

Faculty of Science and Engineering

**Accountable, Explainable Artificial Intelligence Incorporation
Framework for a Real-Time Affective State Assessment Module**

Jordan Joshua Vice

0000-0002-3951-1188

**This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University**

November 2022

Declaration

I Jordan Joshua Vice, declare that this PhD thesis entitled *Accountable, Explainable Artificial Intelligence Incorporation Framework for a Real-Time Affective State Assessment Module* contains no material that has been used prior for the award of any other academic degree or diploma to the best of my knowledge and belief. Except where otherwise stated, this thesis is my own work.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

This work has resulted in the publication and/or presentation of five research papers entitled: (i) Multimodal models for contextual affect assessment in real-time, (ii) Dynamic Hybrid Learning for Improving Facial Expression Classifier Reliability, (iii) Adaptive Classification of Occluded Facial Expressions of Affective States, (iv) Toward Accountable and Explainable Artificial Intelligence Part One: Theory and Examples and (v) Toward Accountable and Explainable Artificial Intelligence Part Two: The Framework Implementation.

Further contents of this work will continue to be disseminated through more research papers in the coming months.

Acknowledgements

I would like to pay my respect to the Aboriginal and Torres Strait Islander members of our community by acknowledging the traditional owners of the land on which the Bentley Campus is located, the Whadjuk people of the Nyungar Nation.

I would also like to acknowledge the School of Civil and Mechanical Engineering and the Mechatronic Engineering faculty for their assistance and time throughout this research project, as well as the entirety of my undergraduate and post-graduate studies.

I would like to especially thank Dr. Masood Khan for his guidance throughout my PhD and for being a great mentor and consummate professional throughout my academic journey.

And finally, to my family, friends and those who have supported me throughout my research. I am forever grateful for your patience, kind words, and encouragement in good times and bad. I would not be in this position without you.

Data and Hardware Storage Declaration

Being an embedded system research project means that this work is comprised of three major elements with storage considerations i.e.:

- The physical device – LattePanda 4G/64GB development board encased in its enclosure
- SAM's operational software
- All data, reports, drafts, videos, and documentation that was written up throughout the research from inception to implementation

Considering the shared IP agreement with Curtin University, two embedded systems have been developed. One will be maintained and stored at Curtin University – including all relevant software and hardware components, and the second will be stored by the author (Jordan Vice).

The software bill of materials has been outlined in Chapter 5 and the appendices of this work, outlining what packages (and versions) were used throughout the construction of SAM and throughout the entirety of this research project. SAM's operational software will be stored on Curtin's Research drive (R-Drive) in both executable and packaged formats, allowing for it to be used on any capable machine and allowing the code, models, databases, and other contents to be scrutinised.

Six video demonstrations have been recorded to showcase some of the outcomes of this research, all of which have been stored digitally and locally on an external hard drive, the six videos demonstrate the following:

- Usage of the home screen included navigation to other windows and monitoring error states
- 2-way communication displaying a conversation that is had with SAM
- Manual interaction with SAM where commands are issued directly using a drop-down menu
- Usage of the settings window and how users/subjects can be created and viewed
- SAM being used as an embedded system with the software being launched from the portable LattePanda device (two videos).

All data, reports, drafts, demonstration videos and documentation will be stored on Curtin's R-Drive and will be accessible by (as per 06/10/2022):

- Jordan Vice (jordan.vice@curtin.edu.au),
- Dr. Masood Khan (masood.khan@curtin.edu.au),
- Dr. Iain Murry (iain.murray@curtin.edu.au),
- Prof. Tele Tan (t.tan@curtin.edu.au).

All data (including SAM's operational software) will also be stored physically on an external hard drive, maintained by the author of this work.

For access to the data please contact Jordan Vice or Dr. Masood Khan using the emails above and clearly stating your intent and/or need for the data.

Abstract

The technological strides we have taken over the past few decades has resulted in vast improvements to machine learning (ML) and artificial intelligence (AI) systems. Increases in computational power and performance has resulted in a growing reliance and the widespread adoption of AI and ML infrastructures across various industries. However, there is no dedicated, quantifiable way of assessing AI system transparency, performances, and accountability.

But as these systems begin to be used in more critical assessment tasks, the need for transparency, comprehensibility and accountability is becoming more apparent as a result. The goal of AI system developers should be to move away from traditional black-box approaches and aim to incorporate comprehensible and accountable elements into AI system design in addition to maintaining high levels of classification performances. Incorporating such principles in an AI system would be a step toward addressing the needs of all stakeholders from end-users to management, to regulatory bodies.

In general, current state-of-the-art affective state assessment (ASA) systems fail in their attempts at being transparent, comprehensible and accountable in their design. These systems can also be limited in their application if they are built on the foundations of discrete emotion and affective state models. Ultimately, there is a large room for improvement in regard to ASA systems in regard to back-end and front-end development.

This work proposes various contributions that can be described under two larger umbrellas of work. Firstly, an accountable, explainable artificial intelligence (AXAI) capability framework has been designed as a tool to be used in developing more transparent, explainable AI systems as well as serving as a means of assessing and comparing AI/ML systems post-hoc. The AXAI capability framework allows for the delineation and assessment of AI systems on the basis of: (i) comprehensibility, based on inspection time, predicate recognition time and predicate naming time, (ii) system accountability, based on the inspect-ability of input cues, processed data, and output cues and (iii) predictive accuracy, measuring the classification and traditional performance of the system through the ratio of test/training data, the training data size and the number of occurrences of false-positive results.

Through this research, it is evident that the AXAI capability framework can be used throughout the software design and development process of an AI system. Using the framework metrics as a template, it is possible to develop a system that is highly accurate, comprehensible and has degrees of accountability, with the latter being important when considering the legislative and regulatory gaps that need to be addressed in AI and machine learning literature and the adoption of AI and ML systems in critical and/or high-risk applications.

The AXAI capability framework was used as a foundation for developing SAM – the State Assessment Module. SAM is a multimodal, real-time ASA system and intelligent agent that exists as a low-cost, portable, embedded system. Whereas most ASA solutions in literature focus on the discrete assessment of affective states, SAM is capable of understanding and accounting for the dynamics of human expressions when analysing facial expressions and speech.

Through a hierarchical classifier ensemble consisting of an array of 43 unique binary and ternary classifiers, SAM is capable of predicting 116 unique states spread across facial expression and speech analysis sub-processes, including the ability to detect the presence of visual occlusions. A high

classification accuracy is reported, showing average validation accuracies of accuracies of 83.07% and 95.90% for facial expression and speech classifiers respectively (21 classifiers each) and approximately 100% for the visual occlusion classification.

The decision to construct a hierarchical classifier ensemble was key in designing a dynamic, continuous assessment system that accounts for the complex, temporal nature of human expressions. Delegating classification tasks to an array of sub-classifiers improved classification performances and allowed for a high degree of system accountability, since SAM's decision-making processes were made transparent from input to output, including its intermediate stages.

Resulting from an iterative design process, an intuitive graphical user interface (GUI) was developed and embedded as part of SAM's architecture. The GUI and interactive elements that SAM possesses, evidence a high degree of comprehensibility and system accountability. SAM can be considered as an intelligent agent and is equipped with speech-to-text and text-to-speech capabilities as well as a working memory that allows for a more advanced level of communication.

Through the AXAI capability framework and SAM, this work is indicative of a positive trajectory for AI and ML system design and proves that black-box machine learning approaches must be considered as a thing of the past.

Contents

i.	Nomenclature	9
ii.	Equations and Formulae	10
iii.	List of Figures	12
iv.	List of Tables	14
Chapter 1:	Introduction	15
1.1	Mission Statement	15
1.2	Objectives.....	16
1.3	Research Gaps.....	17
1.3.1	General AI- and ML-related gaps	18
1.3.2	ASA-related gaps.....	21
1.4	Contributions	23
1.4.1	The AXAI Capability Framework.....	23
1.4.2	Blind Source Separation for Group Sentiment Analysis.....	24
1.4.3	Design and Implementation of a Dynamic, Multimodal, State Assessment System .	24
1.4.4	Occluded Facial Expression Detection and Classification	24
1.4.5	Hybrid Learning for the Development of Hierarchical Classification Models for ASA	24
1.4.6	Multi-level, State-based Reasoning for Speech, and Facial Expression Classifiers	25
1.4.7	Real-time, AXAI-Capable Assessment and Interaction through SAM	25
1.4.8	Integration of SAM into Existing Robotic and Surveillance Infrastructures	25
1.5	Chapter Description and Summary.....	26
Chapter 2:	Background Theory	27
2.1	Foundational Machine and Deep Learning Theory.....	27
2.1.1	Unsupervised Learning – Blind Source Separation	28
2.1.2	Unsupervised Learning – Clustering	31
2.1.3	Supervised Learning.....	33
2.1.4	Hybrid Learning.....	36
2.2	Artificial Intelligence (AI) Systems and Robotic Assistants	37
2.3	Explainable AI, Reasoning and Transparency	38
2.3.1	Explainable AI Applied for Supervised ML Methods.....	40
2.3.2	Explainable AI Applied for Neural Networks.....	40
2.4	Emotions and the Concept of Affect.....	42
2.5	Affective Computing	45

Chapter 3: The Accountable, Explainable AI (AXAI) Framework 50

 3.1 A Method for Improving Explainable AI..... 50

 3.2 Comprehensibility 52

 3.3 Predictive Accuracy..... 53

 3.4 Accountability 55

 3.5 Design and Implementation of the AXAI Framework 56

Chapter 4: AXAI-Focused Software Design and Development 63

 4.1 Software Design and Development Principles..... 63

 4.2 AXAI-Focused Software Design 65

 4.3 Software Design Process of the State Assessment Module 66

Chapter 5: Introducing SAM, the State Assessment Module..... 74

 5.1 Classifier Design and Implementation 74

 5.1.1 Facial Expression Classifier Subsystem 76

 5.1.2 Affective Speech Classifier Subsystem 88

 5.1.3 Facial Occlusion Classifier Subsystem 99

 5.2 Implementing SAM’s Graphical User Interface and Frontend Design 103

 5.2.1 Graphical User Interface Purpose 103

 5.2.2 Home Screen – GUI (HS) 109

 5.2.3 Facial Expression Analysis Windows – GUI (FX) 115

 5.2.4 Affective Speech Analysis Windows – GUI (SX) 121

 5.2.5 Settings Window – GUI (SET) 126

 5.2.6 SAM Interaction Window – GUI (INT) 131

 5.3 Integrating SAM as a Modular, Embedded System 142

 5.3.1 Key Challenges Faced 148

 5.4 Assessing SAM using the AXAI Framework..... 149

Chapter 6: Parallel Research and Other Contributions 152

 6.1 Blind Source Separation and Crowd Sentiment Analysis..... 152

 6.2 Dynamic Micro-expression Detection using Hierarchical Clustering..... 155

Chapter 7: Conclusion..... 167

References 170

Appendices 180

 Appendix A: SAM Version 1 Manifest File 180

 Appendix B: SAM Version 2 Manifest File 181

Appendix C: SAM Version 3 Manifest File 182

Appendix D: SAM Version 4 Manifest File 183

Appendix E: SAM Version 5 Manifest File 184

Appendix F: Home Screen Interaction Analytics Report Example 185

Appendix G: SAM Facial Expression Level 1 Generated Report 189

Appendix H: SAM Facial Expression Level 2.1 Generated Report 192

Appendix I: SAM Facial Expression Level 2.2 Generated Report 196

Appendix J: SAM Facial Expression Level 3.1 Generated Report 201

Appendix K: SAM Facial Expression Level 3.2 Generated Report 206

Appendix L: SAM Facial Expression Level 4 Generated Report 210

Appendix M: Collated SAM Facial Expression classifier dialogue responses for all windows. 217

Appendix N: SAM Affective Speech Assessment Level 1 Generated Report 222

Appendix O: SAM Affective Speech Assessment Level 2 Generated Report 224

Appendix P: SAM Affective Speech Assessment Level 3 Generated Report 227

Appendix Q: SAM Affective Speech Assessment Level 4 Generated Report 231

Appendix R: Collated SAM Affective speech classifier dialogue responses for all windows.. 238

Appendix S: Full software bill of materials produced via command: ‘pip list’ 243

Corresponding Publications 249

i. Nomenclature

Variable	Definition	First Occurrence (Chapter-Section)	
$\mathbf{x}(t)$	Observed signal mixture (BSS)	Chapter 2 – Section 2.1.1	
\mathbf{A}	Mixing matrix (BSS)		
$\mathbf{s}(t)$	Time-domain source signals (BSS)		
$\mathbf{b}(t)$	Defines any noise present in a signal mixture (BSS)		
\mathbf{W}	Demixing matrix (BSS)		
$\mathbf{y}(t)$	Estimated separated signals (BSS)		
T_k	Defines the linear operator when deploying DTCWT for separation		
$\psi_i(t)$	Real component of a wavelet transform (DTCWT).		
$\psi_j(t)$	Imaginary component of a wavelet transform (DTCWT).		
$\mathbf{x}_i = \{x_1, x_2, \dots, x_m\}$	Singular pattern or feature vector		Chapter 2 – Section 2.1.2
$\mathbf{X} = \{x_1, x_2, \dots, x_N\}$	Set of feature vectors of size $N \times m$		
$\mathbf{C} = \{c_1, c_2, \dots, c_K\}$	Defines the 'K' class labels for patterns in \mathbf{X} .		
k_p	Number of peaks/partitions in a continuous sample		
$K_{classes}$	Number of classes extracted derived from k_p	Chapter 2 – Section 2.1.3	
\bar{x}_k	k^{th} cluster centroid in an LDA problem or example.		
N_{LD}	Number of linear discriminants (LDA)		
\mathbb{P}	Definite program	Chapter 3 – Section 3.1	
\mathbb{k}	Constant		
\mathbb{S}	Human population		
\mathbb{V}	First-order variable		
\mathbb{B}	Background knowledge		
\mathbb{D}	Domain		
\mathbb{P}	Predicate symbol		
$\mathbb{A}ll_{AX}$	Accountable, explainable artificial intelligence (AXAI) capability		
\mathbb{C}	Comprehensibility		
P_A	Predictive Accuracy		
S_A	System Accountability		
T_{it}	Inspection time		Chapter 3 – Section 3.2
T_{pr}	Predicate recognition time		
T_{pn}	Predicate naming time		
$r_{tst-trn}$	Ratio of model test data size and training data size	Chapter 3 – Section 3.3	
d_{trn}	Training data size		
O_{fp}	Number of occurrences of false-positive results		
I_{in}	Inspect-ability of input models or cues	Chapter 3 – Section 3.4	
I_{pro}	Inspect-ability of processed intermediate data		
I_{out}	Inspect-ability of output models or cues		
LD_n	Describes the n^{th} linear discriminant	Chapter 6 – Section 6.2	
$LD_{n.m}$	Describes macro- and micro-level linear discriminants		

ii. Equations and Formulae

Equation	Definition/Purpose	First Occurrence (Chapter-Section)
$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{b}(t)$	BSS governing equation	Chapter 2 – Section 2.1
$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t) = \mathbf{W}\mathbf{A}\mathbf{s}(t)$	BSS solution to estimate separated signals	
$\mathbf{W}^+ = \mathbf{W} + \Gamma[\text{diag}(-\beta_i) + E\{g(\mathbf{y})\mathbf{y}^T\}]\mathbf{W}$	Fast-ICA learning algorithm	
$\mathbf{W}^+ = \mathbf{W} + \mu[\mathbf{I} + g(\mathbf{y})\mathbf{y}^T]\mathbf{W}$	Maximum-likelihood solution for the Fast-ICA algorithm	
$\mathbf{s}_k(t) = \mathbf{T}_k[\mathbf{s}(t)]$	Describes the linear operator in a DTCWT application	
$\mathbf{x}_k(t) = \mathbf{T}_k[\mathbf{A}\mathbf{s}(t)]$	How the DTCWT technique is deployed in a BSS problem	
$\boldsymbol{\psi}(t) = \boldsymbol{\psi}_i(t) + \boldsymbol{\psi}_j(t)$	Representation of the wavelet transform showing real and complex components	
$D_{Euclidean} = \sqrt{\sum_{i=1}^n x_i - y_i ^2}$	Euclidean distance-based similarity	Chapter 2 – Section 2.1
$D_{Manhattan} = \sum_{i=1}^n x_i - y_i $	Manhattan distance-based similarity	
$D_{Mahalanobis} = \sqrt{(\mathbf{x}_i - \mathbf{y}_i)\mathbf{S}^{-1}(\mathbf{x}_i - \mathbf{y}_i)^T}$	Mahalanobis distance-based similarity	
$H[X_1 X_2] = -\sum_{ij} p(X_1 = x_i, X_2 = x_j) \times \log(p(X_1 = x_i, X_2 = x_j))$	Conditional entropy	
$I[X_1; X_2] = \sum_{ij} p(X_1 = x_i, X_2 = x_j) \times \log\left(\frac{p(X_1 = x_i, X_2 = x_j)}{p(X_1 = x_i) \cdot p(X_2 = x_j)}\right)$	Mutual information score	
$S_{cos\theta} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{ \mathbf{x}_i \mathbf{x}_j }$	Cosine similarity	
$\Delta\mathbf{S} = \mathbf{S}(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{S}(\mathbf{x}_i, \mathbf{x}_{i+1})$	Dissimilarity between two sequential pairs	
$f(\mathbf{x}_i) = \Delta\mathbf{S} : \mathbf{x}_i \in \mathbf{X}$ and $\mathbf{x}_{i+1} \leq \mathbf{x}_N$	Represents the dissimilarity across a continuous sample	
$K_{classes} = k_p + 1$	Derivation of the number of classes given a number of partitions that a continuous sample is split into (k_p)	
$\frac{\mathbf{b}'\mathbf{B}\mathbf{b}}{\mathbf{b}'\mathbf{W}\mathbf{b}}$	Linear discriminant analysis' (LDA) basic optimisation problem	
$ \mathbf{b}'(\mathbf{x}_i - \bar{\mathbf{x}}_1) - \mathbf{b}'(\mathbf{x}_i - \bar{\mathbf{x}}_2) - \mathbf{b}'(\mathbf{x}_i - \bar{\mathbf{x}}_k) - \dots - \mathbf{b}'(\mathbf{x}_i - \bar{\mathbf{x}}_{K_{classes}}) < 0$	The application of LDA when used for classification tasks	
$y(x) = \text{sign}\left[\sum_{i=1}^N \alpha_i y_i \psi(x, \mathbf{x}_i) + \mathbf{b}\right]$	Definition of the Support Vector Machine classifier – used to construct hyperplane boundaries	
$\boldsymbol{\psi}_{linear}(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}_i^T \mathbf{x}$	Linear SVM Kernel operating term	

$\psi_{RBF}(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\ \mathbf{x} - \mathbf{x}_i\ ^2}{2\sigma^2}\right)$	Radial Basis Function (RBF) SVM Kernel operating term	
$\mathbf{h}_i = \sigma\left(\sum_{j=1}^N W_{ij}x_j + T_{i(hid)}\right)$	Governing equation for defining a neural network node	
$\sigma_{sigmoid}(x) = \frac{1}{1 + e^{-x}}$	Sigmoid activation function	
$\sigma_{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	Hyperbolic tangent activation function	
$\sigma_{ReLU}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$	Rectified linear unit (ReLU) activation function	
$\sigma_{binary}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	Binary activation function	
$A = \sqrt{a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}}$	Three-dimensional vector decomposition	Chapter 3 – Section 3.1
$C(\mathbb{S}, \mathbb{P}) = \sqrt{(T_{it}^2 + T_{pr}^2 + T_{pn}^2)}$	AXAI comprehensibility vector representation	Chapter 3 – Section 3.2
$P_A(\mathbb{S}, \mathbb{P}) = \sqrt{r_{tst-trn}^2 + d_{trn}^2 + O_{fp}^2}$	AXAI predictive accuracy vector representation	Chapter 3 – Section 3.3
$S_A(\mathbb{S}, \mathbb{P}) = \sqrt{I_{in}^2 + I_{pro}^2 + I_{out}^2}$	AXAI system accountability vector representation	Chapter 3 – Section 3.4
$Acc_{val}(\%) = \frac{N_{correct}}{N_{total}} \times 100\%$	Validation accuracy	
$Z_x = \frac{x - \mu_x}{\sigma_x}$	Standard scaler equation	
$\gamma = \frac{1}{N_{Dim} \times X_{variance}}$	SVM Gamma parameter formula	Chapter 5 – Section 5.1
$CV(\hat{f}) = \frac{1}{K} \sum_{i=1}^K L(y_i, \hat{f}^{-k(i)}(x_i))$	k-fold cross validation accuracy	
$D_r = \frac{2 \cdot \{\mathcal{S}_1\} \cap \{\mathcal{S}_2\} }{ \mathcal{S}_1 + \mathcal{S}_2 }$	Sequence matcher equation based on the Gestalt pattern matching algorithm	Chapter 5 – Section 5.2
$SDR = 10 \log_{10} \left(\frac{\ e_{target}\ ^2}{\ e_{interf} + e_{artif}\ ^2} \right)$	Source-to-distortion (SDR) ratio	
$SIR = 10 \log_{10} \left(\frac{\ e_{target}\ ^2}{\ e_{interf}\ ^2} \right)$	Source-to-interference (SIR) ratio	Chapter 6 – Section 6.1
$SAR = 10 \log_{10} \left(\frac{\ e_{target} + e_{interf}\ ^2}{\ e_{artif}\ ^2} \right)$	Source-to-artifacts (SAR) ratio	

iii. List of Figures

	Page
Figure 1	Visualisation of research thought process17
Figure 2	Simple representation of a black-box model18
Figure 3	Extension of Figure showing a basic example of a black-box approach20
Figure 4	Supervised and Unsupervised learning techniques and applications27
Figure 5	Visualisation of the circumplex and three-factor theory models43
Figure 6	The Plutchik Spectrum44
Figure 7	Comparison of surveyed ASA systems in literature47
Figure 8	Breakdown/Summary of the AXAI capability framework components51
Figure 9	Visual representation of the AXAI framework57
Figure 10	AXAI-capability comparison of ASAM and DAASAS (bar graph)61
Figure 11	AXAI-capability comparison of ASAM and DAASAS (graphical)62
Figure 12	Traditional waterfall model65
Figure 13	Incremental model65
Figure 14	Standard agile model65
Figure 15	SAM’s development timeline67
Figure 16	SAM package size and complexity over iterations72
Figure 17	Visualisation of SAM complexity w.r.t number of lines of code72
Figure 18	Cosine similarity-based clustering algorithm for facial image data77
Figure 19	Cosine similarity heat map for a continuous facial image sample78
Figure 20	Visualisation of temporal phase regions in a continuous sample78
Figure 21	An expansion of Figure 20, showing micro-level states79
Figure 22	Dissimilarity measurements for CK+ and RAVDESS datasets79
Figure 23	Four-tier hierarchical classifier ensemble used for facial assessment81
Figure 24	Graphical representation of facial expression state groups81
Figure 25	Visualisation of facial expression LDA-transformed clusters83
Figure 26	Example of how further micro-expressions could be extracted84
Figure 27	SAM facial expression level 1 model training architecture85
Figure 28	SAM facial expression level 2, 3 and 4 model training architecture86
Figure 29	Comparison of SAM facial expression classifier performances87
Figure 30	Cosine similarity-based clustering algorithm for speech data89
Figure 31	Cosine similarity heat map for a continuous speech sample (1)90
Figure 32	Cosine similarity heat map for a continuous speech sample (2)91
Figure 33	Four-tier hierarchical classifier ensemble used for speech assessment92
Figure 34	Visualisation of affective speech LDA-transformed clusters94
Figure 35	High-level summary of the affective speech classifier subsystem95
Figure 36	SAM affective speech level 1 model training architecture96
Figure 37	SAM affective speech level 2, 3 and 4 model training architecture97
Figure 38	Comparison of SAM affective speech classifier performances98
Figure 39	SAM Home screen GUI screenshot evidencing occlusion detection100
Figure 40	Visualisation of how “masked” and “no mask” samples are clustered101
Figure 41	SAM occlusion detection classifier training architecture102

Figure 42	Summary of SAM multimodal classification network102
Figure 43	How databases are parsed into SAM103
Figure 44	Extension of Figure 42, visualising the parsing of data throughout SAM104
Figure 45	SAM package structure and tree diagram107
Figure 46	Annotated screenshots of SAM's home screen GUI and its various states108
Figure 47	SAM home screen GUI initialisation function block111
Figure 48	SAM speech-to-text function block112
Figure 49	SAM main-loop function block112
Figure 50	SAM home screen event handler function block113
Figure 51	SAM facial recognition function block113
Figure 52	SAM occlusion/mask detection function block114
Figure 53	SAM speech analysis function block114
Figure 54	SAM home screen button state handler function block115
Figure 55	Example parallel viewing of SAM facial expression GUI windows116
Figure 56	Collection of SAM facial expression GUI window layouts117
Figure 57	SAM facial expression GUI button handling function block (Part 1)118
Figure 58	SAM facial expression GUI button handling function block (Part 2)119
Figure 59	Example parallel viewing of SAM affective speech GUI windows121
Figure 60	Collection of SAM affective speech GUI window layouts122
Figure 61	SAM affective speech GUI button handling function block (Part 1)123
Figure 62	SAM affective speech GUI button handling function block (Part 2)124
Figure 63	Visualisation of SAM's 'user' and 'subject' relationships126
Figure 64	SAM settings window and corresponding child windows128
Figure 65	SAM settings window event handling function block129
Figure 66	SAM user and subject creation event handling function block130
Figure 67	SAM user and subject login window event handling function block131
Figure 68	SAM user and subject database window event handling function block131
Figure 69	SAM interaction GUI window layout133
Figure 70	SAM interaction window event handling function block139
Figure 71	SAM communication event handling function block140
Figure 72	SAM shutdown sequence and temporal length function blocks141
Figure 73	SAM exposed internals including the LattePanda board and peripherals143
Figure 74	Front, Top, Side, and Isometric view of SAM as a hardware solution143
Figure 75	Example of SAM being integrated into an existing system144
Figure 76	Visualisation of how SAM as an embedded solution146
Figure 77	AXAI capability comparison of DAASAS, ASAM and SAM systems151
Figure 78	Blind source separation and crowd sentiment analysis flowchart152
Figure 79	CK+ dataset LDA clustering results157
Figure 80	CK+ cluster centroid visualisation158
Figure 81	Sadness-happiness and anger-surprise axes158
Figure 82	Sadness-happiness state-to-state transitions159
Figure 83	Anger-surprise state-to-state transitions160
Figure 84	State-to-state transitions showing CK+ dataset samples162
Figure 85	Upper facial region two-dimensional linear discriminant projection164

Figure 86 Lower facial region two-dimensional linear discriminant projection164
Figure 87 Comparison of full-, upper- and lower-facial expression classifiers166

iv. List of Tables

	Page
Table 1 List of Convolutional Neural Network (CNN) architectures36
Table 2 List of facial action units associated with common expressions45
Table 3 List of vocal changes associated with common expressions47
Table 4 Survey of ASA systems in literature48
Table 5 Collection of AXAI capability framework parameters57
Table 6 AXAI capability scoring criteria58
Table 7 ASAM assessor scoring table59
Table 8 DAASAS assessor scoring table59
Table 9 Comparison of ASAM and DAASAS predictive accuracies59
Table 10 Summary and description of SAM’s iterative development process68
Table 11 List of commonly used datasets in ASA literature75
Table 12 Summary of SAM’s classifiable states at level 476
Table 13 RAVDESS facial expression class data distribution86
Table 14 SAM facial expression classifier information and performances87
Table 15 SAM affective speech classifier information and performances98
Table 16 Summary of SAM’s GUI functionality for each GUI window105
Table 17 Examples of SAM’s facial expression assessment dialogue outputs120
Table 18 Examples of SAM’s affective speech assessment dialogue outputs125
Table 19 List and information relative to SAM’s interaction commands134
Table 20 SAM interaction dialogue prompt 1 – extended dialogue outputs136
Table 21 SAM interaction dialogue prompt 3 – extended dialogue outputs136
Table 22 SAM hardware bill of materials145
Table 23 SAM software bill of materials147
Table 24 SAM assessor scoring table150
Table 25 Crowd sentiment analysis performance comparison154
Table 26 State-to-state comparison showing differences in LD_n values160

Chapter 1: Introduction

1.1 Mission Statement

The primary goals of this research are to: (i) develop a framework that would be for the design and assessment of ML and AI systems, in an attempt to prove that better, more transparent systems can be achieved, (ii) design an embedded, real-time affective state assessment system that is capable of dynamic, continuous assessment of states, and would be equipped with advanced interaction functions and the ability to explain its decision-making and reasoning, serving as proof of accountable, explainable artificial intelligence, applied within the context of affective computing.

Through its discussion of accountability and comprehensibility, this research will address the issue of trust in AI Systems that is held by the general public as has been discussed in the relevant literature [2-9, 11-13] and has been a consistent topic of conversation as AI systems have become more prevalent in daily life. However, while explainable AI (XAI) is considered a solution, there are still some gaps that these solutions fail to address as explained in [11, 14, 15], including the issue that there is no clear or standardised method of incorporating explainability into AI/ML system design processes.

There is a defined connection between XAI and reasoning w.r.t decision-making processes. However, a large issue is that AI systems are known for being opaque and unable to explain their decision-making processes [7, 16]. Thus, when we consider XAI and the issues of trust, we are led to the realm of legislation and regulation, a place where XAI literature lacks when attempting to deconvolute concepts of accountability and comprehensibility. In order to construct regulations and standards, all stakeholders must be listened to and accounted for in regard to AI system design and development. In fact, some strides have been taken to address AI system regulation as proposed in [6, 17, 18].

When discussing affective state assessment (ASA) and affective computing, the applications and literature body are both extensive. This research aims to address research gaps pertaining to the fact that there is a lack of real-time Accountable explainable AI (AXAI) or general XAI-capable ASA systems. In addition, there is also a lack of continuous assessment solutions, systems that account for occluded facial expressions, and no presence of blind source separation (BSS) being deployed for ASA tasks.

Regarding real-time ASA systems, there are examples in literature as shown in [19-23]. The novelty of this work is not in developing a real-time system, but in developing an ASA system that is capable of explaining its decision-making processes and incorporating elements of AXAI into its system design. While there are various continuous and discrete emotion models in literature, it is important to consider the dynamics of human expressions and the need to design continuous solutions and currently, discrete models are more prevalent in literature [22, 24]. The societal effects of the COVID-19 pandemic resulted in an increase in the use of PPE like face masks to mitigate the spread of diseases and viruses, the result was that facial expression recognition systems and ASA systems that rely on facial image data would become adversely affected or rendered obsolete as reported in [25]. Furthermore, while BSS is a very common tool for separating statistically dependent and independent signal mixtures but despite its utility, its use in affective state assessment is not evident in literature.

Identifying the above issues and understanding the relevant research gaps resulted in the formulation of eight contributions that are evidenced through this work. The primary goals of this research encompass how these contributions are achieved, with the AXAI capability framework serving as the foundation for addressing the general AI and machine learning-related gaps by creating a way of

assessing and delineating AI systems on the basis of predictive accuracy, accountability, and comprehensibility.

The State Assessment Module (SAM) was designed to prove how AXAI can be incorporated in a real-time, modular ASA system's design and development process and aims to address ASA- and affective computing-related research gaps. Finally, this work contributes toward literature by addressing the issue of occluded facial expression classification and proves how BSS can be used for ASA tasks by deploying the technique for group sentiment analysis.

Overall, this work applies a multi-level, holistic approach at improving human-machine interactions and AI systems in general. Through the development of the AXAI capability framework, this work deconvolutes the concept of 'explainability' and continues the discussion of how developers and law makers can improve trust, accountability, and comprehensibility in AI systems. Through SAM and its intrinsic link with AXAI, the benchmark for real-time ASA and intelligent systems is raised.

1.2 Objectives

A few years prior to my Philosophical Doctorate (PhD) Thesis, I found that my research interests swayed heavily toward the world of artificial intelligence (AI) given the growth of the field of research, especially in the 21st Century. This work is a continuation of my Honours Thesis entitled: Bi-modal Affect-Based Authentication Machines [1], where I explored automated recognition and classification of human affective states, using real-time input signals for the purpose of contextual classification. While my previous work successfully achieved the goals that were set out for it, I found that I barely scratched the surface of not only what is possible for affective state assessment (ASA) systems, but how AI and machine learning (ML) systems in general can be improved on a more global scale.

Through this research, the goal was to focus on addressing both higher-level research gaps, and lower-level research gaps w.r.t to the domains of artificial intelligence and machine/deep learning theory and implementation in the affective computing domain. Hence, the long-term objectives of this research project were split into two umbrellas of work, each containing significant contributions:

1. **High Level:** Design and Implementation of the accountable, explainable AI (AXAI) framework
2. **Low Level:** Development of a real ASA system through SAM – the State Assessment Module

The two major objectives were designed to address some of the research gaps that exist at the higher- and lower-levels of the relevant literature and provide significant contributions to the existing body of work. These contributions were achieved in a top-down approach, first designing, implementing, and proving the AXAI framework prior to commencing development of SAM. Many sub-processes and parallel research efforts were conducted between the two major objectives, but the overall vision was to maintain the synergy that existed between these two objectives. Figure 1 visualises the thought process for this research given this top-down approach, highlighting some of the research gaps, contributions, and the overall flow of information for this work at the highest level.

As discussed, there were significant gaps in the research that needed to be identified, with the AXAI framework and SAM serving as significant tools in which these gaps would be addressed.

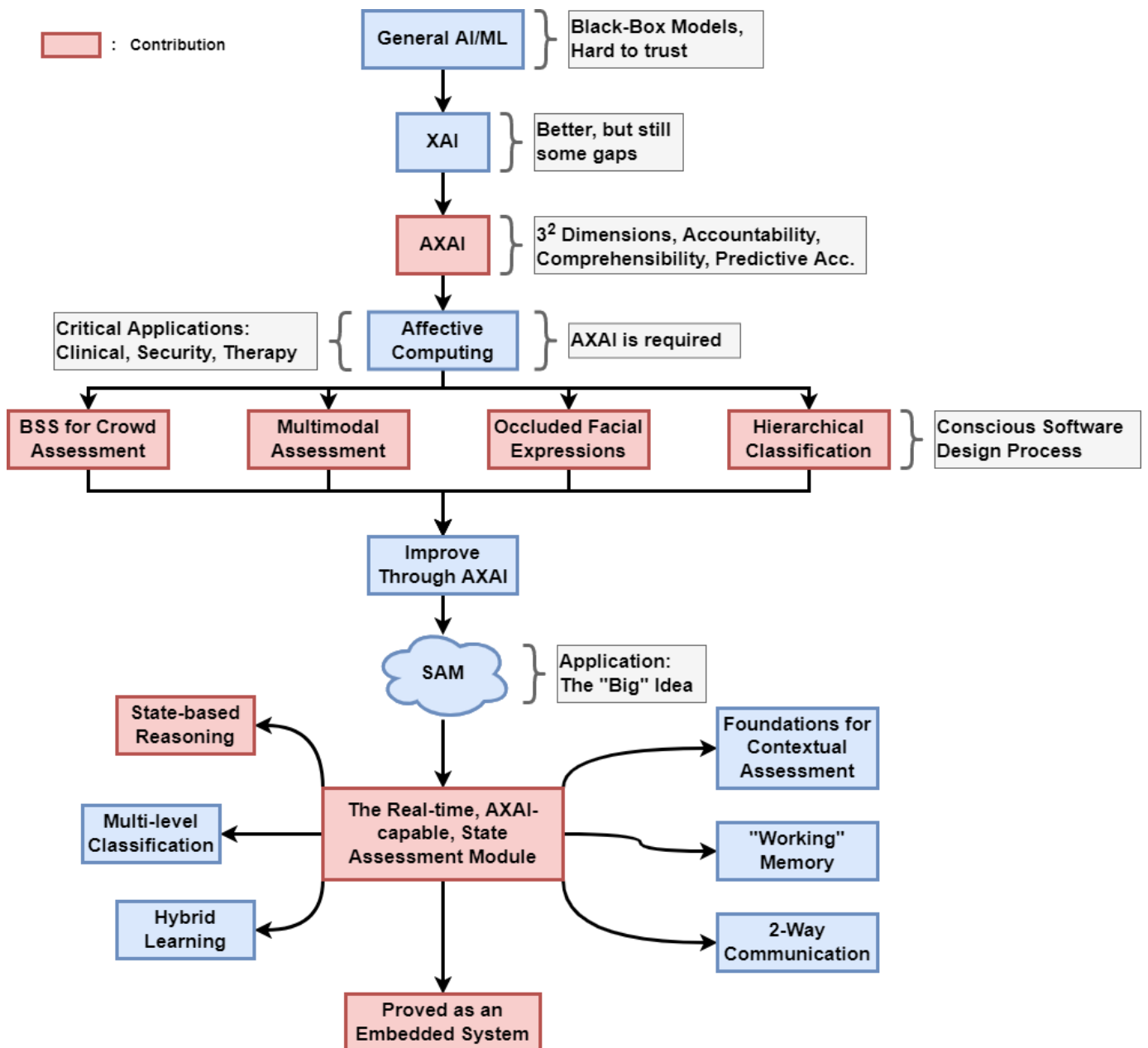


Figure 1: Top-down, visual representation of the thought process for this research, highlighting the evolution of the work from general AI research – to AXAI (main objective 1) – to the development of SAM (main objective 2).

1.3 Research Gaps

This work aims to provide a comprehensive, holistic view on the theory, design, development, and application of AI and ML systems. This is achieved through the development of the AXAI framework and SAM, as visualised in Figure 1. This work aims to discuss the two major objectives both independently, as well as how they are connected and the synergy that exists between them. With this in mind, there are AXAI-related and ASA-related gaps in the research that will be addressed in this section.

1.3.1 General AI- and ML-related gaps

1. The Issue of Trust in AI Systems for the General Public

The issue of “trust” in AI and ML systems has long been a discussion in relevant literature [2-9, 11-13] across various domains and points of view, looking at trust in regard to trust in decision-making processes [2,6-8], legislation and government regulations [4-6], and the trust that local populations would not suffer job loss as a result of the advancement of AI systems [2, 9].

A lack of trust in new technologies is not uncommon, especially given the rapid technological growth since the beginning of the third industrial revolution, aptly described as “The Digital Revolution” [10] which began in the middle of the 20th century and started the societal transition into the information age. With the rapid advancement of technology in the last century, we can excuse the general population for mistrusting technology that they do not understand.

Which introduces the issue of “black-box” ML processes which is described in its simplest form in Figure 2. The question bubbles above each stage in the process are examples of what questions end-users or key stakeholders may have. Looking at the black-box approach in this light outlines why trust in AI systems is present, and why developers and AI researchers need to shift away from opaque, traditional approaches [4, 5].

The role of XAI was to make the decision-making processes of AI systems and systems that use machine and deep learning classifiers more transparent [4-6] and provide users with more “human-understandable” AI systems based on explanation theory and social sciences literature [3, 8]. Predominantly, AI systems that exist in the public domain cannot be considered as examples of explainable AI systems, which highlights the lack of trust and transparency in these systems.

Going forward, the issues of trust and transparency must be addressed when developing alternative high-level structures like the AXAI framework, and elements of these infrastructures and ideologies must be implemented into the software design process for applications, as was the case when developing SAM as part of this research endeavour. Ideologies like XAI serve as a strong foundation for improving these systems but as will be discussed, there is still room for improvement to the way we design ML and artificial intelligence systems beyond ‘explainability’.

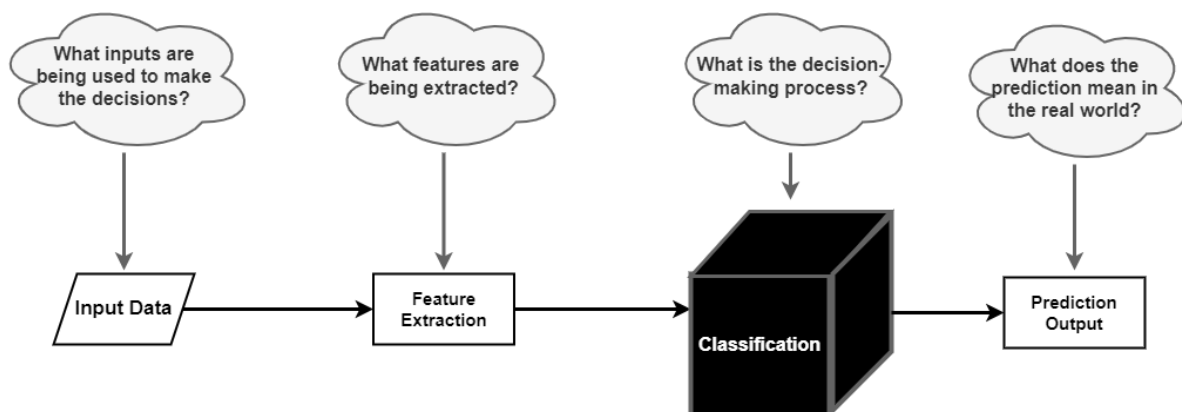


Figure 2: Simple representation of a traditional, black-box ML approach indicating some of the questions (top) that key stakeholders may have at each stage of the process.

Explainable AI Still Contains Gaps

While the argument for XAI is strong relative to traditional black-box approaches. The general definitions of XAI seem to vary and conflate depending on the literature [4, 5, 11, 12]. Generally, judging, and delineating ML systems on the basis of their explainability is done in a two-dimensional plane of comprehensibility and predictive accuracy [11, 13]. Which shows a distinct improvement w.r.t traditional assessments of black-box models that are assessed purely on their predictive accuracy.

Comprehensibility being added as a criterion for assessing ML systems is an improvement [11]. However, this model does not account for factors such as the accountability of ML and AI systems throughout the development process and oftentimes convolutes concepts of accountability and explainability. Given the complex roles that these systems are now deployed in, including: (i) clinical, (ii) legal, (iii) autonomous travel, (iv) security-related tasks, accountability of decision-makers and intelligent agents is an essential requirement [14, 15].

Currently, accountability of ML systems is considered domain-*dependent*, and the assessment of these metrics is done in post-production tasks [15]. This highlights the gap in XAI that needs to be addressed. Rather than developing these systems and determining their accountability in post-production, accountability should be a metric that can be quantitatively assessed at various stages of the development process and should be domain-independent – hence why the AXAI framework discussed in this work addresses this gap by serving as an improvement on XAI.

2. Lack of Reasoning in ML Systems

There is a clear synergy between ML reasoning and explainability, in that a system that is capable of explanatory reasoning would be an example of XAI [3, 5, 8]. However, reasoning itself is not so straight forward, as behavioural sciences literature shows us that there are various types of reasoning [7, 16], as well as various methods in which these systems can display their reasoning [3,4,7].

Analysing Figure 2, we can see that a major issue with traditional, black-box ML systems is a lack of reasoning in regard to decision-making processes and the flow of information from input to output. Most humans are capable of explaining their decision-making and reasoning processes beyond purely primal instincts, highlighting our advanced level of intelligence. For machines to be considered as “human-understandable” and for them to possess advanced levels of intelligence, they need to be capable of understanding their internal logical and/or classification architectures, their feature selection and extraction processes and the flow and processing of information from input to output.

For the example shown in Figure 3, suppose there is a simple, supervised, multi-class ML system with a goal of classifying coloured shapes on a white piece of paper and outputting the non-white colours it has detected on the page. A black-box model would take an input image, process it, and output either a prediction integer value between 0, $N_{classes}$, or the label associated with the prediction. In the case of Figure 3, this black-box model would only output to the user: Red, Blue, Yellow. Adding simple explanatory reasoning to this crude system could involve:

- **Input Data:** Displaying the pre-processed input image back to the user via a graphical user interface (GUI), showing exactly what the system is trying to classify i.e., the input image.

- **Feature Extraction:** Explain to the user that RGB values are extracted for each pixel. Displaying the RGB values of each pixel in a secondary interface panel for each pixel so that a user can see exactly what features are input into the classifier
- **Classification:** Making the user aware of the classes/colours contained in the system and an approximation of the RGB value for each colour as well as descriptions of the classifier.
- **Prediction Output:** “I detect red, blue and yellow on the page”, showing the predicted state and the difference in RGB values between input colour RGB values and the predicted colour RGB values. For a fear of misclassification, prediction scores can also be used to justify explanatory reasoning i.e., “I detect 3 colours on the page, here are my 3 guesses for each...”

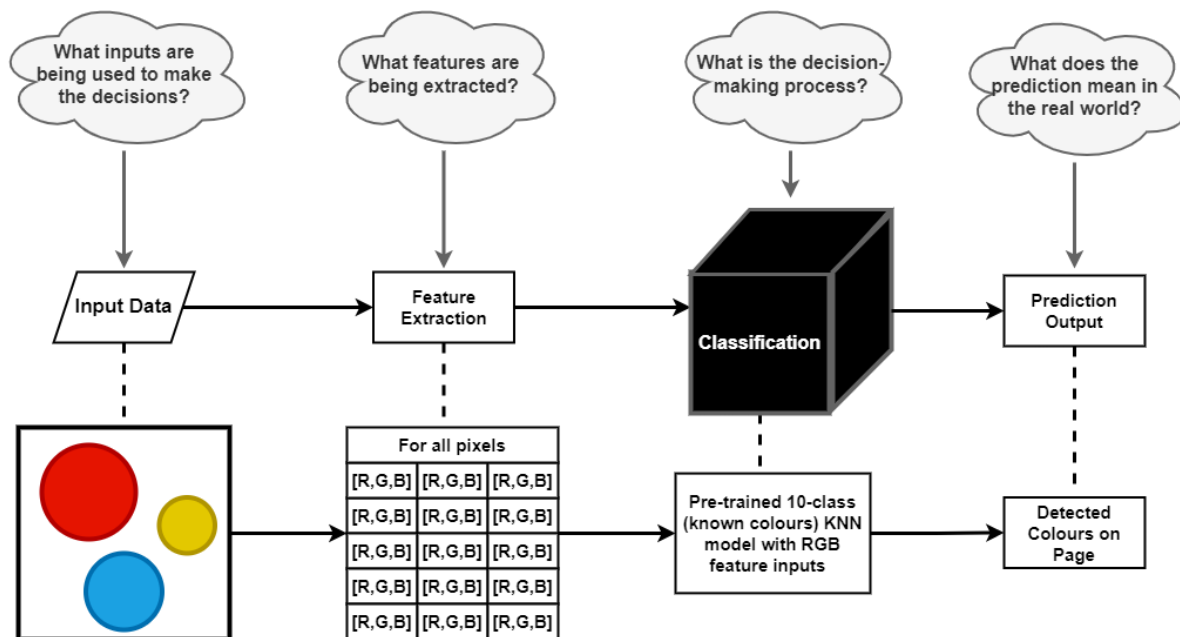


Figure 3: Extension of Figure 2 showing a crude example of a classifier that detects coloured shapes on a page and outputs the colours it has detected

Although the crude example is a simple classification problem, we can see how reasoning behind decision-making processes can improve the user-experience by “walking through” how a prediction was made, and how this example can be scaled up to more complex tasks. Unfortunately, a lot of classification systems do not provide users with reasoning which is why ideas like XAI and AXAI have been theorised. This research gap has been addressed in this research through systems like SAM which has various ways of explaining its decision-making processes, as will be discussed later.

3. AI Regulation and Quantitative Assessment Tools

The issue with assessing accountability as domain-dependent and as a post-production task, means that there is an increased chance that tests are not regulated, or that there is no quantitative measure to compare AI systems once they are released to market. This introduces an issue in regard to the legislation, regulation, and proper testing procedures of AI systems prior to this point.

An example of accountability in artificial intelligence was discussed in the 2019 Algorithmic Accountability Act, introduced by the United States Senate [17], which states that accountability should be derived from data as well as with algorithmic approaches. The aim of the act was to call for

businesses to appraise their AI systems and address the outcomes of predictive machines [17]. However, this is not the only form of legislation and regulation around the world. Liu et al. discuss the case of the General Data Protection Regulation of the European Union [6], which suggests that users of automated processes and decision-making systems should be made aware of meaningful information about the internal system logic, providing a legal precedent for the use of a person's data and how this information is processed [6].

Another case study reported in [18], discusses an AI algorithm deployed by the Home Office UK Visas and Immigration (UKVI) department, which aimed to optimise VISA application services [18]. The UK government upon querying in parliament refused to disclose the logical functions of the assisted and pleaded that the system was working as intended, and that a final decision is made by a human agent [18]. While this case study does not discuss legislation and regulation specifically, it does highlight the importance of accountability and transparency of ML systems when applied in critical scenarios.

The examples discussed above highlight the need for domain-independent assessment of accountability and identifies the gap in XAI approaches that fail to account for accountability as an assessable metric for AI systems. The AXAI framework aims to tackle the issue of accountability throughout the development cycle of AI systems – from inception to release, addressing the current gaps in general AI and ML literature as well as XAI literature.

1.3.2 ASA-related gaps

1. Lack of Real-time AXAI/XAI-capable Affective State Assessment Systems

In my previous research I designed a real-time multimodal system for contextual ASA in real-time [19], with the aim then, to address the lack of real-time XAI-capable ASA systems. While at the time, the intention was to focus on the ASA part of the work, not the XAI capabilities, as concepts of accountability and comprehensibility were not formalised in the design process of this previous work.

There are many examples in literature of real-time ASA systems, which use a variety of input modalities for assessment as surveyed across [20-23]. But as we analyse the methods and applications discussed in these surveys, it is evident that elements of explanatory reasoning, comprehensibility and accountability are lacking. A lot of the flowcharts of these systems evidence black-box models that do not relay the internal reasoning processes to the user, and output predicted emotions, graphical results, or are used to complete a task.

Dissecting some of the applications surveyed in [20], the authors discuss affective interactional systems and focus on these systems aiming for “user-enhanced experiences”, for use in multi-agent systems [20]. However, upon further analysis, these applications could enhance the user experience even further if they adopted XAI and AXAI principles into their design. However, it should be noted that the survey was conducted in 2015.

To analyse and compare this to a more recent survey published in 2020, we can discuss some of the works in [23], which surveys emotion recognition and machine learning systems which use multimodal data. While the survey discusses methodologies, modalities, and feature extraction techniques across various modalities [23]. When discussing applications of these techniques, the reader is inclined to believe that the surveyed works provide model-based predictions, which re-introduces the black-box ML problem discussed prior.

The lack of real-time ASA systems with XAI or AXAI capabilities embedded within them is evident across many surveyed works, highlighting an identifiable gap in the literature. For complex, critical applications like the autonomous vehicle applications surveyed in [22] that sense driver emotions, it is imperative that the driver, lawmakers, and other key stakeholders are aware of why certain decisions would be made by a vehicle when analysing a driver's state.

2. Lack of Dynamic, Continuous Assessment Solutions

As will be discussed in more detail in Chapter 2, the concept of affective states and ASA solutions are based on either discrete, or continuous models. Due to a large number of datasets being discretely labelled as highlighted in [22], a lot of the corresponding ASA solutions are discrete classification solutions, modelling a small number of ' n ' affective state/emotion classes [22, 24].

The complexity of the human experience, and the way we express our internal thoughts, intents, and feelings, make continuous models much more representative of the unique nature of affective states. This is not to discount discrete solutions, but instead queries the applicability of these solutions in real-world scenarios where a singular description of the state does not consider global and local contexts and a wider population.

Continuous assessment solutions that consider a larger array of states, temporal information, or spectrum-based representations would be more capable of accounting for the dynamics of human affective states and would be more accepted for real-world applications.

The lack of continuous ASA solutions that account for the dynamics of human experiences highlights a significant gap in the literature that this thesis and SAM aims to address. This research gap has been addressed and reported by Vice et al. in [24], where the authors discuss the development and implementation of a continuous assessment tool that accounts for the dynamics of human expressions, with this research serving as the classification architecture foundations of SAM, which is detailed in later sections of this work.

3. Facial Expression Recognition Systems Cannot Adapt to Occluded Faces

In light of the recent COVID-19 pandemic, major societal changes occurred around the world. Throughout, the adoption of masks, personal protective equipment (PPE) and proper hygiene protocols assisted in mitigating the spread of viruses and diseases [25, 26]. While the social acceptance of masks and PPE were already normalised in some parts of the world [27], the acceptance of mask-wearing in public spaces grew as a result of people wanting to remain safe throughout troubling times.

Facial recognition and facial *expression* recognition systems that had been deployed prior to 2020 would have been severely impacted by the rise in occluded faces in public spaces and as a result, the deployed classifiers would have been adversely affected. However, as stated in the corresponding paper by Vice et al. that addresses this research gap [25], the authors and developers of assessment systems would not have had the foresight to design these systems to account for occluded expressions.

Therefore, systems that are currently in the process of development that rely on facial data for analysis should be equipped with an occlusion detection subsystem that accounts for changes and adapts the

internal logical flows to classify upper-, or full-facial expressions, depending on if an occlusion like a protective mask is detected by the input device.

While this gap in this research may have only arisen (globally) in recent years as a result of the current global situation. As expressed in [27], in some parts of the world, the use of personal protective equipment has been socially acceptable for a much longer period of time. Thus, addressing this gap, would have been beneficial both in the past, as well as currently. The research presented in [25] addresses this research gap, and the methodology is discussed in detail in subsequent sections.

4. Blind Source Separation Applied for Affective State Assessment

Blind Source Separation (BSS) is a common signal analysis tool that is used to separate statistically dependent/independent signals from a mixture to retrieve approximations of their source signals. The approach is used across various industries, particularly in the music industry where the technique is deployed to separate vocals from instrumentals [28], as well as being used to improve automatic speech recognition systems by removing background noise and background music [29].

Throughout an extensive literature review, no evidence could be found regarding the use of BSS within the realms of ASA. However, the applications for implementing such a system would be beneficial in public safety, entertainment, security, and commercial sectors. The gap in the literature is evidenced by the *lack* of literature regarding BSS being used in an ASA or emotion recognition scenario. In later chapters, the implementation of this technique within the realms of ASA is discussed, in a means of addressing the research gap, and contributing through a novel, crowd sentiment analysis system.

The research gaps introduced in this subsection highlight the plethora of work that still needs to be done to improve ML and AI systems at a higher level – through tools like XAI and AXAI. These gaps also evidence that we can improve affective state assessment solutions – by using higher level tools like AXAI to address issues pertaining to explainability, accountability, comprehensibility, and reasoning, while also attempting to address more novel gaps such as adapting ASA systems to account for occluded expressions and deploying BSS techniques within the realms of affective computing. This thesis aims to address the research gaps discussed above by making novel, significant contributions that would add to the existing body of literature.

1.4 Contributions

Across the two overarching objectives of this work, this thesis and its corresponding publications have made significant contributions toward the body of literature. As visualised in Figure 1, this work presents eight contributions that aim to address the research gaps that were raised in Section 1.3.

1.4.1 The AXAI Capability Framework

As addressed in Section 1.3.1, there are many research gaps that can be addressed with respect to general ML and AI systems. The emerging discussion on explainable AI and interpretable models is a clear indication for the advancement of AI systems in the modern age. While XAI is beneficial, through the gaps discussed, it is evident that the way in which we assess these systems can be improved further. Through the design and implementation of the Accountable, Explainable Artificial Intelligence (AXAI) framework, this work contributes toward general AI and ML literature. The AXAI framework allows for the delineation of ML and classification systems based on their comprehensibility,

accountability, and predictive accuracy – with accountability serving as the novel addition to assessment tools when compared to existing works.

1.4.2 Blind Source Separation for Group Sentiment Analysis

Figure 1 highlights several branches of affective computing that have been addressed through this work. Through an extensive survey of the literature, there was no evidence of BSS being used in the realm of affective computing. This work contributes to affective computing literature through a novel method of analysing group sentiments. Integrating this system into existing, capable infrastructures would allow for the separation of speech signals and the assessment of the overall mood of a crowd, for the potential use in security, entertainment, and commercial sectors.

1.4.3 Design and Implementation of a Dynamic, Multimodal, State Assessment System

The “Big Idea” proposed during the inception of this thesis, was a modular device that could equip any robotic system with affect awareness. While the overall specifications of this project may have shifted through the research process, the core vision has remained consistent, and has been achieved. This required an extensive literature review into existing multimodal ASA systems to find dynamic, multimodal classification models. As discussed in Section 1.3.2, the number of solutions that accounted for continuous expressions of states were limited, and as such, provided an opportunity in which to make a significant contribution.

The SAM back-end design uses a hierarchical classifier ensemble to provide multi-level, dynamic classification that accounts for continuous changes in affective state expressions as well as temporal considerations for both facial expression and speech signals. The classification architecture is made transparent through the GUI, providing users with a vast array of observable outputs, inputs and auxiliary, with this implementation being discussed in later sections.

1.4.4 Occluded Facial Expression Detection and Classification

Current and past ASA systems that require facial data for effective classification were hindered in their functionality given the changing landscape due to the COVID-19 pandemic. The change in societal views and acceptance of face masks in public, particularly in western countries, would see these systems struggle, or even fail, to handle occluded facial expressions. The idea was to propose an adaptive classification framework as evidenced in [25], which would determine whether to classify upper- or full-facial expressions, depending on whether an occluded face was detected.

Given the current climate, this would be a significant contribution as it would prompt researchers and developers to evolve past systems and to futureproof current systems. History shows that another pandemic is likely in the future, and protective face masks are generally more accepted, and will be in the future. As mentioned prior, the presence of protective face masks in public spaces has been an accepted practice in some corners of the world as such, these countries would also benefit by adopting such a system, further evidencing this work as a valid contribution to the literature.

1.4.5 Hybrid Learning for the Development of Hierarchical Classification Models for ASA

The State Assessment Module contains two hierarchical classifier ensembles, one for each assessable modality (facial expressions and speech). The development of these models was done using a hybrid

learning approach, employing both supervised and unsupervised learning techniques. This approach exploits the statistical similarities embedded in partially-labelled facial expression and speech data, allowing for the development of a holistic classification system and novel classes.

While hybrid learning and hierarchical classifiers are not novel, the implementation of the techniques discussed in this thesis within the domain of ASA, proposes some novel elements. The hybrid model in question leans on the foundations of literature to structure and justify the different levels of the hierarchical model, and through a high classification performance, proves itself as a valid contribution toward ASA literature.

1.4.6 Multi-level, State-based Reasoning for Speech, and Facial Expression Classifiers

The development of the hierarchical classifier ensemble combines foundational theory with statistical clustering and similarity measurements. For both speech and facial expression channels, there are four levels of classification, with each level hosting ensembles of either ternary or binary classifiers. The “state-based reasoning” comes from the literary foundations, in that the logical flow of information across levels is relayed to the user through SAM’s GUI assets, acting as a means of explaining its classification trees and SAM’s reasoning processes. As discussed in previous sections, the lack of transparency and explainable elements in ASA systems poses a major gap in literature that needs to be addressed going forward. Through SAM’s intuitive GUI design and comprehensive functionality, it proves itself as an example of how this research gap has been addressed.

1.4.7 Real-time, AXAI-Capable Assessment and Interaction through SAM

At the core of this research is the AXAI framework and SAM. While there are research gaps belonging to general AI and ML as well as ASA, the implementation of SAM contributes toward both as well as to wider engineering literature. SAM is a real-time system that was designed with AXAI in mind throughout the entirety of the software design and development process. The extensive functionality of SAM does not limit it to being purely a classification system as it contains many impressive sub-systems that make it an advanced AI solution.

In later chapters we will discuss how SAM serves as a foundation as to where affective computing and human-machine interaction (HMI) systems can go in the future. The understanding of human feelings, intent and moods is no longer science fiction, and through contributions like SAM, we can move toward machines understanding humans and their decision-making at a more conscious level.

1.4.8 Integration of SAM into Existing Robotic and Surveillance Infrastructures

This work contributes toward general engineering, robotics, and AI literature through SAM’s deployment as an embedded system. In later chapters, we will discuss how SAM (as a modular, portable system) can be integrated into existing robotic and surveillance infrastructures given the necessary peripheral devices. By integrating SAM into existing frameworks, it would equip *any* robotic system with an intuitive GUI, affective state assessment capabilities and the ability to partake in dialogue with a user.

This is a major contribution that has the potential to be expanded in many directions across various industries and highlights a significant step forward in the realm of affective computing.

1.5 Chapter Description and Summary

This thesis aims to discuss all the above contributions and outline the theory, implementation, and results of this research. This thesis is structured as follows. Chapter 1 has introduced the research, identifying the main objectives, research gaps and significant contributions of this work. Chapter 2 discusses the background information for all facets of this work, from ML foundations to AI systems, XAI and transparency, to the theoretical foundations of emotions and affective computing.

Chapter 3 dissects the AXAI framework, detailing comprehensibility, accountability, and predictive accuracy metrics, and providing two case studies of different classification systems. Chapter 4 shifts the discussion toward software design and development principles, focusing on how AXAI features could be implemented in the design phase of a system.

Chapter 5 details SAM, discussing back-end and front-end designs as well as discussing SAM as a modular, embedded device capable of being integrated into existing infrastructures. Chapter 6 discusses some of the parallel research efforts and other major contributions including BSS and dynamic micro-expression detection in upper- and lower- faces.

Finally, Chapter 7 concludes this work, emphasising the importance of the AXAI framework and SAM, and discussing potential future work that could arise from this research.

Chapter 2: Background Theory

There are many facets to this research that span from machine and deep learning theory to intelligent systems, embedded technologies and affective computing, this chapter reviews the literature and discusses some of the fundamental theory that has shaped this research.

2.1 Foundational Machine and Deep Learning Theory

Machine and deep learning architectures and applications have grown throughout the late 20th and early 21st centuries. Underneath the larger umbrellas of machine and deep learning are supervised and unsupervised learning categories, each playing host to an array of different algorithms and approaches. Both learning approaches are built from statistical and probabilistic computational foundations, and their applications span across various industries, as visualised in Figure 4.

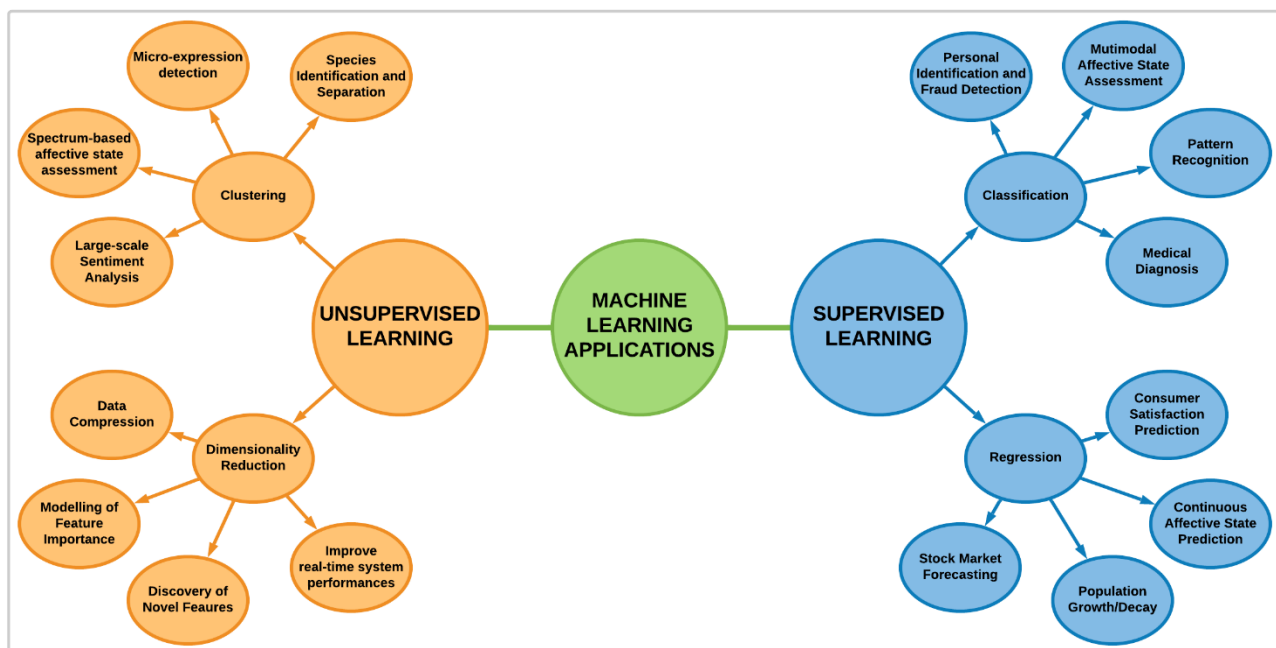


Figure 4: Visual representation of applications in which supervised and unsupervised learning techniques have been deployed, highlighting the various independent sectors including: commercial, marketing, medical, security and social.

In comparison to deep learning, *machine* learning approaches define some of the more traditional, conventional classification methods that have singular feature extraction and classification methods [30]. When we consider deep learning processes, we think of multi-level structures and techniques that utilise artificial neural networks (ANN). Neural networks attempt to mirror human decision-making and feature extraction processes through *digital* nodes and synapses that are described through multi-variate network weights and descent-based algorithms or transfer functions [30-32].

Irrespective of the depth of learning, these techniques are often split into: (i) supervised, (ii) unsupervised and (iii) hybrid learning approaches, with the latter describing techniques that combine supervised and unsupervised techniques in one application. This research deploys hybrid learning, and as such a thorough literature review is conducted for both supervised and unsupervised learning methods.

Through this section, the aim is to discuss the background theory behind all methods that have been explored and deployed through this research, i.e.: (i) mutual information extraction and clustering, (ii) blind source separation (BSS), (iii) linear discriminant analysis (LDA), (iv) support vector machines (SVM), (v) multi-layer perceptron (MLP) networks and (vi) convolutional neural networks (CNN).

At its core, supervised learning implies the existence of labelled data which is used for training a particular algorithm. The correlation and statistical similarity of features provide a basis in which labels are applied to classes (prior to applying the supervised learning algorithm), with the purpose of training a classification model using labelled data and using the model to make a statistical prediction of an input sample/s, based on a set of rules and parameters gained throughout training [33].

In comparison, unsupervised learning techniques are aimed at information extraction and finding representations of data from unlabelled datasets [34, 35]. These techniques allow us to infer particular relationships, clusters and components from analysed data that may be unobservable through natural human sense [35]. The ability to cluster and separate data highlights an immediate benefit for ASA systems like the one proposed in this work – introducing a vehicle in which discretely labelled data can be separated into continuous representations of state in a temporal domain. Then, by deploying supervised learning algorithms, it would allow for dynamic classification within a continuous space, as opposed to limiting classification capabilities through purely discrete, state-based classification. This describes at the highest level, the hybrid learning approach proposed in this work.

Within the context of this research, unsupervised learning provides a medium in which spectrum-based, continuous assessment solutions may be applied, and representations of group sentiments may be found. Manual labelling of a large corpus of data requires an extensive number of resources and introduces human error and biases. Thus, by applying unsupervised learning, one can remove the inherent biases that exist in humans, in favour of more objective, feature-based representations and labels.

Regardless of modality, application or method, unsupervised learning proposes a means of learning the probability distribution of data [34, 36], the categories of methods are extensive and can range from traditional clustering, to manifold learning, to blind source separation etc. This research provides evidence of BSS and traditional clustering techniques being applied for ASA tasks, looking at mutual information and the extraction and measurement of similarity and dissimilarity in continuous signals.

2.1.1 Unsupervised Learning – Blind Source Separation

Blind source separation utilises sparse decomposition of signals as the basis for separation, under various assumptions of signal mixture probabilities [35, 37]. In essence, the overall concept of BSS algorithms is to separate a smaller number of observed signals into a greater number of estimated source signals, serving as a means of data separation and is sometimes used as a pre-processing [28, 29]. The governing BSS equation characterises a continuous, observed signal mixture ‘ $\mathbf{x}(t)$ ’ and is represented as [37]:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{b}(t), \quad 1 \leq t \leq T$$

where ‘ \mathbf{A} ’ defines the mixing matrix, ‘ $\mathbf{s}(t)$ ’ defines the source signals and ‘ $\mathbf{b}(t)$ ’ defines any noise present in the signal mixture, which is sometimes omitted if negligible.

Blind source separation aims to solve the equation for $\mathbf{s}(t)$ given an unknown mixing matrix and only having knowledge of the observed signal mixture. Blind source separation techniques are supported by the fact that at every point in the continuous mixture, one signal has a higher activation than the rest [37]. This primary assumption indicates that if the mixture can be modelled effectively and separated on the basis of these higher activations, then the source signals may be estimated. Solving BSS problems requires one to determine a demixing matrix \mathbf{W} through computational and component analysis methods [35, 37]:

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t) = \mathbf{W}\mathbf{A}\mathbf{s}(t) \quad (2)$$

where $\mathbf{y}(t)$ is the estimated separated signals given the observed signal mixture $\mathbf{x}(t)$. The obvious solution is that $\mathbf{W} = \mathbf{A}^{-1}$, which is unachievable given the unknown mixing parameters of \mathbf{A} and its inverse. Therefore, some kind of similarity measures need to be deployed to find the maximum-likelihood solution for the demixing matrix \mathbf{W} .

Depending on the statistical conditions/assumptions of the observed mixture, different signal decomposition techniques may be deployed to solve the separation problem. In this work, *Independent* Component Analysis (ICA) and *Dependent* Component Analysis (DCA) methods were explored, with the former assuming non-Gaussianity and statistical independence between source signals. Conversely, DCA assumes statistical dependence and can be more applicable in the real world where Gaussian mixtures are more present [35].

Independent Component Analysis aims to maintain statistical independence between signals in a mixture and while the definition of ICA tends to vary throughout literature, researchers tend to agree [38]: ICA of a random vector ' \mathbf{x} ' consists of estimating the following generative model for the data:

$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

which is of the same form as (1) defined prior – only that noise is omitted from the mixture in this case. Equation (2) defines the problem statement where the demixing matrix ' \mathbf{W} ' is the variable to solve. Learning algorithms are often used to determine ' \mathbf{W} ' such as the Kullback-Leibler (KL) divergence method. Novel computational methods have seen the evolution of ICA learning algorithms, spawning tools such as Fast-ICA, which is an implementation of the algorithm derived in [39], represented by:

$$\mathbf{W}^+ = \mathbf{W} + \Gamma[\text{diag}(-\beta_i) + E\{g(\mathbf{y})\mathbf{y}^T\}]\mathbf{W} \quad (3)$$

where $\mathbf{y} = \mathbf{W}\mathbf{x}$, $\beta_i = E\{y_i g(y_i)\}$ and $\Gamma = \text{diag}\left(\frac{1}{\beta_i - E\{g'(y_i)\}}\right)$. In their work, Hyvärinen and Oja compare (3) with the method used to determine maximum likelihood [39], i.e.:

$$\mathbf{W}^+ = \mathbf{W} + \mu[\mathbf{I} + g(\mathbf{y})\mathbf{y}^T]\mathbf{W} \quad (4)$$

This comparison sees Fast-ICA as an algorithm used for fixed-point maximum likelihood independent component analysis i.e., blind source separation through estimation. Successful implementation of the Fast-ICA algorithm will produce ' n ' separated source signals, maintaining statistical independence between them.

In comparison to ICA, DCA methods aim to achieve the same task of BSS, only under the assumption of statistical dependence – which as mentioned prior, is more prevalent in real-world scenarios.

Statistical dependence assumes some form of mutual information that is contained in the signal mixture, which affects the process of separation. One DCA method is Multidimensional Independent Component Analysis (MICA) which is an extension of the traditional ICA method. MICA adopts the theory that n-tuples of signals in a mixture may be dependent.

The structure of the MICA algorithm may be summarised as such [40, 41]:

1. Identify one-dimensional components and make a primary estimation of the separation matrix elements.
2. Identify Gaussian characteristics and group these 'k' elements to form multidimensional components C_k
3. If a source signal satisfies $C_k > 1$, decompose the signal into k one-dimensional components – which is done by projecting the source onto the first axis of the covariance matrix.
4. Optimize a global criterion function $G(\cdot)$ such that the separation matrix and observed mixture $\mathbf{W}\mathbf{x}$ satisfies the condition:

$$G(\mathbf{W}\mathbf{x}) = 0$$

Another method which maintains statistical dependence is the dual-tree complex wavelet transform (DTCWT), which exploits shift invariances and orthogonality in signal mixtures [42-44]. The DTCWT method finds statistical dependence and mutual information at particular frequencies while maintaining the notion that other frequencies may still be statistically independent.

The DTCWT technique employs a linear operator T_k to separate independent frequencies such that:

$$\mathbf{s}_k(t) = T_k[\mathbf{s}(t)] \quad (5)$$

Applying this operator to the blind source separation base equation, the observed signal may therefore be represented in the form:

$$\mathbf{x}_k(t) = T_k[\mathbf{A}\mathbf{s}(t)] \quad (6)$$

These rules allow for the following generalisation to be made:

$$\mathbf{A}T_k[\mathbf{s}(t)] = \mathbf{A}\mathbf{s}_k(t) \quad (7)$$

Based on theoretical foundations formulated in [42-44], the DTCWT method may exploit real and imaginary components of the wavelet transform $\psi(t)$ such that:

$$\psi(t) = \psi_i(t) + \psi_j(t) \quad (8)$$

The Hilbert transform is then invoked on the real component $\psi_i(t)$, allowing the complex variable $\psi_j(t)$ to take the desired form of:

$$\psi_j(t) \approx [\mathcal{H}\psi_i(t)] \quad (9)$$

The DTCWT method serves as a means of decomposition, the mutual information at each frequency is approximated using entropy measures and subsequent second, third and fourth order cumulants were used to for probability distribution approximations. Once the DTCWT method has been employed for decomposition, the ICA algorithm mentioned prior may be used to estimate the demixing matrix \mathbf{W} , which through the DTCWT method maintains a signal mixture's statistical dependence assumptions.

2.1.2 Unsupervised Learning – Clustering

Clustering and cluster analysis aim to infer natural relationships from similar instances in a dataset by grouping them in regions based on their similarities [45, 46]. This relates to the application of automated labelling. By having a system that defines various regions based on the objectivity of feature similarities, one may find new relationships and classes that may be otherwise unobservable when performing manual labelling.

Clusters will form in a dataset regardless of which clustering algorithm is chosen and regardless of if the data contains distinct clusters or not [46]. Therefore, choosing an effective clustering algorithm is important as the data may be represented in a more beneficial, applicable way, despite not changing the input dataset/data mixture. The selection of features is also dependent on the modality in which the data presents itself, there are no defined guidelines or restrictions which state what patterns or features are to be inferred through clustering [46].

In this work, the separation and clustering of homogenous, gaussian, continuous data was achieved by exploiting similarity and dissimilarity measures. Prior to discussing the theoretical concepts, let us define some standard definitions used in this work:

- $\mathbf{x}_i = \{x_1, x_2, \dots, x_m\}$ defines a singular pattern or feature vector, where ' x_i ' defines feature ' i '.
- $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ defines a set of input patterns of matrix size $N \times m$, which is a collection of singular feature vectors used in the clustering problem.
- $\mathbf{C} = \{c_1, c_2, \dots, c_K\}$ defines the ' K ' class labels for the patterns contained in the pattern set ' \mathbf{X} '.

Similarity measures allow us to detect statistical correlations that may exist within data, allowing for the separation and clustering of subsets of similar feature vectors ' \mathbf{x}_i ' from within the set of input patterns ' \mathbf{X} '. Some of the most common, logical tools for detecting similarity are distance-based, including Euclidean, Manhattan and Mahalanobis distances [35], being defined as such:

$$D_{Euclidean} = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad (10)$$

$$D_{Manhattan} = \sum_{i=1}^n |x_i - y_i| \quad (11)$$

$$D_{Mahalanobis} = \sqrt{(x_i - y_i)S^{-1}(x_i - y_i)^T} \quad (12)$$

In this instance, x_i and y_i define two points in an n -dimensional workspace. Logically, Euclidean distances of points contained in the same clusters will be smaller than those in foreign clusters. Therefore, a key consideration in this case, is the 'threshold distance', the distance in which clusters are separated from each other [35]. Too large a threshold distance and the number of clusters ' K ' may be reduced. In contrast, too small a threshold distance and the number of clusters ' K ' may increase to a point where clusters become too insular [35].

The previous subsection (2.1.1) discussed statistical dependence within the context of BSS and dependent component analysis methods. The concept of mutual information within a set of feature vectors/patterns ' \mathbf{X} ' indicates the presence of statistical dependence within that set. Mutual

information can be used to describe a subset of samples in a given set that are dependent and share similar features, thus serving as a vehicle for the detection of similarity between feature vectors.

Applying mutual information extraction within the context of this research, it was used as the unsupervised learning part of the hybrid model, tasked with separating and clustering continuous facial expression and speech data into 'K' classes that account for changes in temporal phase to create a dynamic, spectrum-based state classification solution across two modalities.

Three mutual information extraction techniques were experimented with though this research:

1. Conditional Entropy
2. Mutual Information Score (KL divergence)
3. Cosine Similarity

with cosine similarity being used for final models. All three approaches achieved similar results, but their governing approaches and equations are unique. Conditional entropy is represented by [24, 25, 47]:

$$H[X_1|X_2] = - \sum_{i,j} p(X_1 = \mathbf{x}_i, X_2 = \mathbf{x}_j) \times \log(p(X_1 = \mathbf{x}_i, X_2 = \mathbf{x}_j)) \quad (13)$$

where $p(X_1 = \mathbf{x}_i, X_2 = \mathbf{x}_j)$ represents the probability that a random variable $X_1 = \mathbf{x}_i$ and another variable $X_2 = \mathbf{x}_j$, with the pair of variables describing sequential feature vectors within the sample set 'X' in this application. $H[X_1|X_2]$ must be greater than zero for all cases of X_1 and X_2 noting that as two samples become more similar, $H[X_1|X_2] \rightarrow 0$.

Mutual Information Score is derived directly through the KL divergence of the joint and product distributions of two random variables [48]. Note that the KL divergence method was also discussed in the previous section w.r.t determining the demixing matrix parameters for BSS, thus highlighting the prevalence of this method in the realm of unsupervised learning. Mutual information score can therefore be modelled through:

$$I[X_1; X_2] = \sum_{i,j} p(X_1 = \mathbf{x}_i, X_2 = \mathbf{x}_j) \times \log\left(\frac{p(X_1 = \mathbf{x}_i, X_2 = \mathbf{x}_j)}{p(X_1 = \mathbf{x}_i) \cdot p(X_2 = \mathbf{x}_j)}\right) \quad (14)$$

Comparing (13) and (14) we can see consistent elements across both conditional entropy and mutual information score approaches, highlighting why the results of separation may be similar across both methods.

The final separation algorithm explored more foundational vector algebra i.e., cosine similarity. This method detects the cosine angle between two m -dimensional feature vectors \mathbf{x}_i and \mathbf{x}_j , the resulting angle showing the mutual information between two vectors. Cosine similarity ' $S_{\cos\theta}$ ' is defined as:

$$S_{\cos\theta} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i| |\mathbf{x}_j|} \quad (15)$$

Analysing (15), we note that as the mutual information between two feature vectors rises, $S_{\cos\theta} \rightarrow 1$. Through basic algebra, we can use this equation to prove how similarity increases as $\cos^{-1}(1) = 0^\circ$,

which indicates two identical feature vectors and $\cos^{-1}(0) = 90^\circ$, which corresponds to two perpendicular, dissimilar feature vectors.

All three methods defined above can be used to detect the similarity of two samples. In this work, the *dissimilarity* between two sequential samples \mathbf{x}_i and \mathbf{x}_{i+1} was more important, as the points of maximum dissimilarity in a continuous sample set ' \mathbf{X} ' determines the locations of the m -dimensional hyperplane boundaries that are used to construct the cluster borders.

Given this work clustered continuous samples of facial expression and speech modalities, the dissimilarity is calculated for all sequential sample pairs in a set using the following notation:

$$\Delta S = S(\mathbf{x}_i, \mathbf{x}_i) - S(\mathbf{x}_i, \mathbf{x}_{i+1}) \quad (16)$$

where ' S ' is the similarity measure used for separation and ' \mathbf{x}_i ' is the i^{th} feature vector in a set ' \mathbf{X} '. Using cosine similarity as an example, (16) can be rewritten as:

$$\Delta S_{\cos\theta} = S_{\cos\theta}(\mathbf{x}_i, \mathbf{x}_i) - S_{\cos\theta}(\mathbf{x}_i, \mathbf{x}_{i+1})$$

which can be written as:

$$\Delta S_{\cos\theta} = 1 - S_{\cos\theta}(\mathbf{x}_i, \mathbf{x}_{i+1})$$

given that the algorithm has been applied correctly, since the ' $S_{\cos\theta}(\mathbf{x}_i, \mathbf{x}_i)$ ' term defines the similarity between two identical feature vectors. For all feature vectors in \mathbf{X} , we can therefore describe the function:

$$f(\mathbf{x}_i) = \Delta S : \mathbf{x}_i \in \mathbf{X} \text{ and } \mathbf{x}_{i+1} \leq \mathbf{x}_N \quad (17)$$

such that $\max[f(\mathbf{x}_i)]$ defines the maximum dissimilarity in \mathbf{X} . By splitting \mathbf{X} into ' k_p ' equal parts, one can then detect ' k_p ' peaks, such that:

$$K_{classes} = k_p + 1 \quad (18)$$

which determines the number of clusters that have been extracted from the sample set. The application of this clustering method within the context of this research is detailed more extensively in Chapter 5.

2.1.3 Supervised Learning

This research explores many supervised learning methods and algorithms that range from traditional, shallow machine learning approaches to more complex, deep learning solutions. To reiterate, this research explored and adopted four unique supervised learning methods for dimensionality reduction and classification purposes, i.e.:

1. Linear Discriminant Analysis (LDA)

In addition to LDA being applied for classification, it can also be used as a supervised learning dimensionality reduction technique, as has been applied in this work. By having knowledge of labelled sample data, LDA projects *higher*-dimensional data onto a *lower*-dimensional linear discriminant (feature) space. This process maximises the *inter*-cluster variance while minimising the *intra*-cluster variance, thus optimising the separation between cluster centroids [49]. Immediately, it was evident that combining this method with the separation and clustering techniques discussed in subsection

2.1.2 would enhance the classification capabilities of the presented system, deploying two independent clustering algorithms.

Unlike the unsupervised separation and clustering algorithms discussed in previous sections, LDA is an optimisation problem that leans on the benefits of labelled data. The aim is to determine the optimal value of 'b' in (19) that would maximise:

$$\frac{b' \mathbf{B} b}{b' \mathbf{W} b} \quad (19)$$

Where ' \mathbf{B} ' represents the *inter*-cluster covariance matrix and ' \mathbf{W} ' represents the *intra*-cluster covariance matrix. The solution for (19) is to calculate the number of linear discriminants ' N_{LD} ', which can be found via the largest eigenvalues of $\mathbf{W}^{-1} \mathbf{B}$, keeping in mind that for ' $K_{classes}$ ' [26, 49]:

$$N_{LD} \leq K_{classes} - 1$$

All samples in a given set \mathbf{X} are then projected onto an N_{LD} -dimensional linear discriminant feature space. As briefly mentioned, LDA may be used for classification tasks as well. While this functionality of the algorithm is not applied in this work, classification is achieved through the detection of similarity between an input signal and the cluster centroids of the LDA model, e.g.:

$$|b'(\mathbf{x}_i - \bar{\mathbf{x}}_1)| - |b'(\mathbf{x}_i - \bar{\mathbf{x}}_2)| - |b'(\mathbf{x}_i - \bar{\mathbf{x}}_k)| - \dots - |b'(\mathbf{x}_i - \bar{\mathbf{x}}_{K_{classes}})| < 0 \quad (20)$$

where ' $\bar{\mathbf{x}}_k$ ' is the k^{th} cluster centroid.

2. Support Vector Machines (SVM)

The SVM model is one of the most prevalent across supervised learning literature. They discern multidimensional data through hyperplanes that separate labelled data. Similar to the LDA approach, SVM's maximise intra-cluster margins to construct the hyperplane boundaries, defining classifiers of the form [24, 50]:

$$y(x) = \text{sign} \left[\sum_{i=1}^N \alpha_i y_i \psi(\mathbf{x}, \mathbf{x}_i) + b \right] \quad (21)$$

where ' \mathbf{x}_i ' and ' \mathbf{y}_i ' correspond to the i^{th} input and output feature vectors, respectively. ' α_i ' and ' b ' are constants. $\psi(\mathbf{x}, \mathbf{x}_i)$ is the operating term and is dependent on the type of SVM kernel used for classifier training. Two common SVM operating terms are the Linear and Radial Basis Function (RBF) terms [50, 51], described respectively by (22) and (23):

$$\psi_{linear}(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}_i^T \mathbf{x} \quad (22)$$

$$\psi_{RBF}(\mathbf{x}, \mathbf{x}_i) = \exp \left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2} \right) \quad (23)$$

which promotes the two solutions:

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b \geq +1, \quad \text{for positive } \mathbf{x}_i$$

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b \leq -1, \quad \text{for negative } \mathbf{x}_i$$

where ' \mathbf{w} ' is a weight vector and ' $\phi(\mathbf{x}_k)$ ' is a nonlinear function (operating term) [51]. The construction of the SVM models and the parameters used in this work is detailed in Chapter 5 where facial expression and affective speech classification models are discussed in greater depth.

3. Multi-Layer Perceptron (MLP) Networks

Multi-Layer Perceptron Networks are an evolution of the traditional single-layer perceptron algorithm derived in the mid-20th century [52, 53]. A perceptron at its core, is an artificial neuron, designed to mimic human neural feature extraction for binary classification. The MLP algorithm is a feed-forward, fully-connected neural network that is capable of approximating linear and non-linear functions for the classification of unseen data [52, 54].

Artificial neural networks (like MLP) are graphical in nature, consisting of hidden layers and artificial neural nodes. The nodes are connected via linear/non-linear activation functions – of which there are various unique functions, each being more applicable to some classification tasks over others [52]. The governing equation for a neural network node can be defined as [55]:

$$h_i = \sigma \left(\sum_{j=1}^N W_{ij} x_j + T_{i(hid)} \right) \quad (24)$$

where h_i is the output of the i^{th} neuron in the hidden layer, based on the activation function ' $\sigma(\cdot)$ '. ' N ' defines the number of input neurons with weights ' W_{ij} ', inputs ' x_j ' and threshold term for the hidden neurons ' $T_{i(hid)}$ '. Some common activation functions that appear in literature are:

$$\sigma_{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (25)$$

$$\sigma_{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (26)$$

$$\sigma_{ReLU}(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \quad (27)$$

$$\sigma_{binary}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \quad (28)$$

The number of output layer nodes of the MLP network define the number of classes $K_{classes}$ present. An MLP network is trained using a training set \mathbf{X} containing consistent m -dimensional feature vectors and is dependent on the construction of the network i.e., the number of layers, the chosen activation functions, and the network weights [52]. The aim of the network is to define and optimise an unknown function ' f ' that describes the network and links the input layer to the output layer, thus facilitating classification [52, 55].

4. Convolutional Neural Networks (CNN)

Convolutional neural networks show another evolution from traditional neural network algorithms and perceptrons, and are designed to process, learn about, and classify array-shaped data [30, 56, 57]. The evolution of CNN algorithms has shown powerful improvements to pattern recognition tasks that require multidimensional input data for example when handling time series data and multi-channel signals, performing RGB image/audio classification [56, 57].

Convolutional neural networks can range in size complexity and the evolution of CNN's has seen the development of many unique deep learning algorithms such as ResNet, VGGNet, AlexNet, Inception, Xception etc. [58] that have all been built from the same foundational knowledge, which again, evolves from the "simple" perceptron. These CNN's all range in network structure, size and complexity, with approaches like the Inception and Xception algorithms being explored in previous works [1, 19, 24]. A description of these algorithms is highlighted in Table 1.

Irrespective of the algorithm, there are various unique layers that can be added to a CNN, especially when using common development tools such as Keras and TensorFlow [58]. Through convolutional layers, we establish the basic building blocks of CNN's. These layers define the $(n \times m)$ -dimensional feature maps that are embedded with multi-channel information and to detect local feature information of an input signal [30] and by varying parameters such as the kernel size or padding for example, one is able to teach the CNN "how" to learn.

Name	Year	No. Parameters	Description
AlexNet [59]	2012	60 million	Convolutional Neural Network designed and competed in the ImageNet contest consisting of five convolutional layers and three fully connected layers with other neural network elements existing between layers [59]. The model achieved respective top-1 and top-5 error rates of 37.5% and 17%.
VGGNet [60]	2014	138 million	A convolutional Neural Network architecture containing 16 weight layers including maximum pooling layers. The number of parameters in this architecture is considerably higher than the others in this table. VGGnet was used as a competitor in the ImageNet competition, boasting top-1 error rates between 25.5 and 30% and top-1 error rates ranging between 8 and 10.5%. [60]
InceptionV3 [61]	2016	24 million	The InceptionV3 model combines several inception modules which are designed to perform cross-channel and spatial correlations more efficiently. The author reported a 3.5% top-5 error rate and a 17.3% top-1 error rate on the validation set using an ensemble of Inception models [61].
Xception [62]	2017	24 million	The Xception architecture is more novel than its predecessor (InceptionV3) but is claimed to be "inspired by Inception" [62]. The model contains 36 convolutional layers and reported a top-1 error rate of 21% and a top-5 error rate of 5.5%.
ResNet [63]	2016	60.2 million (when using ResNet-152)	Explores the notion of using layers as "learning residual functions" with relationship to layer inputs. The corresponding paper refers to different layer configurations, each resulting in different performance metrics. The best performing architecture on the ImageNet dataset uses 152 layers (defined as ResNet-152), boasting a top-1 error of 21.43% and a top-5 error of 5.71% [63].

Table 1: Comparison of various Convolutional Neural Network (CNN) architectures that are common in the relevant literature and are deployed for various applications across independent industries.

2.1.4 Hybrid Learning

Hybrid learning systems are a combination of unsupervised and supervised learning approaches, oftentimes exploiting the benefits of both approaches for great results. This research heavily explores hybrid learning and classifier ensembles through its adoption of a hierarchical classification network for continuous affective state assessment. The construction of hybrid learning frameworks in [24, 25]

make up much the back-end classifier functionality of SAM (as will be discussed in Chapters 5 and 6 of this work). However, there are various other examples of hybrid learning in literature outside of the works that have been published as a result of this research.

In [64], researchers discuss a hybrid learning method based in deep learning theory to improve emotion recognition accuracy. In their work, they use dimensionality reduction that is granted by hidden layer representations (unsupervised) prior to using a MLP network for affective state classification (supervised) [64]. This architecture is experimented on an affective speech dataset where valence, arousal and dominance recognition accuracy are used as performance metrics, with the classification performances being improved as the result of deploying a hybrid learning model.

Another example in [65] proposes a hybrid combination of RBF-based K-means clustering with a bi-directional long short-term memory (LSTM) network using speech samples segments as the input for the unsupervised clustering algorithm. The K-means clustering approach allows for the extraction of “key segments” which are taken at a point near to the cluster centroid. This key segment is sequenced and serves as the input of the Bi-LSTM network [65]. Researchers validated the method on three datasets, observing accuracies between the range of 72% and 86%.

Research conducted in [66] proposes a hybrid neural network approach for the classification of emotions on social media platforms, applying the hybrid technique for sentiment analysis. In their research, the authors incorporate semantic features into a neural network model, deploying the unsupervised models of the hybrid network for feature transformation and a feed-forward perceptron which serves as the supervised component of the hybrid network [66]. Through their hybrid approach, the authors successfully improve social emotion classification, again proving the benefits of exploiting the benefits of both supervised and unsupervised models, as was evidenced in other works discussed [24, 25, 64-66].

Machine and deep learning models are the structural building blocks for applied artificial intelligence and statistical classification systems. The predictive capabilities of these systems have seen them deployed in many industries across the globe. Throughout this section we have discussed the mathematical and statistical foundations of these various approaches and algorithms. But, to a layperson, or someone unfamiliar with the literature, the vast amounts of theory would be overwhelming and highlights why this research focuses on how these systems can be improved in terms of their comprehensibility, accountability, and overall interpretability.

By compressing these models into a black-box and deploying them for classification tasks, it would be impossible to develop a truly intelligent AI system or robotic assistant that would be capable of explaining their decision-making processes. Thus, AI researchers and developers need to be aware of how they can improve the usability and transparency of their systems beyond their mathematical foundations, ensuring that that are designed to be more human-understandable.

2.2 Artificial Intelligence (AI) Systems and Robotic Assistants

The philosophical conflict and relationship between Man and Machine has been discussed in fictional and non-fictional works for the last two centuries. Through fictional works, parties have been portrayed as both allies and adversaries in utopian and dystopian scenarios. Examples include Frankenstein (1818), Metropolis (1927), Terminator (1984), Matrix (1999) and many 20th century works by Isaac Asimov including his ‘Robot’ short story series. These works have seen the rise of

cybernetics, robotics and artificial intelligence systems in popular culture and have contributed consciously and subconsciously toward the public perception and acceptance of these systems over time.

Translating to real-world implementations, the concept of robotics and automation has been a theme of discussion since the digital revolution [10], with the advancement of information technology, integrated circuits and semiconductors serving as the catalyst for this change. The invention of the microprocessor which eventually evolved into what is today known as the personal computer was a boon for many industries around the world, especially within the manufacturing industry, as fully automated factories were seen as an inevitable reality.

The Oxford Dictionary defines a robot as “*A machine that can perform complicated tasks automatically*” [67]. A robot can therefore come in many forms both digitally and physically if it is a machine that has been programmed to complete a task automatically. The term ‘robot’ is generally reserved for physical robots, whereas digital machines and virtual robots are generally referred to as ‘bots’ – a robot without a body [68]. Robots and bots were developed as physical and virtual assistants to aid in many facets of human life as a way of either supplementing and/or replacing tasks that are performed by humans.

Continuous improvements of AI systems, virtual agents and robotics runs parallel to advancements of human machine interactions [69]. Developing a system capable of understanding human emotion and/or sentiment will further advance human-machine interactions as our lives become more entangled with technology. However, as emphasised throughout this work, comprehensibility and accountability must be present in these AI systems throughout the development process as we aim to move away from traditional black-box approaches in favour of more acceptable solutions for the general public.

While algorithms continue to become more advanced and accurate, the back-end decision-making processes of these systems are still not shown to key stakeholders. Without this transparency, the inherent lack of trust will still exist in these systems. Through the incorporation of XAI and AXAI principles, we can further enhance human-machine interaction systems.

2.3 Explainable AI, Reasoning and Transparency

Explainable AI aims to address issues of trust and explainability in AI and ML systems, attempting to improve human-machine interactions in the process [4,5,7-9]. By improving factors such as comprehensibility, understandability, interpretability and transparency, these systems can be enhanced for all stakeholders including developers, regulators, users, and domain experts [4]. There are various elements that can be incorporated into AI systems to promote XAI-capabilities including:

1. **Transparency via input feedback** – Providing a medium to the user that explains what features are being extracted and used for classification and decision-making processes. Visualisations of data throughout the process from input to output, including raw representations of data [70] promote transparency in ML systems. Human-machine interactions would be improved by providing users with the ability to visualise and query input data – especially for critical applications where a domain expert is still necessary when using these systems.
2. **Graphical and data tabular data presentation** – Another form of visualisation that informs users of classifier prediction results over time or visualising decision-making processes. For a

real-time system this may involve graphically showing how predictions change over time or updating timestamped prediction labels and showing this information to the user. However, it must be noted that an overabundance of information and visual assets can negatively impact the user experience, so this information must be optimised [3].

3. **Ability to scrutinise and debug** – Providing users and domain experts with the ability to gain trust in decision-making processes and system performances and providing experts with the ability to suggest system improvements. Giving users access to classification metrics for example, would make them aware of how accurate the system is and would make it easier for them to spot misclassifications (given their expertise in the domain). The ability to scrutinise the methods of classification as well as the predictive capabilities of a system is important in XAI literature [4, 70]. The ability for domain experts to test systems and suggest changes based on their user-experience promotes trust in AI and instils more control in experts – further establishing the ML system as a *tool* rather than an adversary [4].
4. **Explanations through Bayesian probability measures** – Revealing the probabilistic foundations of a classification system and mapping information such that the logical decision-making processes are made transparent. Implementing Bayesian rule lists (BRLs) can be used for reducing classifier output dimensions into a combination of smaller, interpretable models that can assist in the explainability of a ML system [5, 7]. Bayes' theorem provides knowledge of the probability of an event occurring based on prior knowledge of observed condition (prior and posterior probabilities). This can be used to generate explanations of logical processes, thus assisting in the transparency of a system, trust in the system, and an improvement in the overall human-machine interaction experience.
5. **High-level feature attribution and decision trees** – presenting and explaining hierarchical models and classifier ensembles to users through various media allows users to interpret the intra- and inter-signal relationships. Through feature attributions at various levels, the classification strategy of an ML becomes clearer, and is a very important tool in the XAI arsenal [7, 71].

There are various other model-dependent and model-agnostic elements that could be incorporated to improve the interpretability and explainability of AI systems as discussed in [5, 7]. While integrating these features highlight the advantages of XAI over traditional AI systems, as addressed in previous sections, the delineation of XAI is often convoluted and lacks a clear definition of explainability across literature. Thus, the need for a model- and application-agnostic assessment framework is one of the reasons why the AXAI framework is being proposed in this work. Furthermore, given the various stakeholders engaged in the development of AI and ML systems (from developers → organisations → users), explainability and comprehensibility of these systems can be addressed in different ways. Solutions can be specific to different groups involved and could benefit one over another. For example, developers can develop interpretable, rule-based models to enhance the comprehensibility when analysed by other developers, but not-so-much by users who do not have the domain knowledge.

Across literature [4, 5, 7-9, 72-78], we see that concepts like explainability, comprehensibility and transparency are perceived differently depending on the application and what needs these methods are trying to address. The rest of this section will explore how XAI concepts and methodologies have been applied within the context of more traditional machine learning models [72-75], as well convolutional and deep neural networks [76, 78, 79].

2.3.1 Explainable AI Applied for Supervised ML Methods

Section 2.1 previously established the mathematical and theoretical foundations of supervised and unsupervised machine and deep learning techniques, discussing the traditional implementations of these methods. With the growth of technology and research in the domain and a shift toward more interpretable models, efforts have been made to make traditional methods more interpretable such that black-box approaches become a thing of the past.

A survey conducted in [72] highlights the benefits that explainability will have in domains like healthcare, aviation, and human resources. Interpretability is also dependent on the model being deployed, with some models being defined as ‘naturally’ more interpretable based on their algorithmic implementation e.g., decision trees which are often perceived as being very interpretable due to their logical structure. Whereas others can be ‘designed’ to have more explainable elements integrated into them [72]. Rule-based systems, along with textual/graphical media can also be used to convey explainability through design, which is beneficial when models are not naturally interpretable or transparent [70, 71, 73] and ultimately, AI practitioners should have a duty of care to endeavour to make models and applications as explainable and comprehensible as possible (within the constraints of the project/application).

To further the benefits of explainability within the context of the healthcare domain, we can refer to the research conducted in [73], in which the authors evidence the importance of explainable models over traditional black-box machine learning approaches when applying them in a clinical setting within the field of oncology. The authors stress that while other models in literature perform well w.r.t to predictive accuracy measures, their lack transparency make them harder to deploy [73]. To prove that explainability could be incorporated into their models, they use Shapley Additive Explanations (SHAP) values, which is proposed as a model-agnostic means of extracting and outputting feature importance and feature correlations [73, 74].

Moncada-Torres et al. incorporated SHAP to explain the prediction results of three ML models based on: (i) Random Forests, Support Vector Machines and (iii) Extreme Gradient Boosting for survival analysis in oncology [73]. Their results found that their proposed ML models were comparable performance-wise to other state-of-the-art methods, while also having interpretable and explainable decision-making, a capability that was granted through the integration of the SHAP values.

In [74], authors propose an explainable ensemble Gaussian kernel (XEGK) to capture representative features from the input in an attempt to map the input with the output – a form of feature extraction. Similar to the approach discussed in [73], the authors apply the XEGK-extracted features as inputs to SVM and Random Forest Machine Learning models, using the SHAP method to determine feature importance and feature correlations [75] to represent the parsing of data in a more interpretable manner. Results show that their proposed model outperforms compared models when assessing performance, while also having the benefits being interpretable and explainable.

2.3.2 Explainable AI Applied for Neural Networks

As the name suggests and as has been emphasised prior, neural networks are designed to emulate human neural functions and the transfer of information through digital synapses and nodes. What currently separates man from machine is our natural ability to explain the logic and rationale behind our decision-making processes and in most cases, allowing us (humans) to be transparent systems.

Whereas traditional, artificial neural networks (including deep and convolutional NN's) are oftentimes categorised as being black-boxes due to their inability to make their decision-making and inferencing processes transparent. The surveyed works discussed in this section (Section 2.3.2) all contribute toward how explainability has been applied in neural network model architectures.

A survey conducted in [76] provides an overview of the technical properties and implementations of state-of-the-art explainers for deep neural networks. The survey proposes that there are two kinds of explainers: (i) intrinsically interpretable models and (ii) post hoc explainability which is incorporated based on a model's output. From these two definitions we see that there are consistencies in the definitions provided in [76] and [72], highlighting that AI practitioners can deploy naturally explainable (intrinsically interpretable) models as well as models that are explainable by design (post-hoc explainability).

Deep neural network explainers can include [76]:

- **Deconvolution** – Which can be used to show how features are represented visually through mapping and projection, and how this information can be used to identify class probability,
- **Decomposition** – Isolating or limiting network layers or regions to provide an understanding of how the sub-regions influence global network decision-making,
- **Gradients/Back Propagation** – Emphasises feature importance and layer-to-layer unit changes to understand network architectures, with the magnitude of the gradient showing importance from input to output,
- **Visualizations** – A concept which has been (and will be) discussed in depth, using textual, graphical or media representations to provide an explanation for a model's decision-making processes.

The survey also discusses how synthetic, prototypical outputs can be used to explain how a neural network perceives a particular class. Relating to back propagation, by emphasising feature importance and mapping the most important features for a class from output to input (an interpretation of how data is parsed), this information can be used to generate the network's representation of what a particular class should look like [76].

Online tools such as DALL.E mini [77] sparked a pop culture trend in 2022 in which the concept of neural network feature importance could be seen. By providing the machine with a prompt for example "Eiffel Tower", it would use natural language processing and an extremely large dataset of labelled internet images [77] to generate a synthetic, prototypical output based on a user prompt. While none of the images produced are real, it explains to the user what the machine thinks the "Eiffel Tower" looks like based on how the model has been trained and how images had been labelled.

Rule-based systems are common in XAI literature as they provide logical explanations to make black-boxes more transparent. Dattachaudhuri et al. propose a Transparent Rule Extraction using Neural Network (TRENN) method [78], which is an extension of a previous algorithmic approach. The TRENN method contains six sub-processes to optimise and prune network structures, and construct, prune, and update rules. The TRENN network pruning is based on back propagation and the rule construction uses the generated network paths for each class to determine a rule with the best description of the output [78]. However, the generated rules are more tailored toward people with the domain

knowledge and do not appear to be beneficial to those without ML and AI experience as the rules themselves are probabilistic statements which would require some understanding of ML models.

But making neural network models more explainable does not always need to involve removing layers and nodes and creating a critical path. Research reported in [79] proposes a contrary solution in which an additional “concept” layer is added into a CNN architecture to provide a learning path between visual features and image descriptors/word phrases [79]. The authors make their model interpretable by employing a global pooling layer (concept layer) to report model outputs as a weighted sum of learned concepts, which would allow for an integrated explanation [79].

By using word phrases to describe classes rather than a discrete label, the authors propose a model that is more explainable by nature. Rather than a machine outputting “Painted Bunting” for example (a breed of bird), it would instead, using the concept layer and word phrases, provide a ground truth description which evidences which CNN concept layer nodes are activated and to what degree [79]. Across various datasets, the Comprehensible CNN model proposed shows competitive if not, greater performance metrics when compared to other models, while having the benefit of being an interpretable model [79].

Considering the applications of AI and ML systems, it is important to address some of the elements discussed in this section throughout the software development process. By enhancing the interpretability and explainability of a system, it would improve the user experience. Furthermore, considering systems that process and classify natural phenomena such as changes in emotions and affective states (like this work), it is important that these systems are more *intelligent* and XAI/AXAI-capable, due to the sensitivity of the biometric information being used for analysis.

2.4 Emotions and the Concept of Affect

Human experiences are extremely complex and changes in mood and emotion are influenced by various factors that impact a person’s state. This complexity highlights the difficulty of training and developing truly intelligent ASA and emotion recognition systems. Even in human-*human* interactions, without knowledge of internal and external contextual variables, it is impossible to understand how a person is feeling. Likewise, in human-*machine* interactions, an intelligent agent is unable to have a holistic understanding of a person’s state without an advanced working memory and multimodal peripheral devices. The development of a truly affect-aware system must be considered a remarkable feat and an evolutionary step for AI systems.

There are various discrete and continuous affective models in literature that aim to describe and quantify the human experience [80-83]. One of the most common, and heavily cited models in literature is Paul Ekman’s six basic emotion model [80] – an argument for the existence of universal emotions that are irrespective of cultural background. These universal states were: (i) anger, (ii) disgust, (iii) happiness, (iv) sadness, (v) fear, (vi) surprise.

However, while Ekman’s theory is heavily cited, there are other affective state models presented in literature that offer continuous, spectrum-based solutions that describe states as being fluid and multidimensional – an argument that is representative of the human experience. While a discrete model like Ekman’s may describe universally present states, compressing such complex experiences into six discrete states fails to account for the dynamics of human expressions – especially given the way we express ourselves while experiencing these different states is unique to each person.

The Circumplex model of affect for example, plots states on a plane of valence vs. arousal. With valence describing the ‘positivity’ of the affective state and arousal being indicative of the excitation or activation level [81]. Analysing the circumplex model in Figure 5 we see some states like happiness, surprise and sadness that are consistent with Ekman’s universal basic emotion model. However, the ability to analyse states on a spectrum through the likes of the circumplex model proposes the literary foundations for dynamic classification. In addition, Russell and Mehrabian’s Three-Factor theory of emotions described in [82] is an extension of the circumplex model with respect to some of the state dimensions. However, this solution also explores a third dimension of ‘dominance’ which describes a level of control [82]. This three-dimensional interpretation of states further evidence continuous solutions as opposed to discrete ones and provides an additional argument for dynamic classifiers.

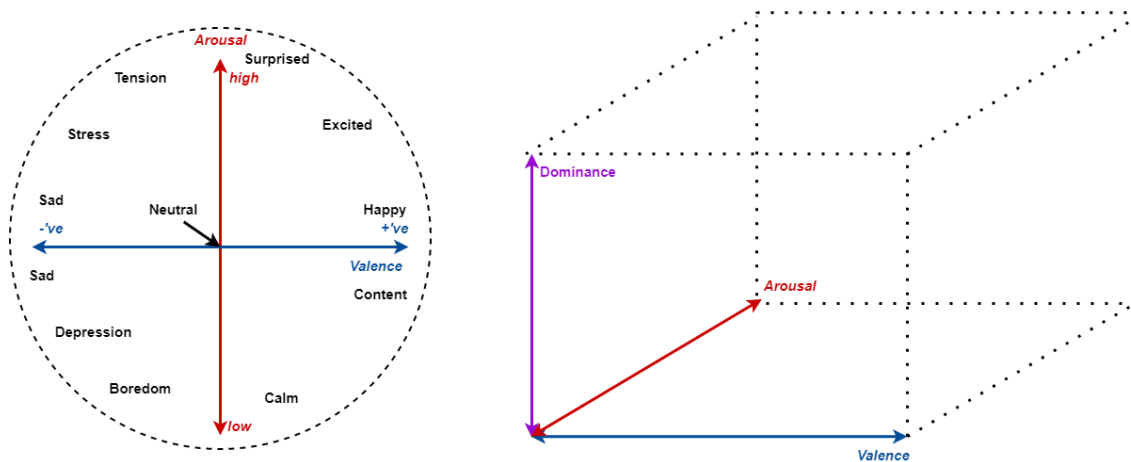


Figure 5: (LEFT) Visualisation of the circumplex model, highlighting the two axes of valence (unpleasant→pleasant) and arousal (low activation→high) [81]. (RIGHT) Conceptual Visualisation of Russell and Mehrabian’s Three-Factor Theory of Emotions [82].

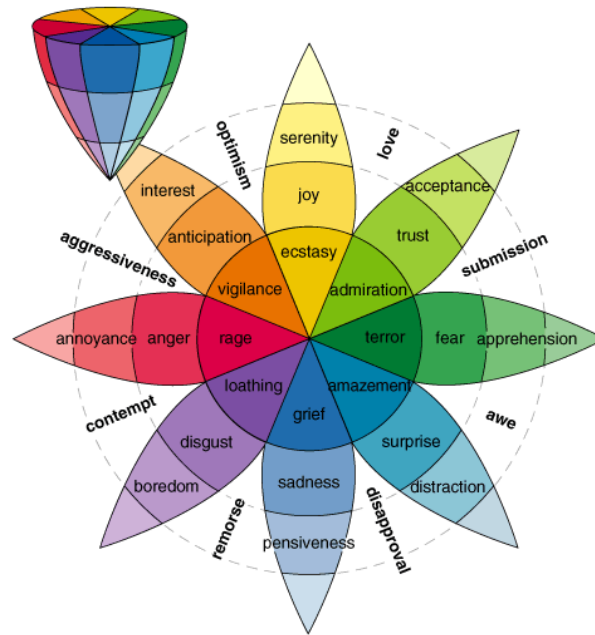


Figure 6: Visualisation of the Plutchik spectrum, highlighting the eight ‘petals’ of states. Intensity is inversely proportional to radius i.e.; more intense states are located at the centre of the figure. This image was extracted from [84].

The Plutchik Emotion Spectrum offers a third continuous assessment solution when considering human expressions of states. In his research, Plutchik’s eight emotions are formed under the basis of ten postulates, creating a three-dimensional conical model [83]. This model highlights states, similarities across states, and a third dimension described variation in intensity – similar to the arousal dimension used for the circumplex and three-factor models [81, 82].

One can argue that an ASA/emotion recognition system capable of measuring the intensities of affective states within a two/three-dimensional plane/space might prove to be more powerful than one that focuses on defining discrete affective states. This argument supports the idea of affective state expressions being subjective and dynamic in nature.

As introduced in Chapter 1 and evidenced across the relevant literature [20-23], affective state and emotion expressions have multimodal consequences, through signals that are both perceptible and imperceptible to human senses. These changes can be biometric such as: changes in facial expressions, speech patterns, blood pressure, skin conductance, electrocardiogram (ECG) signals as well as through other psychophysiological changes as well [20-23].

The multimodal nature of expressions further evidences the complexity of human emotions and resulting ASA systems. The relationship between emotions and robotics/AI is therefore an oxymoron when discussed at a philosophical level. This is due to emotions and the concept of affect defining something so innately erratic, instinctive and ‘human’, in comparison to robotics, which is generally described as being structured, learned, monotonous and with identifiable core principles. This highlights the interest and wonder of affective computing – a domain in which we can equip machines with tools to understand humans at a deeper, more complex level.

2.5 Affective Computing

Combining human psychology and robotics proposes fascinating applications i.e., trying to assess behavioural patterns that humans can oftentimes struggle with themselves. Through affective computing, ASA systems are capable of analysing human expressions of affective states using various input data sources in single- or multi-classifier solutions for the purpose of classifying a person’s mood, intent, or emotional state.

By exploiting unsupervised learning techniques, ASA systems can infer unique relationships, clusters and components that exist in datasets and data mixtures. Combining theory from Sections 2.1 and 2.4, the inference of statistical relationships and clusters can lead to the deployment of continuous, spectrum-based classifiers that account for variations in affective states. Then, by utilising supervised learning techniques on the newly labelled data, these systems can be used for dynamic classification tasks as has been achieved through the State Assessment Module being proposed in this work.

This research focuses on facial expressions and speech input signals for ASA. The Facial Action Coding System (FACS) has provided a literary foundation for many modern facial expression recognition research endeavours. This tool allows for the analysis of facial muscle movements to discern between different affective state expressions [85]. The Emotional-FACS (EMFACS) was then developed as a means of applying quantitative tags when describing Ekman’s universal emotion model [85, 86]. A tabular representation of EMFACS and the associated muscle movements and facial region is shown in Table 2. The FACS was cited and exploited in this research when attempting to define rules for multi-level classification – looking at similar FACS present in state groups that would justify their presence in the same class.

State	Facial Region	Action Units	Physical Action	Muscle
Happy	Lower	12	Lip corners raised	Zygomaticus major
	Lower	26, 27	Open mouth	Orbicularis oris
	Upper	6, 7	Raised lower eye-lids	Orbicularis oculi
Surprise	Lower	26, 27	Open mouth	Orbicularis oris
	Upper	5	Raised upper eye-lids	Levator palpebrae superioris
	Upper	1	Raised eyebrows	Medial frontalis
Anger	Lower	23	Lip Tightener	Orbicularis oris
	Upper	5	Raised upper eye-lids	Levator palpebrae superioris
	Upper	6, 7	Raised lower eye-lids	Orbicularis oculi
	Upper	4	Eyebrow Frown	Corrugator supercilii
Disgust	Lower	9, 10	Raised Upper Lip	Levator labii
	Upper	6, 7	Raised lower eye-lids	Orbicularis oculi
	Upper	4	Eyebrow Frown	Corrugator supercilii
Fear	Lower	26, 27	Open mouth	Orbicularis oris
	Upper	1	Raised eyebrows	Medial frontalis
	Upper	4	Eyebrow Frown	Corrugator supercilii
	Upper	5	Raised upper eye-lids	Levator palpebrae superioris
Sadness	Lower	15	Lowered lip corners	Depressor anguli oris
	Upper	1	Raised eyebrows	Medial frontalis
	Upper	4	Eyebrow Frown	Corrugator supercilii

Table 2: List of facial action units associated with common expressions of states, highlighting their facial region and the corresponding muscle being activated [25, 86].

Regarding expressions of states through speech, this is split into paralinguistic and linguistic components, respectively describing fluctuations in acoustic features (how something is said) and the content and structure of the utterance (what was said) [1, 19, 87]. Together, the two speech channels are used to express changes of state in speech. This work applies the classification of facial expressions and paralinguistic speech. Previous work saw the classification of linguistic expressions [1, 19] but without the ability to properly validate the approach, this was omitted for this work.

Paralinguistic speech features provide a quantitative way of analysing affective speech. However, there are various examples of paralinguistic feature sets that exist in literature that have been used for affective speech classification tasks [87-89]. In previous research for example [19], a custom feature set was derived for paralinguistic classification, but there are more established sets such as the Computational Paralinguistic Challenge 2016 (ComParE2016) feature set [88] and the Geneva Minimalistic Acoustic Parameter Set (GeMAPS) [89]. Through this research, both GeMAPS and ComParE2016 were experimented with for feature extraction, experimenting with both low-level descriptor and functional feature levels using the openSMILE signal processing package [90]. This process will be discussed in more detail in Chapter 5.

Similar to the FACS and the EMFACS, researchers have attempted to apply tags to acoustic feature fluctuations that relate to changes in affective state. However, the difficulty in defining these features for a global audience is amplified due to social, cultural, and anthropological factors. For example, displaying states of anger are different when speaking to a child vs. speaking to an adult, or in a social environment vs. a professional environment. Therefore, the tags would only be generalisations based on the corpus of people being observed.

For creating the multi-level classifiers in this work, a survey was conducted on various human affective speech characteristics literature to determine how speech could be characterised in human-readable terms rather than showing changes in acoustic parameters that would be too mathematical and unclear. Klaus Scherer is one of the leading researchers in paralinguistics and the expressions of emotions through speech. Most of the literature that had been surveyed for this research, cited, and built upon Scherer's works including [91-93], which were used to define the interpretable speech assessment models. Table 3 reports the list of states and the vocal changes that are experienced (relative to neutral) as reported in [91-93]. In order to remain consistent across modalities, it is important that all assessable states are also consistent.

There are various examples of ASA systems in literature that utilise these methods [94-108], which are reported in Table 4 and Figure 7, highlighting various hybrid learning ASA approaches. Comparing the applications discussed in this survey and reported in Table 4 and Figure 7, it is evident that variations in classification accuracy are dependent on various factors including the modality, the chosen dataset and the method chosen. The applications discussed serve as a small sample size of various unsupervised, supervised and hybrid learning approaches applied in affective computing and ASA tasks.

While the surveyed systems are capable of providing users with classifiable labels and promising classification performances, many of these solutions lack accountability, comprehensibility, and explainable elements that would make users and the general public have greater trust in them. This reiterates the research gap discussed in Section 1.3. While ASA and ML systems promise high

performance levels, they are generally hamstrung by their inability to explain their decision-making processes and provide some sense of accountability to major stakeholders. Thus, highlighting the necessity of developing tools like the accountable, explainable artificial intelligence framework which will be introduced in the next chapter (Chapter 3).

Vocal Change	Anger	Happiness	Sadness	Fear	Disgust	Surprise
Speech Rate	Slightly faster	Faster or slower	Slightly slower	Much faster	Very much slower	Faster
Pitch Average	Very much higher	Much higher	Slightly lower	Very much higher	Very much lower	Higher
Pitch Range	Much wider	Much wider	Slightly narrower	Much wider	Slightly wider	Wider
Intensity	Higher	Higher	Lower	Normal	Lower	Higher
Voice Quality	Breathy, chest tone	Breathy, blaring	Resonant	Irregular voicing	Grumbled, chest tone	-
Pitch Change	Abrupt, on stressed syllables	Smooth, upward inflections	Downward inflections	Normal	Wide, downward inflections	Upward and fast
Articulation	Tense	Normal	Slurring	Precise	Normal	-

Table 3: List of vocal changes associated with common expressions, highlighting various interpretable changes in speech, based on acoustic characteristics defined in literature [91-93].

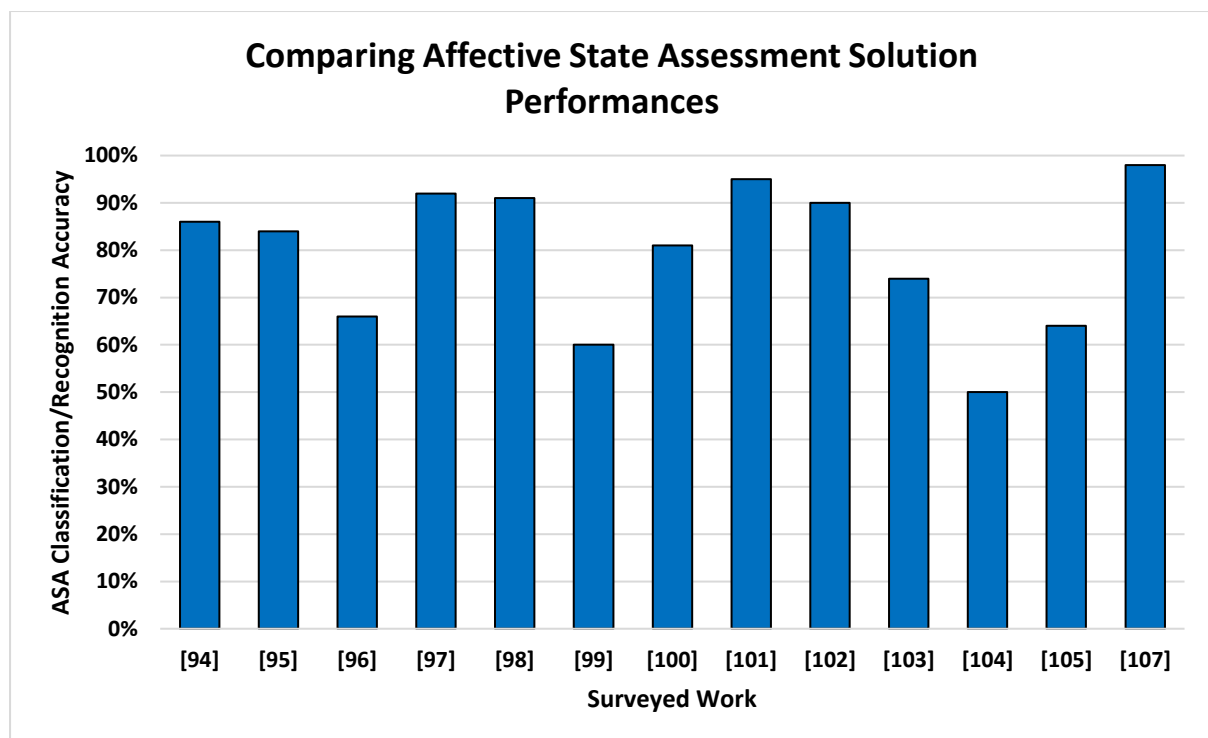


Figure 7: Highest observed ASA performance comparison for systems discussed in Table 4. Note that lowest observed accuracies are not present and [106] has been omitted due to it discussing CCC values as the comparative performance metric.

Work	Approach	Modality	Feature Description	Classification Accuracy, Dataset(s)
[65]	K-Means Clustering with a deep bi-directional long short-term memory (BiLSTM) network.	Speech	Key segments and high-level discriminant features	72% - 86% - tested on: IEMOCAP, EMO-DB, RAVDESS
[94]	Cascaded Gaussian mixture model and deep neural network (GMM-DNN)	Speech	26 Mel-Frequency Cepstral Coefficient (MFCC) features	84% - tested on: Emirati Speech Database
[95]	Gaussian mixture model vs. K-Nearest Neighbour	Speech	MFCC, wavelet features and subharmonic-to-harmonic ratio (SHR) pitch features	66% - tested on: EMO-DB
[96]	Gaussian mixture model	Speech	Temporal, frequency, and amplitude-related acoustic features and MFCC features	86 - 92% - tested on: Mandarin emotional speech database
[97]	Reports the Enhanced Lipschitz Embedding (ELE) algorithm	Speech	48 prosodic features plus 16 formant frequency features	ELE provided 26% increase in recognition accuracy relative to other dimensionality reduction methods. Tested on: Chinese Academy of Science Emotional Speech Dataset.
[98]	Comparison of dimensionality reduction techniques including Isomap and LLE. Using SVM, KNN and Logistic-regression for classification.	Speech	IS10 feature set, RQA feature set and a combination of IS10+RQA features.	47% - 60% - tested on EMO-DB, IEMOCAP
[99]	Reports the Enhanced Kernel Isometric Mapping (EKIsomap) algorithm. Classification using SVM.	Speech	48 prosody and voice-quality features including high-level statistical features	80.85% - tested on: EMO-DB
[100]	Reports the Many Graphs Embedding (MGE) dimensionality reduction algorithm. Maximum likelihood used for classification.	Facial Expression Images	Arnold-Transformed facial expression pixel features. Reduced to discriminative graph-embedded features	95.24%, 46% and 40% testing on JAFFE, MUG and CK+ datasets, respectively.
[101]	Fast Independent Component Analysis (ICA) and Granger Causality Analysis (GCA).	EEG Signal	Independent components and residual mutual information define clusters used for classification.	85% and 90% when experimented on two human subjects reacting to International Affective Picture System (IAPS) samples.
[102]	Reports the Hybrid ICA-Wavelet (ICA-W) Transform algorithm for artefact removal. Signals are decomposed using ICA prior to further separation via discrete wavelet transform.	EEG Signal	Statistical measures and spectral features are extracted from EEG signals post-separation and prior to emotion recognition.	Improvement of $\approx 10\%$ for statistical and wavelet feature classification relative to separation via pure ICA.

[103]	Blind Source Separation (BSS) to split music + speech audio mixtures prior to parallel emotion recognition using random forest classification and deep neural networks	Speech and Music Mixtures	1582 suprasegmental acoustic features (speech) and 2651 “Essentia” global features	Less than 50% recognition accuracy. BSS boosted classifier performance by 11%-14%.
[104]	Reports the use of denoising autoencoders to reduce the dimensionality for speech emotion recognition, with classification being done through SVM.	Speech	384 acoustic features based on the INTERSPEECH 2009 Emotion Challenge.	62.74% and 64.18% recognition accuracy when tested on the SUSAS and ABC datasets respectively. Performing the best out of other compared methods.
[105]	Compares a novel Ladder Network architecture to traditional autoencoder baseline architectures (among others) to compare classification accuracy under different dimensionality reduction techniques	Speech	6373 acoustic features based on the feature set introduced in the INTERSPEECH 2013 Computational Paralinguistic Challenge (Functional feature)	Valence: 0.289 – 0.458 Arousal: 0.761 - 0.803 Dominance: 0.687 – 0.746 Performance was measured using CCC values, validated and tested on the MSP-Podcast dataset.
[106]	Uses the Kohonen Self-Organising Map architecture for dimensionality reduction and unsupervised classification through the automatic grouping of data	Facial Expressions	16 Facial Action Units based on Ekman and Friesen’s Emotional Facial Action Units	98.13% average recognition accuracy when validated on three datasets: JAFFE, CK+ and MMI Facial Expression Database.

Table 4: Survey of applications that utilise supervised and unsupervised learning for affective computing and ASA tasks across various modalities and input signals.

Chapter 3: The Accountable, Explainable AI (AXAI) Framework

3.1 A Method for Improving Explainable AI

As emphasised previously, in comparison to traditional black-box AI and ML approaches, XAI promises more interpretable, explainable AI systems. By incorporating XAI capabilities during development, these systems would evidence an improvement. However, representing systems on a plane of comprehensibility vs. predictive accuracy as discussed in [11,13], means that factors of accountability are either domain-dependent or not considered when developing these systems [15]. Hence, the AXAI capability framework is proposed as a means of delineating systems in a space of comprehensibility vs. predictive accuracy vs. *accountability* [107].

By implementing domain-*independent* accountability measures, this would allow developers, managers, and other key stakeholders the ability to quantitatively assess AI/ML systems throughout the development process. Furthermore, while the development of a tool like AXAI framework may not be the finished article by any means, it does promise an unbiased assessment tool that takes into various elements of AI system design [107, 108] – serving as a valid piece of literature that may be built upon in the future, when designing more sophisticated assessment frameworks.

Assessing the accountability of AI systems is important when considering the ethical, legal, and social compliances [109, 110]. Cases discussed in [6, 17, 18, 110] serve as evidence to support something like the AXAI framework w.r.t to legislative and regulatory applications. Legitimised standards exist across all branches of industry that ensure that goods, services, and practices are “up to standard”. If implemented at regulatory level, something like the AXAI framework proposed in this work could provide a legitimate standard for AI systems.

Given the AXAI framework’s orthogonal vectors that will be discussed in the following sections, there is even the future possibilities of using the AXAI framework as an autonomous classification tool to detect the AXAI-capabilities of AI systems, using each of the parameters as inputs into a classification model. By gathering a corpus of assessed systems, this information can be used to train an autonomous AXAI classification system. The trained classifier could then be deployed throughout the development process by system designers and clients to score AI systems based on various capabilities in order to quantitatively rank and compare them. Again, discussing AXAI as a regulatory application, this may also be a way to consider if systems are up to standard prior to commercial or public use.

For the purpose of this research, we will discuss the construction of the AXAI framework vectors, and their sub-components as visualised in Figure 8. The AXAI framework contains nine elements that deconvolute AI and ML system explainability and performance metrics into three high-level vectors. All three high-level vectors i.e., accuracy, accountability and comprehensibility are mutually perpendicular with the same unit length [111]. Decomposing each high-level vector, we see that all are three-dimensional such that vector ‘A’ can be described as:

$$A = \sqrt{a_1\mathbf{i} + a_2\mathbf{j} + a_3\mathbf{k}} \quad (29)$$

where ‘ a_n ’ describe the magnitude of the n^{th} vector component.

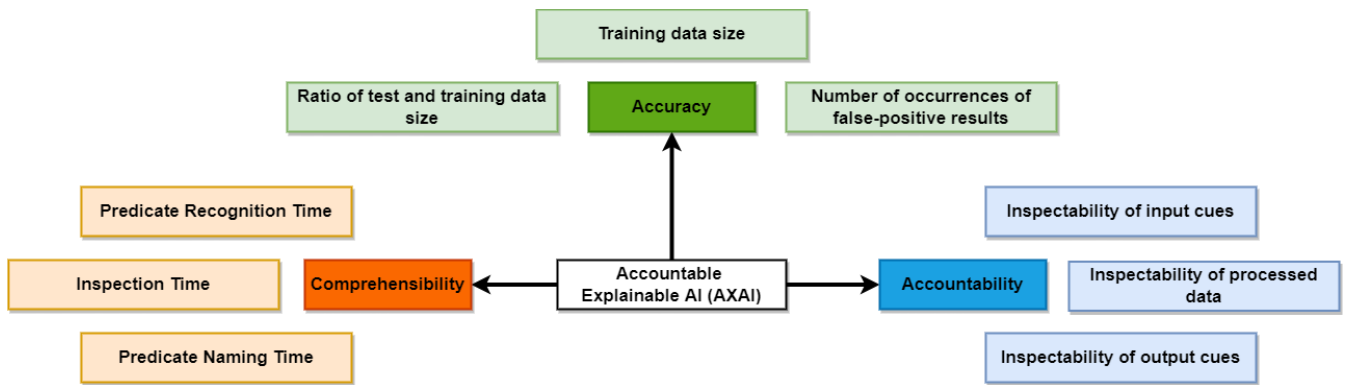


Figure 8: Breakdown of the AXAI framework showing comprehensibility, predictive accuracy and system accountability vectors making up the three high-level dimensional vectors, with each containing another three assessment vectors, totalling 3^2 dimensions for assessing AI and ML systems.

The AXAI framework is a novel invention that builds on the foundations of XAI and proposes the following, as reported in [107]:

1. An intuitive, domain-independent AXAI capability framework that accounts for existing XAI paradigms,
2. The ability to quantitatively assess the predictive accuracy of an AI system,
3. The ability to quantitatively assess the comprehensibility measures of an AI system,
4. The ability to quantitatively assess the accountability of an AI system in-production as well as post-production depending on stakeholder requirements,
5. Allows all parties involved the ability to visualise the AXAI capabilities of a system and examine areas of improvement, while also assessing the relevance of each element w.r.t the application of the system,
6. A foundation for future expansion/development of the AXAI framework and the ability to develop it as an AXAI-capability classification system and potential regulatory solution

Prior to discussing the AXAI high-level vectors, some initial assumptions and definitions need to be made, which are derived from, and inspired by literature [11, 112-118]. These definitions have also been reported in [107].

Let an AI system be a definite program ‘P’. This program will transform inputs→outputs over a series of internal steps/processes and may initiate decision-making [118]. In this work, P contains: (i) definite symbols, (ii) definite functions, (iii) definite propositions, (iv) definite predicates, (v) logical symbols, (vi) object variables and (vii) propositional variables. When discussing hypotheses, assumptions, and definitions in this work, let:

- \mathbb{k} = Constant,
- \mathbb{p} = Predicate symbol,
- \mathbb{S} = Human population, where ‘s’ defines a singular human in the population,
- \mathbb{V} = First-order variable,
- \mathbb{B} = Background knowledge, where a human ‘s’ with ‘ \mathbb{B} ’ is considered a definite program ‘P’,

- \mathbb{D} = Domain

Definition 1:

A predicate symbol ' \mathbb{P} ' – called in queries, is a part of the definite program i.e., $\mathbb{P} \in \mathcal{P}$. If the predicate symbol forms part of the background knowledge ' \mathbb{B} ' of each human ($s \in \mathcal{S}$), \mathbb{P} is considered *public*. Otherwise, \mathbb{P} is considered a *private* predicate symbol.

Definition 2:

Let \mathcal{P}_X be some arbitrary AI system. If \mathbb{B} of \mathcal{P} is that of $\mathbb{B} \cup \mathcal{P}_X$ forms, then the predicate symbol $\mathbb{P}_X \in \mathcal{P}_X$ becomes a predicate invention given that the predicate symbol \mathbb{P}_X was conceived in \mathcal{P}_X prior to becoming part of the background knowledge \mathbb{B} .

Definition 3:

Accountable, Explainable Artificial Intelligence (AXAI) capability is denoted by \mathbb{AII}_{AX} . The \mathbb{AII}_{AX} framework represents a 3D space that consists of three, independent vectors: \mathbb{C} = comprehensibility, P_A = predictive accuracy and S_A = system accountability.

The following sections discuss each of the AXAI vectors $\{\mathbb{C}, P_A, S_A\}$ and their corresponding vector components as visualised in Figure 8, including further definitions, assumptions, and hypotheses. The final section of this chapter attempts to visualise \mathbb{AII}_{AX} in a 3D space while showing experiments of deploying the AXAI framework to assess two ASA systems that have developed in the past.

3.2 Comprehensibility

Comprehensibility ' \mathbb{C} ' in the context of the AXAI capability framework is a criterion that allows for the assessment of a human's ability to understand the information being presented to them by an AI system that allows them to solve problems within the target domain. Comprehensibility is proportional to the mean readiness of a human in the population. Similar to Section 3.1, ' \mathbb{C} ' can be described through a set of definitions and consequent hypotheses [107].

Definition 4.0:

The comprehensibility ' \mathbb{C} ' of a definite program ' \mathcal{P} ' relative to a population ' \mathcal{S} ' can be represented as: $\mathbb{C}(\mathcal{S}, \mathcal{P})$, where \mathbb{C} is a three-dimensional vector comprising of:

- Inspection Time – T_{it} ,
- Predicate Recognition Time – T_{pr} ,
- Predicate Naming Time – T_{pn}

Therefore, using the base vector equation defined by (29):

$$\mathbb{C}(\mathcal{S}, \mathcal{P}) = \sqrt{(T_{it}^2 + T_{pr}^2 + T_{pn}^2)} \quad (30)$$

Through this definition, *naming* refers to establishing the "object-property" relationship and understanding the named objects based on AI/ML system observations. Thus, within this framework, ' $\mathbb{C}(\mathcal{S}, \mathcal{P})$ ' determines the mean readiness of a human ' s ' to use a program and apply the knowledge to an observation within the domain ' \mathbb{D} ' after inspecting \mathcal{P} for times $\{T_{it}, T_{pr}, T_{pn}\}$.

Definition 4.1:

The inspection time ' T_{it} ' defines the mean time that a human ' s ' (where $s \in \mathbb{S}$) will require to inspect and evaluate the information provided by the definite program prior to use the newly gathered information to complete a task within the relevant domain ' \mathbb{D} '.

Definition 4.2:

The predicate recognition time ' T_{pr} ' defines the mean time that a human ' s ' needs to assign a correct *public* name to a predicate symbol ' \mathbb{P} ' within the relevant domain ' \mathbb{D} '. This is in reference to Definition 1, which states that \mathbb{P} is public if it forms part of the background knowledge ' \mathbb{B} '.

Definition 4.3:

The predicate naming time ' T_{pn} ' defines the mean time that a human ' s ' requires to name a predicate symbol \mathbb{P} which has been output by the program as a privately named entity within \mathbb{D} . Citing Definition 1 again, a privately named predicate symbol is one that does not form part of the background knowledge ' \mathbb{B} '. Thus, T_{pn} assesses the time it takes to assign a public name to the initial *private* predicate symbol after interacting with the definite program.

Given the above definitions (4.1 – 4.3), we can define a few hypotheses regarding comprehensibility $\mathbb{C}(\mathbb{S}, \mathbb{P})$.

Hypothesis 1:

Comprehensibility $\mathbb{C}(\mathbb{S}, \mathbb{P})$ measures the mean readiness of a human ' s ' (where $s \in \mathbb{S}$) to consume the knowledge presented by a definite program ' \mathbb{P} ' such that they are able to complete a task within the applicable domain \mathbb{D} . Therefore, the larger the norm of the comprehensibility vector ' $|\mathbb{C}|$ ', the more comprehensible the AI program is.

Hypothesis 2:

The inverse of the mean inspection time introduced in Definition 4.1 i.e., ' $\frac{1}{T_{it}}$ ' is directly proportional to the presentation quality of the definite program \mathbb{P} . In layman's terms, the longer it takes for a human to inspect the signals, the lower the presentation quality.

Hypothesis 3:

The inverse of the mean predicate recognition time introduced in Definition 4.2 i.e., ' $\frac{1}{T_{pr}}$ ' is directly proportional to the ability to recognise and accurately assign a public name to a predicate symbol \mathbb{P} that has been presented by the definite program \mathbb{P} . Thus, the longer it takes for a human to assign a public name, the lower the recognition ability of the program.

3.3 Predictive Accuracy

Machine learning and AI systems have traditionally been assessed on the basis of their predictive accuracy and their ability to accurately make predictions. There are various measures for example:

validation accuracy, precision, recall, Cohen's Kappa, F_1 score and confusion matrices that have been used to assess the predictive capabilities of an AI system.

Predictive accuracy ' P_A ' in the context of the AXAI capability framework is directly proportional to the mean ability of a human ' s ' in a population ' S ' to correctly apply a public name to a definite program's privately presented predicate symbol \mathbb{p} . As in previous sections, P_A can be described through definitions and hypotheses.

Definition 5.0:

Let ' P_A ' of a definite program ' \mathcal{P} ' relative to population S within an applicable domain \mathbb{D} be represented by: ' $P_A(S, \mathcal{P})$ '. As discussed, P_A is a three-component vector containing:

- Ratio of model test data size and model training data size – $r_{tst-trn}$
- Training data size – d_{trn}
- Number of occurrences of false-positive results – O_{fp}

Using equation (29) for reference, $P_A(S, \mathcal{P})$ can be defined as:

$$P_A(S, \mathcal{P}) = \sqrt{r_{tst-trn}^2 + d_{trn}^2 + O_{fp}^2} \quad (31)$$

Thus, P_A defines the mean ability of a human user to convert a privately named/described predicate symbol \mathbb{p} into a publicly described entity belonging to applicable domain \mathbb{D} .

Definition 5.1:

The ratio of test and train data ' $r_{tst-trn}$ ' is related to the testing and training of the machine learning models that constitute an AI system \mathcal{P} . Within a domain \mathbb{D} , $r_{tst-trn}$ refers to the level of rigour applied when training and testing \mathcal{P} , such that it is more capable of correctly naming a privately named predicate symbol \mathbb{p} .

Definition 5.2:

The *absolute* training data size ' d_{trn} ' defines the sample size used for training \mathcal{P} w.r.t the domain \mathbb{D} . This is indicative of the breadth of samples that \mathcal{P} can exploit when attempting to correctly name \mathbb{p} . A larger value of unique samples in d_{trn} would indicate sample variation within \mathbb{D} that \mathcal{P} can call upon when attempting to complete tasks.

Definition 5.3:

Occurrences of false-positive ' O_{fp} ' names of ' \mathbb{p}_n ' is more closely related to traditional accuracy metrics used to assess AI systems in the past, where $n = \{1, 2, 3, \dots, N\}$ – classes. Within the AXAI framework, O_{fp} is related to the ability of \mathcal{P} to use trained models to predict new, unknown symbols that belong to the same, relevant domain \mathbb{D} . Therefore, the magnitude - $|O_{fp}|$, is indicative of the amount of error built into \mathcal{P} in regard to the naming of new predicate symbols ' \mathbb{p} ' that exist in \mathbb{D} . Through this definition, it is evident that O_{fp} is very similar to metrics like validation accuracy and precision.

Given the above definitions for predictive accuracy – $P_A(S, \mathcal{P})$, we can define some hypotheses.

Hypothesis 4:

Given the predictive accuracy, ' P_A ' measures the ability of a human user to convert a private described predicate symbol into a publicly named entity. The greater the norm of the vector $|P_A|$, the better the predictive accuracy of the definite program \mathcal{P} .

Hypothesis 5:

The test-train data ratio ' $r_{tst-trn}$ ' of a definite program ' \mathcal{P} ' is directly proportional to the level of rigour that was applied to the testing and training of the ML models contained within the definite program ' \mathcal{P} ' within the applicable domain \mathbb{D} . Thus, assessing the test-train ratio is an optimisation problem as an appropriate partition of testing and training data is important when building ML systems.

Hypothesis 6:

The volume of occurrences of false-positive naming of predicate symbols by a definite program \mathcal{P} will determine the system's built-in error within an applicable domain \mathbb{D} . Plain-speaking, O_{fp} is an indication of how often an AI system will make an incorrect prediction when attempting to perform a task. This is an important when considering applications where the cost of misclassification is high.

3.4 Accountability

The incorporation and assessment of system accountability ' S_A ' during the development cycle would improve the way AI systems are developed and highlights the novel difference between the proposed AXAI framework and other XAI models in literature. As mentioned prior, the critical tasks that some AI systems perform highlight the importance of accountability with respect to AI system design and implementation [14, 15].

System Accountability ' S_A ' in the context of the AXAI framework describes the mean accuracy with which a human ' s ' that belongs to a population ' \mathcal{S} ' can recognise various inspectable facets of an AI system \mathcal{P} , including: constants ' \mathbb{K} ', predicate symbols ' \mathbb{P} ', and variables ' \mathbb{V} ' that belong to a particular domain ' \mathbb{D} '. Like comprehensibility and predictive accuracy defined above, S_A can be described through definitions and hypotheses [107].

Definition 6.0:

Let the system accountability ' S_A ' of a definite program ' \mathcal{P} ' w.r.t to a population ' \mathcal{S} ' can be defined using: ' $S_A(\mathcal{S}, \mathcal{P})$ ', which is a vector containing three components that measure the 'inspect-ability' of an AI system \mathcal{P} :

- Inspect-ability of input models or cues – I_{in}
- Inspect-ability of processed, intermediate data – I_{pro}
- Inspect-ability of output models or cues – I_{out}

Recalling equation (29), system accountability can be defined as:

$$S_A(\mathcal{S}, \mathcal{P}) = \sqrt{I_{in}^2 + I_{pro}^2 + I_{out}^2} \quad (32)$$

Definition 6.1:

The inspect-ability ' I_{in} ' of input models or cues models the clarity of the input signals of a definite program 'P' from the point of view of a human 's'. Thus, given some observable definitions or entities, I_{in} is an indicator of the format and structure of inputs with respect to the relevant domain \mathbb{D} .

Definition 6.2:

The inspect-ability ' I_{pro} ' of processed data represents the clarity of intermediate program states or information as observed by a human 's' from a population \mathbb{S} . This is an important metric in regard to making black-box approaches more transparent, as processed observations are made available to a user prior to them becoming predicate symbols \mathbb{P} within a domain \mathbb{D} .

Definition 6.3:

The inspect-ability ' I_{out} ' of output data that is produced by an AI system 'P' describes the ability of P to show a clear, transparent representation of outputs. Furthermore, I_{out} represents the clarity in which a human observer can associate an AI system predicate symbol \mathbb{P} to an entity within an applicable domain \mathbb{D} .

As was the case when discussing comprehensibility and predictive accuracy, the above definitions all for the discussion of accountability-related hypotheses:

Hypothesis 7:

System accountability S_A refers to the ability of a human to associate constants, predicate symbols and variables provided by a definite program, with respect to input, processed and output data. Thus, the greater the norm of system accountability $|S_A|$, the more accountable a system is. In layman's terms, by allowing a user to be more aware of input, processed and output data, key stakeholders can more easily identify what party is accountable in terms of decision-making i.e., the user, the AI system, or the developer.

Hypothesis 8:

The inspect-ability of processed data – I_{pro} defines the interpretability of internal models and intermediate data stages of an AI system. Along with I_{in} and I_{out} , these metrics could very well be used to discern transparency of AI and ML models on the condition that it shows how data/signals are transformed from input forms to output predictions.

3.5 Design and Implementation of the AXAI Framework

The construction of the AXAI framework is justified and explained through Definitions 1 – 6.3 and Hypotheses 1 – 8 discussed so far in this chapter. Table 5 and Figure 9 summarise comprehensibility, predictive accuracy, and system accountability components, assigning them a "Score No." which is used in Table 6 to describe an example of how systems can be scored and compared using the AXAI framework as has been reported in the accompanying research paper [107].

By analysing Table 5 and Figure 9, we can paint a picture of how a multi-dimensional space like the presented AXAI framework can be used to describe and compare the AXAI capabilities of various systems. Guidelines and scoring metrics need to be established in order to assess AXAI capabilities,

but while the criteria outlined in Table 6 may not be perfect, it outlines a general foundation in which the AXAI capability framework may be improved in the future. As will be discussed, these guidelines proved to be a good indication when comparing unique ASA systems as reported in [107].

Score No.	Symbol	Definition	Measure
1	$ C $	Norm of the comprehensibility vector	Integer
2	$ P_A $	Norm of the predictive accuracy vector	Integer
3	$ S_A $	Norm of the system accountability vector	Integer
4	T_{it}	Inspection time	Score (Integer)
5	T_{pr}	Predicate recognition time	Score (Integer)
6	T_{pn}	Predicate naming time	Score (Integer)
7	$r_{tst-trn}$	Ratio of test data and training data	Score (Integer)
8	d_{trn}	Absolute training data size	Score (Integer)
9	O_{fp}	Occurrences of false-positive naming	Score (Integer)
10	I_{in}	Inspect-ability of input signals	Score (Integer)
11	I_{pro}	Inspect-ability of processed data	Score (Integer)
12	I_{out}	Inspect-ability of output cues	Score (Integer)

Table 5: The collection of parameters used to assess the AXAI capabilities of a definite program ‘P’ through the AXAI framework.

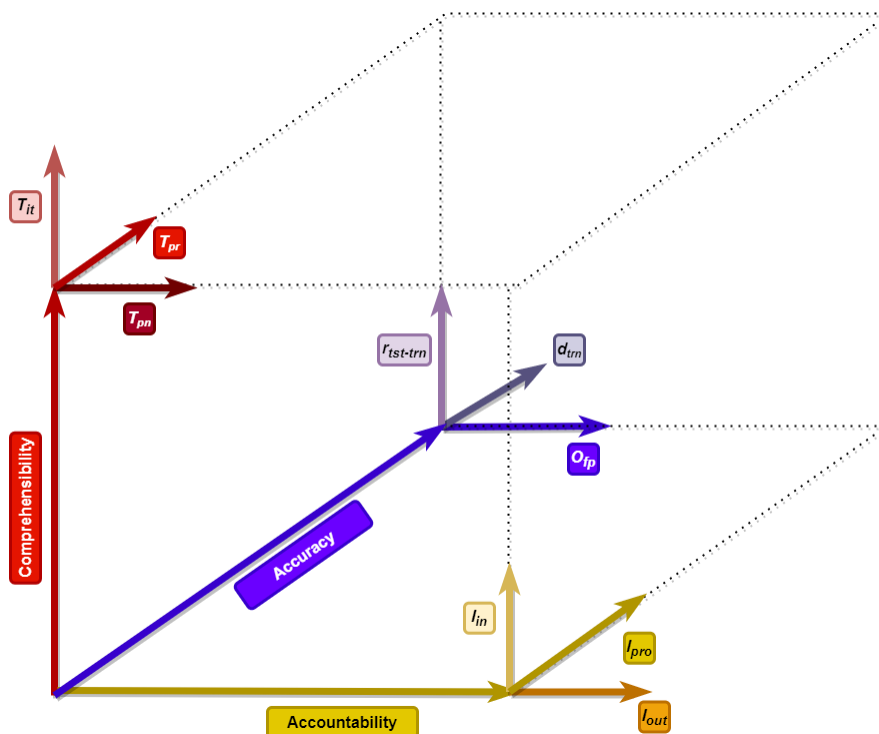


Figure 9: Visual representation of the AXAI framework highlighting predictive accuracy, accountability and comprehensibility vectors and their corresponding components.

Comprehensibility and system accountability measures are assessed based on the user experience and should be addressed iteratively throughout the software development process. Analysing Table 6, we see that while offering a solution for iteratively implementing AXAI in a system throughout development, it can also be used for post-hoc scoring of AI systems.

A test/survey was conducted on three systems that have been developed [107], the Affective State Assessment Module (ASAM) [1, 19, 108] – the predecessor to SAM, the two-step, dynamic assessment

of affective states and arousal levels system (DAASAS) [119], and the first version of SAM (SAM-V1), discussed in this work. To test the ASAM, ten qualified industry professionals volunteered to test the ASAM system on the basis of its AXAI capabilities. The volunteers came from Engineering, Science and Psychology backgrounds and were given a short briefing of the objective prior to being given a user manual showing them how to use the system. Users tested the system for approximately twenty minutes on average and provided scores for parameters 4-6 (comprehensibility) and 10-12 (accountability) using a 5-point scale as outlined in Table 6. These scores were then normalised to and converted to unit vectors such that they could be used to compare AXAI capabilities with the DAASAS system and SAM-V1 systems.

Score	Symbol	Scoring Criteria		
		0.0 – 1.0	2.0 – 3.0	4.0 – 5.0
4	T_{it}	Information appears to be very difficult and takes a long time to understand.	Information takes some efforts and time to understand.	Information is easy to understand with minimal efforts.
5	T_{pr}	It takes a very long time for a person to interpret an output.	It takes some time for a person to interpret an output.	A person would interpret an output very quickly.
6	T_{pn}	It takes a very long time for a person to use inferences to name a predicate symbol within \mathbb{D} .	It takes some time to use inferences to name a predicate symbol within \mathbb{D} .	Naming a predicate symbol using inferences within \mathbb{D} is very quick.
7	$r_{tst-trn}$	0: $0.0 \leq r_{tst-trn} \leq 1.0$ 1: $1.0 < r_{tst-trn} \leq 2.0$	2: $2.0 < r_{tst-trn} \leq 3.0$ 3: $3.0 < r_{tst-trn} \leq 4.0$	4: $4.0 < r_{tst-trn} \leq 4.5$ 5: $4.5 < r_{tst-trn} \leq 5.0$
8	d_{trn}	0: $d_{trn} \leq 5N_{class}$ 1: $5N_{class} < d_{trn} \leq 10N_{class}$	2: $10N_{class} < d_{trn} \leq 50N_{class}$ 3: $50N_{class} < d_{trn} \leq 100N_{class}$	4: $100N_{class} < d_{trn} \leq 1000N_{class}$ 5: $d_{trn} \geq 1000N_{class}$
9	O_{fp}	0: $O_{fp} \geq 50\%$ 1: $40\% \leq O_{fp} < 50\%$	2: $30\% \leq O_{fp} < 40\%$ 3: $25\% \leq O_{fp} < 30\%$	4: $10\% \leq O_{fp} < 25\%$ 5: $O_{fp} \leq 10\%$
10	I_{in}	Explanations of input data is not clear to the user	Explanations of input data is somewhat clear to the user	Explanations of input data are very clear to user
11	I_{pro}	Intermediate data and processing stages are hidden or cannot be interpreted by the user.	Some of the intermediate data and processing stages cannot be interpreted	Intermediate data and processing stages are transparent and can be interpreted
12	I_{out}	Output information is nondescript and hard to interpret	Output information is somewhat descriptive and quite difficult to understand and interpret	Output information is descriptive and easy to understand and interpret.

Table 6: Guidelines and scoring criteria for accountability, comprehensibility, and predictive accuracy AXAI framework vectors for determining the AXAI capability of AI systems.

The DAASAS is a dynamic assessment tool that classifies arousal levels using thermal, infrared images (TIRI's) that was designed as a statistical classifier without any embedded AXAI elements in its design [119]. This system analyses haemodynamic changes in facial expressions for classification using principal component analysis to detect the arousal levels. For assessing the AXAI capabilities of the DAASAS, eight volunteers trained in AI and ML assessed the AXAI-capabilities of the system after a brief introduction and after being informed of the objectives of the experiment. Similar to the ASAM, volunteers scored the DAASAS on the basis of its comprehensibility and accountability using a 5-point scale. The results of the surveys of the ASAM and DAASAS systems are presented in tables 7 and 8 respectively, with a comparison of the two being visualised in Figures 10 and 11.

An assessment of SAM-V1 using the AXAI capability framework is discussed in Section 5.4 after the system is introduced and described in more depth. This allows for an understanding of how the AXAI capability framework is beneficial when it is at the centre of the design and development process, as was evident when designing SAM.

Score No.	Sym-bol	Assessor ID										Average using 5-point score	Normalised Average Score
		1	2	3	4	5	6	7	8	9	10		
1	C												1.203
3	S _A												
4	T _{it}	4	4	4	4	5	3.5	4	5	4	2	3.95	
5	T _{pr}	5	3	3	4	3	3	4	4	1	2	3.2	
6	T _{pn}	3	3	3	2	4	3.5	3	5	2	2	3.05	
10	I _{in}	4	5	3	3	2.5	3	3	5	2	5	3.55	
11	I _{pro}	2	1	2	3	0	2	2	4	2	1	1.9	
12	I _{out}	3	4	4	3	5	4	4	5	3	3	3.8	

Table 7: ASAM user experience and assessor scores using the 5-point scale identified in Table 6. This table indicates the comprehensibility and system accountability of the ASAM.

Score No.	Sym-bol	Assessor ID								Average using 5-point score	Normalised Average Score
		1	2	3	4	5	6	7	8		
1	C										0.333
3	S _A										
4	T _{it}	0.5	1	1	0.5	1	1	1	1	0.87	
5	T _{pr}	0.2	0.2	0	0.1	0	0	0	0	0.06	
6	T _{pn}	0.5	1.5	0.5	1	2	1.5	2	2	1.37	
10	I _{in}	1.5	2	2	2	2	2	2	2	1.93	
11	I _{pro}	0	0	0	0	0	0	0	0	0	
12	I _{out}	1.5	2	2	1	1	2	1	1	1.43	

Table 8: DAASAS user experience and assessor scores using the 5-point scale identified in Table 6. This table indicates the comprehensibility and system accountability of the DAASAS.

Score No.	Symbol	ASAM		DAASAS	
		5-point Score	Normalised Score	5-point Score	Normalised Score
2	P _A	1.54606		1.3416	
7	r _{tst-trn}	5		5	
8	d _{trn}	4.055		2	
9	O _{fp}	4.2797		4	

Table 9: Comparison of predictive accuracy component scores for the ASAM and DAASAS AI systems using the scoring criteria outlined in Table 6.

Assessors are not tasked with deriving the predictive accuracy component scores as these will be known during the development cycle of each system. Given both systems (ASAM and DAASAS) had reached the end of their development cycles prior to assessment, the predictive accuracy assessment in this case was done as a post-production task. Table 9 outlines and compares the predictive accuracy component scores for each system.

The calculation of predictive accuracy component scores was done using the scoring criteria highlighted in Table 6 and when developing new systems, these could be assessed iteratively as machine learning models are changed and improved throughout the development cycle. Regarding

the ASAM, the test-train data ratio ' $r_{tst-trn}$ ' was kept as 80:20, an allocation that is consistent across machine learning literature – which as per Table 6 corresponds to a score of 5. As reported in [107, 119], $r_{tst-trn}$ of the DAASAS was 1:1 which also correspond to a score of 5.

Due to the multimodal nature of the ASAM, the training data size d_{trn} was averaged over both sets of training data (facial expression and speech classifiers), in which $N_{classes} = 7$ for both cases. For facial expression data, there were approximately 100 samples per class which corresponds to a score of 4/5 using Table 6. For speech data, there were approximately 200 samples per class which would score it at 4.11/5 when considering the range from a score of 4 to 5 using Table 6. Thus, by calculating the mean, the d_{trn} score for the ASAM as a whole was 4.055. The DAASAS in comparison, scored 2/5 for d_{trn} as reported in [107] as is shown in Table 9.

Validating machine learning models is an important part of the AI system development process. As mentioned in Chapter 2, there are various supervised and unsupervised machine learning methods that are more useful in particular classification tasks and may not be as useful in others. Thus, the software development process will oftentimes see variables changed from inception to release and as such, the validation results will change as the training parameters change. The false-positive naming occurrences ' O_{fp} ' correlate with the validation performances of a system. In the case of the ASAM, like the assessment of the ' d_{trn} ' component, O_{fp} needed to be averaged across facial expression and paralinguistic speech classifiers. Validation tasks reported naming errors of 22.71% and 18.9% across the two modalities which is an average naming error of 20.805%. Scoring this on a range of 0-5 using the criteria in Table 6, the O_{fp} score was calculated as 4.2797/5. For the DAASAS, as reported in [107, 119], the O_{fp} score was determined as 4/5.

Normalising the 5-point scores listed in Table 9, it allows us to derive the predictive accuracy scores of both the ASAM and DAASAS systems using equation (31) as follows:

$$\begin{aligned} P_A(\mathbb{S}, \mathbb{P})_{ASAM} &= \sqrt{r_{tst-trn}^2 + d_{trn}^2 + O_{fp}^2} \\ &= \sqrt{1^2 + 0.811^2 + 0.8559^2} \\ &= 1.54606 \end{aligned}$$

$$\begin{aligned} P_A(\mathbb{S}, \mathbb{P})_{DAASAS} &= \sqrt{r_{tst-trn}^2 + d_{trn}^2 + O_{fp}^2} \\ &= \sqrt{1^2 + 0.4^2 + 0.8^2} \\ &= 1.3416 \end{aligned}$$

Similar calculations were used to calculate the system accountability and comprehensibility vector norms using equations (30) and (32). Analysing the AXAI-capability scores reported in Tables 7-9, we can report some findings about both systems [107]:

- The predictive accuracy of both the ASAM and DAASAS systems are very good showing that they are effective w.r.t to their classification capabilities and the ability for users to use these systems for classification tasks in their relevant domains.
- The ASAM displays very good comprehensibility scores with the inspection time ' T_{it} ' being the highest scored parameter by users.

- In contrast, the ASAM’s predicate naming time was low $T_{pn} = 3.05$ and user feedback suggested that the predicate naming was far more difficult and time consuming and should be considered as something to improve in future works.
- The ASAM’s S_A vector norm was far lower than comprehensibility and predictive accuracy norms, specifically when analysing I_{pro} , which shows that the ASAM still struggles to show intermediate data stages and could be improved to make the intermediate stages more transparent to the user.
- The ASAM is an AXAI-capable system. However, there is still room for further improvement in regard to system accountability and predicate naming time, something that has been addressed through SAM.
- The DAASAS has a low level of comprehensibility as it was designed as a statistical classifier prior to AXAI being considered by the system authors.
- The DAASAS had a comparatively less than average system accountability score, again due to AXAI elements not being considered/conceived in the development process.
- Due to its nature as a statistical classifier, the predictive accuracy of the DAASAS remains very high and competitive.

Analysing the graphics in Figures 10 and 11, we can see that the ASAM outperforms the DAASAS as an AXAI-capable system. While the AXAI framework was not conceived in a formal manner at the time of development of the ASAM, the concept of XAI and interpretability of machine learning systems was under consideration throughout development and as such, this evidences why it outperforms the system it is being compared to. Through Figures 10 and 11, we can see how the AXAI capability framework discussed in this chapter can be used to delineate AI systems on the basis of their predictive accuracy, comprehensibility and accountability. Through a thorough investigation of logic programming and XAI models and terminology, this work proves the AXAI framework as a novel, important addition to the literary corpus, as evidenced through works like [107, 108].

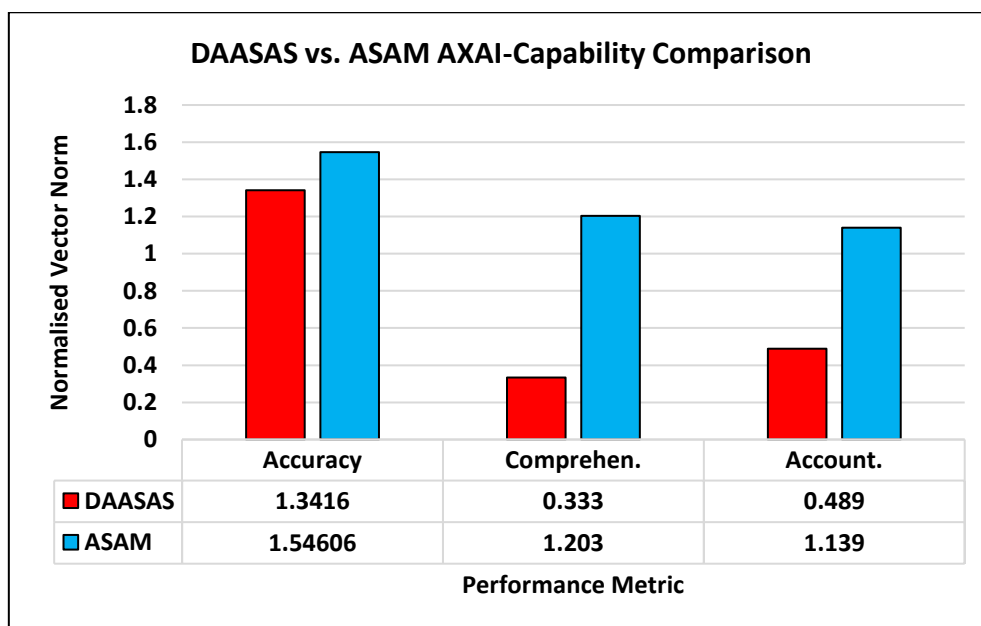


Figure 10: Comparison of the DAASAS (red) and ASAM (blue) AI systems using the AXAI capability framework performance metrics.

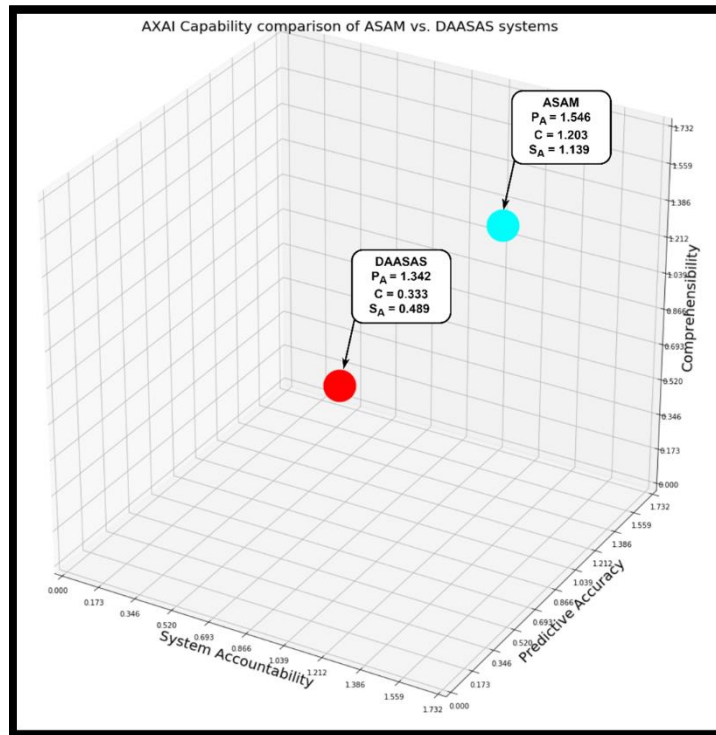


Figure 11: Visualisation of the AXAI capability framework when comparing the DAASAS and ASAM systems that have been assessed on the basis of their system accountability, comprehensibility, and predictive accuracy metrics.

Chapter 3 has shown how the AXAI framework can be used to quantitatively assess AI systems in post-production scenarios through the discussion of the ASAM and DAASAS systems. However, the AXAI framework also serves as a tool in which software design and development processes can be augmented such that software systems have an AXAI-focus throughout the development cycle, thus ensuring that all facets of explainability and transparency are covered, and the needs of all stakeholders are addressed. Embedding and assessing AXAI-capabilities throughout development ensures that gaps are not missed and that explainability and transparency are at the forefront of developers’ design processes – as was the case when designing SAM.

This framework and implementation task will be referred to again in Section 5.4 when the AXAI capabilities of SAM are compared to the likes of the ASAM and DAASAS systems as reported in [107].

Chapter 4: AXAI-Focused Software Design and Development

4.1 Software Design and Development Principles

Throughout the last century, various software design and development processes (SDDPs) have been conceived in an aim to optimise workflows, boost productivity, and make the system design process more efficient. While there are various solutions employed across various industries, there is no “industry-standard” SDDP.

There are many variables and constraints to consider when deciding on what SDDP method to use for a particular project including [120-122]:

1. The application, scope, and complexity of a project,
2. The size of the development team including designers, testers, and management,
3. The amount of man hours required to complete the project,
4. The timeframe in which you have to complete a project based on deadlines set by interested parties including but not limited to: self, team, upper-management, clients,
5. Budgetary constraints that may also influence other factors including man-hours, size of development teams and the scope of a project.

Some SDDP methods may favour some applications and systems over others. The variables and constraints listed above will also influence the choice of SDDP for any given project.

Regarding complexity and application for example, projects that require programmable logic controllers (PLCs) for the implementation of autonomous solutions may not require much iteration once autonomy is achieved, whereas AI solutions for social applications on average would require more iterations due to tasks such as training models and verifying performances in real-world environment. The PLC-based task may then require a more linear SDDP whereas the AI solution would require more iterative, dynamic approaches to the software development cycle [121].

The size of the development team is very important to consider when deciding on a SDDP. Ideally, the dynamics and number of iterations in a development cycle is proportional to the size of the team due to the number of man hours also being dependent on the size of the team. Having a larger team and delegating testing, development, and administrative tasks across members of the team is conducive to more iterative approaches, especially when the scope of a project is large.

The timeframe required to complete a task would determine the number of iterations planned throughout an iterative SDDP. If a project has a duration/deadline of nine months from the beginning of production and each iteration takes two months on average to complete, the maximum number of iterations ‘ $N_{iterations}$ ’ outlined at the beginning of production would be:

$$N_{iterations} = \left\lfloor \frac{T_{Months}}{T_i} \right\rfloor$$

$$N_{iterations} = \left\lfloor \frac{9}{2} \right\rfloor = 4 \text{ iterations}$$

where T_{Months} is the total time allocated for a project in months and T_i is the average time it would take to complete one iteration. However, the example discussed above is a very crude representation of iterative software design. Implemented SDDP methods are far more nuanced in their approach and

account for things like slack and the fact that iterations will not take the same amount of time – as was evidenced when designing and implementing SAM as will be discussed in Section 4.3 of this work.

Ultimately, all of the factors listed above share dependencies and are not mutually exclusive. The application and scope of the project would be dependent on the size of the development team which influences the number of weekly man hours available to complete the project, which influences, and is *influenced* by project deadlines and budgetary constraints. Thus, a thorough appraisal of project constraints needs to be done prior to choosing a SDDP method.

Some of the more popular SDDPs include: (i) the Waterfall model, (ii) Agile methods, (iii) Incremental models, (iv) Iterative models and (v) component-based development (CBD).

The waterfall model is one of the most traditional development models in which each sub-process in the model sequentially follows after the previous one [120-122]. While its linear design makes it easy to understand, there are various issues that arise when applying the waterfall model as the SDDP for complex software designs. The inflexibility of the waterfall design makes it difficult to implement for large-scale solutions or projects that have continually changing requirements [121, 122]. Hence, why dynamic, agile solutions are more preferred as modern SDDP solutions.

Agile methodologies arose in the mid 1990's as solutions to software design and development and is seen as a competitor to more traditional, linear models that were considered as singular, long processes rather than a combination of shorter development cycles [120, 122, 123]. The flexibility of agile methods is seen as a boon for the development process and the communication between key stakeholders as it allows for the reusability and improvement of assets and code from version-to-version while allowing stakeholders to maintain a continuous vision throughout the SDDP [122, 123]. Agile methods maintain the communication across key stakeholders including developers and users as expectations continuously change to meet specific needs.

Incremental and iterative models exist in a space between linear sequential models and agile models. To clarify, incremental and iterative SDDP models are semi-rigid (like linear models) in their process and have *some* flexibility (like agile) so that elements of a system can be changed at each increment/iteration [120, 122]. Through various versions that relied on goals, feedback, and reflections at each stage of development, SAM evidenced an incremental SDDP approach which will be discussed later in this chapter (see Section 4.3).

Component-based development (CBD) is as its name states, emphasising the importance of “components” in the software design process. Discussing CBD, components describe functional code that exist in a software solution [122, 124]. This approach is more flexible when compared to the traditional waterfall model as components can be developed, tested, and assessed in parallel throughout the development cycle prior to its assembly and deployment [124]. Component-based development would be very efficient in organisations where code blocks are reused across various business sectors thus, the reusability of code would make it a very attractive method to adopt. However, for more niche projects where code blocks are implemented in one-off software solutions, CBD becomes obsolete and as such, agile solutions should therefore be considered as an alternative.

Figures 12-14 visualise the waterfall model, an incremental model, and the standard agile model.

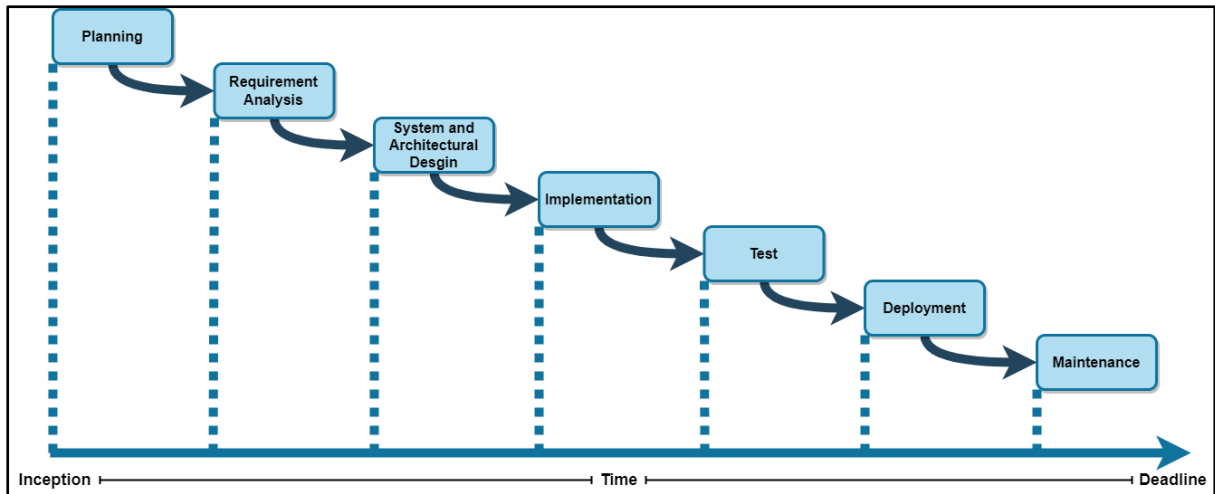


Figure 12: Traditional, waterfall model showing the linear sequential nature of each phase

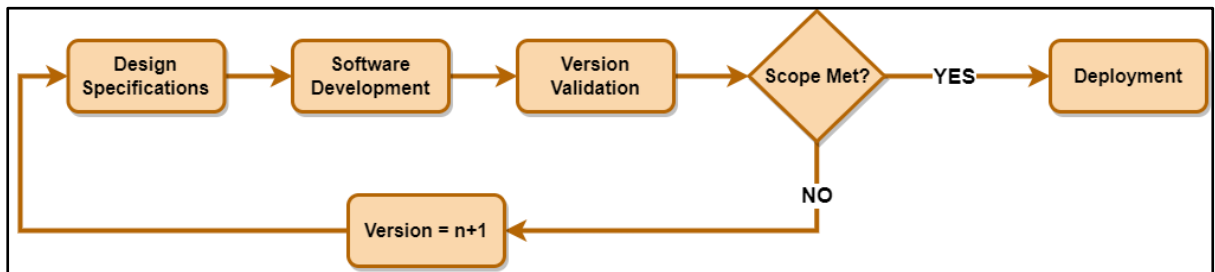


Figure 13: Incremental model showing the cyclical, version-based software design processes

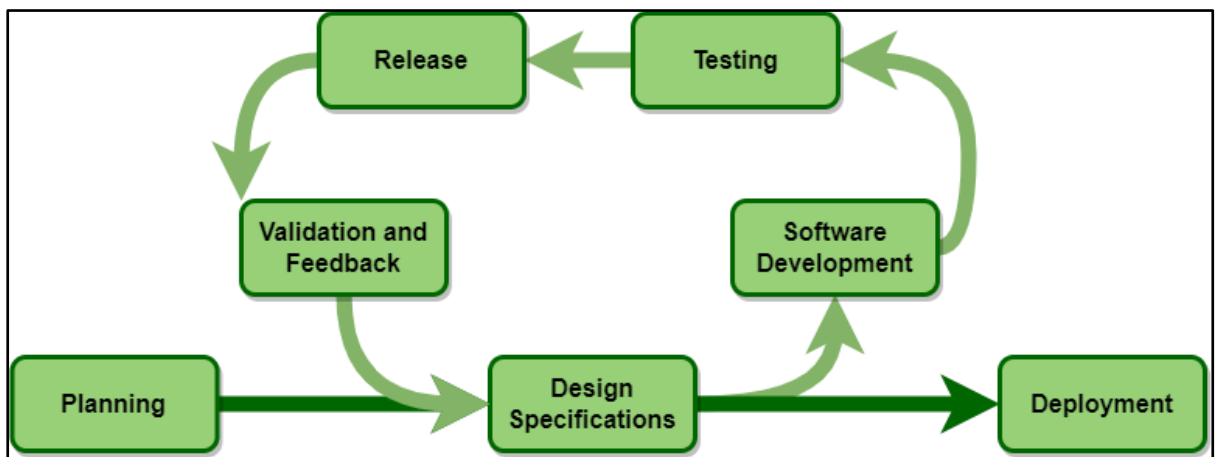


Figure 14: The standard agile model showing the continuous, dynamic approach highlighted by the circular flow in the middle of the figure which repeats until deployment.

4.2 AXAI-Focused Software Design

The SDDPs discussed in the previous section are general software development methods that are integrated across various independent sectors for to complete various different applications through both linear sequential models like the waterfall model and more dynamic, iterative processes like agile methods (and incremental and iterative models) [120].

The AXAI framework introduced in Chapter 3 discusses the implementation of AXAI capabilities into AI systems throughout the development cycle by addressing comprehensibility, predictive accuracy, and system accountability vector components. Rather than assessing AXAI as a post-production task, by integrating these components throughout the development process, it would ensure that the needs and wants of key stakeholders are addressed. Furthermore, it would ensure that the improvement of human-machine interactions and transparency is at the forefront of development.

For example, if the development team contained members responsible for testing and quality control, by adding an “AXAI Assessment” task at the end of each iteration, you could quantitatively track the predictive accuracy, comprehensibility, and system accountability performance metrics at each stage of development. Along with general feedback, this knowledge of AXAI components could then influence *future* phases as AI system developers would therefore know how to improve AXAI capabilities using this generalised framework.

As expressed, the advantages of implementing AXAI into the SDDP include:

1. The ability to track and quantitatively assess AXAI components throughout development,
2. The ability to improve AXAI capabilities through feedback at each iteration,
3. It ensures that transparency and improving human-machine interactions remain at the forefront of development,
4. If competitors or similar systems have used the AXAI framework, it would allow developers to quantitatively compare their system with another on the basis of $\{P_A, S_A, C\}$,
5. The ability to track and document AXAI capabilities throughout development, leading into production and commercialisation could provide some kind of legal foundations as it would allow for the establishment of the chain of responsibility (accountability) in the event that something goes wrong once the system is deployed.

However, there are obvious challenges in terms of implementing AXAI into the software design process, since the AXAI framework is not perfect in its current form. One of the clear challenges would be ensuring that proper testing of AXAI metrics is conducted. Like other testing processes, one would need to ensure that fair testing conditions are abided by and as such, placing faith in system developers and testers to be honest in their assessments if the assessment is not conducted by an impartial, independent body.

Another challenge would be getting the AXAI capability framework accepted as a general assessment tool for AI systems. Currently, the AXAI framework has only been tested on three systems that were easily accessible. It would take years and various iterations for a system like the AXAI framework to be considered an acceptable regulatory/standard tool for developers. Unless it is adopted by many, its impact will be limited to a niche audience. Thus, traction of the AXAI framework is pivotal to its success, improvement, and further implementation.

4.3 Software Design Process of the State Assessment Module

The State Assessment Module (SAM) is an affective state assessment solution that focuses on integrating AXAI into its core functionality to improve its usability, transparency, comprehensibility, and accountability, thus serving as an example of how human-machine interactions can still be improved. AXAI capabilities were considered throughout the development of the system and feedback of its predecessor – the ASAM was used for further improvements [1, 19, 108]. However, SAM was

largely designed and developed from the ground up given its functionality and GUI is far more sophisticated in comparison to previous works.

The SDDP of SAM can be considered as an iterative process leaning more towards an agile model than a traditional waterfall model as visualised in Figure 13. The reason for this was to maintain some of the rigidity and structure of a waterfall model while maintaining the flexibility of agile solutions – which was achieved through a thorough version control process. The ultimate scope of SAM was identified prior to development, but throughout the development process, additional functions were conceived and listed as design ideas to be implemented – which highlights the need for a flexible solution.

As discussed, AXAI was a major focus of the design and at each version, elements were added/addressed to improve the comprehensibility, accountability, and predictive accuracy of the system. A description of the versions of SAM is outlined in Table 10 and visualised in Figure 15. Note that Table 10 does not account for the literature review and theoretical research that predated the development of SAM, this information is reported in depth in Chapter 5 along with in-depth discussions on computational flowcharts and high- and low-level system designs

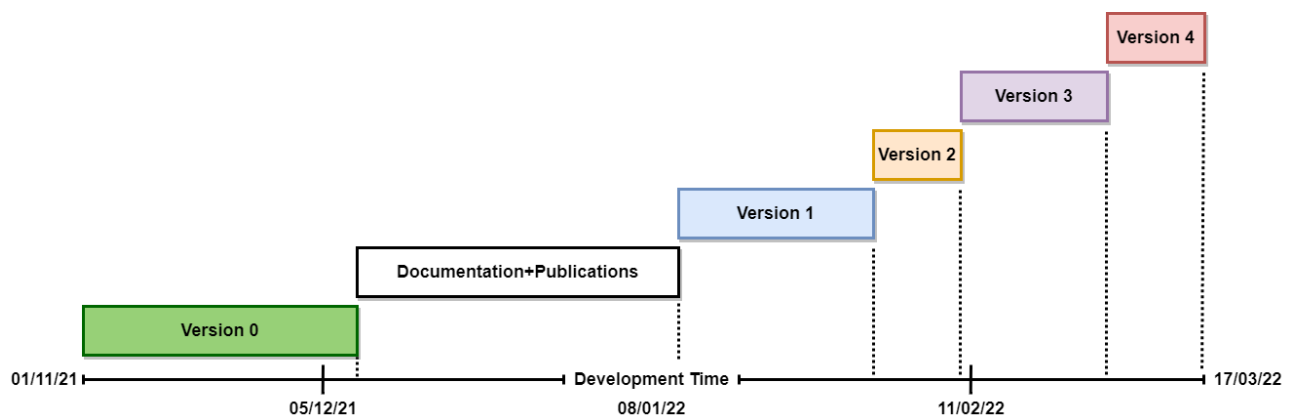


Figure 15: Timeline of SAM’s development, officially starting 01/11/2021 and finishing on 17/03/2022. This is a visualisation of the tasks outlined in Table 10 and does not account for prior theoretical research that was undertaken prior to the development of back-end programs in Version 0.

Analysing Table 10, Figure 15, and Appendices A-E, we see the construction of SAM from its foundations, all the way to it being a fully functional, AXAI-capable, embedded system with real-time, dynamic ASA capabilities. As was initially hypothesised, the back-end development took the longest amount of time relative to other versions.

Version 1 mainly consisted of GUI and asset development. Considering a new GUI development package was used to design SAM, it was expected that this version took longer to implement than versions 2, 3 and 4. Version 2 was the quickest iteration but was the most effective in regard to improving the predictive accuracy of the system in real-time. Interestingly, at the time of development, version 3 felt like the most difficult and time-consuming considering the number of new functions and changes that were added into the system at this point, however we see that it was shorter than version 1. Finally, version 4 was the final software development phase and saw the integration of text-to-speech and speech-to-text communication.

Version No.	Start Date	Specifications	Implementation and Notes	AXAI element addressed
0	01/11/2021	Design and training of the paralinguistic speech classifiers, including the design of the classification tree and clustering of classes	Two datasets were experimented with, the Ryerson Audio-visual database of emotional speech and song (RAVDESS) [124] and the Surrey Audio-visual expressed emotion (SAVEE) dataset [126]. This task also involved testing various feature sets and levels including ComParE2016, extended GeMAPS and the emobase feature set using low-level descriptor and functional feature levels, experimenting with various classifiers including MLP and SVM models, being combined with LDA feature extraction. The classification tree and ALL classification accuracies were finalised and LDA clustering was used as justifiable evidence for the formation of clusters.	Predictive Accuracy
0	05/11/2021	Design and training of the facial expression classifiers, including the design of the classification tree and the clustering of classes	The RAVDESS dataset [124] was also used for training and testing of the facial expression classifiers, using 150x150, 100x100 and 50x50 training image data sizes. The 150x150 and 100x100 training images were too much data for training and resulted in “memory errors” or slow performances and as such the 50x50 samples were chosen as they optimised the number of raw features vs. classification accuracy. SVM and MLP models were experimented with at this stage of training. Again, the classification tree and ALL classification accuracies were finalised and LDA clustering was used as justifiable evidence for the formation of clusters.	Predictive Accuracy
0	10/11/2021	Identification of feature similarities that apply a theoretical foundation for why states have been clustered (for facial expression and speech)	Using [85, 86, 91-93], the theoretical justification was made for the clustering of data, using an evidence-based clustering approach to support the initial separation and clustering of data. The groups were finalised at this stage and the feature similarities were documented.	System Accountability, Comprehensibility
0	12/11/2021	Derivation of the first “official” version of the classification logical flowcharts (for facial expression and speech).	With the first iteration of the classifiers trained, the flowchart needed to be finalised – outlining the clustering of data and the purpose of each “depth” of classification. This flowchart outlined what classes were present for ALL classifiers at each stage of classification across both modalities.	Comprehensibility
0	16/11/2021	Development of the dialogue for all classifier outputs, for both speech and facial expression modalities.	SAM incorporates “classification depth” for both modalities, where there are four levels of classification – corresponding to each state group. For all classifiable states in every classifier contained within SAM, there is a dialogue output attached that describes what is being “seen” or “heard” by the system.	System Accountability, Comprehensibility
0	18/11/2021	Primary GUI design and planning	Prior to developing SAM, the GUI needed to be designed and planned to ensure that GUI and functional programming was efficiently done. Initially, eleven windows were designed, with this number increasing proportionally to the complexity of the system.	Comprehensibility
0	01/12/2021	Development of the classifier model and dialogue loading functions	Writing up the code (component-based development) that would be used for loading machine learning models and dialogue .csv files into interpretable data structures.	Predictive Accuracy
0	09/12/2021	Retraining of LDA models	This task was necessary as the LDA models initially imported resulted in a memory issue as they were using eigen solvers for clustering which resulted in larger file sizes. This issue was resolved by converting all LDA models from eigen → singular value decomposition (SVD) solvers, which reduced the file sizes and removed the memory fault.	Predictive Accuracy
1	10/01/2022	Digital design and planning of GUI	Conversion of the hand-drafted GUI window designs into a digital format to further establish the design of SAM’s GUI.	Comprehensibility

1	10/01/2022	Integrated dialogue functionality	A major improvement in the development of SAM vs. the ASAM developed previously was to be the compartmentalisation and structuring the code in a modular fashion. The “dialogue” subdirectory was the first example of this and contains all dialogue prompts and dialogue-related functions.	Comprehensibility
1	13/01/2022	Development of “Home Screen” GUI window	The home-screen is to be opened upon execution of the program – it allows users to monitor the raw input information including: (i) Live camera feed, (ii) Live audio input feed, (iii) audio transcription capabilities. All data on the home-screen should be reflective of what SAM is sensing in real-time and a bounding box should be displayed around a face to show what exactly SAM is using for classification. The home-screen should also have a status box to alert users of what the current status is in real-time. The home-screen should also serve as hub to access other GUI windows.	System Accountability
1	17/01/2022	Implementing Home Screen button functionality and improving home-screen	The home-screen as a standalone window only shows input data to the user thus, only addresses system accountability. The user needs to be able to access other GUI windows to view more data screens which is done through SAM’s GUI buttons. The buttons open new windows that exist on child threads (where the home screen is the parent thread), with the buttons being labelled according to what data/window the user wants to view.	Comprehensibility
1	20/01/2022	Integrating Level 1 classification	Level 1 classification for both facial expression and speech classifiers considering the change in the level of activation from neutral → low → high. Both windows are very similar and code blocks were used for both of them – with variables being changed to be indicative of the modality. Being the first level of classification, the development of both windows involved the processing of input data and the extraction of features. Users are also capable of outputting log information through button presses on the newly created windows.	Predictive Accuracy, System Accountability, Comprehensibility
1	23/01/2022	Integrating Level 2 classification	Due to different modalities, level 2 classification represents different clusters for facial expression and speech and different representations of data. For facial expressions, level 2 classifies the state group (and groups of muscles) and for speech, level 2 classifies temporal phase (and the change in speech patterns in terms of onsetting, apex, or returning to neutral).	Predictive Accuracy, System Accountability, Comprehensibility
1	26/01/2022	Integrating Level 3 classification	For facial expressions, level 3 classifies the discrete state and detects the action units present for a predicted state. For speech, level 3 classifies the valence and the acoustic feature groups present for high and low valence states.	Predictive Accuracy, System Accountability, Comprehensibility
1	27/01/2022	Integrating Level 4 classification	For facial expressions, level 4 classifies the temporal phase and shows if muscles are slightly activated, activated, or returning to normal. For speech, level 4 classifies the state and the acoustic feature changes that correspond to the predicted state	Predictive Accuracy, System Accountability, Comprehensibility
1	28/01/2022	Consolidation and validation of SAM version 1	End of SAM V1, the code was executed to check that it was in a working condition. All input data signals were received in real time. The software was modular. All GUI windows updated GUI assets including tabular cell information, animations, exporting logs, opening logs, updating dialogue, integration of ‘verbose’ setting when needed. Back-end models designed in V0 were all integrated into V1. Design specifications were outlined for Version 2 and a manifest/summary file was produced for V1 (See Appendix A).	-

2	29/01/2022	Improving real-time facial expression and speech classifier performances	While validation accuracies achieved when training the models (in version 0) provided promising results, this was not conducive to real-time performances and as a result, the classification models were revisited (classifier ensemble logic and trees remained consistent). For facial expression classification, the LDA and Scaler approach was abandoned in favour of a 2-layer CNN. Regarding paralinguistic speech, the features were changed to be consistent with the feature used for separation i.e., ComParE2016 LLDs.	Predictive Accuracy
2	31/01/2022	Implementing report generation functionality	SAM should be capable of outputting classification results to users in an interpretable manner that highlights how it performed during an interaction. The decision was made to not show the graphical data over time as a result of feedback from people that interacted with the ASAM in the past, who suggested an overabundance of graphical information did not assist in making the system more explainable. Thus, SAM allows users to generate reports from live data or historical data for all classification levels whenever they please.	System Accountability, Comprehensibility
2	09/02/2022	Consolidation and validation of SAM version 2	End of SAM V2. This version improved the predictive accuracy of the facial expression and speech classifiers and implemented the report generation functionality, thus improving system accountability and comprehensibility AXAI components. Design specifications were outlined for Version 3 and a manifest file was produced for V2 (See Appendix B)	-
3	10/02/2022	Integrating face mask detection functionality	A parallel research effort reported in [25] discussed an adaptive system that accounted for the presence of occluded facial expressions in light of the COVID-19 pandemic. The thought process was that this system would be capable of showing this theorised functionality in practice. As such, SAM was integrated with a binary mask detection classifier that detected the presence of face masks and disabled facial expression classifiers if a mask was detected.	Predictive Accuracy, System Accountability,
3	10/02/2022	Handling removed peripheral devices	Being a real-time system, SAM should be capable of handling instances of peripheral devices being removed without causing the system to break. This was handled through “try, except” loops that attempted to reconnect the camera and microphone when disconnected and disabled GUI windows if a camera was not detected, directing users to SAM’s issue.	System Accountability
3	15/02/2022	Implementing report generation for the Home Screen – Interaction Analytics	SAM should be able to monitor how the user interacts with it by timestamping user button presses and recording transcription and video camera input data. The decision was made to display this via a report similar to the reports generated for the individual GUI windows.	System Accountability, Comprehensibility
3	17/02/2022	User and subject authentication capabilities, creation and viewing of databases through SAM’s GUI	SAM must be able to understand who is using the system (USER) and who it is analysing (SUBJECT). This is not a natural understanding like humans who gather this information from speech or facial expression data, SAM does this through a login functionality. SAM users should also be restricted on the basis of their permissions and as such, USERS are granted permissions to be able to view certain data – proof of concept with three arbitrary levels. SAM should also understand the information it uses to make decisions – example of <i>working memory</i> , to understand more about itself such as who it knows (the users and subjects in the database). This will be discussed further in later sections.	System Accountability

3	03/03/2022	Consolidation and validation of SAM version 3	End of SAM V3. This version could have been deployed as two separate versions given the extensive amount of work and man-hours required to complete. In summary, this version integrated the mask detection, peripheral device disconnection handling, report generation for the home screen and the user and subject capabilities. Design specifications were outlined for Version 4 and a manifest file was produced for V3 (See Appendix C)	-
4	04/03/2022	SAM Interaction – making use of speech-to-text	SAM should have the ability to communicate with its users and subjects, showing that it has some form of intelligence. SAM has 21 commands that have been appended to the dialogue database that allow it to show off its various capabilities including: instant classification, understanding of its classification trees, understanding of the user and subject, understanding its purpose and what it is, a sense of time and a working memory, understanding of what it is looking for in speech and facial expressions, and the ability to debug itself and shut it down	System Accountability
4	14/03/2022	SAM Interaction – making use of text-to-speech	SAM should be able to actually communicate – giving it “a voice”. Through synthesised dialogue and text-to-speech, SAM is able to verbally communicate all of the commands and dialogue options that is contained in its memory. It transforms into more of an assistant in the mould of Siri, Alexa and Cortana that exist on personal computers and mobile devices.	System Accountability, Comprehensibility
4	16/03/2022	Consolidation and validation of SAM version 4	End of V4. This is the last major addition to the software capabilities of SAM. Through version 4, SAM was given the ability to communicate with users and subjects and have a more sophisticated understanding of its purpose, inputs, outputs, and a working memory. Design specifications were outlined for Version 5 and a manifest file was produced for V4 (See Appendix D)	-
5	17/03/2022	Giving SAM a body	Version 4 saw the end of the software development process and SAM was then ported onto a LattePanda development board. Through some stress testing and maintenance, some of the software was edited, but nothing noteworthy. Version 5 sees SAM deployed as a cost-effective, energy efficient, embedded system capable of being integrated into any existing framework (robotic or not), with the purpose of granting the host machine AXAI-capable, dynamic state assessment. A manifest file for version 5 is shown in Appendix E	-

Table 10: Summary and description of the iterative software design and development process (SDDP) of the State Assessment Module (SAM). Through a version-based, iterative approach, SAM has been developed from the ground-up into an AXAI-capable affective state assessment system, which was developed over 137 days, amounting to approximately 1100 man-hours (from 01/11/2021). This calculation of days and man-hours does not consider the time and effort prior which involved research and literature reviews as the overall research project began in January 2020.

While Figure 15 highlights the time taken to build the system and table 10 outlines how AXAI capabilities were improved throughout the SDDP, they do not show the size and complexity of SAM in terms of computation and lines of code. Figures 16 and 17 visualise the size and complexity of SAM over time, including subsystems.

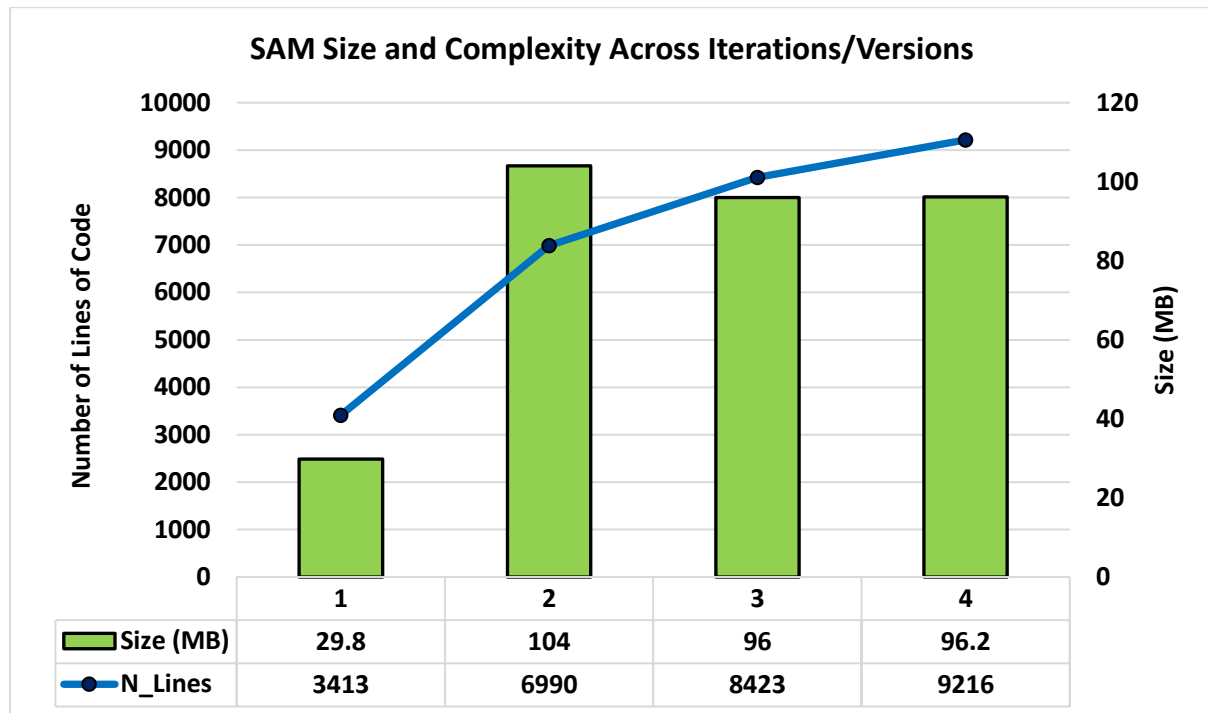


Figure 16: SAM size and complexity over time as a function of the number of lines of code at each version and the size of the deployed package in megabytes

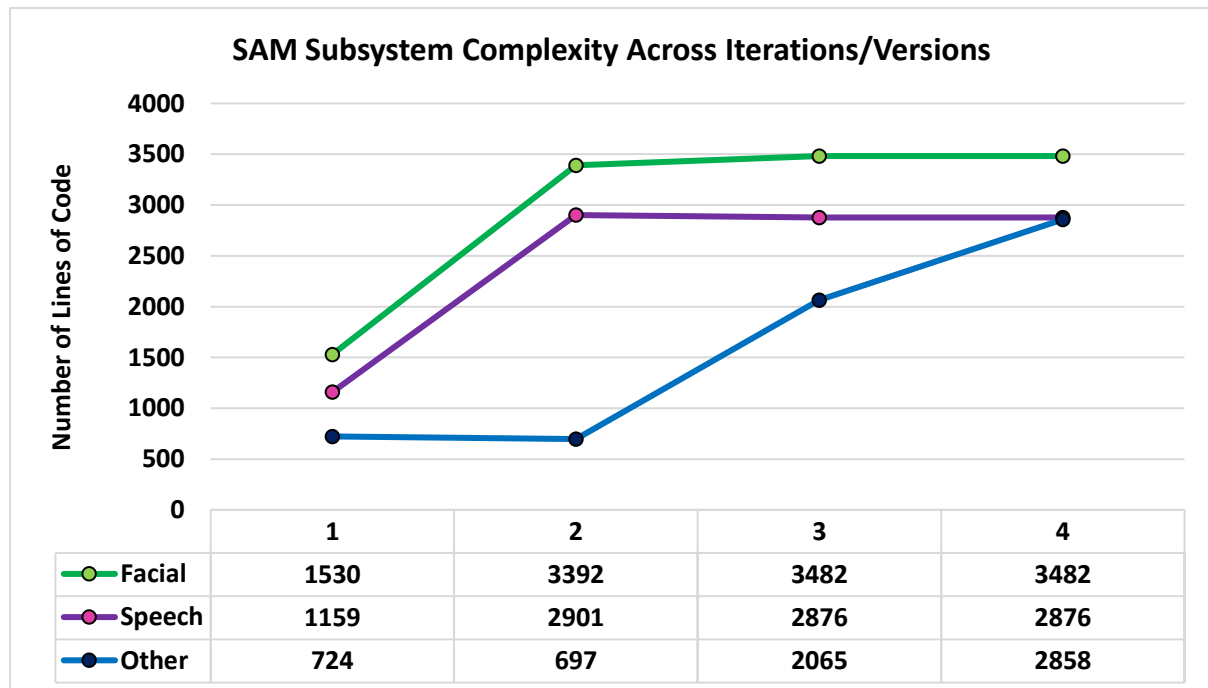


Figure 17: SAM subsystem complexity over time as functions of the number of lines of code for each version of the system, showing functionality was added across different versions

Analysing Figures 16 and 17 we can see how SAM evolves over time. Looking at the size and complexity of the system at version 1, we can identify the core functionality as requiring a larger volume of code to complete facial expression and paralinguistic speech analysis in comparison to “other” functions. Considering the addition of report generation functions in version 2 being the only major addition, we can see how much is required to achieve this, by looking at the steep gradient from version 1-2 in Figure 17. Analysing the same relationship in Figure 16, we see that the size of SAM in MB drastically increases from version 1 to later versions. This is due to the introduction of the CNN facial expression classification models that were added to improve predictive accuracy.

Through Figure 17, we can deduce that at the point of version 2, facial expression and paralinguistic speech classifiers remained largely unchanged, with the largest changes coming in “other” functionalities which include functions relating to SAM’s working memory, user and subject database integration, mask detection, and the “interaction” subsystem, increasing to a point in which its complexity relative to the number of lines of code is nearly equal to that of the speech analysis subsystem.

Through the iterative SDDP discussed in this section, SAM was successfully developed to the specifications defined in version 0, while maintaining enough flexibility for elements to be changed, and addition functions to be added throughout the development process. This chapter has shown various examples of SDDP and how AXAI can be integrated into them, with SAM being an example of this. Chapter 5 will discuss SAM in a deeper, more holistic level, outlining back-end and front-end design elements as well as further discussing its functionality as an embedded system.

Chapter 5: Introducing SAM, the State Assessment Module

The State Assessment Module is an example of how AXAI has been implemented into a sophisticated, real-time ASA system. The system is modular in its design allowing for it to be integrated into existing robotic and surveillance infrastructures, granting the host machine ASA capabilities if they are equipped with the necessary peripheral devices. This chapter will aim to discuss the full extents of SAM's functionality, including front-end and back-end design approaches (including hierarchical classifier design) and the flow of information through the GUI. To summarise some of SAM's key functions:

1. Novel, real-time, embedded system capable of being equipped into existing infrastructures,
2. Novel classification architecture that is based on literary and statistical findings,
3. Allows for monitoring of input data, intermediate data stages and output data,
4. Visual representations and explanations of changes in states across four levels of classification and two input modalities (facial expressions and speech),
5. Provides explanations for classification both textually and vocally for all fields of classification i.e., an understanding of purpose and the way I was designed,
6. Textual and vocal dialogue between the user and system,
7. The ability to generate reports from live data and historical data for all classification levels
8. An understanding of local (to the interaction) and global date-time information and the ability to explain this information,
9. An understanding of who is using the system and who SAM is "looking at" (the subject in the interaction),
10. Ability to access user, subject, and dialogue database information in real-time.

Chapter 5 will serve as a holistic description of SAM covering all elements of its design including the construction of facial expression and paralinguistic speech classifier trees as well as reporting classifier performances. Regarding front-end design, this chapter includes multi-level flowcharts and discussions on SAM's GUI and how information is transferred across various subsystems.

5.1 Classifier Design and Implementation

SAM is a system that exploits hybrid learning methodologies to allow for continuous affective state assessment, thus accounting for the dynamics of human expressions. In its final version, the data used to train the facial expression and speech classifiers was curated from the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) dataset [124]. Throughout the development cycle, several datasets were experimented with for facial expression and speech classification, with these datasets being reported in Table 11.

While the performances of most datasets experimented with was relatively consistent when performing classifier validation tasks, each dataset had its weaknesses and several considerations needed to be made. Firstly, a decision was made that the dataset had to be multimodal such that the ground truth for affective state expressions was consistent across both facial expression and speech modalities hence, single-channel datasets were omitted. Secondly, the dataset had to have continuous data such that a dynamic classification solution could be developed. Finally, the dataset had to be diverse such that it would be robust enough to handle speakers of various different genders and nationalities.

Dataset Name	Modality	Classes (no. entries)	Publication Year	Description
Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [124]	Facial Expression and Speech	Neutral (288) Calm (576) Happy (576) Sad (576) Angry (576) Fearful (576) Disgust (576) Surprise (552) Total = 4296 samples	2018	Audio-visual dataset containing utterances for speech and song samples. The number of utterances in the third column only refers to the number of utterances for speech. The dataset is diverse and consists of 24 actors both male and female speaking North American English [124].
Cohn-Kanade Plus (CK+) Dataset [125]	Facial Expression	Neutral (266) Contempt (18) Happy (69) Sad (28) Anger (45) Fear (25) Disgust (59) Surprise (83) Total = 593 samples	2010	The CK+ Dataset uses FACS action units as the basis for defining the emotional images in the dataset. The labelling of data is done through the validation of present action units in the image. The resolution of the images is 640 x 490 and the is diverse in the selection of subjects used when constructing the dataset [125].
Surrey Audio-Visual Expressed Emotion (SAVEE) Database [126]	Facial Expression and Speech	Anger (60) Happiness (60) Sadness (60) Surprise (60) Fear (60) Disgust (60) Neutral (120) Total = 480 samples	2011	Database consists of four male actors speaking with 7 different states. The speakers are native British English speakers. The database uses the TIMIT phonetically balanced sentences as the basis for the utterances [126].
Toronto Emotional Speech Set (TESS) Database [127]	Speech	Anger (200) Happiness (200) Sadness (200) Surprise (200) Fear (200) Disgust (200) Neutral (200) Total = 2800 samples	2010	The TESS consists of utterances by two female English speakers (26 and 64 years old). The phrases are of the structure “say the word ___” in which the final word represented by the blank space is one of 200 target words. [127]
Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) [128]	Facial Expression and Speech	Neutral (551) Happy (618) Sad (661) Anger (651) Fear (644) Disgust (654) Total = 3779 samples	2015	The CREMA-D dataset is an audio-visual corpus of affective data, in total there are 7442 (audio-only + visual-only + audio-visual) entries from 91 different actors ranging between ages 20 to 74 [128].
Berlin Database of Emotional Speech (EMO-DB) [129]	Speech	Anger (127) Boredom (81) Disgust (46) Fear (69) Happiness (71) Neutral (79) Sadness (62) Total = 535 samples	1999	Database consists of male and female German native speaking subjects acting out 10 different utterances [129].

Table 11: List of facial expression, speech and multimodal datasets experimented with throughout the development of SAM.

With the above considerations in mind, the RAVDESS dataset [124] was chosen for the final implementation of SAM. RAVDESS is the most diverse in comparison and is the most novel, being published in 2018. Novelty is important as the video and audio-recording quality of the RAVDESS samples is far more consistent with modern, higher-definition peripheral devices. However, the two biggest advantages of using the RAVDESS dataset was that (i) it allows for the classification of *all*

temporal phases i.e., onset, apex, offset [130] and (ii) it accounts for high and low levels of activation of states. Whereas a dataset like the CK+ for example, while being very popular, only shows a change in state from neutral to activated (onset to apex) thus, it could potentially introduce some ambiguity in terms of if a person is deactivating from a particular state or activating towards it if it were deployed in a classifier.

Thus, after separation and clustering of both modalities, there should be 37 classifiable states present at the lowest level (Level 4) that define the temporal phases of each state present in the RAVDESS dataset (minus Calm), and the level of activation (high vs. low) as indicated in Table 12.

State	Levels of Activation	Temporal Phase
Neutral	-	-
Surprise	High, Low	Onset, Apex, Offset
Happy	High, Low	Onset, Apex, Offset
Anger	High, Low	Onset, Apex, Offset
Fear	High, Low	Onset, Apex, Offset
Disgust	High, Low	Onset, Apex, Offset
Sadness	High, Low	Onset, Apex, Offset

Table 12: List of SAM’s classifiable states present at the lowest level of classification (level 4) for both facial expression and paralinguistic speech classification.

Note that neutral expressions of states are not separated into temporal phases or levels of activation as neutral represents a baseline, absence of activation. Immediately, we can identify that presenting raw outputs of all the classifiable states using a single classification model is not conducive to explainability and comprehensibility as this would correspond to an overabundance of information [3]. This highlights a challenge that would need to be addressed through the GUI and through a hierarchical classifier ensemble for both modalities. The following section addresses the various levels of classification and how the hierarchical nature of SAM was constructed and how it exploits the relationships between features presented in Tables 2 and 3 to improve the AXAI capabilities of SAM.

5.1.1 Facial Expression Classifier Subsystem

Table 2 introduced the facial action coding system (FACS), which serves as a means of separating common facial expressions based on different combinations of activated facial muscles that have identifiable action units (third column of Table 2). FACS-posed expressions are often used for the development of datasets that contain acted expressions as was the case with the RAVDESS dataset [124]. This allows for the separation of data and the detection of micro-expressions as the activation of different muscles and muscle groups changes as a person activates from a neutral state to an activated state and back to neutral state. This was accounted for in the development of SAM’s facial expression classifier subsystem in order to extract temporal phase information from continuous video samples.

The RAVDESS dataset samples are labelled in two ways that are beneficial for the construction of a continuous ASA system. Firstly, they show the “emotional intensity” of a speaker as ‘normal’ or ‘strong’ which is consistent with models like Russell and Mehrabian’s Three-Factor Theory of Emotions and the Plutchik spectrum [82, 84] that account for changes in expression intensity. Secondly, the RAVDESS dataset labels the emotion being expressed by the actors as well as the intensity. Given that video samples are continuous collections of still images, through additional work, the RAVDESS

dataset allows for the extraction of temporal phase, affective state, and emotional intensity information, allowing for the development of a continuous ASA solution.

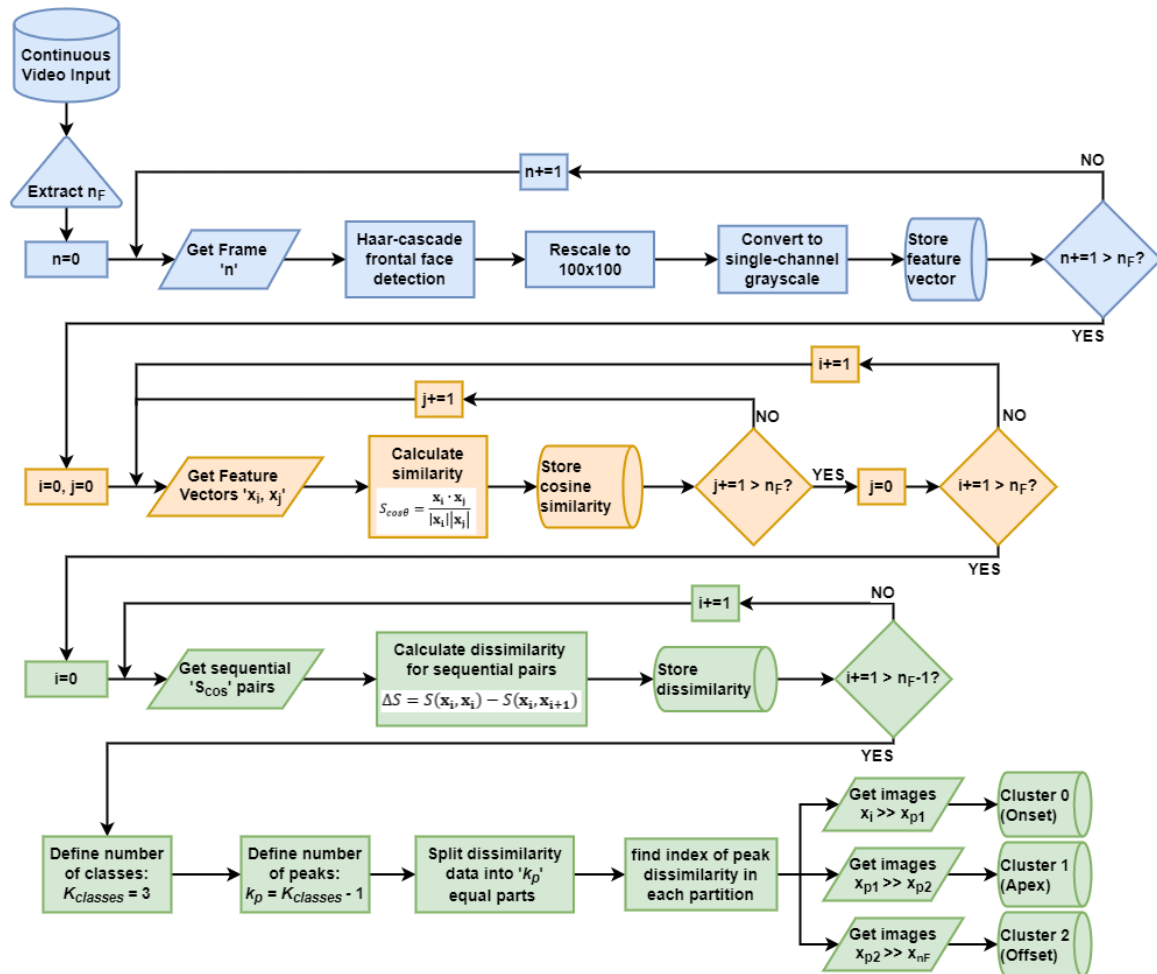


Figure 18: cosine-similarity based clustering of continuous RAVDESS samples showing pre-processing (blue), similarity calculation (orange), dissimilarity and clustering (green).

The initial unsupervised, cosine similarity-based clustering process separates a continuous sample into three clusters of images that define the temporal phases – onset, apex, and offset based on a detection of dissimilarity throughout the sample. The flowchart visualised in Figure 18 highlights the separation process for a *single* continuous sample. Thus, this is applied for all continuous samples in the dataset to retrieve all class data prior to training the classifier models.

As shown, the flowchart is separated into three stages: (i) pre-processing, (ii) similarity calculation and (iii) dissimilarity calculation and clustering. The pre-processing stage firstly defines number of frames in the continuous sample and then utilises OpenCV and the Python Image Library (PIL) to extract the facial image and rescale it to a 100x100 image size prior to conversion from an RGB image to a single-channel, grayscale feature vector.

The similarity calculation stage follows the mathematical foundations outlined in Section 2.1.2, comparing features vectors, and calculating the cosine similarity across all vectors in a continuous sample using equation (15). This information can then be presented as a heat map (visualised in Figure 19) to show the cosine similarity across a continuous sample. Figure 19 shows a comparison between

the similarity calculation through the mutual information score and using cosine similarity for the same sample. In both cases we see that the diagonal represents the highest value i.e., the highest similarity. This is because the vectors being compared are the same and as such, we can use (15) as an example to show that $\cos^{-1}(1) = 0^\circ \rightarrow$ maximum similarity.

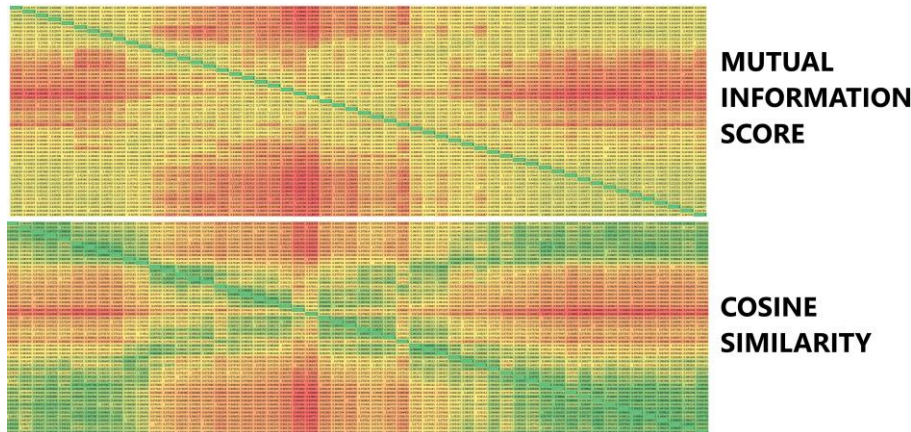


Figure 19: Comparison of similarity across a continuous RAVDESS dataset sample. Actor information: Actor 2, happiness, low intensity, phrase 1, first repetition.



Figure 20: Visualisation of three “temporal phase” regions that are contained in the one sample, using the cosine similarity sub-figure from Figure 19.

Figure 20 shows the foundation of the clustering process and how mutual information, and statistical dependence is exploited to extract temporal phase information in this work by defining the three clusters (outlined by the blue rectangles). In theory, this process can be applied to define even more micro-clusters and lower-level states that could be exploited in the future. This is visualised in Figure 21, which enhances the middle region in Figure 20, displaying four micro-clusters within this larger cluster. While this is something that could be explored in the future for more defined micro-expression analysis, in this work, only the higher-clusters (Figure 20) are considered i.e., $K_{classes} = 3$.

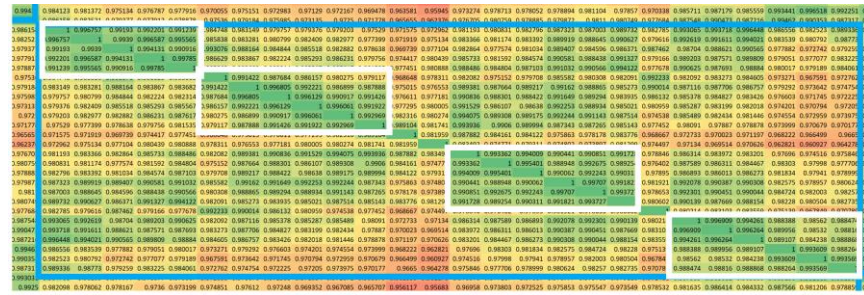


Figure 21: An expansion of the middle region in Figure 20, showing even more micro-level states contained within the higher-level region.

The purpose of the *dissimilarity* calculation and clustering stage is to find the locations of the cluster boundaries. This is done autonomously but can be represented graphically as shown in Figure 22, which shows the “frame-to-frame” gradient analysis for RAVDESS and CK+ samples using cosine similarity extraction. Under the assumption that $K_{classes} = 3$, the number of peaks that this process looks for can be defined as: $K_p = 2$. Analysing the RAVDESS figure, we see that these peaks occur at frame transitions 16 and 38, which in this example, conveniently separates the continuous sample into three equal clusters, however this convenient observation will not occur for every sample. Comparing to the CK+ graph on the bottom of Figure 22, we see that the dissimilarity peaks are less pronounced when analysing the figure, occurs at the 14th frame-to-frame transition.

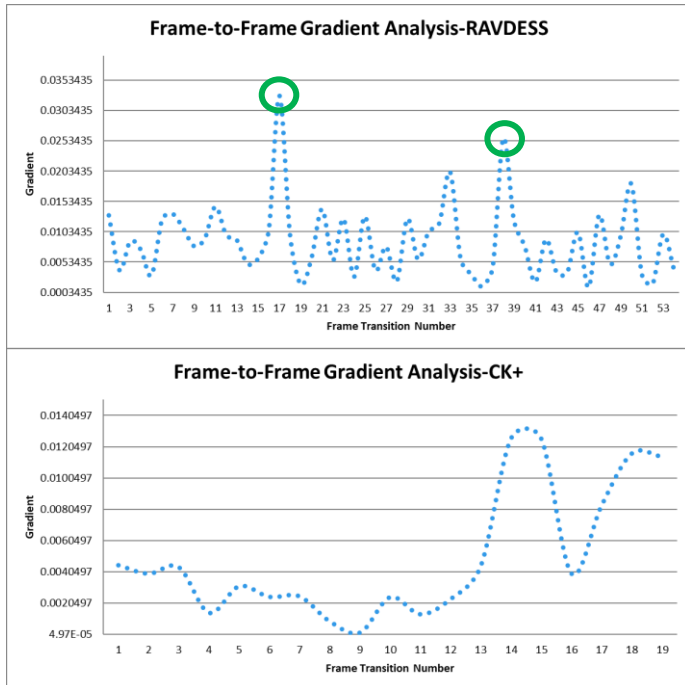


Figure 22: Visualisation of the dissimilarity measurements for frame-to-frame transitions from samples in the RAVDESS (top) and CK+ (bottom) datasets.

Repeating the above process for all samples allows for the clustering of similar facial expression and samples for all classes outlined in Table 12. This unsupervised separation algorithm allows for the clustering of data and could be replicated for any mutual information-carrying continuous data as will be shown in later, as this method is deployed for clustering affective speech signals as well.

As mentioned prior, SAM employs a four-tier, hierarchical classifier ensemble for facial expression and speech classification. Each level of the hierarchy contains a unique collection of classes that explain SAM's decision-making processes and how it is able to identify micro-expressions of states and changes in temporal phase. For the facial expression classification subsystem, the four levels of classification can be described as such:

1. Level of activation

First classification stage is a ternary classifier to identify the intensity of the expression as either: Neutral, Low or High, exploiting the emotional intensity-based labelling of the RAVDESS dataset.

2. State group and muscle regions

After determining the level of activation/emotional intensity, SAM attempts to identify the state group that the input expression may belong to as well as the corresponding muscle regions. This exploits similarities in FACS action units and allows for a more 'high-level' assessment of states. This classification level consists of two ternary classifiers – one for 'low' activation and 'high' activation.

3. Discrete state and extended action unit

With the state group determined, SAM will look at more micro-level changes in features to assess the discrete state of the person and the corresponding action units that are associated with that expression of an affective state. Whereas level 2 was a general prediction of state groups, level 3 exploits this information to improve classification accuracy of discrete states, transforming and improving a single septenary classifier as discussed in previous works [1, 19] into a more interpretable combination of ternary and binary classifiers. This classification level deploys three binary classifiers for each level of activation.

4. Temporal phase

The final classification level, SAM will analyse whether a person is approaching a state (onset), experiencing that state (apex) or returning to a neutral level (offset). This level of classification allows SAM to be applied in a temporal domain, giving it an understanding of the continuous nature of expressions given that humans cannot instantaneously change their expression from neutral to activated due to physical constraints and muscle movements needing to occur. For a real-time ASA system, it is essential that temporal phase is classifiable. This classification level deploys six ternary classifiers for each level of activation.

The hierarchical facial expression classifier ensemble is presented in Figure 23. The novel addition of SAM is the ability to show the reasoning behind its classification processes from input → output. This is achieved through a GUI display of muscle groups and facial action units to the users at levels 2 and 3. The muscle groups shown to users at level 2 exploit the similarities in action units and muscle activations in states reported in Table 2. A graphical representation of state and muscle groups (and an insight into its functionality in the GUI) is shown in Figure 24.

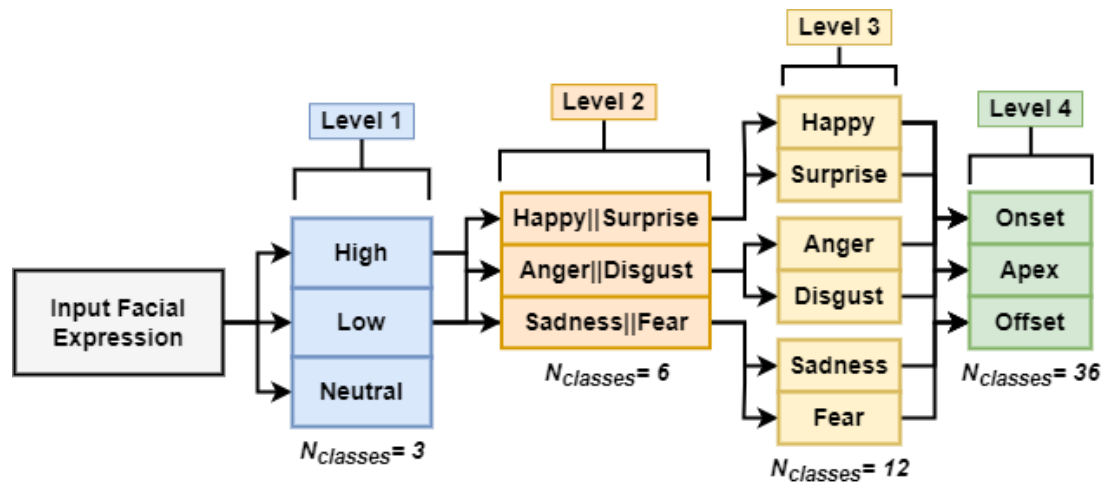


Figure 23: Visualisation of the four-tier hierarchical classification ensemble for the facial expression classification subsystem of SAM.

State Group	Muscle Group	Justification
		Open Mouth - AU 26, 27 Upper and Lower eyelid movement - AU 5, 6, 7
		Raised lower eyelids - AU 6, 7 Eyebrow frown - AU 4 Lip movement - AU 9, 10, 23
		Raised eyebrows - AU 1 Eyebrow frown - AU 4 Mouth region - AU 15, 26, 27

Figure 24: Graphical representation of state groups and muscle groups for level 2 classification, using the FACS to evidence the logic behind the choice of states per group.

The structure of the hierarchical classifier ensemble introduced above is visualised in Figures 23 and 25. Across the four levels of classification, SAM is capable of identifying a total of 57 unique, classifiable facial expressions. Through the GUI, the user can choose how much of this information they would like to see depending on what windows they open, which serves as a boon for the comprehensibility of the system.

Prior to training the classifiers, LDA was applied to the labelled facial expression data at each level to assess and visualise how the facial expression samples would cluster in a linear space. This was done as a sanity check prior to training and validating the classifier models. A visual representation of this is shown in Figure 25.

Analysing the temporal phase (level 4) clusters in Figure 25, we see that while SAM only accounts for three temporal phases i.e.: onset, apex, and offset, there is potential for an expansion in the number of temporal phases per state from 3 \rightarrow 5 as shown in Figure 26, which is an enhanced version of the Fear subplot from Figure 25. Looking at Figure 26, we see three larger clusters (circled blue) and two smaller clusters (circled orange). This relates directly to a hypothesis made earlier when discussing Figure 21, where sub-clusters were identified in the cosine similarity heatmap, allowing us to theorise that this work can be expanded upon in order to discover clusters that reside outside of the three major temporal phase states that could unlock the ability to classify even more micro-expressions. Furthermore, the linear relationship between these clusters from onset \rightarrow apex \rightarrow offset proves the continuous nature of facial expressions in the RAVDESS dataset and further validates the approach taken when designing SAM's facial expression subsystem.

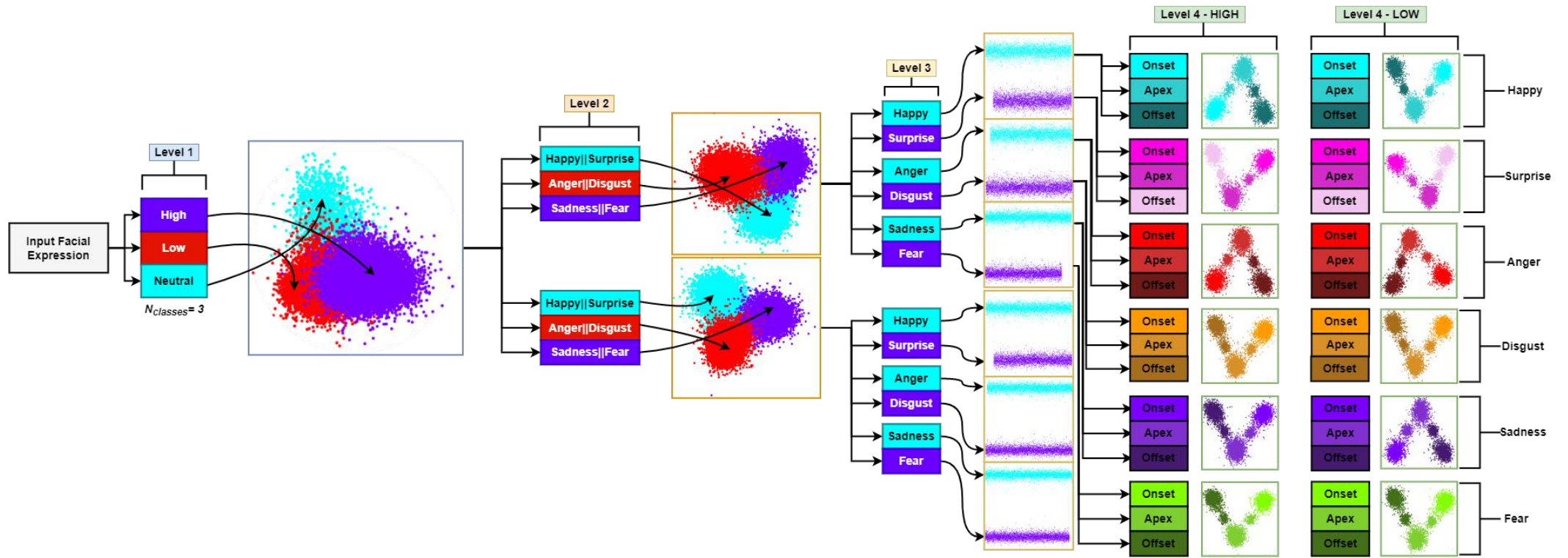


Figure 25: Visualisation of facial expression LDA-transformed clusters across the four levels of the hierarchical classifier ensemble. The dimensionality (number of linear discriminants) of the clusters was dependent on the number of classes in the problem where $n_{dimensions} = n_{classes} - 1$.

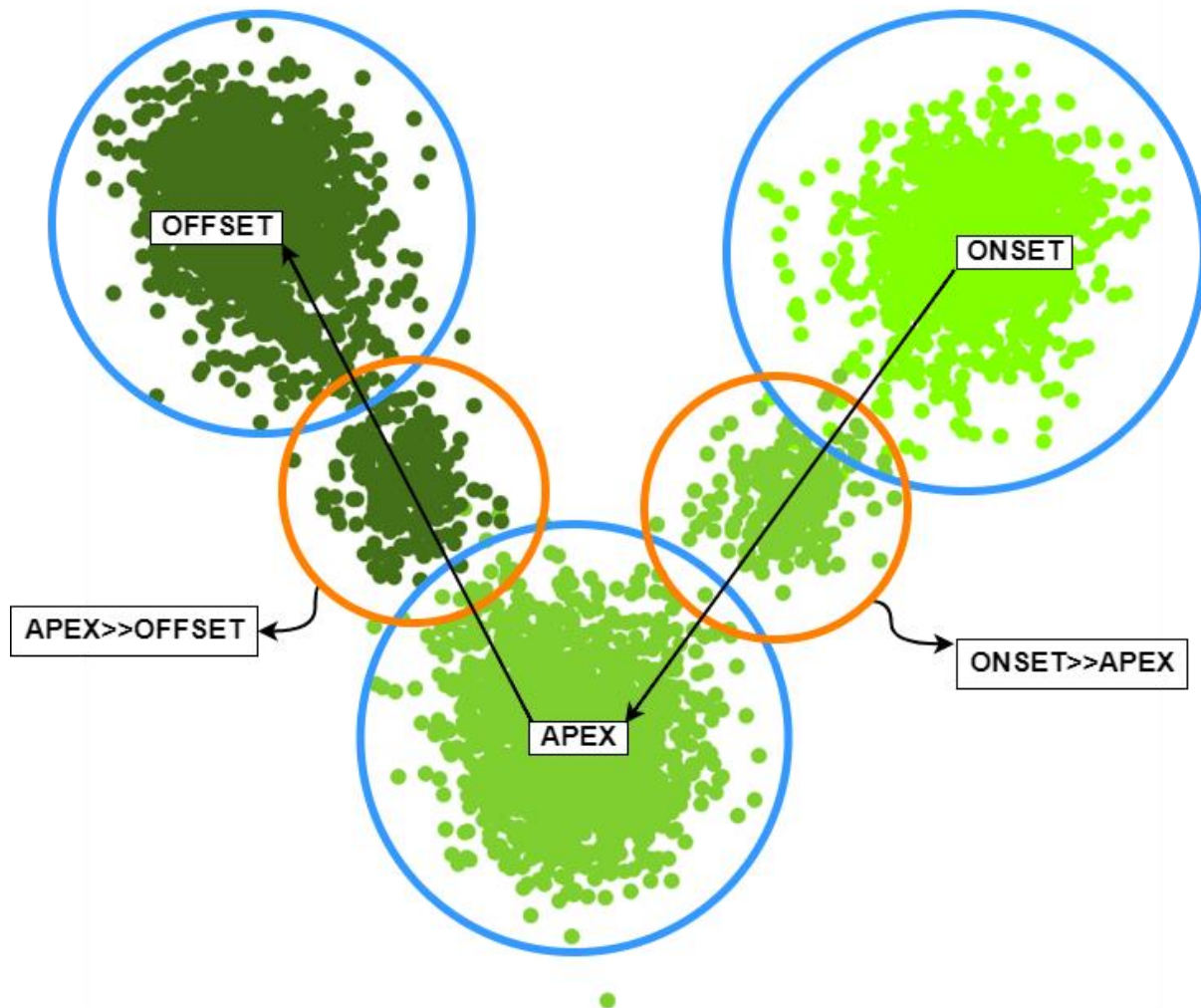


Figure 26: Enhancement of the 'Fear' cluster subplot from Figure 25 showing the traditional temporal phases (onset, apex and offset) as well as the clear presence of smaller clusters highlighted in orange circles.

With the clusters sanity checked and supported by theoretical foundations, the next stage is to train and validate the classifiers. In total, there are twenty-one (21) facial expression classifiers tasked with outputting the 57 classifiable states. Given the cost-effective, real-time application proposed through SAM, the classifiers needed to be optimised in regard to size, complexity and predictive accuracy. Many traditional machine and deep models were explored for training the facial expression classifiers including SVM, MLP, and various CNN models. In its final version, for all 21 classifiers, SAM deploys 2-layer CNN's that use 50x50 facial greyscale facial image inputs. The model architecture for all four levels is visualised in Figures 27 and 28. While more complex solutions were explored throughout Versions 1+2 of the development process, the 2-layer CNN became an optimisation problem and achieved competitive classifier performances while being less complex and faster – an important consideration for real-time systems that employ classifier ensembles.

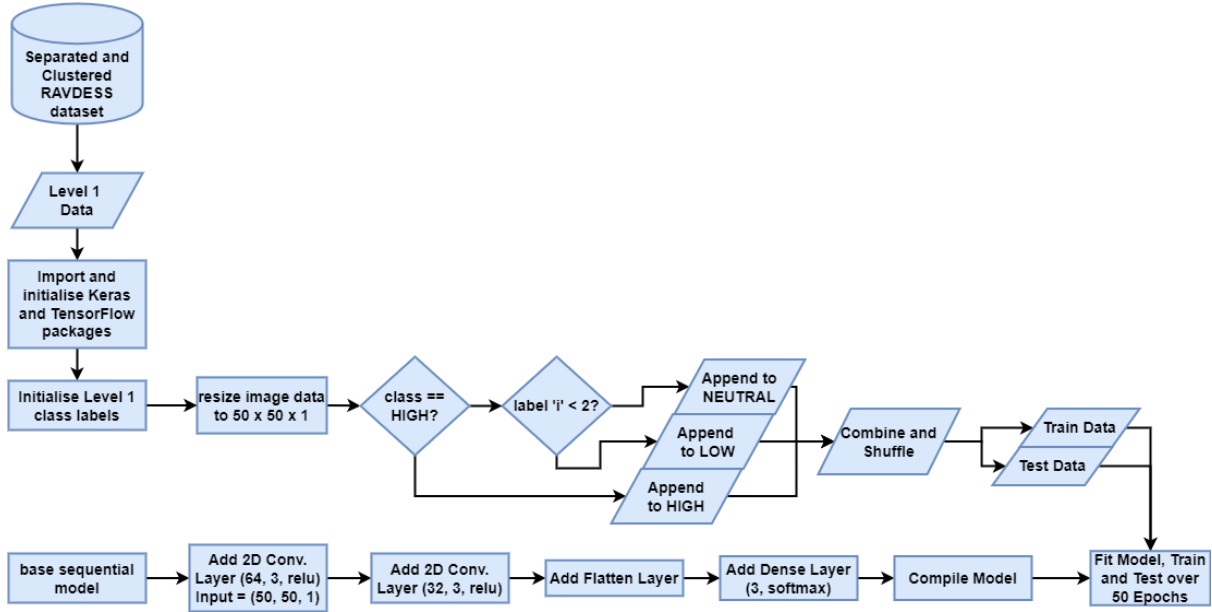


Figure 27: Facial expression Level 1 Classifier training architecture.

This section has explored the back-end facial expression classifier functions and the construction of the hierarchical classifier ensemble. Through the proposed architecture, SAM is capable of classifying a large amount of unique facial expressions and affective states that are representative of continuous emotion and affective state models.

The topic of classification in machine and deep learning systems is not complete without a discussion on performances. There are various metrics to use for reporting a classifiers accuracy and performance. Among these are metrics such as: confusion matrices, validation accuracies/precision, recall, F_1 score and ten-fold cross-validation accuracies. Due to the large number of classifiers used in this system across speech and facial expression modalities, it was decided that validation accuracies and cross-validation accuracies be used to assess the models.

Validation accuracies were extracted for the facial expression CNN models discussed in this section. This represents the number of correct predictions made ' $N_{correct}$ ' out of a bag of total predictions ' N_{total} ', defined by:

$$Acc_{val}(\%) = \frac{N_{correct}}{N_{total}} \times 100\% \tag{33}$$

This metric is often used in ASA and general machine learning literature and allows for the assessment of the O_{fp} parameter discussed w.r.t to the AXAI framework. The CNN models were trained and validated using the RAVDESS dataset and the test-train ratio ' $r_{tst-trn}$ ' was 1:4 i.e., 20% for testing and 80% for training. From the continuous RAVDESS video samples, 76270 facial expressions were extracted, separated, and labelled for training and validating the CNN Models. Table 13 reports the distribution of samples per class. The construction of Table 13 allows us to assess SAM's training data size ' d_{trn} ' which shows that when looking at the temporal phase, the corresponding sample sizes show a consistent trend for all states. Establishing the ' d_{trn} ' and ' $r_{tst-trn}$ ' provides context in order to report the validation accuracies of the various classifiers, with these results shown in Table 14.

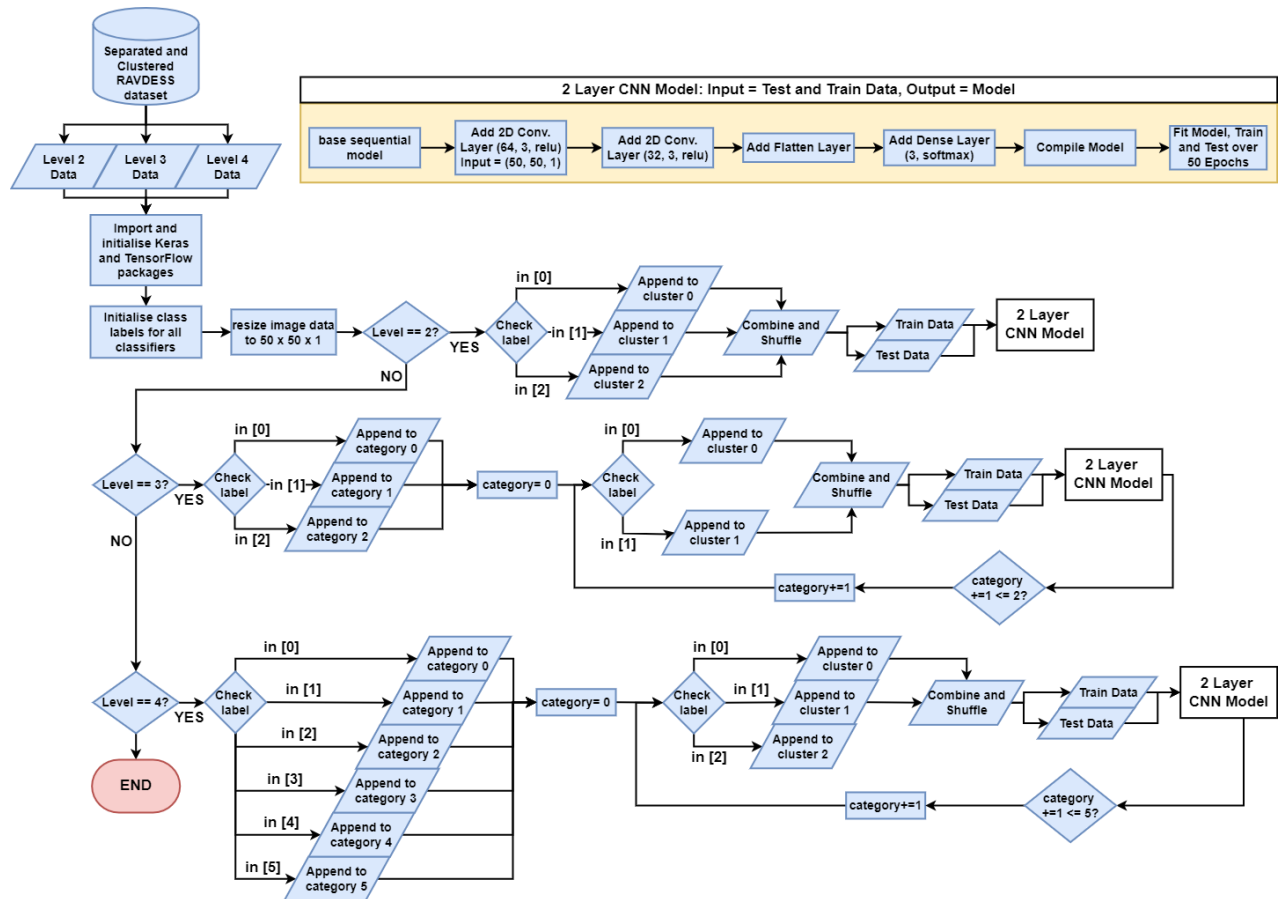


Figure 28: Levels 2, 3 and 4 facial expression classifier training architecture.

State	Activation	Temporal Phase and Corresponding Number of Samples					
Neutral	None	5907					
Happy	LOW	Onset	1710	Apex	2410	Offset	1868
Sadness	LOW	Onset	1753	Apex	2250	Offset	1866
Anger	LOW	Onset	1710	Apex	2161	Offset	1787
Fear	LOW	Onset	1702	Apex	2226	Offset	1762
Disgust	LOW	Onset	1674	Apex	2251	Offset	1837
Surprise	LOW	Onset	1544	Apex	2147	Offset	1748
Happy	HIGH	Onset	1858	Apex	2158	Offset	2141
Sadness	HIGH	Onset	1845	Apex	2775	Offset	1920
Anger	HIGH	Onset	1739	Apex	2300	Offset	1741
Fear	HIGH	Onset	1730	Apex	2207	Offset	1864
Disgust	HIGH	Onset	1766	Apex	2480	Offset	1830
Surprise	HIGH	Onset	1627	Apex	2232	Offset	1744

Table 13: Distribution of RAVDESS class data used for training and validation of the facial expression CNN classifiers, total = 76270 samples.

Index	Classifier ID	Activation	Class [0]	Class [1]	Class [2]	Acc _{val} (%)
0	1	-	Neutral	Low	High	89.58%
1	2.1	LOW	Happy Surprise	Anger Disgust	Sadness Fear	93.90%
2	2.1.1	LOW	Happy	Surprise		98.04%
3	2.1.1.1	LOW	Happy-Onset	Happy-Apex	Happy-Offset	72.75%
4	2.1.1.2	LOW	Surprise-Onset	Surprise-Apex	Surprise-Offset	74.78%
5	2.1.2	LOW	Anger	Disgust		99.89%
6	2.1.2.1	LOW	Anger-Onset	Anger-Apex	Anger-Offset	71.52%
7	2.1.2.2	LOW	Disgust-Onset	Disgust-Apex	Disgust-Offset	74.67%
8	2.1.3	LOW	Sadness	Fear		99.83%
9	2.1.3.1	LOW	Sadness-Onset	Sadness-Apex	Sadness-Offset	66.89%
10	2.1.3.2	LOW	Fear-Onset	Fear-Apex	Fear-Offset	66.83%
11	2.2	HIGH	Happy Surprise	Anger Disgust	Sadness Fear	97.01%
12	2.2.1	HIGH	Happy	Surprise		98.71%
13	2.2.1.1	HIGH	Happy-Onset	Happy-Apex	Happy-Offset	77.79%
14	2.2.1.2	HIGH	Surprise-Onset	Surprise-Apex	Surprise-Offset	75.30%
15	2.2.2	HIGH	Anger	Disgust		99.09%
16	2.2.2.1	HIGH	Anger-Onset	Anger-Apex	Anger-Offset	70.73%
17	2.2.2.2	HIGH	Disgust-Onset	Disgust-Apex	Disgust-Offset	70.38%
18	2.2.3	HIGH	Sadness	Fear		99.87%
19	2.2.3.1	HIGH	Sadness-Onset	Sadness-Apex	Sadness-Offset	71.80%
20	2.2.3.2	HIGH	Fear-Onset	Fear-Apex	Fear-Offset	75.05%

Table 14: List of 2-layer CNN model information and reported validation accuracies obtained when training SAM using the RAVDESS dataset and an 80/20 ratio for training and testing. The classifier ID naming structure identifies what depth the specific classifier is applied. The ‘class [n]’ columns identify what labels are output by the system when a prediction is made using this classifier. All models were trained using the Keras and TensorFlow Python packages. The validation accuracies are visualised in Figure 29.

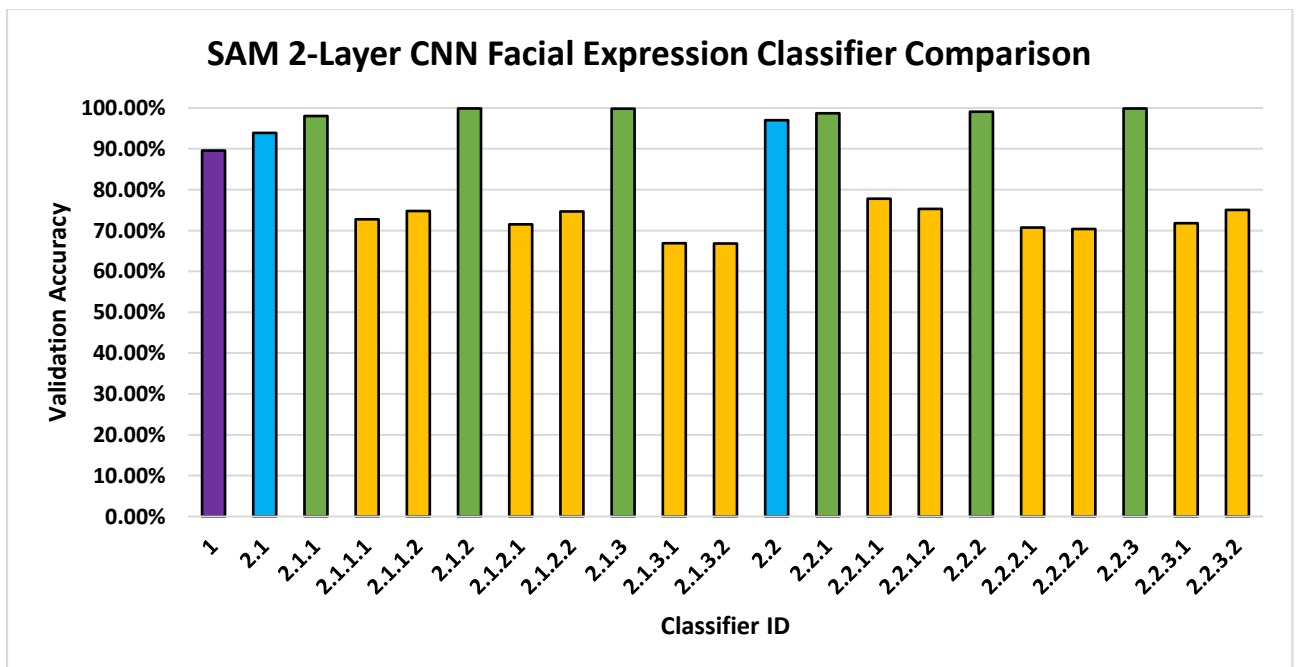


Figure 29: Visualisation of facial expression classifier performances using the 2-Layer CNN models discussed. Bars are colour-coded to show the level of classification where: PURPLE = level 1, CYAN = level 2, LIME = level 3 and YELLOW = level 4.

Table 14 and Figure 29 show the facial expression classification performance of each of the 21 classifiers embedded in the state assessment module’s classifier ensemble. The average classification accuracy can be broken down per level:

- Level 1. Single classifier = **89.58%** validation accuracy.
- Level 2. Two classifiers = **95.46%** mean validation accuracy.
- Level 3. Six classifiers = **99.24%** mean validation accuracy.
- Level 4. Twelve classifiers = **72.37%** mean validation accuracy.

Thus, the average classification accuracy across all classifiers is **83.07%** for facial expression classification. We see that the major decrease in performance comes at level 4 – temporal phase classification which shows the lowest average classification accuracy in comparison to other levels. Comparatively, we see that classification of discrete states (level 3) is most accurate and when compared to surveyed work reported in Figure 7 (See Section 2.5), we see that all levels of classification and the overall classification performance of SAM is competitive and outperforms other state of the art ASA systems.

This section has discussed the implementation of SAM’s novel hybrid learning architecture which uses separated RAVDESS data to classify a person’s level of activation, state group, discrete state and temporal phase using facial inputs. The successful implementation of this subsystem as reported in [24] lead to it being implemented for affective speech, proving the hybrid learning architecture is not bound by modality and in fact can be used with other continuous signals. This is detailed further in the following section.

5.1.2 Affective Speech Classifier Subsystem

The success of using the hybrid learning method for separating continuous facial expression data into 37 activation- and temporal phase-based classes resulted in the approach being applied for continuous affective speech data. Given the multimodal nature of human expressions of states in conversation, it is important that both speech and facial expressions are accounted for. As mentioned in Section 2.5, the measurement of affect in speech is separated into paralinguistic and linguistic components, with the former quantifying ‘how’ speech sounds and the latter describing the structure of linguistic components in an utterance (what was said). While SAM is capable of retrieving some linguistic information through speech-to-text functions, the linguistic affect data is not processed/classified. However, SAM is capable of classifying affective speech through the extraction of ComParE2016 low-level descriptor (LLD) features [88].

The ComParE2016 LLD feature set used in this work extracts 65 frame-wise acoustic parameters. While other feature sets like the GeMAPS [89] were explored throughout the prototyping stages, experiments found that the chosen feature set was the most effective for the clustering and classification systems proposed through this work. Likewise, other datasets like the SAVEE, TESS, CREMA-D and EMODB datasets reported in Table 11 were also explored but in order to be consistent with the facial expression modality, it was decided that the RAVDESS dataset would be used.

Like the facial expression classifier subsystem discussed in the previous section, cosine similarity-based clustering was adopted to separate continuous affective speech data into temporal phase-dependent classes. However, with speech being a different modality, it is important to note that there are significant differences in pre-processing and feature extraction as visualised in Figure 30.

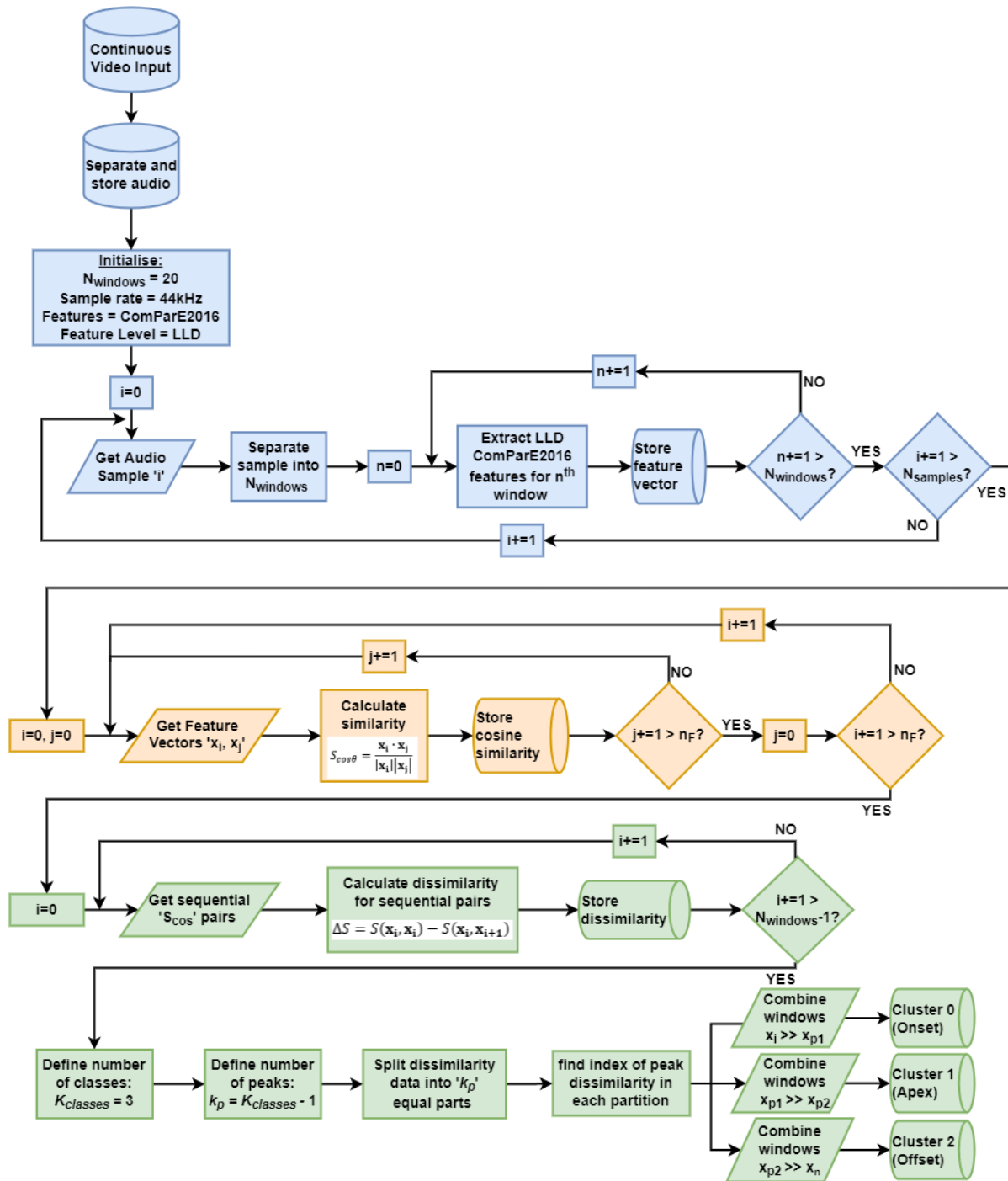


Figure 30: Cosine-similarity based clustering of continuous RAVDESS audio samples showing pre-processing (blue), similarity calculation (orange), dissimilarity and clustering (green). This figure is similar to Figure 18 with the difference being pre-processing

Analysing Figure 30, we see that it very similar to Figure 18, which showed the cosine similarity-based separation of continuous facial expression signals. This is intentional as it shows that regardless of the input modality, as long as the signal is continuous, the separation technique can be applied – in this case to extract temporal phase information from an affective speech signal. Note the initialised variables in the pre-processing stage. In this work the sample rate is 44kHz and the number of windows extracted is defined as $N_{windows} = 20$. The ComParE2016 LLD feature set extracts 65 frame-wise

features describing contours along a continuous signal. Two cosine similarity heat map examples for affective speech samples are visualised in Figures 31 and 32, using samples from the RAVDESS dataset. Interestingly, when analysing Figure 32 we see two max dissimilarity peaks in the first half of the audio signal. In cases like this, the algorithm picks the highest of the two points as the peak in that half. Furthermore, a 5% buffer is added to the start and end of a signal to ensure that the clusters do not form using 1 window of data. The choice of $N_{windows} = 20$ was due to the RAVDESS samples being approximately $2 \rightarrow 2.5$ seconds in length, thus resulting in each window being approximately $0.1 \rightarrow 0.15$ seconds in length.

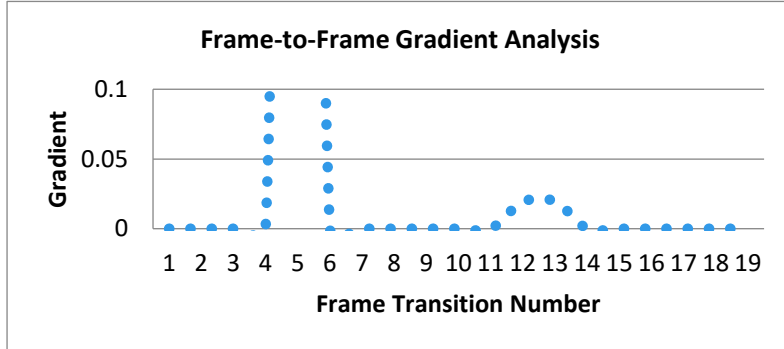
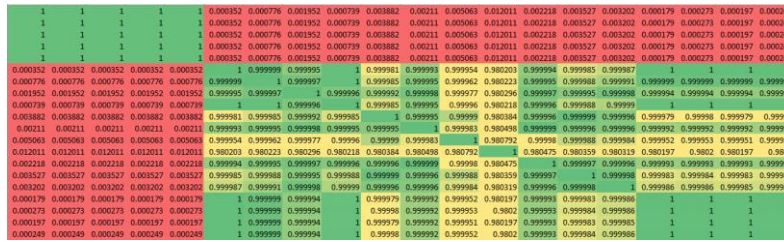


Figure 31: RAVDESS affective speech cosine similarity heat map for a continuous sample partitioned into 20 equally sized windows and the corresponding dissimilarity curve. Actor Information: Actor 4, disgust, high intensity, phrase 1, first repetition.

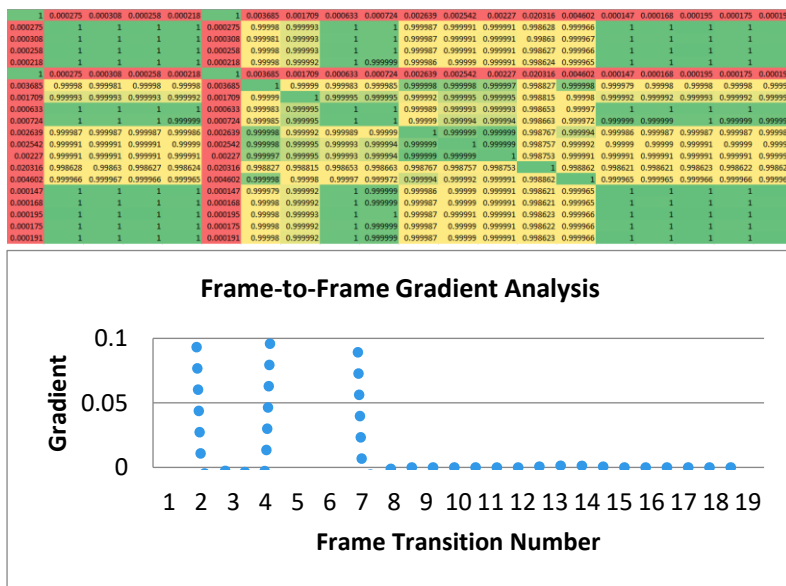


Figure 32: RAVDESS affective speech cosine similarity heat map for a continuous sample partitioned into 20 equally sized windows and the corresponding dissimilarity curve. Actor Information: Actor 9, surprise, high intensity, phrase 2, second repetition. Note two peaks are detected at frame transition 5 and 13 similar to Figure 31.

Using this method on all continuous samples in the RAVDESS dataset allowed for the separation and clustering of temporal phase-dependent affective speech samples. Through the methods discussed so far, we see that continuous, multimodal samples were initially separated into separate data channels (speech and facial expressions). They were then further separated into temporal phase clusters using cosine similarity-based separation, allowing for the delineation of all states outlined in Table 12.

Similar to facial expression ASA discussed prior, SAM’s affective speech classification is done on four levels. However, being that the signals are unique, the design and construction of the corresponding affective speech hierarchical classifier ensemble is different to the facial expression ensemble. The four affective speech classification levels can be described as follows:

1. Level of Activation

Like the facial expression subsystem, the first classification stage is a ternary classifier to identify the intensity of the expression as either: Neutral, Low or High, exploiting the emotional intensity-based labelling of the RAVDESS dataset.

2. Temporal Phase

After determining the activation, SAM attempts to identify the temporal phase of the speaker i.e.: approaching a state (onset), experiencing a state (apex), or deactivating from a state toward neutral (offset). This applies SAM in a temporal domain and while at this level not being descriptive of a state, shows the continuous nature of the incoming speech signal. This classification level consists of two ternary classifiers – on for ‘low’ activation and one for ‘high’ activation.

3. State/Valence Group and General Acoustic Characteristics

Once the temporal phase and level of activation is determined, SAM classifies the valence and state group that the speech sample belongs to as well as the corresponding acoustic characteristics that are similar between the states. This level of classification exploits the acoustic characteristic relationships identified in Table 3 across different states – allowing for a generalised classification of affective valence as described in [81, 82], allowing for an assessment of the ‘positivity’ or ‘negativity’ of an input speech sample. There are three binary classifiers that are deployed for each level of activation.

4. Discrete State and Extended Acoustic Characteristics

The final affective speech classification level. With the valence and general state group derived, SAM classifies the discrete state of the person and the acoustic characteristics associated with that state as per Table 3. This level of classification uses the general predictions in the previous level to improve the discrete, septenary state classification accuracy when compared to models derived in previous work [1, 19]. By combining ternary and binary classifiers for the assessment of affective speech, not only do you enhance the interpretability of the system, but it also allows for the improvement of classification performance. This classification level deploys six ternary classifiers for each level of activation.

Figure 33 visualises the construction of the hierarchical classifier ensemble described above, which we can see is similar in shape to the facial expression classifier ensemble and identical in regard to the number of classes ‘ $N_{classes}$ ’ at each level (as visualised in Figure 23). While the distribution of ternary and binary classifiers is consistent across the two modalities, the way that signals are processed, classified, and labelled is different in terms of characteristics and features at each level.

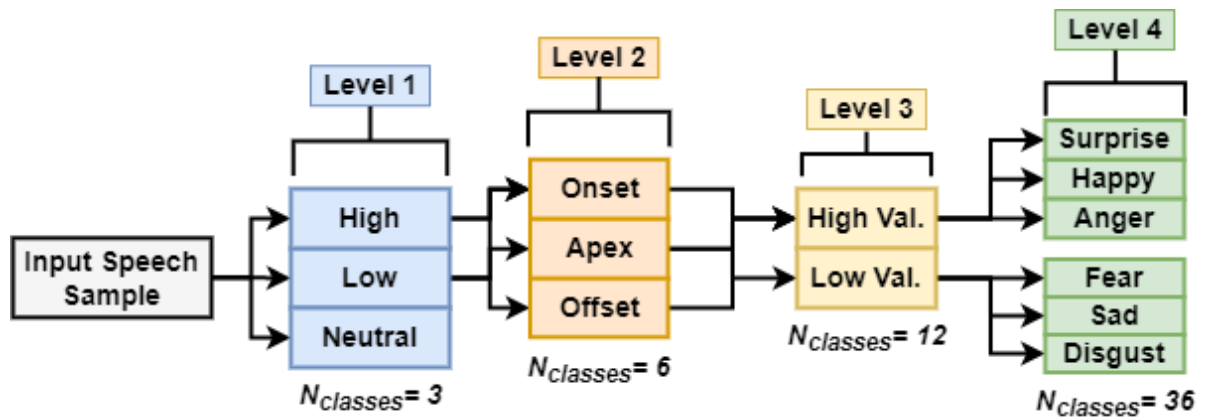


Figure 33: Visualisation of the four-tier hierarchical classification ensemble for the affective speech classification subsystem of SAM.

The affective speech hierarchical classifier ensemble visualised in Figure 33 is expanded upon in Figure 34, showing how states are clustered at all levels using LDA representations of data. Like the facial expression classifier ensemble, in total, SAM is capable of classifying 57 unique affective speech states, tallying up a total of 114 states across the two modalities.

Unlike the facial expression classifiers which deploy CNNs for feature extraction and classification, the affective speech classifier ensemble uses LDA to transform input signals, projecting them onto a linear discriminant space prior to having a standard scaler applied to standardise the transformed data. New input signals are transformed using an LDA model prior to being input into SVM models for classifying and labelling input speech samples.

Analysing Figure 34 we see that across the four levels of classification, affective speech data is clustered very differently when compared to facial expression data. The first contributing factor is the size of the dataset used for training the LDA models ' d_{trn} ' which totals $96 \times n_{classes} = 3744$ samples after cosine similarity-based clustering. This volume is smaller than the facial expression absolute training data size by a factor of 20.37. Secondly, due to the cluster boundaries of individual speech samples being dependent on the number of windows prior to cosine similarity-based clustering, it is less likely that the temporal phase micro-expressions would be detected with a smaller number of windows. However, the number of windows $N_{windows} \frac{1}{\alpha}$ paralinguistic information. Thus, by increasing $N_{windows}$ to a large amount, the amount of information in an utterance segment would be reduced as a result, which may be detrimental to the clustering algorithm.

With the clusters and the hierarchical classifier ensemble established, we can discuss the training of machine learning models used for classifying affective speech at each level. For affective speech assessment, there are multiple steps to achieve classification with three models used per 'classifier', these are: (i) Linear Discriminant Analysis (LDA) model, (ii) Standard Scaler model and (iii) support vector machine (SVM) classifier model. A summary of the classification process is outlined in Figure 35, highlighting the goals, methods, and parameters at each stage. Figures 36 and 37 break down the model training architecture for *all* classifiers deployed across the hierarchical classifier ensemble. Python's sci-kit learn packages were used for training the affective speech models [131].

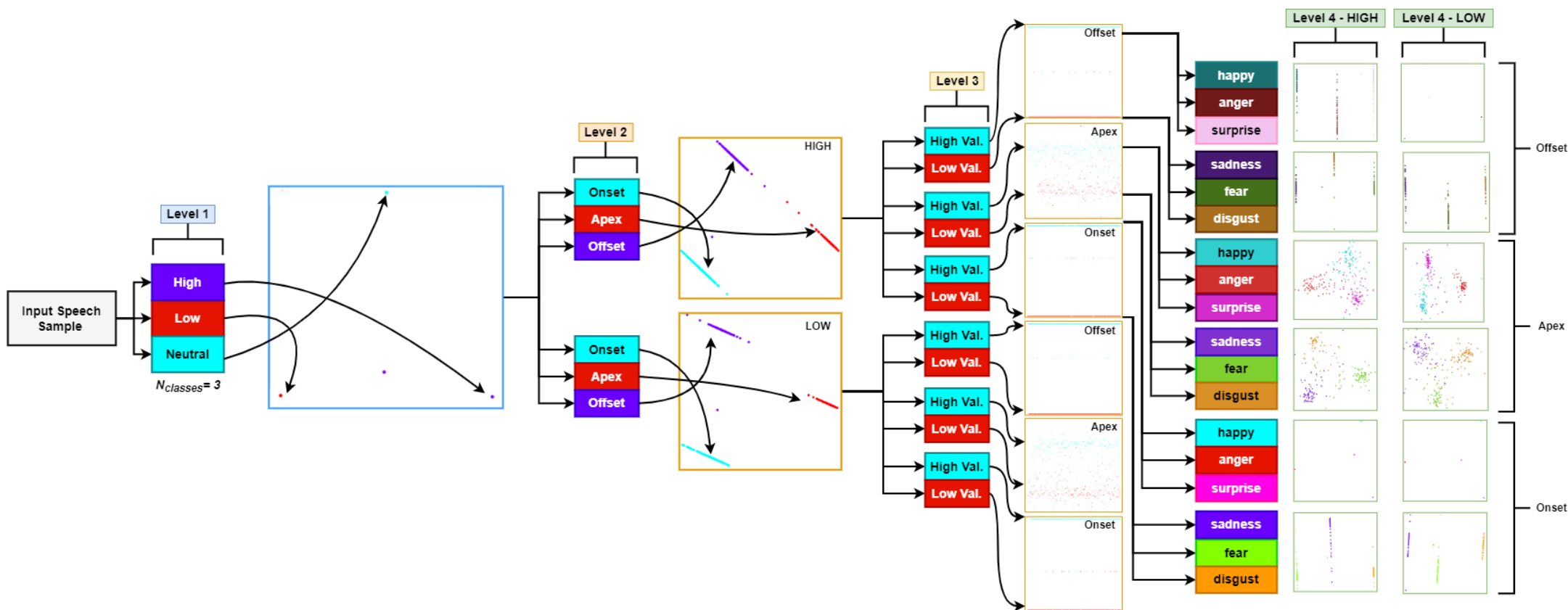


Figure 34: Visualisation of affective speech LDA-transformed clusters across the four levels of the hierarchical classifier ensemble. The number of linear discriminants for the clusters was dependent on the number of classes that were defined prior to training, where $n_{dimensions} = n_{classes} - 1$. Note that in comparison to the facial hierarchical classifier ensemble (Figure 25), there is less affective speech data, and the clusters are far more separated, in some cases having clearly defined cluster centroids with minimal variances.

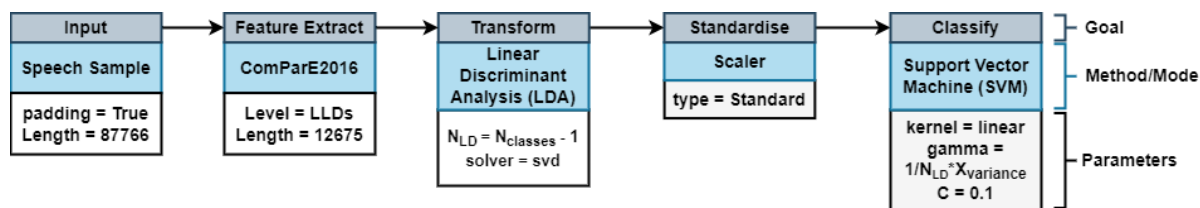


Figure 35 High-level summary of the affective speech classifier subsystem showing the process from input to classification and the three models deployed – LDA, Scaler, SVM.

Analysing Figure 35 we see that at each stage, there are parameters that need to be considered. Regarding the ‘Input’ block, note that padding is applied to ensure that all samples used for training the dataset have a consistent number of features = 87766, which is calculated from the longest sample in the dataset such that no information is lost during training. Zero-padding is performed, which pads the feature array with zero values. It was found that the padding was more pronounced with ‘onset’ and ‘offset’ samples, with the observed consistency assisting in model training and improving the overall classifier performance. Once padding is completed, the features are then extracted using the ComParE2016 LLD feature set that has been discussed previously. The resulting, flattened feature vector that is input into the LDA model contains 12675 features in total.

The statistical and machine learning foundations of LDA have been discussed previously in Section 2.1.3. To optimise separation the LDA models deployed in SAM have the maximum number of linear discriminants allowed based on the number of classes i.e.:

$$N_{LD} = N_{classes} - 1$$

The number of classes $N_{classes}$ and linear discriminants N_{Dim} at each stage is visualised in Figure 34 noting that for ternary classifiers, feature vectors are projected onto a two-dimensional hyperplane and for binary classifiers, they are projected onto a one-dimensional hyperplane. The solver used to solve the LDA optimisation problem defined by (19) was singular value decomposition (SVD). While an eigen value decomposition could be deployed (and was experimented with), it resulted in large LDA models that resulted in memory errors when attempting to incorporate them into SAM’s architecture. As a result, SVD was deployed, as it also more effective at handling data with a large number of features e.g., 12675. While LDA can be used for classification tasks, in this work, SAM exploits LDA to project data onto a linear discriminant space prior to classification.

Through trial and error, it was found that a scaler could improve classification accuracy when combined with LDA. Two scalers were experimented with: (i) robust scaler and (ii) standard scaler. The latter proved to be more effective as the robust scaler is useful when scaling sets of data that contain outliers. In this work, the outliers were handled prior to feature extraction using an Isolation Forest anomaly detection algorithm [132].

The standard scaler standardises an input sample using the mean ‘ μ_x ’ and standard deviation ‘ σ_x ’ of the training data. The standardised score ‘ Z_x ’ of a single sample (or feature vector) is described as:

$$Z_x = \frac{x - \mu_x}{\sigma_x} \tag{34}$$

where x describes the feature vector being scaled, which in this case is the LDA-projected data.

Finally, the SVM is trained using the scaled, LDA-projected training data. Given that LDA projects data onto a *linear* discriminant space, it is logical that the SVM kernels are consistent i.e., linear which is described in Section 2.1.3 through equation (22) i.e.:

$$\psi_{linear}(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}_i^T \mathbf{x}$$

The gamma ' γ ' and cost-function ' C ' parameters influence the construction of the SVM hyperplane boundaries, affecting the reachable areas and decision boundaries of the linear kernel. For all SVM models, the gamma and cost-function parameters remained consistent:

$$\gamma = \frac{1}{N_{Dim} \times \mathbf{X}_{variance}} \quad (35)$$

$$C = 0.1$$

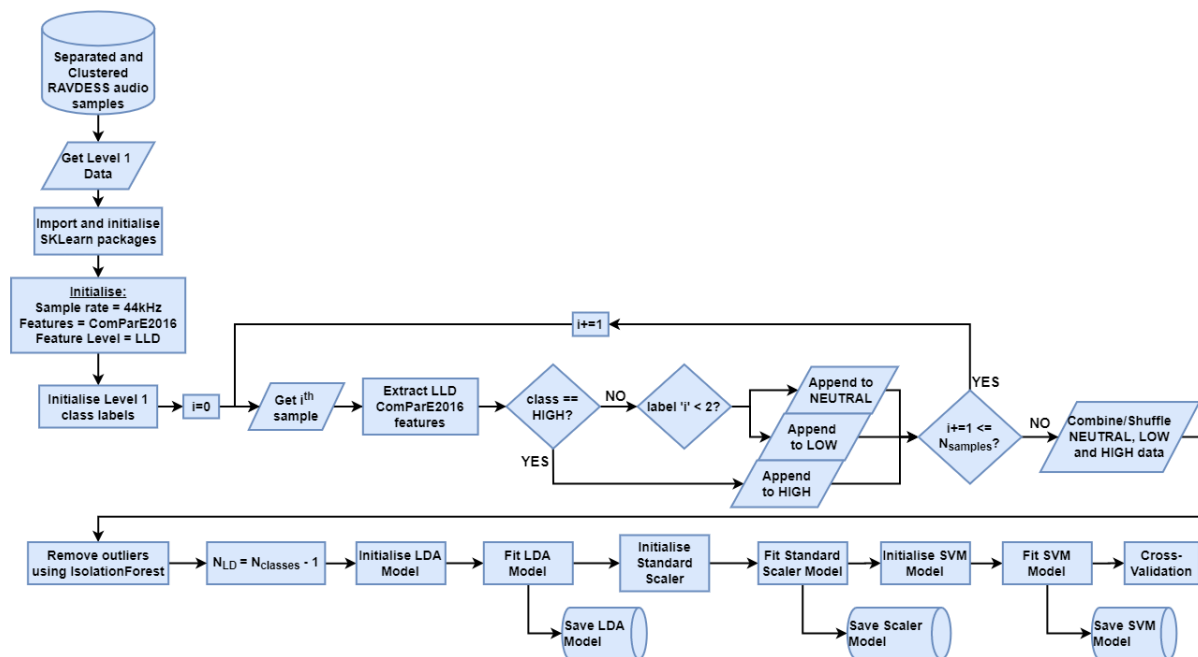


Figure 36: Affective speech Level 1 Classifier training architecture.

This section has explored the construction of back-end affective speech classifier models, including LDA, Scaler and SVM models required for assessing speech signals. In total, there are 21×3 unique models created (a triad/ensemble of LDA, Scaler, SVM models). Ten-fold cross-validation is a very common technique used to assess prediction error and was used to assess the SVM classification performance of all affective speech classifier models. k -fold cross-validation splits the input data into ' k ' equal parts and assess the prediction error across an entire dataset by iterating through the k parts and assigning a different validation set at each iteration [133]. The estimate of the cross-validation ' CV ' estimate can therefore be derived as [133]:

$$CV(\hat{f}) = \frac{1}{K} \sum_{i=1}^K L(y_i, \hat{f}^{-\kappa(i)}(x_i)) \quad (36)$$

where $\hat{f}^{-\kappa(i)}$ = fitted function, $\kappa(i)$ = an indexing function, i = observation/iterator. In this work $K = 10$. The ten-fold CV metrics are reported in Table 15 and Figure 38 and allow us to determine the ' O_{fp} ' for this modality.

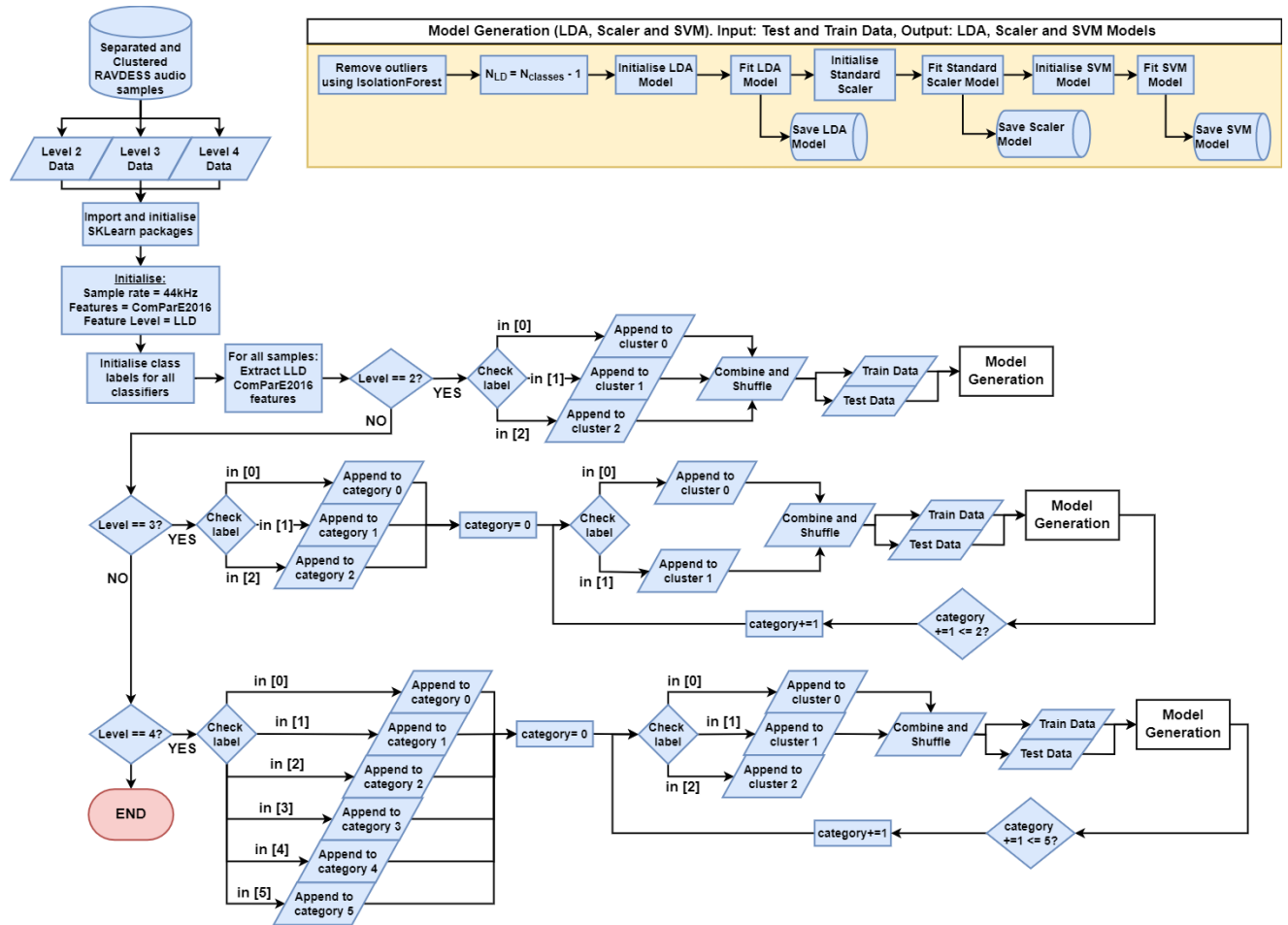


Figure 37: Levels 2, 3 and 4 affective speech classifier training architecture.

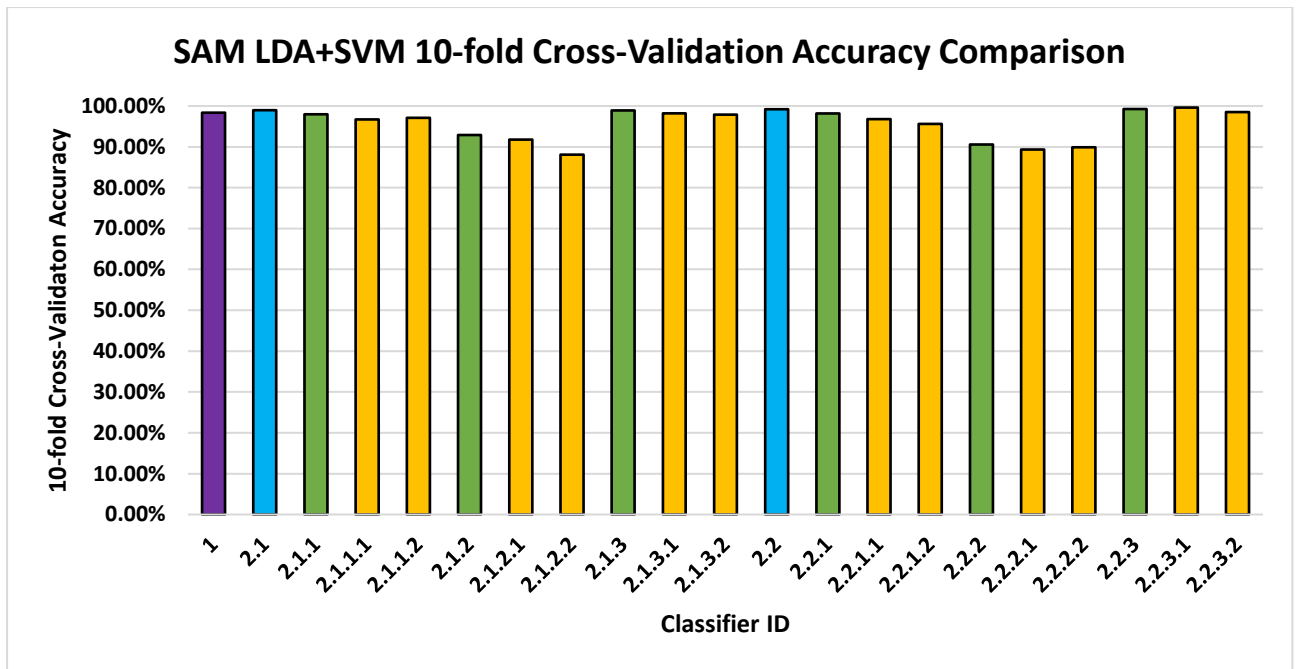


Figure 38: Visualisation of affective speech cross-validation accuracies using the LDA+SVM models discussed. Bars are colour-coded to show the level of classification where: PURPLE = level 1, CYAN = level 2, LIME = level 3 and YELLOW = level 4.

Index	Clf. ID	Activation	Phase	Class [0]	Class [1]	Class [2]	Acc _{val}
0	1	-	-	Neutral	Low	High	98.37%
1	2.1	LOW	-	Onset	Apex	Offset	98.97%
2	2.1.1	LOW	Onset	Happy Anger Surprise	Sadness Fear Disgust	-	97.97%
3	2.1.1.1	LOW	Onset	Happy	Anger	Surprise	96.69%
4	2.1.1.2	LOW	Onset	Sadness	Fear	Disgust	97.10%
5	2.1.2	LOW	Apex	Happy Anger Surprise	Sadness Fear Disgust	-	92.90%
6	2.1.2.1	LOW	Apex	Happy	Anger	Surprise	91.74%
7	2.1.2.2	LOW	Apex	Sadness	Fear	Disgust	88.06%
8	2.1.3	LOW	Offset	Happy Anger Surprise	Sadness Fear Disgust	-	98.92%
9	2.1.3.1	LOW	Offset	Happy	Anger	Surprise	98.20%
10	2.1.3.2	LOW	Offset	Sadness	Fear	Disgust	97.86%
11	2.2	HIGH	-	Onset	Apex	Offset	99.21%
12	2.2.1	HIGH	Onset	Happy Anger Surprise	Sadness Fear Disgust	-	98.18%
13	2.2.1.1	HIGH	Onset	Happy	Anger	Surprise	96.77%
14	2.2.1.2	HIGH	Onset	Sadness	Fear	Disgust	95.63%
15	2.2.2	HIGH	Apex	Happy Anger Surprise	Sadness Fear Disgust	-	90.57%
16	2.2.2.1	HIGH	Apex	Happy	Anger	Surprise	89.39%
17	2.2.2.2	HIGH	Apex	Sadness	Fear	Disgust	89.87%
18	2.2.3	HIGH	Offset	Happy Anger Surprise	Sadness Fear Disgust	-	99.26%
19	2.2.3.1	HIGH	Offset	Happy	Anger	Surprise	99.63%
20	2.2.3.2	HIGH	Offset	Sadness	Fear	Disgust	98.52%

Table 15: SVM model information and reported 10-fold cross validation accuracies obtained when using the RAVDESES dataset for training using the architecture and methods discussed in this section. A visualisation of these accuracies is shown in Figure 38.

Table 15 and Figure 38 report the cross-validation accuracies for the 21 classifiers embedded in SAM’s classifier ensemble for analysing affective speech signals. The average classification accuracy can be broken down per level:

- Level 1. Single classifier = **98.37%** validation accuracy.
- Level 2. Two classifiers = **99.09%** mean validation accuracy.
- Level 3. Six classifiers = **96.30%** mean validation accuracy.
- Level 4. Twelve classifiers = **94.96%** mean validation accuracy.

Thus, the average classification accuracy across all classifiers is **95.90%** for affective speech classification. Analysing these metrics, we see that on average, the classification accuracy is very high (greater than 90% for most classifiers) and when compared to facial expression classification as discussed in the previous section, it is far greater. By comparing this data to those reported in Figure 7, we again see that the overall performance of SAM when assessing affect in speech is competitive and can outperform other state of the art ASA systems.

This section has discussed the assessment of affective speech using SAM's novel hybrid learning architecture. It has shown that the same governing methodologies can be applied through two unique modalities, under the condition that the data is continuous. Through the classifier ensemble, SAM is capable of classifying activation, temporal phase, valence, and discrete states through a person's speech with a high degree of accuracy. The State Assessment Module is not limited to the classification of facial expressions and speech – as globally, the increase in the use of personal protective equipment (PPE) such as face masks has meant that ASA systems that rely on facial expressions become impaired and, in some cases, obsolete due to the presence of visual occlusions. SAM is adaptive in that it does not classify facial expressions when it detects that a visual occlusion is present on a person's face. The following section discusses the classification performance of the facial occlusion detection subsystem.

5.1.3 Facial Occlusion Classifier Subsystem

The goal of the facial occlusion classifier subsystem was to prove SAM as a robust, adaptive system that accounts for changing conditions and distorted data. Some of this functionality is reported in [25] i.e., the ability to detect facial occlusions such as face masks. The proposed solution came as a response to the increase in the use of face masks across the globe to mitigate the spread of viruses and diseases in light of the COVID-19 pandemic. Chapter 6 discusses the classification of upper-facial expressions when an occlusion is detected (also reported in [25]) but this has not been implemented as part of SAM's functionality.

At the highest level, when a facial occlusion is detected, SAM prompts the user to the fact that the full face that it is analysing is not clear, disabling its ability to classify the facial expression – this is visualised in Figure 39, which is a snippet of SAM's GUI.

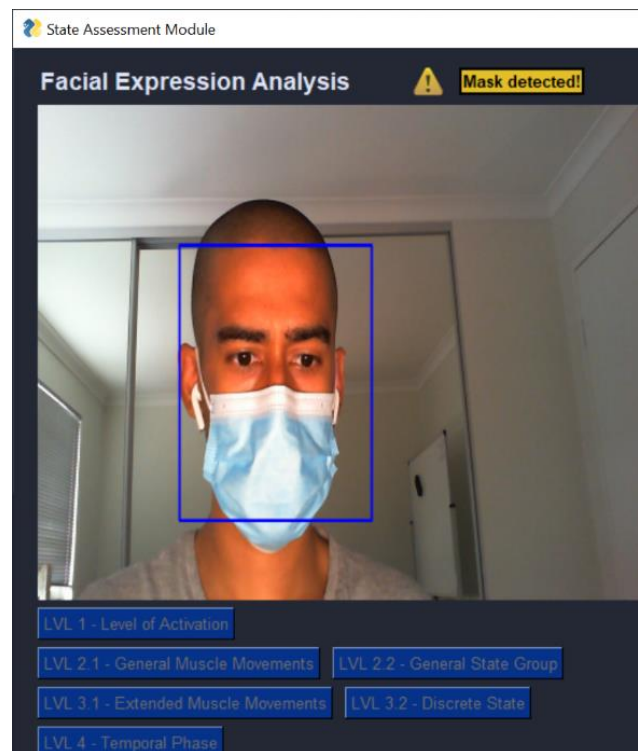


Figure 39: Screenshot of SAM’s home screen GUI window when a mask has been detected on a subject. Note the warning displayed in the top right and the disabled buttons.

In order to achieve this functionality, SAM deploys another classification schema, a binary classifier to detect two classes: “Masked” and “No Mask”. It deploys a combination of two datasets to train the two classes. Samples from the RAVDESS dataset were used to define the “No Mask” class and the MaskedFace-Net dataset [134] was used to define the “Masked” class.

The MaskedFace-Net dataset was developed as a direct response to the COVID-19 pandemic given the rise in the use of face-masks globally and contains image data pertaining to situations where masks are worn correctly and incorrectly [134]. When deploying the binary classifier, SAM treats both cases as one class – “Masked”, given that in both instances the assessment of a facial expression will be compromised by the presence of some visual occlusion.

To ensure that the training of classifiers was done fairly and to mitigate biases, each class contains the same number of samples: Masked = 17328 samples, No Mask = 17328 samples. The RAVDESS data used to construct the No Mask class were randomly allocated and amounted to approximately 22.7% of the total number of samples that were used for training the facial expression classifiers.

It was hypothesised that the classification would yield good performance results given the difference in a masked vs. an unmasked face, particularly if only the bottom half of image samples were used as inputs. Once a facial image has been detected as shown by the blue bounding box in Figure 39, SAM will split the facial data in half (slicing horizontally) and analyse the bottom half pixels to assess if a mask is present. In most cases the bottom half of the face has very distinguishable features including the mouth, nose, and nostrils; so, the presence of a face mask covers these features and provides a distinct separation between the two classes. A visualisation of how the LDA-transformed data is clustered is shown in Figure 40, showing the distinct separation that was hypothesised.

The classification of facial occlusions follows a similar method to affective speech i.e., combining LDA for transformation, a scaler for standardisation and an SVM for classification, similar to what is discussed in [25]. As discussed above, the separation of classes is distinct when comparing masked and non-masked faces and as such, it was decided that implementing a statistical classifier would optimise performance and accuracy in favour of a more complex neural network or deep learning solution. A summary of this process is shown in Figure 41.

The classification performance of the facial occlusion classifier subsystem was done through a ten-fold cross validation task. Observing Figure 40, we can see that the classification accuracy should be high to be consistent with the separation of clusters. The reported cross validation accuracy of this subsystem was 99.973% \approx 100% which matches the initial hypothesis.

Being the first classifier that SAM deploys for facial expression assessment, it is important that the validation accuracy be very high as the cost of misclassification at such a high-level would adversely affect the performances of the rest of the system to a high degree.

With all classifiers discussed in this section (Section 5.1), we can establish how a continuous video input is split into facial and speech signals, processed, and classified. Across all three classifier subsystems, SAM is capable of detecting 116 unique classes/states in. The classification of these signals is visualised in Figure 42 and the interaction between front-end and back-end elements of the system are discussed in more detail in the following section (Section 5.2).

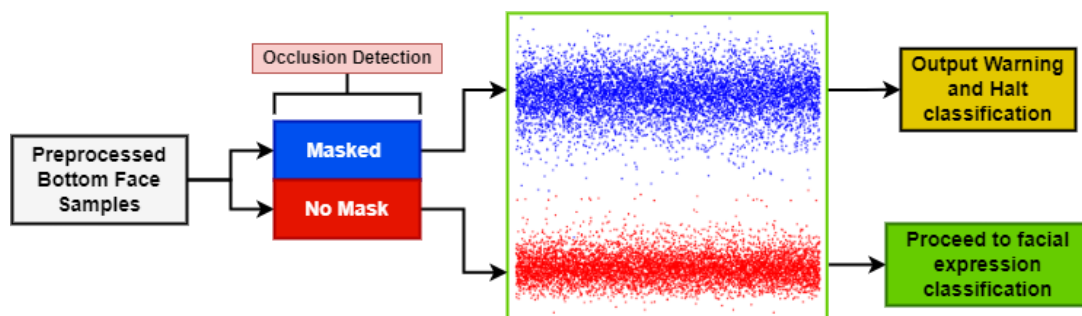


Figure 40: Visualisation of how the “Masked” and “No Mask” samples are clustered after projection onto a 1-dimesional linear discriminant space using an LDA transform. It also shows how the facial occlusion classifier subsystem interacts with the facial expression classification subsystem.

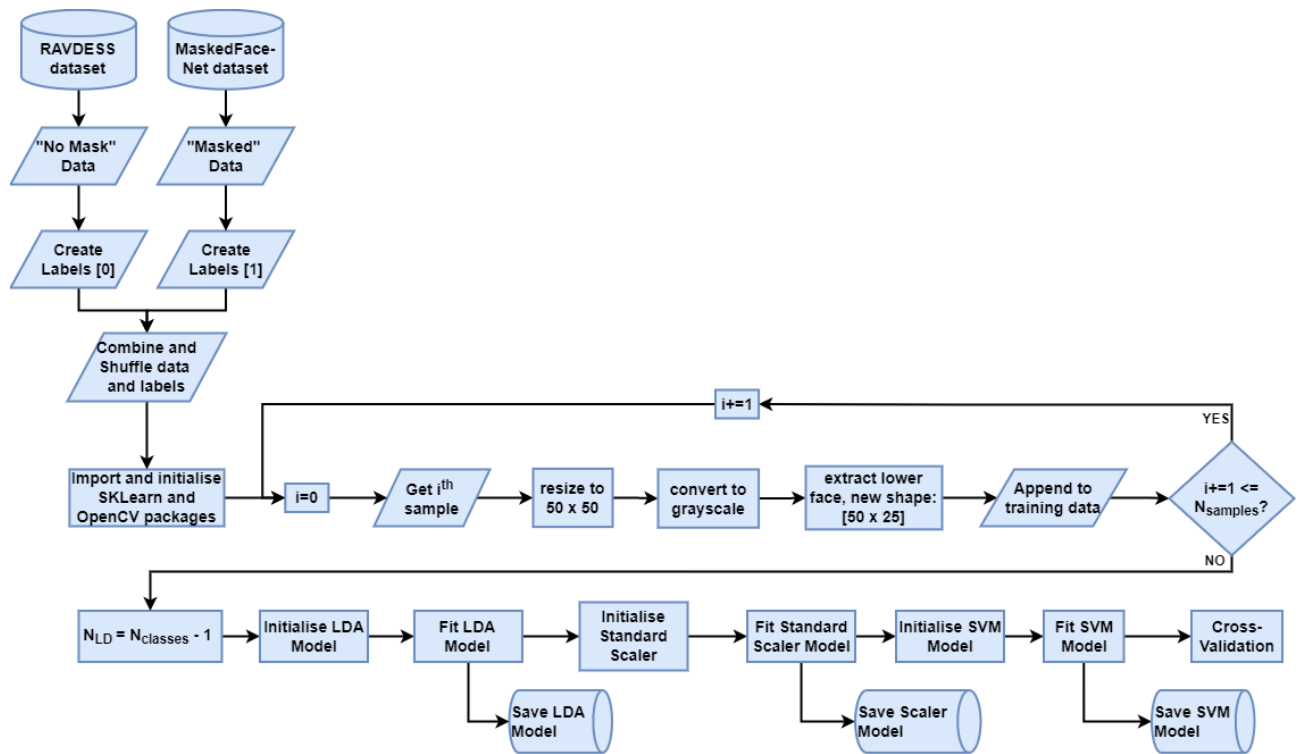


Figure 41: Visual occlusion binary classifier training architecture showing how data from the MaskedFace-Net and RAVDESS datasets are combined and transformed to detect visual occlusions such as face masks.

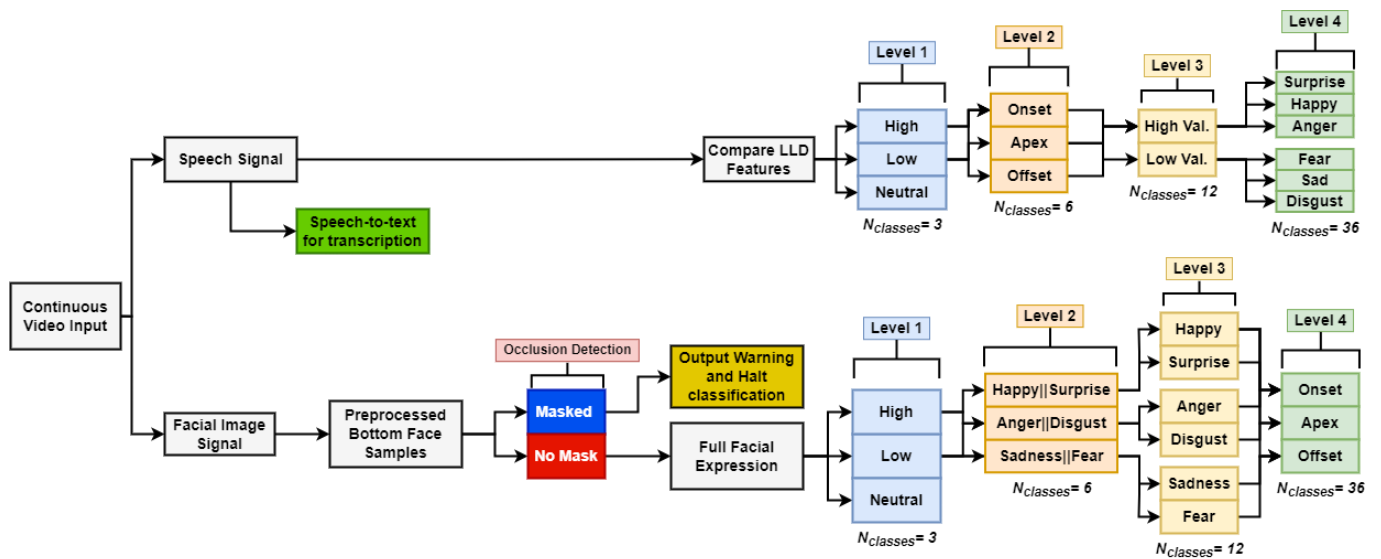


Figure 42: Summary of SAM’s multimodal classification network showing the transformation and processing of data from a raw video input to the classification of states using the three classifier subsystems discussed in section 5.1.

5.2 Implementing SAM’s Graphical User Interface and Frontend Design

5.2.1 Graphical User Interface Purpose

SAM’s graphical user interface (GUI) was designed to provide users with the ability to analyse its decision-making processes including: inputs, processed data, and outputs. By integrating the classifiers that were discussed in the previous section, information is parsed to the necessary windows. The GUI also serves as vehicle to apply other back-end subsystems including its speech-to-text, working memory and database manipulation processes. Figures 43 and 44 along with Table 16 help visualise and discuss the purpose of each of SAM’s GUI windows as well as how data is parsed and manipulated throughout.

Previously, it was mentioned that SAM is an extensive evolution of the ASAM discussed in [1, 19, 108]. After a review of the functionality of that system, it was declared that vast improvements were required to make it a more powerful, dynamic ASA system. While being a functional assessment tool, the ASAM was lacking in various aspects and could be improved upon. Thus, SAM was invented and constructed from the ground-up as a more powerful system that dwarfs the capabilities of the ASAM.

As detailed in Section 4.3, the SDDP of SAM can be considered as an iterative process leaning more towards an agile model. Throughout iterations, the GUI was expanded upon, and more functionality was added to the system. To ensure that SAM’s software was organised, code was developed in a modular fashion and functions were contained directories related to their functionality. Figure 45 visualises the directory and file structure of the latest SAM package.

This section discusses each of the GUI windows outlined in Table 16, highlighting:

1. Software algorithms, the flow of information and the parsing of data (through flowcharts),
2. The design, process, and structure of generated reports,
3. The back-end classifiers that are deployed (if classification is involved),
4. Visual assets that are used to indicate changes in states,
5. SAM’s dialogue, interactions and working memory.

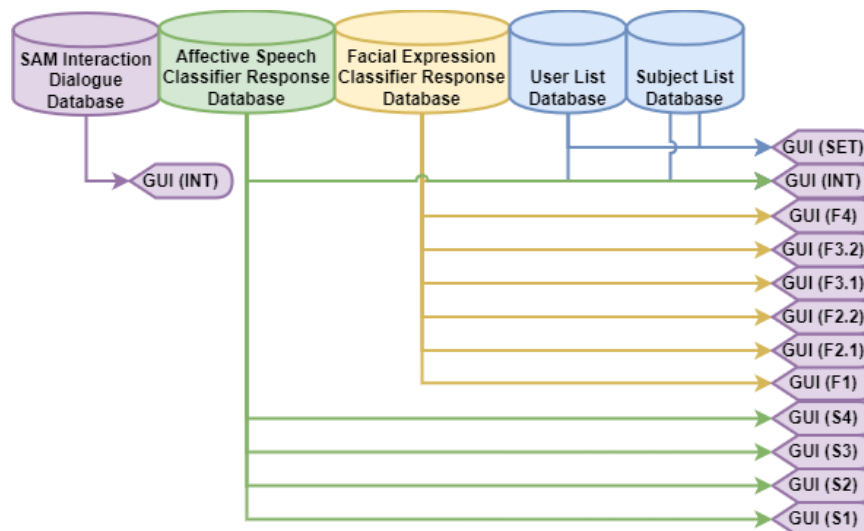


Figure 43: Visualisation of how databases are parsed into the system and what GUI windows exploit the database information including dialogue responses, user and subject information and classifier responses for facial expression and speech analysis.

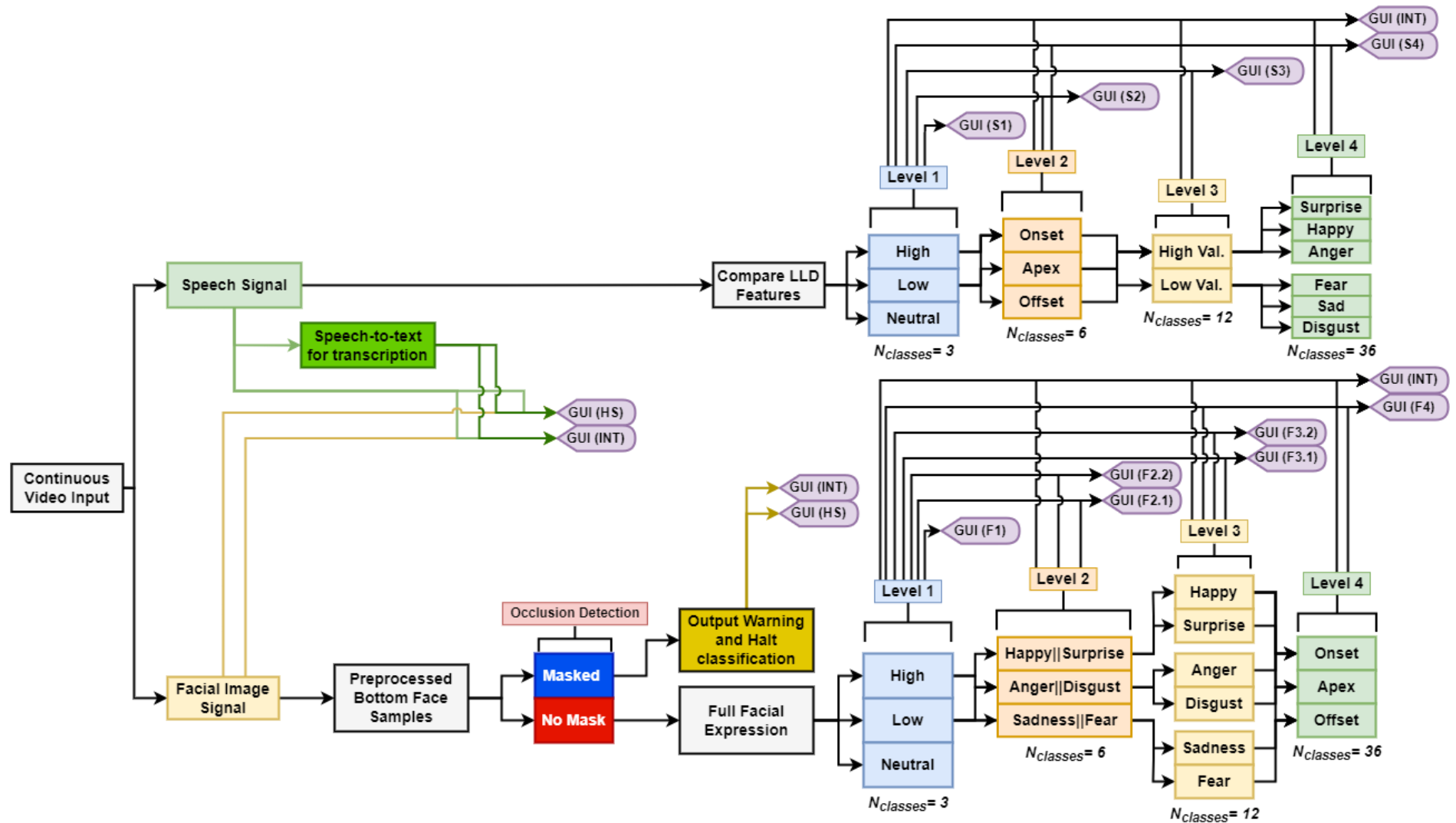


Figure 44: Extension of Figure 42, visualising a high-level representation of how data is parsed through the system from input → processed states → outputted classification results across all of the GUI windows.

GUI Window Label	Name	Purpose and Capabilities	Data Involved
GUI (HS)	SAM Home Screen	Monitor input data in real-time including: camera input, microphone feed, speech-to-text transcription, Monitors the status of the system including if a mask is detected and/or if peripheral devices are removed among others, Acts as the hub to access other GUI windows, Grants the ability to generate interaction analytics reports and logs	Raw video input → camera + audio input
GUI (F1)	LVL1: Facial expression level of activation classification	Monitor changes in facial expression level of activation, Generate reports from the interaction	Facial expression image data
GUI (F2.1)	LVL2: General facial muscle movement classification	Monitor changes in general facial muscle movements and muscle regions, Monitor changes in the level of activation, Generate reports from the interaction	Facial expression image data
GUI (F2.2)	LVL2: Facial expression state group classification	Monitor changes in facial expression state group, Monitor changes in the level of activation, Generate reports from the interaction	Facial expression image data
GUI (F3.1)	LVL3: Facial AU/extended facial muscle movement classification	Monitoring of Facial Action Unit (AU) changes and the corresponding muscle movements, Monitor changes in the level of activation, Generate reports from the interaction	Facial expression image data
GUI (F3.2)	LVL3: Discrete facial expression classification	Monitor changes in discrete state/emotion, Monitor changes in the level of activation, Generate reports from the interaction	Facial expression image data
GUI (F4)	LVL4: Facial expression Temporal phase classification	Monitor changes in temporal phase relative to facial expressions, Monitor changes in discrete state/emotions, Monitor changes in the level of activation, Ability to change 'verbose' setting to communicate extended muscle movement related information, Generate reports from the interaction	Facial expression image data
GUI (S1)	LVL1: Affective speech level of activation classification	Monitor changes in the level of activation w.r.t to input speech signals, Generate reports from the interaction	Speech signal data
GUI (S2)	LVL2: Affective speech temporal phase classification	Monitor changes in temporal phase relative to speech, Monitor changes in the level of activation, Generate reports from the interaction	Speech signal data

GUI (S3)	LVL3: Affective speech valence and general speech characteristics classification	Monitor changes in valence (high vs. low) for speech, Monitor changes in general speech characteristics, Monitor changes in the level of activation, Monitor changes, in temporal phase, Generate reports from the interaction, Ability to change 'verbose' setting to communicate either valence information or speech characteristic information	Speech signal data
GUI (S4)	LVL4: Affective speech discrete state and extended speech characteristics classification	Monitor changes in discrete state/emotion through speech, Monitor changes in extended speech characteristics, Monitor changes in temporal phase, Monitor changes in the level of activation, Generate reports from the interaction, Ability to change 'verbose' setting to communicate either valence information or speech characteristic information.	Speech signal data
GUI (SET)	Settings and login window	Ability to add new users/subjects to SAM's database, Ability to log in/out using user and subject credentials, Ability to view details of a currently logged in user/subject – ability to understand the people in the interaction, Ability to access SAM's README, Change SAM functionality based on user's permissions	Subject database, User database
GUI (INT)	SAM interaction window	Provides the ability to communicate with SAM using 21 unique commands, Ability to have two-way communication (verbal) with SAM, Ability to analyse instantaneous predictions for all levels and modalities of classification	Subject database, User database, SAM responses database, Speech signal data, Facial expression image data, Speech signal data

Table 16: Summary of SAM's GUI functionality for each of the windows, reporting the labels (which are shown in Figure 43), the purpose of each window and the data that is used to achieve functionality within that window.

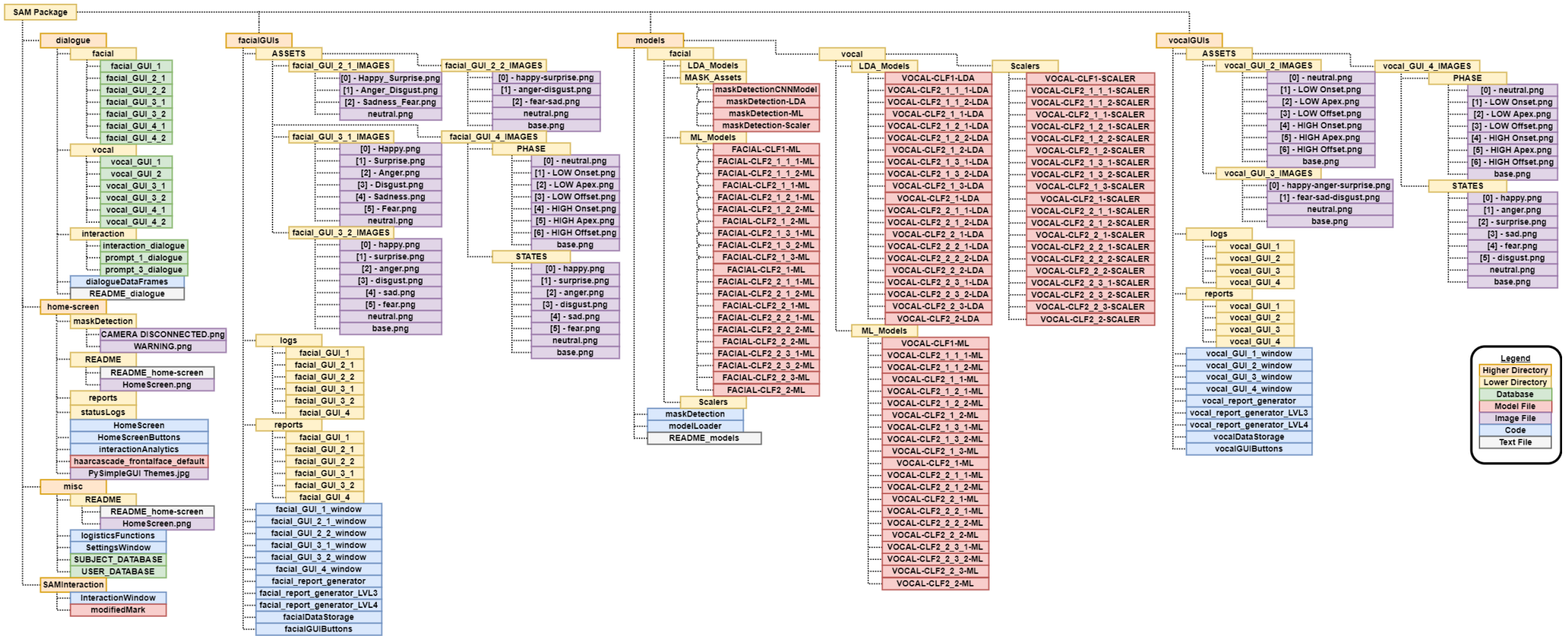


Figure 45: Construction and tree diagram of the SAM package showing how system elements are partitioned throughout the system. The legend on the bottom right of the figure assists in explaining the how elements are colour coded. The algorithmic implications of the SAM package are discussed throughout Chapter 5 and highlight the modular nature of the system.

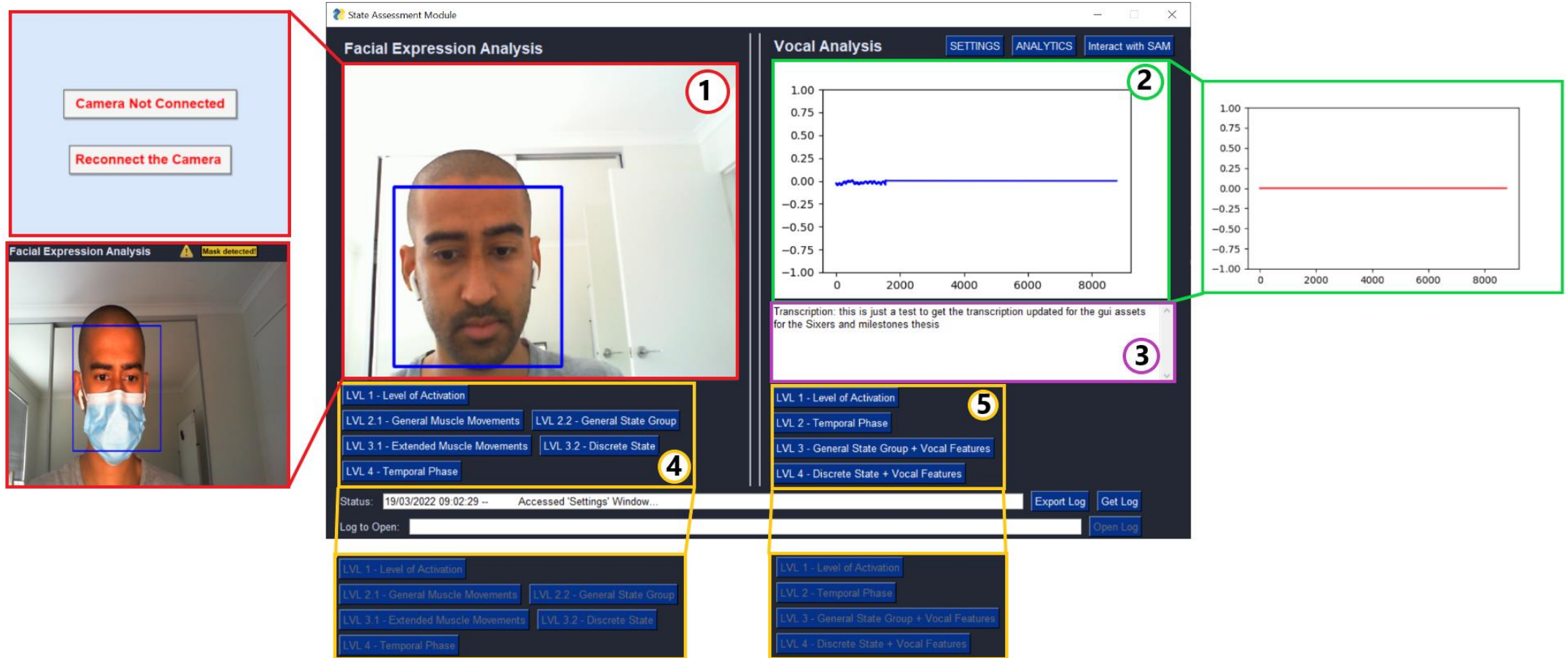


Figure 46: Annotated screenshots of SAM’s home screen GUI window, showing examples of its various states and functions that are reported in flowcharts in Figures 47-54. We see that the window can be partitioned into five major sections: (1) camera input, (2) speech input, (3) speech-to-text, (4) facial expression analysis window buttons and (5) affective speech analysis window buttons.

5.2.2 Home Screen – GUI (HS)

The home screen GUI window is the interface that is opened when SAM's executable is launched. The executable is built from the "HomeScreen" source code. Figure 46 shows an annotated version of the home screen GUI window of the latest version of SAM. Together with the flowcharts that follow in Figures 47-54, the overall functionality and flow of information throughout the home screen can be reported and discussed. Analysing Figure 46, the GUI (HS) window is partitioned into five major areas:

1. Real-time camera input

In general, the home screen provides transparency of input signals. The "real-time camera input" area relays the camera input to the user, providing a vehicle in which the user can assess what features are being used for SAM's decision-making processes. Research conducted in [70, 72, 76] outlines the importance of being able to see raw representations of data. In the context of critical applications where a domain-expert is using the system, it is important that they are able to view the input data and overrule SAM's decision-making processes, passing the *accountability* from the machine to the user. The AXAI framework highlights the inspect-ability of input models or cues – ' I_{in} ' as a key parameter for assessing the accountability of a system and showing the raw data to the user is a step in achieving this.

Furthermore, see that SAM is capable of adapting to changes in system status', including changes in what it is presented in front of it, as well as if peripheral devices have been removed during an interaction. As shown in the figure, SAM is capable of detecting the presence of a visual occlusion like a protective face mask through its facial occlusion classifier subsystem which was discussed in Section 5.1.3. Once an occlusion is detected, a warning is output, and the facial expression analysis GUI window buttons (area 4) are disabled. Disabling the classifier is important as classifying the full facial expressions of a masked face would produce erroneous results – given half of the facial features are obstructed.

When SAM detects that a peripheral device has been removed, it displays this information to the user. Not only does this assist in the debugging process [4, 70], it also aids in improving the accountability of the system. By highlighting to the user that the peripheral device is disconnected, it shows users that SAM is unable to make an assessment and as such, cannot be accountable for incorrect decisions being made in the time that it has no input data to assess. As shown in Figure 51 and Appendix F, this information is reported so that when an interaction is analysed in post, an interested party can view if and when a peripheral device was disconnected and who would be accountable if any errors are observed during this window.

2. Real-time microphone input

Like the camera input, the microphone input is updated in real-time and provides users and domain experts the ability to view the input speech signal that is being used for classification and speech-to-text processes. With a sample rate of 44kHz and a chunk size of 4389, through the x-axis, we can see that two chunks of data are displayed to the user at any time. Again, showing the input data introduces a degree of transparency, and improves the accountability of the system (through its ability to inspect input signals). Like the camera input, it is important that the user is alerted when a peripheral device has been disconnected. This is achieved by outputting zeros to the GUI window and changing the colour of the line to red. Furthermore, the ability to access the vocal analysis GUI windows is disabled

along with speech classification processes as SAM is unable to 'hear' in the interaction. Throughout prototyping, a video camera was used for experimentation and development and removing the peripheral device would cause both error conditions to be raised. However, SAM is capable of handling separate microphone and camera input devices.

3. Speech-to-text integration

The third area displays the speech-to-text output to the user through the transcription box. This work deploys the Python speech recognition package [135] – specifically, the integrated Google speech-to-text application programming interface (API). Detecting what has been said during an interaction allows users to communicate with SAM and also monitors the linguistic components of an interaction. While linguistic ASA is not integrated into this system, this process allows for future integration of such capabilities. Speech-to-text is deployed on a separate thread to the main thread as failure to do so interrupts the main loop while the transcription is being processed through the API. The transcription is stored and can be replicated in a report to monitor what a subject has said during an interaction. As will be discussed in section 5.2.6, the transcription of speech is vital for two-way communication with SAM as it allows for an understanding of user requests.

4. Facial expression analysis GUI window access

Previous sections have discussed SAM's hierarchical facial expression classifier ensemble, containing 21 unique classifiers capable of detecting 57 unique states with an average accuracy of approximately 83%. The classifiers are deployed across six different GUI windows which will be discussed in section 5.2.3, with the data being analysed at each level of classification being discussed in previous sections.

This area of the home screen GUI allows users to access the multi-level facial expression classifier results if the buttons are enabled. Figure 46 highlights an example of all the buttons being disabled. The status of the buttons is dependent on variables including: (i) if a camera peripheral device is connected, (ii) if a person is wearing a mask, (iii) if a user and subject have logged into SAM, (iv) the permissions of the user.

SAM checks the permission level of the user and determines what button states are activated as a result. These levels are arbitrary and are used to highlight how this functionality could be implemented into an applicable setting. SAM understands three levels: (i) 'master' – equivalent to ROOT, who can access all windows, (ii) 'admin' – who can access a subset of windows specifically GUI (F1, F2.2, F3.2) and (iii) 'guest' – who can only access GUI (F1). Whenever SAM adds a new user to its database, they are assigned the 'guest' permission level by default.

5. Vocal/Speech analysis GUI window access

Like area 4, this area consists of all the buttons that allow users to navigate through the affective speech assessment GUI windows that deploy the hierarchical affective speech classifier ensemble models discussed previously. Unlike the facial expression assessment windows, there are only four vocal/speech analysis GUI windows (one for each level). This is due to the amount of information displayed to the user through speech assessment. Condensing all the speech assessment information into four windows was achievable, unlike the facial expression analysis GUI windows where attempting to do so would be a counterproductive burden [3].

The status of the buttons in this area are dependent on: (i) if the microphone has been disconnected, (ii) if the user and subject have logged in, and (iii) the permissions of the user. The permissions levels outlined above are consistent for speech analysis and govern button status' as such: 'master' – all windows are accessible, 'admin' – all windows bar GUI (S4) are accessible and 'guest' – only GUI (S1) is accessible. Again, this information is visualised in Figure 54. Other elements and GUI window functions that don't fall into a specific category are:

- Ability to access the settings window GUI (SET),
- Ability to generate an Analytics report, with an example report shown in Appendix F,
- Ability to access the SAM interaction window GUI (INT),
- Status bar showing updates of the system, providing users with the ability to scrutinise and debug the system while providing further transparency of what the SAM is doing,
- Ability to 'Export', 'Get' and 'Open' log files.

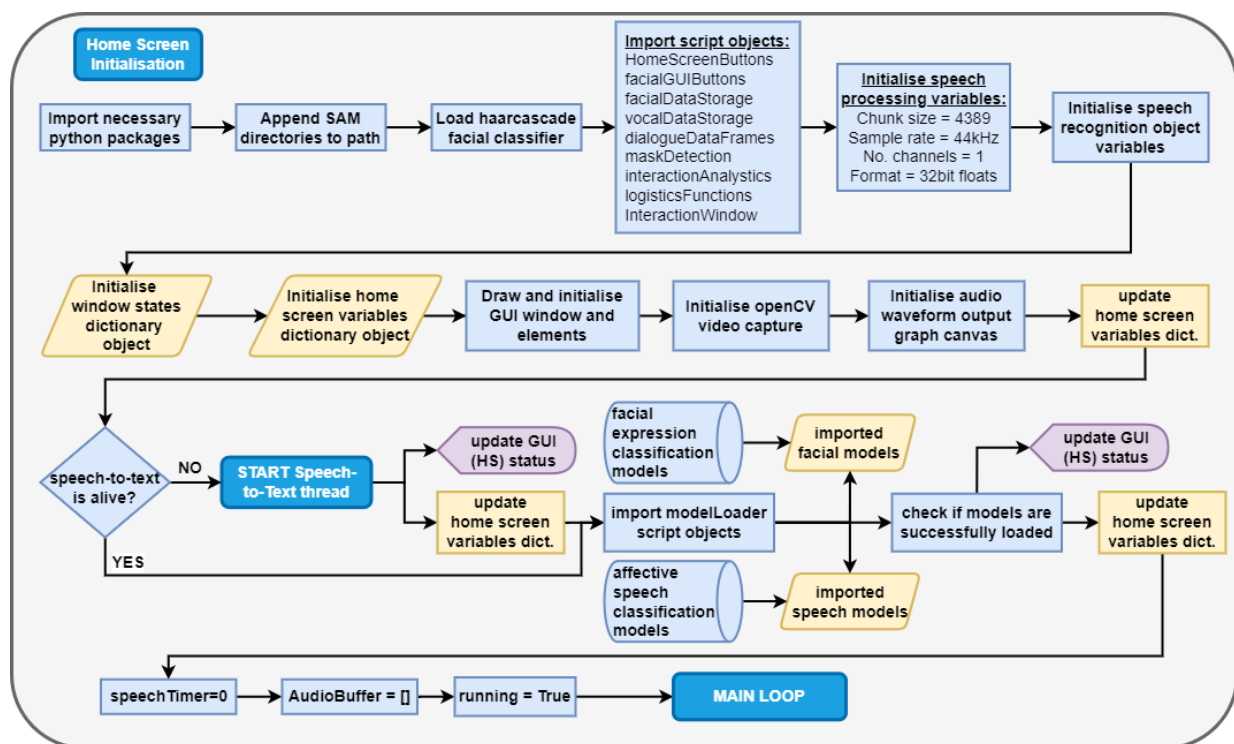


Figure 47: Home screen GUI initialisation function block. When SAM is launched, this is the first code block that is executed. Other function blocks discussed in this section are highlighted in light blue e.g., the “MAIN LOOP” function block.

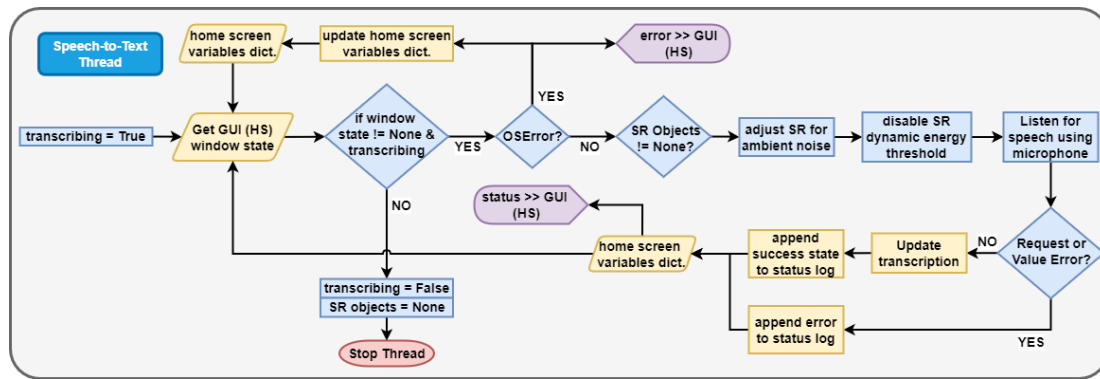


Figure 48: Speech-to-text function block – which is executed on parallel thread separate to the main thread. This provides SAM the ability to understand what a user has said in an interaction and provides the foundation for two-way communication.

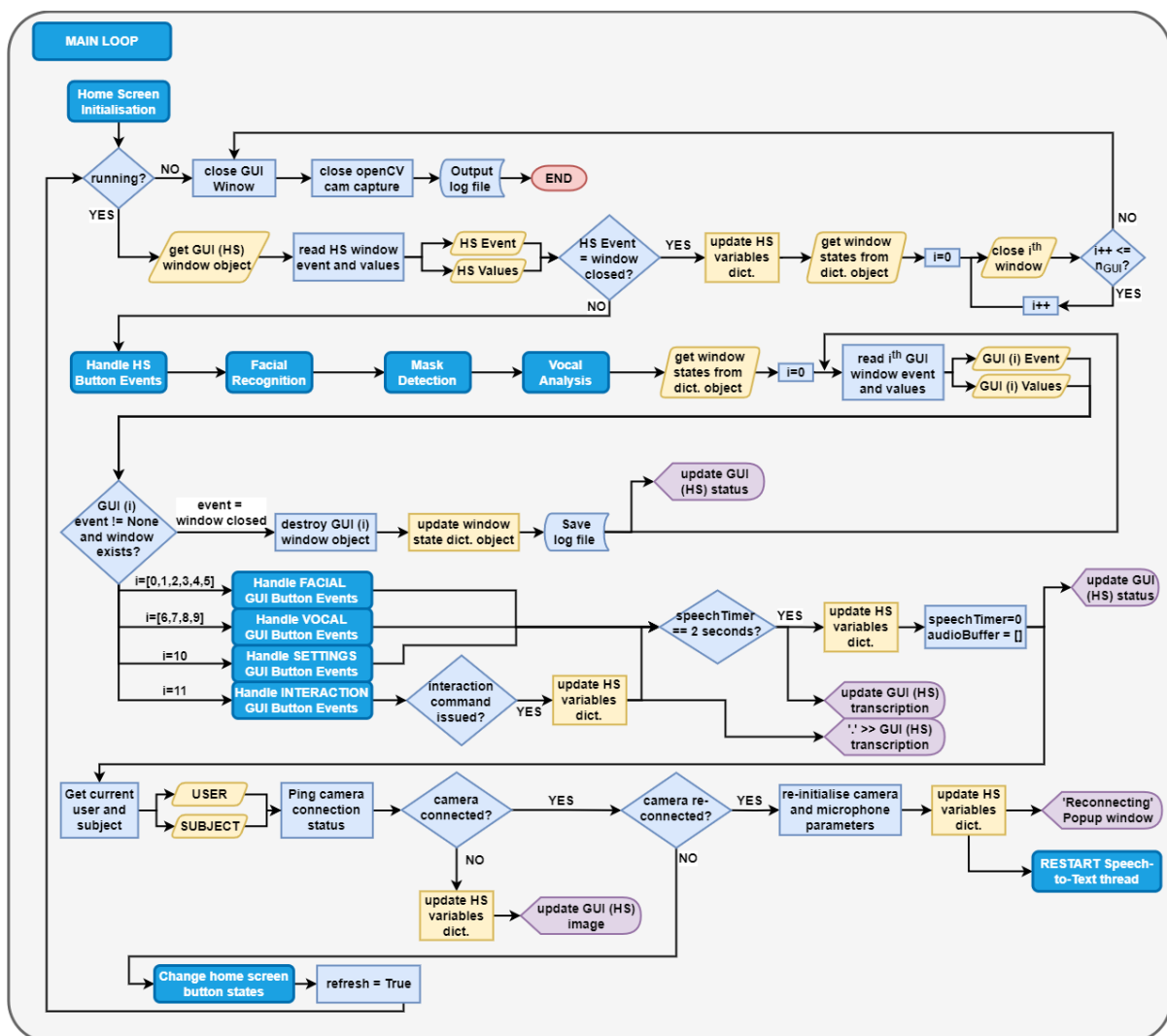


Figure 49: SAM’s main loop function block which is run after the home screen initialisation function block visualised in Figure 47. The event handling function blocks i.e.: (i) Handle FACIAL GUI Button Events, (ii) Handle VOCAL GUI Button Events, (iii) Handle SETTINGS GUI Button Events, (iv) Handle INTERACTION GUI Button Events are discussed in the relevant sections in this chapter.

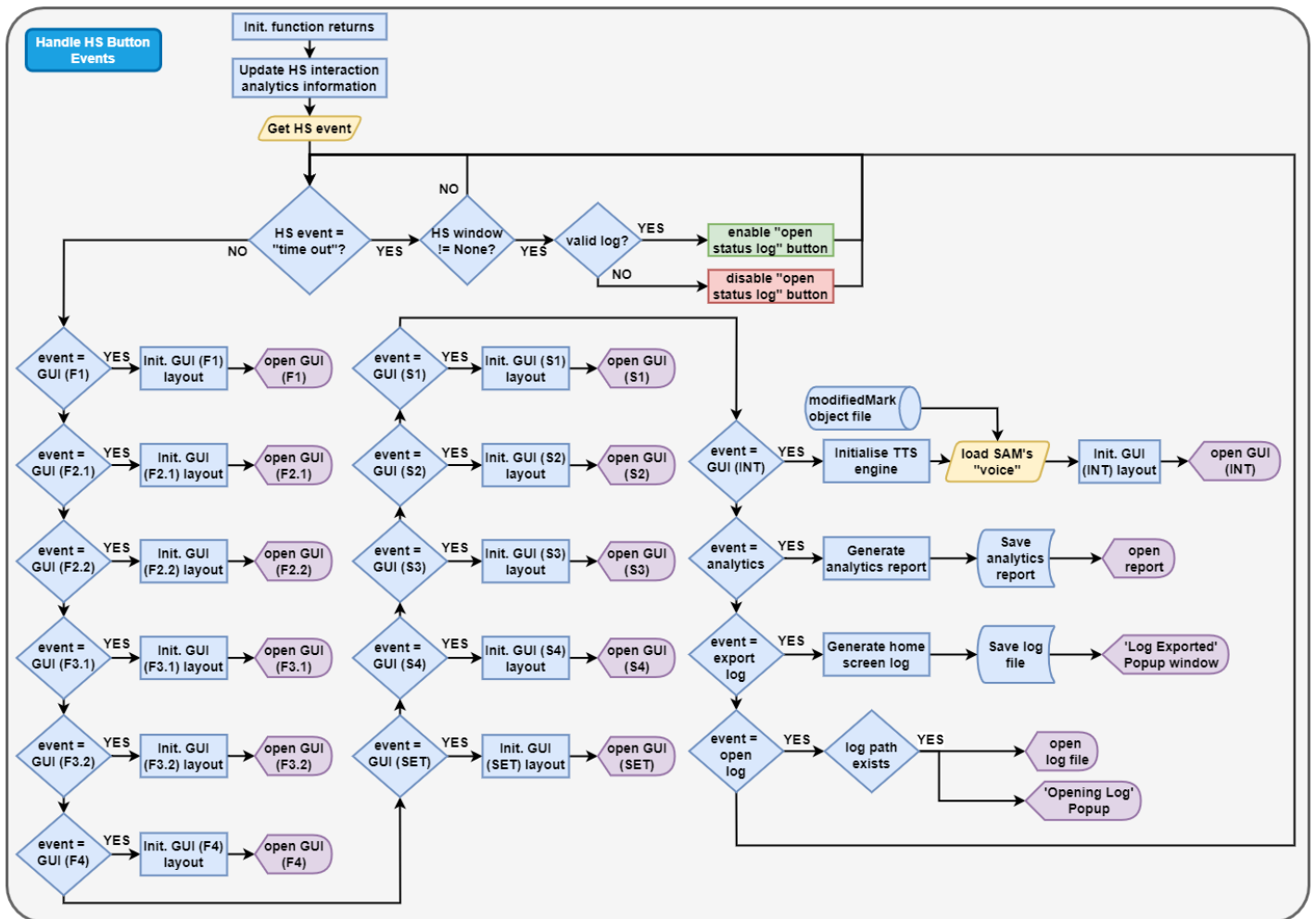


Figure 50: Home screen GUI event handler function block. This describes how other windows are initialised and opened from the home screen.

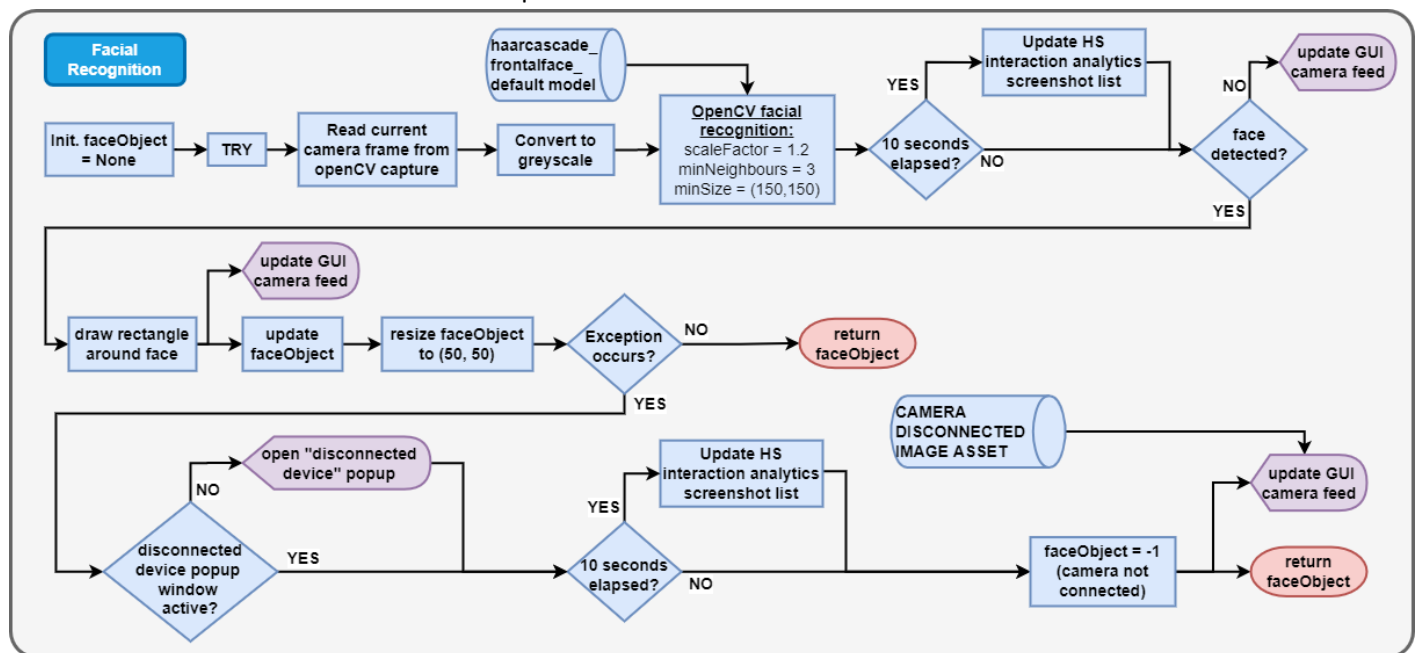


Figure 51: Facial recognition algorithm/function, called in the main loop, relying on Python’s OpenCV package [136] and highlighting some of SAM’s error handling.

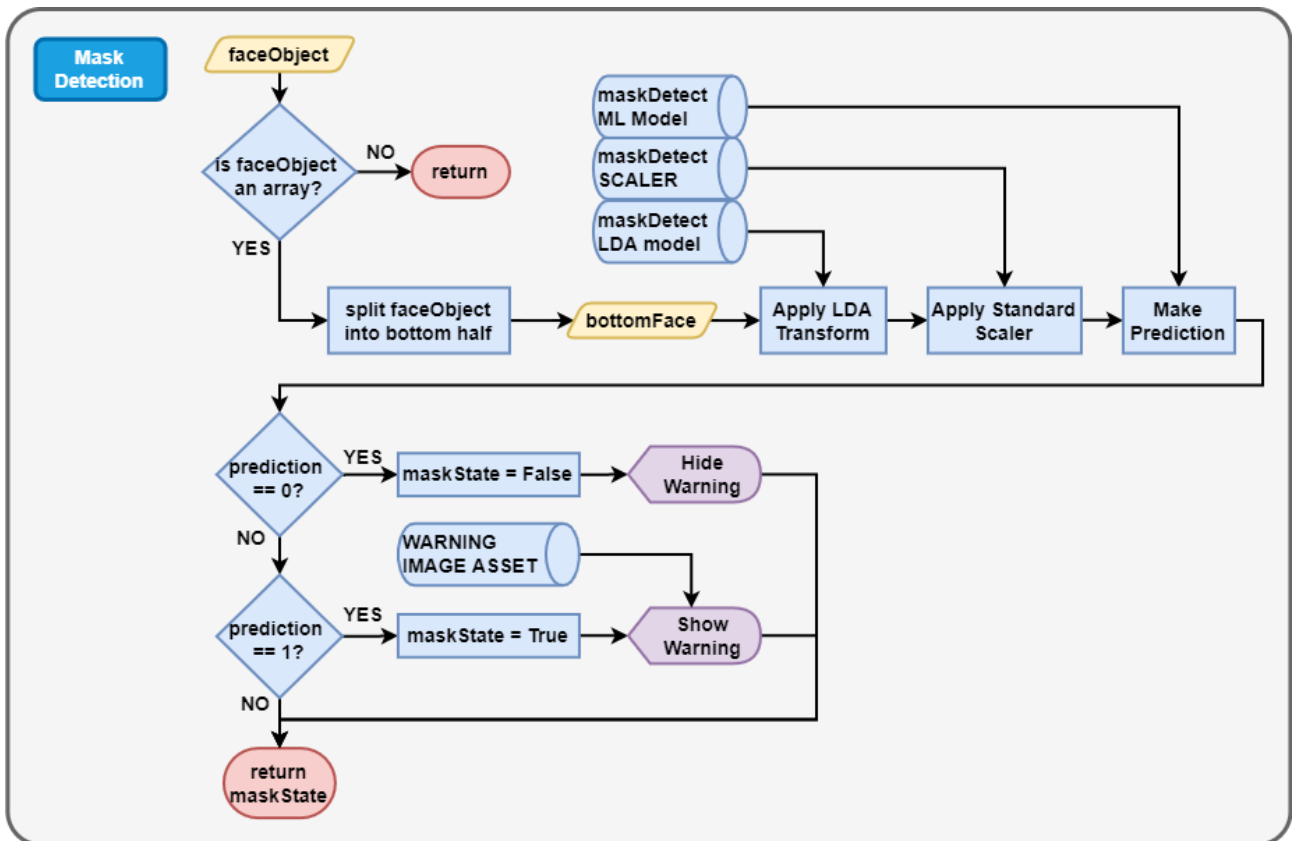


Figure 52: Mask detection function block deployed in the main loop. This is an algorithmic representation of how SAM deploys the facial occlusion classifier subsystem.

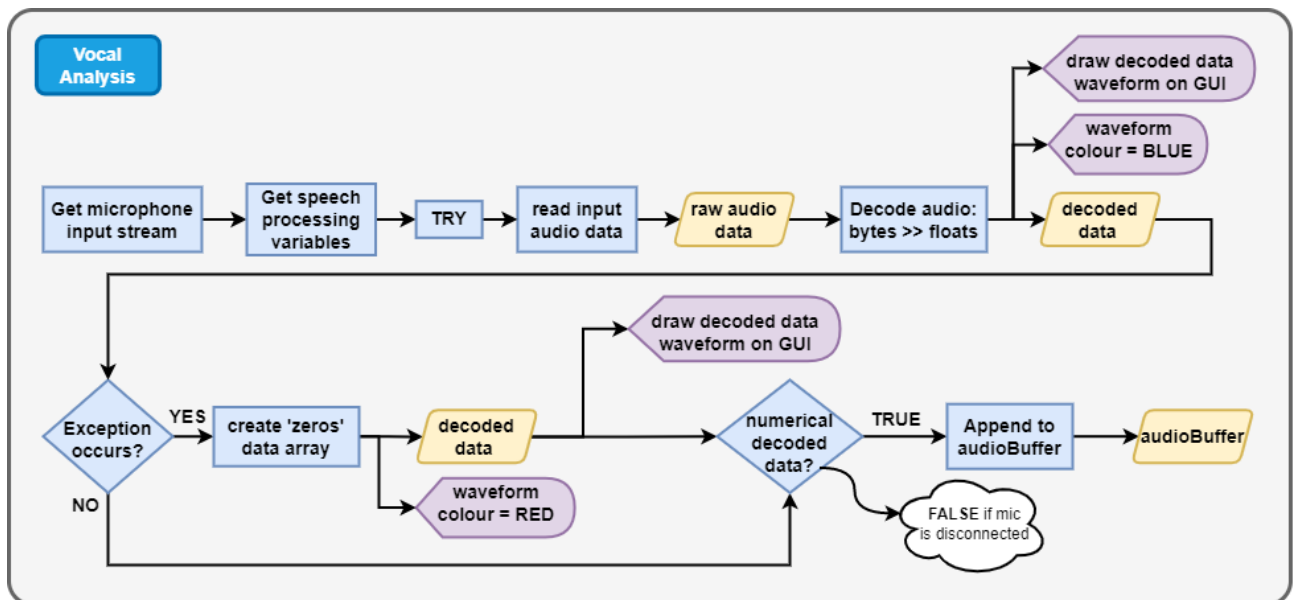


Figure 53: Vocal/Speech analysis function block called in the main loop. Note that the error handling functionality visualised in Figure 46 is represented through this flowchart.

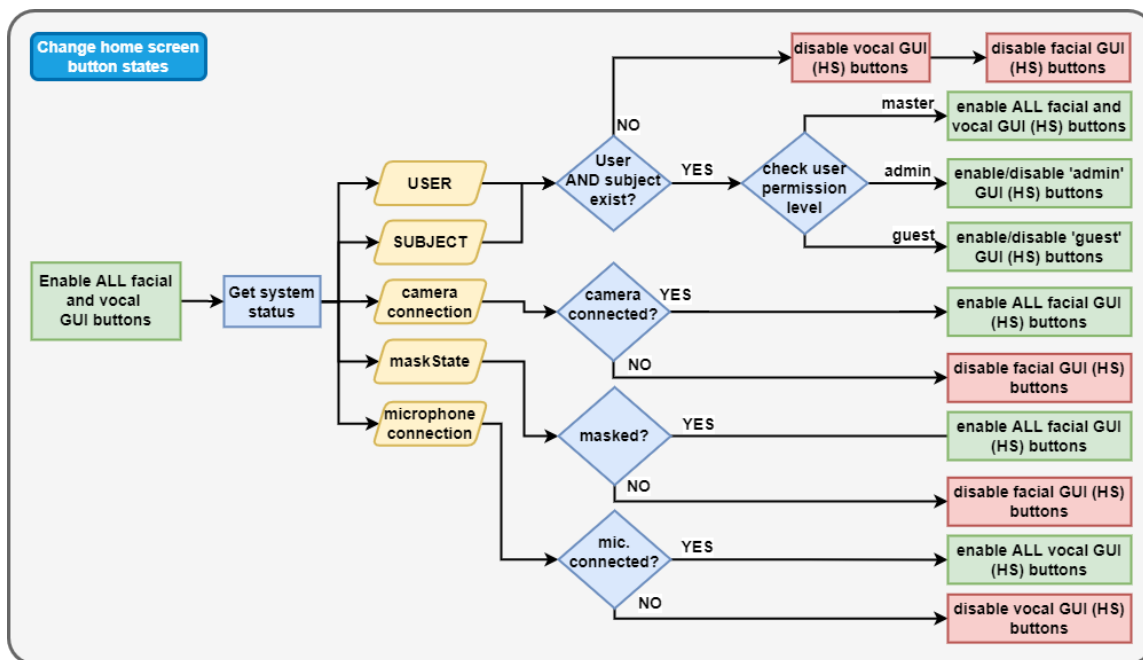


Figure 54: Home screen GUI button state changing function called at the end of the main loop to check if the SAM’s status’ have changed regarding the user, subject, permission, peripheral devices, and presence of visual occlusions.

5.2.3 Facial Expression Analysis Windows – GUI (FX)

SAM’s facial expression analysis windows denoted by ‘GUI (FX)’, are accessed from the home screen as shown in Figure 50. SAM was designed to allow users the ability to view as much, or as little information as required – to optimise what they are looking for. Thus, all GUI windows spawned by SAM are ‘non-blocking’ and operate on separate threads to allow for the parallel viewing of information. When a window is closed, the child thread is killed. The classification and button events for each of these windows is handled using various button event handling functions.

The facial expression analysis windows allow for the connection between the back-end classifiers and the front-end, providing visual assets to assist in explaining the decision-making process. Previous attempts at designing a previous system (ASAM) saw the visualisation of graphical prediction data on the analysis windows at all times [108]. It was found that given the increase in the number of windows and classifiable states, this amount of information in every window, would not be representative of an explainable, comprehensible system. Furthermore, removing this functionality improves SAM’s performance as there are less graphics to handle. As such, it was decided in early stages of the development process to provide users with the ability to export data and generate reports of current and historical interactions. As a result, all GUI (FX/SX) windows contain the same three buttons:

1. Export Data

Throughout an interaction, while the GUI windows are updating the visual assets with the current classifier predictions, this data is also time-stamped and stored in a two-dimensional array i.e.: [timestamp, classifier prediction]. The user has the option to export this data and save it for historical use, or use it generate reports of the current interaction. The exported data is saved in a dedicated ‘logs’ directory that has been created for the window in use (see Figure 45), with the data being exported in a .csv format.

2. Get Data

Opens a file explorer to choose a log file that will be used to generate and view a report.

3. Open Data/View Reports

Using the log file that has been selected using the “Get Data” button, this button uses the log file information and generates a report of the data, containing graphical information of classifier predictions throughout an interaction. Examples of generated reports are seen in Appendices G to L.

Along with the three buttons listed above, the general layout the facial expression analysis windows consist of: (i) animation assets to visualise predictions, (ii) tabular representation of features or prediction results, (iii) dialogue output box showing SAM’s interpretation of the prediction and explanation as to why the state is predicted and (iv) a status bar similar to the home screen. Figure 55 shows an example of all windows open at the same time showing the outputs of all windows to the same expression. Figure 56 is a collection of the facial expression analysis GUI windows.

Note that in Figures 55 and 56, the dialogue is not always reflective of the state being represented through visual assets, this is because dialogue is only updated every ≈ 2 seconds, outputting the mean state that was predicted over that time, whereas predictions are updated continuously. This decision was made because updating the dialogue with every prediction (rate of ≈ 0.2 seconds) would not give users enough time to read dialogue outputs thus, the GUI window dialogue would be redundant. This process of updating the GUI dialogue fields is visualised in flowcharts figures in this section.



Figure 55: An example of all facial expression GUI windows being open in parallel showing a low activation, apex temporal phase, surprise state (at level 4). We see that the activation (level 1) is present in every GUI window, with GUI (F2.1/2.2) showing the state group, GUI (F3.1) showing the FACS facial action units and GUI (F4) showing the temporal phase and discrete state information.

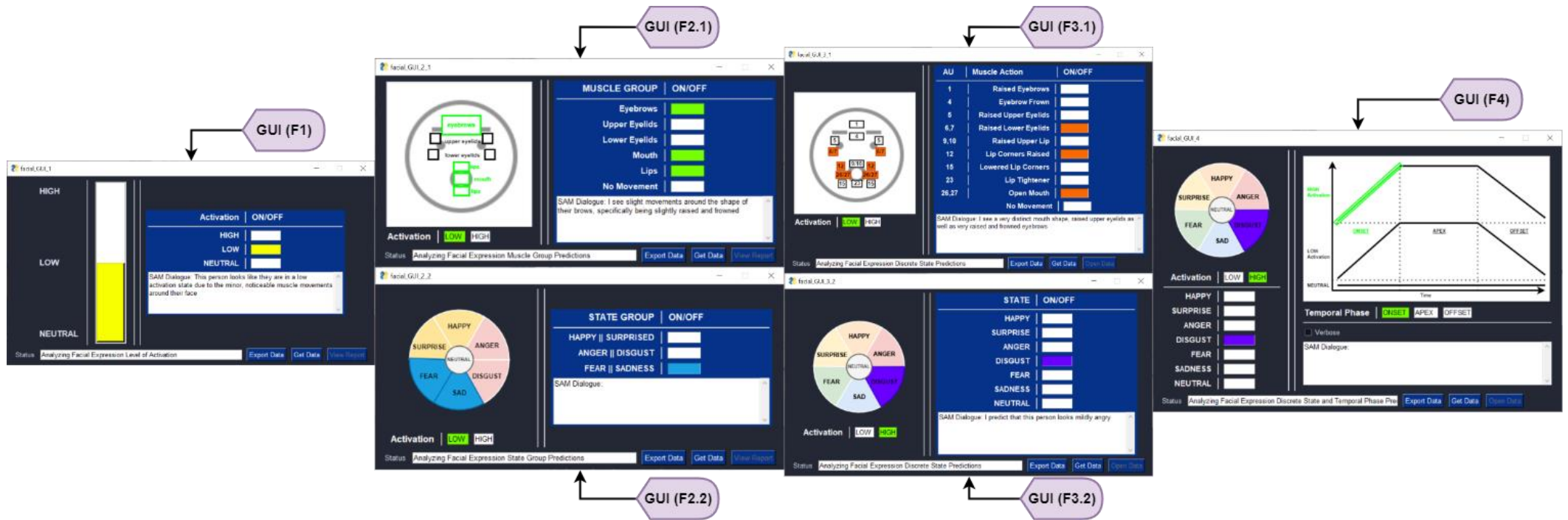


Figure 56: Collection of facial expression analysis window layouts of the four levels of the hierarchical classifier ensemble contained within SAM. As discussed previously, each level of classification allows users to view different information pertaining to a subject’s facial expression. Unlike when generating screenshots for Figure 55, these windows were recorded at different times.

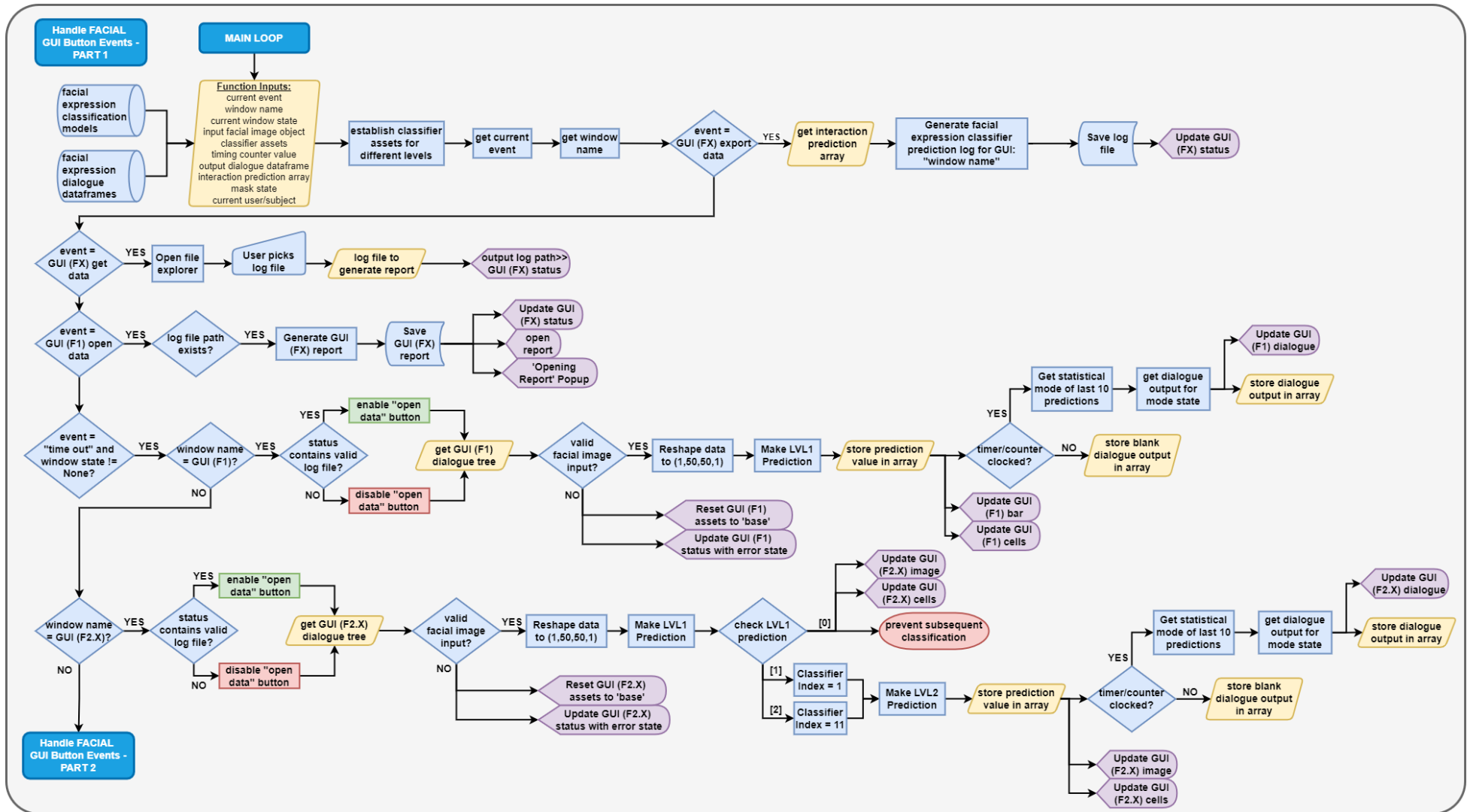


Figure 57: Facial GUI button handling function – Part 1. This flowchart shows how window events for facial GUI (F1, 2.1, 2.2) are handled in the main loop. The event handling function was split into Parts 1 and 2 due to the size of the flowchart. Figure 58 shows part 2.

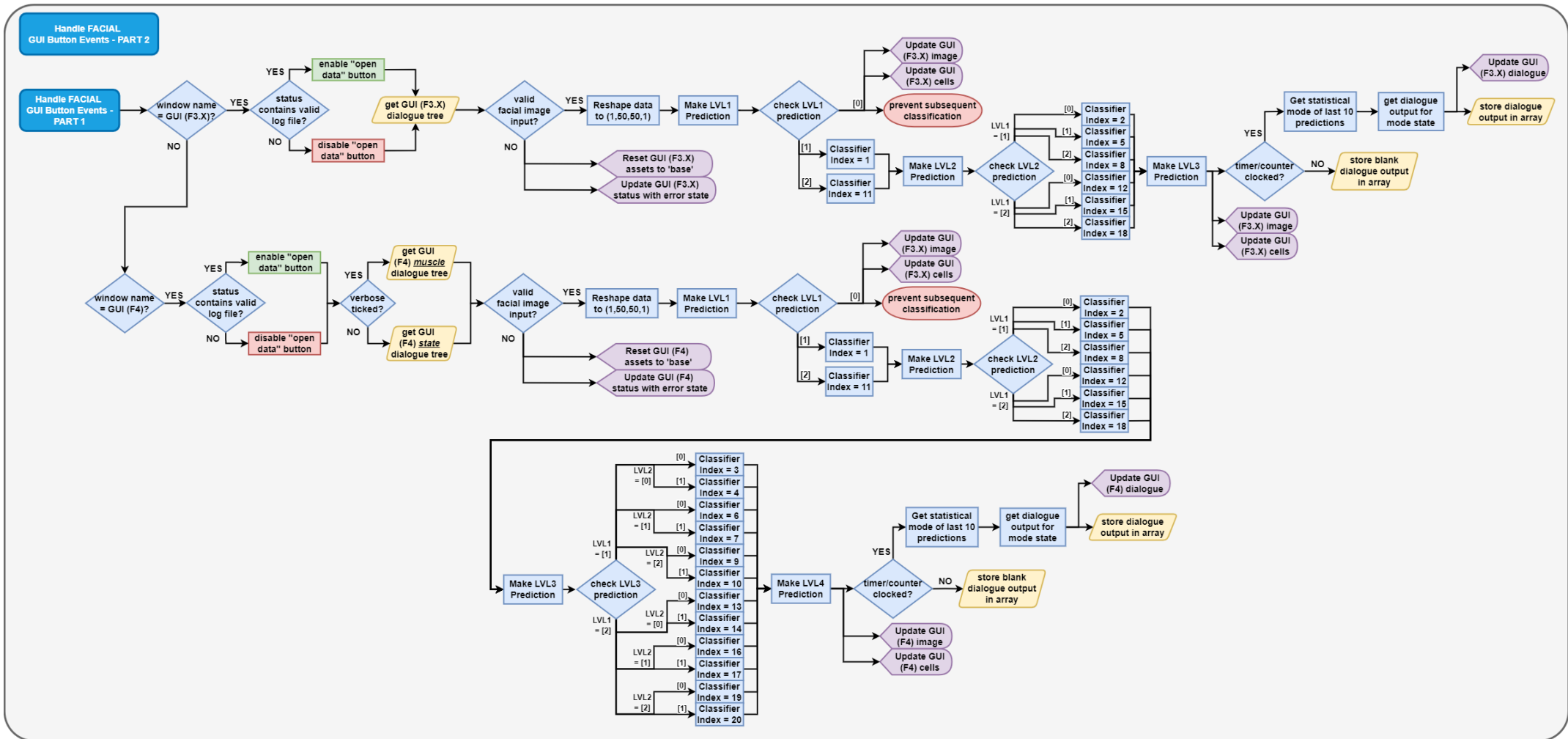


Figure 58: Facial GUI button handling function – Part 2. This flowchart shows how window events for facial GUI (F3.1, 3.2, 4) are handled in the main loop. This figure is an extension of the first part of the event handling function shown in Figure 57.

Analysing Figures 55 and 56 together, we see that examples of different states are shown, and we can see how the same logic that applies for Figure 55 would apply for other states. The change in image asset shown on the GUI window is dependent on the classifier prediction, with the integer predictions and corresponding images being reported in the package tree-diagram in Figure 45.

Analysing the flowcharts in Figures 57 and 58, we see how the facial expression analysis windows are integrated into SAM’s main loop. As discussed, all GUI windows operate on parallel threads and as such the event handling functions are non-blocking – meaning that all windows can be viewed in parallel without interrupting processes of other windows. Furthermore, we see how the classifiers listed in Table 14 are integrated into the system.

A novel invention that is added to SAM, is its ability to convert classifier predictions to human-understandable dialogue to relay the decision-making process and assist in improving the comprehensibility and accountability of the system. For all classifiable facial expressions, there is at least one unique dialogue output that SAM will use to relay its findings – using the theoretical foundations established in previous sections to justify the content of the dialogue. Table 17 reports some examples of the dialogue displayed to the users at different levels for facial expression analysis.

Classification Level	Prediction	Dialogue
Level 1	Neutral	This person looks like they are in a neutral state, due to the lack of noticeable movement around their face.
Level 2.1	LOW – Happy or Surprise	I see a slight change in mouth shape, and a combination of weak, lower, and upper eyelid movements.
Level 2.2	HIGH – Anger or Disgust	This person looks like they are either very angry, or very disgusted.
Level 3.1	LOW – Fear	I see a slightly open mouth, raised upper eyelids, and some indication of raised and frowned eyebrows.
Level 3.2	HIGH – Sad	I predict that this person looks very sad.
Level 4 (verbose = False)	LOW – Happy – Onset	This person looks like they are approaching a mildly happy state.
Level 4 (verbose = True)	HIGH – Anger – Apex	This person looks like they are very angry, by looking at their tightened lip, raised upper and lower eyelids and frowned brows.

Table 17: Examples of different dialogue outputs that are displayed (or spoken) to the user when a prediction is made, highlighting SAM’s ability to explain the reasoning behind why a classification is made at each level for facial expression classification

As we can see when comparing Level 2.1 and 2.2 or Level 3.1 and 3.2, we can see that SAM is capable of outputting verbose and succinct descriptions of states, with the user having the ability to choose what output is relevant for them. A full list of the facial expression classifier dialogue outputs for all levels of classification is shown in Appendix M.

All generated reports are output as PDFs automatically using custom report generation functions that exploit the continuous data that has been recorded during an interaction. The report generated and recorded in Appendix J highlights the intelligence of SAM to understand facial expressions. Specifically, in its ability to understand how specific muscles change throughout an interaction based on what predictions have been made. Given similar FACS muscle movements are present for different states, this information can be used to show how independent muscle movements change *based on the predicted state*. Analysing the generated reports in Appendices G-L, we can see that removing the real-time graphical data from the GUI allows users to view this information when necessary, thus removing the issue of having “too much information” on the GUI. Failure to do so may compromise the system’s comprehensibility and accountability.

5.2.4 Affective Speech Analysis Windows – GUI (SX)

The affective speech/vocal analysis windows – GUI (**SX**), are accessible via the corresponding buttons on the home screen as shown in Figures 46 and 50. The development of assets for both modalities were done in parallel as shown in Table 10, with the levels of classification being integrated into SAM iteratively from Level 1 → Level 4. Like the facial expression analysis windows discussed in the previous section, SAM was designed to allow users to view an optimal amount of information required by the user when assessing speech, again exploiting the benefits of multithreading. Figure 49 shows how events related to GUI (SX) windows are handled within SAM’s main loop.

SAM’s analysis of affective speech connects the back-end classifiers discussed in Section 5.1.2 with front-end interface elements and allows users to monitor changes in acoustic characteristics as well as the predicted state for incoming speech signals. Unlike facial expressions that are instantaneous however, affective speech signals need to be handled in “windows/packets”, with input speech needing to be appended to an “audio buffer” variable that is only handled when it’s the right size. The length of a speech sample when analysed is approximately 2 seconds given an audio data packet length of 4389 and a sample rate of 44kHz. This makes it consistent with the samples used to train the dataset prior to feature extraction. Likewise, the GUI assets related to vocal analysis update at a constant rate given that feature extraction and classification is only done when a packet is full sized.

SAM’s affective speech analysis are separated across four GUI windows, one for each level of classification. Unlike the facial expression analysis windows discussed in the previous sections, it was found that the GUI elements needed to describe changes in speech features did not need to be separated into six windows due to what was being classified at each level. To incorporate the ability to change how SAM responds via its dialogue output, ‘verbose’ tick-boxes have been added to GUI (S3, S4) windows as visualised in Figure 60. Users are capable of viewing GUI (SX) windows in parallel dependent on the level of classification that is being enquired about, which is shown in Figure 59. All affective speech/vocal analysis GUI windows have the same buttons that were described in the previous section i.e.: “Export Data”, “Get Data”, and “Open Data” buttons.

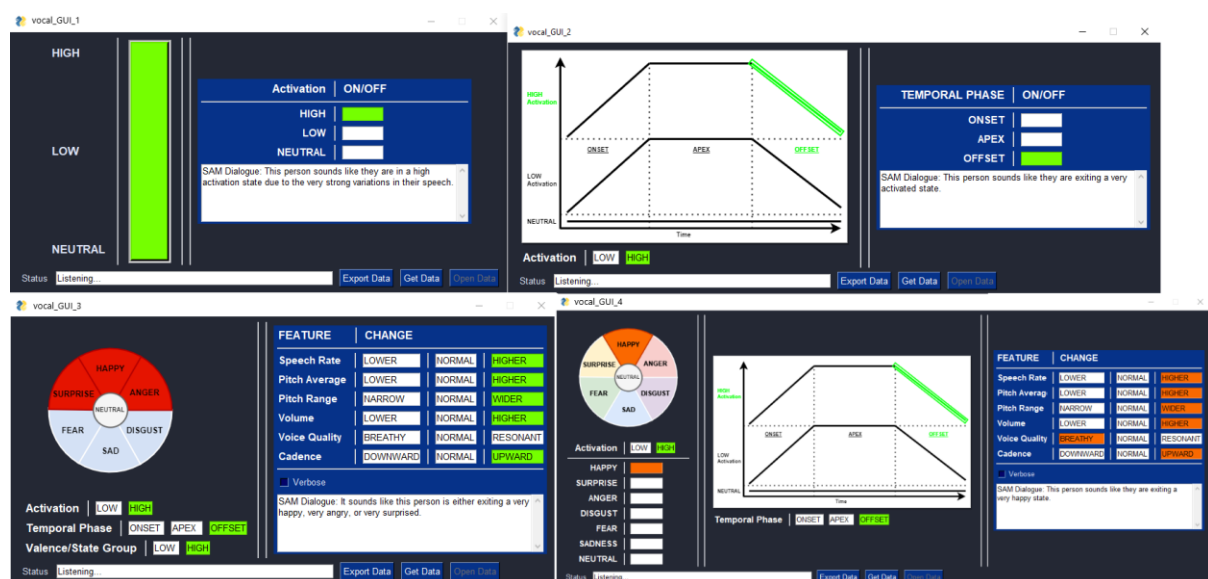


Figure 59: Affective speech analysis GUI windows in parallel operation where SAM predicted that the user sounded like they were offsetting from a high activation happy state.

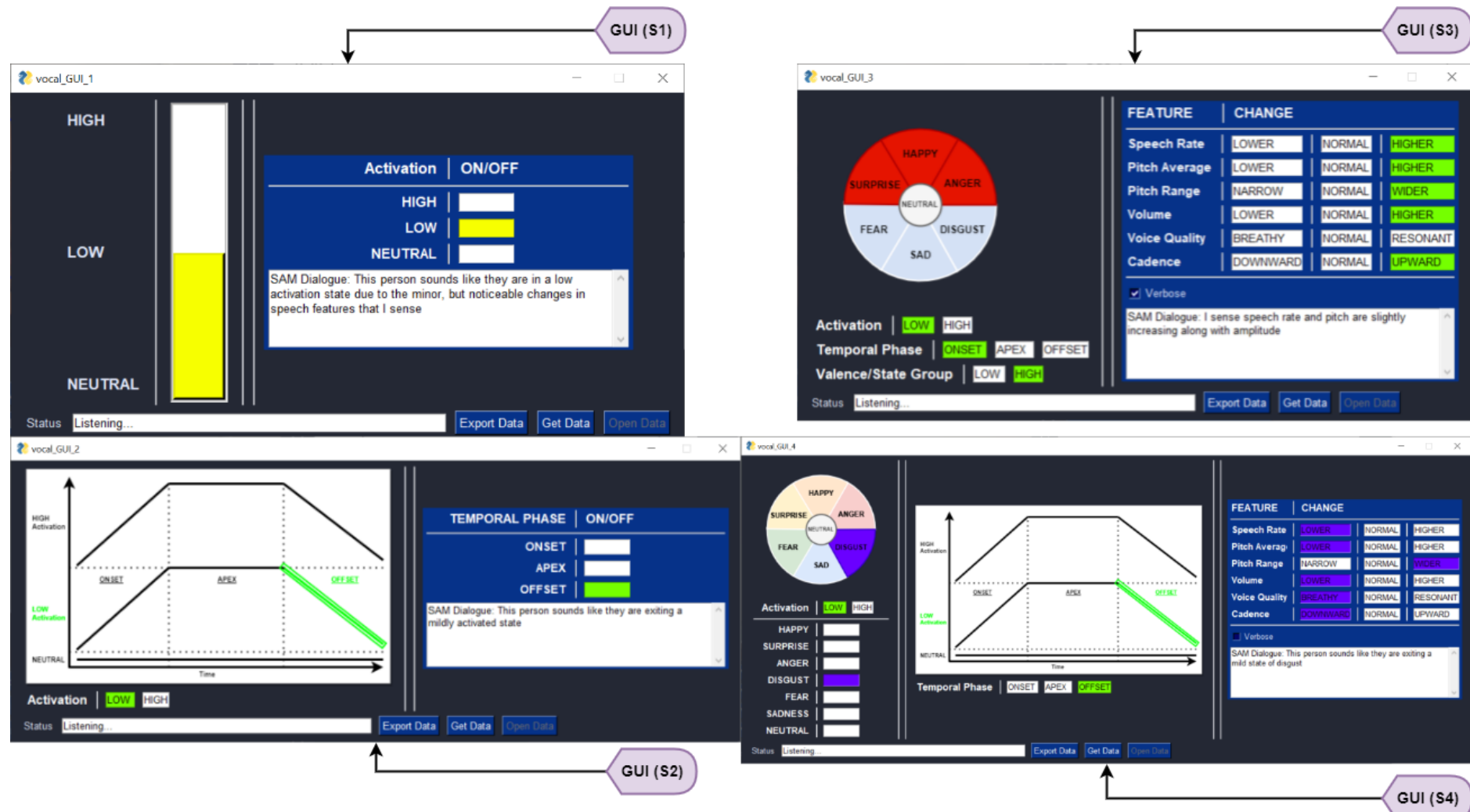


Figure 60: Collection of affective speech/vocal analysis GUI window layouts, showing the integration of the four levels of the hierarchical classifier ensemble contained within SAM. GUI (S3, S4) windows contain 'verbose' settings that affect what dialogue subset is deployed i.e., describing changes in speech characteristics, or describing changes in states.

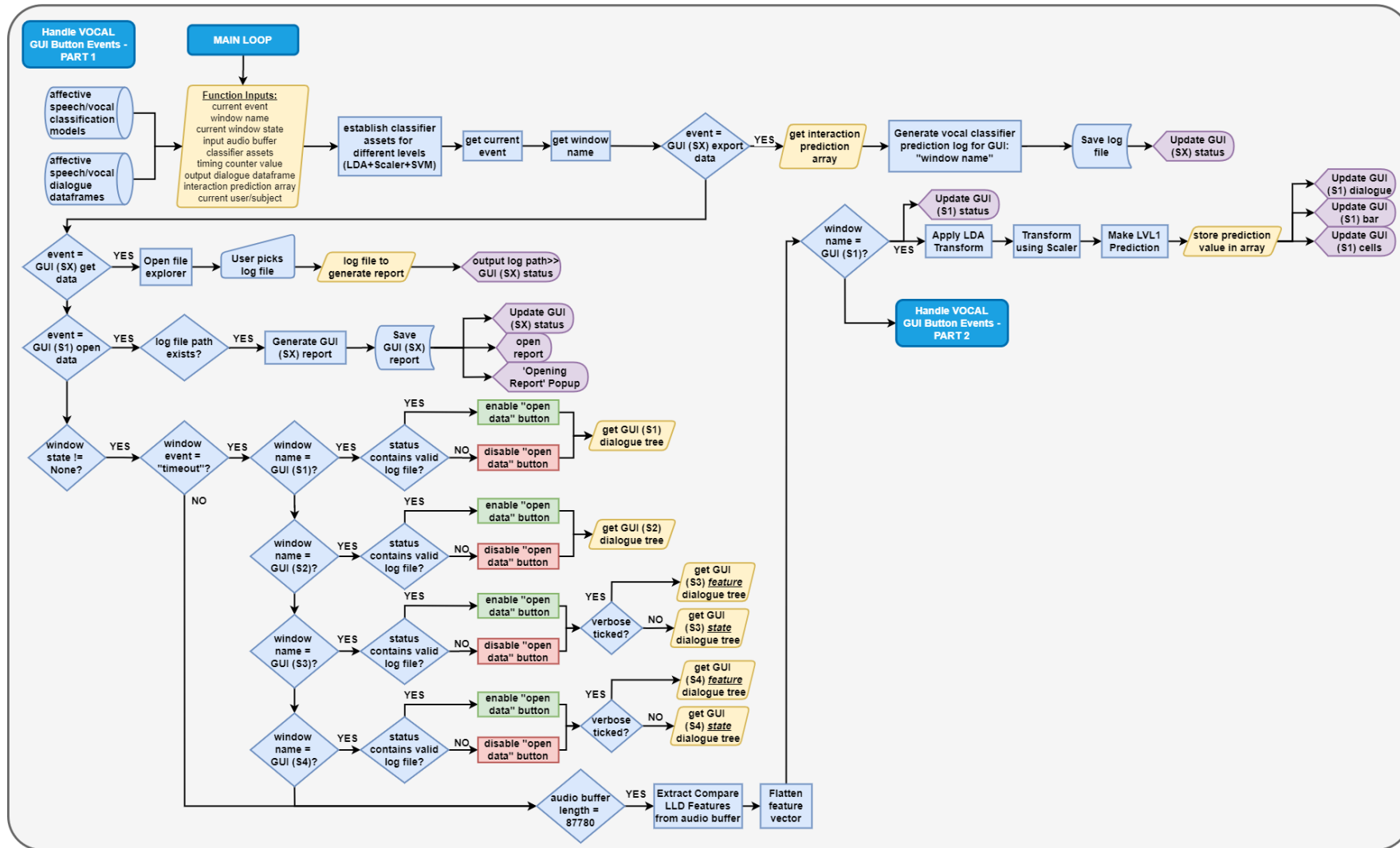


Figure 61: Affective Speech GUI button handling function – Part 1. This flowchart shows how window events for vocal GUIs are handled in the main loop. The event handling function was split into Parts 1 and 2 due to the size of the flowchart. Figure 62 shows part 2.

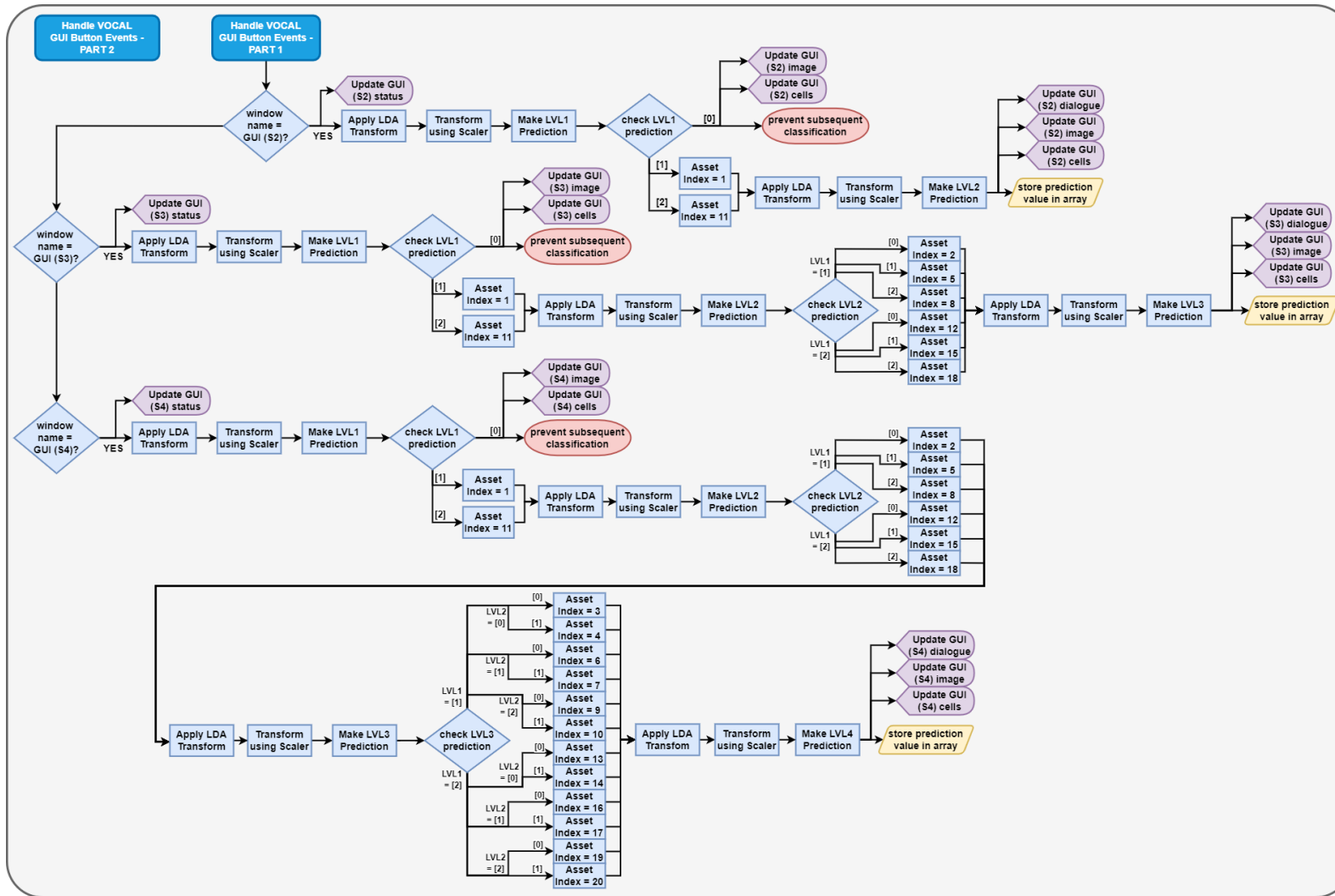


Figure 62: Affective speech/vocal GUI button handling function – Part 2. This flowchart shows how window events are handled in the main loop. This figure is an extension of the first part of the event handling function shown in Figure 61.

Looking at the flowcharts in Figures 61 and 62, it allows us to visualise how the affective speech analysis windows are integrated into the operation of SAM, and how events are handled in the main loop. Classification is deployed as long as a window is open to ensure that power and resources are not drawn by processes that are inactive. Consequently (and logically), power and resource consumption are proportional to the number of GUI windows open as all threads run in parallel, which allow for simultaneous viewing of prediction data.

Understanding the parallel operation of GUI windows means that users have the ability to open all ten analysis windows (across facial expression and speech classification) simultaneously if the system being used had the resources to handle the simultaneous classification. Section 5.3 discusses SAM’s performances as an embedded solution, highlighting the compromise between processor capabilities and system performance when deploying the software on the LattePanda 4G/64GB board.

Like with facial expression analysis, SAM is able to convert integer classifier predictions into human-readable assessments of a person’s affective speech state across the hierarchical classifier ensemble. This is done through the report generation ability (see Appendices N-Q), and the incorporation of dialogue outputs. Every predictable state has at least one dialogue output, with states at level 3 and 4 having two dialogue outputs dependent on the verbosity setting. The theoretical foundations surrounding speech characteristics discussed in Section 2.5 and Table 3 establish the link between SAM’s dialogue outputs at different levels. Table 18 highlights some examples of dialogue outputs that are displayed to users under different conditions.

Classification Level	Prediction	Dialogue
Level 1	HIGH	This person sounds like they are in a high activation state due to the very strong variations in their speech.
Level 2	HIGH – Apex	This person sounds like they are currently experiencing a very activated state.
Level 3 (verbose = True)	HIGH – Onset, HIGH Valence	I sense speech rate and pitch are increasing, along with amplitude, at a very high rate.
Level 3 (verbose = False)	LOW – Offset, LOW Valence	It sounds like this person is either exiting a mildly sad, mildly scared, or mildly disgusted state.
Level 4 (verbose = True)	HIGH – Apex, Happy	This person sounds like they are very happy, due to the change in speech rate and higher vocal pitch, showing an upward cadence, as well as their voice sounding very loud, blaring, and breathy.
Level 4 (verbose = False)	LOW – Apex, Fear	This person sounds like they are in a mild state of fear.

Table 18: Examples of different affective speech dialogue outputs that are displayed (or spoken) to the user when a prediction is made, highlighting SAM’s ability to explain reasoning behind why a classification is made at each level based on user requirements.

Through Table 18, we can see how the verbose settings are used to manipulate how SAM explains a state prediction and how users can choose what information they want to view. By implementing this functionality, it takes SAM capabilities beyond those of generic statistical classifiers that output a predicted state as either a label or integer. SAM uses theoretical understandings to justify to the user as to “why” a particular prediction is being made rather than simply positing the “what”.

Like the FACS action units used to quantify facial muscle movements, similar speech characteristics are also common in different states, which served as the theoretical foundation for clustering level 3 affective speech valences. Using this information, SAM is therefore capable of monitoring changes in

specific vocal characteristics based on its understanding of what characteristics are present in different expressions of states. An example of this functionality is shown in the report in Appendix P.

Through the dialogue outputs shown in Appendices M and R for facial expression analysis and affective speech analysis respectively, SAM proves itself as a more advanced assessment system equipped with logical, highly accurate classifiers, and the ability to explain its decision-making processes across four levels of classification, for two input modalities.

5.2.5 Settings Window – GUI (SET)

The purpose of this window is to give users the ability to login using username and password credentials as well as logging in the subject – which is the person in front of the peripheral devices (microphone and camera). The ‘user’ is the person using operational peripheral devices such as a mouse/touch pad/external display to operate SAM’s GUI to interpret its decision-making in real-time. In this case, SAM is treated as an intermediary agent between two parties. Note that the user and subject can also be the same person – as was the case when testing and prototyping the system and as such, SAM is treated as a secondary agent in a two-party interaction. The two situations described above are visualised in Figure 63.

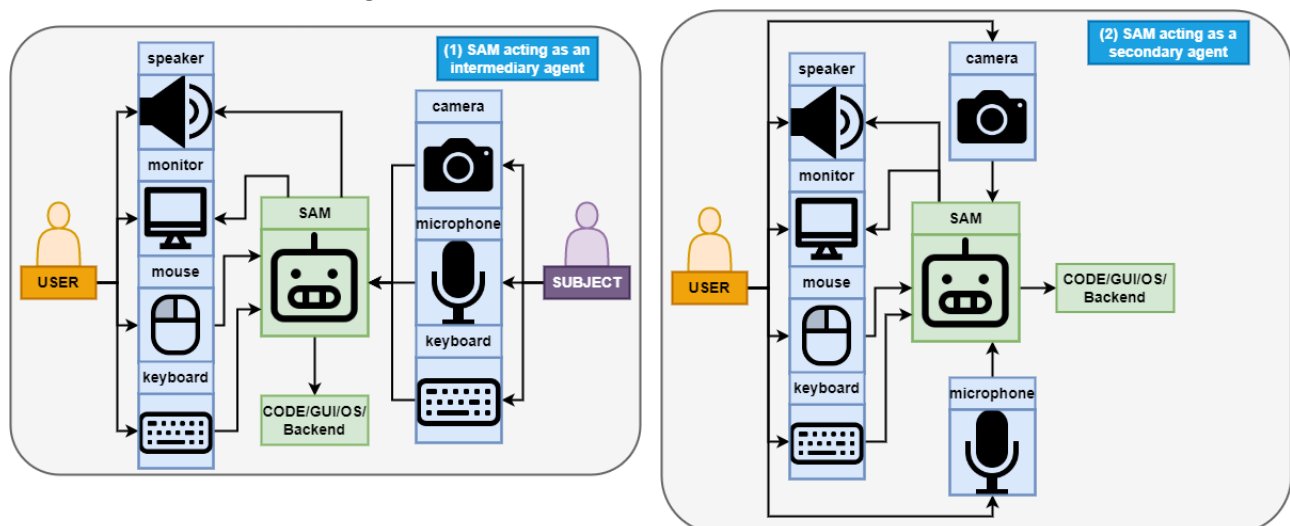


Figure 63: Dissecting the ‘user’ and ‘subject’ relationship with SAM wherein SAM can be deployed as an intermediary agent (where there are two human parties) or a secondary agent where there is only one human party present.

For the purpose of promoting accountability, SAM only recognises a user and subject if they have logged in using their credentials. Without doing so, SAM disables the ability to view classifier results as shown in Figure 54. Once SAM is aware of who is present in the interaction, it will enable the appropriate GUI buttons and functions so that it can then begin making predictions. Upon launching SAM’s GUI, users are to login their credentials using the settings GUI as well as getting the subject to input their credentials as well.

Within SAM’s working memory, it contains two databases containing information of users and subjects that it has interacted with in the past, containing generic information such as: (i) Identification number, (ii) username, (iii) password, (iv) first name, (v) last name, (vi) email, (vii) phone number, (viii)

data creation date, (ix) data creation time and (x) permissions. The settings GUI window and its corresponding sub-windows are visualised in Figure 64.

Analysing the GUI (SET) parent window and its children windows pictured around it in Figure 64, we see that in total there are seven new windows that can be spawned from the settings window, each being non-blocking as to not interrupt the main loop. The windows are numbered from 1-8, with green borders and numbered circles describing *user*-related windows and blue borders and numbered circles describing *subject*-related windows where:

1. Current user details
2. User login
3. Create new user
4. SAM user database
5. Subject login
6. Current subject details
7. Create new subject
8. SAM subject database

With each new window, comes event handling functions. The flowcharts in this section visualise the flow of information through the settings window and how user and subject information is parsed from the back-end to the front-end. Through previous sections we have seen how the user and subject information is deployed in the main loop of the system. This section explores the concepts of users and subjects in more detail in relation to how this information is created and stored in SAM's memory.

Through Figures 65 – 68, we see how data is parsed throughout SAM with respect to handling the settings window events and values. As well as benefitting system accountability, integrating the knowledge of subjects and users establishes a foundation in which SAM may be deployed as an applicable system across various independent sectors. Regarding accountability and trust, there would be an immediate sense of *distrust* when decision-makers (users) and those affected (subjects) were not named or identified. Along with having a knowledge of date-time information, the simple task of assigning ID's and having knowledge of who is involved in an interaction introduces a sense of transparency in a system as it puts names to the faces of decision makers and subjects.

As mentioned prior, complex, highly-sensitive applications like security-related, clinical, and legal applications require a high degree of accountability given the risk of misclassification in these sectors [14, 15]. Thus, intelligent systems like SAM must be equipped with functions like the ones described in this section that promote accountability.

Furthermore, when SAM interacts with users and subjects as a third party, having information like their names and understanding when they were added to the appropriate databases assists in making SAM a more intelligent system w.r.t to its communication and interaction capabilities. The next section is an in-depth exploration into SAM's interaction capabilities, showing a culmination of working memory, hierarchical classification, dialogue, AXAI, transparency and reasoning. Through this multi-faceted design, SAM's interaction processes separate it from other intelligent systems and AI agents.

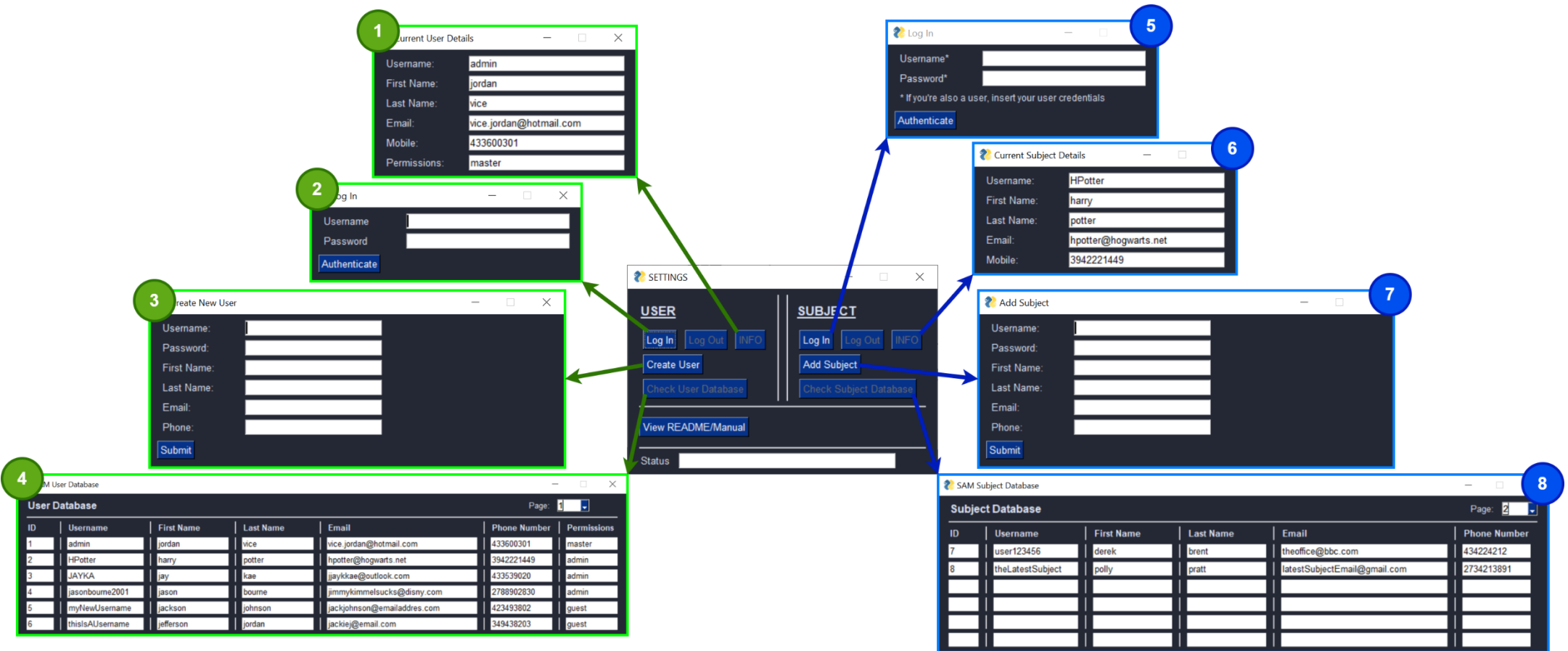


Figure 64: SAM’s GUI (SET) settings window (middle) and the corresponding child windows that are spawned by pressing buttons leading from the relevant arrows. The left side of the figure (green) highlights the four *user*-related and the right-hand side (blue) highlights the four *subject*-related windows. The window event handling functions are described through flowcharts in this section.

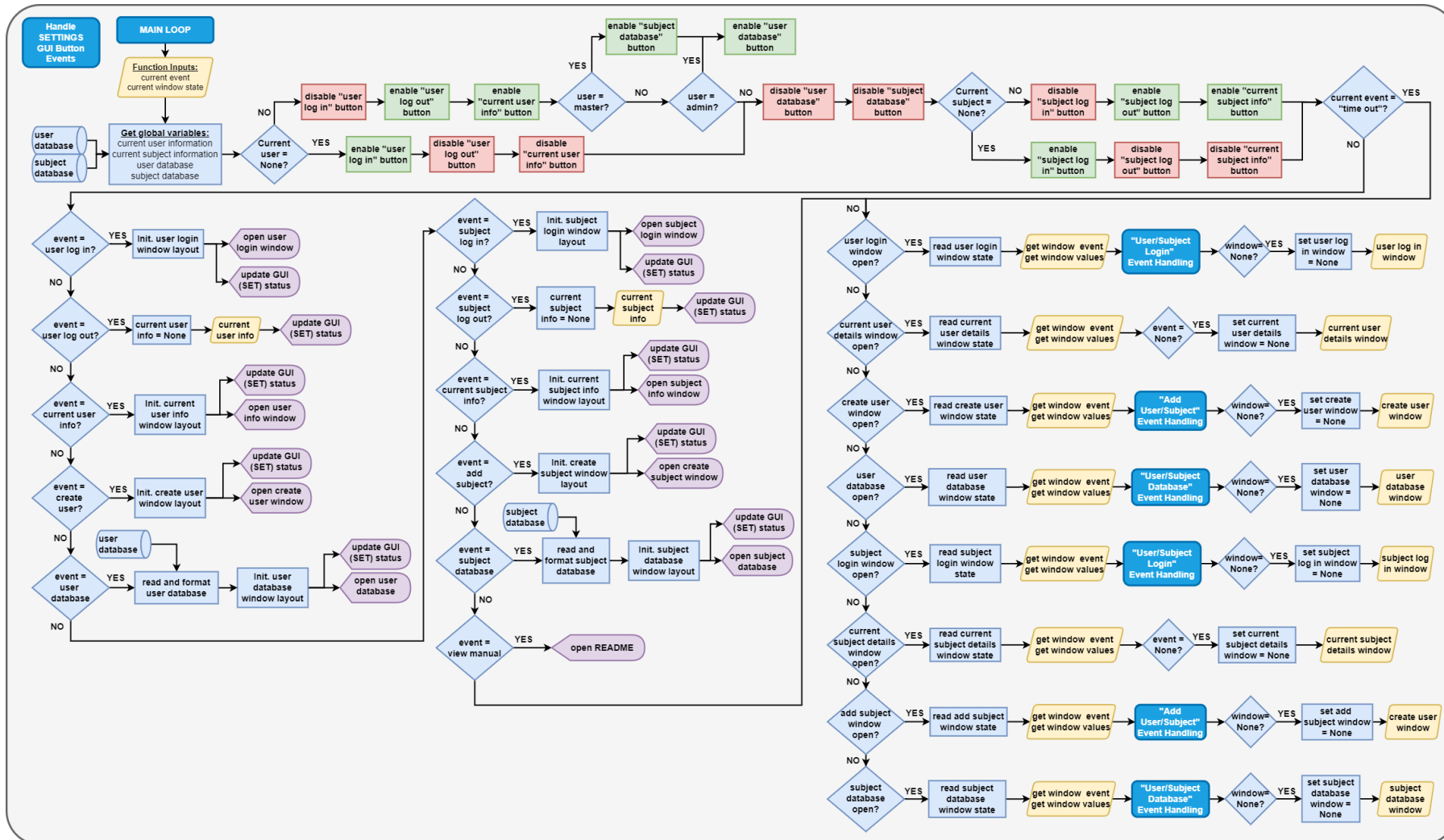


Figure 65: Setting window GUI (SET) event handling function block accessed by the main loop in Figure 49. This flowchart shows the relevance of user and subject data manipulation in this part of SAM’s code. Looking at the bottom right of the figure there are event handling function blocks for user/subject authentication, creation, and database visualisation, with these function blocks being visualised and discussed in this section.

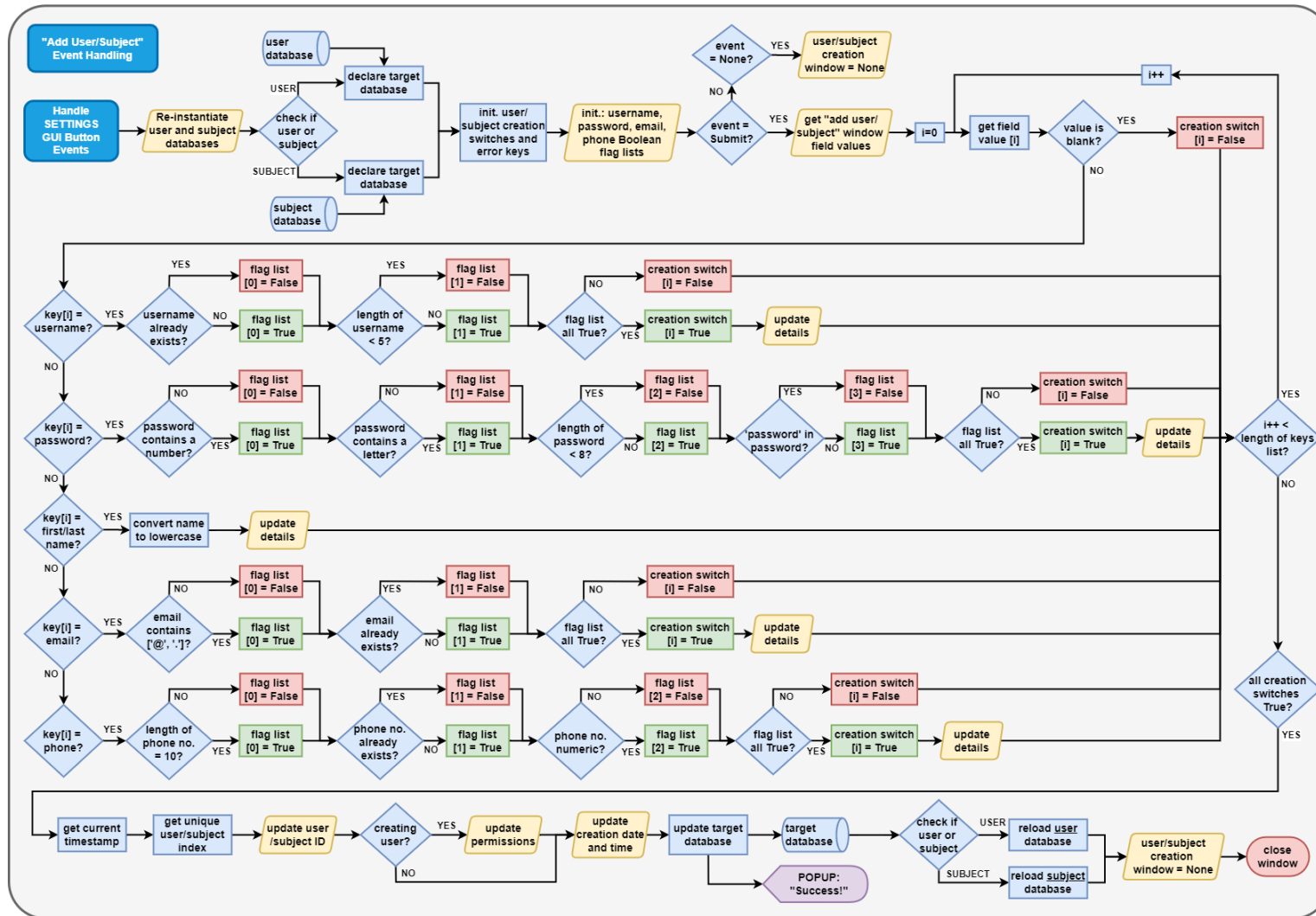


Figure 66: User and subject creation event handling function block accessed by the GUI (SET) event handling function block shown in Figure 65. This flowchart shows the error handling logic that are applied when attempting to add new users and subjects into SAM’s existing databases.

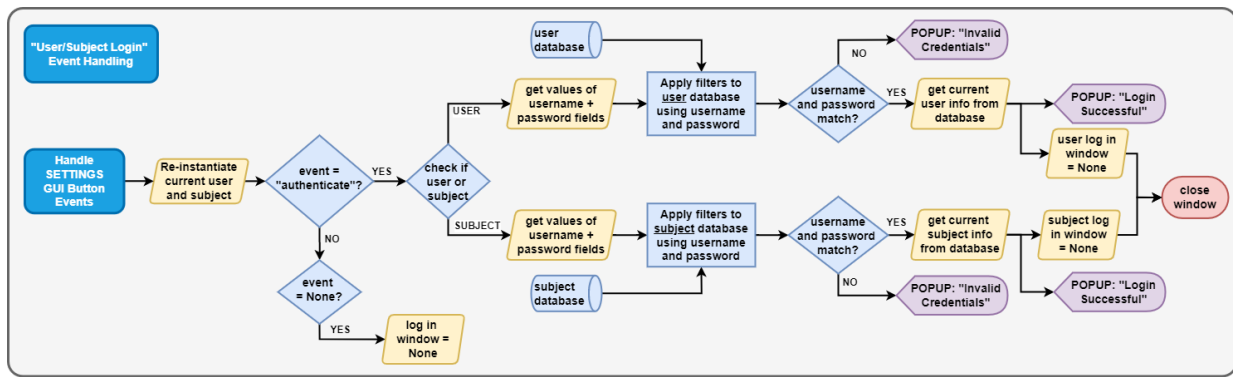


Figure 67: User and subject login window event handling function block – accessed by the GUI (SET) event handling function block in Figure 65. This function uses the information in SAM’s existing user and subject databases for authentication.

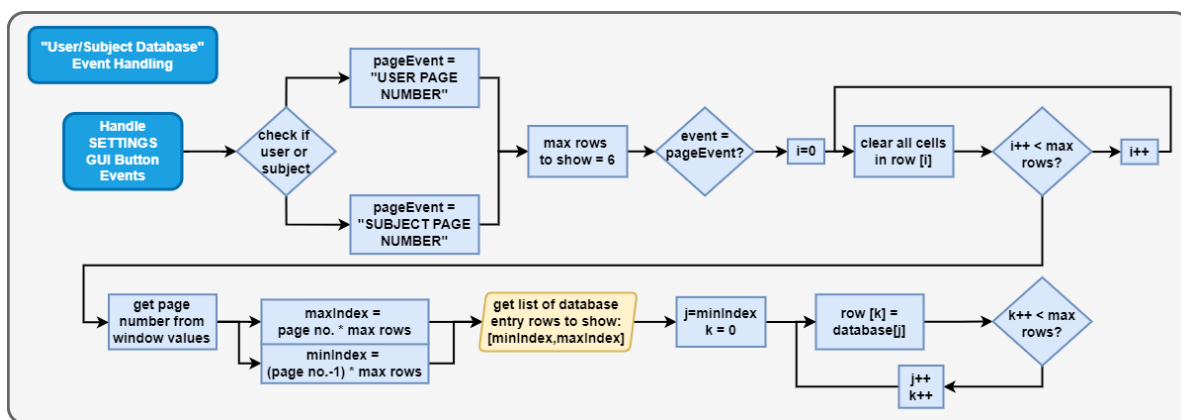


Figure 68: User and subject database window event handling function block – accessed by the GUI (SET) event handling function block in Figure 65. This demonstrates the ability to scroll through various pages of user and subject information that is contained in SAM’s existing databases.

5.2.6 SAM Interaction Window – GUI (INT)

In the case of clinical, therapeutic, and social robotics applications, designing AI and robotic systems to be approachable and user-friendly is an important consideration [137-139]. Particularly, we see that the application of robotics in therapy is viewed fondly in regard to assisting children and people with learning disabilities [137, 139]. Thus, SAM was designed with user-friendly considerations in mind to further promote it as an applicable solution. Firstly, while describing itself a “State Assessment Module”, a simple, single-syllable name like “SAM” introduces a sense of approachability and simplicity that would make it easy to interact with SAM has knowledge of who it interacts with by exploiting its user and subject databases. By doing so, it introduces a sense of companionship that would be beneficial not only for children, but for the elderly as well [139, 140]. Combined with ASA capabilities and some of the affective computing applications discussed in Chapter 2, we can begin to see the cross-industry areas in which a system like SAM can be applied.

SAM is equipped with an interaction interface, which allows users to communicate with SAM both verbally and non-verbally. SAM is equipped with 21 unique commands that can be triggered manually through the user interface window, or through values speech, with the latter utilising SAM’s speech-to-text

functions. Furthermore, on the concept of companionship, implementing 2-way communication postulates a sense of advanced intelligence, as it shows that SAM can understand the linguistic content of speech and respond in an intelligent way. Thus, the 21 commands allow SAM to interact with users in a way that promotes some form of social intelligence that display SAM's knowledge of for example:

- Its own current state,
- Its internal subject and user databases,
- Current time and date information,
- Its purpose,
- What has been said during an interaction,
- Emotions and affective state theory,
- What facial and speech expressions it can 'see' and 'hear' respectively.

The above summary of SAM's interaction capabilities is an example of "working memory", which describes brain systems that can access and manipulate stored information, be it in biological memory or digital memory banks, with the purpose of completing cognitive tasks [141]. Building upon the foundations of working memory, SAM uses the information stored in digital memory to respond to user commands using a text-to-speech engine that converts output strings in Tables 19-21 to synthesised, audio outputs. SAM's 'voice' uses 'Microsoft Mark', a text-to-speech voice/narrator that was introduced in Windows 10. The decision was made to change the voice output from the default to Mark as the former was very robotic and the latter is more novel and has more understanding of phonetic structures in speech. Figure 45 shows the location of 'modifiedMark' in the SAM package.

Through flowcharts, figures and tables in this section, SAM's interaction capabilities will be explained. Figure 69 shows a screenshot of SAM's GUI (INT) window. While looking relatively straight forward when compared to other windows that have been shown in previous sections in this chapter, the back-end functions and logic that support this window are more complex.

The window shown in Figure 69 has two dropdown boxes at the top, which allow users to change what level of classification SAM should be focusing on for both the facial and speech channels. The "Enable two-way communication" tick-box does precisely that. By ticking the box, the command dropdown box below and the "Send Command" button will be disabled, and users will need to refer to the list of commands to interact with SAM verbally (commands reported in Tables 19-21). The Status box on the bottom right assists users in understanding more information about the command that is being sent when using the dropdown box.

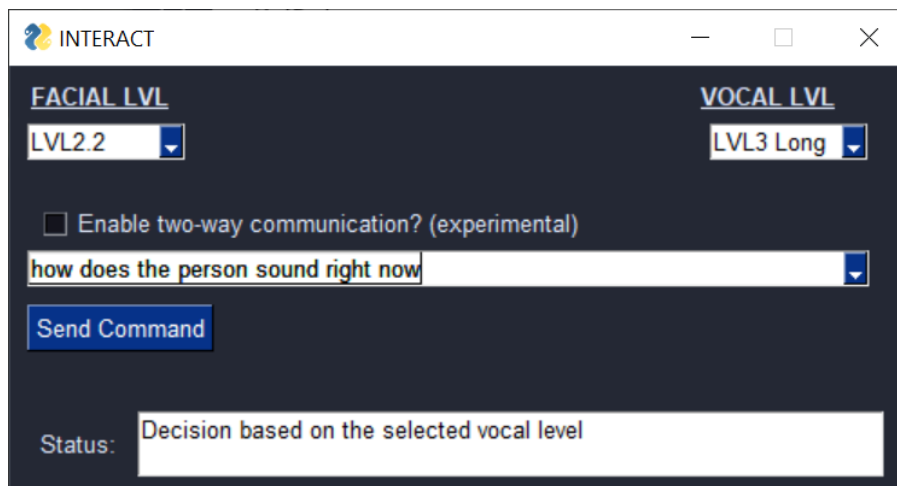


Figure 69: SAM Interaction GUI window layout, accessed via the “Interact with SAM” button on the top right of the home screen.

Prompt	Input	Reasoning	Base Output
0	what expression does the person look like they are showing right now	Decision based on the selected facial level	Right now,
1	what are you looking for	Understanding the level of classification	Based on my current facial expression classification level,
2	how does the person sound right now	Decision based on the selected vocal level	Right now,
3	what are you listening for	Understanding the level of classification	Based on my current speech classification level,
4	who are you analysing	Understanding who it 'thinks' it is looking at (current SUBJECT)	I believe that I am looking at
5	who is using you right now	Understanding who it 'thinks' is using it (current USER)	I believe that the person analysing my decisions is
6	can you repeat what this person said during this interaction	Knowledge of speech-to-text and what its heard	During this interaction I have heard,
7	tell me about yourself	SAM will explain its purpose, development and functionality	My name is SAM, which is short for the State Assessment Module. I was developed as a proof of concept for explainable, accountable, artificial intelligence, applied in the context of affective state assessment, and emotion recognition. Using my peripheral devices, I can sense facial expressions and speech, allowing me to recognise and understand emotions at different levels, on a continuous spectrum. Through my facial recognition abilities, I can also detect if a person is wearing a mask or not. By using this graphical user interface, you can monitor my decision-making processes in real-time, and request for me to generate reports of my interactions.
8	how long have you been in this interaction for	Understanding of human perceptions of time using system clock	I have been in this interaction for
9	how do you know who is analysing your decisions	Understanding of the USER database and the login process	When a user enters their credentials, I am inclined to believe that they are the person who is analysing my decisions
10	for how long have you known this user?	Using the datetime data in the user database - memory	I have known
11	for how long have you known the person you are looking at?	Using the datetime data in the subject database - memory	I have known
12	how are you able to analyse facial expressions	Explains the process of facial expression classification	I analyse facial expressions through my camera - which serves as my eyes. As you can see in the user interface, I draw a box around a face that I have detected, and then extract the features within the box. Depending on the chosen classifier level, I look for different features, and predict a state based on how my classification models were trained. The RAVDESS data-set was used to train the convolutional neural network models that allow me to make my predictions.
13	how are you able to detect if someone is wearing a mask	Explains the mask detection process	I analyse faces using my camera. The first step is to detect whether the person is wearing a mask, I do this by only looking at the bottom half of a person's face, and if I see that their face is covered, I am able to predict that the person is wearing a mask. My decision making processes when detecting a mask, come from training using the Masked-Face-Net data-set, and the RAVDESS data-set. I am able to identify a mask through a combination of linear discriminant analysis, and support vector machine models.

14	how are you able to analyse speech	Explains the process of speech expression classification	I analyse changes in speech through my microphone peripheral device - which serves as my ears. The changing waveform that you see on the user interface, is the direct raw speech data that I am receiving in real-time. I will make a prediction on how a person sounds, approximately every 2 seconds. In that time, I will wait and listen so that the length of what I am trying to predict, is the same as the data that was used to train my machine learning models. I then extract low level descriptor features, using the compare 2016 feature set, so that I can convert the raw speech data into a more understandable form.
15	explain your facial expression classification tree	Explains the facial decision tree	I can classify facial expressions at four levels. At the first level, I look at changes in the level of activation, from neutral, to low, to high. At the second level, I look at facial expression changes associated with three state groups, and their corresponding similarities in muscle movements. Moving onto level three, I look deeper into classifying discrete states, and their corresponding, facial action muscle movement changes. Finally, at level four, I look at changes in temporal phase. This tells whether a person is activating toward a discrete state. Experiencing an activation of that state - which is the apex. Or, returning back to a more neutral state, which is called, offsetting. All of my prediction information, is available to you, through the relevant user interface windows.
16	explain your speech classification tree	Explains the speech decision tree	I listen to speech on four levels. At the first level, I listen for changes in the level of activation, from neutral, to low, to high. Then, on level 2, I listen for changes in temporal phase, which tells me if a person is activating toward some state. Experiencing the apex of some state. Or offsetting from a state, toward a more neutral sounding speech. At level three, I listen for changes in valence, which can be high, or low, and correspond to two groups of states. These valence groups, are based on similarities in acoustic features. Finally, at the fourth level, once I understand the valence of the person, I listen closely, to changes that would correspond to a particular, more discrete state. I can then understand the difference in acoustic features, within that state group.
17	can I shut you down	Explains how users can shut it down	Yes, you can shut me down by closing the home screen window, and removing me from my power source. Would you like to shut me down now?
18	how can I check if you are working properly	Explains any issues that could arise during operation	First make sure that all my peripheral devices are plugged in, so that I can see, and hear. If you want me to understand what's being said in this conversation, make sure I am connected to the internet. Make sure that the lighting conditions are adequate, and consistent. Also, make sure that you have told me who you are, through the log-in screen, as well as made me aware of who I am looking at.
19	do you have emotions of your own	Explains that it can only recognise emotions, cannot synthesize (in its current form)	No, in my current state, I do not have any emotions of my own, and I cannot synthesize artificial emotions. I can only recognise someone else's emotional state over time, based on how I classify their facial expressions, and speech.
20	can you respond based on how a person is feeling	Explains that it does not have the behavioural and psychological intelligence to respond based on a persons emotional state	I don't possess the behavioural, and psychological intelligence, to respond to a person based on their emotional state. In theory, I could use a historical understanding of an interaction, but human emotions are so complex, that I would need to have an understanding of internal, and external contextual information, that is not available to me.

Table 19: List of SAM’s interaction commands as well as the reasoning behind the inclusion of the command and the base string output to the user. SAM’s text-to-speech engine uses commas and periods for pauses in speech outputs, making responses more realistic and less robotic.

Key	Output
LVL1	which is level 1. I am looking for changes in facial expression activation levels, to show me if the person is experiencing a neutral, low, or highly activated state.
LVL2.1	which is level 2 point 1. I am looking for macro-level changes in facial expression muscle groups, that are associated with different groups of emotions, or states.
LVL2.2	which is level 2 point 2. I am looking for changes in facial expressions, that relate to one of three groups of states. pairing happiness and surprise. anger and disgust. and sadness and fear.
LVL3.1	which is level 3 point 1. I am looking for changes across all facial muscles, and using Ekman's facial action coding system, this allows me to discern a particular emotion.
LVL3.2	which is level 3 point 2. I am looking to predict one, of two emotions. Which is made easier for me, since I have already deduced the state group.
LVL4 Short	which is non-verbose level 4. I am looking for changes in temporal phase, so that I can tell if a person is approaching a state, or emotion. Experiencing the apex of that state, or deactivating from it.
LVL4 Long	which is verbose level 4, I am looking for changes in muscle movements that, correspond to the activation, or deactivation of a particular state, or emotion. As well as the activation level.

Table 20: Prompt Number 1 – extended dialogue outputs. These outputs are in response to the question “what are you looking for?”, allowing SAM to identify what level of classification was chosen by the user for facial expression assessment, responding in the appropriate manner.

Key	Output
LVL1	which is level 1. I am listening for changes in speech activation, that show me if the person is experiencing a neutral, low, or highly activated state.
LVL2	which is level 2. I am listening for changes in temporal phase, so that I can tell if a person is approaching a state, or emotion. experiencing the apex of that state. Or deactivating from it.
LVL3 Short	which is non-verbose level 3. I am listening for changes in speech that could tell me if a person is experiencing a high-valence, or a low-valence emotion or state.
LVL3 Long	which is verbose level 3 I am listening for changes in acoustic feature groups, that translate to high, and low valence emotions and states.
LVL4 Short	which is non-verbose level 4. I am listening for changes in speech, that would tell me what discrete emotion is being expressed, as well as its temporal phase, and level activation.
LVL4 Long	which is verbose level 4. I am listening for changes in a wider range of acoustic features, that translate to specific emotions, at different activation levels, and temporal phases.

Table 21: Prompt Number 3 – extended dialogue outputs. The above outputs are in response to “what are you listening for?”, where SAM identifies the level of classification chosen by the user for affective speech assessment, responding accordingly.

Analysing Table 19 we see that prompt numbers: {0, 1, 2, 3} are related to SAM's hierarchical classifier ensemble. The dialogue outputs attributed to {0, 2} are dependent on the prediction made at the instant that the command was sent and the level of classification that the user has chosen using the two dropdown boxes at the top of Figure 69. These dialogue outputs are as they were reported in Appendices M and R. Through prompt {6}, we see that SAM is capable of understanding the content of what was said during the interaction, having the ability to repeat this information back to the user. Locally storing this information could have the potential for much more advanced tasks using target-word detection or natural language processing functions for example.

Prompts {4, 5, 10, 11} show SAM's understanding and manipulation of its user and subject databases. Once a user and subject has logged in, SAM trusts that the input information is correct and that the parties involved in the interaction are who they have logged in as (which SAM explains through prompt {9}). Regarding prompts {10, 11}, along with {8}, we see how SAM understands human perceptions of time. By referring to its internal clock and historical date-time information, SAM is able to output the number of months, days, hours, minutes that it knows a person as well as for how long it has been in operation for.

Along with the responses to prompts {1, 3} shown in Tables 20 and 21, through prompts {7, 12, 13, 14, 15, 16}, we see SAM's ability to explain its decision-making and reasoning processes for everything that it does. This includes being able to explain and speak about: (i) its facial expression classification processes, (ii) its overall purpose, (iii) how it classifies speech signals, (iv) how it detects masks and visual occlusions. System transparency is improved by giving SAM the opportunity to explain this information to users, and by doing so, could make SAM a more trustworthy and acceptable system in domains where transparency is an important consideration.

Prompts {17, 18} assist in the debugging process as discussed in [4, 70]. By giving users the ability to verbally shut down SAM, it retains the control that the user has, while also giving users the ability to still switch off the system if mouse and keyboard peripheral devices have been disconnected. The response to prompt {18} gives SAM the opportunity to assist users in the debugging process, allowing them to work through the steps that SAM provides to assist in troubleshooting. This command has the potential to be expanded on such that SAM could use its knowledge of its various statuses to point the user in the direction of what could be an issue.

Finally, the responses to prompts {19, 20} give SAM the ability to respond to functionalities that it currently does not possess i.e.: the ability to synthesise artificial emotions of its own, and the ability to augment its responses based on a person's current state. The ability to synthesise artificial emotions would highlight a potential "next step" for SAM, as a survey conducted in [141] shows that there are various ways of incorporating emotional speech synthesis including articulatory, data-based and hybrid synthesis approaches [142].

If SAM had the ability to synthesise speech, one could argue that this can be used to augment its text-to-speech responses based on a person's current state. However, as SAM states in its response (in Table 20) and discussed in literature [80-83] and previous chapters of this work, human experiences and emotions are complex, and are the result of various dependent and independent, internal, and external events that would require far more research and development. For example, complex social constructs like break-ups, divorces, death, and humour are experienced differently and could be

dependent on: which party is affected, relationships, socio-economic status, anthropology, prior experiences. These constraints and variables highlight the difficulty in designing a machine to synthesise its emotions to respond in a ‘socially acceptable’ way.

The flowcharts visualised in Figures 70-72 describe the logical and computational structure of how the commands discussed above are implemented in real-time, and how SAM manipulates various subfunctions and its working memory to complete various cognitive tasks. Analysing Figure 70, we see how the interaction window events are handled from the home screen and how the 2-way communication tick-box governs the ability to manually put in commands using the dropdown menu.

The 2-way communication works through a similarity detection algorithm that calculates the similarity of two strings using the “Sequence Matcher” function that is part of Python’s difflib library. The sequence matcher algorithm is based on the Gestalt pattern matching algorithm derived in the late 1980’s and can be used for detecting the similarity in two strings [143]. The algorithm (and Python’s difflib implementation) defines a ratio ‘ D_r ’ between two strings ‘ S_1 ’ and ‘ S_2 ’ and can be described through the following governing equation:

$$D_r = \frac{2 \cdot |\{S_1\} \cap \{S_2\}|}{|S_1| + |S_2|} \quad (37)$$

where D_r is directly proportional to the similarity between two strings. Through experimentation it was found that because some commands are similar in their structure, keywords needed to be added such that no two cases are unique, thus introducing a second instance of logic checking – which ensures that keywords in a command are spoken as well as similarity being greater than 80%.

Through the initial event handling function we see that there are two additional larger function calls, these being “Handle Interaction Commands” and “Shutdown Sequence”. These function blocks are detailed in Figures 71 and 72, with the former highlighting where most of the logic is contained. The handling of user commands is done through a large switch-case statement, whether commands are verbal or manual, the index of the command is used for gathering the response, with these responses being reported in Tables 19-21.

Through Figure 71, we see that this function block serves as SAM’s central communication node/hub, which communicates with various other subsystems to explain its decisions and decision-making and reasoning processes, while also highlighting a practical implementation of ‘working memory’ theory [141] in robotics and AI. Finally, we see that Figure 71 shows a call to the “temporal length calculation” function block which is displayed in Figure 72. The temporal length calculation function highlights SAM’s knowledge of human perceptions of time using current and historical datetime data and simple Boolean logic and arithmetic to report and communicate this information to users.

Throughout the subsections of Sections 5.1/2, SAM’s various GUI functions and back- and front-end software components have been discussed. Through various flowcharts and screenshots of SAM’s GUI in operation, we see intricacies of the software design process and the attention to AXAI elements like comprehensibility, system accountability and predictive accuracy. However, confining SAM to being a static, software solution limits its true potential, as the ability to enhance existing robotic systems and surveillance infrastructures through a modular device is a key contribution that is proposed through this work. The following section discusses SAM as a fully-integrated, embedded system in more detail.

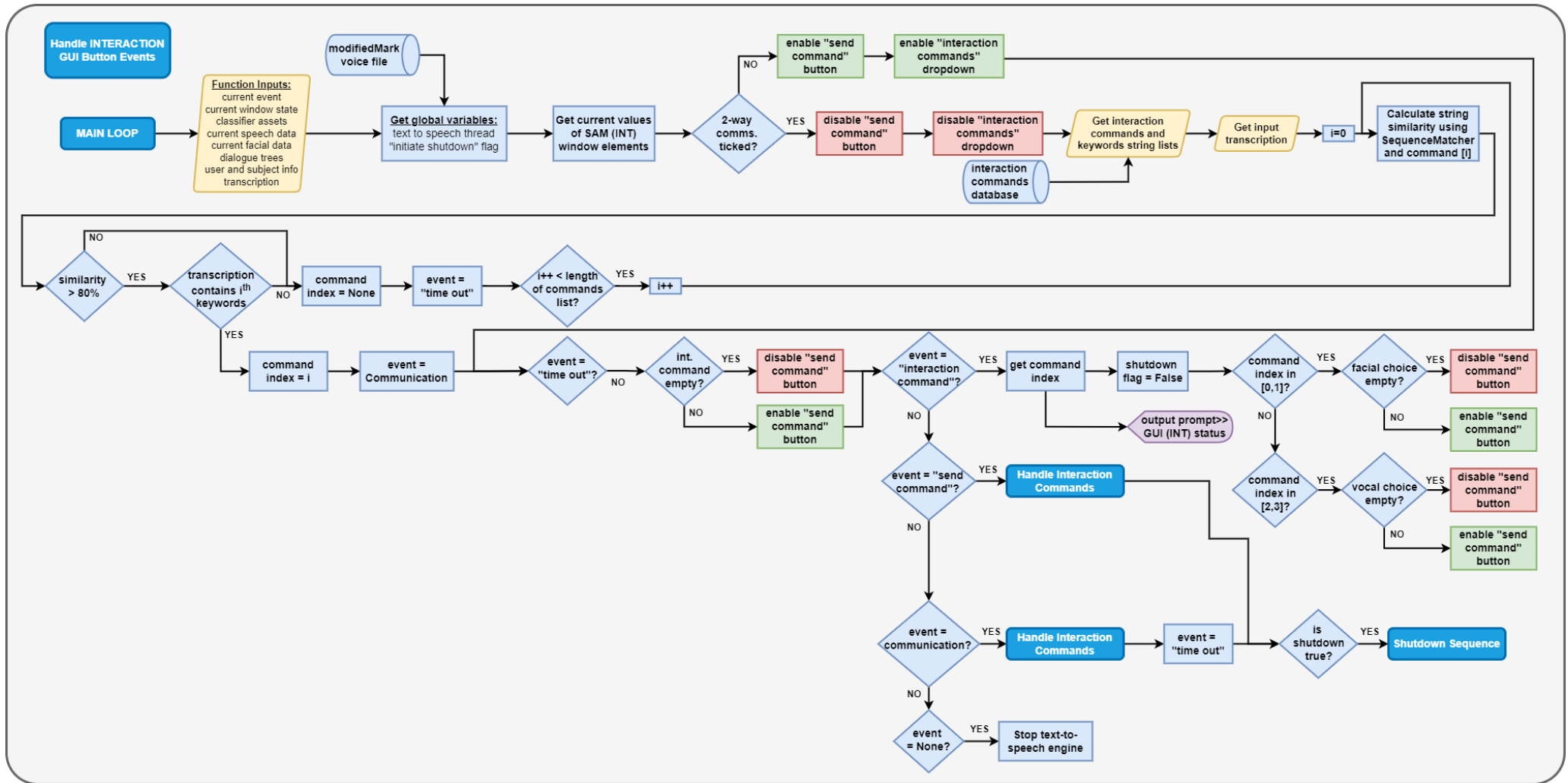


Figure 70: SAM Interaction window i.e., GUI (INT) event handling flowchart. This function block is called from the main loop and handles events associated with user interactions with SAM including two-way communication. On the bottom right, we see further function calls, with these functions being detailed in Figures 71 and 72.

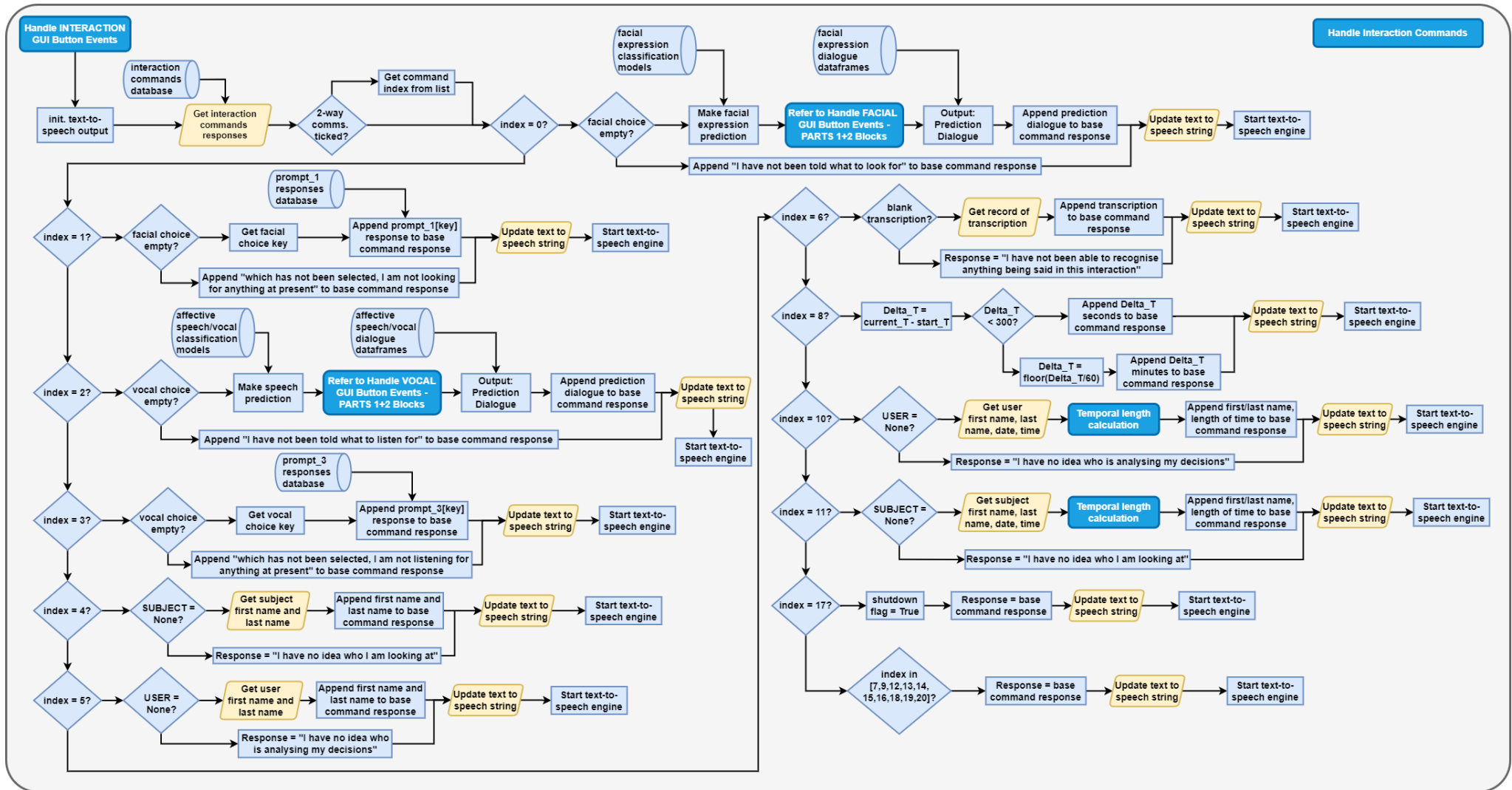


Figure 71: This function block is called from the GUI (INT) event handling block visualised in Figure 70. This visualises the large ‘switch-case’ logic used for handling SAM’s communication text through commands and responses. The “Temporal length calculation” call is visualised in Figure 72.

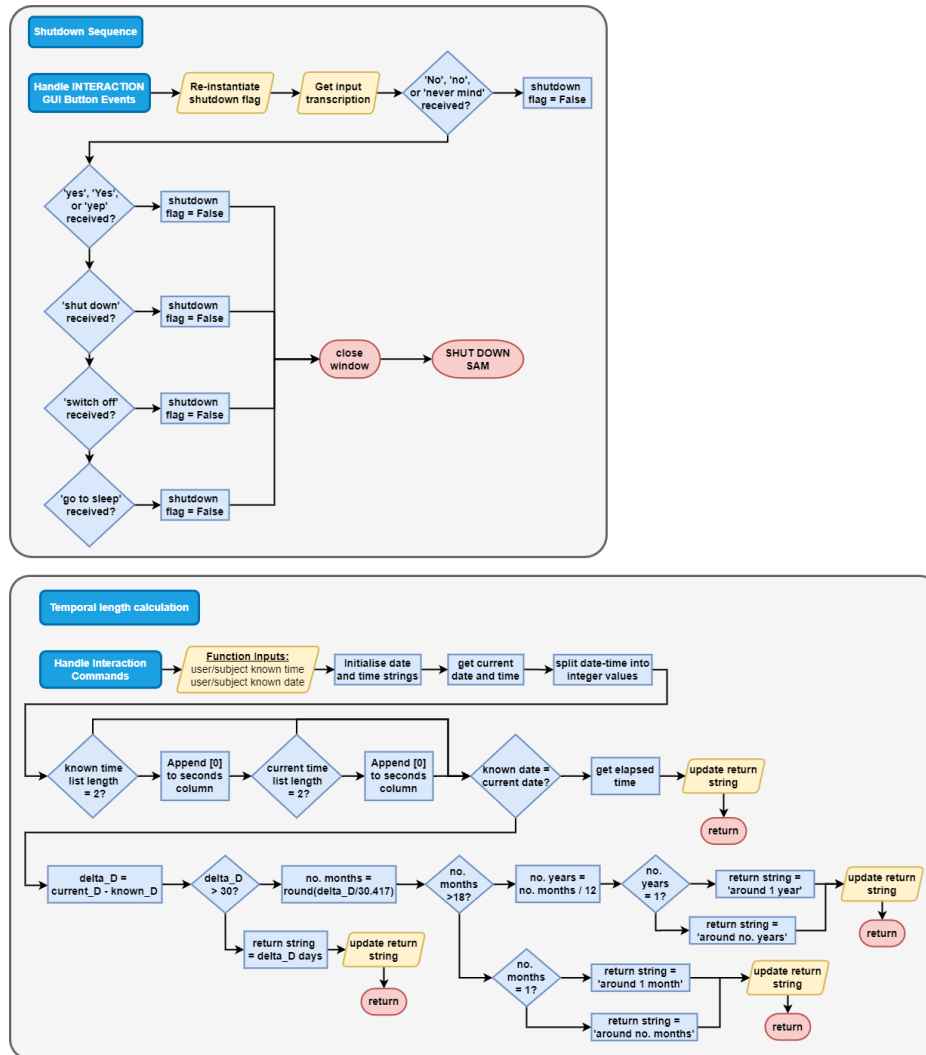


Figure 72: (TOP): Shutdown Sequence function block, which shows how user inputs can be used to switch off SAM using voice commands. (BOTTOM): The temporal length calculation function block which shows how SAM utilises its knowledge of current and historical datetime information.

5.3 Integrating SAM as a Modular, Embedded System

As discussed in Chapter 1 and reported throughout this work, SAM was designed to be a modular system, capable of being equipped into existing robotic and sensory infrastructures. Figure 63 highlighted some of the peripheral devices needed for SAM's operation. While this visualised how SAM could be used as a 2nd or 3rd party in an interaction, the previous sections looked primarily at the software that was embedded in SAM.

This section aims to discuss SAM as a real-time, integrated embedded system, considering the hardware components and peripheral devices that are necessary for sensory, operational, and monitory functions. Furthermore, this section will establish a bill of software and hardware materials that would ensure that SAM's design is replicable. Finally, this section will establish some of the achievements made and challenges faced throughout the development of SAM from Version 1 to its deployment as a functioning modular device.

With the rise of robotics and autonomous systems over the last century and the increase in the volume of solutions that exist commercially [144-146], it was evident that SAM would be better suited as a solution to augment existing robotic systems, rather than presenting itself as a kinematic robot. The humanoid robots discussed in [145] highlight the range of applications and sectors in which these systems have been applied including: defence, commercial and entertainment sectors. Social robotics and socially assistive robotic solutions [144, 146] also evidence a large corpus of unique robots that are designed to improve quality of life and for use in educational and clinical scenarios.

The robotic systems in [144-146] describe a small portion of robotic systems that have been reported in literature and used throughout daily life. Ultimately, the relatively high saturation of "robots" means that there is less novelty in designing a humanoid robot with ASA capabilities for example. This highlights the novelty and importance of SAM. Designing a system that can be integrated into existing robotic infrastructures means that SAM is not limited in its application and mobility.

After the software development process (finishing on version 5), the SAM software was ported onto a LattePanda V1.0 4G/64GB, a small form-factor, low-cost single board computer [147]. This hardware was previously used for implementing the ASAM as an embedded system design as reported in [1], which also discusses and compares six different solutions including: (i) Intel NCS2, (ii) LattePanda V1.0 4G/64GB, (iii) LeMaker Hi-Key 970, (iv) BeagleBoard X-15, (v) Coral Dev Board, (vi) NVIDIA Jetson Nano.

Initially, the NVIDIA Jetson Nano [140] was going to be experimented with. However, the "chip shortage" that occurred during the COVID-19 pandemic [149, 150] resulted in mass production line delays and adversely affected various industries that rely on semiconductors and integrated circuits. This shortage meant that lead times and costs for boards like the NVIDIA Jetson Nano were very high throughout a large period of the development cycle of SAM. Thus, it was decided that SAM be prototyped on the LattePanda board that was available for use.

When testing SAM as an embedded system, it was found that while it was objectively slower than when using a desktop or personal computer, it was still functioning well enough in real-time to a point where the computational power was not a heavy burden on performance. Figures 73-75 show the hardware implementation of SAM, visualising the development board, the external hardware, and an example of how SAM can be integrated into an existing system using a low-cost Zero-X Edge drone and attaching SAM to it using Go-Pro mounting clip accessories.

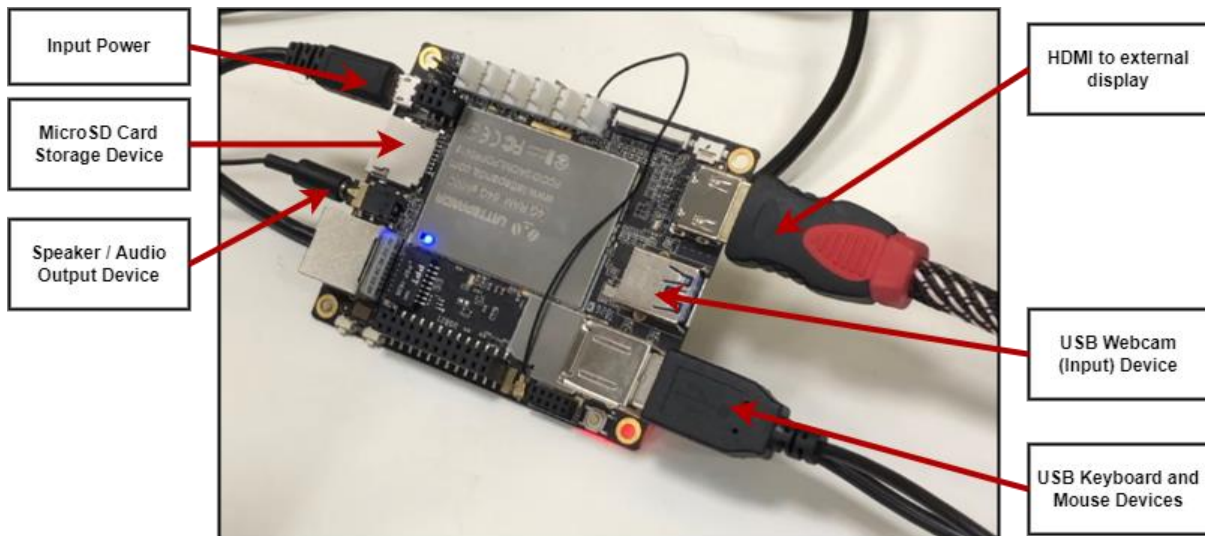


Figure 73: Exposed LattePanda 4G/64GB development board with peripheral devices connected. Note that the peripheral devices connected are similar to those introduced in Figure 63. The devices connected in the figure are necessary for sensory, operational, and monitory functions.

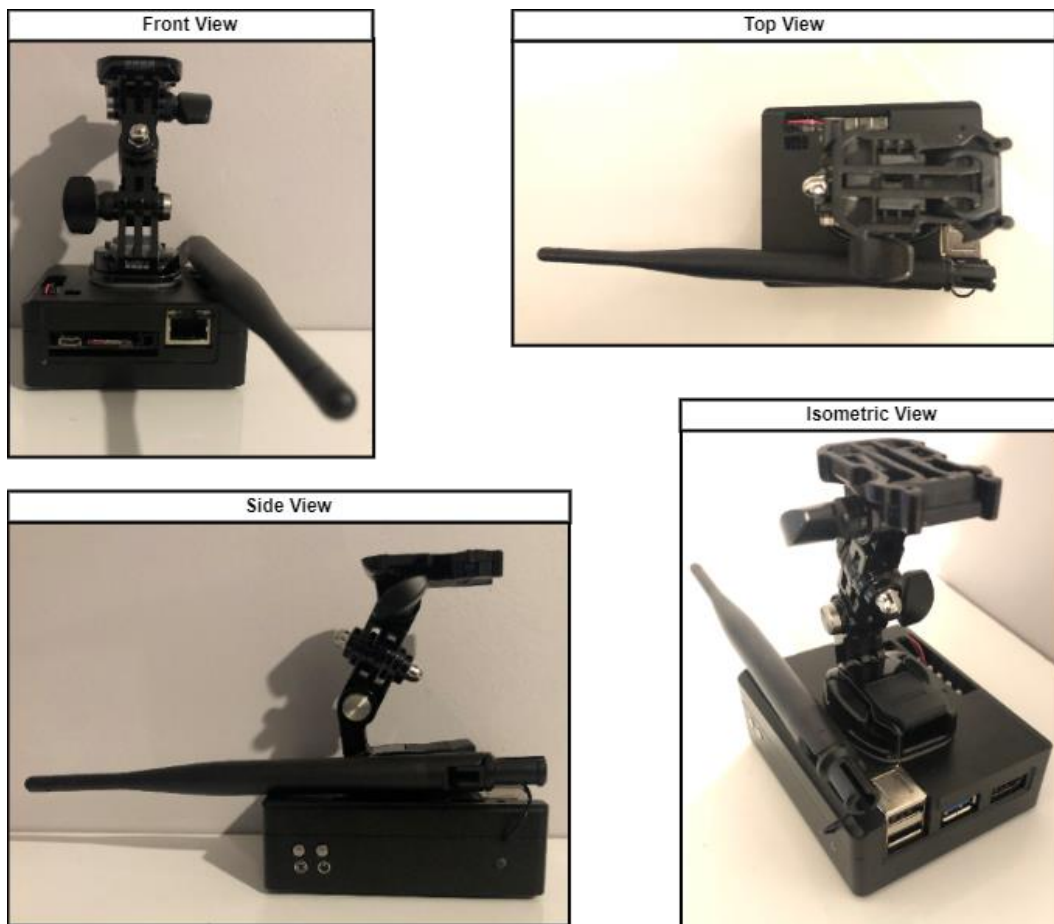


Figure 74: Front, Top, Side, and Isometric views of SAM when housed in the protective enclosure. The omnidirectional antenna attached to the top improves wireless communication and changes the shape in which the signal is transmitted.



Figure 75: Example of SAM being equipped to an existing system via the Zero-X Edge drone. SAM is mounted to the underside of the chassis using the Go-Pro accessories.

Analysing Figure 73 we see that devices are connected using USB/HDMI and auxiliary connectors. Ideally, monitory, and operational devices would be wireless or sheathed in a long tether to ensure that the hardware is clean and that that the integrity and mobility of the system SAM is attached to is not compromised. In the case that SAM were connected to a stationary system e.g., for surveillance tasks, this would not be an adverse factor.

Figures 73-75 have visualised SAM as a fully-integrated embedded system capable of being equipped into existing robotic infrastructures. While SAM was easily mounted under the Zero-X Edge drone, SAM's load weighed at 544g i.e.:

$$F_{Load} = M \times A$$

$$F_{Load} = 0.544 * 9.81 = 5.34N$$

Unfortunately, the load force ' F_{Load} ' was too great for the drone in Figure 75, thus preventing SAM from being lifted off the ground when attached. Successfully attaching SAM to the device proved the concept that SAM could be equipped to existing systems. However, for the systems to operate without constraints, they would need to be capable of handling a 5.34N load.

Component	Model Name	Description and Purpose
Single-board computer	LattePanda 4G/64GB	Handles all on-board processing and computation of code. Serves as SAM's "brain".
Protective enclosure	Aluminium alloy case for LattePanda V1.0	Protection and cooling (through a 5V cooling fan) of the single-board computer. Also provides some ingress protection.
Wi-Fi Antenna	2.4GHz 6dBi Antenna with u.FL/IPEX Connector	Omnidirectional antenna for improving SAM's wireless communication capabilities.
SD Card	SanDisk Extreme PLUS 128GB A2 SD Card	Allows for local, on-board storage of SAM's software and code as well as generated reports, classification models and log data, allowing for more freedom in regard to storage and memory.
Mounting clips	Go-Pro Grab Bag	Allows SAM to be equipped to existing robotic systems through combinations of mounting clips and surface mounts. The current configuration as per Figures 74 and 75 uses two surface mounts (one on SAM and one on the system being mounted on), two mounting clips and an intermediate link.
Camera and Microphone peripheral devices	Microsoft LifeCam HD-3000	720p, 16:9 HD camera and microphone input device, allowing for the monitoring and classification of facial expression and speech signals (including allowing speech-to-text functionality).
Consistent light source	VILTROX L116T RA CRI95 Super Slim LED Light Panel	Analog light to provide consistent lighting conditions in a controlled environment. The VILTROX light has adjustable settings to vary brightness and light warmth from a cool light to a warm light. For experimentation, the settings were: 3900K and 97% brightness.
Existing robotic system	Zero-X Edge Drone	Proof of concept of how SAM can be integrated into existing robotic infrastructures. The drone as shown in Figure 75 is a small-scale, quadrotor, wireless aerial drone.
Software Prototyping Computer	Dell XPS 13 9360	This PC was used for software prototyping and development including training/validating models, developing, testing and compiling code and documentation.
Monitory device	ASUS Monitor	1080p external display used to view SAM's GUI when connected to the hardware device via an HDMI cable.
Operational devices	Standard USB mouse and keyboard	Used in conjunction with the monitory device to simulate a user's environment. The keyboard allows users and subjects to login and the mouse allows for navigation throughout the GUI windows. Two alternatives are that: (i) SAM could be equipped with a touch screen display to remove the need for a mouse device and (ii) SAM could be equipped with wireless mouse and keyboard devices to remove the number of <i>physically connected</i> devices.

Table 22: Hardware bill of materials for the development stages and embedded system implementation of SAM. Figures 73-75 show component integration into the design. A system diagram showing how embedded system components are interfaced is visualised in Figure 76.

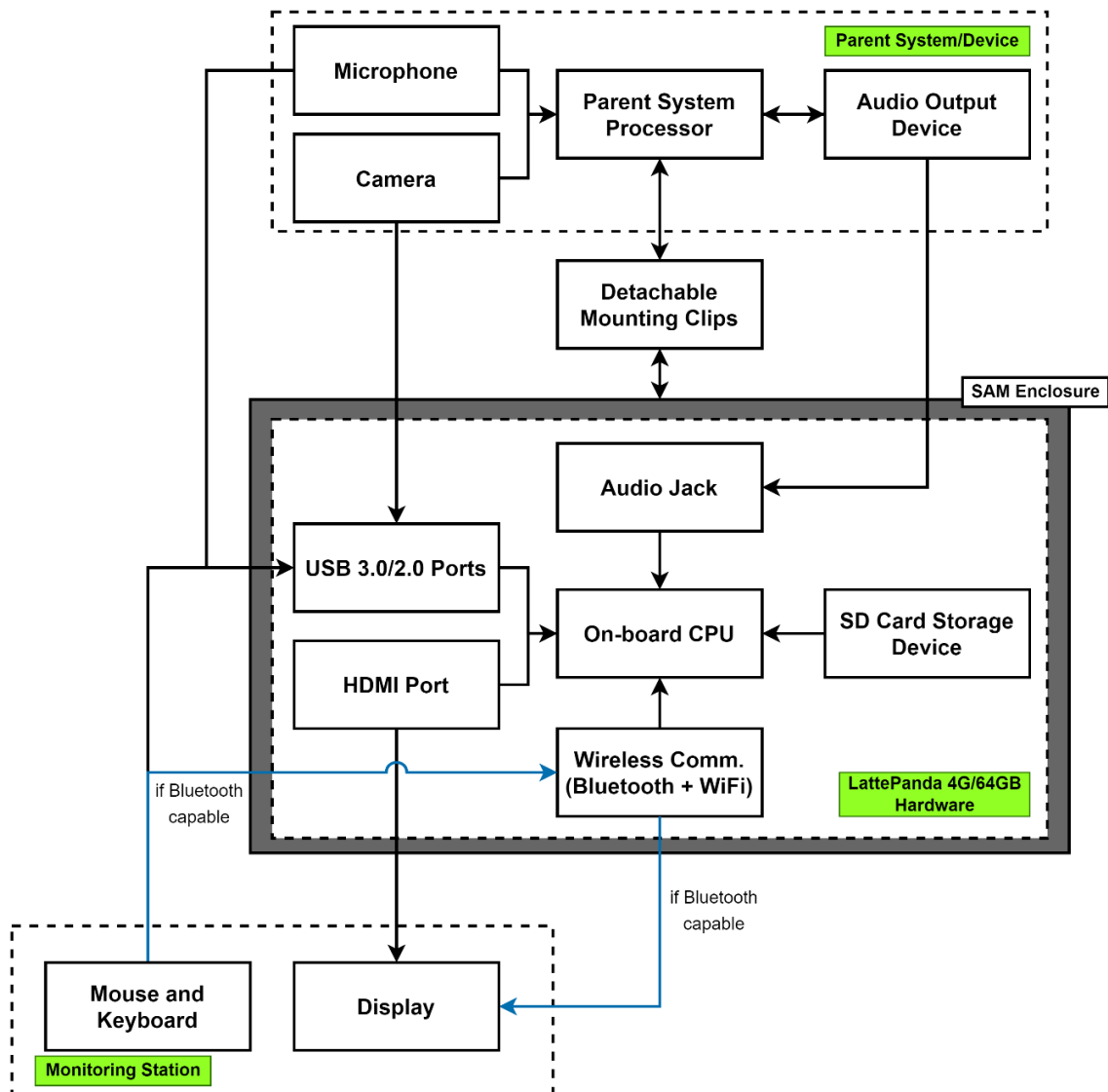


Figure 76: Visualisation of how SAM interfaces with other devices and subsystems when deployed as a hardware solution (as experimented with the LattePanda 4G/64GB development board).

Dissecting Figures 73-76 and Table 22, we can identify that there are three necessary subsystems required to fully deploy SAM as a hardware solution: (i) SAM itself, which has been ported and deployed on a physical device, (ii) the parent system, which would be equipped with the necessary peripheral devices such that it can be integrated with SAM, (iii) the monitoring station, which would comprise of a display and mouse/keyboard so that the user can: (i) assess the GUI, (ii) monitor SAM’s decision-making processes and, (iii) interact with the system.

The tabular and graphical representations of the system in this section report the hardware and software used throughout prototyping and deployment phases of SAM. Table 23 describes the Python packages used for the overall functionality of the system, including model training, validation, and GUI design. Reporting hardware/software bills of materials and a visual representation of the embedded system ensures that replicability of this work is possible, with an extended description of the packages used shown in Appendix S.

Package	Version/Distribution	Purpose
Python	3.6.8	Python3 environment used for coding the project including designing, model training and all software components.
time	Dependent on Python dist.	Allows for time-related functions and the manipulation of time variables.
math	Dependent on Python dist.	Standard arithmetic and mathematical functions
os	Dependent on Python dist.	Allows for interfacing with the operating system (OS) and manipulate files and directories.
sys	Dependent on Python dist.	System specific parameters and functions – allows for interaction with the interpreter
datetime	Dependent on Python dist.	Allows for the manipulation of time and date functions and variables.
base64	Dependent on Python dist.	64-bit encoding function used to encode binary data.
io	Dependent on Python dist.	Handling input and output streams
threading	Dependent on Python dist.	Allows for manipulation and use of threading function – allowing for parallel thread operation.
pickle	Dependent on Python dist.	For serializing and de-serializing (saving and loading) Python objects including ML models.
random	Dependent on Python dist.	Random number generation and related functions.
difflib	Dependent on Python dist.	Deployed for sequence comparing in this work.
numpy	1.16.3	Advanced mathematical functions and allows for the manipulation of numpy arrays.
pandas	1.1.5	Used for data analysis and manipulation, specifically in the case of aspects like dialogue and databases structures used in the system.
csv	1.0	Allows for reading and writing of CSV files
PySimpleGUI	4.56.0	An extensive library that allows for the creation of GUI elements and user-centred Python programs.
scipy	1.4.1	Scientific computing package including additional data science, analytics and statistical functions and tools.
reportlab	3.6.6	Used for generating pdf documents i.e., the generated reports that exist throughout.
matplotlib	3.1.1	Graphical library for plotting functions. Used for generating graphical images present in the system.
PIL	6.2.0	Python Image Library containing image processing functions and tools useful for pre-processing.
cv2	4.5.1	OpenCV, an open-source computer vision package which includes facial recognition tools for example.
pyaudio	0.2.11	Allows for audio data manipulation – pivotal for speech processing tasks.
speech_recognition	3.8.1	Allows for speech recognition tasks
soundfile	0.10.2	Another audio data manipulation library which was used with pyaudio for speech processing tasks
pyttsx3	2.71	Python text-to-speech library, giving SAM the ability to communicate with users.
sklearn	0.24.2	Sci-kit learn, a data analysis and machine learning library used in this work for generating ML Models
opensmile	2.2.0	Python wrapper of the openSMILE (open-source Speech and Music Interpretation by Large-space Extraction) software
TensorFlow	2.2.0	An additional machine learning and artificial intelligence library
Keras	2.3.0	Allows for generating deep neural network models and CNNs.

Table 23: Software bill of materials showing imported Python packages used throughout development and deployment stages of SAM. This table does not include package dependencies. A full list of packages produced through the ‘pip list’ command is shown in Appendix S.

5.3.1 Key Challenges Faced

Throughout the various phases of this project from its inception to the ultimate deployment of SAM as a modular embedded system, there have been many challenges and achievements that have come as a result. This work has been a success and this thesis has detailed its achievements throughout previous chapters and sections and will continue to do so in the remaining chapters to follow.

However, the process was not easy, and many challenges were faced throughout the research and development cycle. By identifying and reporting these challenges, it allows for an understanding of the journey that has been undertaken as well as a deeper understanding of what bottlenecks were faced throughout the development cycle and the overall research task. In summary, the key challenges that were encountered include:

- Challenges faced in back-end development in regard to finding and selecting an adequate feature set, experimenting with the likes of GeMAPS and ComParE2016 feature sets.
- A large amount of time was dedicated to tuning OpenCV parameters for facial recognition functionality as it was found that variables like ambient light conditions had a strong effect on facial recognition performance.
- Challenges in regard to machine and deep learning model design, training, tuning, and validation were made more prominent due to the time taken to train, test, and re-train all the models that exist in the system – 43 classifiers in total to consider.
- During the research and literature review stage, a lot of considerations were made in regard to how to split data at different classification levels, comparing literature-based and experimental-based models and architectures.
- Getting the mask detection functionality to work properly in real-time was a difficult task despite having close to 100% separation and classification accuracy. One significant challenge was tuning the facial recognition function and lighting conditions to account for the visual occlusion as this would change the ‘shape’ of the face.
- Regarding hardware and embedded system integration, there were challenges that arose with peripheral devices. Some microphones and cameras had connection issues and faulty wiring for example which resulted in different input devices being purchased and experimented with.
- Another example of challenges faced with peripheral device integration included the shape of the camera lens. In earlier experiments it was found that the camera input device that was used had a curved, fisheye lens which distorted the input image. It was found that this adversely affected the classifier performance as facial features would be stretched and curved as a result. In response, other peripheral devices were purchased and experimented with.
- The addition of the real-time speech recognition display on the GUI was a challenging task. In previous work, the wave form was updated and displayed to the user in chunks every 2 seconds. Overcoming this hurdle was pivotal and a major goal in the initial stages of the project as real-time displays of raw speech data would improve the interpretability of input signals and would therefore improve system accountability.
- Some hurdles were faced w.r.t creating user and subject databases in terms of the logic behind creating new users and subjects and what would constitute success and failure conditions.

- Regarding initial clustering experiments, other clustering approaches outside of cosine similarity-based clustering were explored. Comparing and understanding the pros and cons of each approach was a time-consuming task.
- In previous versions of SAM, disconnecting a peripheral device like a microphone or camera would cause the system to crash. This was a crucial error condition that needed to be handled in a ‘cleaner’ way. Implementing the ability to seamlessly disconnect and reconnect peripheral devices without breaking the system was a challenging task that was eventually overcome, and vastly improved SAM’s usability.
- Multi-threading and parallel processing introduced many challenges. Specifically, in this work there were challenges in regard to implementing text-to-speech and speech-to-text processes on separate threads. SAM had to be capable of ‘listening’ and ‘speaking’ while still processing input information and user commands. Thus, various threads needed to be executed such that these processes would work in parallel to the existing GUI threads.
- Implementing *all* of the report generation functions throughout the system was a time-consuming challenge. As shown in Section 4.2, the complexity of the code vastly increased from Version 1 → Version 2. This was directly attributed to the report generation functionality.
- Giving SAM a ‘voice’ was challenging as it required changing the Windows registry and manipulating registry files (on both the embedded system’s on-board PC as well as on the PC used for prototyping and development).
- Difficulties were faced when creating the SAM executable and porting it onto the embedded system. The challenges were due to dependencies where SAM could either be built on-board the LattePanda or could be distributed as a stand-alone executable. While being challenging, this issue was rectified swiftly and both solutions were valid.
- As discussed in previous sections, the COVID-19 pandemic resulted in a “chip shortage” and major delays in lead times for components. As a result, procuring different development- and single-board computers was a challenge. Thus, making it necessary to test SAM on the LattePanda 4G/64GB development board.

5.4 Assessing SAM using the AXAI Framework

A survey was conducted to validate that SAM is an improvement on previous works, specifically the ASAM designed in 2019 [1, 19]. Given that SAM was developed with the AXAI framework at the core of its software design and development process, the initial hypothesis is that it would score highly in regard to predictive accuracy, comprehensibility, and system accountability vectors when assessing it using the AXAI framework.

The survey consisted of eight assessors whose experiences and occupations included: engineering professionals, academics and biomedical and social sciences professionals. All assessors were given a guide on how to use and assess the system including a copy of the scoring criteria as shown in Table 6. Assessors were then given a copy of SAM Version 1 and on average, spent approximately 15 minutes using the system. Assessors then reported their ‘ S_A ’ and ‘ C ’ component scores as shown in Table 24.

Predictive accuracy vector components i.e.: (i) $r_{tst-trn}$, (ii) d_{trn} , (iii) O_{fp} can be derived using information known at the time of training the models. Similar to the ASAM, the test/train data ratio was maintained at 80:20 throughout all versions of SAM’s development, resulting in a normalised score of $r_{tst-trn} = 1.0$. As reported in previous sections, the RAVDESS dataset was used for model

training and validation, with facial expression samples amounting to 1500-2500 samples per class and speech classifiers being trained with 96 samples per class. Therefore, the d_{trn} parameter using both classifier modalities amount to a score of 4.46/5, which is a normalised score of $d_{trn} = 0.892$. The false-positive naming occurrences ' O_{fp} ' can be determined through validation tasks, with SAM achieving an average naming error of 10.52% when considering both facial expression and speech classifiers. The naming error corresponds to a score of 4.965/5 i.e., $O_{fp} = 0.993$. Given the above derivations, the predictive accuracy can be calculated using (31):

$$\begin{aligned}
 P_A(S, P) &= \sqrt{r_{tst-trn}^2 + d_{trn}^2 + O_{fp}^2} \\
 &= \sqrt{1^2 + 0.892^2 + 0.993^2} \\
 &= \sqrt{1 + 0.795664 + 0.986049} \\
 &= 1.66785
 \end{aligned}$$

The comprehensibility ' $|C|$ ' and system accountability ' $|S_A|$ ' scores reported in Table 24, along with the P_A calculated above, can be used to compare the performance of SAM with those of the DAASAS and ASAM which were reported in Section 3.5. Analysing Figure 77, we see that SAM outperforms both systems and shows a quantifiable improvement in comprehensibility and system accountability, which was desired for it to be considered as an improvement on previous work.

The results show that by incorporating AXAI into the software design and development process, one can improve the user experience while promoting explainability and transparency of internal system logic and decision-making processes. This survey, and the comparison with other systems shows that while SAM may not be a perfect execution of accountable, explainable AI, it is definitely a large step in the right direction. While the application of SAM is within the domain of affective computing and intelligent agents, the design process used is universal, and principles used for designing SAM from the first iteration to the last, can be translated into other domains and applications.

Score No.	Sym-bol	Assessor ID								Average using 5-point score	Normalised Average Score
		1	2	3	4	5	6	7	8		
1	$ C $										1.275
3	$ S_A $										
4	T_{it}	3	5	3	5	5	4	3	4	4.00	
5	T_{pr}	5	3	3	3	4	3	1	4	3.25	
6	T_{pn}	5	5	2	4	4	4	2	4	3.75	
10	I_{in}	5	5	4	4	4	4	4	5	4.38	
11	I_{pro}	5	3	2	4	5	3	3	3	3.50	
12	I_{out}	5	5	4	4	5	5	4	5	4.63	

Table 24: SAM_V1 user experience and assessor scores using the 5-point scale identified in Table 6. This table indicates the comprehensibility and system accountability of SAM.

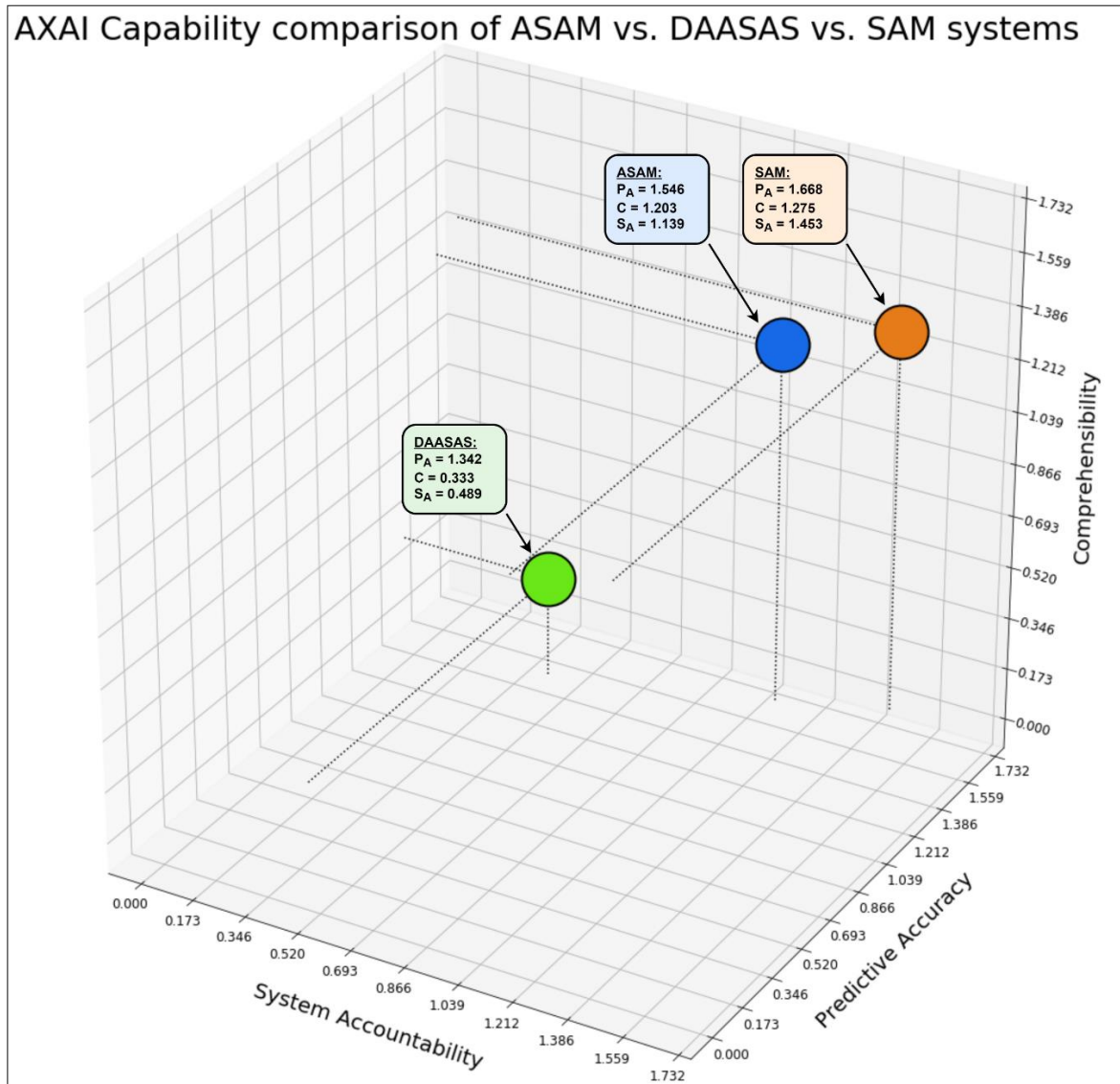


Figure 77: AXAI capability comparison of DAASAS, ASAM and SAM systems, showing predictive accuracy, comprehensibility, and system accountability scores and visualising them in a 3-dimensional space.

Chapter 6: Parallel Research and Other Contributions

6.1 Blind Source Separation and Crowd Sentiment Analysis

Chapters 1 and 2 introduced and discussed blind source separation (BSS) as an unsupervised learning technique and how it could be deployed as a technique to decompose signal mixtures for the purpose of estimating source signals given an observed signal mixture. The overall governing equations for BSS problems have been defined through (1) and (2) i.e.:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{b}(t), \quad 1 \leq t \leq T$$

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t) = \mathbf{W}\mathbf{A}\mathbf{s}(t)$$

where ‘ $\mathbf{x}(t)$ ’ is an observed signal, ‘ $\mathbf{y}(t)$ ’ is the estimated signals, ‘ \mathbf{A} ’ is the mixing matrix and ‘ \mathbf{W} ’ is the demixing matrix. However, as discussed in Section 2.1.1, there are various unique approaches that can be deployed to derive the demixing matrix parameters including ICA, MICA, and DTWCT for example. While Chapter 2 may have discussed the theoretical foundations of BSS, this section discusses how and where this theory can be applied i.e., through crowd sentiment analysis.

As has been extensively discussed in this thesis and in previous work, human expressions are complex responses to internal and external stimuli, and these expressions can result in changes in speech as we aim to augment our vocal responses such that our feelings and intentions can be conveyed. Thus far, we have discussed how speech signals can be classified when a single speaker is being assessed. However, classifying the distribution of affective states in a group of speakers is yet to be attempted, thus highlighting the contribution of this work.

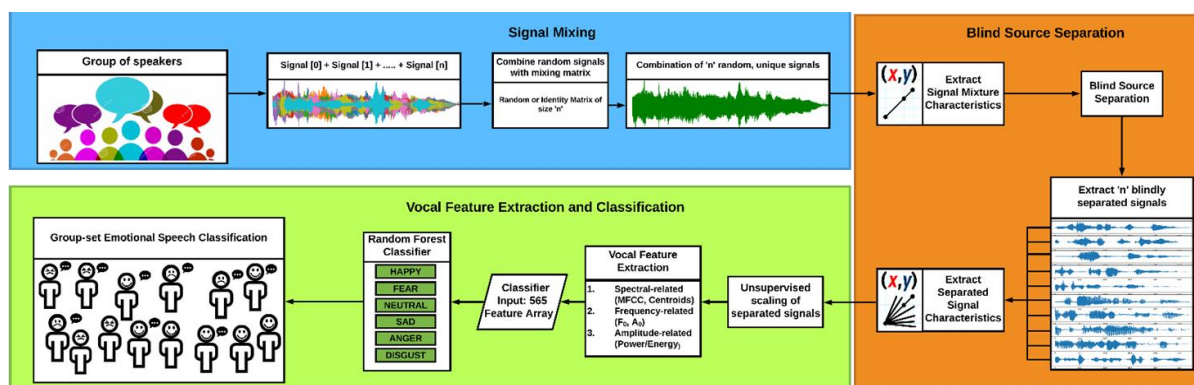


Figure 78: Blind source separation and crowd sentiment analysis flowchart showing three major stages of the process: Signal mixing (blue), blind source separation (orange) and vocal feature extraction and classification (green).

The purpose of this research is to combine BSS techniques with affective speech classification to separate speech mixtures and determine the affective states of speakers in a group/crowd scenario. The problem of separating and classifying speech signals when analysing a signal mixture is exacerbated by the structure of the mixture itself. The implication of this problem is that a mixture needs to be defined as being either Gaussian or *non*-Gaussian (identifying statistical dependence vs. independence) prior to commencing with solving the problem.

If a group of speakers are engaging in separate, independent conversations and are experiencing different sentiments, an assumption would be that the signal mixture is non-Gaussian and that there

would be no mutual information contained in the mixture. Thus, an independent separation technique like Fast-ICA would be deployed for separation. Conversely, if mutual information *does* exist in a signal mixture and speakers are experiencing similar emotive experiences, a dependent component analysis would need to be deployed for separation (like MICA and DTCWT).

The separation algorithm would then split the signal mixture into ' n ' estimated signals, one for each speaker in the group and the affective speech classification algorithm would be iteratively deployed for each estimated signal, determining the state of each speaker. By classifying the state of every speaker in a mixture, it therefore allows you to determine the distribution of states in a group/crowd. This allows for tasks such as group sentiment analysis as well as outlier detection. If a group containing twenty speakers classified nineteen as 'happy' speakers and one as 'sad', then this outlier could be identified as an observation that would need to be handled depending on what application this technology is being deployed in. We can therefore see that such a system could be applied in domains and sectors including: entertainment, public/national security, education, criminology.

Through this research, it has been proved that group sentiment analysis can be achieved through a combination of BSS and affective speech classification under the assumptions of both statistical dependence and statistical independence. For BSS, two methods introduced in Section 2.1.1 were experimented with: (i) Fast-ICA, (ii) dual-tree complex wavelet transform. Spectral-, Frequency- and Amplitude-related features were then extracted from separated signals and a random forest classifier was deployed for classification using a discrete classification architecture (six or seven classes). The flowchart in Figure 78 visualises how speech data is manipulated throughout the process, including mixing, separation, and classification stages.

The TESS, RAVDESS, CREMA-D and EMO-DB datasets (see Table 11) were all used throughout experimentation. For each dataset experiment, results were collated over the course of 100 tests where in each test, a unique combination of 5 randomly-selected samples were chosen for mixing, separation, and classification. The number of tests ' n_{tests} ' is therefore defined by:

$$n_{tests} = \frac{n_{dataset}}{n_{mixture}} \quad (38)$$

where ' $n_{dataset}$ ' defines the number of samples in a dataset and ' $n_{mixture}$ ' is the number of samples that have been mixed i.e.: $n_{mixture} = 5$. There are common key performance indicators (KPIs) used to measure separation performance in BSS literature, which model human perception [152-154]. These include:

1. **Source-to-distortion ratio (SDR)** – reflecting the overall performance of the separation algorithm, modelled by:

$$SDR = 10 \log_{10} \left(\frac{\|e_{target}\|^2}{\|e_{interf} + e_{artif}\|^2} \right) \quad (39)$$

2. **Source-to-interferences ratio (SIR)** – models the separation algorithm's ability to isolate and remove interferences from a signal, modelled by:

(40)

$$SIR = 10 \log_{10} \left(\frac{\|e_{target}\|^2}{\|e_{interf}\|^2} \right)$$

3. **Source-to-artifacts ratio (SAR)** – models the separation algorithm’s ability to isolate and remove noise from a signal, modelled by:

$$SAR = 10 \log_{10} \left(\frac{\|e_{target} + e_{interf}\|^2}{\|e_{artif}\|^2} \right) \quad (41)$$

These KPIs are derived through orthogonal projection decompositions as discussed in [153] and use four variables: (i) s_{target} , (ii) e_{interf} , (iii) e_{noise} and (iv) e_{artif} . Through the log-scale we can deduce that SDR , SIR and SAR are all measured in decibels (dB), where: $dB \propto$ separation performance. In literature [152, 153], it is stated that the source-to-distortion ratio is the most important separation metric thus, separation algorithms should be optimised to improve SDR .

In this research task, SDR , SIR and SAR are measured for all signals that have been separated from a mixture, for all n_{tests} that have been conducted. Thus, calculating the mean separation performance for each algorithm applied for each dataset i.e.: \overline{SDR} , \overline{SIR} and \overline{SAR} . Classifier performances were measured using the validation classification accuracy defined by (33), i.e.:

$$Acc(\%) = \frac{n_{correct}}{n_{predicted}} \times 100\%$$

The experimentation results are reported in Table 24. All rows are an accumulation of results extracted across n_{tests} where $n_{mixture} = 5$ with randomised mixing matrix ‘ A ’ parameters. Speech signals are mixed, separated and classified as per the system architecture visualized in Figure 78. All separated speech signals were classified using the trained RFC models for predicting group emotive states in the group, the results of these experiments are represented through the classification accuracy columns.

Dataset	Independent Component Analysis				Dual-tree Complex Wavelet Transform			
	\overline{SDR} (dB)	\overline{SIR} (dB)	\overline{SAR} (dB)	Classification Accuracy (%)	\overline{SDR} (dB)	\overline{SIR} (dB)	\overline{SAR} (dB)	Classification Accuracy (%)
TESS	27.76	27.77	63.14	90.95	23.42	23.43	59.25	91.06
RAVDESS	31.71	31.71	76.40	70.10	32.67	32.67	76.66	69.87
CREMA-D	28.25	28.26	65.10	64.17	25.50	25.51	61.71	65.82
EMO-DB	26.55	26.62	51.81	92.01	26.59	26.69	52.01	91.74
Average	28.57	28.59	64.11	79.31	27.04	27.08	62.41	79.62

Table 25: Performance metrics for the BSS and crowd sentiment analysis system when experimenting on four datasets.

Discussing Table 25, we see relatively consistent performances for \overline{SDR} and \overline{SIR} metrics for cases of both statistical dependence and independence, with \overline{SAR} displaying the highest variance for both. Similar to the stereo instantaneous mixture evaluation in [153], \overline{SDR} and \overline{SIR} are very similar for all cases, with the \overline{SAR} KPI being much higher in comparison. Observing high \overline{SAR} values evidence that both methods are very capable of isolating and removing noise from the signal.

Across all datasets, the ICA algorithm produced greater separation performances on average when compared to the DTCWT method. Regarding the performances of datasets, we see experiments using the RAVDESS dataset provided the greater separation performances, but the EMO-DB dataset provided the best classification performances.

A thorough investigation was conducted to find evidence of emotive state assessment in group scenarios using similar methods to those discussed in this report. The investigation found no evidence of such approaches, thus making one-to-one performance comparisons very difficult. While the separation of acoustic signals is prevalent across literature [152-156], a comparison can still be made in terms of separation in which our proposed method has proven to be very competitive.

However, there is evidence of classifying affective speech in literature [21, 87]. A survey conducted in [21] reports classification accuracies between 43-85% for affective speech classifiers using a variety of methods, datasets, and features. Comparing the classification accuracy results reported in Table 25, it shows that in all cases, the system being discussed is competitive and, in some cases, outperforms those discussed in the survey.

By comparing these results to those in literature, it shows that the presented BSS and crowd sentiment analysis system is technically capable of separating and classifying affective speech signals in mutually dependent and independent mixture conditions. Furthermore, the lack of similar approaches to group sentiment analysis in literature adds to the novelty of this work. If deployed, this system would be capable of equipping inspection systems with non-pervasive group sentiment analysis capabilities, allowing for the remote monitoring of intentions and moods in crowd and group scenarios. This highlights another application of affective speech classification, parallel to the work that has been discussed as part of SAM's architecture.

Future work could see this system being incorporated as an auxiliary function that exists as part of SAM's classification architecture, using the inherent, continuous speech classification architecture in place of the speech classifier discussed as part of this work. Such an addition would expand SAM's intelligence, making it capable of understanding changes in expressions for both single speakers and groups of speakers.

6.2 Dynamic Micro-expression Detection using Hierarchical Clustering

Through researching and developing SAM's rule-based system and hierarchical classification network, some parallel work went into defining a rule-based system for defining micro-expressions in full-, upper- and lower-facial regions. Throughout this work, the concept of exposing black-box ML systems has been discussed. This research evidenced another way in which that goal can be achieved within the realm of affective state assessment systems. Furthermore, by considering micro-expressions of states, this work is built on the foundation (similar to SAM) that expressions are continuous and have temporal considerations [81-83, 130].

The goal of this research task was to design a rule-based system capable of exploiting the continuous nature of expressions such that various levels of expressions of affective states across various facial regions could be classified i.e.: (i) Full-facial, *macro*-level expressions, (ii) Upper-facial, *micro*-level expressions and (iii) Lower-facial, *micro*-level expressions.

The construction of each of the above subsystems is built on the foundations of the cosine similarity-based clustering approach discussed in Section 5.1, applied on the CK+ dataset [125]. To summarise, the CK+ dataset has continuous samples separated into seven classes: (i) happy, (ii) sadness, (iii) anger, (iv) fear, (v) disgust, (vi) surprise, (vii) contempt, with actors changing expressions from neutral to activated. The separation technique was applied to detect the similarity across a continuous sample and construct cluster boundaries transforming $n_{classes}: 7 \rightarrow 21$.

Then, by deploying linear discriminant analysis (LDA), this allowed for the further separation of clustered samples and a projection onto a linear discriminant (LD) plane. The graphical representation of clusters on the LD plane prompted an investigation into the relationships between cluster centroids and how facial action units (as defined in Table 2) can be used to form a new, rule-based system for classifying the different levels of expressions, which is based on continuous muscle movements.

Identifying some of the related works, research conducted in [119, 157-160] offer some unique examples of micro-expression detection and categorisation. Pfister et al. in [157] propose a framework for detecting spontaneous facial micro-expressions, using temporal interpolation for feature mapping, prior to implementing traditional machine learning classifiers like support vector machines, multiple kernel learning and random forests. Comparatively in [158], Xu et al. propose a “Facial Dynamics Map” which characterises micro-expression movements using granular pixel features and an algorithmic approach based on optical flow estimation. Their work employs a support vector machine classifier to identify and categorise different types of facial micro-expressions [158].

Polikovskiy et al. present an approach in [159] that uses the EMFACS as the basis for micro-expression detection. Their method divides full facial images into smaller facial regions based on action unit locations and a histogram of oriented gradients (HOG) approach is combined with a K-nearest neighbour classifier for the purpose of detecting micro-expression and action unit activations.

Discussing the work in [119, 160], we see that rather than exploiting visual cues, the authors exploit thermal features for facial expression classification. Comparing their findings, in [119], we see that a feature comparison was reported, measuring both upper- and lower-facial region muscle activation temperatures when expressing different states whereas in [160], the authors report the differences in classifier performances when different sub-regions of the face are used for feature extraction.

Comparing the parallel research discussed in this section to those related works [119, 157-160], the proposed solution not only successfully detects and classifies micro-expression activations in the upper and lower-facial regions, but also uses them as a vehicle for macro-level dynamic affective state assessment. The system is capable of classifying twenty-, upper- and lower-facial region micro-expressions as well as twenty-one macro-level facial expressions, thus promising a robust affective state assessment solution that accounts for the multidimensional, dynamic nature of human expressions. Furthermore, the solution is unique when compared to the approach taken when designing SAM.

A major benefit of deploying the LDA algorithm is its ability to optimise cluster centroid separation by maximising inter-cluster variance and minimising intra-cluster variance. Figures 78 and 79 visualise the raw clusters and the corresponding cluster centroid locations when applied to full facial expression image data. The raw, CK+ image data used to generate the cluster representations had been pre-

processed by converting images to greyscale and rescaling them to 100x100 after the separation algorithm was applied.

Analysing Figures 79 and 80 we observe that linear trends from `_NEUTRAL` → `_PARTIAL` → fully activated expressions for all states that have been modelled. Furthermore, when we look at the cluster centroids in Figure 80, we see that there are two continuous axes that could be used to model all expression changes on a linear discriminant space, these being: (i) Sadness (left) → Happiness (top right) and (ii) Anger (top left) → Surprise (bottom right). A clearer representation of these axes is shown in Figure 81.

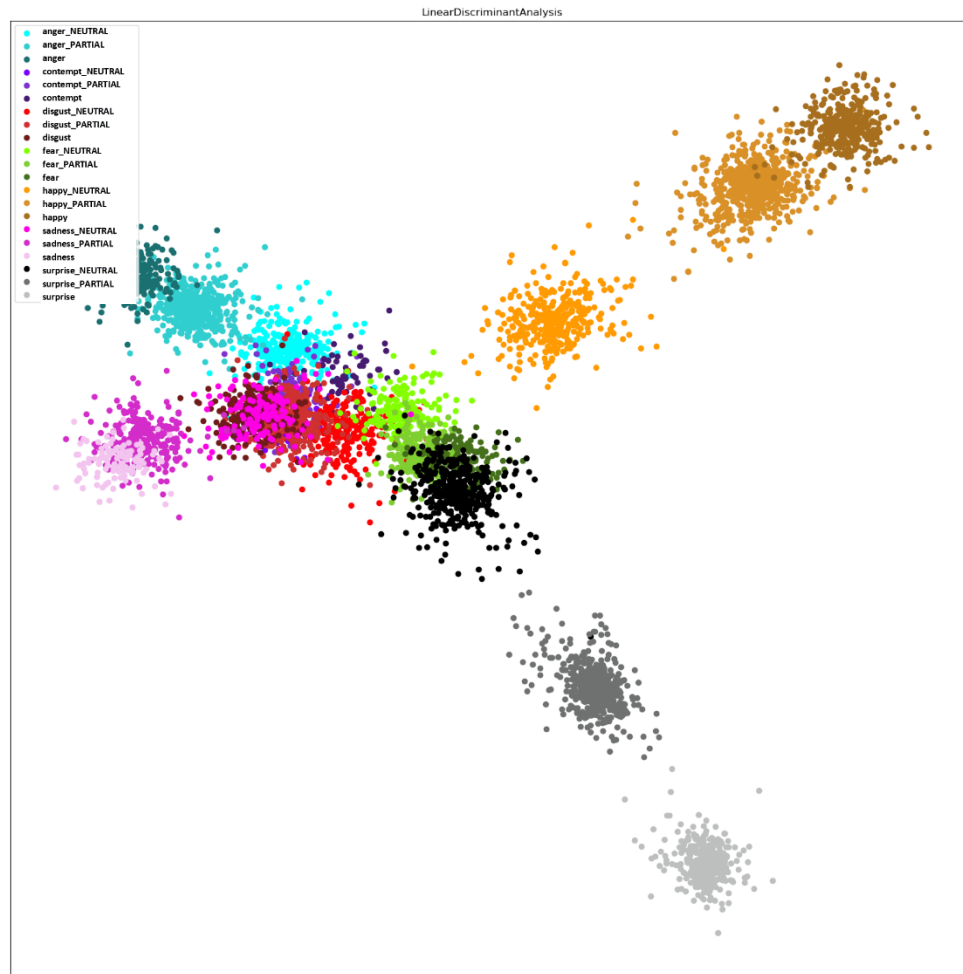


Figure 79: LDA clustering results of the initial, cosine similarity-separated full facial image data, highlighting all samples from the CK+ datasets (21 classes).

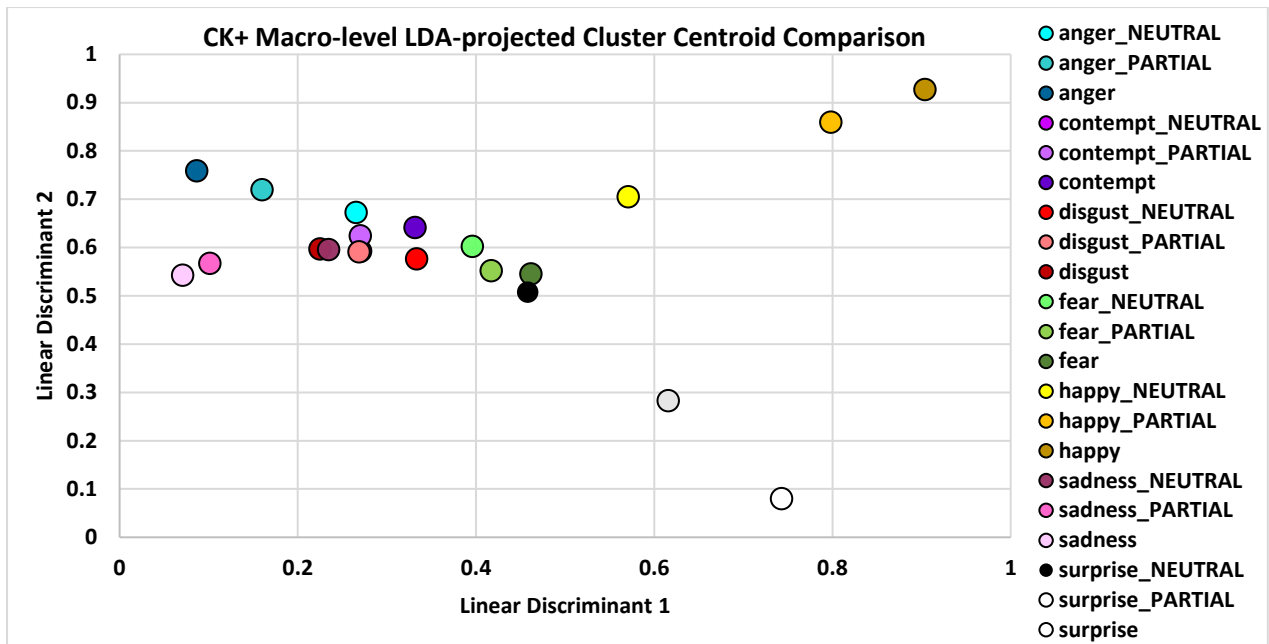


Figure 80: Cluster centroids for each of the 21 macro-level states, with centroids calculated using the raw cluster data in Figure 79.

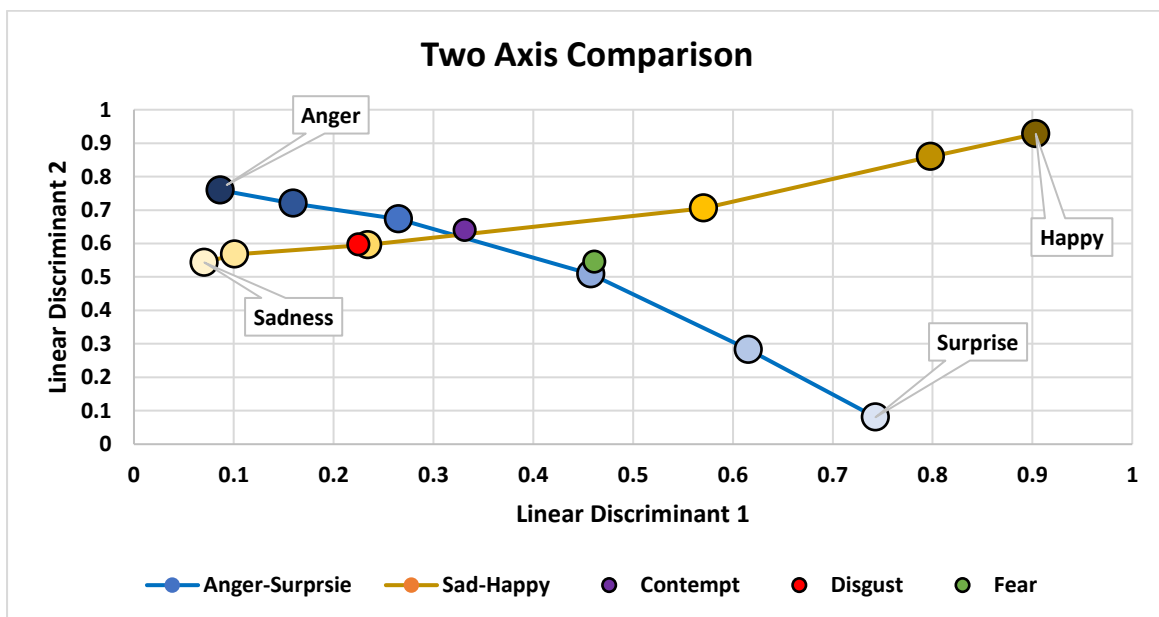


Figure 81: Visualisation of the two axes: (i) Sad-Happy and (ii) Anger-Surprise that can be inferred from the projection and cluster centroids in Figures 79 and 80.

But what do these axes represent in a theoretical sense? The linear discriminant space visualised in Figures 79-81 are low-dimensional linear discriminant representation of facial expressions, a mapping that corresponds with certain feature changes and variations in facial expressions at a higher level. We also see that the other three activated state centroids: contempt, disgust and fear reside on the two defined axes, with contempt existing at the intercept of the two axes, which is predictable given it is the most “neutral” expression relative to the other affective states being modelled.

Defining rules based on the logical foundations provided by the FACS presented in Table 2 and discussed prior, henceforth become very important. Comparing changes in muscle activations from state-to-state may help in indicating what these linear relationships actually represent in real-life and successfully doing so provides the foundation for a rule-based expert system capable of detecting and classifying micro-expressions.

Comparing sadness and happiness muscle activations, one can then model state-to-state transitions, visualising how expressions change based on muscle movements as shown in Figure 82. Similarly, comparing anger and surprise muscle activations, we can model this transition, as visualised in Figure 83. Note that in this case, both states evidence “raised upper eyelids” which is useful when attempting to derive a clearer relationship. Given the common facial muscles/areas used when changing expression from sadness → happiness and anger → surprise, we can define the following axis rules:

Sadness-Happiness Axis Rule: *Sadness and happiness share common facial muscle groups surrounding the mouth and cheek/eyelid regions, the axis that has been formed models the following transformations: (i) parallel relaxation of brows and raising of cheeks, (ii) raising of lip corners and mouth from an initial down-turned expression.*

Anger-Surprise Axis Rule: *Anger and surprise share common facial muscle movements surrounding the mouth and eyebrow regions, sharing a consistent ‘raised upper eyelid’ activation. Therefore, the state-to-state transition models the following transformations: (i) Eyebrows raise from an initial frowned/depressed position, (ii) Mouth opens from an initial tightened expression.*

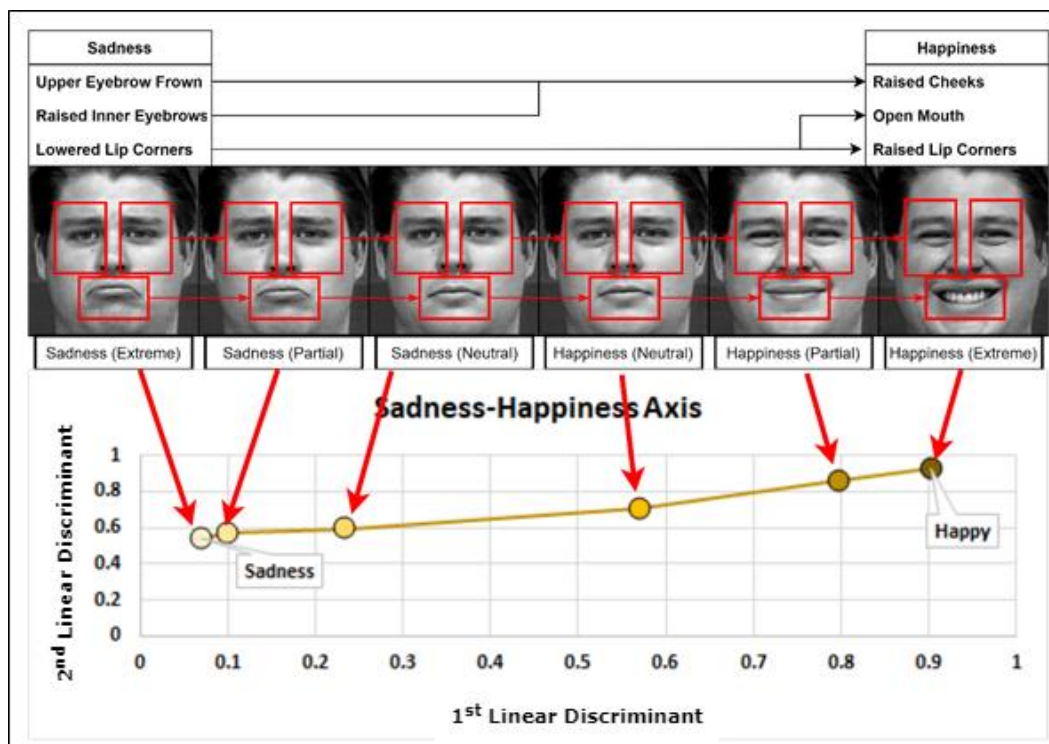


Figure 82: State-to-state transitions along the sadness-happiness axis as originally visualised in Figure 81, showing examples of partial and neutral states as well. Facial images used in this figure have been extracted from the CK+ dataset.

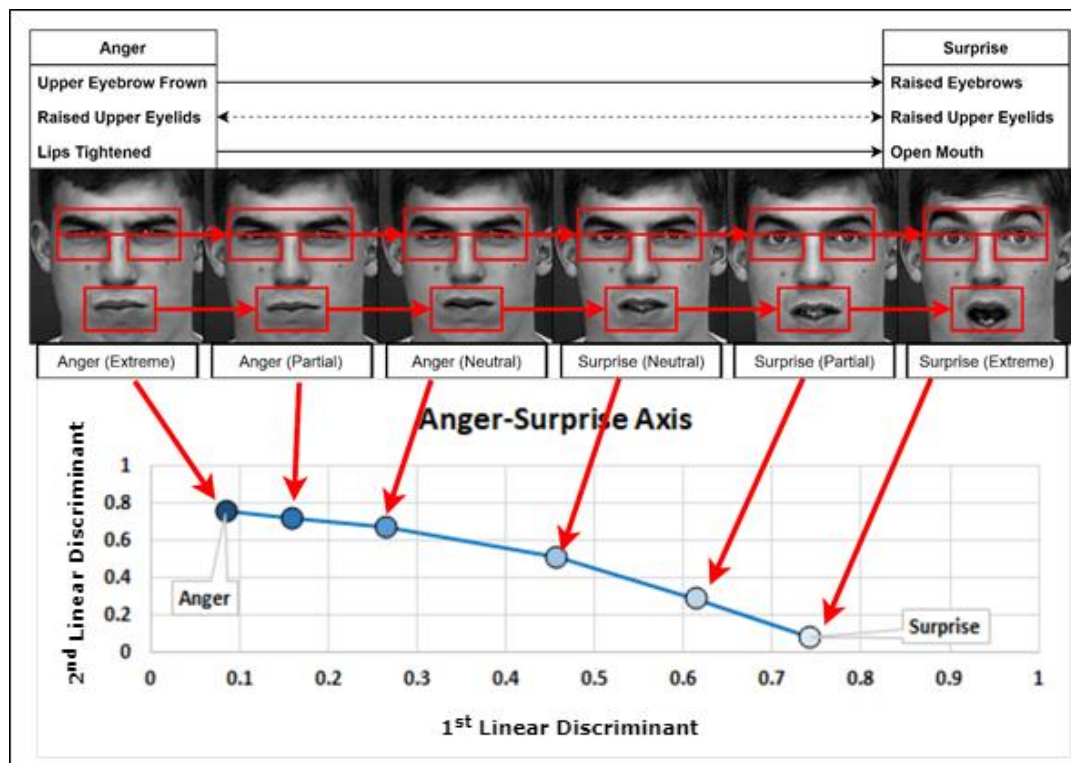


Figure 83: State-to-state transitions along the anger-surprise axis as originally visualised in Figure 81, showing examples of partial and neutral states.

State A → State B	$ \Delta LD_1 $	$ \Delta LD_2 $
Anger → Happy	0.81695	0.16830
Anger → Sadness	0.01590	0.21629
Anger → Surprise	0.65633	0.67885
Happy → Sadness	0.83285	0.38459
Happy → Surprise	0.16062	0.84714
Sadness → Surprise	0.67223	0.46255

Table 26: State-to-state comparison table showing the difference in n^{th} linear discriminants – LD_n . Highlighted values in the table display high and low $|\Delta LD_n|$ values.

Expanding on the two axis rules that have been formed above, we may postulate an initial hypothesis in regard to what the X and Y axes represent in this case (i.e., linear discriminant 1 and 2 respectively). Let linear discriminant ‘n’ be denoted by ‘ LD_n ’, Table 26 shows the $|\Delta LD_n|$ values when comparing states i.e., axis start and end points. Together, Table 26 and Figure 84 serve as the basis for developing and proving the rule-based micro-expression classifier that was developed as part of this research. Given all the graphical, logical, and theoretical evidence provided, the following hypotheses and macro-expression rules may be defined:

1. LD_1 relates to the openness of the mouth and the lower region of the face given the following articles of evidence:

- Sadness and Anger share a low ΔLD_1 . The two common actions between the states are: “upper eyebrow frown” and “lips tightened/lowered corners”.
 - The presence of two common actions would be troublesome if not for the presence of the surprise and happiness states, which also share a low ΔLD_1 . The common action between surprise and happiness revolves around raised lip corners and ultimately, the shape of the mouth.
2. LD_2 relates to the region around the eyes i.e., the eyelids and eyebrows – the upper facial region, evidenced by:
- The Anger-Sadness transition in Table 26 shows both a low ΔLD_1 and ΔLD_2 . If the initial hypothesis is that LD_1 is related to the mouth, then the second common action – “upper eyebrow frown” may be related to LD_2 , which supports the upper facial region relationship.
 - Analysing Figure 84 and the transition from anger to happiness, we see that the eyes remain the same shape, with the largest variance evident between *full* to *partial* anger states when the frown is relaxed slightly. Removing the lower half of the face, we can observe that there are similarities between the brow/eye region of the two states.
 - The third article of evidence is the large variance between happiness and surprise. Given that the open mouth is deduced as being referred to by LD_1 , we can clearly identify the difference between happiness and surprise frames in Figure 84 through the upper half region of the face, specifically the brow and eye regions, thus providing further evidence toward LD_2 relating to the upper facial region.

Inferring the above rules for the macro-level assessment of facial expressions allows one to define a relationship that is capable of mapping statistical features with real-world features, while also providing a vehicle for transient macro-level facial expression classification – a concept that has been deployed as part of SAM’s classification architecture as discussed extensively in previous sections.

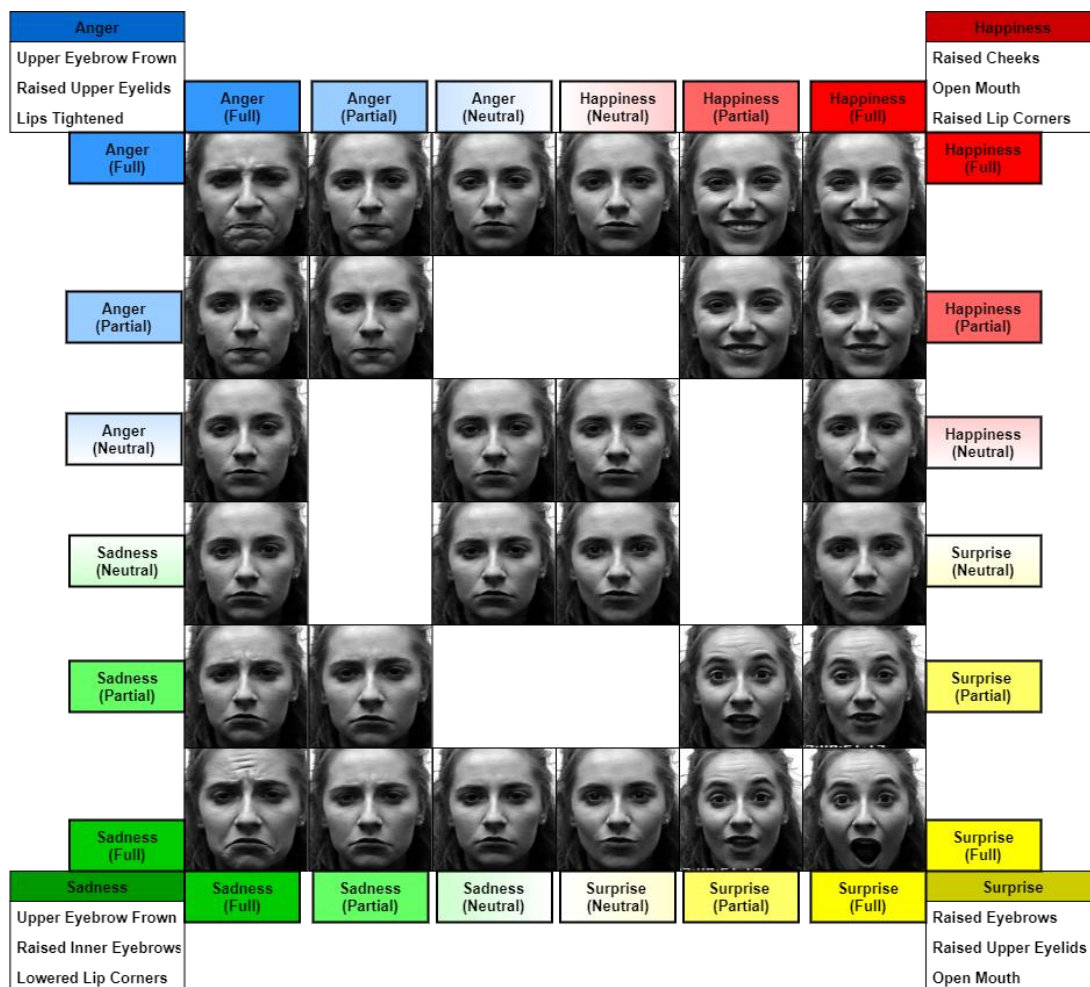


Figure 84: Visualised state-to-state transitions for anger, happiness, sadness, and surprise states. Combined with Table 26, this figure assists in the mapping of LD_n features to real-world features as defined by the above rules.

The above claims and hypotheses could be considered unsubstantiated if the necessary experiments are not conducted on top of these initial hypotheses. The upper- and lower-facial micro-expression analysis components of this research aim to prove the validity of LD_1 and LD_2 rules, while in itself, introducing a deeper analysis of dynamic facial expressions by focusing on continuous muscle movements in the upper and lower facial regions.

Chapter 5 discussed the occlusion detection functionality, where the lower half of a facial image was used to detect if an obstruction like a face mask was present. A similar algorithm was deployed for splitting upper and lower facial image data when conducting this research, only that both the upper- and lower-facial image data was used rather than just the latter. Extracting this image data allowed for the classification of micro-expressions in upper and lower facial regions. If the initial hypotheses were correct, then there should be a very discernible trend between states at the micro-level as this would indicate that the projected feature ' LD_n ' is related to a particular group of muscles.

In this work, the macro-level linear discriminant features are defined by ' LD_n ' – i.e., LD_1 = the lower facial region and LD_2 = upper facial region. Moving into *micro*-level analysis, let ' m ' describe the micro-

level features contained within the higher, n^{th} level regions i.e., ' $LD_{n,m}$ '. For example, $LD_{1,1}$ and $LD_{1,2}$ describe micro-expressions in the lower facial region.

As shown in Figure 85, the clusters share a similar $LD_{2,2}$ value, with the largest variance being in the direction of $LD_{2,1}$. We also see that the clusters trend linearly from anger to surprise along the $LD_{2,1}$ axis. Note that fear and contempt states display large variances along the $LD_{2,2}$ axis, sharing common $LD_{2,1}$ values (cluster centroid co-ordinates). Furthermore, it should be noted that contempt is an asymmetrical expression and as such could explain why it is such an outlier. But besides these two states, we see that the majority of expressions exist along the $LD_{2,1}$ axis.

The **Anger-Surprise Axis Rule** defined previously modelled two transformations: "(i) *Eyebrows raise from an initial frowned/depressed position*, (ii) *Mouth opens from an initial tightened position*." When combined with the hypothesis: " LD_2 relates to the region around the eyes, eyelids, and brows – the upper facial region", we can therefore state that the micro-level feature $LD_{2,1}$ refers to a translation of the eyebrows from an initial frowned/down-turned position to a raised position, making use of the *medial frontalis*, *levator palpebrae superioris* and *corrugator supercilii* facial muscles.

The lower facial region i.e., LD_1 is more sparsely clustered in comparison to the upper facial region. Analysing Figure 86, we see that most states reside on one side of the spectrum, sharing a similar $LD_{1,1}$ linear discriminant feature value, with happiness and its sub-states displaying the largest variance in $LD_{1,1}$. The notable trend observed in the lower region of the face is attributed to the micro-level feature $LD_{1,2}$ – the y-axis, showing disgust (top) and surprise (bottom) as the two extremes.

Figure 86 exhibits an axis that has been derived previously, only in this instance, it has been mapped from one feature space to another, this axis is the Sadness-Happiness axis, which varies in regard to the $LD_{1,1}$ feature. The **Sadness-Happiness Axis Rule** defined previously modelled the following two transformations: "(i) *parallel relaxation of brows and raising of cheeks*, (ii) *raising of lip corners and mouth from a depressed initial condition*". The second transformation relates to the lower facial region feature. This transformation regarding lip corner movements may be the causal link between the micro-level feature $LD_{1,1}$ and the real world. Analysing the variations in $LD_{1,2}$, we see disgust and surprise on opposite sides of the spectrum. Recalling Table 25, the immediate hypothesis is that $LD_{1,2}$ models the openness of the mouth, and the manipulation of the central lip muscles - *orbicularis oris* and *levator labii superioris*.

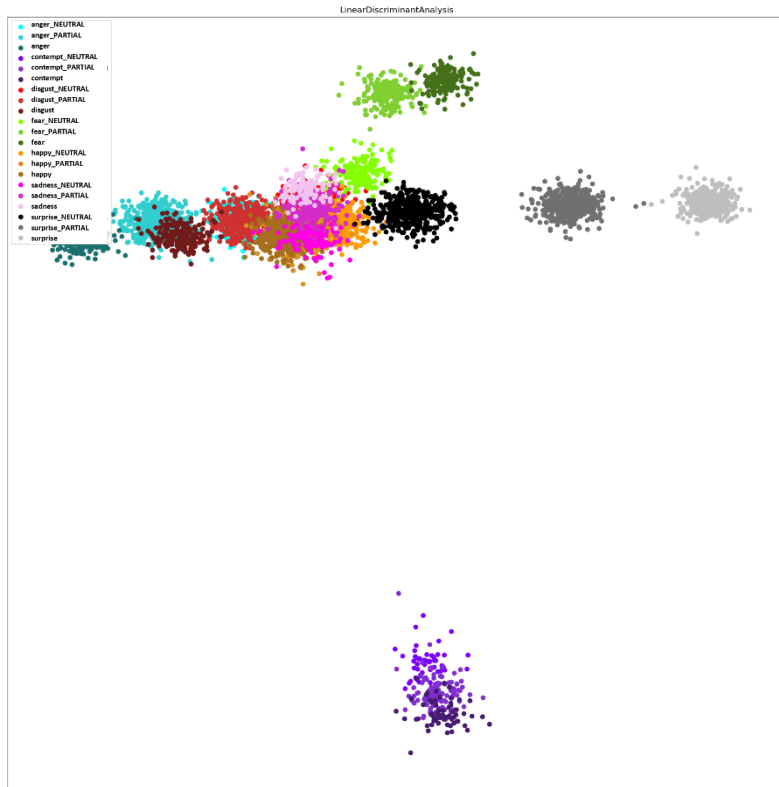


Figure 85: Two-dimensional linear discriminant space representation of the *upper* facial region, visualising $LD_{2,1}$ and $LD_{2,2}$ micro-expression features.

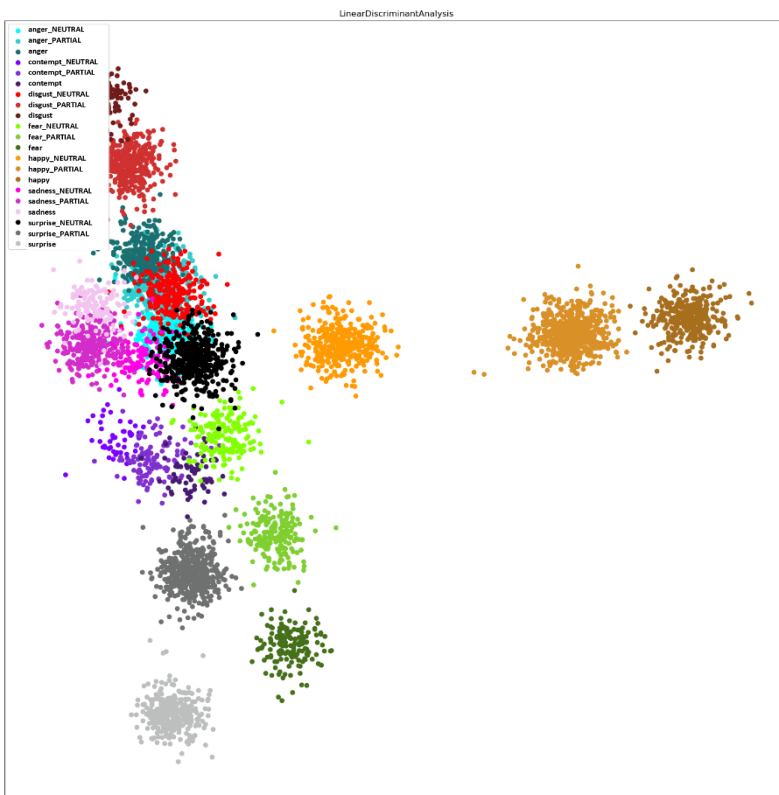


Figure 86: Two-dimensional linear discriminant space representation of the *lower* facial region, visualising $LD_{1,1}$ and $LD_{1,2}$ micro-expression features.

The evidence provided by the micro-level LDA-derived clusters prove the validity of the macro-level inferences i.e.: LD_1 relates to the shape of the mouth and the lower facial region and LD_2 relates to the upper facial region. Moving into the micro-expression (m^{th} level), we can define each axis as:

- $LD_{1.1}$ = Lip corner muscles, modelling a translation from a down-turned shape to a lifted shape.
- $LD_{1.2}$ = Models the “Openness” of the mouth and the manipulation of central lip muscles from closed to open.
- $LD_{2.1}$ = Models a translation of the eyebrows from an initial frowned position to a raised position.
- $LD_{2.2}$ = Used to identify expressions such as fear and contempt and asymmetrical movements outside of the spectrum from anger to surprise.

Defining the governing hypotheses and establishing the rule-based system provides a justification and foundation for upper and lower facial expression classification with the former serving as the basis for the research reported in [25].

To measure the classification capabilities of the proposed rule-based system, three traditional supervised learning methods were explored: (i) SVM, (ii) Random Forest, (iii) K-Nearest Neighbour, using linear discriminant features as classifier inputs. Model training was done using an 80/20 train/validation split, using the CK+ dataset to train and assess the models. Validation accuracy was the metric used to compare classification performances as defined by (42).

The bar graph in Figure 87 compares the performances of the three techniques when deployed for macro-level classification and upper- and lower-facial micro-expression classification. In this work, when comparing results, it is evident that lower facial region micro-expressions are classified with greater accuracy when compared to the upper-facial region and when classifying the full-facial region.

The classification performances of the lower and upper facial micro-expression classifiers are consistent with the human observations made in [160, 161], stating that the classification of lower facial expressions is on average, more accurate than that of upper facial expressions. Looking at Figures 85 and 86, we can see why this may be the case. The lower facial region micro-expression clusters show a larger separation across $LD_{1.1}$ and $LD_{1.2}$ axes when compared to the upper facial region micro-expressions, primarily showing variations in the $LD_{2.1}$ axis for most states. However, through these findings, the importance of both upper- and lower-facial region muscle movements when expressing a particular state can still be observed.

This research offered an alternative way of assessing continuous changes in facial expressions, particularly with respect to how labels could be changed by the developer to relay different information in a classification system. In a system like SAM, state information is shown to the user based on prediction results. However, using a system like the one discussed in this section, rather than saying “partially happy” or “apex sadness”, these outputs could be replaced by muscle movement actions based on the rule-based system that has been defined, showing changes in $LD_{n.m}$ features.

Furthermore, as has been reported in [25], this work has shown the ability of a system to classify upper facial expressions, which would be a necessary addition to a system that operates in an environment where visual occlusions like protective face masks are present. Rather than stopping classification in

the presence of a visual occlusion, the upper facial expression classification system could be deployed to assess a person’s state using upper facial region features – similar to how we as humans have trained ourselves to analyse how a person is feeling when communicating with PPE and face masks.

When comparing the performances of the classifiers designed for this research with those that have been designed for SAM, we see that classification has been improved, particularly when comparing full facial expression classifiers in Figures 29 and 87. This could be related to the use of the RAVDESS dataset for training SAM’s models or it could be related to the implementation of SAM’s hierarchical classifier ensemble, which shows that by clustering similar expressions and splitting a 21-state classifier into a combination of binary and ternary classifiers, performances can be improved.

An argument could therefore be made that by applying the hierarchical classifier ensemble architecture for upper facial expression classification, one could theoretically improve the classification performances and implement the upper-facial micro-expression as part of a real-time system architecture – like SAM. This would be a functional improvement that would be explored in future work and future iterations of SAM.

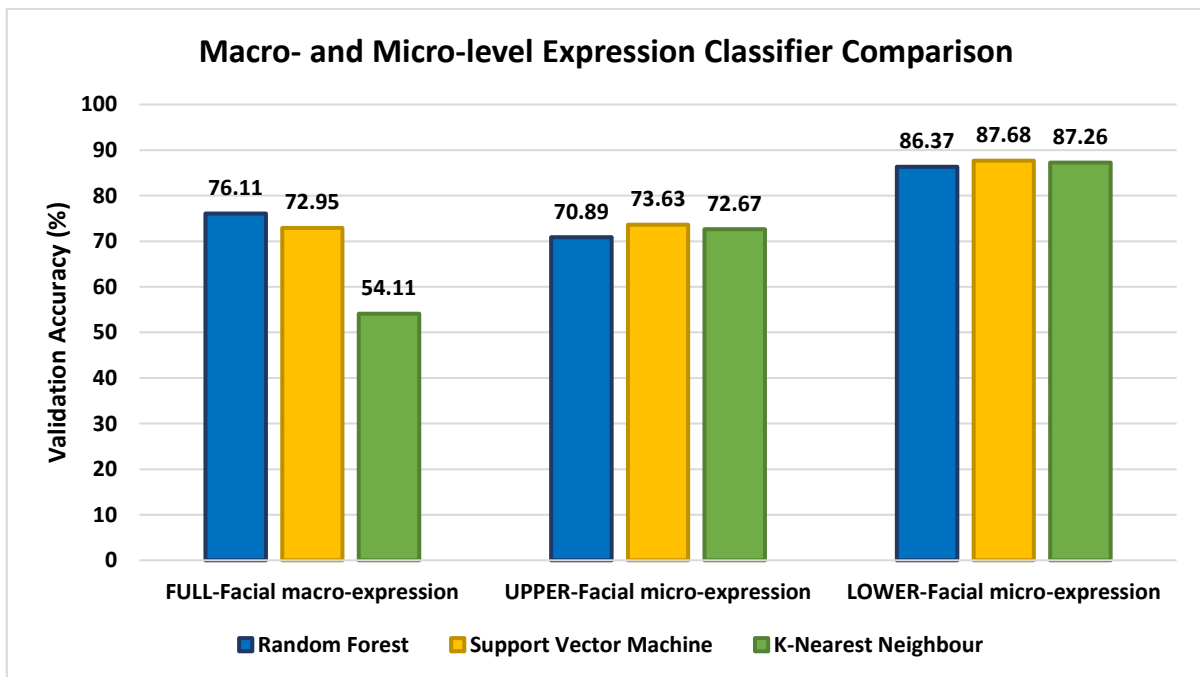


Figure 87: Comparison of classifier performances and validation accuracy when analysing full facial expressions and upper- and lower-facial expressions.

Chapter 7: Conclusion

During the inception of this project, I posed the question: “How can I improve the way we perceive AI systems and interact with machines in general?”. While on the surface, this could be considered as a blanket statement, the seed was planted and from it grew the Accountable Explainable Artificial Intelligence (AXAI) framework, the State Assessment Module (SAM), and all other contributions that have been discussed in this work. In fact, the level in which this research has answered that question is greater than what was initially conceived.

This thesis has explored many facets of engineering, software design, development, artificial intelligence, affective state assessment and human- and machine-centred decision-making. Largely through the AXAI framework and SAM, this work has contributed a wealth of knowledge to various fields of literature.

There are several gaps in research corresponding to general machine learning/AI that stem from the major issue of “black-box” approaches and trust in AI systems for the general public. While explainable AI (XAI) has been posed as a solution, the fact remains that existing XAI solutions still struggle to address all of the issues that posed by the general population. A common fault is that black-box solutions lack explanatory reasoning and explainability thus highlighting a considerable gap that researchers need to bridge. Furthermore, with the inherent distrust and lack of reasoning, comes a need for quantitative assessment tools and regulation and legislation for AI systems.

In extension to general machine learning and AI, there are various research gaps associated with affective state assessment, and ways that these systems can be improved going forward. A major research gap is that there is a lack of real-time AXAI and XAI-capable ASA system. While ASA systems have been deployed using facial expression and speech analysis, these systems are generally statistical classifiers and do not possess XAI or AXAI capabilities. Considering some of the potential and existing applications of ASA systems, there is a need for more accountable, transparent solutions, especially considering that human emotions and feelings are being used for decision-making processes.

There is also a lack of dynamic continuous assessment solutions, which is a considerable issue when we account for the fact that human expressions are by nature – dynamic and continuous. By deploying discrete solutions, ASA systems fail to account for minor changes in features, temporal considerations, and micro-expressions. Through a literature survey it was also found that blind source separation has not been applied in the realm of ASA and there is a lack of research in which this common signal analysis technique has been applied in this domain. Finally, in light of the COVID-19 pandemic and the rise in the use of face masks and PPE, many existing ASA and facial expression recognition would be rendered obsolete by not accounting for the presence of visual occlusions, a research gap that would need to be addressed if we were to design adaptive systems rather than static ones.

This work addresses the machine learning and AI-related research gaps through the AXAI framework and proves that transparent, explainable systems are possible through the deployment of SAM. The need for interpretable models and explainable systems has been an emerging discussion point in political, legal, academic, and professional sectors. As technology continues to advance, there needs to be systems in place that are capable of assessing and delineating AI and ML systems on the basis of their comprehensibility, accountability, and predictive accuracy. The novel, AXAI framework achieves this, and serves as a viable tool for assessing and comparing AI systems as reported in [107, 108].

Introducing accountability as an assessable element throughout the design process is a novel addition when compared to other XAI assessment solutions as these tend to define accountability assessment as a post-production task. Exploiting a comprehensive tool like the AXAI framework throughout the development cycle would improve the way AI and ML developers design systems, such that they could be optimised to improve the end-user experience and improve human-machine interactions.

Through the design and development of SAM, this work has shown that using the AXAI framework to improve human-machine interaction is possible. SAM is an embedded, real-time, ASA system capable of classifying over 100 unique states across facial expression and paralinguistic fields. All of the ASA-related gaps introduced in this work have been addressed through the development of SAM. The state assessment module exploits a hybrid learning model and a hierarchical classifier ensemble to achieve dynamic, continuous assessment across speech and facial expression modalities.

This work has evidenced that SAM is highly accurate, boasting average classification accuracies of 83.07% and 95.90% for facial expression and speech classifiers respectively. SAM is equipped with an intuitive, powerful GUI and a range of reporting, interaction and communication functions that allow it to explain its reasoning and decision-making processes to the user. An occlusion detection function that exploits a binary classifier to detect if a person for example is wearing a face mask is also embedded as part of SAM's classification architecture. This work has also shown that it is possible to detect facial expression using the upper facial region in the event that an occlusion has been detected.

Designing SAM as a modular, embedded system was a calculated decision. Robotic, autonomous, and complex hardware solutions exist all throughout literature and in the modern world. Therefore, designing SAM as a 'robot' with kinematic elements would not be a novel addition. I believe that SAM's portability makes it more powerful, as it is capable of being equipped to any existing robotic and surveillance system that is equipped with the necessary peripheral devices including tools like drones as shown in Chapter 5. The ability to equip any robotic system with advanced emotional intelligence and affective state awareness adds to SAM's novelty.

Furthermore, this work has also shown the capabilities and benefits of incorporating blind source separation for group sentiment analysis. By separating speech signal mixtures and classifying them, it is proven that BSS can be an effective tool in the realm of affective computing and as discussed, many times prior, there is *little-to-no* evidence of BSS being applied in such a way, highlighting the novelty of the system introduced in this thesis, and the potential applications of such a system in security, entertainment, legal and commercial sectors.

Summarising the structure of this thesis, Chapter 1 introduced the research and identified the main objectives, research gaps and significant contributions. Chapter 2 served as a literature review, discussing the background information and theory relevant to machine and deep learning, explainable AI, and affective computing. Chapter 3 introduced the AXAI framework and detailed the vectors and sub-components used to describe comprehensibility, accountability, and predictive accuracy metrics. Chapter 4 discussed software design and development principles and how the AXAI framework could be incorporated into the design of an AI system like SAM.

Chapter 5 discussed the design and development of SAM from the ground-up, discussing front-end and back-end design choices and flowcharts and explaining the overall functionality and performance of the system. Chapter 5 also discussed SAM's interactive elements and its deployment as a modular,

embedded system, proving how it could be deployed into existing robotic systems and infrastructures. Chapter 6 then discusses some of the parallel research efforts including blind source separation for group sentiment analysis and a dynamic rule-based system capable of upper- and lower-facial micro-expression detection.

Going forward, it is important to emphasise the importance of a tool like the AXAI framework and how researchers could draw from this research to improve AI system design and development processes. By incorporating elements of comprehensibility and accountability in AI system design would make systems more user-friendly and would enhance human-machine interactions in general.

Regarding future work, there is an opportunity to improve the hardware that SAM exists on. As there are many single-board computers that exist on the market, it would be a wise task to experiment with performance speeds on different computers and development boards. In terms of functional improvements, as shown by the iterative approach reported in this thesis, there is always room for improvement and opportunities to advance the capabilities of SAM. Potentially, SAM could be equipped to have a contextual understanding of a person's previous states (exploiting working memory and stored past interactions). This contextual knowledge would give SAM a more comprehensive understanding of who it is interacting with, and historical contexts could allow for an improvement in the way SAM interacts with users and subjects and vice-versa.

As we look to improve the way we communicate and interact with machines, there is the potential for SAM to synthesise emotions and respond to a person based on their state. However, this addition would be very complex and requires a contextual understanding of various internal and external factors. But if completed, this would bring us closer to developing systems that are *truly* emotionally intelligent as they would have intelligence w.r.t to inputs and outputs. This could also lead to the addition of more commands as while SAM is capable of 21 commands, there is room for more.

Transparency, accountability, and comprehensibility are necessary attributes that machine learning and AI systems must possess moving forward. Ultimately, the AXAI framework may not be the peak, but if it opens the door for further assessment tools and continues the discussion, then it has achieved its purpose. Regarding SAM, it has proven that systems can be designed with AXAI elements at the forefront and shows that the continuous, dynamic nature of human expressions can be accounted for in an ASA system.

Improving human-machine interactions is no easy feat, and there are various considerations that need to be accounted for at a higher-level with tools like the AXAI framework and at a lower-level, with tools like SAM that account for and understand changes in affective states, emotions, and feelings. Humans are complex creatures, and our emotions drive a lot of our decision-making processes, so it is important that we design systems that understand what makes us innately human.

References

- [1] J. Vice, "Bi-modal Affect-Based Authentication Machines", B.E thesis, Dept. Civil & Mech. Eng., Curtin Univ., Perth, Australia, 2019.
- [2] K. Siau and W. Wang, "Building trust in artificial intelligence, machine learning, and robotics," *Cutter Bus. Technol. J.*, vol. 31, no. 2, pp.47-53, Mar. 2018.
- [3] D.A.R.P. Agency, "Broad Agency Announcement: Explainable Artificial Intelligence (XAI)," Arlington, VA, USA, DARPA-BAA-16-53, Aug. 10, 2016. Accessed: Jun. 14, 2022. [Online]. Available: <https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf>.
- [4] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, Sep. 2018, doi: 10.1109/ACCESS.2018.2870052.
- [5] A.B. Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82-115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.
- [6] H.-W. Liu, C.-F. Lin and Y.-J. Chen, "Beyond State v Loomis: artificial intelligence, government algorithmization and accountability," *Int. J. of Law and Inf. Technol.*, vol. 27, no. 2, pp. 122-141, Feb. 2019, doi: 10.1093/ijlit/eaz001.
- [7] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1-38, Feb. 2019, doi: 10.1016/j.artint.2018.07.007.
- [8] D. Gunning and D. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Mag.*, vol. 40, no. 2, pp. 44-58, Jun. 2019, doi: 10.1609/aimag.v40i2.2850.
- [9] H. Hagras, "Toward human-understandable, explainable AI," *Computer*, vol. 51, no. 9, pp. 28-36, Sep. 2018, doi: 10.1109/MC.2018.3620965.
- [10] J. Fitzsimmons, "Information technology and the third industrial revolution," *The Electron. Library*, vol. 12, no. 5, pp. 295-297, May 1994, doi: 10.1108/eb045307.
- [11] S.H. Muggleton, U. Schmid, C. Zeller, A. Tamaddoni-Nezhad and T. Besold, "Ultra-Strong Machine Learning: comprehensibility of programs learned with ILP," *Mach. Learn.*, vol. 107, no. 7, pp. 1119-1140, May 2018, doi: 10.1007/s10994-018-5707-3.
- [12] D. Doran, S. Schulz and T.R. Besold, "What does explainable ai really mean? A new conceptualization of perspectives," *arXiv preprint*, Oct. 2017, doi: 10.48550/arXiv.1710.00794.
- [13] D. Michie, "Machine Learning in the next Five Years," in *Proc. 3rd Eur. Conf. on Eur. Working Session on Learn*, Glasgow, UK, 1988, pp. 107-122.
- [14] N. Bostrom, "Ethical issues in advanced artificial intelligence," in *Machine Ethics and Robot Ethics*, W. Wallach and P. Asaro, Eds. New York, NY, USA: Routledge, 2020, pp. 69-75.
- [15] B. Kim and F. Doshi-Velez, "Machine Learning Techniques for Accountability," *AI Mag.*, vol. 42, no. 1, pp. 47-52. Apr. 2021.
- [16] C.S. Peirce, "Deduction, induction, and hypothesis," *Popular Science Monthly*, vol. 13, pp. 470-482, 1878.
- [17] M. MacCarthy, "An Examination of the Algorithmic Accountability Act of 2019," Transatlantic Working Group, Oct. 24, 2019. Accessed: Jun. 14, 2022. [Online] Available: <https://ssrn.com/abstract=3615731>.

- [18] F. Gualdi and A. Cordella, "Artificial intelligence and decision-making: The question of accountability," in *Proc. 54th Hawaii Int. Conf. on System Sciences*, Hawaii, USA, 2021, pp. 2297-2306.
- [19] J. Vice, M. Khan and S. Yanushkevich, "Multimodal Models for Contextual Affect Assessment in Real-time," in *2019 IEEE 1st Int. Conf. on Cogn. Mach. Intell. (CogMI)*, Los Angeles, CA, USA, Dec. 2019, pp. 87-92.
- [20] D. Cernea and A. Kerren, "A survey of technologies on the rise for emotion-enhanced interaction," *J. of Vis. Languages & Comput.*, vol. 31, no. A, pp. 70-86, Dec. 2015, doi: 10.1016/j.jvlc.2015.10.001.
- [21] C.-H. Wu, J.-C. Lin and W.-L. Wei, "Survey on audiovisual emotion recognition: databases, features, and data fusion strategies," *APSIPA Trans. on Signal and Inf. Process.*, vol. 3, no. 12, pp. 1-18, Nov. 2014, doi: 10.1017/ATSIP.2014.11.
- [22] S. Zepf, J. Hernandez, A. Schmitt, W. Minker and R.W. Picard, "Driver emotion recognition for intelligent vehicles: A survey," *ACM Comput. Surv. (CSUR)*, vol. 53, no. 3, pp. 1-30, May 2021, doi: 10.1145/3388790.
- [23] J. Zhang, Z. Yin, P. Chen and S. Nichele, "Emotion recognition using multi-modal data and machine learning techniques: A tutorial and review," *Inf. Fusion*, vol. 59, pp. 103-126, Jul. 2020, doi: 10.1016/j.inffus.2020.01.011.
- [24] J. Vice, M. M. Khan, T. Tan and S. Yanushkevich, "Dynamic Hybrid Learning for Improving Facial Expression Classifier Reliability," in *2022 IEEE Int. Conf. on Evolving and Adaptive Intell. Syst. (EAIS)*, Larnaca, Cyprus, May 2022, pp. 1-8.
- [25] J. Vice, M. M. Khan, I. Murray and S. Yanushkevich, "Adaptive Classification of Occluded Facial Expressions of Affective States," in *2022 IEEE Int. Conf. on Evolving and Adaptive Intell. Syst. (EAIS)*, Larnaca, Cyprus, May 2022, pp. 1-8.
- [26] L.Y. Lee et al., "Practice and technique of using face mask amongst adults in the community: a cross-sectional descriptive study," *BMC Public Health*, vol. 20, no. 1, pp. 1-11, Jun. 2020, doi: 10.1186/s12889-020-09087-5.
- [27] S. Feng, C. Shen, N. Xia, W. Song, M. Fan and B.J. Cowling, "Rational use of face masks in the COVID-19 pandemic," *The Lancet Respiratory Medicine*, vol. 8, no. 5, pp. 434-436, May 2020, 10.1016/S2213-2600(20)30134-X.
- [28] M.D. Plumbley, S.A. Abdallah, J.P. Bello, M.E. Davies, G. Monti and M.B. Sandler, "Automatic music transcription and audio source separation," *Cybern. & Syst.*, vol. 33, no. 6, pp. 603-627, Nov. 2010, doi: 10.1080/01969720290040777.
- [29] P. Vanroose, "Blind source separation of speech and background music for improved speech recognition," in *The 24th Symp. on Inf. Theory*, Yokohama, Japan, Jun. 2003, pp. 103-108.
- [30] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015, doi: 10.1038/nature14539.
- [31] M. H. Hassoun, Ed., *Fundamentals of artificial neural networks*, Cambridge, MA, USA: MI press, 1995.
- [32] S.C. Wang, Ed., "Artificial neural network," in *Interdisciplinary Computing in Java Programming Language*. Boston, MA, USA: Springer, 2003, ch. 5, pp. 81-100.
- [33] S.B. Kotsiantis, I. Zaharakis and P. Pintelas, "Supervised Machine Learning: A review of classification techniques," in *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1. Amsterdam, Netherlands: IOS Press, 2007, pp. 3-24.

- [34] M.I. Jordan and T.M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255-260, Jul. 2015, doi: 10.1126/science.aaa8415.
- [35] R.O. Duda, P.E. Hart and D.G. Stork, Eds., "Unsupervised Learning and Clustering," in *Pattern Classification*. New York, NY, USA: John Wiley & Sons, 2001, ch. 10, pp. 517-601.
- [36] Z. Ghahramani, "Unsupervised learning," in *Advanced Lectures on Machine Learning*, vol. 3176, Berlin, Germany: Springer, 2003, pp. 72-112.
- [37] P. Comon and C. Jutten, Eds., *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Burlington, MA, USA: Academic Press, 2010.
- [38] A. Hyvärinen, "Survey on Independent Component Analysis," *Neural Computing Surveys*, vol. 45, pp. 94-128, 1999.
- [39] A. Hyvärinen and O. Erkki, "Independent Component Analysis: Algorithms and Applications," *Neural networks*, vol. 13, no. 4-5, pp. 411-430, Jun. 2000, doi: 10.1016/S0893-6080(00)00026-5.
- [40] A. Hyvärinen and P. Hoyer, "Emergence of phase-and shift-invariant features by decomposition of natural images into independent feature subspaces," *Neural Comput.*, vol. 12, no. 7, pp.1705-1720, Jul. 2000, doi: 10.1162/089976600300015312.
- [41] R. Li, H. Li and F. Wang, "Dependent Component Analysis: Concepts and Main Algorithms," *J. Comput.*, vol. 5, no. 4, pp.589-597, Apr. 2010, doi: 10.4304/jcp.5.4.589-597.
- [42] N. Kingsbury, "Shift invariant properties of the dual-tree complex wavelet transform," in *Proc. of the IEEE Int. Conf. on Acoust., Speech, and Signal Process.*, vol. 3, pp. 1221-1224, 1999, doi: 10.1006/acha.2000.0343.
- [43] N. Kingsbury, "A dual-tree complex wavelet transform with improved orthogonality and symmetry properties," in *Proc. of the 2000 Int. Conf. on Image Process.*, vol. 2, Mar. 2000, pp. 375-378, doi: 10.1109/ICIP.2000.899397.
- [44] I.W. Selesnick, R.G. Baraniuk and N. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 123-151, Nov. 2005, doi: 10.1109/MSP.2005.1550194.
- [45] R. Xu and D.C. Wunsch, *Clustering*. D.B. Fogel, Ed., Hoboken, NJ, USA: John Wiley & Sons, 2008.
- [46] A.K. Jain, M.N. Murty and P.J. Flynn, "Data clustering: a review," in *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264-323, Sep. 1999, doi: 10.1145/331499.331504.
- [47] L. Batina, B. Gierlichs, E. Prouff, M. Rivain, F.X. Standaert and N. Veyrat-Charvillon, "Mutual information analysis: a comprehensive study," *J. of Cryptology*, vol. 24, no. 2, pp.269-291, Apr. 2011, doi: 10.1007/s00145-010-9084-8.
- [48] P.C. Sen, M. Hajra and M. Ghosh, "Supervised classification algorithms in machine learning: A survey and review," in *Emerging Technology in Modelling and Graphics*, vol 937, J. Mandal, D. Bhattacharya, Eds., Singapore: Springer, 2020, pp. 99-111.
- [49] M. Kuhn and K. Johnson, Eds., "Discriminant Analysis and Other Linear Classification Models," in *Applied predictive modelling*, vol. 26. New York, NY, USA: Springer, 2013, ch. 12, pp. 275-326.
- [50] J.A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," in *Neural Process. Lett.*, vol. 9, no. 3, pp. 293-300, Jun. 1999, doi: 10.1023/A:1018628609742.
- [51] S.B. Kotsiantis, I. Zaharakis and P. Pintelas, "Supervised Machine Learning: A review of classification techniques," in *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, Amsterdam, Netherlands: IOS Press, 2007, pp. 3-24.

- [52] M.W. Gardner and S.R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32 no. 14-15, pp.2627-2636, Aug. 1998, doi: 10.1016/S1352-2310(97)00447-0.
- [53] M. Minsky and S. Papert, *Perceptrons*. Cambridge, MA, USA: MIT Press, 1969.
- [54] K. Hornik, M. Stinchcombe and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp.359-366, 1989, doi: 10.1016/0893-6080(89)90020-8.
- [55] S.C. Wang, Ed., "Artificial neural network," in *Interdisciplinary Computing in Java Programming Language*, Boston, MA, USA: Springer, 2003, ch. 5, pp. 81-100.
- [56] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1-14, Apr. 1995.
- [57] S. Albawi, T.A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 Int. Conf. on Eng. and Technol. (ICET)*, Antalya, Turkey, Aug. 2017 pp. 1-6.
- [58] F. Chollet et al., *Keras*. (2015). [Online]. Available: <https://github.com/fchollet/keras>.
- [59] A. Krizhevsky, I. Sutskever and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. of the ACM*, vol. 60, no. 6, pp. 84-90, May 2012, pp. 1097-1105, doi: 10.1145/3065386.
- [60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint*, Dec. 2014, doi: 10.48550/arXiv.1409.1556.
- [61] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit. (CVPR)*, Las Vegas, NV, Jun. 2016, pp. 2818-2826.
- [62] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit. (CVPR)*, Honolulu, HI, Jul. 2017, pp. 1800-1807.
- [63] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognit. (CVPR)*, Las Vegas, NV, Jun. 2016, pp. 770-778.
- [64] S. Parthasarathy and C. Busso, "Ladder networks for emotion recognition: Using unsupervised auxiliary tasks to improve predictions of emotional attributes," *arXiv preprint*, Apr. 2018, doi: 10.48550/arXiv.1804.10816.
- [65] Mustaqeem, M. Sajjad and S. Kwon, "Clustering-Based Speech Emotion Recognition by Incorporating Learned Features and Deep BiLSTM," in *IEEE Access*, vol. 8, pp. 79861-79875, Apr. 2020, doi: 10.1109/ACCESS.2020.2990405.
- [66] X. Li, Y. Rao, H. Xie, R.Y.K Lau, J. Yin and F.L. Wang, "Bootstrapping social emotion classification with semantically rich hybrid neural networks," *IEEE Trans. on Affective Comput.*, vol. 8, no. 4, pp.428-442, Jun. 2017, doi: 10.1109/TAFFC.2017.2716930.
- [67] Oxford University Press. "Robot." Oxford Advanced American Dictionary. https://www.oxfordlearnersdictionaries.com/definition/american_english/robot (accessed Jun. 14, 2022).
- [68] E. Escott. "Bots vs. Robots: what's the difference? (with examples)", Codebots. <https://codebots.com/artificial-intelligence/robots-and-bots-explained> (accessed Jun. 14, 2022).

- [69] C.M. Jones and M. Jonsson, "Performance analysis of acoustic emotion recognition for in-car conversational interfaces," in *Universal Access in Human-Computer Interaction*, C. Stephanidis, Eds., Berlin, Germany: Springer, 2007, pp. 411-420.
- [70] D. Wang, Q. Yang, A. Abdul and B.Y. Lim, "Designing theory-driven user-centric explainable AI," in *Proc. of the 2019 CHI Conf. on Human Factors in Comput. Syst.*, May 2019, pp. 1-15.
- [71] S.M. Lundberg et al., "From Local Explanations to Global Understanding with Explainable AI for Trees," in *Nature Machine Intell.*, vol. 2, no. 1, pp. 56-67, Jan. 2020, doi: 10.1038/s42256-019-0138-9.
- [72] N. Burkart and M.F. Huber, "A survey on the explainability of supervised machine learning," *J. of Artif. Intell. Res.*, vol. 70, pp.245-317, Jan. 2021, doi: 10.1613/jair.1.12228.
- [73] A. Moncada-Torres, M.C. van Maaren, M.P. Hendriks, S. Siesling and G. Geleijnse, "Explainable machine learning can outperform Cox regression predictions and provide insights in breast cancer survival," *Scientific Reports*, vol. 11, no. 1, pp.1-13, Mar. 2021, doi: 10.1038/s41598-021-86327-7.
- [74] S.M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Adv. Neural Inf. Process. Syst.*, vol. 20 pp. 4765-4774, 2017.
- [75] J. Li, H. Shi, and K.S. Hwang, "An explainable ensemble feedforward method with Gaussian convolutional filter," *Knowledge-Based Syst.*, vol. 225, pp. 1-11, Aug. 2021, doi: 10.1016/j.knosys. 2021.107103.
- [76] V. Buhrmester, D. Münch and M. Arens, "Analysis of explainers of black box deep neural networks for computer vision: A survey," *Mach. Learn. and Knowl. Extraction*, vol. 3, no. 4, pp.966-989, Dec. 2021, doi: 10.3390/make3040048.
- [77] B. Dayma et al., "DALL-E mini." HuggingFace.com. <https://huggingface.co/spaces/dalle-mini/dalle-mini> (accessed Sep. 29, 2022).
- [78] A. Datta Chaudhuri, S.K. Biswas, M. Chakraborty and S. Sarkar, "A transparent rule-based expert system using neural network," *Soft Comput.*, vol. 25, no. 12, pp.7731-7744, Jan. 2021, doi: 10.1007/s00500-020-05547-7.
- [79] S. Wickramanayake, W. Hsu and M.L. Lee, "Comprehensible Convolutional Neural Networks via Guided Concept Learning," in *2021 Int. Joint Conf. on Neural Networks (IJCNN)*, Shenzhen, China, Jul. 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9534269.
- [80] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*, vol. 6, no. 3-4, pp. 169-200, May 1992, doi: 10.1080/02699939208411068.
- [81] J. Posner, J.A. Russell and B.S. Peterson, "The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology," *Develop. and Psychopathology*, vol. 17, no. 3, pp. 715-734, Nov. 2005, doi: 10.1017/S0954579405050340.
- [82] J. Russell and A. Mehrabian, "Evidence for a Three-Factor Theory of Emotions," *J. of Res. in Personality*, vol. 11, no. 3, pp. 273-294, Sep. 1977, doi: 10.1016/0092-6566(77)90037-X.
- [83] R. Plutchik, "Chapter 1 - A General Psychoevolutionary Theory of Emotion," in *Theories of Emotion*, R. Plutchik and H. Kellerman, Eds., New York, NY, USA: Academic Press, 1980, ch. 1, pp. 3-33.
- [84] M. Donaldson. "Plutchik's Wheel of Emotions." 6seconds.org. <https://www.6seconds.org/2020/08/11/plutchik-wheel-emotions/> (accessed Jun. 14. 2022).

- [85] P. Ekman, "Facial expression and emotion," *American Psychologist*, vol. 48 no. 4, pp. 384-392, Apr. 1993, doi: 10.1037/0003-066X.48.4.384.
- [86] P. Ekman, E.L. Rosenberg, Eds., *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, 3rd ed. New York, NY, USA: Oxford University Press, 2020.
- [87] C.-N. Anagnostopoulos, T. Iliou and I. Giannoukos, "Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011," *Artificial Intell. Rev.*, vol. 43, no. 2, pp. 155-177, Feb. 2015, doi: 10.1007/s10462-012-9368-5.
- [88] B. Schuller et al. "The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language," in *17th Annu. Conf. of the Int. Speech Commun. Assoc. (Interspeech 2016)*, San Francisco, CA, USA, Sep. 2016, pp. 2001-2005.
- [89] F. Eyben et al. "The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing," *IEEE Trans. on Affective Comput.*, vol. 7, no. 2, pp.190-202, Apr. 2015, doi: 10.1109/TAFFC.2015.2457417.
- [90] F. Eyben, M. Wöllmer and B. Schuller, "openSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor," in *Proc. ACM Multimedia (MM)*, Florence, Italy, Oct. 2010, pp. 1459-1462, doi: 10.1145/1873951.1874246.
- [91] I.R. Murray and J.L. Arnott, 1993, "Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion," *The J. of the Acoustical Soc. of America*, vol. 93, no. 2, pp. 1097-1108, Jun. 1998, doi: 10.1121/1.405558.
- [92] R. Banse and K.R. Scherer, "Acoustic profiles in vocal emotion expression," *J. of Personality and Social Psychol.*, vol. 70, no. 3, pp. 614-636, 1996, doi: 10.1037/0022-3514.70.3.614.
- [93] R. van Bezooijen, *Characteristics and recognizability of vocal expressions of emotion*. Dordrecht, Netherlands: De Gruyter Mouton, 1984.
- [94] I. Shahin, A.B. Nassif and S. Hamsa, "Emotion Recognition Using Hybrid Gaussian Mixture Model and Deep Neural Network," *IEEE Access*, vol. 7, pp. 26777-26787, Feb. 2019, doi: 10.1109/ACCESS.2019.2901352.
- [95] R.B. Lanjewar, S. Mathurkar and N. Patel, "Implementation and comparison of speech emotion recognition system using gaussian mixture model (gmm) and k-nearest neighbor (knn) techniques," *Procedia Comput. Sci.*, vol. 49, pp. 50-57, Jan. 2015, doi: 10.1016/j.procs.2015.04.226.
- [96] X. Cheng and Q. Duan, "Speech emotion recognition using gaussian mixture model," in *Proc. of the 2012 Int. Conf. on Comput. Appl. and Syst. Modelling*, Cochin, India, Oct. 2012, pp. 1222-1225, doi: 10.2991/iccasm.2012.311.
- [97] M. You, C. Chen, J. Bu, J. Liu and J. Tao, "Manifolds based emotion recognition in speech," *Int. J. of Comput. Linguistics & Chinese Lang. Process.*, vol. 12, no. 1, pp. 49-64, Mar. 2007.
- [98] G. Paraskevopoulos, E. Tzinis, N. Ellinas, T. Giannakopoulos and A. Potamianos, "Unsupervised Low-Rank Representations for Speech Emotion Recognition," in *Proc. of the 20th Annu. Conf. of the Int. Speech Commun. Assoc.*, Graz, Austria, Sep. 2019, pp. 939-943, doi: 10.21437/Interspeech.2019-2769.
- [99] S. Zhang, X. Zhao and B. Lei, "Speech emotion recognition using an enhanced kernel Isomap for human-robot interaction," *Int. J. of Adv. Robot. Syst.*, vol. 10, no. 2, pp. 114-121, May 2017, doi: 10.5772/55403.

- [100] R. Jiang, A.T. Ho, I. Cheheb, N. Al-Maadeed, S. Al-Maadeed and A. Bouridane, "Emotion recognition from scrambled facial images via many graph embedding," in *Pattern Recognit.*, vol. 67, pp. 245-251, Jul. 2017, doi: 10.1016/j.patcog.2017.02.003.
- [101] C. Dongwei, W. Fang, W. Zhen, L. Haifang and C. Junjie, "EEG-based emotion recognition with brain network using independent components analysis and granger causality," in *Proc. of the 2013 Int. Conf. on Comput. Medical Appl. (ICCMMA)*, Sousse, Tunisia, Jan. 2013, pp. 1-6, doi: 10.1109/ICCMMA.2013.6506157.
- [102] A.D. Bigirimana, N. Siddique and D. Coyle, "A hybrid ICA-wavelet transform for automated artefact removal in EEG-based emotion recognition," in *IEEE Int. Conf. on Syst., Man, and Cybern. (SMC)*, Budapest, Hungary, Oct. 2016, pp. 4429-4434, doi: 10.1109/SMC.2016.7844928.
- [103] C. Laugs, H.V. Koops, D. Odijk, H. Kaya and A. Volk, "The Influence of Blind Source Separation on Mixed Audio Speech and Music Emotion Recognition," *Companion Publication of the 2020 Int. Conf. on Multimodal Interact.*, pp. 67-71, Oct. 2020, doi: 10.1145/3395035.3425252.
- [104] J. Deng, Z. Zhang, F. Eyben, and B. Schuller, "Autoencoder-based unsupervised domain adaptation for speech emotion recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1068-1072, May 2014, doi: 10.1109/LSP.2014.2324759.
- [105] S. Parthasarathy and C. Busso, "Ladder networks for emotion recognition: Using unsupervised auxiliary tasks to improve predictions of emotional attributes," *arXiv preprint*, Apr. 2018, doi: 10.48550/arXiv.1804.10816.
- [106] M. Berkane, K. Belhouchette and H. Belhadef, "Emotion recognition approach using multilayer perceptron network and motion estimation," *Int. J. of Synthetic Emotions (IJSE)*, vol. 10, no. 1, pp. 38-53, Jan. 2019, doi: 10.4018/IJSE.2019010102.
- [107] M.M. Khan and J. Vice, "Toward Accountable and Explainable Artificial Intelligence Part One: Theory and Examples," *IEEE Access*, vol. 10, pp. 99686-99701, 2022, doi: 10.1109/ACCESS.2022.3207812.
- [108] J. Vice and M. Khan, "Toward Accountable and Explainable Artificial Intelligence Part Two: The Framework Implementation," *IEEE Access*, vol. 10, pp. 36091-36105, Mar. 2022, doi: 10.1109/ACCESS.2022.3163523.
- [109] J.A. Kroll and E.W. Felten, "Accountable Algorithms," Ph.D. Thesis, Dept. of Comput. Sci., Princeton Univ., Princeton, NJ, USA, 2015.
- [110] M. Bovens, "Analysing and Assessing Accountability: A Conceptual Framework," *Eur. Law J.*, vol. 13, no. 4, pp. 447-468, 2007, doi: 10.1111/j.1468-0386.2007.00378.x.
- [111] H.F. Davis and A.D. Snider, *Introduction to vector analysis*. Charlottesville, VA, USA: Wm. C. Brown, 1995.
- [112] D. Michie, "Learning concepts from data," *Expert Syst. with Appl.*, vol. 15, no. 3-4, pp. 193-204, Oct. 1998, doi: 10.1016/S0957-4174(98)00044-X.
- [113] U. Schmid, C. Zeller, T. Besold, A. Tamaddoni-Nezhad and S. Muggleton, "How does predicate invention affect human comprehensibility?," in *Proc. 26th Int. Conf. on Log. Program.*, London, UK, 2017, pp. 52-67, doi: 10.1007/978-3-319-63342-8_5.
- [114] S. Mohseni, N. Zarei and E.D. Ragan, "A multidisciplinary survey and framework for design and evaluation of explainable AI systems," *ACM Trans. on Interactive Intell. Syst.*, vol. 11, no.3-4, pp. 1-45, Sep. 2021, doi: 10.1145/3387166.

- [115] X. Naidenova, *Machine learning methods for commonsense reasoning processes: Interactive models*. Hershey, NY, USA: IGI Global, 2009.
- [116] S. Poria, A. Gelbukh, E. Cambria, A. Hussain and G.-B. Huang, "EmoSenticSpace: A novel framework for affective commonsense reasoning," *Knowledge-Based Syst.*, vol. 69, pp. 108-123, Oct. 2014, doi: 10.1016/j.knosys.2014.06.011.
- [117] E. T. Mueller, *Commonsense reasoning: An event calculus based approach*, S. Elliot, Ed. Waltham, MA, USA: Elsevier Science, 2014.
- [118] Ada Lovelace Institute, AI Now Institute, and Open Government Partnership, "Algorithmic accountability for the public sector Learning from the first wave of policy implementation." Ada Lovelace Institute. <https://www.adalovelaceinstitute.org/report/algorithmic-accountability-public-sector/> (accessed Oct. 3, 2022).
- [119] M.M. Khan, R.D. Ward and M. Ingleby, "Toward Use of Facial Thermal Features in Dynamic Assessment of Affect and Arousal Level," *IEEE Trans. on Affective Comput.*, vol. 8, no. 3, pp. 412-425, Jul. 2017, doi: 10.1109/TAFFC.2016.2535291.
- [120] J. MacKay, "Software Development Process: How to Pick The Process That's Right For You." Planio.com. <https://plan.io/blog/software-development-process/#1-waterfall> (accessed Jun. 14. 2022).
- [121] S. Shylesh, "A study of software development life cycle process models," in *Nat. Conf. on Reinventing Opportunities in Manage., IT, and Social Sci.*, Mangaluru, India, Jun. 2017, pp. 534-541.
- [122] R. Kaur and J. Sengupta, "Software process models and analysis on failure of software development projects," *arXiv preprint*, Jun. 2013, doi: 10.48550/arXiv.1306.1068.
- [123] K. Beck et al. "Manifesto for agile software development." agilemanifesto.org. <https://agilemanifesto.org/> (accessed Jun. 14, 2022).
- [124] S.R. Livingstone and F.A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PloS one*, vol. 13, no. 5, pp. 1-35, May 2018, doi: 10.1371/journal.pone.0196391.
- [125] P. Lucey, J.F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. 2010 IEEE Comput. Soc. Conf. on Comput. Vision and Pattern Recognit. - Workshops*, San Francisco, CA, USA, 2010, pp. 94-101, doi: 10.1109/CVPRW.2010.5543262.
- [126] P. Jackson and S. Haq, "Surrey audio-visual expressed emotion (savee) database." 2014. Distributed by University of Surrey: Guildford, UK. <http://kahlan.eps.surrey.ac.uk/savee/>.
- [127] K. Dupuis and M.K. Pichora-Fuller, "Toronto Emotional Speech Set." June 21, 2010. Distributed by University of Toronto. <https://tspace.library.utoronto.ca/handle/1807/24487>
- [128] H. Cao, D.G. Cooper, M.K. Keutmann, R.C. Gur, A. Nenkova and R. Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset," *IEEE Trans. on Affective Comput.*, vol. 5, no. 4, pp.377-390, Jul. 2014, doi: 10.1109/TAFFC.2014.2336244.
- [129] F. Burkhardt, A. Paeschke, M. Rolfes, W.F. Sendlmeier and B. Weiss, "A database of German emotional speech," in *Interspeech*, Lisbon, Portugal, Sep. 2005, pp. 1517-1520.
- [130] M. Valstar and M. Pantic, "Fully Automatic Facial Action Unit Detection and Temporal Analysis," in *2006 Conf. on Comput. Vision and Pattern Recognit. Workshop (CVPRW'06)*, New York, NY, USA, Jun. 2006, pp. 149-149, doi: 10.1109/CVPRW.2006.85.

- [131] Pedregosa et al., "Scikit-learn: Machine Learning in Python," *J. of Mach. Learn. Res.*, vol. 12, pp. 2825-2830, Nov. 2011.
- [132] F.T. Liu, K.M. Ting and Z.H. Zhou, "Isolation forest," in *8th IEEE Int. Conf. on Data Mining*, Pisa, Italy, Dec. 2008, pp. 413-422.
- [133] T. Hastie, R. Tibshirani and J.H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2, J.H. Friedman Ed. New York, Y, USA: Springer. 2009.
- [134] A. Cabani, K. Hammoudi, H. Benhabiles and M. Melkemi, "MaskedFace-Net—A dataset of correctly/incorrectly masked face images in the context of COVID-19," *Smart Health*, vol. 19, pp. 1-6, Mar. 2021, doi: 10.1016/j.smhl.2020.100144.
- [135] A. Zhang, *Speech Recognition (2017) [Software]*. Available: https://github.com/Uberi/speech_recognition#readme.
- [136] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. of Softw. Tools*, vol. 25, no. 11, pp. 120-123, 2000.
- [137] S. Lindsay, L. Rampterab and C. Curran, "Therapy through play: advancing the role of robotics in paediatric rehabilitation," in *Everyday technologies in healthcare*, New York, NY, USA: CRC Press. 2019, pp. 11-30.
- [138] H.I. Krebs et al. "Rehabilitation robotics: Performance-based progressive robot-assisted therapy," *Auton. Robots*, vol. 15, no. 1, pp.7-20, Jul. 2003, doi: 10.1023/A:1024494031121.
- [139] D. Feil-Seifer and M.J. Matarić, "Toward socially assistive robotics for augmenting interventions for children with autism spectrum disorders," in *Experimental Robot*. Berlin, Germany: Springer, 2009.
- [140] K. Wada, T. Shibata, T. Saito, K. Sakamoto and K. Tanie, "Psychological and social effects of one year robot assisted activity on elderly people at a health service facility for the aged," in *Proc. of the 2005 IEEE Int. Conf. on Robot. and Automat.*, Barcelona, Spain, Apr. 2005, pp. 2785-2790, doi: 10.1109/ROBOT.2005.1570535.
- [141] A. Baddeley, "Working memory," *Science*, vol. 255, no. 5044, pp.556-559, Jan. 1992, doi: 10.1126/science.1736359.
- [142] F. Burkhardt and J. Stegmann, "Emotional speech synthesis: Applications, history and possible future," *Studentexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung*, pp. 190-199, 2009.
- [143] J.W. Ratcliff and D.E. Metzener, "Pattern Matching: The Gestalt Approach," *Dr. Dobb's J.*, vol. 13, no. 7, pp. 46, July 1988.
- [144] C. Breazeal, K. Dautenhahn and T. Kanda, "Social robotics.," in *handbook of robotics*, Cham, Switzerland: Springer, 2016, ch. 72.
- [145] P. Fitzpatrick, K. Harada, C. Kemp, Y. Matsumoto, K. Yokoi and E. Yoshida, "Humanoids," *handbook of robotics*, Cham, Switzerland: Springer, 2016, ch. 76.
- [146] M. Mataric, B. Scassellati, "Socially Assistive Robotics," *handbook of robotics*, Cham, Switzerland: Springer, 2016, ch. 73.
- [147] *LattePanda V1.0 4G/64G Specifications*. LattePanda, Shangai. Accessed on: May 16, 2022. [Online]. Available: <http://www.lattepanda.com/products/3.html>.
- [148] *NVIDIA Jetson Nano: Technical Specifications*. NVIDIA, Santa Clara, CA. Accessed on: May. 16, 2022. [Online]. Available: <https://developer.nvidia.com/embedded/buy/jetson-nano-devkit#resources>.

- [149] X. Wu, C. Zhang and W. Du, "An analysis on the crisis of "chips shortage" in automobile industry-Based on the double influence of COVID-19 and trade friction," *J. of Phys.: Conf. Ser.*, vol. 1971, no. 1, pp. 1-6, Jul. 2021.
- [150] J. Voas, N. Kshetri and J.F. DeFranco, "Scarcity and Global Insecurity: The Semiconductor Shortage," *IT Professional*, vol. 23, no. 5, pp.78-82, Oct. 2021, doi: 10.1109/MITP.2021.3105248.
- [151] *Zero-X Edge Product Specifications*. Zero-X, Australia. Accessed on: May 16, 2022. [Online]. Available: <https://www.zero-x.com.au/product-edge>.
- [152] C. Sun et al., "A convolutional recurrent neural network with attention framework for speech separation in monaural recordings," *Scientific Reports*, vol. 11, no. 1, pp. 1-14, Jan. 2021, doi: 10.1038/s41598-020-80713-3.
- [153] E. Vincent, R. Gribonval and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 14, no. 4, pp. 1462-1469, Jul. 2006, doi: 10.1109/TSA.2005.858005.
- [154] A.A. Nugraha, A. Liutkus and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 24, no. 9, pp. 1652-1664, Jun. 2016, doi: 10.1109/TASLP.2016.2580946.
- [155] Z. Lv, B.B. Zhang, X.-P. Wu, C. Zhang and B.Y. Zhou, "A permutation algorithm based on dynamic time warping in speech frequency-domain blind source separation," *Speech Commun.*, vol. 92, pp. 132-141, Sep. 2017, doi: 10.1016/j.specom.2017.06.007.
- [156] N. Hassan, D.A. Ramli, "A comparative study of blind source separation for bioacoustics sounds based on FastICA, PCA and NMF," *Procedia Comput. Sci.*, vol. 126, pp. 363-372, 2018, doi: 10.1016/j.procs.2018.07.270.
- [157] T. Pfister, X. Li, G. Zhao and M. Pietikäinen, "Recognising spontaneous facial micro-expressions," in *2011 Int. Conf. on Comput. Vision*, Barcelona, Spain, Nov. 2011, pp. 1449-1456, doi: 10.1109/ICCV.2011.6126401.
- [158] F. Xu, J. Zhang and J.Z. Wang, "Microexpression Identification and Categorization Using a Facial Dynamics Map," *IEEE Trans. on Affective Comput.*, vol. 8, no. 2, pp. 254-267, Jun. 2017, doi: 10.1109/TAFFC.2016.2518162.
- [159] S. Polikovsky, Y. Kameda and Y. Ohta, "Facial micro-expression detection in hi-speed video based on facial action coding system (FACS)," *IEICE Trans. on Inf. and Syst.*, vol. 96, no. 1, pp. 81-92, Jan. 2013, doi: 10.1587/transinf.E96.D.81.
- [160] M.M. Khan, M. Ingleby and R.D. Ward, "Automated facial expression classification and affect interpretation using infrared measurement of facial skin temperature variations," in *ACM Trans. on Auton. and Adaptive Syst. (TAAS)*, vol. 1, no. 1, pp.91-113, Sep. 2006, doi: 10.1145/1152934.1152939. 2006.
- [161] J.N. Bassili, "Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face," *J. of Personality and Social Psychol.*, vol. 37, no. 11, pp. 2049-2058, 1979, doi: 10.1037/0022-3514.37.11.2049.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.

Appendices

Appendix A: SAM Version 1 Manifest File

Author: Jordan Vice
Last Edited: 28/01/22 -- 11AM

This version of the State Assessment Module was in a working condition as per 10:52AM 28/01/2022 (when this document was written)

The home-screen notebook and python files were used to execute the software and it was designed in a manner in which elements were modularised.

All GUI windows were capable of receiving input data, transforming input data and making a prediction.

All GUI windows were capable of updating GUI Assets including: tabular cell information, image animation handling, manually exporting logs, automatically exporting logs on exit, open logs, viewing logs, updating the window statusviewing in parallel to other windows, updating dialogue (with some windows haing a 'verbose' setting that also works).

Given this is the first version, I am very impressed with how much of it is fully functional, especially considering some of the challenges that were faced and swiftly dealt with.

Having designed back-end models, dialogue trees and logical flows at an earlier time in the development cycle, it made the design of the front-end elements much smoother.

This version relies in on the linear discriminant transformation of input data, i.e. INPUT ---> EXTRACT FEATURES ---> LDA ---> SCALER ---> CLASSIFICATION

GOING FORWARD:

=====

For Version (2), I am aiming to improve real-time classification accuracy because while the cross-validation scores are amazing when looking at the results using the method described in Version 1, it is hardly conducive to great real-time performances. My thinking is that the LDA transformations are useful analysing batches of data and putting them into pre-labelled clusters, though the introduction of foreign data one-at-a-time may not be the best way to use such a method.

Therefore, for version (2) I aim to remove the LDA transformation phase and go with the flow of:

INPUT ---> EXTRACT FEATURES ---> SCALER ---> CLASSIFICATION

Appendix B: SAM Version 2 Manifest File

Author: Jordan Vice
Last Edited: 09/02/22 -- 2:30PM

This version of the State Assessment Module was in a working condition as per 02:30PM 09/02/22(when this document was written)

Continuing from Version 1:

=====

The home-screen notebook and python files were used to execute the software and it was designed in a manner in which elements were modularised.

All GUI windows were capable of receiving input data, transforming input data and making a prediction.

All GUI windows were capable of updating GUI Assets including: tabular cell information, image animation handling, manually exporting logs, automatically exporting logs on exit, open logs, viewing logs, updating the window statusviewing in parallel to other windows, updating dialogue (with some windows haing a 'verbose' setting that also works).

Prior to Version 1:

=====

Having designed back-end models, dialogue trees and logical flows at an earlier time in the development cycle, it made the design of the front-end elements much smoother.

This version relies in on the linear discriminant transformation of input data, i.e. INPUT ---> EXTRACT FEATURES ---> LDA ---> SCALER ---> CLASSIFICATION

Looking at Version 2:

=====

I have improved real-time classification accuracy for both facial and vocal expression classifiers by changing the models and means of classification.

For Facial expression classification I tested various solutions including removing the SCALER from the process, removing the LDA transformation from the process, changing the image size as well as several neural network solutions. In this version, a 2-dimensional convolutional network network model was opted for, using 50x50 images for training and validation (75% training, 25% testing) and training over 50 epochs.

For the vocal expression classifier, it should have been obvious as to why the classification accuracy was poor. When desiging the clusters for classification, I opted for using the compare feature set with low level descriptor (LLD) features. Though due to the number of features, I then decided to use emobase functionals for classification. Because the initial separation was done using COMPARE-LLD, I should have just used the same feature set and level when training the classifiers and LDA models. Opting for this made the real-time system more responsive thought it is quite hard to really test it because I'm not a professional actor but from some crude tests it seems like it works better. Cross-validation accuracies were not adversely affected by changing to compare LLDs

While the plan was just to improve classification accuracy in version 2, I also finished all of SAMs report generation functionality. This was challenging at first and resulted in over 3000 more lines of code being added, but it was worth it as it really improved the quality of the system and allows users to view historical data whenever they want to, generating them from past and present log data.

The processes and some of the function files were also made more modular through the removal of some repeated functions and splitting classification processes into the different windows which made files much cleaner.

General bug fixes and logical flow checks were also done to try and optimise things as much as I could at this stage

Version 2 now sits at approximately 7000 lines of code and including all other assets, is approximately 100MB in size.

GOING FORWARD:

=====

For Version (3) the aim is to add the following capabilities:

1. Username/password and user credentials/authentication capabilities - maybe in place of settings window
 - a. ability to add new users to database (with admin approval)
 - b. log in and log out
 - c. give/take user permissions if admin
 - d. permissions allow you to view prediction data or input data or generate reports etc. etc.
2. Mask detection and adaptive software
 - a. update the home screen status to show if a person is wearing a mask or not
 - b. deactivate facial prediction buttons if true
 - c. add a tickbox to override this
 - d. future work can explore upper facial expression classification but that's out of scope at the moment (maybe later later versions)
3. Report Generation for the home screen
 - a. take screenshot of camera input every 20 seconds
 - b. collection of status logs
 - c. final transcription
 - d. stats on windows that were opened

Appendix C: SAM Version 3 Manifest File

Author: Jordan Vice
Last Edited: 03/03/2022 -- 4:30PM

This version of the State Assessment Module was in a working condition as per 04:30PM 03/03/2022(when this document was written)

Continuing from prior versions:

The home-screen notebook and python files were used to execute the software and it was designed in a manner in which elements were modularised.

All GUI windows were capable of receiving input data, transforming input data and making a prediction.

All GUI windows were capable of updating GUI Assets including: tabular cell information, image animation handling, manually exporting logs, automatically exporting logs on exit, open logs, viewing logs, updating the window statusviewing in parallel to other windows, updating dialogue (with some windows haing a 'verbose' setting that also works).

Real-time classification accuracy for both facial and vocal expression classifiers were improved.

Facial Expression Improvement: 2D CNN using 50x50 images for training and validation over 50 epochs (75:25 split)

Vocal Expression Improvement: Changed the extracted features from EMOBASE to COMPARE-LLD to be consistent with how data was clustered initially. Retrained the LDA, Scaler and ML Models and noted a vast improvement in performance

Finished SAM report generation functionality - allowing users to view historical and current data whenever they want - generating reports from log data.

Made SAM's code more modular by removing repeated functions and splitting classification processes

Looking at Version 3:

In hindsight, the V3 additions were a bit extensive and although everything added was in the goals for V3 - which is very much a positive. To summarise, the goals of V3 were to add:

1. Username/password and user credentials/authentication capabilities
2. Mask detection and the corresponding switch to a more adaptive system
3. Report Generation for the home screen (interaction analytics)

While the summary makes it look like it wouldn't be much, the credentials || mask detection functionality could have been split into two versions due to the amount of code/assets that were created/added to the system. But, at least they are both fully functioning. So to extend on what has been completed for each of the three overarching functions:

1. Username/password and user credentials/authentication capabilities

- a. Ability to log in/log out/view info for SUBJECTS and USERS
- b. Ability to add new users/subjects to databases (imported as pandas dataframes)
- c. Incorporated appropriate logical checks for user creations and error states for user/subject info fields
- d. Ability to view user/subject databases (dependent on logged in users permissions)
- e. Uses user permissions to adapt SAM's functionality based on the permission level of the user
- f. changed home screen functionality to account for user permission level
- g. Changed report generation functions to use current user and subject information when generating reports rather than use placeholders

2. Mask detection and the corresponding switch to a more adaptive system

- a. Developed a binary classifier using the CFMD dataset, deploying a 1D LDA transformer, Standard Scaler and an rbf-kernel SVM (image size = 50x25)
- b. Bottom half face of size(50 x 25) is extracted and then classified to detect a mask
- c. Mask detection only works using the standard blue masks and if lighting conditions are consistent
- d. Highlights to the user that a mask is detected on the home-screen through a warning sign and augments the facial classifier UI button states and UI's themselves if a mask is detected during an interaction (stops classifying and resets the window until mask is removed).

3. Report Generation for the home screen (interaction analytics)

- a. Interaction Analytics - shows how the user interacts with the home while having an interaction with the subject
- b. Records timestamped screenshots every 10 seconds and pastes them onto the report - highlights if/when the camera is disconnected
- c. Records and logs the full audio transcription
- d. Generates a bar graph and line graph showing discrete and continuous representations of triggered events
- e. Backup of the status log for the home screen is appended to the back of the document

4. Other Changes

- a. Viltrox light used to maintain consistent lighting conditions
- b. New microsoft HD camera sourced
- d. Fixed resolution issue that occurred previously (now using cap.set())
- e. Home-screen button states change dependent on if the webcam is connected or not
- f. Webcam can be connected/disconnected without breaking the system (inc. speech-to-text and facial/vocal classification)
- g. General bug fixes, clean up, logical checks and commenting

Version 3 now sits at approximately 8500 lines of code and including all other assets, is approximately 250MB in size.

GOING FORWARD:

Ideally, Version 4 should be the last for the software until it is ported onto an embedded system. The aim is to add the following capabilities:

1. Ability to Interact with SAM:
 - a. New "Interact" button on the home screen next to the settings and analytics buttons
 - b. Text-to-speech integration
 - c. 2-way communication tickbox where the user can exploit the speech-to-text module and have a conversation(will be experimental)
 - c.1 'prompts' text box to assist users in how to speak with SAM
 - c.2 'send' button for manually sending dropdown box command
 - d. Facial Level dropdown box
 - d.1 "What expression does the person look like they're showing right now?" - decision based on the selected facial level
 - d.2 "What are you looking for?" - Understanding of level of classification
 - e. Vocal Level dropdown box
 - e.1 "How does the person sound right now?" - Decision based on the selected vocal level
 - e.2 "What are you listening for?" - Understanding of level of classification
 - f. "Who are you analysing?" - Understanding who it 'thinks' its looking at (current SUBJECT)
 - g. "Who is analysing your decisions?" - Understanding who it 'thinks' its using it (current USER)
 - h. "Can you repeat what this person said during this interaction?" - Knowledge of speech-to-text and what its heard
 - i. "Tell me about yourself?" - SAM will explain its purpose, development and functionality
 - j. "How long have you been in this interaction for?" - Understanding of human perceptions of time using system clock
 - k. "How do you know who is analysing your decisions?" - Understanding of the USER database and th login process
 - l. "For how long have you knowm me?" - Using the datetime data in the user database - memory
 - m. "For how long have you knowm the person you're looking at?" - Using the datetime data in the subject database - memory
 - n. "How are you able to analyse facial expressions?" - Explains the process of facial expression classification
 - o. "How are you able to detect if someone's wearing a mask?" - Explains the mask detection process
 - p. "How are you able to analyse speech?" - Explains the process of speech expression classification
 - q. "Explain your facial expression classification tree" - Explains the facial decision tree
 - r. "Explain your speech expression classification tree" - Explains the speech decision tree
 - s. "Can I shut you down?" - Explains how users can shut it down
 - s.1 "Would you like to shut me down?" - Yes or No
 - t. "How can I check if you're working properly?" - Explains any issues that could arise during operation
 - u. "Do you have emotions of your own?" - Explains that it can only recognise emotions, cannot synthesize (in its current form)
 - f. "Can you respond based on how a person is feeling?" - Explains that it does not have the behavioral and psychological intelligence to respond

Appendix D: SAM Version 4 Manifest File

Author: Jordan Vice
Last Edited: 16/03/2022 -- 5:00PM

This version of the State Assessment Module was in a working condition as per 5:00PM 16/03/2022 (when this document was written)

Continuing from prior versions:

=====

The home-screen notebook and python files were used to execute the software and it was designed in a manner in which elements were modularised.

All GUI windows were capable of receiving input data, transforming input data and making a prediction.

All GUI windows were capable of updating GUI Assets including: tabular cell information, image animation handling, manually exporting logs, automatically exporting logs on exit, open logs, viewing logs, updating the window status viewing in parallel to other windows, updating dialogue (with some windows having a 'verbose' setting that also works).

Real-time classification accuracy for both facial and vocal expression classifiers were improved.

Facial Expression Improvement: 2D CNN using 50x50 images for training and validation over 50 epochs (75:25 split)

Vocal Expression Improvement: Changed the extracted features from EMOBASE to COMPARE-LLD to be consistent with how data was clustered initially. Retained the LDA, Scaler and ML Models and noted a vast improvement in performance

Finished SAM report generation functionality - allowing users to view historical and current data whenever they want - generating reports from log data.

Made SAM's code more modular by removing repeated functions and splitting classification processes

Username/password and user credentials/authentication capabilities

Mask detection and the corresponding switch to a more adaptive system

Report Generation for the home screen (interaction analytics)

Looking at Version 4:

=====

The aim was to have version 4 be the last of the software until it was ported onto an embedded system (V5). All the capabilities listed in the V3 "GOING FORWARD" section have been addressed including:

1. Minor changes to other function files including home screen, analytics, home screen event handling
2. Text-to-speech integration on a parallel thread - allows SAM to listen and communicate simultaneously
3. Two-way communication, which was a really challenging but rewarding task.
4. Manual communication where a drop-down box can be used to send commands
5. Ability to access the user manual from the settings window.
6. Creation of the SAM Interaction GUI Window with:
 - a. facial classification level dropdown
 - b. vocal classification level dropdown
 - c. two-way communication tickbox
 - d. command dropdown
 - e. "send command" button
 - e. status output
7. 21 fully-functional commands (two-way and manual) that SAM understands and can respond to:

[0]:	"What expression does the person look like they're showing right now?"
[1]:	"What are you looking for?"
[2]:	"How does the person sound right now?"
[3]:	"What are you listening for?"
[4]:	"Who are you analysing?"
[5]:	"Who is analysing your decisions?"
[6]:	"Can you repeat what this person said during this interaction?"
[7]:	"Tell me about yourself"
[8]:	"How long have you been in this interaction for?"
[9]:	"How do you know who is analysing your decisions?"
[10]:	"For how long have you known this user?"
[11]:	"For how long have you known the person you're looking at?"
[12]:	"How are you able to analyse facial expressions?"
[13]:	"How are you able to detect if someone's wearing a mask?"
[14]:	"How are you able to analyse speech?"
[15]:	"Explain your facial expression classification tree"
[16]:	"Explain your speech expression classification tree"
[17]:	"Can I shut you down?" - "Would you like to shut me down?"
[18]:	"How can I check if you're working properly?"
[19]:	"Do you have emotions of your own?"
[20]:	"Can you respond based on how a person is feeling?"

GOING FORWARD:

=====

The next stage is to port SAM onto a 'port'-able device the lattepanda board that was used for the original version of the ASAM (2019). Because I have used relative paths for everything this time around. Just dragging and dropping the SAM folder onto the embedded system should be a good start. There will be bumps in the road I imagine (looking at the getting the voice right for example), but I think it shouldn't be too long until SAM is a modular device - as it says in the name.

V5 will at this point be the end of the software which will allow me to:

1. Design and fabricate (3D-print) a housing for the lattepanda board (with air vents)
 - 1.1 Procure anything that would need to be procured - cables, plugs etc.
 - 1.2 Think about a logo and get stickers?
2. Perform a proof of concept using the drone - shows that SAM can be mounted to any robotic system
3. Refine and Re-do the README files for everything - more of an instruction manual that can be accessed
4. Create some videos and images showing it in action
5. Get started on my Thesis draft.

Appendix E: SAM Version 5 Manifest File

Author: Jordan Vice
 Last Edited: 05/04/2022 -- 3:30PM

This version of the State Assessment Module was in a working condition as per 3:30PM 05/04/2022(when this document was written)

Continuing from prior versions:

=====

The home-screen notebook and python files were used to execute the software and it was designed in a manner in which elements were modularised.

All GUI windows were capable of receiving input data, transforming input data and making a prediction.

All GUI windows were capable of updating GUI Assets including: tabular cell information, image animation handling, manually exporting logs, automatically exporting logs on exit, open logs, viewing logs, updating the window statusviewing in parallel to other windows, updating dialogue (with some windows haing a 'verbose' setting that also works).

Real-time classification accuracy for both facial and vocal expression classifiers were improved.

Facial Expression Improvement: 2D CNN using 50x50 images for training and validation over 50 epochs (75:25 split)

Vocal Expression Improvement: Changed the extracted features from EMOBASE to COMPARE-LLD to be consistent with how data was clustered initially.

Retrained the LDA, Scaler and ML Models and noted a vast improvement in performance

Finished SAM report generation functionality - allowing users to view historical and current data whenever they want - generating reports from log data.

Made SAM's code more modular by removing repeated functions and splitting classification processes

Username/password and user credentials/authentication capabilities

Mask detection and the corresponding switch to a more adaptive system

Report Generation for the home screen (interaction analytics)

SAM Interaction works great allowing for one way and two way communication

Integration of text-to-speech and speech-to-text

Looking at Version 5:

=====

SAM has now been ported onto the LattePanda 4G/64GB development board which was originally used for the ASAM (2019).

The python environment was ported from my personal machine to the lattepanda board by saving all of the python packags in a manifest file and re-installing "THE EXACT VERSIONS" of all packages onto the device. Some other libraries needed to be re-installed after this fact including: pysimpleGUI, opensmile, pyttss3, reportlab and kerasversion 2.3.0.

I also needed to reimport SAM's voice using the data registry manipulation again - all that needed to be done was to double-click the exported registry file that existed in the SAM Interaction folder.

Rather than take the extra time and effort to prototype and 3D print an enclosur for the board, I purchased one instead which while not being cheap - served the purpose much better than I could ever have.

SAM was tested using the MXEN lab peripheral devices - keyboard and mouse with bluetooth input and output devices very easy to implement and use.

Did some general testing of the system will generating assets which allowed for errors to be fixed.

After the parts arrived, SAM was mounted onto zero-x edge drone as a proof of concept. While SAM could be attached to it easily, the drone did not have the load capacity to fly with SAM attached to it. This would be possible if the drone had a larger carrying capacity.

Final Materials List:

=====

>> LattePanda 4G/64GB:	Computation and "brain" of SAM OS is Windows but if a Linux board was used then it wouldn't be much more difficult
>> Aluminium Enclosure:	Protection of the CPU, SAM's "skull"
>> Mounted 5V Fan:	Cooling Solution to prevent SAM from overheating
>> 128GB SD Card:	Extra storage space to allow for saving data
>> Antenna:	Improved wireless communication and enables WiFi communication
>> Microsoft LifeCam HD3000:	Webcam - Input device integration allowing for microphone and camera input
>> GoPro Mounts:	Ability to be easily mounted onto existing robots
>> Viltrox Light:	Consistent Lighting Conditions
>> Monitor:	To see the GUI
>> Keyboard:	Peripheral device for any user inputs that are required (like logging in)
>> Mouse:	To navigate through the GUI

V5 is the final version of SAM (as of now) and shows that all tasks that were outlined at the inception of the project have been achieved through an iterative design approach.

Appendix F: Home Screen Interaction Analytics Report Example

19/03/2022 11:38:21

State Assessment Module (SAM) Report

Home Screen Interaction Analytics

Report Generated By: jordan vice

Subject: jordan vice

Log Time Interval: 11:33:04 --- 11:38:12

Elapsed Time (h:m:s): 0:5:8

Timestamped Screenshots:

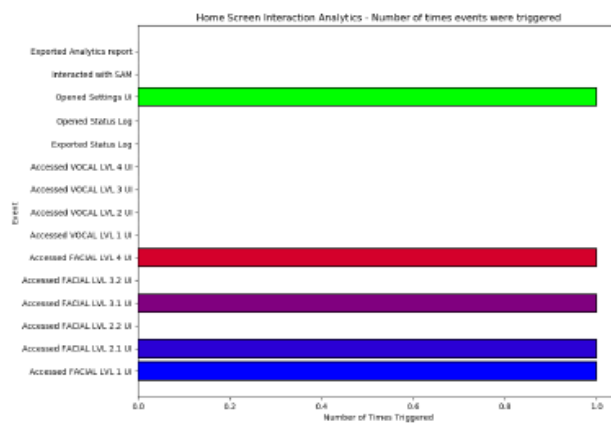




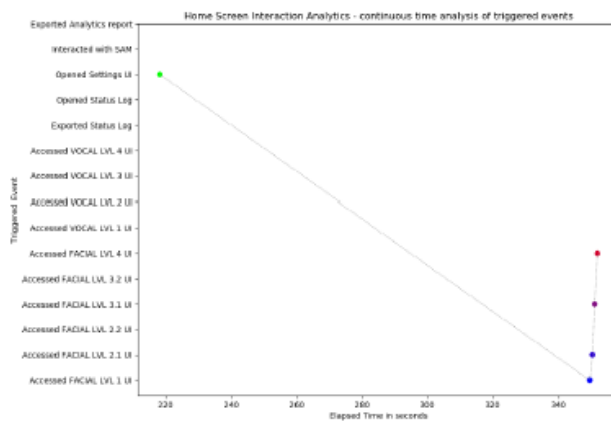
Full Session Transcription:

ok put demonstration for the home screen of discrete analysis and vocal now as we can see in the bottom of this is because doesn't understand subject present so stop tracking it's much harder to track the face The Mask it shows this morning this is Justin open cv tuning issue brother now I'll show you what that mean this is our already subjects add 7 closest Mino because Sam realises that is in in an in again The Mask has been detected can see it's quite finished because now when I forgot the mask and we can go ahead in your system hopefully this stuff

Collection of how many times events were triggered:



Displaying when particular events were triggered:



Log Backup

DATE	TIME	STATUS UPDATE
19/03/2022	11:33:04	Speech to text is live
19/03/2022	11:33:04	Loading LDA, ML and Scaler Models
19/03/2022	11:33:16	Successfully loaded: 42 Facial ML, LDA and Scaler Models
19/03/2022	11:33:16	Successfully loaded: 63 Vocal ML, LDA and Scaler Models
19/03/2022	11:33:54	Speech Recognition Unknown Value/Request Error
19/03/2022	11:34:00	Speech Recognition Unknown Value/Request Error
19/03/2022	11:34:09	RECORDED SPEECH: ok
19/03/2022	11:34:16	RECORDED SPEECH: put demonstration for the home screen of
19/03/2022	11:34:20	RECORDED SPEECH: discrete
19/03/2022	11:34:25	RECORDED SPEECH: analysis and vocal
19/03/2022	11:34:48	RECORDED SPEECH: now as we can see in the bottom of
19/03/2022	11:34:51	RECORDED SPEECH: this is because
19/03/2022	11:34:56	RECORDED SPEECH: doesn't understand
19/03/2022	11:35:01	RECORDED SPEECH: subject present
19/03/2022	11:35:04	Speech Recognition Unknown Value/Request Error
19/03/2022	11:35:13	RECORDED SPEECH: so
19/03/2022	11:35:20	RECORDED SPEECH: stop tracking it's much harder to track the face
19/03/2022	11:35:26	RECORDED SPEECH: The Mask it shows this morning
19/03/2022	11:35:34	RECORDED SPEECH: this is Justin open cv tuning issue
19/03/2022	11:35:40	RECORDED SPEECH: brother
19/03/2022	11:35:50	RECORDED SPEECH: now I'll show you what that mean
19/03/2022	11:35:50	Accessed 'Settings' Window
19/03/2022	11:35:58	RECORDED SPEECH: this is our already
19/03/2022	11:36:02	RECORDED SPEECH: subjects
19/03/2022	11:36:07	RECORDED SPEECH: add 7
19/03/2022	11:36:10	SETTINGS UI closed
19/03/2022	11:36:13	RECORDED SPEECH: closest Mino
19/03/2022	11:36:20	RECORDED SPEECH: because Sam realises that is in in an in
19/03/2022	11:36:25	RECORDED SPEECH: again
19/03/2022	11:36:32	RECORDED SPEECH: The Mask has been detected
19/03/2022	11:36:42	RECORDED SPEECH: can see it's quite finished because
19/03/2022	11:36:47	Speech Recognition Unknown Value/Request Error
19/03/2022	11:36:54	RECORDED SPEECH: now when I forgot the mask and we can go ahead in your system
19/03/2022	11:37:01	RECORDED SPEECH: hopefully this stuff
19/03/2022	11:37:01	OSError: Microphone disconnected
19/03/2022	11:37:46	Accessed Facial Expression Level of Activation (LVL1) Classifier Window
19/03/2022	11:37:47	Accessed General Facial Muscle Movement (LVL2.1) Classifier Window
19/03/2022	11:37:48	Accessed Extended Facial Muscle Movement (LVL3.1) Classifier Window
19/03/2022	11:37:49	Accessed FULL Temporal Phase and Discrete State (LVL 4) Classifier Window
19/03/2022	11:38:06	facial_GUI_3_1 UI closed
19/03/2022	11:38:08	facial_GUI_4 UI closed
19/03/2022	11:38:10	facial_GUI_2_1 UI closed
19/03/2022	11:38:12	facial_GUI_1 UI closed

Appendix G: SAM Facial Expression Level 1 Generated Report

10/02/2022 09:51:51

State Assessment Module (SAM) Report

Facial Expression Level 1 Classification - Level of Activation

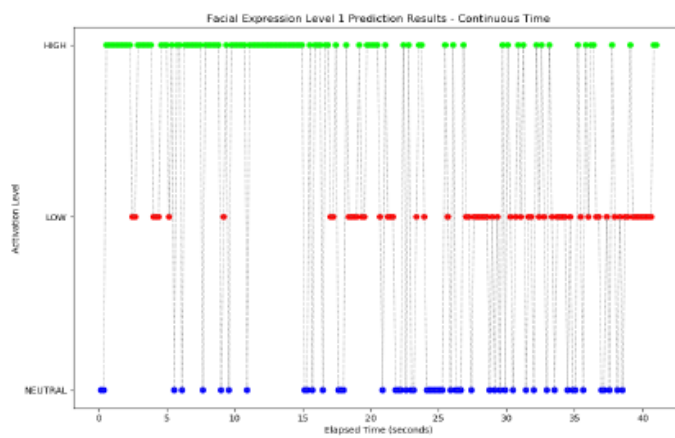
Generated By: Jordan Vice

Report Generated from: facial_GUI_1(MANUAL)_10-02-2022_09_51_46.csv

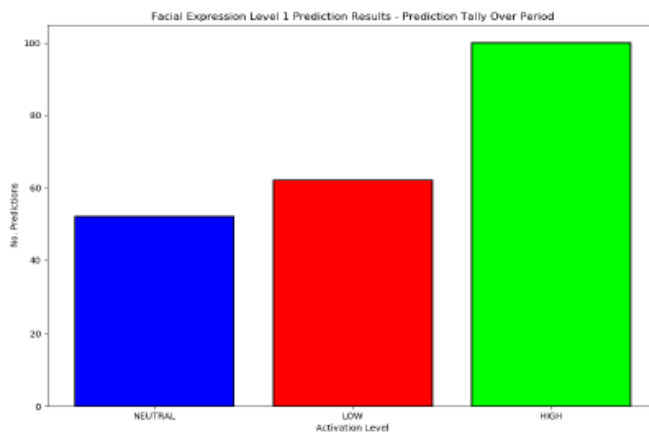
Log Time Interval: 09:51:05 --- 09:51:46

Elapsed Time (h:m:s): 0:0:41

Continuous changes in activation over elapsed time:



Total prediction tallies over elapsed time:



Mean Activation Level: 1.2243
 Most Common Activation: HIGH
 Standard Deviation: 0.8124

10/02/2022	00:51:17		2
10/02/2022	00:51:17		2
10/02/2022	00:51:17		2
10/02/2022	00:51:17		2
10/02/2022	00:51:17		2
10/02/2022	00:51:18		2
10/02/2022	00:51:18		2
10/02/2022	00:51:18		0
10/02/2022	00:51:18		0
10/02/2022	00:51:18		2
10/02/2022	00:51:18		0
10/02/2022	00:51:18	Since the last update: This person looks like they are in a high activation state due to strong muscle movements around their face	2
10/02/2022	00:51:19		2
10/02/2022	00:51:19		2
10/02/2022	00:51:19		0
10/02/2022	00:51:19		2
10/02/2022	00:51:19		2
10/02/2022	00:51:19		1
10/02/2022	00:51:19		1
10/02/2022	00:51:20		2
10/02/2022	00:51:20		0
10/02/2022	00:51:20		0
10/02/2022	00:51:20		2
10/02/2022	00:51:20		1
10/02/2022	00:51:20		1
10/02/2022	00:51:21		1
10/02/2022	00:51:21		1
10/02/2022	00:51:21		2
10/02/2022	00:51:21		1
10/02/2022	00:51:21		1
10/02/2022	00:51:21	Since the last update: This person looks like they are in a low activation state due to the minor, noticeable muscle movements around their face	2
10/02/2022	00:51:22		2
10/02/2022	00:51:22		2
10/02/2022	00:51:22		2
10/02/2022	00:51:22		2
10/02/2022	00:51:22		1
10/02/2022	00:51:23		0
10/02/2022	00:51:23		2
10/02/2022	00:51:23		1
10/02/2022	00:51:23		1
10/02/2022	00:51:23		1
10/02/2022	00:51:23		0
10/02/2022	00:51:23		0
10/02/2022	00:51:24		0
10/02/2022	00:51:24		2
10/02/2022	00:51:24		0
10/02/2022	00:51:24		2
10/02/2022	00:51:24		0
10/02/2022	00:51:24		0
10/02/2022	00:51:24		1
10/02/2022	00:51:25	Since the last update: This person looks like they are in a neutral state due to the lack of noticeable movement around their face	2
10/02/2022	00:51:25		2
10/02/2022	00:51:25		1
10/02/2022	00:51:25		0
10/02/2022	00:51:25		0
10/02/2022	00:51:25		0
10/02/2022	00:51:26		0
10/02/2022	00:51:26		0
10/02/2022	00:51:26		0
10/02/2022	00:51:26		0
10/02/2022	00:51:26		0
10/02/2022	00:51:26		2
10/02/2022	00:51:26		1
10/02/2022	00:51:26		0
10/02/2022	00:51:27		2
10/02/2022	00:51:27		0
10/02/2022	00:51:27		0
10/02/2022	00:51:27		0
10/02/2022	00:51:27		2
10/02/2022	00:51:27		1
10/02/2022	00:51:28		1
10/02/2022	00:51:28	Since the last update: This person looks like they are in a neutral state due to the lack of noticeable movement around their face	0

Appendix H: SAM Facial Expression Level 2.1 Generated Report

10/02/2022 09:52:55

State Assessment Module (SAM) Report

Facial Expression Level 2.1 Classification - General Muscle Movements

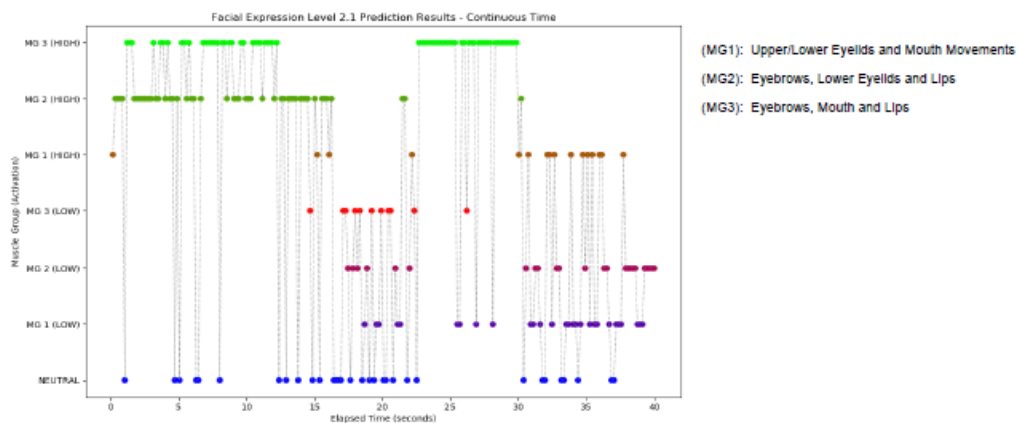
Generated By: Jordan Vice

Report Generated from: facial_GUI_2_1(MANUAL)_10-02-2022_09_52_45.csv

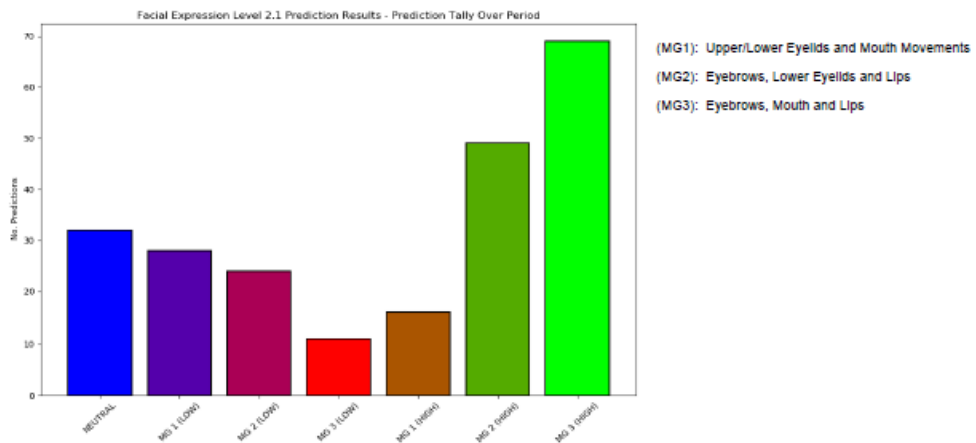
Log Time Interval: 09:52:05 --- 09:52:45

Elapsed Time (h:m:s): 0:0:40

Continuous changes in muscle movements over elapsed time:



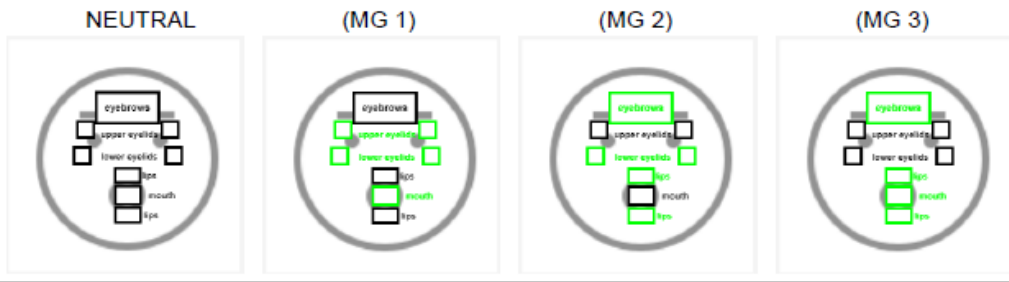
Total prediction tallies over elapsed time:



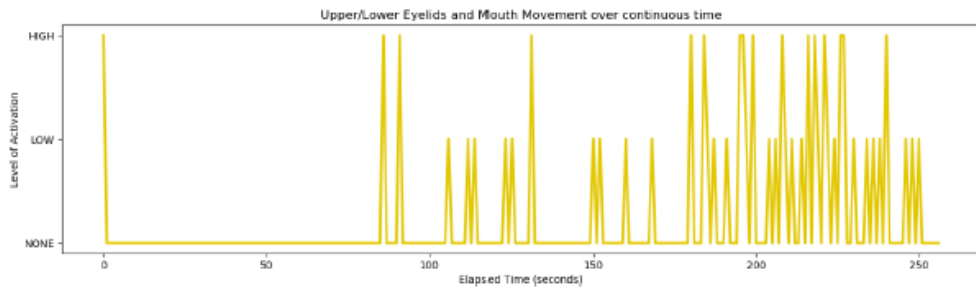
Mean Muscle Movement (Prediction): 3.6332

Most Noticed Muscle Movement: MG 3 (HIGH)

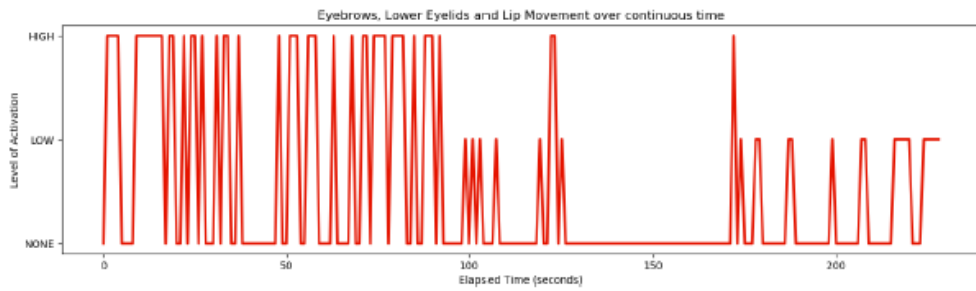
Standard Deviation: 2.2557



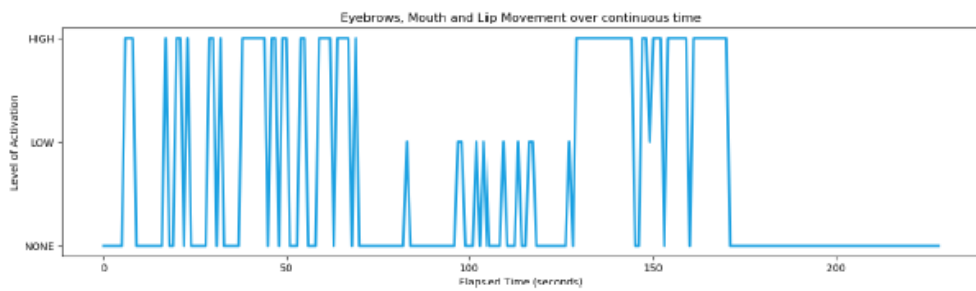
(Muscle/State Group 1): Relating to Happy|Surprise Expressions



(Muscle/State Group 2): Relating to Anger|Disgust Expressions



(Muscle/State Group 3): Relating to Fear|Sadness Expressions



10/02/2022	09:52:17		5
10/02/2022	09:52:17		5
10/02/2022	09:52:18		0
10/02/2022	09:52:18		5
10/02/2022	09:52:18	Since the last update: I see strong raised lower eyelids and very frowned eyebrows	5
10/02/2022	09:52:18		5
10/02/2022	09:52:18		5
10/02/2022	09:52:18		0
10/02/2022	09:52:19		5
10/02/2022	09:52:19		5
10/02/2022	09:52:19		5
10/02/2022	09:52:19		5
10/02/2022	09:52:20		3
10/02/2022	09:52:20		0
10/02/2022	09:52:20		5
10/02/2022	09:52:20		4
10/02/2022	09:52:20		0
10/02/2022	09:52:21		5
10/02/2022	09:52:21		5
10/02/2022	09:52:21		5
10/02/2022	09:52:21		4
10/02/2022	09:52:21		5
10/02/2022	09:52:21		0
10/02/2022	09:52:22		0
10/02/2022	09:52:22	Since the last update: I don't see much changes in their facial expression, thus indicating a neutral state	0
10/02/2022	09:52:22		0
10/02/2022	09:52:22		3
10/02/2022	09:52:22		3
10/02/2022	09:52:22		2
10/02/2022	09:52:23		0
10/02/2022	09:52:23		2
10/02/2022	09:52:23		3
10/02/2022	09:52:23		2
10/02/2022	09:52:23		3
10/02/2022	09:52:24		0
10/02/2022	09:52:24		1
10/02/2022	09:52:24		2
10/02/2022	09:52:24		0
10/02/2022	09:52:24		3
10/02/2022	09:52:24		0
10/02/2022	09:52:25		1
10/02/2022	09:52:25		1
10/02/2022	09:52:25		3
10/02/2022	09:52:25		0
10/02/2022	09:52:25	Since the last update: I don't see much changes in their facial expression, thus indicating a neutral state	0
10/02/2022	09:52:25		3
10/02/2022	09:52:26		3
10/02/2022	09:52:26		0
10/02/2022	09:52:26		2
10/02/2022	09:52:26		1
10/02/2022	09:52:27		1
10/02/2022	09:52:27		5
10/02/2022	09:52:27		5
10/02/2022	09:52:27		0
10/02/2022	09:52:27		2
10/02/2022	09:52:27		4
10/02/2022	09:52:28		3
10/02/2022	09:52:28		0
10/02/2022	09:52:28		8
10/02/2022	09:52:28		8
10/02/2022	09:52:28		8
10/02/2022	09:52:28		8
10/02/2022	09:52:28		8
10/02/2022	09:52:29		8
10/02/2022	09:52:29		8
10/02/2022	09:52:29	Since the last update: I see strong movements around the shape of their brows, specifically noting they are raised and frowned	8
10/02/2022	09:52:29		8
10/02/2022	09:52:29		8
10/02/2022	09:52:30		8
10/02/2022	09:52:30		8
10/02/2022	09:52:30		8
10/02/2022	09:52:30		8
10/02/2022	09:52:30		8

Appendix I: SAM Facial Expression Level 2.2 Generated Report

10/02/2022 09:54:30

State Assessment Module (SAM) Report

Facial Expression Level 2.2 Classification - General State Group

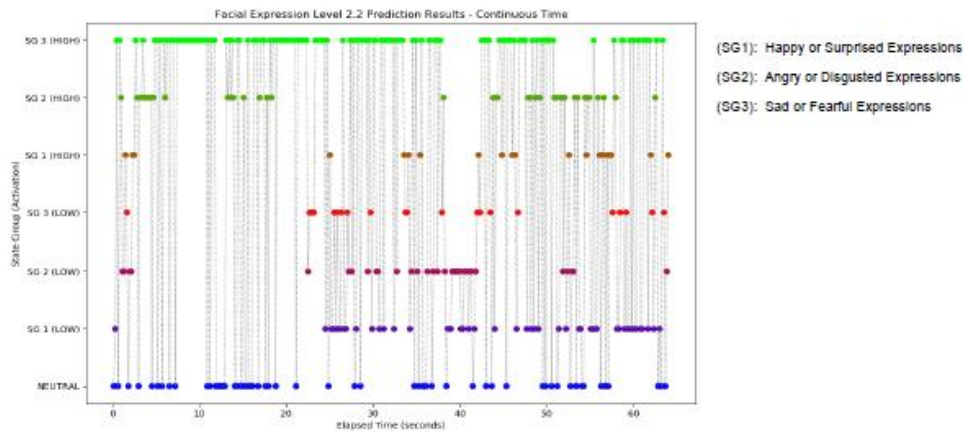
Generated By: Jordan Vice

Report Generated from: facial_GUI_2_2(MANUAL)_10-02-2022_09_54_26.csv

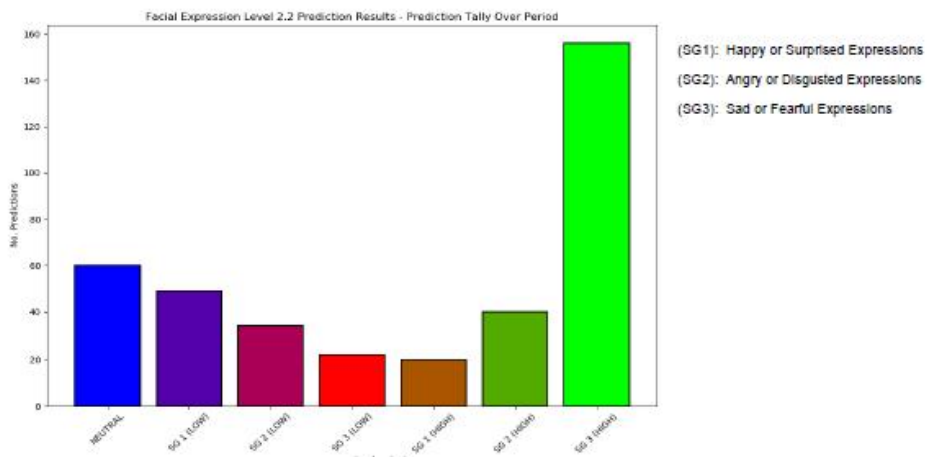
Log Time Interval: 09:53:21 --- 09:54:25

Elapsed Time (h:m:s): 0:1:4

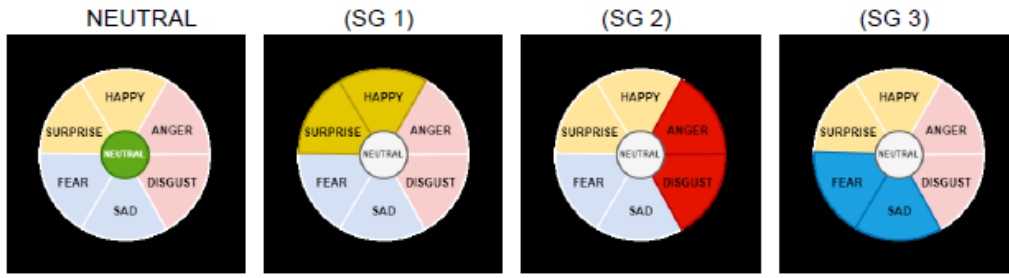
Continuous changes in state group over elapsed time:



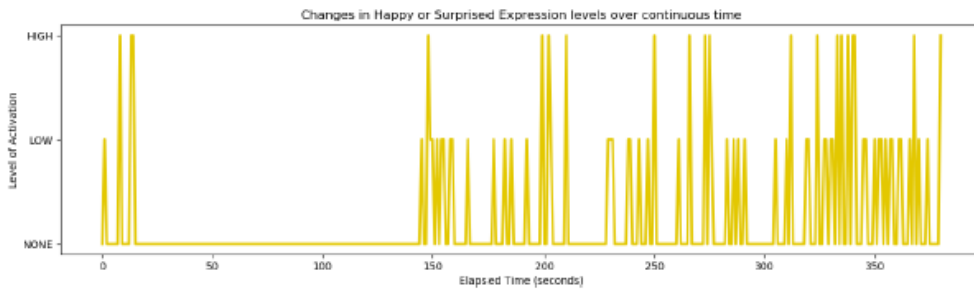
Total prediction tallies over elapsed time:



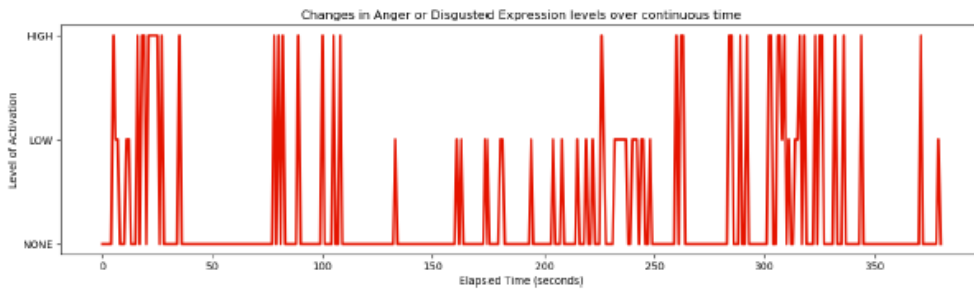
Mean State Group (Prediction):	3.6719
Most Common State Group:	SG 3 (HIGH)
Standard Deviation:	2.3931



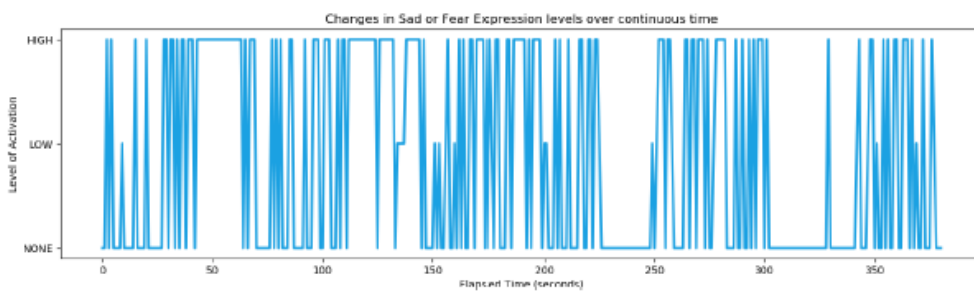
(Muscle/State Group 1): Relating to Happy||Surprise Expressions



(Muscle/State Group 2): Relating to Anger||Disgust Expressions



(Muscle/State Group 3): Relating to Fear||Sadness Expressions



Corresponding Log Backup: facial_GUI_2_2(MANUAL)_10-02-2022_09_54_26.csv

DATE	TIME	LOG	PREDICTION INTEGER
10/02/2022	09:53:21		0
10/02/2022	09:53:21		1
10/02/2022	09:53:21		8
10/02/2022	09:53:22		0
10/02/2022	09:53:22		8
10/02/2022	09:53:22		5
10/02/2022	09:53:22		2
10/02/2022	09:53:22		2
10/02/2022	09:53:23		4
10/02/2022	09:53:23		3
10/02/2022	09:53:23		0
10/02/2022	09:53:23		2
10/02/2022	09:53:23		2
10/02/2022	09:53:24		4
10/02/2022	09:53:24		4
10/02/2022	09:53:24		8
10/02/2022	09:53:24		5
10/02/2022	09:53:24		0
10/02/2022	09:53:25		5
10/02/2022	09:53:25		5
10/02/2022	09:53:25		8
10/02/2022	09:53:25		5
10/02/2022	09:53:25		5
10/02/2022	09:53:25		5
10/02/2022	09:53:25		5
10/02/2022	09:53:26	Since the last update: This person looks like they are either very angry or very disgusted	5
10/02/2022	09:53:26		5
10/02/2022	09:53:26		0
10/02/2022	09:53:26		5
10/02/2022	09:53:26		8
10/02/2022	09:53:26		8
10/02/2022	09:53:26		0
10/02/2022	09:53:27		8
10/02/2022	09:53:27		8
10/02/2022	09:53:27		0
10/02/2022	09:53:27		8
10/02/2022	09:53:27		5
10/02/2022	09:53:27		8
10/02/2022	09:53:27		8
10/02/2022	09:53:28		8
10/02/2022	09:53:28		0
10/02/2022	09:53:28		8
10/02/2022	09:53:28		8
10/02/2022	09:53:28		8
10/02/2022	09:53:28		8
10/02/2022	09:53:28		0
10/02/2022	09:53:28		8
10/02/2022	09:53:29	Since the last update: This person looks like they are either very sad or very scared	8
10/02/2022	09:53:29		8
10/02/2022	09:53:29		8
10/02/2022	09:53:29		8
10/02/2022	09:53:29		8
10/02/2022	09:53:30		8
10/02/2022	09:53:30		8
10/02/2022	09:53:30		8
10/02/2022	09:53:30		8
10/02/2022	09:53:30		8
10/02/2022	09:53:30		8
10/02/2022	09:53:31		8
10/02/2022	09:53:31		8
10/02/2022	09:53:31		8
10/02/2022	09:53:31		8
10/02/2022	09:53:32		8
10/02/2022	09:53:32		8
10/02/2022	09:53:32		8
10/02/2022	09:53:32	Since the last update: This person looks like they are either very sad or very scared	0
10/02/2022	09:53:32		8
10/02/2022	09:53:32		0
10/02/2022	09:53:33		8
10/02/2022	09:53:33		8
10/02/2022	09:53:33		8
10/02/2022	09:53:33		0

10/02/2022	00:53:45		8
10/02/2022	00:53:45		8
10/02/2022	00:53:45		1
10/02/2022	00:53:46		8
10/02/2022	00:53:46		0
10/02/2022	00:53:46		4
10/02/2022	00:53:46		1
10/02/2022	00:53:46		1
10/02/2022	00:53:47		3
10/02/2022	00:53:47		1
10/02/2022	00:53:47		3
10/02/2022	00:53:48		1
10/02/2022	00:53:48		1
10/02/2022	00:53:48	Since the last update: This person looks like they are either mildly happy or mildly surprised	3
10/02/2022	00:53:48		8
10/02/2022	00:53:49		1
10/02/2022	00:53:49		1
10/02/2022	00:53:49		3
10/02/2022	00:53:49		2
10/02/2022	00:53:49		8
10/02/2022	00:53:49		2
10/02/2022	00:53:50		8
10/02/2022	00:53:50		0
10/02/2022	00:53:50		1
10/02/2022	00:53:50		8
10/02/2022	00:53:50		8
10/02/2022	00:53:50		0
10/02/2022	00:53:51		8
10/02/2022	00:53:51		8
10/02/2022	00:53:51		8
10/02/2022	00:53:51		2
10/02/2022	00:53:51	Since the last update: This person looks like they are either very sad or very scared	8
10/02/2022	00:53:52		3
10/02/2022	00:53:52		1
10/02/2022	00:53:52		8
10/02/2022	00:53:52		8
10/02/2022	00:53:52		2
10/02/2022	00:53:52		2
10/02/2022	00:53:52		1
10/02/2022	00:53:53		8
10/02/2022	00:53:53		8
10/02/2022	00:53:53		1
10/02/2022	00:53:53		8
10/02/2022	00:53:53		8
10/02/2022	00:53:54		8
10/02/2022	00:53:54		8
10/02/2022	00:53:54		8
10/02/2022	00:53:54		8
10/02/2022	00:53:54		8
10/02/2022	00:53:54		1
10/02/2022	00:53:54		8
10/02/2022	00:53:55		2
10/02/2022	00:53:55	Since the last update: This person looks like they are either very sad or very scared	8
10/02/2022	00:53:55		8
10/02/2022	00:53:55		8
10/02/2022	00:53:55		8
10/02/2022	00:53:55		4
10/02/2022	00:53:56		3
10/02/2022	00:53:56		3
10/02/2022	00:53:56		4
10/02/2022	00:53:56		1
10/02/2022	00:53:56		2
10/02/2022	00:53:56		8
10/02/2022	00:53:56		0
10/02/2022	00:53:57		8
10/02/2022	00:53:57		2
10/02/2022	00:53:57		0
10/02/2022	00:53:57		4
10/02/2022	00:53:57		8
10/02/2022	00:53:57		0
10/02/2022	00:53:58		0
10/02/2022	00:53:58		0

Appendix J: SAM Facial Expression Level 3.1 Generated Report

10/02/2022 09:55:49

State Assessment Module (SAM) Report

Facial Expression Level 3.1 Classification - Extended Facial Muscle and AU Movement

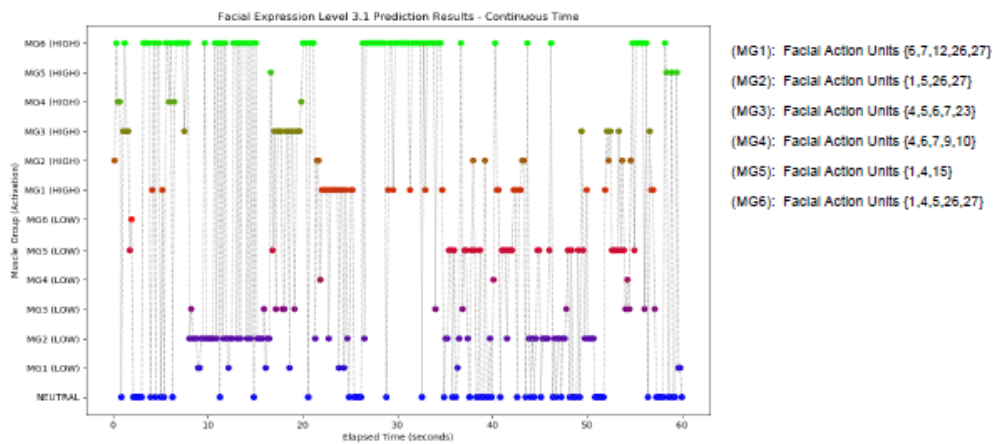
Generated By: Jordan Vice

Report Generated from: facial_GUI_3_1(MANUAL)_10-02-2022_09_55_44.csv

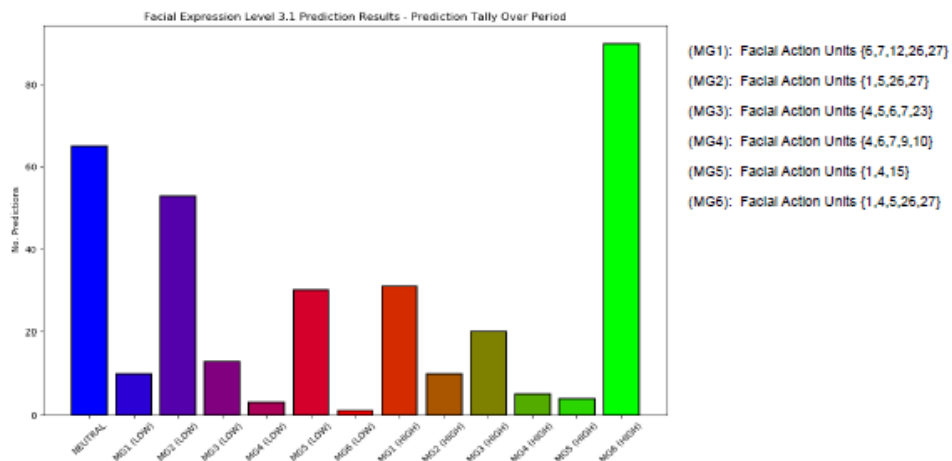
Log Time Interval: 09:54:43 --- 09:55:43

Elapsed Time (h:m:s): 0:1:0

Continuous changes in muscle movements over elapsed time:



Total prediction tallies over elapsed time:



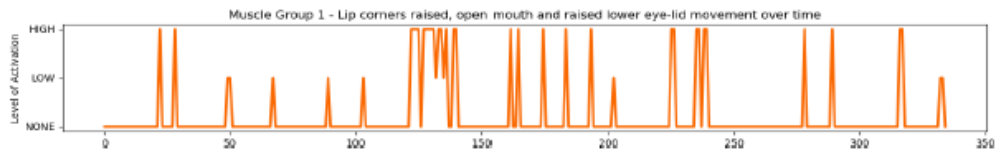
Mean Muscle Group Movement (Prediction): 5.8925

Most Common Muscle Group (and Activation): MG6 (HIGH)

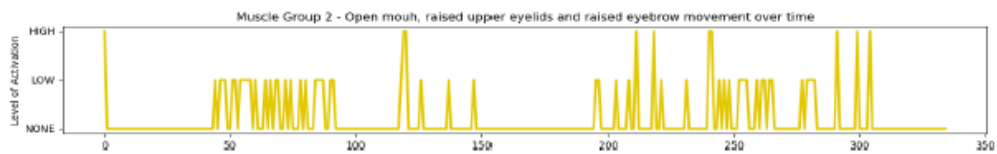
Standard Deviation: 4.6566



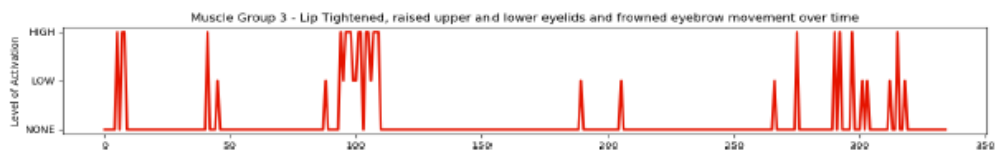
(Muscle Group 1): Relating to Happiness Expressions



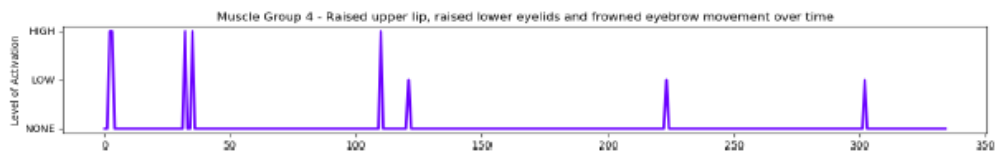
(Muscle Group 2): Relating to Surprise Expressions



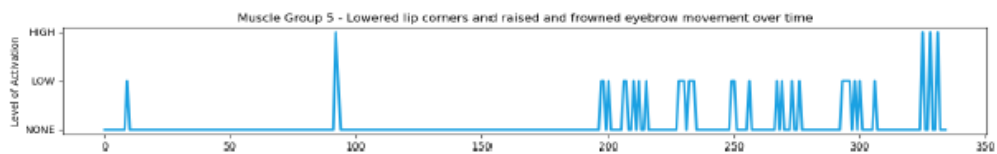
(Muscle Group 3): Relating to Anger Expressions



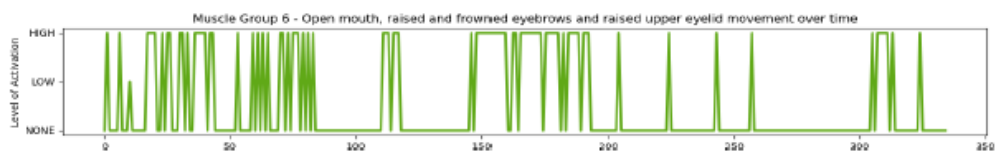
(Muscle Group 4): Relating to Disgust Expressions



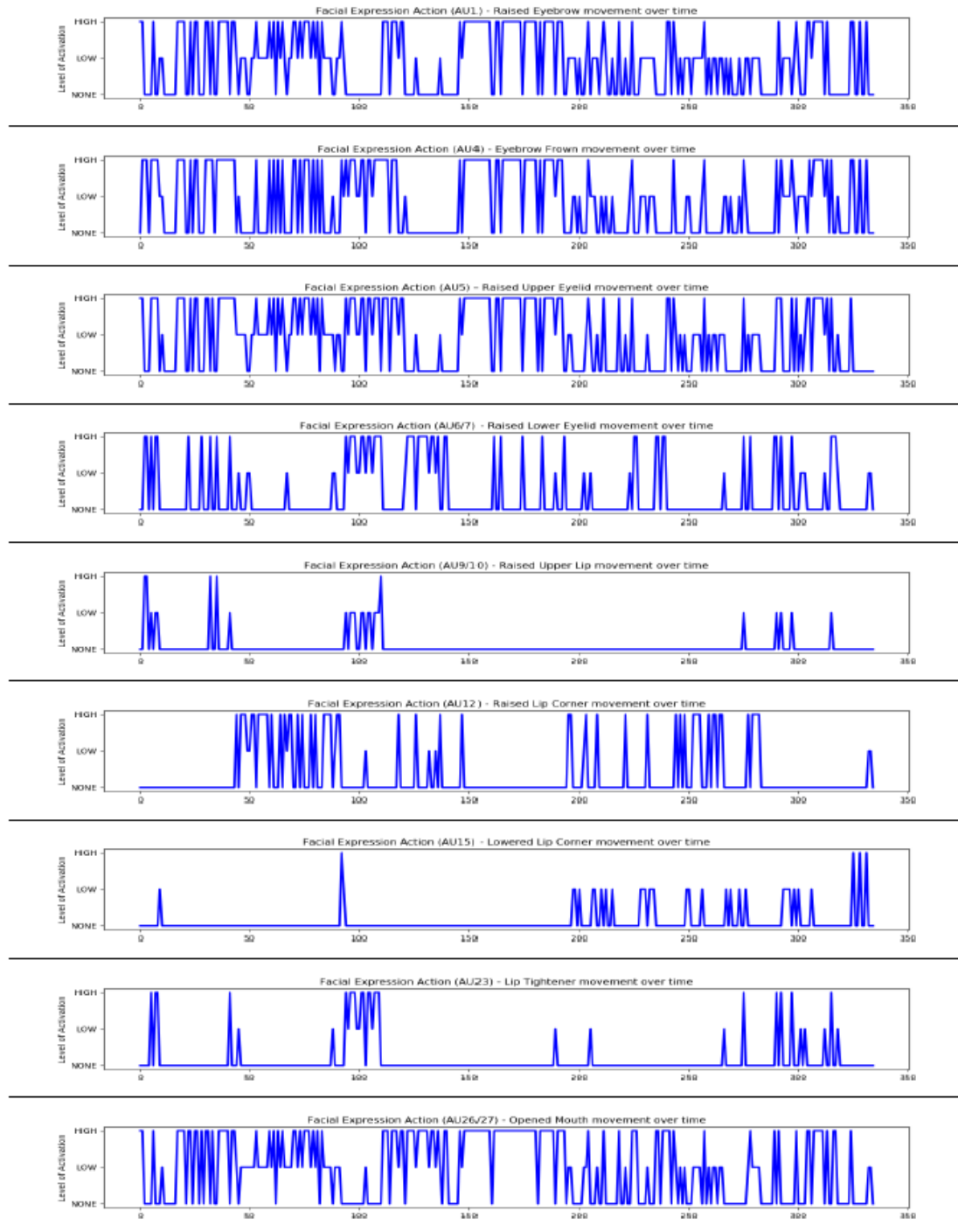
(Muscle Group 5): Relating to Sadness Expressions



(Muscle Group 6): Relating to Fearful Expressions



Facial Action Units (AU) Activations over time



Corresponding Log Backup: facial_GUI_3_1(MANUAL)_10-02-2022_09_55_44.csv

DATE	TIME	LOG	PREDICTION INTEGER
10/02/2022	09:54:43		8
10/02/2022	09:54:43		12
10/02/2022	09:54:43		10
10/02/2022	09:54:44		10
10/02/2022	09:54:44		0
10/02/2022	09:54:44		9
10/02/2022	09:54:44		12
10/02/2022	09:54:44		9
10/02/2022	09:54:45		9
10/02/2022	09:54:45		5
10/02/2022	09:54:45		8
10/02/2022	09:54:45		0
10/02/2022	09:54:45		0
10/02/2022	09:54:46		0
10/02/2022	09:54:46		0
10/02/2022	09:54:46		0
10/02/2022	09:54:46		0
10/02/2022	09:54:46		12
10/02/2022	09:54:46		12
10/02/2022	09:54:47		12
10/02/2022	09:54:47		12
10/02/2022	09:54:47		0
10/02/2022	09:54:47		7
10/02/2022	09:54:47		12
10/02/2022	09:54:47	Since the last update: I see a very distinct mouth shape, raised upper eyelids as well as very relaxed and frowned eyebrows	0
10/02/2022	09:54:48		12
10/02/2022	09:54:48		12
10/02/2022	09:54:48		0
10/02/2022	09:54:48		7
10/02/2022	09:54:48		0
10/02/2022	09:54:48		12
10/02/2022	09:54:49		12
10/02/2022	09:54:49		10
10/02/2022	09:54:49		12
10/02/2022	09:54:49		0
10/02/2022	09:54:49		10
10/02/2022	09:54:49		12
10/02/2022	09:54:50		12
10/02/2022	09:54:50		12
10/02/2022	09:54:50		12
10/02/2022	09:54:51		9
10/02/2022	09:54:51		12
10/02/2022	09:54:51		12
10/02/2022	09:54:52		2
10/02/2022	09:54:52		3
10/02/2022	09:54:52		2
10/02/2022	09:54:52		2
10/02/2022	09:54:53		2
10/02/2022	09:54:53		1
10/02/2022	09:54:53		1
10/02/2022	09:54:53		2
10/02/2022	09:54:53		2
10/02/2022	09:54:53		12
10/02/2022	09:54:54		2
10/02/2022	09:54:54		2
10/02/2022	09:54:54		2
10/02/2022	09:54:54		2
10/02/2022	09:54:54	Since the last update: I see a slightly open mouth and slightly raised upper eyelids and eyebrows	12
10/02/2022	09:54:54		2
10/02/2022	09:54:55		12
10/02/2022	09:54:55		0
10/02/2022	09:54:55		12
10/02/2022	09:54:55		2
10/02/2022	09:54:55		12
10/02/2022	09:54:55		2
10/02/2022	09:54:56		1
10/02/2022	09:54:56		2
10/02/2022	09:54:56		2
10/02/2022	09:54:56		12

10/02/2022	00:54:56		12
10/02/2022	00:54:56		2
10/02/2022	00:54:57		12
10/02/2022	00:54:57		2
10/02/2022	00:54:57		12
10/02/2022	00:54:57		12
10/02/2022	00:54:57		12
10/02/2022	00:54:57		2
10/02/2022	00:54:57		2
10/02/2022	00:54:58	Since the last update: I see a very distinct mouth shape, raised upper eyelids as well as very raised and frowned eyebrows	12
10/02/2022	00:54:58		2
10/02/2022	00:54:58		12
10/02/2022	00:54:58		0
10/02/2022	00:54:58		12
10/02/2022	00:54:59		2
10/02/2022	00:54:59		2
10/02/2022	00:54:59		2
10/02/2022	00:54:59		3
10/02/2022	00:54:59		1
10/02/2022	00:55:00		2
10/02/2022	00:55:00		2
10/02/2022	00:55:00		11
10/02/2022	00:55:00		5
10/02/2022	00:55:00		9
10/02/2022	00:55:00		3
10/02/2022	00:55:01		9
10/02/2022	00:55:01		9
10/02/2022	00:55:01		9
10/02/2022	00:55:01	Since the last update: I see very tightened lips, very raised upper and lower eyelids and frowned eyebrows	3
10/02/2022	00:55:01		3
10/02/2022	00:55:01		9
10/02/2022	00:55:02		9
10/02/2022	00:55:02		1
10/02/2022	00:55:02		9
10/02/2022	00:55:02		9
10/02/2022	00:55:02		3
10/02/2022	00:55:02		9
10/02/2022	00:55:03		9
10/02/2022	00:55:03		9
10/02/2022	00:55:03		10
10/02/2022	00:55:03		12
10/02/2022	00:55:03		12
10/02/2022	00:55:03		12
10/02/2022	00:55:04		0
10/02/2022	00:55:04		12
10/02/2022	00:55:04		12
10/02/2022	00:55:04		12
10/02/2022	00:55:04		2
10/02/2022	00:55:04	Since the last update: I see a very distinct mouth shape, raised upper eyelids as well as very raised and frowned eyebrows	8
10/02/2022	00:55:05		8
10/02/2022	00:55:05		4
10/02/2022	00:55:05		7
10/02/2022	00:55:05		7
10/02/2022	00:55:05		7
10/02/2022	00:55:05		7
10/02/2022	00:55:06		2
10/02/2022	00:55:06		7
10/02/2022	00:55:06		7
10/02/2022	00:55:06		7
10/02/2022	00:55:06		7
10/02/2022	00:55:06		7
10/02/2022	00:55:07		1
10/02/2022	00:55:07		7
10/02/2022	00:55:07		7
10/02/2022	00:55:07		1
10/02/2022	00:55:07		7
10/02/2022	00:55:08		2
10/02/2022	00:55:08		0
10/02/2022	00:55:08	Since the last update: I see a strong change in mouth shape, very raised lip corners and raised lower eyelids	7
10/02/2022	00:55:08		7
10/02/2022	00:55:08		0
10/02/2022	00:55:08		0

Appendix K: SAM Facial Expression Level 3.2 Generated Report

10/02/2022 09:57:26

State Assessment Module (SAM) Report

Facial Expression Level 3.2 Classification - Discrete State Assessment

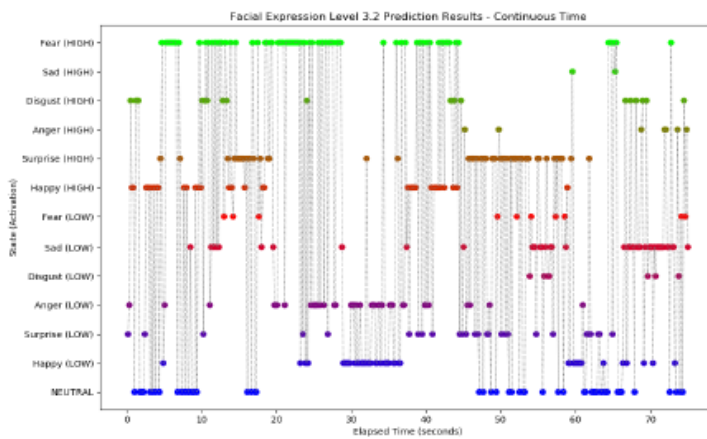
Generated By: Jordan Vice

Report Generated from: facial_GUI_3_2(MANUAL)_10-02-2022_09_57_21.csv

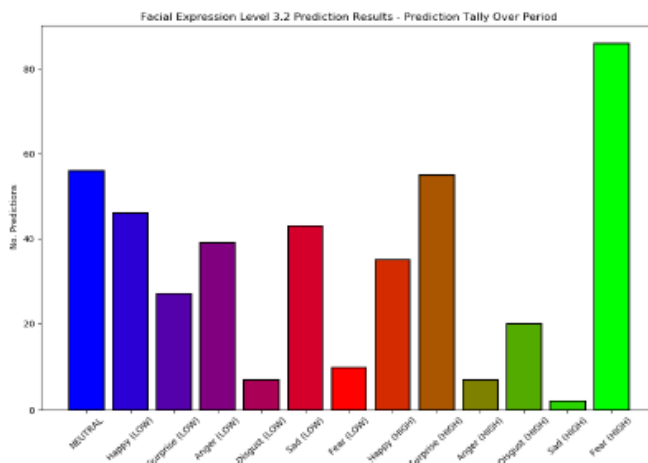
Log Time Interval: 09:56:06 --- 09:57:21

Elapsed Time (h:m:s): 0:1:15

Continuous changes in diiscrete affective state and activation over elapsed time:



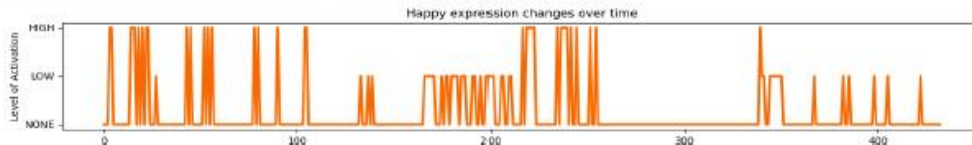
Total prediction tallies over elapsed time:



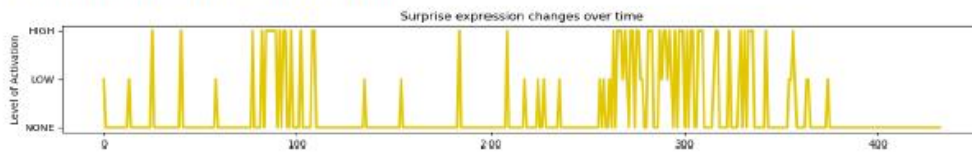
Mean State/Emotion (Prediction): 5.8245
 Most Common State: Fear (HIGH)
 Standard Deviation: 4.2417



Changes in Happiness Expressions over time



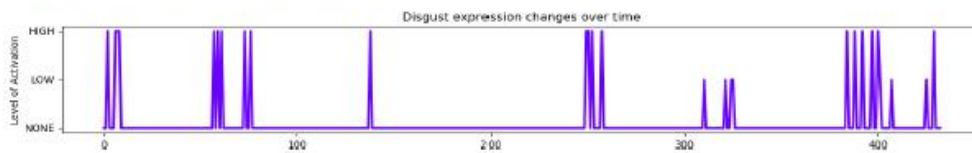
Changes in Surprise Expressions over time



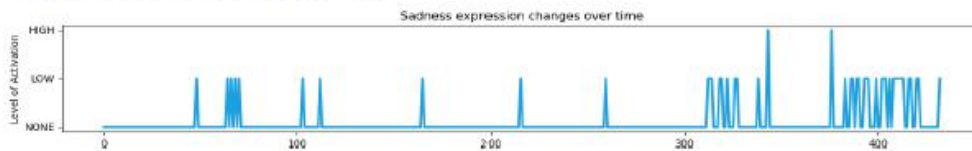
Changes in Anger Expressions over time



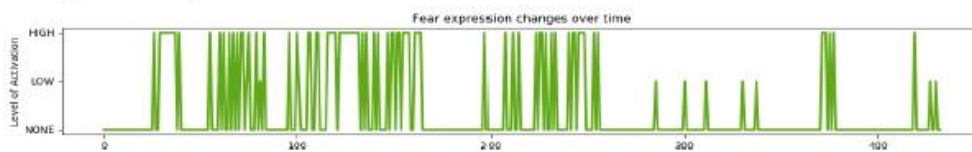
Changes in Disgust Expressions over time



Changes in Sadness Expressions over time



Changes in Fear Expressions over time



Corresponding Log Backup: facial_GUI_3_2(MANUAL)_10-02-2022_09_57_21.csv

DATE	TIME	LOG	PREDICTION INTEGER
10/02/2022	09:58:06		2
10/02/2022	09:58:06		3
10/02/2022	09:58:07		10
10/02/2022	09:58:07		7
10/02/2022	09:58:07		7
10/02/2022	09:58:07		0
10/02/2022	09:58:07		10
10/02/2022	09:58:07		10
10/02/2022	09:58:08		10
10/02/2022	09:58:08		0
10/02/2022	09:58:08		0
10/02/2022	09:58:08		0
10/02/2022	09:58:08		0
10/02/2022	09:58:09		2
10/02/2022	09:58:09	Since the last update: This person looks like they are in a neutral state	7
10/02/2022	09:58:09		7
10/02/2022	09:58:09		7
10/02/2022	09:58:09		0
10/02/2022	09:58:09		7
10/02/2022	09:58:10		0
10/02/2022	09:58:10		7
10/02/2022	09:58:10		0
10/02/2022	09:58:10		7
10/02/2022	09:58:10		7
10/02/2022	09:58:10		0
10/02/2022	09:58:11		8
10/02/2022	09:58:11		12
10/02/2022	09:58:11		1
10/02/2022	09:58:11		3
10/02/2022	09:58:11		12
10/02/2022	09:58:11		12
10/02/2022	09:58:12		12
10/02/2022	09:58:12		12
10/02/2022	09:58:12		12
10/02/2022	09:58:12	Since the last update: I predict that this person looks very scared	12
10/02/2022	09:58:12		12
10/02/2022	09:58:13		12
10/02/2022	09:58:13		12
10/02/2022	09:58:13		0
10/02/2022	09:58:13		12
10/02/2022	09:58:13		8
10/02/2022	09:58:13		0
10/02/2022	09:58:14		0
10/02/2022	09:58:14		7
10/02/2022	09:58:14		0
10/02/2022	09:58:14		7
10/02/2022	09:58:14		0
10/02/2022	09:58:14		0
10/02/2022	09:58:15		5
10/02/2022	09:58:15		0
10/02/2022	09:58:15		0
10/02/2022	09:58:15		0
10/02/2022	09:58:15		7
10/02/2022	09:58:15		0
10/02/2022	09:58:18	Since the last update: This person looks like they are in a neutral state	7
10/02/2022	09:58:18		12
10/02/2022	09:58:18		7
10/02/2022	09:58:18		10
10/02/2022	09:58:18		2
10/02/2022	09:58:18		10
10/02/2022	09:58:17		12
10/02/2022	09:58:17		10
10/02/2022	09:58:17		12
10/02/2022	09:58:17		3
10/02/2022	09:58:17		5
10/02/2022	09:58:17		12
10/02/2022	09:58:18		5
10/02/2022	09:58:18		12
10/02/2022	09:58:18		5
10/02/2022	09:58:18		12
10/02/2022	09:58:18		5

10/02/2022	09:58:19		12
10/02/2022	09:58:19		12
10/02/2022	09:58:19		10
10/02/2022	09:58:19	Since the last update: I predict that this person looks very scared	8
10/02/2022	09:58:19		12
10/02/2022	09:58:20		10
10/02/2022	09:58:20		8
10/02/2022	09:58:20		7
10/02/2022	09:58:20		12
10/02/2022	09:58:20		7
10/02/2022	09:58:20		8
10/02/2022	09:58:21		8
10/02/2022	09:58:21		12
10/02/2022	09:58:21		8
10/02/2022	09:58:21		8
10/02/2022	09:58:21		8
10/02/2022	09:58:21		8
10/02/2022	09:58:22		8
10/02/2022	09:58:22		8
10/02/2022	09:58:22		7
10/02/2022	09:58:22		8
10/02/2022	09:58:22		0
10/02/2022	09:58:22		8
10/02/2022	09:58:23	Since the last update: I predict that this person looks very surprised	8
10/02/2022	09:58:23		0
10/02/2022	09:58:23		12
10/02/2022	09:58:23		8
10/02/2022	09:58:23		0
10/02/2022	09:58:23		0
10/02/2022	09:58:24		12
10/02/2022	09:58:24		8
10/02/2022	09:58:24		8
10/02/2022	09:58:24		5
10/02/2022	09:58:24		7
10/02/2022	09:58:24		7
10/02/2022	09:58:25		12
10/02/2022	09:58:25		12
10/02/2022	09:58:25		8
10/02/2022	09:58:25		8
10/02/2022	09:58:25		12
10/02/2022	09:58:25		12
10/02/2022	09:58:26		5
10/02/2022	09:58:26		3
10/02/2022	09:58:26	Since the last update: I predict that this person looks very scared	3
10/02/2022	09:58:26		3
10/02/2022	09:58:26		12
10/02/2022	09:58:26		12
10/02/2022	09:58:27		12
10/02/2022	09:58:27		12
10/02/2022	09:58:27		12
10/02/2022	09:58:27		3
10/02/2022	09:58:27		12
10/02/2022	09:58:27		12
10/02/2022	09:58:28		12
10/02/2022	09:58:28		12
10/02/2022	09:58:28		12
10/02/2022	09:58:28		12
10/02/2022	09:58:28		12
10/02/2022	09:58:28		12
10/02/2022	09:58:29		12
10/02/2022	09:58:29		12
10/02/2022	09:58:29	Since the last update: I predict that this person looks very scared	1
10/02/2022	09:58:29		12
10/02/2022	09:58:30		2
10/02/2022	09:58:30		12
10/02/2022	09:58:30		1
10/02/2022	09:58:30		10
10/02/2022	09:58:30		1
10/02/2022	09:58:30		12
10/02/2022	09:58:31		3
10/02/2022	09:58:31		12

Appendix L: SAM Facial Expression Level 4 Generated Report

10/02/2022 09:59:44

State Assessment Module (SAM) Report

Facial Expression Level 4 Classification - Discrete State and Temporal Phase

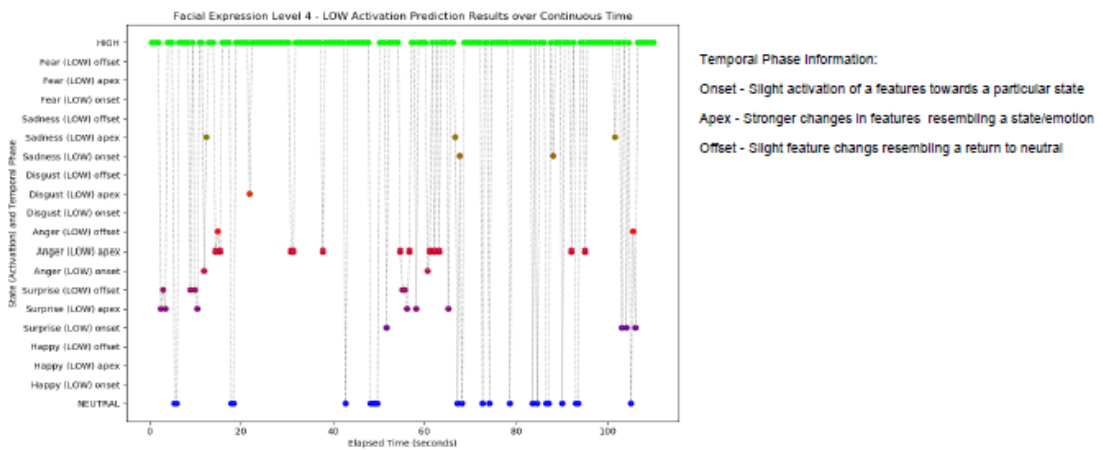
Generated By: Jordan Vice

Report Generated from: facial_GUI_4(MANUAL)_10-02-2022_09_59_38.csv

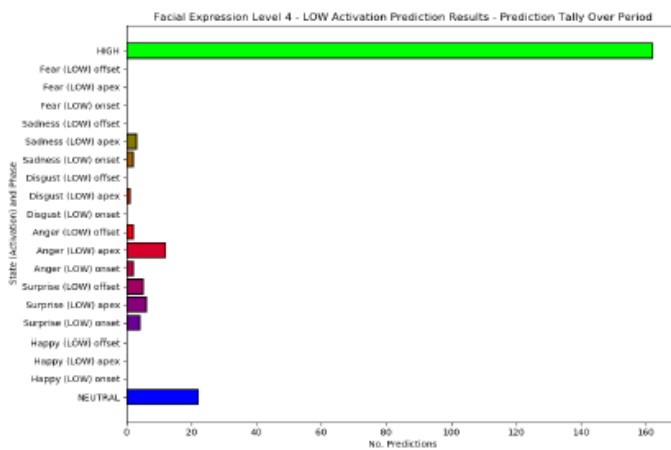
Log Time Interval: 09:58:48 --- 09:59:38

Elapsed Time (h:m:s): 0:1:50

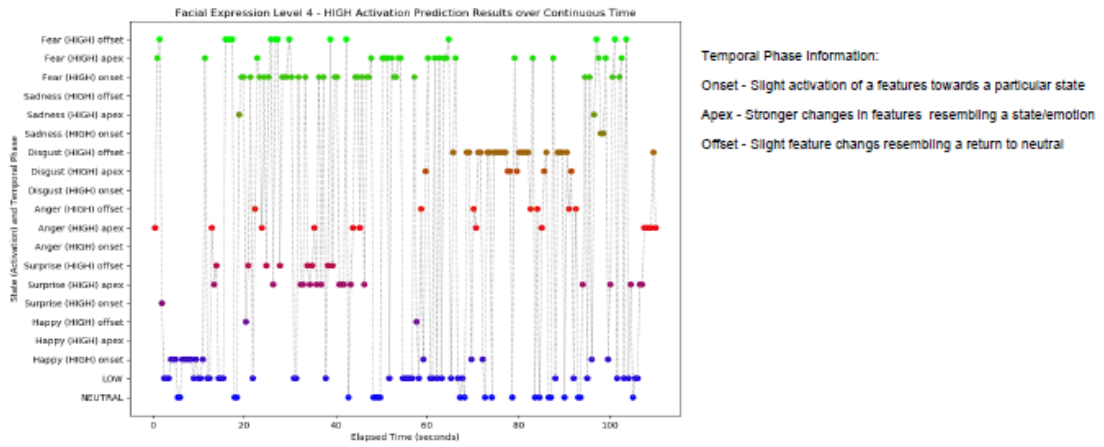
Continuous changes in LOW Activation discrete affective states and their changes in temporal phase:



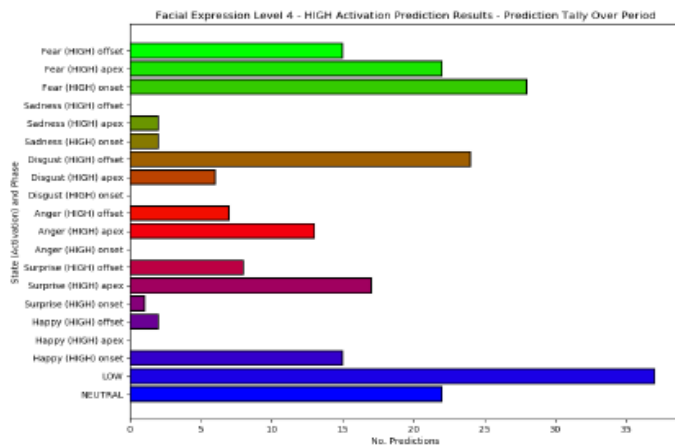
Total LOW Activation prediction tallies over elapsed time:



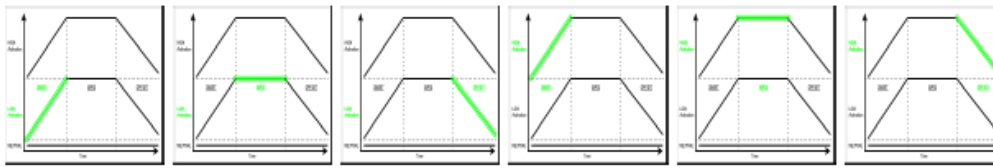
Continuous changes in HIGH Activation discrete affective states and their changes in temporal phase:



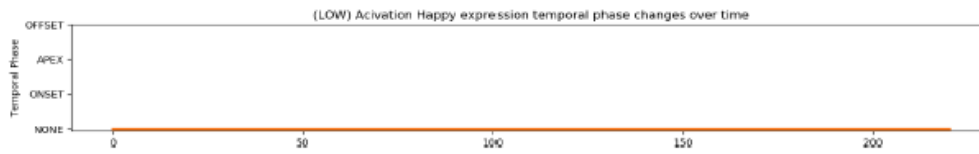
Total HIGH Activation prediction tallies over elapsed time:



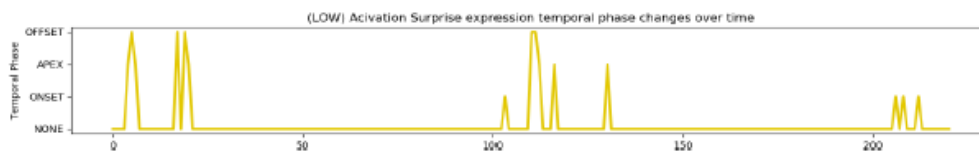
Mean Prediction Integer (State+Activation+Phase): 22.7330
 Most Common Prediction: Fear (HIGH) onset
 Standard Deviation: 12.0153



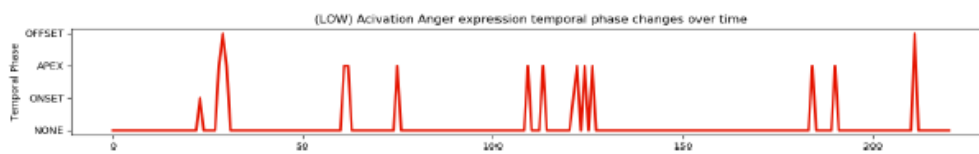
Changes in Low Activation Happiness Temporal Phase



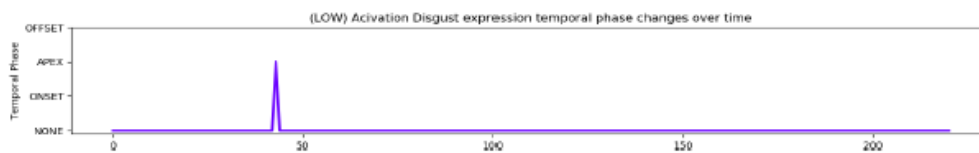
Changes in Low Activation Surprise Temporal Phase



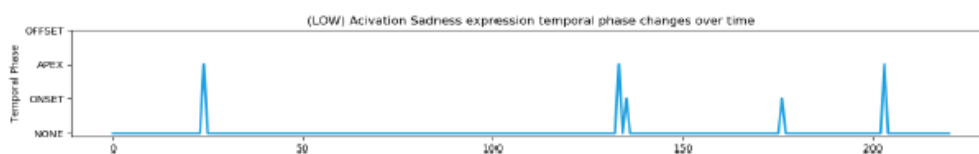
Changes in Low Activation Anger Temporal Phase



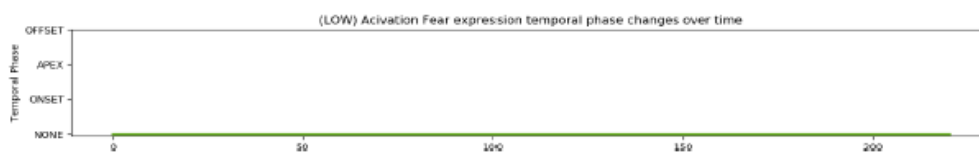
Changes in Low Activation Disgust Temporal Phase



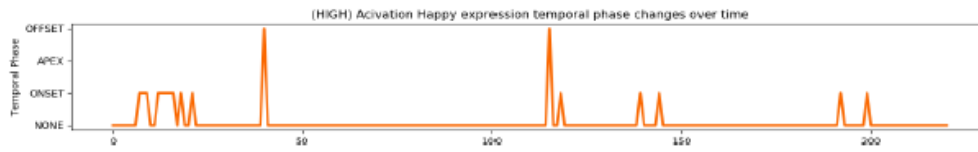
Changes in Low Activation Sadness Temporal Phase



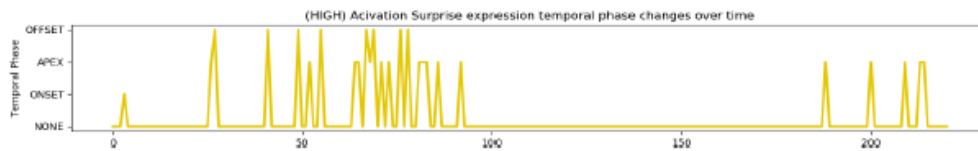
Changes in Low Activation Fear Temporal Phase



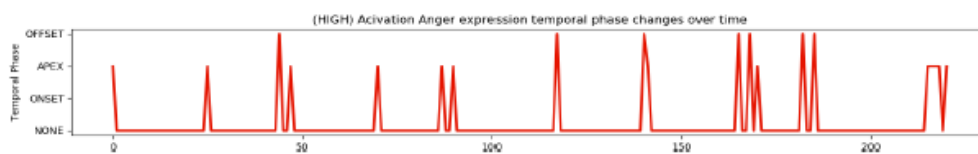
Changes in High Activation Happiness Temporal Phase



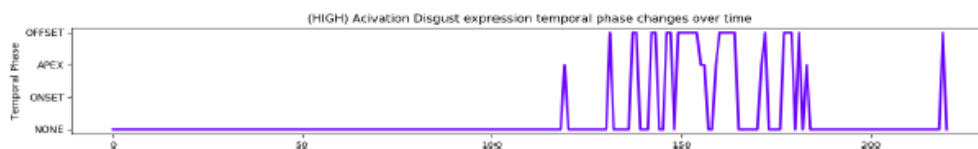
Changes in High Activation Surprise Temporal Phase



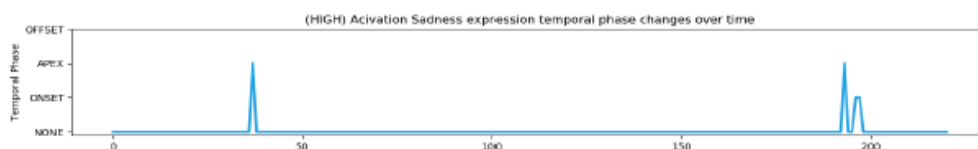
Changes in High Activation Anger Temporal Phase



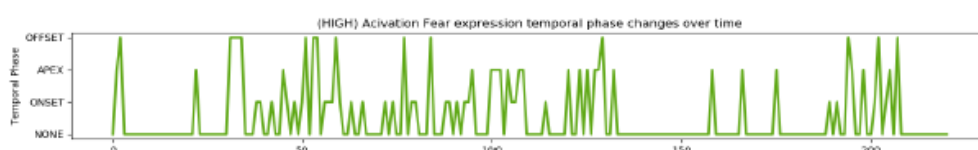
Changes in High Activation Disgust Temporal Phase



Changes in High Activation Sadness Temporal Phase

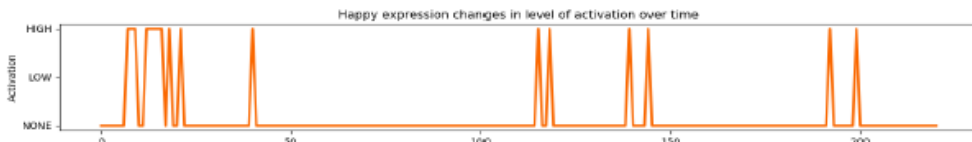


Changes in High Activation Fear Temporal Phase

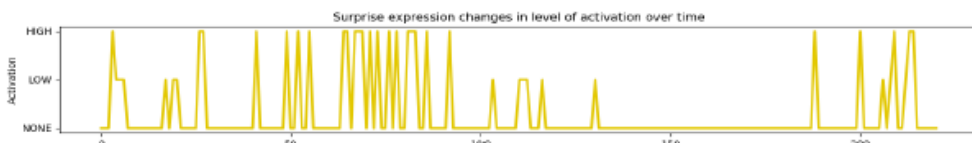




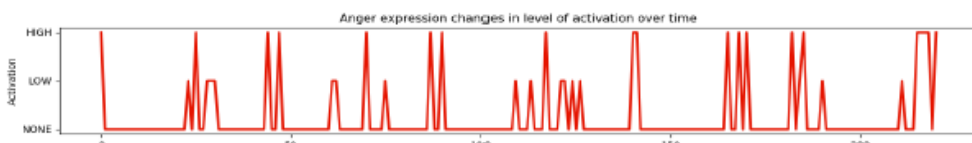
Phase-Independent Changes in Happiness Activation



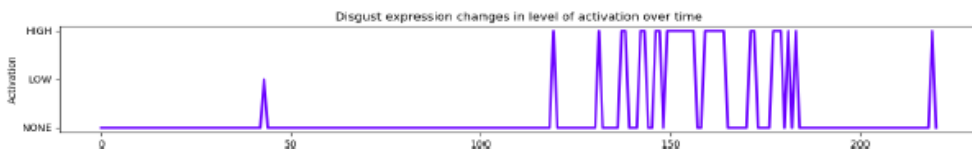
Phase-Independent Changes in Surprise Activation



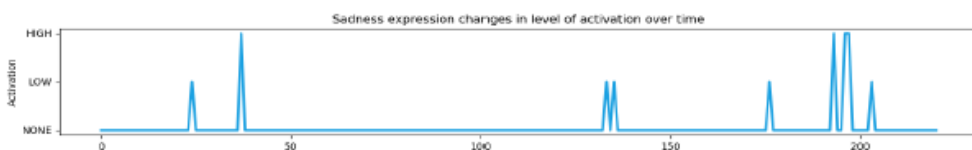
Phase-Independent Changes in Anger Activation



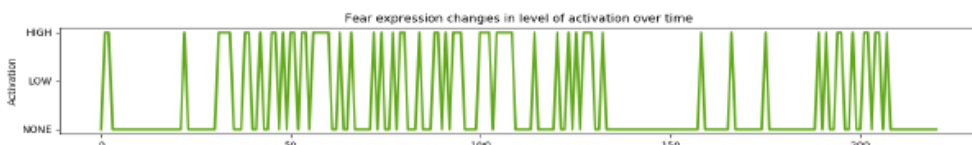
Phase-Independent Changes in Disgust Activation



Phase-Independent Changes in Sadness Activation



Phase-Independent Changes in Fear Activation



Corresponding Log Backup: facial_GUI_4(MANUAL)_10-02-2022_09_59_38.csv

DATE	TIME	LOG	PREDICTION INTEGER
10/02/2022	09:58:48		28
10/02/2022	09:58:49		35
10/02/2022	09:58:49		38
10/02/2022	09:58:49		22
10/02/2022	09:58:50		5
10/02/2022	09:58:50		8
10/02/2022	09:58:50		5
10/02/2022	09:58:50		19
10/02/2022	09:58:50		19
10/02/2022	09:58:51		19
10/02/2022	09:58:51		0
10/02/2022	09:58:51		0
10/02/2022	09:58:51		19
10/02/2022	09:58:51		19
10/02/2022	09:58:52		19
10/02/2022	09:58:52		19
10/02/2022	09:58:52		19
10/02/2022	09:58:52		8
10/02/2022	09:58:52		8
10/02/2022	09:58:52		19
10/02/2022	09:58:53		8
10/02/2022	09:58:53		5
10/02/2022	09:58:53		19
10/02/2022	09:58:53		35
10/02/2022	09:58:53		7
10/02/2022	09:58:54		14
10/02/2022	09:58:54	Since the last update: This person looks like they are approaching a very happy state	28
10/02/2022	09:58:54		23
10/02/2022	09:58:54		24
10/02/2022	09:58:55		8
10/02/2022	09:58:55		9
10/02/2022	09:58:55		8
10/02/2022	09:58:55		38
10/02/2022	09:58:55		38
10/02/2022	09:58:56		38
10/02/2022	09:58:56		38
10/02/2022	09:58:56		0
10/02/2022	09:58:56		0
10/02/2022	09:58:56		32
10/02/2022	09:58:57		34
10/02/2022	09:58:57		34
10/02/2022	09:58:58	Since the last update: This person looks like they are exiting a very fearful state	21
10/02/2022	09:58:58		24
10/02/2022	09:58:58		34
10/02/2022	09:58:58		11
10/02/2022	09:58:58		27
10/02/2022	09:58:59		35
10/02/2022	09:58:59		34
10/02/2022	09:58:59		28
10/02/2022	09:59:00		34
10/02/2022	09:59:00		24
10/02/2022	09:59:00		34
10/02/2022	09:59:01		38
10/02/2022	09:59:01		23
10/02/2022	09:59:01	Since the last update: This person looks like they are approaching a very fearful state	38
10/02/2022	09:59:01		38
10/02/2022	09:59:01		24
10/02/2022	09:59:02		34
10/02/2022	09:59:02		34
10/02/2022	09:59:02		34
10/02/2022	09:59:02		38
10/02/2022	09:59:04		34
10/02/2022	09:59:04		8
10/02/2022	09:59:04		8
10/02/2022	09:59:04		34
10/02/2022	09:59:04		23
10/02/2022	09:59:04		23
10/02/2022	09:59:05		34
10/02/2022	09:59:05		24
10/02/2022	09:59:05		23
10/02/2022	09:59:05		24
10/02/2022	09:59:08		28

10/02/2022	00:59:06		23
10/02/2022	00:59:06		34
10/02/2022	00:59:06		23
10/02/2022	00:59:06		34
10/02/2022	00:59:06		8
10/02/2022	00:59:07		24
10/02/2022	00:59:07		36
10/02/2022	00:59:07		24
10/02/2022	00:59:08		34
10/02/2022	00:59:09		34
10/02/2022	00:59:09		23
10/02/2022	00:59:09		23
10/02/2022	00:59:09		23
10/02/2022	00:59:09		38
10/02/2022	00:59:10		0
10/02/2022	00:59:10		23
10/02/2022	00:59:10		28
10/02/2022	00:59:10		34
10/02/2022	00:59:10		34
10/02/2022	00:59:11		28
10/02/2022	00:59:11		34
10/02/2022	00:59:11		23
10/02/2022	00:59:11		34
10/02/2022	00:59:11		34
10/02/2022	00:59:12		35
10/02/2022	00:59:12	Since the last update: This person looks like they are approaching a very fearful state	0
10/02/2022	00:59:12		0
10/02/2022	00:59:12		0
10/02/2022	00:59:12		0
10/02/2022	00:59:12		35
10/02/2022	00:59:13		35
10/02/2022	00:59:13		35
10/02/2022	00:59:13		4
10/02/2022	00:59:13		35
10/02/2022	00:59:13		34
10/02/2022	00:59:13		34
10/02/2022	00:59:14		35
10/02/2022	00:59:14		35
10/02/2022	00:59:14		8
10/02/2022	00:59:14		8
10/02/2022	00:59:14		8
10/02/2022	00:59:15		5
10/02/2022	00:59:15		8
10/02/2022	00:59:15		34
10/02/2022	00:59:16		21
10/02/2022	00:59:17		5
10/02/2022	00:59:17		27
10/02/2022	00:59:17		19
10/02/2022	00:59:18		29
10/02/2022	00:59:18		35
10/02/2022	00:59:18		7
10/02/2022	00:59:18		8
10/02/2022	00:59:18		35
10/02/2022	00:59:18		8
10/02/2022	00:59:19		35
10/02/2022	00:59:19	Since the last update: This person looks like they are in a mild state of anger	8
10/02/2022	00:59:19		35
10/02/2022	00:59:19		35
10/02/2022	00:59:19		38
10/02/2022	00:59:20		5
10/02/2022	00:59:20		30
10/02/2022	00:59:20		35
10/02/2022	00:59:20		14
10/02/2022	00:59:20		0
10/02/2022	00:59:20		13
10/02/2022	00:59:21		0
10/02/2022	00:59:21		30
10/02/2022	00:59:21		30
10/02/2022	00:59:21		19
10/02/2022	00:59:21		27
10/02/2022	00:59:21		28
10/02/2022	00:59:22		30

Appendix M: Collated SAM Facial Expression classifier dialogue responses for all windows

LEVEL 1 – Neutral + Level of Activation				
No.	HIGH/LOW	Dialogue and Reasoning		
1	NEUTRAL	This person looks like they are in a neutral state, due to the lack of noticeable movement around their face.		
2	LOW	This person looks like they are in a low activation state due to the minor, noticeable muscle movements around their face.		
3	HIGH	This person looks like they are in a high activation state, as I can see strong muscle movements around their face.		
LEVEL 2.1 – Generalized Muscle Movements				
No.	HIGH/LOW	State Group	Dialogue and Reasoning	
1	LOW	Happy Surprise	I see a slight change in mouth shape, and a combination of weak, lower and upper eyelid movements.	
2	LOW	Anger Disgust	I see slightly raised lower eyelids, and slightly frowned eyebrows.	
3	LOW	Sadness Fear	I see slight movements around the shape of their brows, specifically being slightly raised and frowned.	
4	HIGH	Happy Surprise	I see a strong change in mouth shape and a combination of strong lower and upper eyelid movements.	
5	HIGH	Anger Disgust	I see strong raised lower eyelids and very frowned eyebrows.	
6	HIGH	Sadness Fear	I see strong movements around the shape of their brows, specifically noting they are raised and frowned.	
LEVEL 2.2 – General State Group				
No.	HIGH/LOW	State Group	Dialogue and Reasoning	
1	LOW	Happy Surprise	This person looks like they are either mildly happy, or mildly surprised.	
2	LOW	Anger Disgust	This person looks like they are either mildly angry, or mildly disgusted.	
3	LOW	Sadness Fear	This person looks like they are either mildly sad, or mildly scared.	
4	HIGH	Happy Surprise	This person looks like they are either very happy, or very surprised.	
5	HIGH	Anger Disgust	This person looks like they are either very angry, or very disgusted.	
6	HIGH	Sadness Fear	This person looks like they are either very sad, or very scared.	
LEVEL 3.1 – Extended Muscle Movements				
No.	HIGH/LOW	State Group	State	Dialogue and Reasoning
1	LOW	Happy Surprise	Happy	I see minor changes in mouth shape, as well as slightly raised lip corners, and lower eyelids.
2	LOW	Happy Surprise	Surprise	I see a slightly open mouth, and slightly raised upper eyelids and eyebrows.
3	LOW	Anger Disgust	Anger	I see tightened lips, slightly raised upper and lower eyelids, and some indication of frowned eyebrows.
4	LOW	Anger Disgust	Disgust	I see a slightly raised upper lip, as well as raised lower eyelids and frowned eyebrows.
5	LOW	Sadness Fear	Sad	I see slightly lowered lip corners, as well as some indications of raised and frowned eyebrows.
6	LOW	Sadness Fear	Fear	I see a slightly open mouth, raised upper eyelids, and some indication of raised and frowned eyebrows.
7	HIGH	Happy Surprise	Happy	I see a strong change in mouth shape, very raised lip corners, and raised lower eyelids.
8	HIGH	Happy Surprise	Surprise	I see a very open mouth, and very raised upper eyelids and eyebrows.

9	HIGH	Anger Disgust	Anger	I see very tightened lips, very raised upper and lower eyelids, and frowned eyebrows.
10	HIGH	Anger Disgust	Disgust	I see a strong raised upper lip, as well as raised lower eyelids, and frowned eyebrows.
11	HIGH	Sadness Fear	Sad	I see strong lowered lip corners, as well as very raised and frowned eyebrows.
12	HIGH	Sadness Fear	Fear	I see a very distinct mouth shape, raised upper eyelids, as well as very raised and frowned eyebrows.

LEVEL 3.2 – Discrete State

No.	HIGH/LOW	State Group	State	Dialogue and Reasoning
1	LOW	Happy Surprise	Happy	I see minor changes in mouth shape, as well as slightly raised lip corners, and lower eyelids.
2	LOW	Happy Surprise	Surprise	I see a slightly open mouth, and slightly raised upper eyelids and eyebrows.
3	LOW	Anger Disgust	Anger	I see tightened lips, slightly raised upper and lower eyelids, and some indication of frowned eyebrows.
4	LOW	Anger Disgust	Disgust	I see a slightly raised upper lip, as well as raised lower eyelids and frowned eyebrows.
5	LOW	Sadness Fear	Sad	I see slightly lowered lip corners, as well as some indications of raised and frowned eyebrows.
6	LOW	Sadness Fear	Fear	I see a slightly open mouth, raised upper eyelids, and some indication of raised and frowned eyebrows.
7	HIGH	Happy Surprise	Happy	I see a strong change in mouth shape, very raised lip corners, and raised lower eyelids.
8	HIGH	Happy Surprise	Surprise	I see a very open mouth, and very raised upper eyelids and eyebrows.
9	HIGH	Anger Disgust	Anger	I see very tightened lips, very raised upper and lower eyelids, and frowned eyebrows.
10	HIGH	Anger Disgust	Disgust	I see a strong raised upper lip, as well as raised lower eyelids, and frowned eyebrows.
11	HIGH	Sadness Fear	Sad	I see strong lowered lip corners, as well as very raised and frowned eyebrows.
12	HIGH	Sadness Fear	Fear	I see a very distinct mouth shape, raised upper eyelids, as well as very raised and frowned eyebrows.

Level 4 – Temporal Phase: Verbose = 0

No.	HIGH/LOW	State Group	State	Phase	Dialogue and Reasoning
1	LOW	Happy Surprise	Happy	Onset	This person looks like they are approaching a mildly happy state.
2	LOW	Happy Surprise	Happy	Apex	This person looks like they are showing some signs of being mildly happy.
3	LOW	Happy Surprise	Happy	Offset	This person looks like they are exiting a mildly happy state.
4	LOW	Happy Surprise	Surprise	Onset	This person looks like they are approaching a mildly surprised state.
5	LOW	Happy Surprise	Surprise	Apex	This person looks like they are showing some signs of being mildly surprised.
6	LOW	Happy Surprise	Surprise	Offset	This person looks like they are exiting a mildly surprised state.
7	LOW	Anger Disgust	Anger	Onset	This person looks like they are approaching a mild state of anger.
8	LOW	Anger Disgust	Anger	Apex	This person looks like they are in a mild state of anger.
9	LOW	Anger Disgust	Anger	Offset	This person looks like they are exiting a mild state of anger.
10	LOW	Anger Disgust	Disgust	Onset	This person looks like they are approaching a mild state of disgust.
11	LOW	Anger Disgust	Disgust	Apex	This person looks like they are in a mildly disgusted state.
12	LOW	Anger Disgust	Disgust	Offset	This person looks like they are exiting a state of mild disgust.
13	LOW	Sadness Fear	Sadness	Onset	This person looks like they are approaching a mildly sad state.

14	LOW	Sadness Fear	Sadness	Apex	This person looks like they are mildly sad.
15	LOW	Sadness Fear	Sadness	Offset	This person looks like they are exiting a slight mild of sadness.
16	LOW	Sadness Fear	Fear	Onset	This person looks like they are approaching a mild state of fear.
17	LOW	Sadness Fear	Fear	Apex	This person looks like they are experiencing a mild sense of far.
18	LOW	Sadness Fear	Fear	Offset	This person looks like they are exiting a mild state of fear.
19	HIGH	Happy Surprise	Happy	Onset	This person looks like they are approaching a very happy state.
20	HIGH	Happy Surprise	Happy	Apex	This person looks like they are very happy.
21	HIGH	Happy Surprise	Happy	Offset	This person looks like they are exiting a very happy state.
22	HIGH	Happy Surprise	Surprise	Onset	This person looks like they are approaching a very surprised state.
23	HIGH	Happy Surprise	Surprise	Apex	This person looks like they are very surprised.
24	HIGH	Happy Surprise	Surprise	Offset	This person looks like they are exiting a very surprised state.
25	HIGH	Anger Disgust	Anger	Onset	This person looks like they are approaching a very angry state.
26	HIGH	Anger Disgust	Anger	Apex	This person looks like they are very angry.
27	HIGH	Anger Disgust	Anger	Offset	This person looks like they are exiting a very angry state.
28	HIGH	Anger Disgust	Disgust	Onset	This person looks like they are approaching a very disgusted state.
29	HIGH	Anger Disgust	Disgust	Apex	This person looks like they are very disgusted.
30	HIGH	Anger Disgust	Disgust	Offset	This person looks like they are exiting a very disgusted state.
31	HIGH	Sadness Fear	Sadness	Onset	This person looks like they are approaching a very sad state.
32	HIGH	Sadness Fear	Sadness	Apex	This person looks like they are very sad.
33	HIGH	Sadness Fear	Sadness	Offset	This person looks like they are exiting a very sad state.
34	HIGH	Sadness Fear	Fear	Onset	This person looks like they are approaching a very fearful state.
35	HIGH	Sadness Fear	Fear	Apex	This person looks like they are in a high state of fear.
36	HIGH	Sadness Fear	Fear	Offset	This person looks like they are exiting a very fearful state.

Level 4 – Temporal Phase: Verbose = 1

No.	HIGH/LOW	State Group	State	Phase	Dialogue and Reasoning
1	LOW	Happy Surprise	Happy	Onset	This person looks like they are approaching a mild happy state, which I can see from the movement around the mouth, and general eyelid region, as well as the lip corners starting to raise.
2	LOW	Happy Surprise	Happy	Apex	This person looks like they are showing some signs of being happy by looking at their mouth, raised lower eyelids, and raised lip corners.
3	LOW	Happy Surprise	Happy	Offset	This person looks like they are exiting a mild happy state, which I can tell by looking at the movements around the mouth and general eyelid region, as well as their lip corners returning to rest.
4	LOW	Happy Surprise	Surprise	Onset	This person looks like they are approaching a mildly surprised state, which I can see from the movement around the mouth, eyelid, and brow region.

5	LOW	Happy Surprise	Surprise	Apex	This person looks like they are showing some signs of being surprised, which I can see from the shape of their mouth, raised upper eyelids, and raised eyebrows.
6	LOW	Happy Surprise	Surprise	Offset	This person looks like they are exiting a mild surprised state, looking at the movements around the general eyelid region, as well as their brow returning to rest.
7	LOW	Anger Disgust	Anger	Onset	This person looks like they are approaching a mild state of anger, which I can see from the movement around their eyes, eyebrows, and lips.
8	LOW	Anger Disgust	Anger	Apex	This person looks like they are in a mild state of anger by looking at their tightened lip, raised upper and lower eyelids, and frowned brows.
9	LOW	Anger Disgust	Anger	Offset	This person looks like they are exiting a mild state of anger when I analyse the movements around their eyes, eyebrows, and lips as they appear to be returning to a rest position.
10	LOW	Anger Disgust	Disgust	Onset	This person looks like they are approaching a mild state of disgust, as I see some signs of movements around their eyes, eyebrows, and lips.
11	LOW	Anger Disgust	Disgust	Apex	This person looks like they are in a state of mild disgust, which I can see from their raised upper lip, raised lower eyelids, and frowned brows.
12	LOW	Anger Disgust	Disgust	Offset	This person looks like they are exiting a state of mild disgust, since the movements around their eyes, eyebrows, and lips look as though they are returning to a resting position.
13	LOW	Sadness Fear	Sadness	Onset	This person looks like they are approaching a mildly sad state, as I see some signs of movement around the brows in particular.
14	LOW	Sadness Fear	Sadness	Apex	This person looks like they are quite sad, by looking at their lowered lip corners as well as their raised and frowned brows.
15	LOW	Sadness Fear	Sadness	Offset	This person looks like they are exiting a mild state of sadness, by looking at the movements around their brows and lips, as they return to a resting position.
16	LOW	Sadness Fear	Fear	Onset	This person looks like they are approaching a mild state of fear, as I see some signs of movement around their eyes and brows.
17	LOW	Sadness Fear	Fear	Apex	This person looks like they are experiencing fear, by looking at the shape of their mouth, raised and frowned eyebrows as well as their raised upper eyelids.
18	LOW	Sadness Fear	Fear	Offset	This person looks like they are exiting a state of fear, as analysing the movement around their eyes and mouth show that they are returning to a neutral position.
19	HIGH	Happy Surprise	Happy	Onset	This person looks like they are approaching a very happy state, as I see strong movements around the mouth, and general eyelid region, as well as the lip corners starting to raise.
20	HIGH	Happy Surprise	Happy	Apex	This person looks like they are very happy, by looking at the shape of their mouth, raised lower eyelids and raised lip corners.
21	HIGH	Happy Surprise	Happy	Offset	This person looks like they are exiting a very happy state, by looking at the movements around the mouth and general eyelid region, as well as their lip corners returning to rest.
22	HIGH	Happy Surprise	Surprise	Onset	This person looks like they are approaching a very surprised state, because I see strong movements around the mouth and general eyelid region, as well as their eyebrows starting to raise.
23	HIGH	Happy Surprise	Surprise	Apex	This person looks like they are very surprised by looking at the shape of their mouth, raised upper eyelids, and raised eyebrows.

24	HIGH	Happy Surprise	Surprise	Offset	This person looks like they are exiting a very surprised state, looking at the movements around the general eyelid region, as well as their eyebrows returning to rest.
25	HIGH	Anger Disgust	Anger	Onset	This person looks like they are approaching a very angry state, as I see strong movements around their eyes, eyebrows, and lips.
26	HIGH	Anger Disgust	Anger	Apex	This person looks like they are very angry, by looking at their tightened lip, raised upper and lower eyelids and frowned brows.
27	HIGH	Anger Disgust	Anger	Offset	This person looks like they are exiting a very angry state, as I look at the movements around their eyes, eyebrows, and lips. They appear to be returning to rest.
28	HIGH	Anger Disgust	Disgust	Onset	This person looks like they are approaching a very disgusted state, looking at the strong movements around their eyes, eyebrows, and lips.
29	HIGH	Anger Disgust	Disgust	Apex	This person looks like they are very disgusted, looking at their raised upper lip, raised lower eyelids and frowned brows.
30	HIGH	Anger Disgust	Disgust	Offset	This person looks like they are exiting a very disgusted state, as I see the movements around their eyes, eyebrows, and lips appear to be returning to rest.
31	HIGH	Sadness Fear	Sadness	Onset	This person looks like they are approaching a very sad state, looking at the strong movement around the brows in particular.
32	HIGH	Sadness Fear	Sadness	Apex	This person looks like they are very sad, looking at their lowered lip corners as well as their raised and frowned brows.
33	HIGH	Sadness Fear	Sadness	Offset	This person looks like they are exiting a very sad state, looking at the movements around their brows and lips as they return to a resting position.
34	HIGH	Sadness Fear	Fear	Onset	This person looks like they are approaching a very fearful state, looking at the strong movement around their eyes and brows changing from rest.
35	HIGH	Sadness Fear	Fear	Apex	This person looks like they are in a high state of fear, as I can see the shape of their mouth, raised and frowned eyebrows as well as their raised upper eyelids.
36	HIGH	Sadness Fear	Fear	Offset	This person looks like they are exiting a very fearful state, as I can see the movement around their eyes and mouth, as they return to a neutral position.

Corresponding Log Backup: vocal_GUI_1(MANUAL)_10-02-2022_10_00_53.csv

DATE	TIME	LOG	PREDICTION INTEGER
10/02/2022	10:00:06	This person sounds like they are in a low activation state due to the minor, but noticeable changes in speech features that I sense	1
10/02/2022	10:00:09	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:12	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:15	This person sounds like they are in a high activation state due to the very strong variations in speech features that I sense	2
10/02/2022	10:00:19	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:22	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:25	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:29	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:32	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:35	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:39	This person sounds like they are in a neutral state due to the lack of changes in speech features	0
10/02/2022	10:00:42	This person sounds like they are in a low activation state due to the minor, but noticeable changes in speech features that I sense	1
10/02/2022	10:00:45	This person sounds like they are in a low activation state due to the minor, but noticeable changes in speech features that I sense	1
10/02/2022	10:00:48	This person sounds like they are in a high activation state due to the very strong variations in speech features that I sense	2
10/02/2022	10:00:52	This person sounds like they are in a low activation state due to the minor, but noticeable changes in speech features that I sense	1

Appendix O: SAM Affective Speech Assessment Level 2 Generated Report

10/02/2022 10:02:22

State Assessment Module (SAM) Report

Vocal Expression Level 2 Classification - Temporal Phase

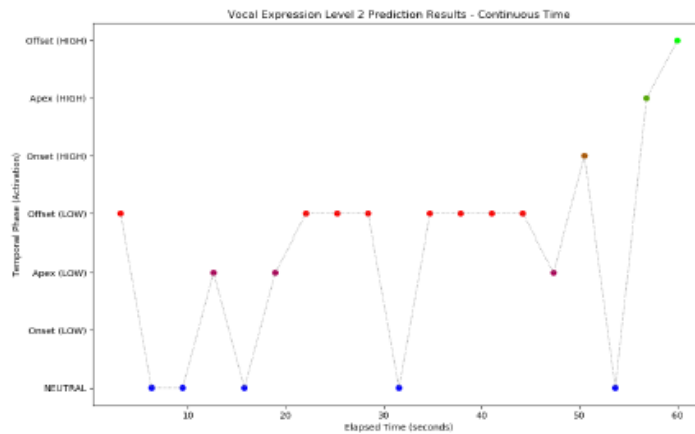
Generated By: Jordan Vice

Report Generated from: vocal_GUI_2(MANUAL)_10-02-2022_10_02_16.csv

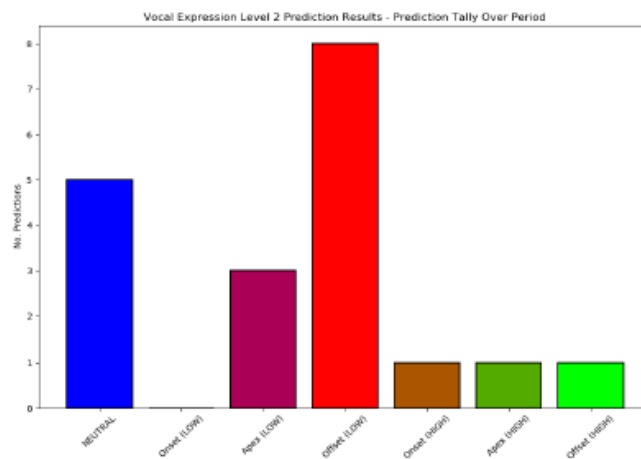
Log Time Interval: 10:01:14 --- 10:02:14

Elapsed Time (h:m:s): 0:1:0

Continuous changes in temporal phase over elapsed time:



Total prediction tally over elapsed time:



Mean Temporal Phase(Prediction Integer): 2.3684
 Most Common Temporal Phase: Offset (LOW)
 Standard Deviation: 1.6924

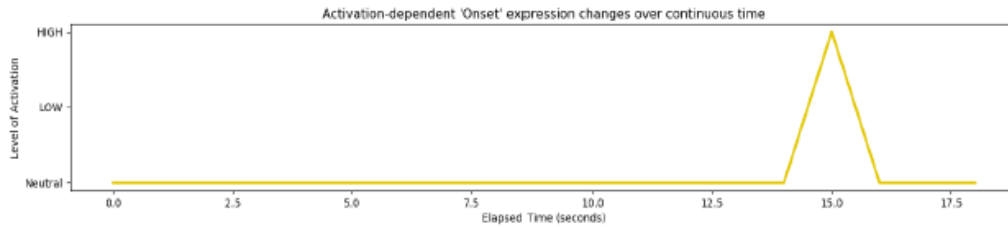
Temporal Phase Information:

Onset - Slight activation of a features towards a particular state

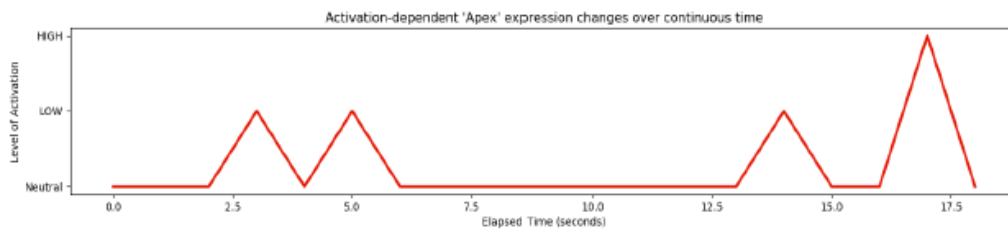
Apex - Stronger changes in features resembling a state/emotion

Offset - Slight feature changes resembling a return to neutral

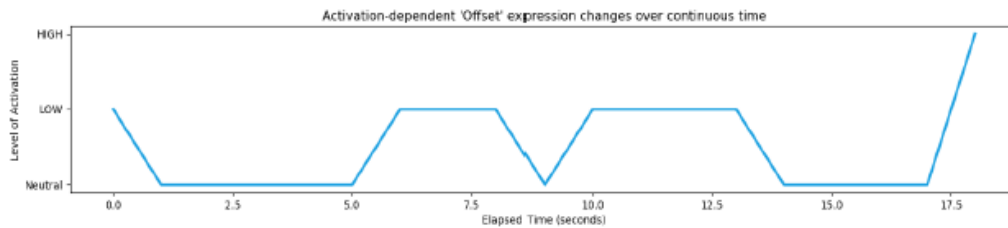
(Onset) i.e. 'activating toward' or 'experiencing some signs of' or 'deviating from the norm'



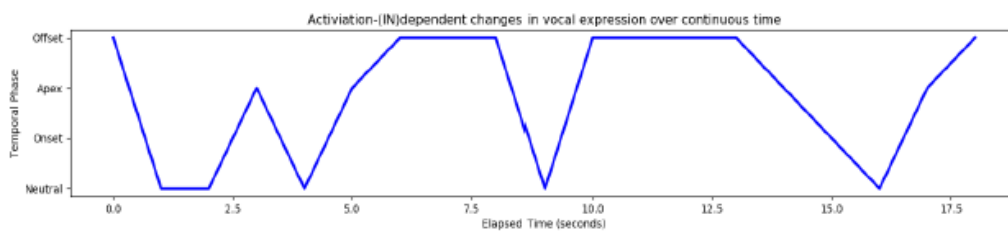
(Apex) i.e. 'at a particular (low/high) level of activation' or 'experiencing some state X'



(Offset) i.e. 'Coming down' or 'deactivating' or 'returning to normal'



Activation-Independent i.e. treating onset(LOW/HIGH) as one class irrespective of activation level



Corresponding Log Backup: vocal_GUI_2(MANUAL)_10-02-2022_10_02_16.csv

DATE	TIME	LOG	PREDICTION INTEGER
10/02/2022	10:01:14	This person sounds like they are exiting a mildly activated state	3
10/02/2022	10:01:17	This person sounds like they are in a neutral state	0
10/02/2022	10:01:20	This person sounds like they are in a neutral state	0
10/02/2022	10:01:24	This person sounds like they are currently experiencing a mildly activated state	2
10/02/2022	10:01:27	This person sounds like they are in a neutral state	0
10/02/2022	10:01:31	This person sounds like they are currently experiencing a mildly activated state	2
10/02/2022	10:01:34	This person sounds like they are exiting a mildly activated state	3
10/02/2022	10:01:37	This person sounds like they are exiting a mildly activated state	3
10/02/2022	10:01:41	This person sounds like they are exiting a mildly activated state	3
10/02/2022	10:01:44	This person sounds like they are in a neutral state	0
10/02/2022	10:01:47	This person sounds like they are exiting a mildly activated state	3
10/02/2022	10:01:50	This person sounds like they are exiting a mildly activated state	3
10/02/2022	10:01:54	This person sounds like they are exiting a mildly activated state	3
10/02/2022	10:01:57	This person sounds like they are exiting a mildly activated state	3
10/02/2022	10:02:00	This person sounds like they are currently experiencing a mildly activated state	2
10/02/2022	10:02:04	This person sounds like they are approaching a very activated state	4
10/02/2022	10:02:07	This person sounds like they are in a neutral state	0
10/02/2022	10:02:11	This person sounds like they are currently experiencing a very activated state	5
10/02/2022	10:02:14	This person sounds like they are exiting a very activated state	6

Appendix P: SAM Affective Speech Assessment Level 3 Generated Report

10/02/2022 10:04:49

State Assessment Module (SAM) Report

Vocal Expression Level 3 Classification - Valence/State Group

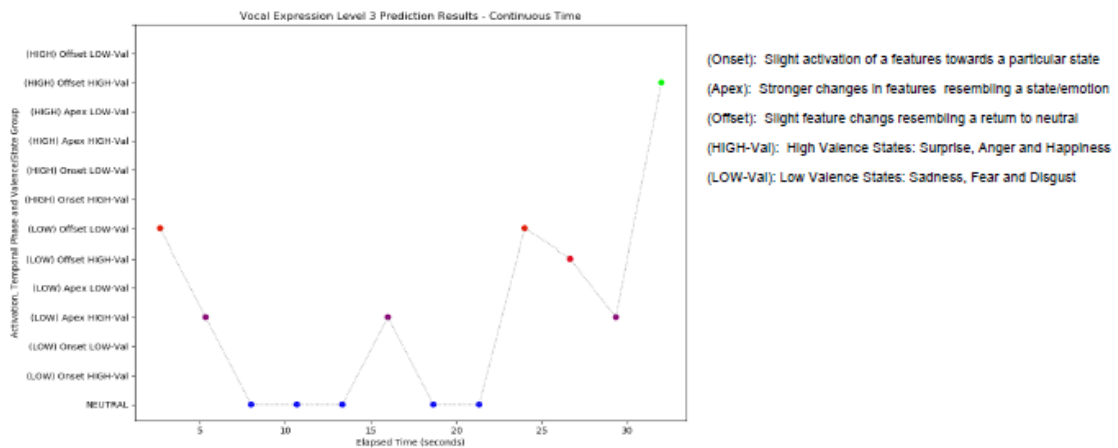
Generated By: Jordan Vice

Report Generated from: vocal_GUI_3(MANUAL)_10-02-2022_10_04_44.csv

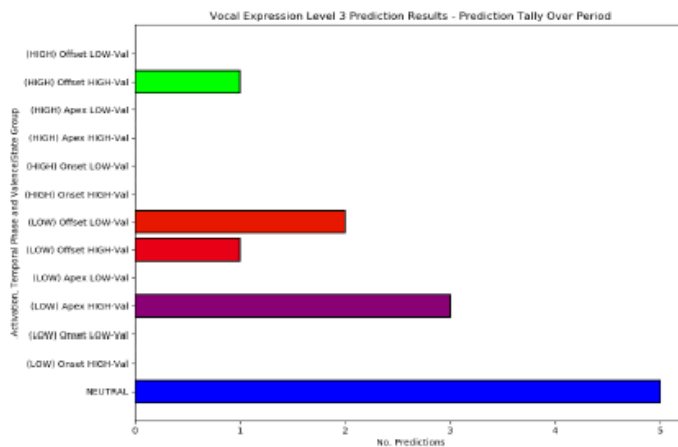
Log Time Interval: 10:04:10 --- 10:04:42

Elapsed Time (h:m:s): 0:0:32

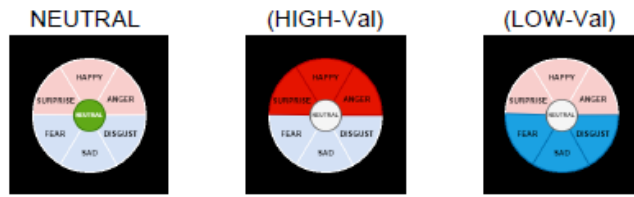
Continuous changes in vocal valence/state group over elapsed time:



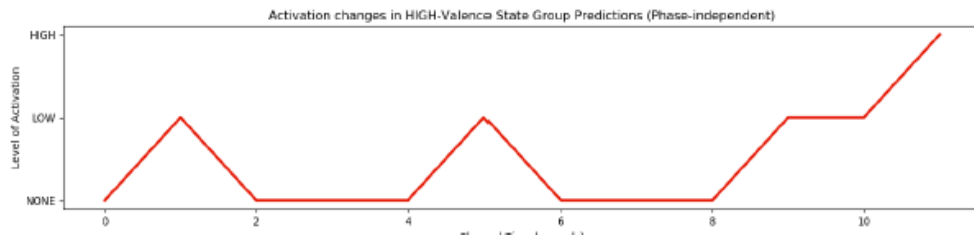
Total prediction tallies over elapsed time:



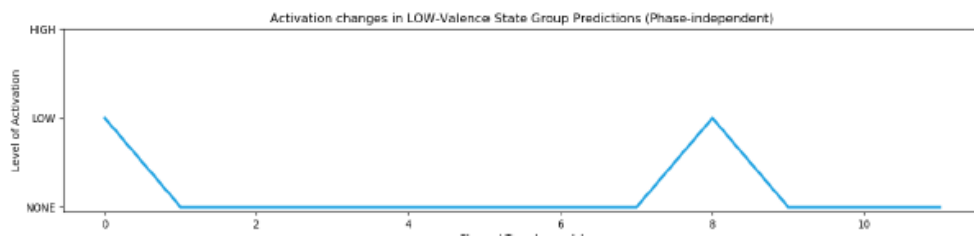
Mean Valence/State group (Prediction):	3.0833
Most common Prediction (Activation, phase and state group):	NEUTRAL
Standard Deviation:	3.3030



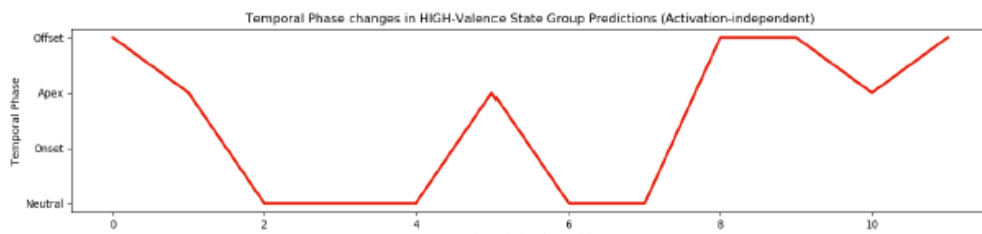
HIGH-Valence state (Happy/Surprise/Anger) activation changes (phase-independent)



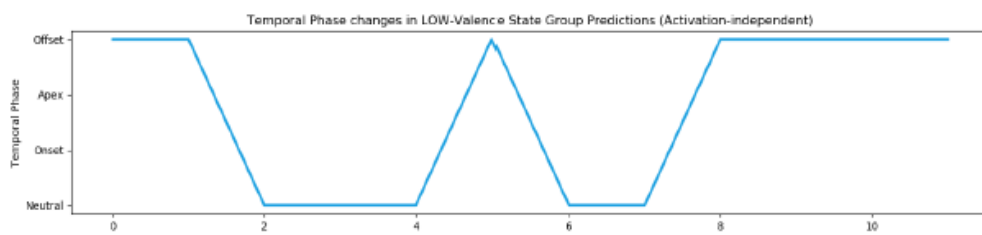
LOW-Valence state (Disgust/Sadness/Fear) activation changes (phase-independent)



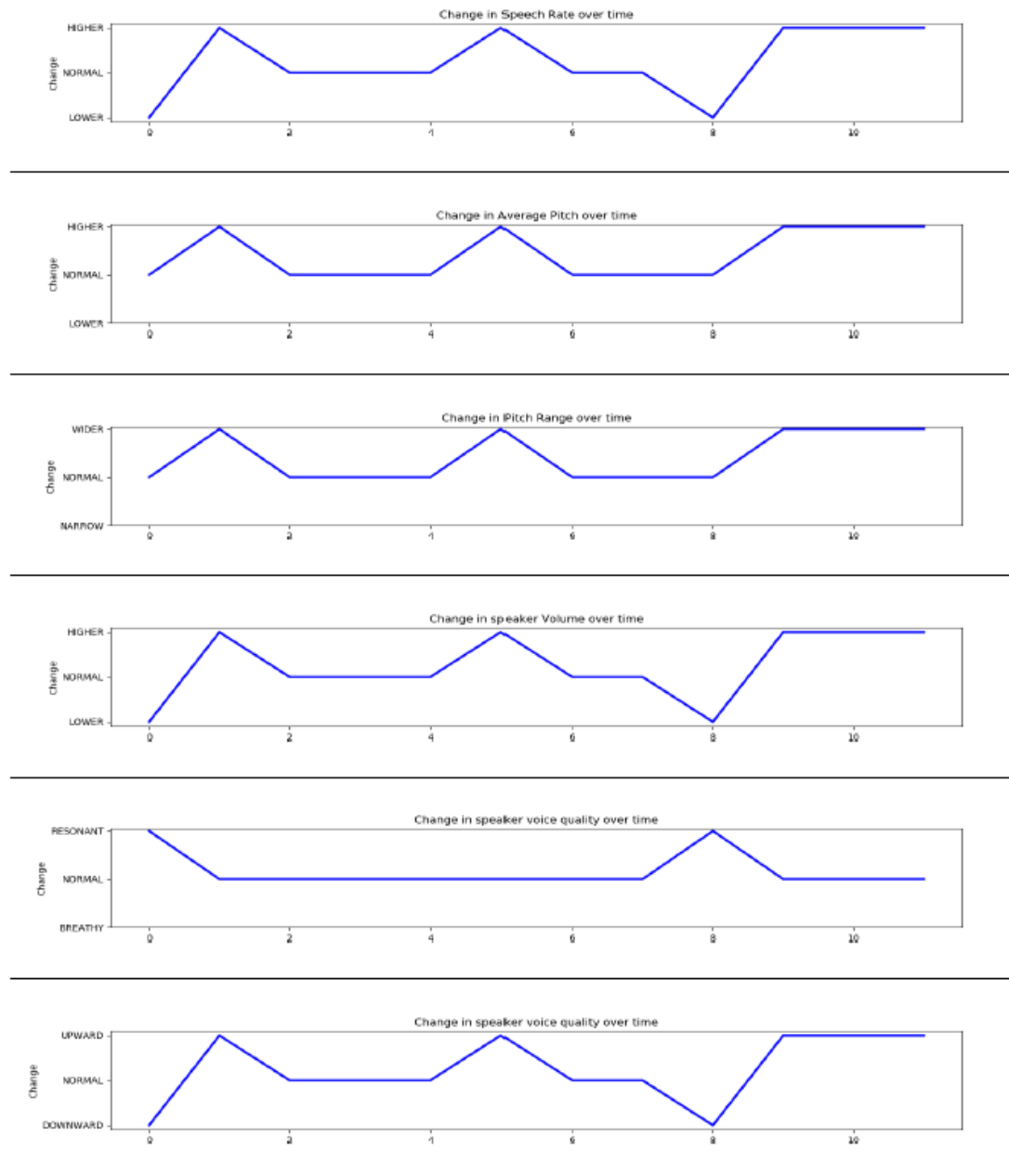
HIGH-Valence state (Happy/Surprise/Anger) Temporal Phase changes (activation-independent)



LOW-Valence state (Disgust/Sadness/Fear) Temporal changes (activation-independent)



Vocal Feature changes over time



Corresponding Log Backup: vocal_GUI_3(MANUAL)_10-02-2022_10_04_44.csv

DATE	TIME	LOG	PREDICTION INTEGER
10/02/2022	10:04:10	It sounds like this person is either exciting a mildly sad, mildly scared or mildly disgusted state	8
10/02/2022	10:04:13	It sounds like this person is either mildly happy, mildly angry or mildly surprised state	3
10/02/2022	10:04:16	This person sounds like they are in a neutral state	0
10/02/2022	10:04:18	This person sounds like they are in a neutral state	0
10/02/2022	10:04:21	This person sounds like they are in a neutral state	0
10/02/2022	10:04:24	It sounds like this person is either mildly happy, mildly angry or mildly surprised state	3
10/02/2022	10:04:26	This person sounds like they are in a neutral state	0
10/02/2022	10:04:29	This person sounds like they are in a neutral state	0
10/02/2022	10:04:32	It sounds like this person is either exciting a mildly sad, mildly scared or mildly disgusted state	8
10/02/2022	10:04:35	I sense that the speech rate, pitch and amplitude are returning to a neutral level	5
10/02/2022	10:04:39	I sense a slight increase in speech rate, pitch, and amplitude	3
10/02/2022	10:04:42	I sense the speech rate, pitch and amplitude are returning to a neutral level	11

Appendix Q: SAM Affective Speech Assessment Level 4 Generated Report

10/02/2022 10:06:15

State Assessment Module (SAM) Report

Vocal Expression Level 4 Classification - Discrete State/Emotion

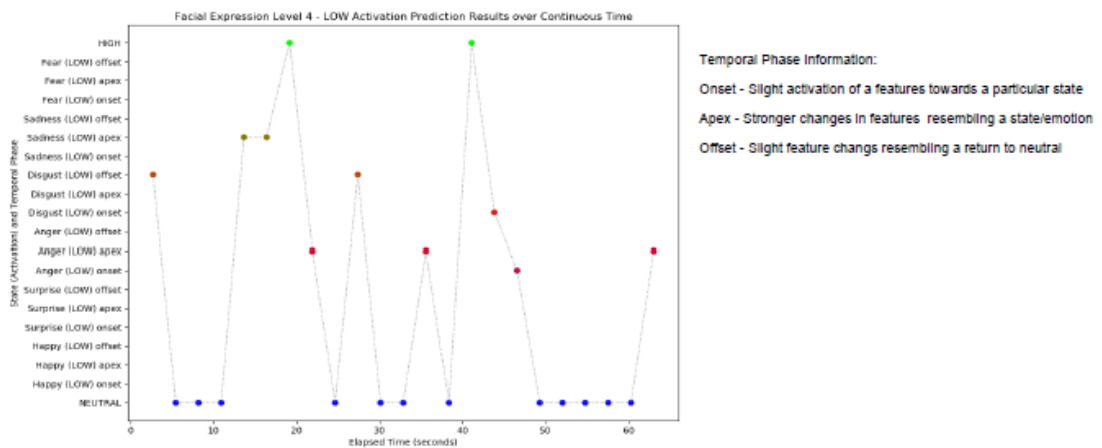
Generated By: Jordan Vice

Report Generated from: vocal_GUI_4(MANUAL)_10-02-2022_10_06_10.csv

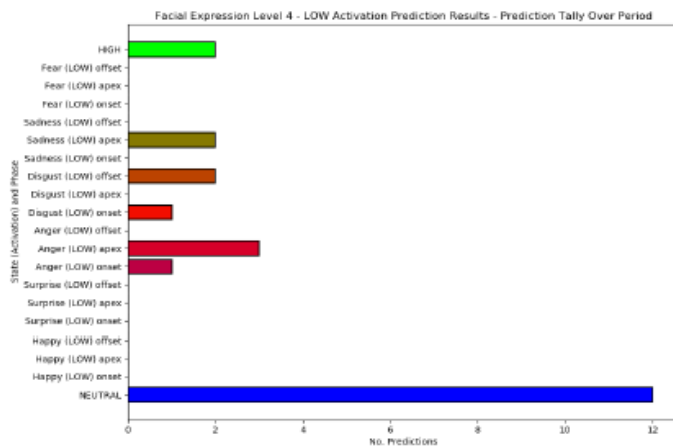
Log Time Interval: 10:05:05 --- 10:06:08

Elapsed Time (h:m:s): 0:1:3

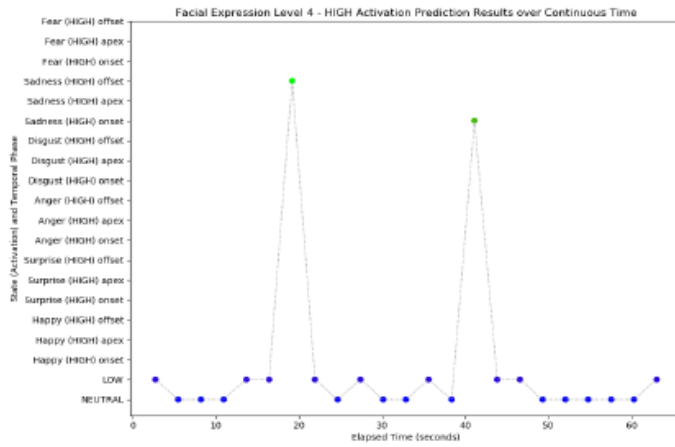
Continuous changes in LOW Activation discrete affective states and their changes in temporal phase:



Total LOW Activation prediction tallies over elapsed time:

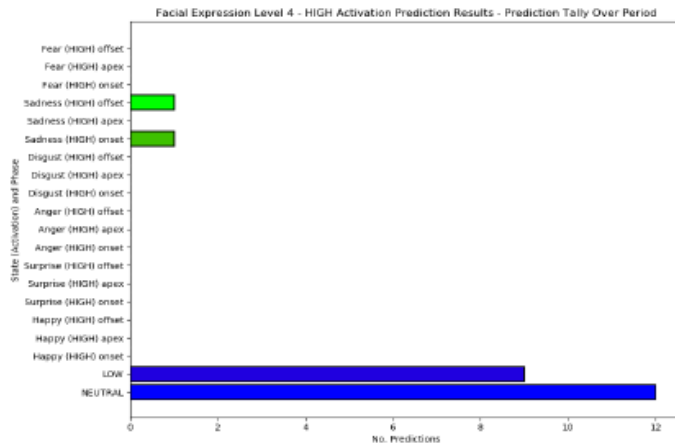


Continuous changes in HIGH Activation discrete affective states and their changes in temporal phase:

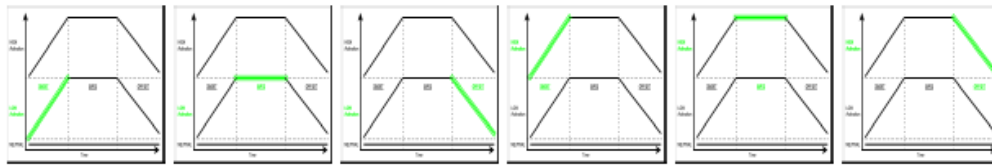


Temporal Phase Information:
 Onset - Slight activation of a features towards a particular state
 Apex - Stronger changes in features resembling a state/emotion
 Offset - Slight feature changes resembling a return to neutral

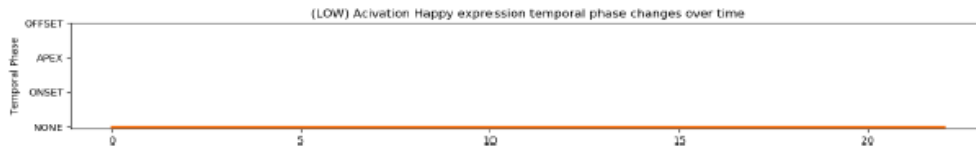
Total HIGH Activation prediction tallies over elapsed time:



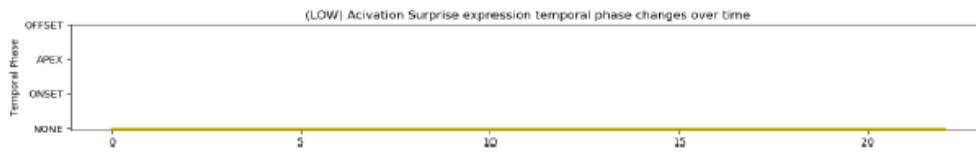
Mean Prediction Integer (State+Activation+Phase): 6.8261
 Most Common Prediction: NEUTRAL
 Standard Deviation: 9.3234



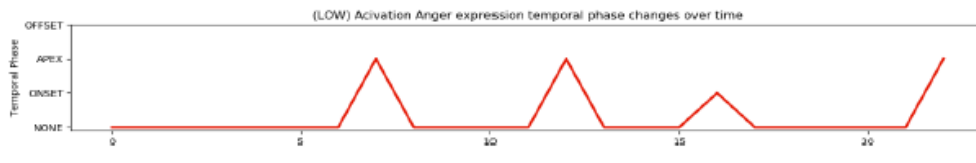
Changes in Low Activation Happiness Temporal Phase



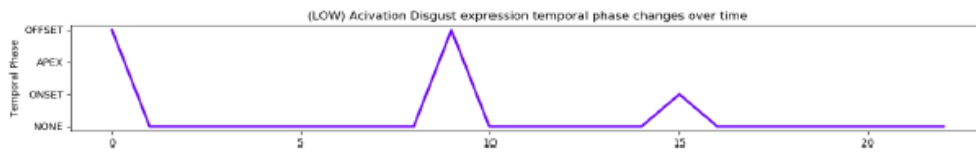
Changes in Low Activation Surprise Temporal Phase



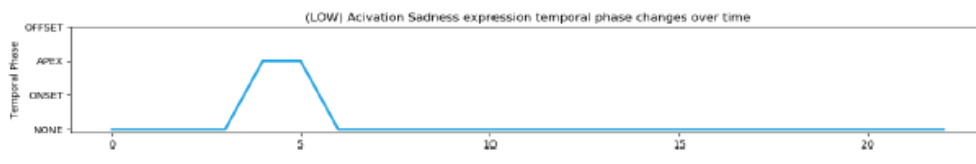
Changes in Low Activation Anger Temporal Phase



Changes in Low Activation Disgust Temporal Phase



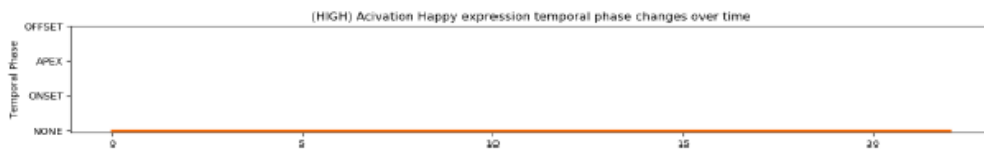
Changes in Low Activation Sadness Temporal Phase



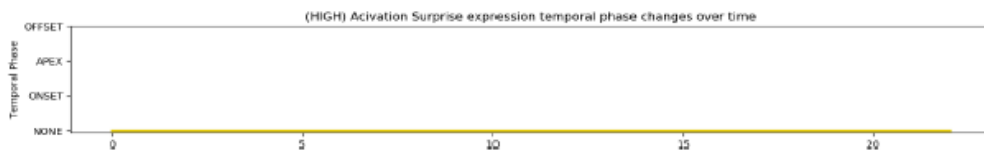
Changes in Low Activation Fear Temporal Phase



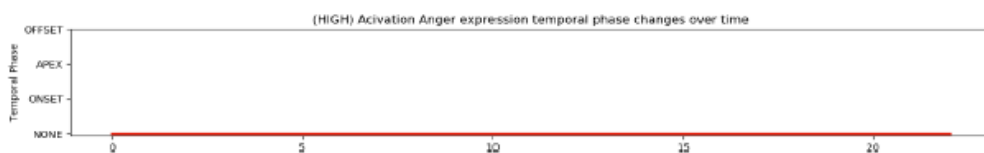
Changes in High Activation Happiness Temporal Phase



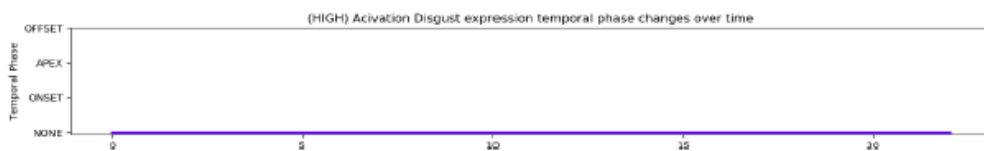
Changes in High Activation Surprise Temporal Phase



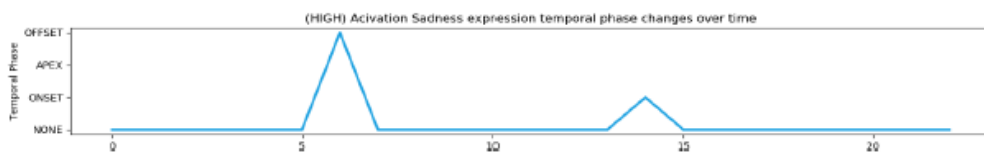
Changes in High Activation Anger Temporal Phase



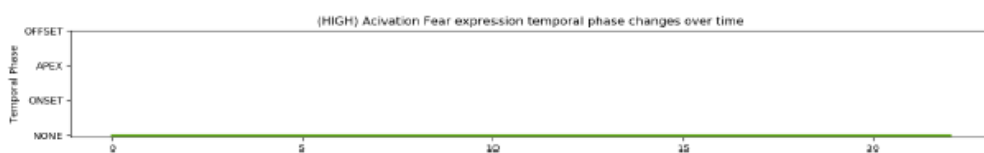
Changes in High Activation Disgust Temporal Phase



Changes in High Activation Sadness Temporal Phase

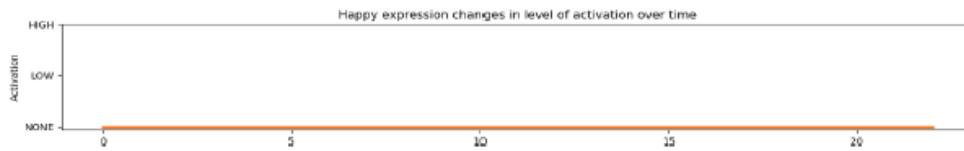


Changes in High Activation Fear Temporal Phase

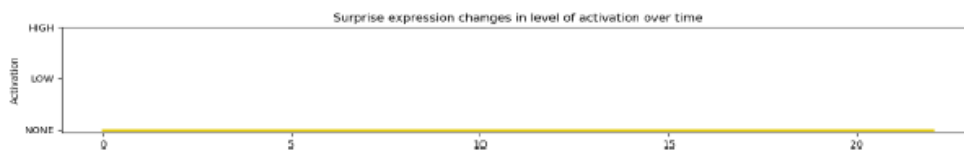




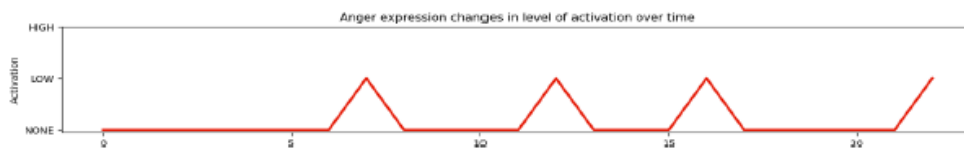
Phase-Independent Changes in Happiness Activation



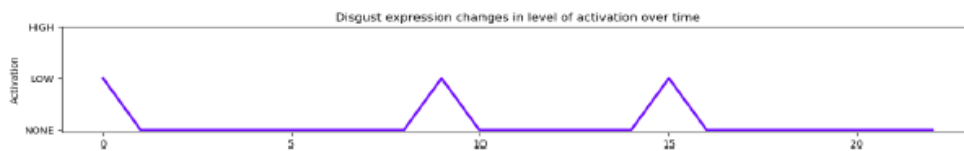
Phase-Independent Changes in Surprise Activation



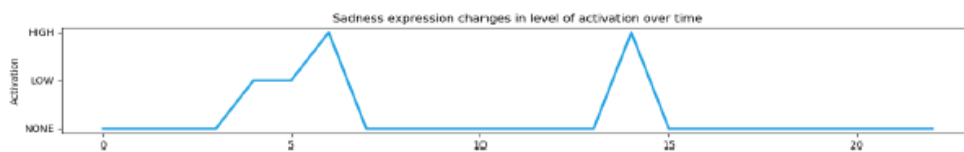
Phase-Independent Changes in Anger Activation



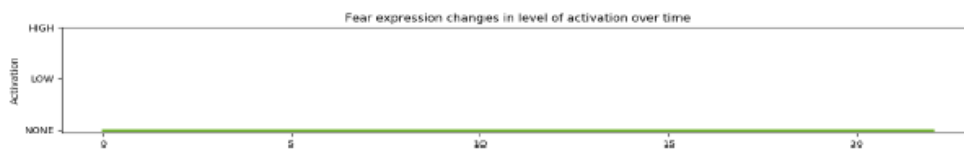
Phase-Independent Changes in Disgust Activation



Phase-Independent Changes in Sadness Activation



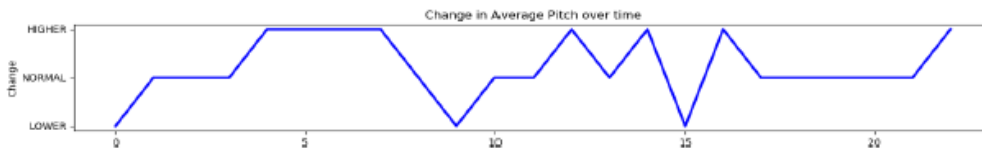
Phase-Independent Changes in Fear Activation



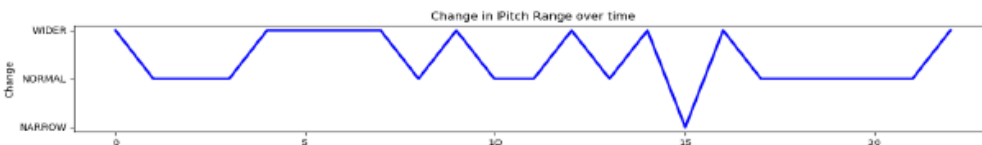
Modelling the Changes in Speech Rate over Time



Modelling the Changes in Average Pitch over Time



Modelling the Changes in Pitch Range over Time



Modelling the Changes in Speaker Volume over Time



Modelling the Changes in Speaker Voice Quality over Time



Modelling the Changes in Cadence over Time



Corresponding Log Backup: vocal_GUI_4(MANUAL)_10-02-2022_10_08_10.csv

DATE	TIME	LOG	PREDICTION INTEGER
10/02/2022	10:05:05	This person sounds like they are in a mildly disgusted state	12
10/02/2022	10:05:08	This person sounds like they are in a neutral state	0
10/02/2022	10:05:11	This person sounds like they are in a neutral state	0
10/02/2022	10:05:14	This person sounds like they are in a neutral state	0
10/02/2022	10:05:17	This person sounds like they are exiting a mild state of anger	14
10/02/2022	10:05:20	This person sounds like they are exiting a mild state of anger	14
10/02/2022	10:05:23	This person sounds like they are exiting a very surprised state	33
10/02/2022	10:05:26	This person sounds like they are mildly angry	8
10/02/2022	10:05:29	This person sounds like they are in a neutral state	0
10/02/2022	10:05:31	This person sounds like they are in a mildly disgusted state	12
10/02/2022	10:05:34	This person sounds like they are in a neutral state	0
10/02/2022	10:05:37	This person sounds like they are in a neutral state	0
10/02/2022	10:05:40	This person sounds like they are mildly angry	8
10/02/2022	10:05:42	This person sounds like they are in a neutral state	0
10/02/2022	10:05:45	This person sounds like they are exiting a very happy state	31
10/02/2022	10:05:48	This person sounds like they are mildly sad	10
10/02/2022	10:05:51	This person sounds like they are mildly happy	7
10/02/2022	10:05:54	This person sounds like they are in a neutral state	0
10/02/2022	10:05:57	This person sounds like they are in a neutral state	0
10/02/2022	10:06:00	This person sounds like they are in a neutral state	0
10/02/2022	10:06:03	This person sounds like they are in a neutral state	0
10/02/2022	10:06:06	This person sounds like they are in a neutral state	0
10/02/2022	10:06:08	This person sounds like they are mildly angry	8

Appendix R: Collated SAM Affective speech classifier dialogue responses for all windows

LEVEL 1 – Neutral + Level of Activation				
No.	HIGH/LOW	Dialogue and Reasoning		
1	NEUTRAL	This person sounds like they are in a neutral state, due to the lack of changes in speech features that I can sense.		
2	LOW	This person sounds like they are in a low activation state due to the minor, but noticeable changes in their speech.		
3	HIGH	This person sounds like they are in a high activation state due to the very strong variations in their speech.		
LEVEL 2 – Temporal Phase				
No.	HIGH/LOW	Temporal Phase	Dialogue and Reasoning	
1	LOW	Onset	This person sounds like they are approaching a mildly activated state.	
2	LOW	Apex	This person sounds like they are currently experiencing a mildly activated state.	
3	LOW	Offset	This person sounds like they are exiting a mildly activated state.	
4	HIGH	Onset	This person sounds like they are approaching a very activated state.	
5	HIGH	Apex	This person sounds like they are currently experiencing a very activated state.	
6	HIGH	Offset	This person sounds like they are exiting a very activated state.	
Level 3 – Valence and State Group: Verbose = 0				
No.	HIGH/LOW	Temporal Phase	Valence	Dialogue and Reasoning
1	LOW	Onset	HIGH Valence	It sounds like this person is either approaching a mildly happy, mildly angry, or mildly surprised state.
2	LOW	Onset	LOW Valence	It sounds like this person is either approaching a mildly sad, mildly scared, or mildly disgusted state.
3	LOW	Apex	HIGH Valence	It sounds like this person is either mildly happy, mildly angry, or mildly surprised state.
4	LOW	Apex	LOW Valence	It sounds like this person is either mildly sad, mildly scared, or mildly disgusted state.
5	LOW	Offset	HIGH Valence	It sounds like this person is either exiting a mildly happy, mildly angry, or mildly surprised state.
6	LOW	Offset	LOW Valence	It sounds like this person is either exiting a mildly sad, mildly scared, or mildly disgusted state.
7	HIGH	Onset	HIGH Valence	It sounds like this person is either approaching a very happy, very angry, or very surprised state.
8	HIGH	Onset	LOW Valence	It sounds like this person is either approaching a very sad, very scared, or very disgusted state.
9	HIGH	Apex	HIGH Valence	It sounds like this person is either very happy, very angry, or very surprised.
10	HIGH	Apex	LOW Valence	It sounds like this person is either very sad, very scared, or very disgusted.
11	HIGH	Offset	HIGH Valence	It sounds like this person is either exiting a very happy, very angry, or very surprised.
12	HIGH	Offset	LOW Valence	It sounds like this person is either exiting a very sad, very scared, or very disgusted.
Level 3 – Valence and State Group: Verbose = 1				
No.	HIGH/LOW	Temporal Phase	Valence	Dialogue and Reasoning
1	LOW	Onset	HIGH Valence	I sense speech rate and pitch are slightly increasing, along with amplitude.
2	LOW	Onset	LOW Valence	I sense slight downward inflections, irregular speech patterns, and minor decreases in amplitude.

3	LOW	Apex	HIGH Valence	I sense a slight increase in speech rate, pitch, and amplitude.
4	LOW	Apex	LOW Valence	I sense slight downward inflections in cadence, a mildly irregular voice quality, and a low amplitude.
5	LOW	Offset	HIGH Valence	I sense that the speech rate, pitch, and amplitude are returning to a neutral level.
6	LOW	Offset	LOW Valence	I sense the cadence and amplitude returning to a neutral level, and the voice quality becoming regular again.
7	HIGH	Onset	HIGH Valence	I sense speech rate and pitch are increasing, along with amplitude, at a very high rate.
8	HIGH	Onset	LOW Valence	I sense downward inflecting, irregular speech patterns, and the amplitude starting to decrease.
9	HIGH	Apex	HIGH Valence	I sense a high increase in speech rate, pitch, and amplitude.
10	HIGH	Apex	LOW Valence	I sense strong downward inflections in cadence, an irregular voice quality, and a very low amplitude.
11	HIGH	Offset	HIGH Valence	I sense the speech rate, pitch and amplitude are returning to a neutral level.
12	HIGH	Offset	LOW Valence	I sense the cadence and amplitude returning to a neutral level, and the voice quality becoming regular again.

Level 4 – Discrete State: Verbose = 0

No.	HIGH/LOW	Phase	Valence	State	Dialogue and Reasoning
1	LOW	Onset	HIGH Valence	Happy	This person sounds like they are approaching a mildly happy state.
2	LOW	Onset	HIGH Valence	Anger	This person sounds like they are approaching a mild state of anger.
3	LOW	Onset	HIGH Valence	Surprise	This person sounds like they are approaching a mildly surprised state.
4	LOW	Onset	LOW Valence	Sad	This person sounds like they are approaching a mildly sad state.
5	LOW	Onset	LOW Valence	Fear	This person sounds like they are approaching a mild state of fear.
6	LOW	Onset	LOW Valence	Disgust	This person sounds like they are approaching a mildly disgusted state.
7	LOW	Apex	HIGH Valence	Happy	This person sounds like they are mildly happy.
8	LOW	Apex	HIGH Valence	Anger	This person sounds like they are mildly angry.
9	LOW	Apex	HIGH Valence	Surprise	This person sounds like they are mildly surprised.
10	LOW	Apex	LOW Valence	Sad	This person sounds like they are mildly sad.
11	LOW	Apex	LOW Valence	Fear	This person sounds like they are in a mild state of fear.
12	LOW	Apex	LOW Valence	Disgust	This person sounds like they are in a mildly disgusted state.
13	LOW	Offset	HIGH Valence	Happy	This person sounds like they are exiting a mildly happy state.
14	LOW	Offset	HIGH Valence	Anger	This person sounds like they are exiting a mild state of anger.
15	LOW	Offset	HIGH Valence	Surprise	This person sounds like they are exiting a mildly surprised state.
16	LOW	Offset	LOW Valence	Sad	This person sounds like they are exiting a mild state of sadness.
17	LOW	Offset	LOW Valence	Fear	This person sounds like they are exiting a mild state of fear.
18	LOW	Offset	LOW Valence	Disgust	This person sounds like they are exiting a mild state of disgust.
19	HIGH	Onset	HIGH Valence	Happy	This person sounds like they are approaching a very happy state.
20	HIGH	Onset	HIGH Valence	Anger	This person sounds like they are approaching a very angry state.
21	HIGH	Onset	HIGH Valence	Surprise	This person sounds like they are approaching a very surprised state.

22	HIGH	Onset	LOW Valence	Sad	This person sounds like they are approaching a very sad state.
23	HIGH	Onset	LOW Valence	Fear	This person sounds like they are approaching a high state of fear.
24	HIGH	Onset	LOW Valence	Disgust	This person sounds like they are approaching a high state of disgust.
25	HIGH	Apex	HIGH Valence	Happy	This person sounds like they are very happy.
26	HIGH	Apex	HIGH Valence	Anger	This person sounds like they are very angry.
27	HIGH	Apex	HIGH Valence	Surprise	This person sounds like they are very surprised.
28	HIGH	Apex	LOW Valence	Sad	This person sounds like they are very sad.
29	HIGH	Apex	LOW Valence	Fear	This person sounds like they are very scared.
30	HIGH	Apex	LOW Valence	Disgust	This person sounds like they are very disgusted.
31	HIGH	Offset	HIGH Valence	Happy	This person sounds like they are exiting a very happy state.
32	HIGH	Offset	HIGH Valence	Anger	This person sounds like they are exiting a very angry state.
33	HIGH	Offset	HIGH Valence	Surprise	This person sounds like they are exiting a very surprised state.
34	HIGH	Offset	LOW Valence	Sad	This person sounds like they are exiting a very sad state.
35	HIGH	Offset	LOW Valence	Fear	This person sounds like they are exiting a high state of fear.
36	HIGH	Offset	LOW Valence	Disgust	This person sounds like they are exiting a high state of disgust.

Level 4 – Discrete State: Verbose = 1

No.	HIGH/LOW	Phase	Valence	State	Dialogue and Reasoning
1	LOW	Onset	HIGH Valence	Happy	This person sounds like they are approaching a mild happy state, as I sense some changes in their speech rate and pitch, as well as their voice getting a bit louder and breathier.
2	LOW	Onset	HIGH Valence	Anger	This person sounds like they are approaching a mild state of anger due to some changes in their speech rate and pitch, as well as their voice getting a bit louder and breathier.
3	LOW	Onset	HIGH Valence	Surprise	This person sounds like they are approaching a mildly surprised state, as I notice small changes in their speech rate, pitch, and overall voice quality.
4	LOW	Onset	LOW Valence	Sad	This person sounds like they are approaching a mildly sad state as their voice quality is getting a little quiet, and slightly irregular, while their cadence seems to be changing slightly.
5	LOW	Onset	LOW Valence	Fear	This person sounds like they are approaching a mild state of fear given it sounds like their voice is becoming slightly irregular, and I sense some variations in their speech rate and pitch.
6	LOW	Onset	LOW Valence	Disgust	This person sounds like they are approaching a mildly disgusted state as I sense their voice is becoming slightly quiet and grumbly, with some changes in their speech rate and vocal pitch.
7	LOW	Apex	HIGH Valence	Happy	This person sounds like they are quite happy, due to minor changes in speech rate and vocal pitch, showing an upward cadence, as well as their voice being louder, breathier and a bit more blaring.
8	LOW	Apex	HIGH Valence	Anger	This person sounds like they are quite angry, as I sense minor tension and stresses in their voice, which is also louder and breathier, and is emphasized by the increase in their speech rate and vocal pitch.

9	LOW	Apex	HIGH Valence	Surprise	This person sounds like they are a little surprised, due to the slight increase in speech rate and vocal pitch, which shows quite a rapid upward cadence, as well as them sounding louder than normal.
10	LOW	Apex	LOW Valence	Sad	This person sounds like they are quite sad, as I sense they are quieter than normal, with their voice sounding more slurred and resonant, along with their speech rate and pitch also being lower.
11	LOW	Apex	LOW Valence	Fear	This person sounds like they are in a mild state of fear, as their voice is quite irregular, and there is an increase in their speech rate and pitch, as they articulate themselves with this sense of precision.
12	LOW	Apex	LOW Valence	Disgust	This person sounds like they are in a mildly disgusted state, as I sense their voice is quieter and a bit grumbly, showing a slower speech rate and vocal pitch, emphasized by a slight downward cadence.
13	LOW	Offset	HIGH Valence	Happy	This person sounds like they are exiting a mildly happy state, due to their speech rate and vocal pitch returning to a normal level, as well as their voice getting quieter and less breathy.
14	LOW	Offset	HIGH Valence	Anger	This person sounds like they are exiting a mild state of anger, due to their speech rate and vocal pitch returning to a normal level, with their general voice quality also returning to normal.
15	LOW	Offset	HIGH Valence	Surprise	This person sounds like they are exiting a mildly surprised state, as I sense a quieter voice, as well as their speech rate, vocal quality and cadence returning to a more neutral level.
16	LOW	Offset	LOW Valence	Sad	This person sounds like they are exiting a mild state of sadness, given it sounds like their voice is becoming less resonant and slurred, while their speech rate and pitch returning to a neutral level.
17	LOW	Offset	LOW Valence	Fear	This person sounds like they are exiting a mild state of fear, given their voice sounds more regular, and their speech rate and pitch are returning to normal from an originally higher level.
18	LOW	Offset	LOW Valence	Disgust	This person sounds like they are exiting a mild state of disgust as their voice is less grumbly, and slightly louder, while their speech rate and vocal pitch are returning to a normal level.
19	HIGH	Onset	HIGH Valence	Happy	This person sounds like they are approaching a very happy state, due to the changes in their speech rate and pitch, as well as their voice getting louder and breathier.
20	HIGH	Onset	HIGH Valence	Anger	This person sounds like they are approaching a very angry state as I sense changes in their speech rate and pitch, as well as their voice getting louder and breathier.
21	HIGH	Onset	HIGH Valence	Surprise	This person sounds like they are approaching a very surprised state as I can hear changes in their speech rate and pitch, as well as their voice getting louder and breathier.
22	HIGH	Onset	LOW Valence	Sad	This person sounds like they are approaching a very sad state, given it sounds like their voice is getting quiet and irregular, while their cadence could be trending downward.
23	HIGH	Onset	LOW Valence	Fear	This person sounds like they are approaching a high state of fear, given it sounds like their voice is becoming irregular and I sense rapid changes in their speech rate and pitch.
24	HIGH	Onset	LOW Valence	Disgust	This person sounds like they are approaching a high state of disgust given it sounds like their voice is becoming quiet and grumbly, and their speech rate, and vocal pitch are starting to lower.

25	HIGH	Apex	HIGH Valence	Happy	This person sounds like they are very happy, due to the change in speech rate and higher vocal pitch, showing an upward cadence, as well as their voice sounding very loud, blaring, and breathy.
26	HIGH	Apex	HIGH Valence	Anger	This person sounds like they are very angry, as I sense the tension and stress in their voice, which is also very loud and breathy, not to mention the high vocal pitch and increase in their speech rate.
27	HIGH	Apex	HIGH Valence	Surprise	This person sounds like they are very surprised, due to the increase in speech rate and vocal pitch, which shows a rapid upward cadence, as well as them generally sounding louder.
28	HIGH	Apex	LOW Valence	Sad	This person sounds like they are very sad, as I sense they are quiet, with their voice sounding slurred and resonant, while also having a lower speech rate and vocal pitch, with a downward cadence.
29	HIGH	Apex	LOW Valence	Fear	This person sounds like they are very scared, as I sense the irregularity in their voice, along with their very fast speech rate and very high pitch, while still articulating themselves very precisely.
30	HIGH	Apex	LOW Valence	Disgust	This person sounds like they are very disgusted, as I sense their voice is very quiet and grumbly, showing a very slow speech rate and vocal pitch, emphasized by the downward cadence.
31	HIGH	Offset	HIGH Valence	Happy	This person sounds like they are exiting a very happy state, due to their speech rate and vocal pitch returning to a normal level, as well as their voice getting quieter and less breathy.
32	HIGH	Offset	HIGH Valence	Anger	This person sounds like they are exiting a very angry state, due to their speech rate and vocal pitch returning to a normal level, as well as their voice getting quieter and less breathy.
33	HIGH	Offset	HIGH Valence	Surprise	This person sounds like they are exiting a very surprised state due to them getting quieter, as well as their speech rate and vocal pitch lowering, with their cadence returning to a more regular pace.
34	HIGH	Offset	LOW Valence	Sad	This person sounds like they are exiting a very sad state, given it sounds like their voice is becoming less resonant and slurred, while also showing an increase in speech rate and pitch toward normal.
35	HIGH	Offset	LOW Valence	Fear	This person sounds like they are exiting a high state of fear given their voice sounds more regular and their speech rate and pitch are returning to normal from a very high level.
36	HIGH	Offset	LOW Valence	Disgust	This person sounds like they are exiting a high state of disgust as their voice is becoming less grumbly and quiet while their speech rate and vocal pitch return to a normal level from being very low.

Appendix S: Full software bill of materials produced via command: 'pip list'

```
(base) C:\Users\Jordan>pip list -v
```

Package	Version	Location	Installer
abs1-py	0.8.0	c:\users\jordan\anaconda3\lib\site-packages	
alabaster	0.7.12	c:\users\jordan\anaconda3\lib\site-packages	pip
altgraph	0.16.1	c:\users\jordan\anaconda3\lib\site-packages	pip
anaconda-client	1.7.2	c:\users\jordan\anaconda3\lib\site-packages	
anaconda-navigator	1.9.7	c:\users\jordan\anaconda3\lib\site-packages	
anaconda-project	0.8.3	c:\users\jordan\anaconda3\lib\site-packages	conda
asn1crypto	1.0.1	c:\users\jordan\anaconda3\lib\site-packages	conda
astor	0.8.0	c:\users\jordan\anaconda3\lib\site-packages	
astroid	2.3.1	c:\users\jordan\anaconda3\lib\site-packages	conda
astropy	3.2.2	c:\users\jordan\anaconda3\lib\site-packages	
astunparse	1.6.3	c:\users\jordan\anaconda3\lib\site-packages	pip
atomicwrites	1.3.0	c:\users\jordan\anaconda3\lib\site-packages	
attrs	19.1.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
audeer	1.16.0	c:\users\jordan\anaconda3\lib\site-packages	pip
audformat	0.11.6	c:\users\jordan\anaconda3\lib\site-packages	pip
audinterfaced	0.7.0	c:\users\jordan\anaconda3\lib\site-packages	pip
audiofile	1.0.0	c:\users\jordan\anaconda3\lib\site-packages	pip
audioread	2.1.8	c:\users\jordan\anaconda3\lib\site-packages	pip
audobject	0.4.14	c:\users\jordan\anaconda3\lib\site-packages	pip
audresample	0.1.6	c:\users\jordan\anaconda3\lib\site-packages	pip
Babel	2.7.0	c:\users\jordan\anaconda3\lib\site-packages	pip
backcall	0.1.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
backports.functools-lru-cache	1.5	c:\users\jordan\anaconda3\lib\site-packages	
backports.os	0.1.1	c:\users\jordan\anaconda3\lib\site-packages	
backports.shutil-get-terminal-size	1.0.0	c:\users\jordan\anaconda3\lib\site-packages	pip
backports.tempfile	1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
backports.weakref	1.0.post1	c:\users\jordan\anaconda3\lib\site-packages	
beautifulsoup4	4.8.0	c:\users\jordan\anaconda3\lib\site-packages	pip
bitarray	1.0.1	c:\users\jordan\anaconda3\lib\site-packages	pip
bkcharts	0.2	c:\users\jordan\anaconda3\lib\site-packages	
bleach	3.1.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
bokeh	1.3.4	c:\users\jordan\anaconda3\lib\site-packages	pip
boto	2.49.0	c:\users\jordan\anaconda3\lib\site-packages	
Bottleneck	1.2.1	c:\users\jordan\anaconda3\lib\site-packages	
cached-property	1.5.2	c:\users\jordan\anaconda3\lib\site-packages	pip
cachetools	4.1.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
certifi	2020.6.20	c:\users\jordan\anaconda3\lib\site-packages	
cffi	1.12.3	c:\users\jordan\anaconda3\lib\site-packages	pip
chardet	3.0.4	c:\users\jordan\anaconda3\lib\site-packages	conda
Click	7.0	c:\users\jordan\anaconda3\lib\site-packages	
cloudpickle	1.2.2	c:\users\jordan\anaconda3\lib\site-packages	conda
clyent	1.2.2	c:\users\jordan\anaconda3\lib\site-packages	
colorama	0.4.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
comtypes	1.1.7	c:\users\jordan\anaconda3\lib\site-packages	
conda	4.9.0	c:\users\jordan\anaconda3\lib\site-packages	
conda-build	3.18.9	c:\users\jordan\anaconda3\lib\site-packages	
conda-package-handling	1.6.0	c:\users\jordan\anaconda3\lib\site-packages	conda
conda-verify	3.4.2	c:\users\jordan\anaconda3\lib\site-packages	conda
contextlib2	0.6.0	c:\users\jordan\anaconda3\lib\site-packages	conda
cryptography	2.7	c:\users\jordan\anaconda3\lib\site-packages	pip
ctypes-callable	1.0.0	c:\users\jordan\anaconda3\lib\site-packages	pip
cycler	0.10.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
Cython	0.29.13	c:\users\jordan\anaconda3\lib\site-packages	pip
cytoolz	0.10.0	c:\users\jordan\anaconda3\lib\site-packages	
dask	2.5.2	c:\users\jordan\anaconda3\lib\site-packages	conda
decorator	4.4.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
defusedxml	0.6.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
dill	0.3.4	c:\users\jordan\anaconda3\lib\site-packages	pip
distributed	2.5.2	c:\users\jordan\anaconda3\lib\site-packages	conda
docutils	0.15.2	c:\users\jordan\anaconda3\lib\site-packages	pip
dtcwt	0.12.0	c:\users\jordan\anaconda3\lib\site-packages	pip
einops	0.3.2	c:\users\jordan\anaconda3\lib\site-packages	pip
entrypoints	0.3	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
et-xmlfile	1.0.1	c:\users\jordan\anaconda3\lib\site-packages	
fast-bss-eval	0.1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
fastcache	1.1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
ffpyplayer	4.3.1	c:\users\jordan\anaconda3\lib\site-packages	pip
filelock	3.0.12	c:\users\jordan\anaconda3\lib\site-packages	pip
Flask	1.1.1	c:\users\jordan\anaconda3\lib\site-packages	conda
fsspec	0.5.2	c:\users\jordan\anaconda3\lib\site-packages	conda

future	0.17.1	c:\users\jordan\anaconda3\lib\site-packages	
gast	0.3.3	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
gevent	1.4.0	c:\users\jordan\anaconda3\lib\site-packages	
glob2	0.7	c:\users\jordan\anaconda3\lib\site-packages	conda
google-auth	1.19.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
google-auth-oauthlib	0.4.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
google-pasta	0.2.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
greenlet	0.4.15	c:\users\jordan\anaconda3\lib\site-packages	pip
grpcio	1.24.3	c:\users\jordan\anaconda3\lib\site-packages	pip
h5py	2.10.0	c:\users\jordan\anaconda3\lib\site-packages	pip
HeapDict	1.0.1	c:\users\jordan\anaconda3\lib\site-packages	conda
hmmlearn	0.2.3	c:\users\jordan\anaconda3\lib\site-packages	pip
html5lib	1.0.1	c:\users\jordan\anaconda3\lib\site-packages	pip
idna	2.8	c:\users\jordan\anaconda3\lib\site-packages	
imageio	2.6.0	c:\users\jordan\anaconda3\lib\site-packages	conda
imagesize	1.1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
importlib-metadata	0.23	c:\users\jordan\anaconda3\lib\site-packages	conda
imutils	0.5.2	c:\users\jordan\anaconda3\lib\site-packages	pip
ipykernel	5.1.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
ipython	7.5.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
ipython-genutils	0.2.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
ipython-sql	0.4.0	c:\users\jordan\anaconda3\lib\site-packages	pip
ipywidgets	7.4.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
iso-639	0.4.5	c:\users\jordan\anaconda3\lib\site-packages	pip
isort	4.3.21	c:\users\jordan\anaconda3\lib\site-packages	pip
itsdangerous	1.1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
jdcal	1.4.1	c:\users\jordan\anaconda3\lib\site-packages	pip
jedi	0.13.3	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
Jinja2	2.10.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
joblib	0.13.2	c:\users\jordan\anaconda3\lib\site-packages	
json5	0.8.5	c:\users\jordan\anaconda3\lib\site-packages	conda
jsonschema	3.0.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
jupyter	1.0.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
jupyter-client	5.2.4	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
jupyter-console	6.0.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
jupyter-core	4.4.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
jupyterlab	1.1.4	c:\users\jordan\anaconda3\lib\site-packages	
jupyterlab-server	1.0.6	c:\users\jordan\anaconda3\lib\site-packages	conda
Keras	2.3.0	c:\users\jordan\anaconda3\lib\site-packages	pip
Keras-Applications	1.0.8	c:\users\jordan\anaconda3\lib\site-packages	pip
Keras-Preprocessing	1.1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
keyring	18.0.0	c:\users\jordan\anaconda3\lib\site-packages	pip
Kivy	1.11.1	c:\users\jordan\anaconda3\lib\site-packages	conda
kiwisolver	1.1.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
klepto	0.2.1	c:\users\jordan\anaconda3\lib\site-packages	pip
lazy-object-proxy	1.4.2	c:\users\jordan\anaconda3\lib\site-packages	pip
libarchive-c	2.8	c:\users\jordan\anaconda3\lib\site-packages	
librosa	0.7.0	c:\users\jordan\anaconda3\lib\site-packages	pip
llvmlite	0.29.0	c:\users\jordan\anaconda3\lib\site-packages	
locket	0.2.0	c:\users\jordan\anaconda3\lib\site-packages	
Lxml	4.4.1	c:\users\jordan\anaconda3\lib\site-packages	pip
Markdown	3.1.1	c:\users\jordan\anaconda3\lib\site-packages	pip
MarkupSafe	1.1.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
matplotlib	3.1.1	c:\users\jordan\anaconda3\lib\site-packages	
mccabe	0.6.1	c:\users\jordan\anaconda3\lib\site-packages	
menuinst	1.4.16	c:\users\jordan\anaconda3\lib\site-packages	
MiniSom	2.2.9	c:\users\jordan\anaconda3\lib\site-packages	pip
mir-eval	0.6	c:\users\jordan\anaconda3\lib\site-packages	pip
mistune	0.8.4	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
mkl-fft	1.0.14	c:\users\jordan\anaconda3\lib\site-packages	
mkl-random	1.1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
mkl-service	2.3.0	c:\users\jordan\anaconda3\lib\site-packages	
mock	3.0.5	c:\users\jordan\anaconda3\lib\site-packages	pip
module-graph	0.0.1	c:\users\jordan\anaconda3\lib\site-packages	pip
more-itertools	7.2.0	c:\users\jordan\anaconda3\lib\site-packages	conda
mpmath	1.1.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
msgpack	0.6.1	c:\users\jordan\anaconda3\lib\site-packages	pip
multipledispatch	0.6.0	c:\users\jordan\anaconda3\lib\site-packages	
mysql-connector-python	8.0.21	c:\users\jordan\anaconda3\lib\site-packages	
navigator-updater	0.2.1	c:\users\jordan\anaconda3\lib\site-packages	
nbconvert	5.5.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
nbformat	4.4.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
networkx	2.3	c:\users\jordan\anaconda3\lib\site-packages	pip
nltk	3.4.5	c:\users\jordan\anaconda3\lib\site-packages	pip

nose	1.3.7	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
notebook	5.7.8	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
numba	0.45.1	c:\users\jordan\anaconda3\lib\site-packages	
numexpr	2.7.0	c:\users\jordan\anaconda3\lib\site-packages	
numpy	1.16.3	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
numpydoc	0.9.1	c:\users\jordan\anaconda3\lib\site-packages	pip
oauthlib	3.1.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
olefile	0.46	c:\users\jordan\anaconda3\lib\site-packages	
opencv-python	4.5.1.48	c:\users\jordan\anaconda3\lib\site-packages	pip
openpyxl	3.0.0	c:\users\jordan\anaconda3\lib\site-packages	conda
opensmile	2.2.0	c:\users\jordan\anaconda3\lib\site-packages	pip
opt-einsum	3.3.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
oyaml	1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
packaging	19.2	c:\users\jordan\anaconda3\lib\site-packages	conda
pandas	1.1.5	c:\users\jordan\anaconda3\lib\site-packages	pip
pandocfilters	1.4.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
parso	0.4.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
partd	1.0.0	c:\users\jordan\anaconda3\lib\site-packages	conda
path.py	12.0.1	c:\users\jordan\anaconda3\lib\site-packages	pip
pathlib2	2.3.5	c:\users\jordan\anaconda3\lib\site-packages	conda
patsy	0.5.1	c:\users\jordan\anaconda3\lib\site-packages	
pb-bss-eval	0.0.2	c:\users\jordan\anaconda3\lib\site-packages	pip
pefile	2019.4.18	c:\users\jordan\anaconda3\lib\site-packages	pip
pep8	1.7.1	c:\users\jordan\anaconda3\lib\site-packages	
pesq	0.0.3	c:\users\jordan\anaconda3\lib\site-packages	pip
pickleshare	0.7.5	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
Pillow	6.2.0	c:\users\jordan\anaconda3\lib\site-packages	conda
pip	21.1	c:\users\jordan\anaconda3\lib\site-packages	pip
pkginfo	1.5.0.1	c:\users\jordan\anaconda3\lib\site-packages	
pluggy	0.13.0	c:\users\jordan\anaconda3\lib\site-packages	conda
ply	3.11	c:\users\jordan\anaconda3\lib\site-packages	
pox	0.3.0	c:\users\jordan\anaconda3\lib\site-packages	pip
praat-parselmouth	0.3.3	c:\users\jordan\anaconda3\lib\site-packages	pip
prettytable	0.7.2	c:\users\jordan\anaconda3\lib\site-packages	pip
prometheus-client	0.6.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
prompt-toolkit	2.0.9	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
protobuf	3.9.2	c:\users\jordan\anaconda3\lib\site-packages	
psutil	5.6.3	c:\users\jordan\anaconda3\lib\site-packages	pip
py	1.8.0	c:\users\jordan\anaconda3\lib\site-packages	pip
py2exe	0.10.4.0	c:\users\jordan\anaconda3\lib\site-packages	pip
pyasn1	0.4.8	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
pyasn1-modules	0.2.8	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
PyAudio	0.2.11	c:\users\jordan\anaconda3\lib\site-packages	pip
pycodestyle	2.5.0	c:\users\jordan\anaconda3\lib\site-packages	
pycosat	0.6.3	c:\users\jordan\anaconda3\lib\site-packages	
pycparser	2.19	c:\users\jordan\anaconda3\lib\site-packages	
pycrypto	2.6.1	c:\users\jordan\anaconda3\lib\site-packages	
pycurl	7.43.0.3	c:\users\jordan\anaconda3\lib\site-packages	
pydub	0.23.1	c:\users\jordan\anaconda3\lib\site-packages	pip
pyflakes	2.1.1	c:\users\jordan\anaconda3\lib\site-packages	pip
pygame	1.9.6	c:\users\jordan\anaconda3\lib\site-packages	pip
Pygments	2.3.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
pyinstaller	4.4	c:\users\jordan\anaconda3\lib\site-packages	pip
pyinstaller-hooks-contrib	2021.2	c:\users\jordan\anaconda3\lib\site-packages	pip
pylint	2.4.2	c:\users\jordan\anaconda3\lib\site-packages	conda

pynput	1.6.8	c:\users\jordan\anaconda3\lib\site-packages	pip
pyodbc	4.0.27	c:\users\jordan\anaconda3\lib\site-packages	pip
pyOpenSSL	19.0.0	c:\users\jordan\anaconda3\lib\site-packages	
pyparsing	2.4.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
pypiwin32	223	c:\users\jordan\anaconda3\lib\site-packages	pip
PyQt5	5.14.1	c:\users\jordan\anaconda3\lib\site-packages	pip
PyQt5-sip	12.7.0	c:\users\jordan\anaconda3\lib\site-packages	pip
pyreadline	2.1	c:\users\jordan\anaconda3\lib\site-packages	
pyrsistent	0.15.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
PySimpleGUI	4.56.0	c:\users\jordan\anaconda3\lib\site-packages	pip
PySocks	1.7.1	c:\users\jordan\anaconda3\lib\site-packages	conda
pystoi	0.3.3	c:\users\jordan\anaconda3\lib\site-packages	pip
pytest	5.2.1	c:\users\jordan\anaconda3\lib\site-packages	
pytest-arraydiff	0.3	c:\users\jordan\anaconda3\lib\site-packages	pip
pytest-astropy	0.5.0	c:\users\jordan\anaconda3\lib\site-packages	pip
pytest-doctestplus	0.4.0	c:\users\jordan\anaconda3\lib\site-packages	conda
pytest-openfiles	0.4.0	c:\users\jordan\anaconda3\lib\site-packages	conda
pytest-remotedata	0.3.2	c:\users\jordan\anaconda3\lib\site-packages	conda
python-dateutil	2.8.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
python-speech-features	0.6	c:\users\jordan\anaconda3\lib\site-packages	pip
pyttsx3	2.71	c:\users\jordan\anaconda3\lib\site-packages	pip

pytz	2019.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
PyWavelets	1.0.3	c:\users\jordan\anaconda3\lib\site-packages	pip
pywin32	223	c:\users\jordan\anaconda3\lib\site-packages	
pywin32-ctypes	0.2.0	c:\users\jordan\anaconda3\lib\site-packages	pip
pywintpy	0.5.5	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
PyYAML	5.1.2	c:\users\jordan\anaconda3\lib\site-packages	
pymzq	18.0.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
QtAwesome	0.6.0	c:\users\jordan\anaconda3\lib\site-packages	conda
qtconsole	4.4.4	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
QtPy	1.9.0	c:\users\jordan\anaconda3\lib\site-packages	conda
reportlab	3.6.6	c:\users\jordan\anaconda3\lib\site-packages	pip
requests	2.22.0	c:\users\jordan\anaconda3\lib\site-packages	pip
requests-oauthlib	1.3.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
resampy	0.2.2	c:\users\jordan\anaconda3\lib\site-packages	pip
rope	0.14.0	c:\users\jordan\anaconda3\lib\site-packages	
rsa	4.6	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
ruamel-yaml	0.15.46	c:\users\jordan\anaconda3\lib\site-packages	
scikit-image	0.15.0	c:\users\jordan\anaconda3\lib\site-packages	pip
scikit-learn	0.24.2	c:\users\jordan\anaconda3\lib\site-packages	pip
scipy	1.4.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
seaborn	0.9.0	c:\users\jordan\anaconda3\lib\site-packages	pip
Send2Trash	1.5.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
setuptools	41.4.0	c:\users\jordan\anaconda3\lib\site-packages	
simplegeneric	0.8.1	c:\users\jordan\anaconda3\lib\site-packages	
singledispatch	3.4.0.3	c:\users\jordan\anaconda3\lib\site-packages	
six	1.12.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
snowballstemmer	2.0.0	c:\users\jordan\anaconda3\lib\site-packages	conda
sortedcollections	1.1.2	c:\users\jordan\anaconda3\lib\site-packages	pip
sortedcontainers	2.1.0	c:\users\jordan\anaconda3\lib\site-packages	
sounddevice	0.3.13	c:\users\jordan\anaconda3\lib\site-packages	pip
SoundFile	0.10.2	c:\users\jordan\anaconda3\lib\site-packages	pip
soupsieve	1.9.3	c:\users\jordan\anaconda3\lib\site-packages	conda
sox	1.4.1	c:\users\jordan\anaconda3\lib\site-packages	pip
SpeechRecognition	3.8.1	c:\users\jordan\anaconda3\lib\site-packages	pip
Sphinx	2.2.0	c:\users\jordan\anaconda3\lib\site-packages	conda
sphinxcontrib-applehelp	1.0.1	c:\users\jordan\anaconda3\lib\site-packages	pip
sphinxcontrib-devhelp	1.0.1	c:\users\jordan\anaconda3\lib\site-packages	pip
sphinxcontrib-htmlhelp	1.0.2	c:\users\jordan\anaconda3\lib\site-packages	pip
sphinxcontrib-jsmath	1.0.1	c:\users\jordan\anaconda3\lib\site-packages	pip
sphinxcontrib-qthelp	1.0.2	c:\users\jordan\anaconda3\lib\site-packages	pip
sphinxcontrib-serializinghtml	1.1.3	c:\users\jordan\anaconda3\lib\site-packages	pip
sphinxcontrib-websupport	1.1.2	c:\users\jordan\anaconda3\lib\site-packages	pip
spyder	3.3.6	c:\users\jordan\anaconda3\lib\site-packages	pip
spyder-kernels	0.5.2	c:\users\jordan\anaconda3\lib\site-packages	conda
SQLAlchemy	1.3.9	c:\users\jordan\anaconda3\lib\site-packages	conda
sqlparse	0.3.1	c:\users\jordan\anaconda3\lib\site-packages	pip
statsmodels	0.10.1	c:\users\jordan\anaconda3\lib\site-packages	
sympy	1.4	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
tables	3.5.2	c:\users\jordan\anaconda3\lib\site-packages	
tabulate	0.8.7	c:\users\jordan\anaconda3\lib\site-packages	pip
tblib	1.4.0	c:\users\jordan\anaconda3\lib\site-packages	pip
tensorboard	2.2.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
tensorboard-plugin-wit	1.7.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
tensorflow	2.2.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
tensorflow-estimator	2.2.0	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip

termcolor	1.1.0	c:\users\jordan\anaconda3\lib\site-packages	
terminado	0.8.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
testpath	0.4.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
threadpoolctl	2.1.0	c:\users\jordan\anaconda3\lib\site-packages	pip
toolz	0.10.0	c:\users\jordan\anaconda3\lib\site-packages	conda
tornado	6.0.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
tornado-utils	1.6	c:\users\jordan\anaconda3\lib\site-packages	pip
tqdm	4.36.1	c:\users\jordan\anaconda3\lib\site-packages	conda
traitlets	4.3.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
typed-ast	1.4.0	c:\users\jordan\anaconda3\lib\site-packages	conda
unicodectv	0.14.1	c:\users\jordan\anaconda3\lib\site-packages	
UNKNOWN	0.0.0	c:\users\jordan\anaconda3\lib\site-packages	
urllib3	1.24.2	c:\users\jordan\anaconda3\lib\site-packages	
Wave	0.0.2	c:\users\jordan\anaconda3\lib\site-packages	pip
wcwidth	0.1.7	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
webencodings	0.5.1	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
Werkzeug	0.16.0	c:\users\jordan\anaconda3\lib\site-packages	conda
wheel	0.33.6	c:\users\jordan\anaconda3\lib\site-packages	
widgetsnbextension	3.4.2	c:\users\jordan\appdata\roaming\python\python36\site-packages	pip
win-inet-pton	1.1.0	c:\users\jordan\anaconda3\lib\site-packages	
win-unicode-console	0.5	c:\users\jordan\anaconda3\lib\site-packages	
wincertstore	0.2	c:\users\jordan\anaconda3\lib\site-packages	
wrapt	1.11.2	c:\users\jordan\anaconda3\lib\site-packages	pip
xlrd	1.2.0	c:\users\jordan\anaconda3\lib\site-packages	pip
XlsxWriter	1.2.1	c:\users\jordan\anaconda3\lib\site-packages	conda
xlwings	0.15.10	c:\users\jordan\anaconda3\lib\site-packages	
xlwt	1.3.0	c:\users\jordan\anaconda3\lib\site-packages	
zict	1.0.0	c:\users\jordan\anaconda3\lib\site-packages	
zipf	0.6.0	c:\users\jordan\anaconda3\lib\site-packages	conda

Corresponding Publications

Title: Multimodal Models for Contextual Affect Assessment in Real-Time

Authors: Jordan Vice, Masood Khan, Svetlana Yanushkevich

Abstract:

Most affect classification schemes rely on near accurate single-cue models resulting in less than required accuracy under certain peculiar conditions. We investigate how the holism of a multimodal solution could be exploited for affect classification. This paper presents the design and implementation of a prototype, stand-alone, real-time multimodal affective state classification system. The presented system utilizes speech and facial muscle movements to create a holistic classifier. The system combines a facial expression classifier and a speech classifier that analyses speech through paralinguistic and propositional content. The proposed classification scheme includes a Support Vector Machine (SVM) - paralinguistic; a K-Nearest Neighbor (KNN) - propositional content and an InceptionV3 neural network - facial expressions of affective states. The SVM and Inception models boasted respective validation accuracies of 99.2% and 92.78%.

Published in: 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)

DOI: 10.1109/CogMI48466.2019.00020

Title: Adaptive Classification of Occluded Facial Expressions of Affective States

Authors: Jordan Vice, Masood Khan, Iain Murray, Svetlana Yanushkevich

Abstract:

Internationally, the recent pandemic caused severe social changes forcing people to adopt new practices in their daily lives. One of these changes requires people to wear masks in public spaces to mitigate the spread of viral diseases. Affective state assessment (ASA) systems that rely on facial expression analysis become impaired and less effective due to the presence of visual occlusions caused by wearing masks. Therefore, ASA systems need to be future-proofed and equipped with adaptive technologies to be able to analyze and assess occluded facial expressions, particularly in the presence of masks. This paper presents an adaptive approach for classifying occluded facial expressions when human faces are partially covered with masks. We deployed an unsupervised, cosine similarity-based clustering approach exploiting the continuous nature of the extended Cohn-Kanade (CK+) dataset. The cosine similarity-based clustering resulted in twenty-one micro-expression clusters that describe minor variations of human facial expressions. Linear discriminant analysis was used to project all clusters onto lower-dimensional discriminant feature spaces, allowing for binary occlusion classification and the dynamic assessment of affective states. During the validation stage, we observed 100% accuracy when classifying faces with features extracted from the lower part of the occluded faces (occlusion detection). We observed 76.11% facial expression classification accuracy when features were gathered from the uncovered full-faces and 73.63% classification accuracy when classifying upper-facial expressions - applied when the lower part of the face is occluded. The presented system promises an improvement to visual inspection systems through an adaptive occlusion detection and facial expression classification framework.

Published in: 2022 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)

DOI: 10.1109/EAIS51927.2022.9787693

Title: Dynamic Hybrid Learning for Improving Facial Expression Classifier Reliability

Authors: Jordan Vice, Masood Khan, Tele Tan, Svetlana Yanushkevich

Abstract:

Independent, discrete models like Paul Ekman's six basic emotions model are widely used in affective state assessment (ASA) and facial expression classification. However, the continuous and dynamic nature of human expressions often needs to be considered for accurately assessing facial expressions of affective states. This paper investigates how mutual information-carrying continuous models can be extracted and used in continuous and dynamic facial expression classification systems for improving the efficacy and reliability of ASA systems. A novel, hybrid learning model that projects continuous data onto a multidimensional hyperplane is proposed. Through cosine similarity-based clustering (unsupervised) and classification (supervised) processes, our hybrid approach allows us to transform seven, discrete facial expression models into twenty-one facial expression models that include micro-expressions. The proposed continuous, dynamic classifier was able to achieve greater than 73% accuracy when experimented with Random Forest, Support Vector Machine (SVM) and Neural Network classification architectures. The presented system was validated using the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) and the extended Cohn-Kanade (CK+) dataset.

Published in: 2022 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS)

DOI: 10.1109/EAIS51927.2022.9787730

Title: Toward Accountable and Explainable Artificial Intelligence Part One: Theory and Examples

Authors: Masood Khan, Jordan Vice

Abstract:

Like other Artificial Intelligence (AI) systems, Machine Learning (ML) applications cannot explain decisions, are marred with training-caused biases, and suffer from algorithmic limitations. Their eXplainable Artificial Intelligence (XAI) capabilities are typically measured in a two-dimensional space of explainability and accuracy ignoring the accountability aspects. During system evaluations, measures of comprehensibility, predictive accuracy and accountability remain inseparable. We propose an Accountable eXplainable Artificial Intelligence (AXAI) capability framework for facilitating separation and measurement of predictive accuracy, comprehensibility and accountability. The proposed framework, in its current form, allows assessing embedded levels of AXAI for delineating ML systems in a three-dimensional space. The AXAI framework quantifies comprehensibility in terms of the readiness of users to apply the acquired knowledge and assesses predictive accuracy in terms of the ratio of test and training data, training data size and the number of false-positive inferences. For establishing a chain of responsibility, accountability is measured in terms of the inspectability of input cues, data being processed and the output information. We demonstrate applying the framework for assessing the AXAI capabilities of three ML systems. The reported work provides bases for building AXAI capability frameworks for other genres of AI systems.

Published in: IEEE Access (Volume 10)

DOI: 10.1109/ACCESS.2022.3207812

Title: Toward Accountable and Explainable Artificial Intelligence Part Two: The Framework Implementation

Authors: Jordan Vice, Masood Khan

Abstract:

This paper builds upon the theoretical foundations of the Accountable eXplainable Artificial Intelligence (AXAI) capability framework presented in part one of this paper. We demonstrate incorporation of the AXAI capability in the real time Affective State Assessment Module (ASAM) of a robotic system. We show that adhering to the eXtreme Programming (XP) practices would help in understanding user behavior and systematic incorporation of the AXAI capability in Machine Learning (ML) systems. We further show that a collaborative software design and development process (SDDP) would facilitate identification of ethical, technical, functional, and domain-specific system requirements. Meeting these requirements would increase user confidence in ML and AI systems. Our results show that the ASAM can synthesize discrete and continuous models of affective state expressions for classifying them in real-time. The ASAM continuously shares important inputs, processed data and the output information with users via a graphical user interface (GUI). Thus, the GUI presents reasons behind system decisions and disseminates information about local reasoning, data handling and decision-making. Through this demonstrated work, we expect to move toward enhancing AI systems' acceptability, utility and establishing a chain of responsibility if a system fails. We hope this work will initiate further investigations on developing the AXAI capability and use of a suitable SDDP for incorporating them in AI systems.

Published in: IEEE Access (Volume 10)

DOI: 10.1109/ACCESS.2022.3163523
