

School of
Electrical Engineering, Computing and Mathematical Sciences

Advanced Deep Learning for Medical Images Analysis

Bayu Adhi Nugroho

This thesis is presented for the Degree of

Doctor of Philosophy

Curtin University

February 2022

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for any other degree or diploma in any university.

Perth, 4 February 2022

Bayu Adhi Nugroho

The thesis is dedicated to
improve the quality of healthcare system
with the use of computing technology

The Publication Arising from Thesis

Bayu Adhi Nugroho. An aggregate method for thorax diseases classification. *Scientific Reports*, 11(1), February 2021.

Statement Contribution of Others in The Publication

The published manuscript [[128](#)] is single-authored by Bayu Adhi Nugroho.

Acknowledgements

Some of the contents from Chapter 2 and Chapter 3 have been published in the refereed Springer Nature's - Scientific Reports journal [128]. The manuscript is also available on the public preprint server [126].

The extension of the journal article [128] into Chapter 3 was involving the use of a private dataset [91].

The author of the thesis is very thankful to Dr. Johannes Herrmann and Prof. Iain Murray for the overall supervision and help.

The thesis was edited by Elite Editing, and editorial intervention was restricted to Standards D and E of the Australian Standards for Editing Practice.

Abstract

The application of deep learning is evolving, including in expert systems for healthcare, such as disease classification. There are several challenges in the use of deep-learning algorithms in application to disease classification. First, deep learning for classification performs supervised training, which means the algorithm requires large amounts of labelled examples. Satisfying this requirement is not easy in health care. Both high-quality samples (e.g. high-resolution images) and human experts may not be readily available to create the required number of samples. Second, deep learning performs end-to-end training, which means the algorithm will require costly computing power and uses inputs directly without pre-processing during the training process. Third, when a particular disease is more health threatening or has rarer samples to analyse than others in a group, a unique deep learning algorithm may need to be applied to provide results specific to the classification of that specific disease.

This thesis aims to improve classification to address the above mentioned problems. First, this thesis proposes a cost-sensitive imbalance training algorithm to address an unequal number of training examples, further improving classification performance. The results from the cost-sensitive algorithm have achieved state-of-the-art performance for a commonly used dataset. Second, the thesis proposes a two-stage Bayesian optimisation training algorithm. The experiment shows that the algorithm can reduce computational cost and maintain the classification performance. Third, the thesis proposes a dual-branch network for the training of a one-class classification scheme. We use 14 one-class classifiers; each of them was trained only with positive examples.

The experiment shows that the algorithm preserves classification performance without the presence of counterexamples during training.

Contents

Chapter 1: INTRODUCTION	1
1.1 Challenges and Gaps	2
1.2 Thesis Objectives and Chapters' Contributions	4
1.3 Structure of the Thesis	6
Chapter 2: BACKGROUND AND LITERATURE REVIEW	8
2.1 Chest Diseases	8
2.1.1 The Taxonomy of Chest Diseases	9
2.1.2 A Brief Overview of Chest Diseases	10
2.1.3 Chest Diseases Classifications	11
2.2 A Taxonomy and History of the Healthcare Expert System	15
2.2.1 The Rule-Based Expert System for Health Care	17
2.2.2 Applications of Traditional Machine-Learning Methods for Health Care	17
2.2.3 The Convolutional Network Applications for Health Care	19
2.2.4 The DenseNet-121 Network	24
2.2.5 EfficientNet	25
2.2.6 The Conclusion: Traditional Classifiers Versus Neural Net- works	26
2.3 The Binary, Multiclass and One-Class Classification Overview	27
2.3.1 Binary Classification	28
2.3.2 Multiclass Classification	29
2.3.3 One-Class Classification	30
2.4 The Computational Cost of the Use of Features for Classification Task	33

2.5	The Datasets of Medical Images	36
2.6	Discussion	37
Chapter 3: IMBALANCE CLASSIFICATION: THE AGGREGATE		
METHOD FOR CHEST CANCER CLASSIFICATION		38
3.1	Introduction	38
3.2	Method	39
3.2.1	The Existing Weights Function and Network Architecture	39
3.2.2	Proposed Weights Function and Network Architecture	40
3.2.2.1	The Weighted Cross-Entropy	42
3.2.2.2	The Weighted Focal Loss	42
3.2.2.3	Progressive Image Resizing	42
3.2.2.4	The Network Backbone	43
3.2.2.5	Baseline	43
3.2.2.6	Performance Evaluation	43
3.3	Research Contribution and Novelty Statement	44
3.4	Experiments and Results	44
3.4.1	Backbone Network Training	44
3.4.2	Weighted Binary Cross-Entropy with Effective Number of Samples	46
3.4.3	Weighted Focal Loss with Positive and Negative Pattern	46
3.4.4	Generalisation of the Weights Formula into the Glaucoma Classification Problem	51
3.5	The Intuitive Theoretical Background and Evidence from Experiment	53
3.6	The Imbalance Metric Evaluation	55
3.7	Third-Phase Training Saturation	56
3.8	Discussion	57
Chapter 4: HYPERPARAMETERS AND NETWORK ARCHITECTURES LEARNING FOR FEATURES CLASSIFICATION		
4.1	Introduction	60
4.2	Method	62
4.2.1	Research Contributions and Novelty Statement	62

4.2.2	The Existing Computational Cost for Neural Network	62
4.2.3	The Proposed Approach for Reducing Computational Cost .	63
4.2.3.1	The Kernel of Gaussian Process	64
4.2.3.2	The Partition of Iterations	65
4.2.3.3	The Acquisition Function	65
4.2.3.4	Gaussian Noise	68
4.2.4	The Total FLOPs/MACCs Calculation	68
4.2.5	The Training Epochs and Total FLOPs Correlation	69
4.3	Experiments and Results	70
4.3.1	The Generalisation of the Method for the Skin Cancer Clas- sification	74
4.3.2	The Applicability into Mobile Device	76
4.4	Discussion	78
Chapter 5: ONE-CLASS CLASSIFICATION		79
5.1	Introduction	79
5.2	Method	80
5.2.1	Research Contribution and Novelty Statement	80
5.2.2	The Existing One Class Classification	81
5.2.3	The Proposed Approach for One-Class Classification	82
5.2.4	The Best Outputs from Two Branches	84
5.2.5	The Fine-Grained Computational-Cost Evaluation	84
5.3	Experiment and Results	87
5.3.1	Effectiveness of the pre-Trained Weights	90
5.4	Discussion	91
Chapter 6: CONCLUSION		93
6.1	Summary of Contributions	93
6.2	The Progressions from Chapters	95
6.3	Future Works	96
Appendix A: THE CODE LISTING		97
Appendix B: THE LIST OF REPOSITORIES		99

Bibliography

100

List of Figures

1.1	Pipeline of The Research	6
2.1	The Anatomy of The Lung [18]	9
2.2	Training Distribution from Official Split	12
2.3	The Taxonomy of CAD Expert System [75, 93, 114, 138]	15
2.4	The Taxonomy of Skin Cancer Lesions [49]	20
2.5	The One Class Classification (OCC) Taxonomy [90]	30
2.6	The Datasets' Timeline	36
3.1	Area Under Precision-Recall Curve	56
4.1	The Width of ω and Balance of Exploration-Exploitation	68
4.2	The Mobile App Detects Cardiomegaly	77
4.3	The Mobile App Detects Infiltration	77
5.1	Two Tier Learning [29]	81
5.2	A Dual-Branch Network with Six Outputs	82
6.1	The Chapters' Progressions	95
A.1	Training Label Modification for One Class	97
A.2	Weighted EI Acquisition Function	98
A.3	Calculate FLOPs from The Model	98

List of Tables

2.1	The Imbalanced Number of Samples	13
2.2	The Comparison of Expert System Approaches	16
2.3	The Comparison of CNN and Non-CNN	16
2.4	The Layer Comparison DenseNet-121 and ChexNet	24
2.5	The EfficientNet-B0 Layer [161]	25
2.6	Decision Boundary	27
2.7	The Datasets	37
3.1	Results from Various γ for Focal Loss	47
3.2	Identical Split Comparison [139]	48
3.3.A	Improvement Rate	49
3.3.B	Improvement Rate (cont.)	49
3.4	Results from Five-Folds Cross-Validation	49
3.5.A	Comparison Results with Previous Research under The Official Splits	52
3.5.B	Comparison Results with Previous Research under The Official Splits (cont.)	52
3.6	Glaucoma Cases	52
3.7	Effectiveness of Weights for Glaucoma Classification	53
3.8	The Improvement of The Proposed Weight Calculation	54
3.9	The AU-PRC Improvement	55
3.10	Third-Phase Training Results from Table 3.2	57
3.11	The Heatmap from Different Methods and Various Networks	59
4.1	Results Comparison with Previous Study under The Offi- cial Splits	71
4.2	Identical Split Comparison [139]	72

4.3.A	Improvement Rate	75
4.3.B	Improvement Rate (cont.)	75
4.4	The Benchmark under Average Precision Metric	75
4.5	The Benchmark under Specificity Metric	75
4.6	The Benchmark under Sensitivity Metric	76
5.1	Decomposition of Backpropagation Algorithm [116]	85
5.2	The Notations [116]	85
5.3	Identical Splits Comparison with Previous Work	87
5.4	The Official Splits Comparison with Previous Works	88
5.5	Identical Splits Comparison with Multiclass Classification	89
5.6	The Official Splits Comparison with Multiclass Classification	90
5.7	Identical Splits Comparison Between from Scratch and Pre- Trained Initialisation	91
5.8	The Official Splits Comparison Between from Scratch and Pre-trained Initialisation	91

Chapter 1

Introduction

There is a crucial shortage of medical experts and many demands to serve patients with adequate diagnoses in a short period. The expert system is the de facto answer to transfer knowledge from medical experts into computer hardware. There are also some cases in which doctors falsely diagnose patients [30]. This research reduces the risk [110] because the supervised learning method uses a dataset provided with labels from multiple experts. This procedure reduces the risk of incorrect labels being applied by a single expert. Different experts may have different opinions about the labelling for an individual item in the dataset; the differences may occur due to the experts' knowledge, the level of expertise or the subjective preferences [167].

An expert system technology is a subset of Artificial Intelligence that simulates the process which experts use to solve problems [24]. The machine-learning-based method is considered state-of-the-art in expert system technology. There are many machine-learning approaches for the expert system, and a prominent subset is deep learning. A primary advantage of deep learning over other machine-learning methods is that it does not require extensive feature engineering to feed the algorithm.

One prominent deep learning algorithm is the convolutional neural network. The convolutional neural network's (CNN's) behaviour, which reduces inputs from the upper layer into the bottom layer, is advantageous. This process automates and supersedes manual feature engineering, which is commonly found in traditional machine-learning approaches. The lowest layers in the deep network only consist of valuable features for the final node's classification. In terms of the capability

to automatically separate only helpful features to perform classification, only a decision-tree-based classifier works similarly with the use of Gini impurity and entropy [92, 179, 188].

Deep networks also inherit an essential strength from multilayer perceptrons (MLP), the capability to distinguish non-linearly separable data. This case is also widely known as “the XOR problem”. The XOR problem requires a classifier to develop a non-linear decision boundary. A support vector machine (SVM) manages this problem by mapping data into the higher dimensions—the approach is widely known as the SVM “kernel trick”. Since a deep neural network is the more complex version of an MLP, it handles the non-linearity of data without further extension.

There are cases when obtaining the labelled examples to suffice the training algorithm is complex. The problem is not rare in medical cases for which the availability of data is also subject to experts’ availability. However, the shortage of high-quality examples may lead to further technical problems, such as imbalance, which later inherently reduces the algorithm’s performance and the limitation of classifying minority cases. In developing a unique method to classify a particular class of interest, traditional classifiers accompany methods such as one-class SVM and one-class nearest neighbour. The further extension to use a deep-network-based classifier to improve the medical classification is a promising subject of study.

Despite the preference for deep learning over other machine-learning approaches, it has numerous challenges. This thesis will discuss the proposed approaches to improve the challenges of deep learning in the field of medical images. This research identifies several significant challenges, which are discussed in Section 1.1.

1.1 Challenges and Gaps

The challenge of the class imbalance within a medical images dataset is that the number of images from some important diseases are much smaller than others. Also, the numbers of positively labelled images with diseases are much smaller than the images that are negatively labelled with diseases (healthy). One conventional approach to overcome the imbalance problem is “class-weighting”, which aims to provide cost-sensitive learning. However, there are various methods to perform

class-weighting. The significance of this research is that it contributes to the calculation of the importance of negatively and positively labelled images into the form of weights in the cost function during training. This research addresses this in [128] and Chapter 3.

The challenge is the limited dataset, owing to the high cost of labelling medical data. Deep-learning algorithms work well when a large amount of training data is available, but have limitations when only a small, annotated dataset is available. Further, the algorithm mostly works in supervised learning, which requires well curated (labelled) data. The significant contribution of this research is that we propose using cost-sensitive learning and one-class classifiers to alleviate the problem. This research addresses this in [128] and Chapter 5.

There is also the challenge of time-consuming training. Deep-learning algorithms are well known for the long training periods required to achieve small improvements. This research proposes the use of extracted feature vectors from deep-network backbones to mitigate the use of high-sized medical images and reduce the time required to train the deep-network. This research addresses this in Chapters 4 and 5.

Network architecture is critical for image classification performance and different architectures are required for various application problems. The architecture learning and hyperparameter tuning will be investigated for efficient classification. This research addresses this in [128] and Chapters 3 and 4.

Some classes of diseases are more dangerous or rarer than others; this research aims to detect the most dangerous or smallest numbered samples of diseases using a one-class deep-network classifier. This research expects to have robust classifiers for specific cases. During our literature review, we did not find any peer-reviewed paper that provides a deep-network classifier solution for this problem ¹. This research addresses this in Chapter 5.

¹The latest query: “one class deep learning neural network for medical classification” through Google (April 2021).

1.2 Thesis Objectives and Chapters' Contributions

This research formulates aims and objectives through this study of deep-learning algorithms:

1. To use novel approaches to classify diseases based on deep learning and improves existing results with state-of-the-art imbalance optimisation techniques. This thesis investigates the effectiveness of using imbalance learning for classification improvements. The primary advantage of imbalance learning optimisation is providing better training procedures for learning deep networks when only limited high-quality original images are available. The works involved are outlined in [128] and Chapter 3. The Chapter 3 contribution is to propose an approach that combines a weights calculation algorithm for deep networks with the optimisation of training strategy from the state-of-the-art architecture.
2. To alleviate the costs of training the neural network using extracted features and provide deeper analysis from the architectural neural network perspective with a novel bayesian-optimisation training scheme. The research analyses disease classification using extracted features from the neural network. Architectural learning and hyperparameters tuning will be investigated. The works that include parameter tuning are in [128] and are listed in Algorithm 1 of Section 4.2.3.3. A primary distinction is that in [128], the proposed method applies to the original images' inputs. In Algorithm 1 of Section 4.2.3.3, it applies to the feature vectors inputs.

The Chapter 4 contributions are (i) the comparable classification performance, using minor FLOPs neural-network architecture and Bayesian Optimisation for the features classification task (ii) the proposed Bayesian iteration-partitioning framework works both for the Chest X-Ray dataset and the skin cancer dataset (iii) the proposed method forces a more deterministic Bayesian-Optimisation, ensuring the maximum magnitude of results is achieved whilst minimising the time required (iv) the proposed method's applicability to mobile devices into implementation. The results are empirically

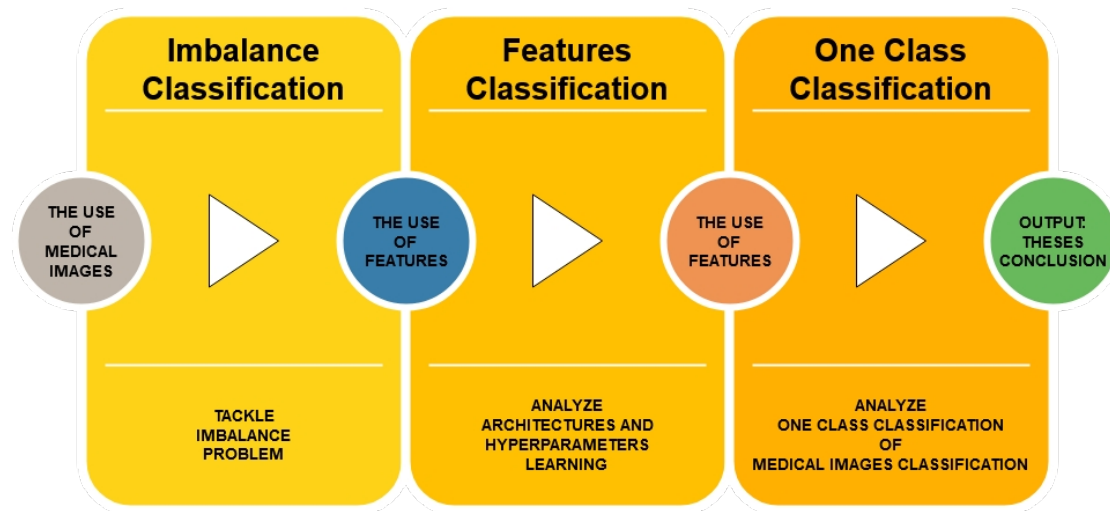
reported and tested both on the Android simulator and the actual device (Samsung Galaxy S8) (v) all the source codes are shared publicly [127] for further study, ensuring that the works are repeatable and well-documented.

3. To use a novel neural network architecture and algorithm for a one-class classification method in classifying diseases. The research investigates the one-class classification method to classify the deadliest class of diseases or the classes of interest or minor cases with deep networks in the proposed medical image domain. This research focuses on a particular class for each disease of interest, since in the medical image domain, most likely there is a particular class that is more lethal than others, or there is a particular class that is rarer than others. Further, there is an urgency to analyse several classes. To provide sound classification performance for severe or minority diseases, unique tuning and network learning will be used. The work involved is outlined in Chapter 5.

The Chapter 5 primary contribution is that it proposes a dual-branch network architecture to train a one-class classifier without the presence of counterexamples. This study can show the advantage of the proposed method to achieve better results than previous studies [26, 62, 64]. It also requires fewer computational costs during training.

The works written in the thesis flow through the pipeline depicted in Figure 1.1.

Figure 1.1: Pipeline of The Research



This research informs the flow of the thesis for better reading and compatibility with the research pipeline, as depicted in Figure 1.1.

1.3 Structure of the Thesis

Chapter 2 provides the background and preliminary knowledge of the topic. It covers several important issues: the comparison of various expert systems, the comparison of traditional machine-learning versus deep-learning algorithms, the comparison of the traditional images classification task with the task of features classification, and details of the network architectures used during the study.

Chapter 3 discusses the research to improve classification performance from the medical images dataset with the proposed cost-sensitive approach. This chapter minimises the effect of imbalance examples in the dataset during the task of disease classification. The primary results from the study have been published in the refereed Springer Nature's Scientific Reports journal [128].

Chapter 4 discusses the research to better understand the trade-off between classification performance and computational cost. This chapter proposes the two phases of Bayesian optimisation for features-based classification.

This approach results in lower computation costs and competitive classification performances.

Chapter 5 explores the proposed network architecture with one-class classification training scheme and its usage in medical images classification tasks. This chapter proposes that the network trained in the one-class fashion have a fine-grained analysis of classification performance and the required computational cost. The results show that the proposed one-class classification is helpful to improve medical image classification. Chapter 6 provides conclusions and recommendations for future research.

Chapter 2

Background and Literature Review

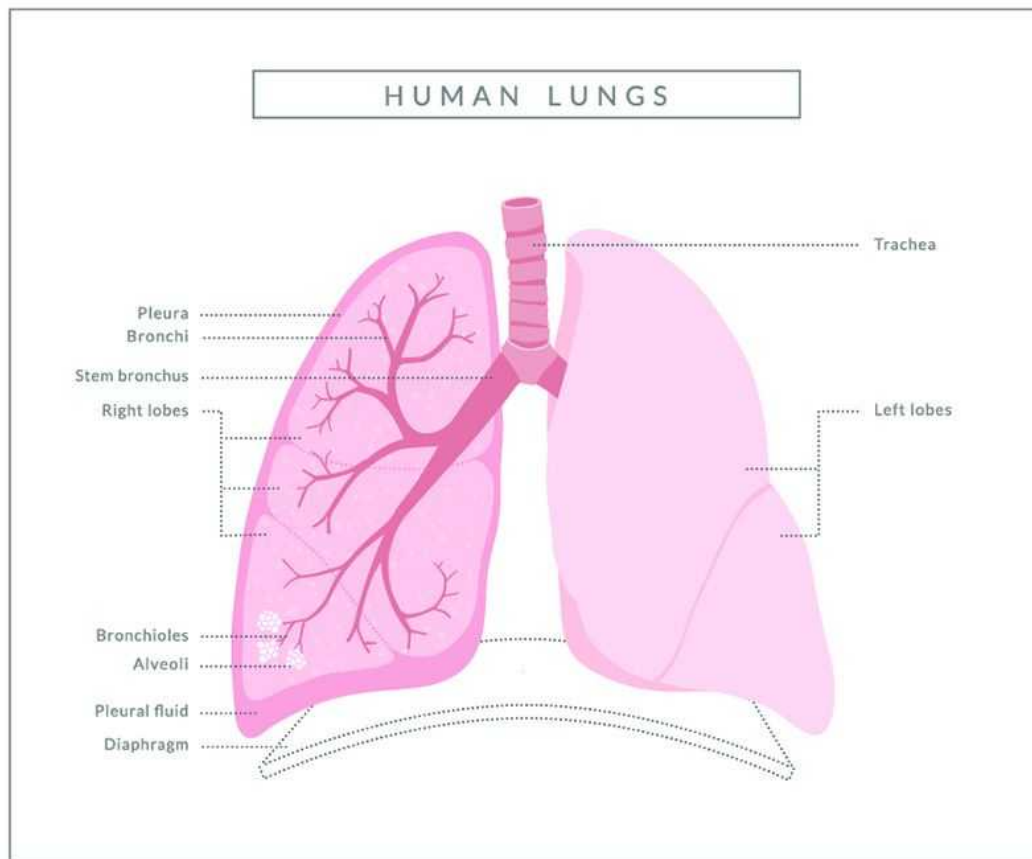
2.1 Chest Diseases

A study of the financial cost of diseases in 2015 showed that respiratory diseases have contributed to \$4 billion in Australian healthcare expenditures [6]. The latter study [16] breaks down the cost of diseases into direct and indirect costs. The direct cost refers to personal expenses related to direct access to health care. The indirect cost refers to personal expenses caused by the reduced workload because of the illness. The direct cost of asthma for each person in a year is up to USD3,000 in the United States (US) [16]. The US population of people over 18 in 2019 is 255,042,109 [36]; 8% of people over 18 severe had asthma in 2019 [19]. This study can infer that in 2019, asthma, among other lung problems in the US, cost USD61 million. The highest indirect cost for asthma is recorded in South Korea, up to USD1,274 for each person in a year [16].

According to the Australian Bureau of Statistics from the 2017 - 2018 National Health Survey, 31 % of Australians suffer from chronic respiratory disease, roughly 7.4 million people [5]. The two most common chronic respiratory diseases in Australia are chronic obstructive pulmonary disease (COPD) and asthma [5]. In 2003 COPD was the primary cause of death in Australia, with 5,400 recorded death cases [7]. Later, in 2018, COPD was the fifth major cause of death in Australia, with 7,113 deaths recorded [3]. Still, in 2018, a related chest disease, lung cancer, was in the fourth position with 8,586 deaths [2]. During 2015 - 2016 COPD

cost \$977 million in the Australian health system [4]. In detail, \$536 million was the cost of hospitals, \$189 million for non-hospital services and \$252 million for pharmaceuticals [4].

Figure 2.1: The Anatomy of The Lung [18]



2.1.1 The Taxonomy of Chest Diseases

From the disorder perspective, there are three types of major lung diseases [20]:

1. airway-related diseases, the conditions in which the disease obstructs the flow of gasses in the lung
2. tissue-related diseases, in which disease influences the structure of lung tissue

3. blood circulation-related diseases—conditions in which the disease attacks the lung’s blood vessels.

In terms of the malignancy perspective, there are two prominent characteristics of lung diseases :

1. non-cancer/non-malignant lung diseases—this includes asthma, interstitial lung diseases (ILD) and chronic obstructive pulmonary disease (COPD). ILD is a group of disorders that produce scars in the lungs. COPD is a chronic obstruction in the lung’s airflow.
2. cancerous lung diseases—these consist of small cell lung cancer and non-small cell lung cancer. The most common cause of lung cancer is smoking behaviour.

2.1.2 A Brief Overview of Chest Diseases

In Section 2.1.2, the thesis reviews specific chest diseases that will be used in the research. The discussion includes diseases that are potentially contagious, like pneumonia and diseases leading to complications and death, like pneumothorax.

Generally, cardiomegaly is the abnormal enlargement of the heart. The clinical signs of this disease are commonly found through chest X-ray diagnosis. According to Amin (2020) [22], “A Chest X-ray with an enlarged cardiac silhouette and a cardiothoracic ratio of more than 50% is suggestive of cardiomegaly”.

Edema is the accumulation of fluids within the lungs in an abnormal condition [15]. Edema collects fluids within air sacs, “alveoli” of the lung, then it causes breath shortness. Another disease, pleural effusion, collects fluids around the chest, surrounded by the cause of the abnormality [12]. The primary difference between edema and pleural effusion is the location where the fluids obstruct the lungs to function properly. In pulmonary edema, the fluids build up in the air sacs, while the pleural effusion collects fluid around the lung’s surroundings.

Pleural thickening is the condition in which the “pleura”—the special area between the lung and chest wall—become thickened then develop scars in the pleural tissues. The most common causes of pleural thickening are the inhalation of special fibres called asbestos. Pleural thickening cannot be cured, but the right

treatment will improve the patient’s life quality. However, pleural thickening has a high likelihood to reduce an individual’s life expectancy. Pulmonary fibrosis is a condition in which the lungs develop scars and become thickened in the air sacs (also widely known as alveoli). When the cause of scars is unknown, the more succinct term is “idiopathic pulmonary fibrosis”. The more recent studies attempt to investigate whether COVID-19 can lead to long-term pulmonary fibrosis [130, 154].

Emphysema is another name for chronic bronchitis; the symptom is a constant cough with phlegm [129]. Because it can cause severe damages to alveoli, emphysema is categorised as a COPD [9].

Atelectasis is a condition in which the lung cannot enlarge properly [1]. The other name for atelectasis is “the collapsed lung” or “the shrinking lung” [73]. The condition mainly results from reduced volume from the lungs, producing further unwanted blocked airways within the lungs. A collapsed lung occurs when the air breaks into the pleura. If the collapse only affects a part of the lungs, it is an atelectasis. However, when the lungs experience total collapse, a more proper term for the condition is pneumothorax [17].

The pulmonary hernia is a condition in which part of the lung pushes the weak spot of the chest’s wall [145]. The effort to cure the patient is through a surgical procedure and medical imaging products (e.g. computed tomography [CT] and magnetic resonance imaging [MRI]) to recommend a surgical road map [37].

Pneumonia is a condition in which the lung has infections caused by virus or bacteria or fungi [13, 14]. Pneumonia results in the alveoli of the lung filling with fluids or pus [14]. A recent study [189] provided conclusions to differentiate common pneumonia and pneumonia resulting from COVID-19 in the chest X-ray.

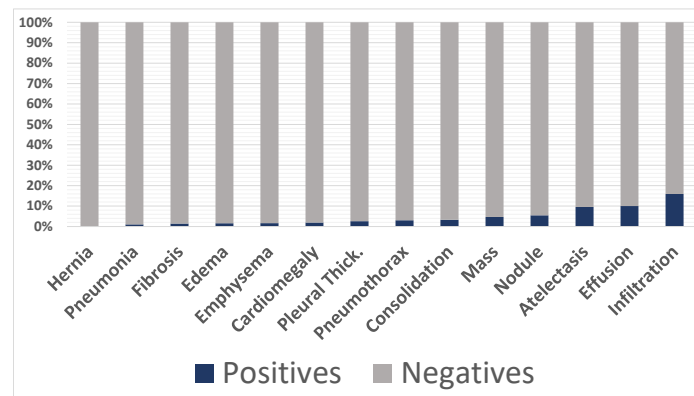
2.1.3 Chest Diseases Classifications

A review paper from Ginneken (2017) [169] discusses several issues from the computing domain in term of its application in the field of chest imaging. These issues involve:

1. rib suppression—the objective is to subtract the bones from the image and provide a better image of tissues for analysis

2. fissure detection—aimed to emphasise the area boundaries that describe the diseased lobes in the chest
3. airway segmentation—since several lung diseases obstruct the airway, to have a precise understanding of the airway’s location is crucial
4. nodule detection and classification—in some cases the existence of nodules in the image may represent the occurrence of cancers.

Figure 2.2: Training Distribution from Official Split



Two deep neural networks, ChexNet and CheXNeXt, are proposed in Rajpurkar et al. (2017) [136] and Rajpurkar et al. (2018) [135] respectively; both use Densenet-121, which has been pre-trained with ImageNet. The primary differences between these two methods are the labelling of the training datasets. The latter relabelled the partially incorrect labels in the dataset, while the first work trained the learning network based on the original National Institutes of Health (NIH) labels. Recent works by Jaipurkar et al. (2018) [82] claimed that their results are better than Rajpurkar et al. (2017) [136], but the results are not equally comparable since the work of Jaipurkar et al. (2018) [82] used a different testing subset of the dataset in the performance evaluation. The experimental results reported in Rajpurkar et al. (2017) [136] are based on the testing subset; those from Jaipurkar et al. (2018) [82] are based on the validation subset.

Table 2.1: **The Imbalanced Number of Samples**

Samples	Number
Healthy	60,361
Hernia	227
Pneumonia	1,431
Fibrosis	1,686
Edema	2,303
Emphysema	2,516
Cardiomegaly	2,776
Pleural Thick.	3,385
Consolidation	4,667
Pneumothorax	5,302
Mass	5,782
Nodule	6,331
Atelectasis	11,559
Effusion	13,317
Infiltration	19,894

However, Jaipurkar et al. (2018) [82] proposed new methods to address the dataset imbalance and improve the signal-to-noise ratio. [82] pruned the training dataset since they found many records that have no classes (unlabelled) and have blurry images. Therefore, they decided to use only the labelled classes with good-quality source images.

Bhatia et al. (2019) [31] used a CT scan in DICOM format of lungs to predict cancer. The method [31] extracts DICOM images with the use of deep residual networks. This stage produces the extracted features. The next step is to perform the classification with the use of ensemble classifiers and extracted features.

According to Baltruschat et al. [26], the current state-of-the-art performance for the dataset [173] classification performance was achieved by Gündel et al. [64]. Further research by Guan et al. [62], which used three-phase training procedures, reported better performance than Gündel et al. [64]. However, the work [62] did not share the split sets, which is critical for the performance evaluation. Further, the re-implementation by another party in Github [139] reported lower results. This study also noticed that the re-implementation [139] of [62] did not share identical sets with the work of Gündel et al. [64]. Baltruschat et al. [26] noticed that different split sets will lead to different performances for the dataset [173]. The chest X-ray dataset [173] was used to evaluate the performance of the proposed

method. It contains 112,120 chest X-ray images from 30,805 unique patients and has multilabels of 14 classes of disease.

The image resolution is 1,024 x 1,024 with the 8-bit channel. This research downsampled the resolution as 224 x 224 and converted the channel into Red-Green-Blue (RGB), which can be adopted to our backbone network. Chest X-Ray 14 only consists of frontal-view images. It does not have any lateral-view cases. The number of positive samples for each class is much less than the negative samples, as depicted in Figure 2.2.

To develop a neural network for medical diagnosis, patient data are necessary; however, the positive class is in a minority and the negative class is in a majority. The neural network is biased to the majority class and has poor performance on the minority class. The common methods to balance the number of positive and negative class for the traditional classifiers is by undersampling and oversampling. After applying those methods, the numbers of each training pattern are equal.

A more sophisticated approach might involve algorithmic techniques to perform cost-sensitive training [85]. Cui et al.'s definition of the effective number of samples: "the effective number of samples is defined as the volume of sample" [44]. Cui et al. [44] developed a metric to determine the effective samples and reformulated the loss function based on the numbers of effective samples in the positive and negative classes. Wang et al. [173] contributed by providing a novel chest X-ray dataset. Both [64, 173] used the same method to balance the dataset; however, the method from [64, 173] is different than the method from [44] to address the imbalance problem.

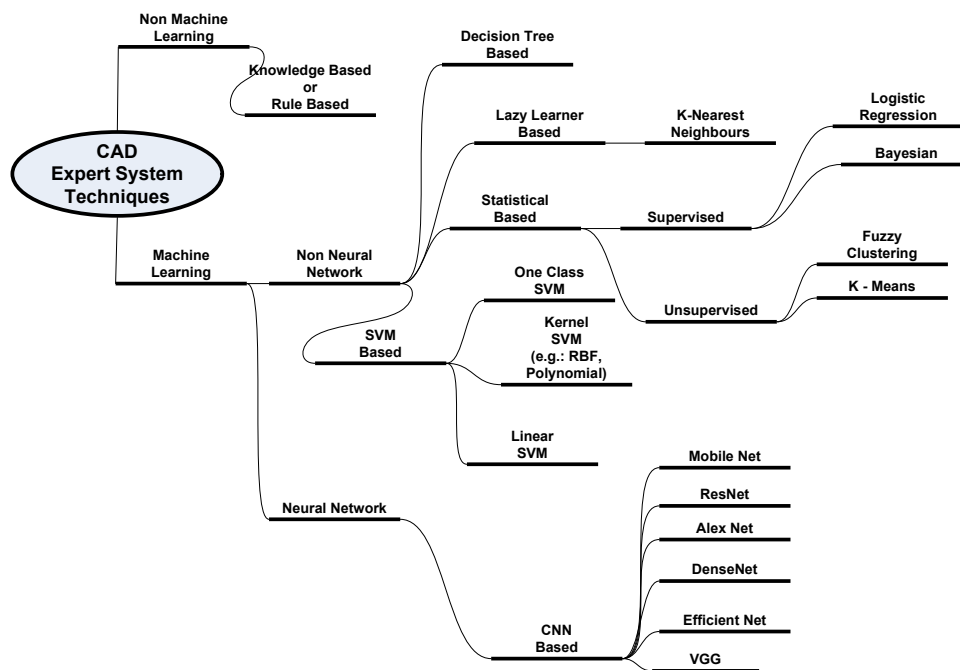
Cui et al.'s approach [44] treats the contributions of training patterns to the loss function equally for all output nodes; this is contrary to Wang and Gündel et al.'s methods [64, 173], which use the distinct weights from positive and negative patterns as the multipliers in the loss function. Although better classification performance can potentially be achieved by Cui et al.'s approach [44], it only addressed effective samples [44] and the imbalances of positive–negative patterns were not tackled.

In summary, Chest X-Ray image specific problems are the minimum labelling from the experts, the data imbalance, and the diseases' visibility in the Chest X-Ray images.

2.2 A Taxonomy and History of the Healthcare Expert System

Yanase and Triantaphyllou (2019) [75] broke down the computer-assisted diagnosis (CAD) system into two broad technical categories. The first is the knowledge-based system, also widely known as the rule-based system. The second is machine-learning types. This type of categorisation is also followed by other literature [93, 114, 138]. The taxonomy of the expert system depicted in Figure 2.3 is our compilation from literature [75, 93, 114, 138].

Figure 2.3: The Taxonomy of CAD Expert System [75, 93, 114, 138]



The machine-learning healthcare-based expert system’s main characteristic is generating a model by learning from a set of data inputs, commonly referred to as the “training” process. The data inputs are further called the “training data”. Conversely, a non-machine learning type—such as in the rule-based system—involves the inference process from a set of rules to apply a supportive medical analysis decision. The rule-based system requires a process called “knowledge engineering”,

which transfers the expertise into a set of rules. In the machine-learning-based system, this has been done through the “feature engineering” and “training” process. Table 2.2 depicts the strengths and weaknesses for each approach. Table 2.3 summarises the details between CNN and non-CNN architectures.

Table 2.2: **The Comparison of Expert System Approaches**

Technical approach	Knowledge transfer method	Knowledge input method	Required computational resources
Rule-based expert system	Knowledge engineering	Expert interviews then hard-code	Low
Non neural network expert system	Training	Through feature selection or feature engineering	Low to high
Neural network expert system	Training	Raw input (text or images) also features (e.g.: extracted features)	High

Table 2.3: **The Comparison of CNN and Non-CNN**

Neural Network Name	Type	Labels	Special Characteristics
RNN / LSTM / GRU	Non CNN	Supervised or Semi Supervised or Unsupervised	Handling sequential time data
Autoencoder	Non CNN	Unsupervised	Encoder decoder
Convolutional Autoencoder	CNN	Unsupervised	Convolutional encoder decoder
Resnet	CNN	Supervised	Image downsampling and residual layer
DenseNet	CNN	Supervised	Image downsampling and dense layer
EfficientNet	CNN	Supervised	Image downsampling and architecture search

2.2.1 The Rule-Based Expert System for Health Care

Several classical CAD expert systems gained popularity because they are the “early pioneers” in the field, such as MYCIN (1975) [150] and INTERNIST-I (1982) [113]. The IF-THEN rules in MYCIN [150] were set based on Bayes theorem, while in the INTERNIST-I [113], the inference comes from a collection of symptoms as the core supporting knowledge.

Bindoff et al. (2006) [32] proposed an incremental updating rule-based recommendation system for a medication review. Whenever a new case arrives as input, the system identifies whether the existing rules need to be updated. If updates are required, the system will require the human expert to make the amendments; then the new rule sets will be applied based on the human expert’s decision.

A rule-based alerting system to provide a remote heart-failure monitoring service from Seto et al. (2012) [146] comprises the initial rule sets, which were created by interviewing 10 clinician experts. The other nine clinician experts validated the initial rule sets. The outputs are eight types of alert messages that provide recommendations on the basis of the input variables taken from the patients (weight, blood pressure and heart rate).

A rule-based system is understandable and straightforward. However, this study can also conclude several disadvantages:

1. There is a bottleneck in term of interviewing the experts to acquire knowledge.
2. It is required to update the rule in the system to accommodate the new knowledge.
3. There is a complicated problem when experts disagree. In the supervised machine-learning system, an ordinary resolution for this problem is to make the decision based on majority vote.

2.2.2 Applications of Traditional Machine-Learning Methods for Health Care

The thesis discusses some of the classical machine-learning approaches for health-care applications in Section 2.2.2. kNN (k-nearest neighbours) is a lazy learner

method that builds a model when the query is submitted [57]; it contradicts the eager learner method that builds the model during training. The letter “k” refers to the k-number of examples used to determine the label outputs for the new examples submitted in the testing query. Examples of applications in the kNN algorithm’s health care are outlined in [48, 131, 170]. The distance between the training and test set was calculated using the Euclidean function [48, 131, 170].

Oliva and Rosa (2016) [131] distinguished normal versus epileptic (abnormal) electroencephalogram (EEG) with kNN using cross-correlogram (CCo) features from 200 EEG segments. The kNN with $k = 1$ achieves the highest negative predictive values 91.18% NPV (91.18% is the normal fraction correctly classified from the actual normal and false-negative normal), and 90.91% sensitivity (90.91% likelihood that abnormalities of abnormal cases will be correctly classified). The kNN with $k = 7$ achieves the positive predictive values 98.88% PPV (98.88% abnormal fraction correctly classified from the actual abnormal and false-positive abnormal) and 99.00% specificity (99.00% likelihood that normalities of the normal cases will be correctly classified). The results were evaluated using 10-fold cross-validation; there were no significant differences in each fold.

Elsayed and Syed (2017) [48] applied the kNN classification for Framingham heart decision support in the cardiology field. The Framingham study set the risk factors for early risk detection of coronary heart diseases. Data were taken from a hospital in Saudi Arabia. The data include the following features for each patient: age, gender, glucose level, total cholesterol, high-density lipoproteins, systolic blood pressure and family history (treatment for hypertension and smoking status). The kNN classifier’s highest accuracy is 66.7%.

Venkataramanaiah and Kamala (2020) [170] performed the kNN classifier on heart rate variability features from the electrocardiogram (ECG) signal, achieving the highest accuracy of 99%. The work [170] used a publicly available dataset [118]. However, Venkataramanaiah and Kamala (2020) [170] did not describe the detailed procedure used to develop the training and testing sets.

SVM is a binary classifier. It optimises the best decision boundary between two patterns with the help of support vectors. SVM is a linear classifier in nature, but kernel function (e.g., radial basis function [RBF], polynomial) will improve its

capability as a non-linear classifier. The examples of the applications in the SVM algorithm's health care are in [27, 181].

Son et al.(2010) [181] applied several SVM kernels (linear, polynomial, RBF, Sigmoid) to perform heart-failure (HF) classification tasks. The features were 11 variables of 76 patients: gender, age, spouse, education, monthly income and duration of HF diagnosis, daily frequency of medication, ejection fraction, minimal status examination-Korean (MMSE-K), medication knowledge and New York Heart Association functional class. The best accuracy achieved was 77.63%, with RBF kernel.

Battineni et al. (2019) [27] used SVM with non-linear RBF kernel to classify dementia using MRI inputs. The input features were MRI's longitudinal data, which refers to the MRI session from 150 subjects (373 MRI data), and resulted in 70% classification accuracy.

The decision-tree classifier splits features based on the criterion. Each split quality is determined by its "impurity". The measurement of impurity is either by the use of Gini index or entropy, with the range values $[0, 0.5)$ and $[0, 1)$ respectively. The Gini index values or entropy from lower nodes of a decision tree are near or equal to zero, which means lower nodes are purer than upper nodes. Chern et al. (2019) [39] attempted to identify patients who are eligible to receive insurance reimbursements for telehealth services (remote health services). The solution is to develop a binary decision-tree classifier. The work [39] applied the C4.5 algorithm, which uses entropy to build the decision tree. The training data are small—200 records. The labels come from three experts in the field; majority vote is used when experts disagree about a particular record's label. The final accuracy achieved was 98.5%.

The thesis has discussed several traditional classifiers approaches. To determine which is the best solution, it very much depends on the case.

2.2.3 The Convolutional Network Applications for Health Care

Esteva et al. (2017) [49] used 1.41 million images to train CNN for malignant skin cancer detection. The base architecture came from GoogleNet Inception v3,

which was pre-trained with 1.28 million images from the 2012 ImageNet recognition competition dataset. Using the concept of transfer learning, they trained and tested the architecture using their own 129,450 images, which were categorised into 757 fine-grained classes of lesion disease. The novelty of fine-grain classes in taxonomy within a disease partitioning scheme itself is a key contribution in this research. Part of the upper taxonomy from Esteva et al.(2017) [49] is shown in Figure 2.4.

The experimental results in [49] under 9-folds training configuration demonstrate that the classification models that have been trained using fine-grain partitions from nine class classification perform $72.1 \pm 0.9\%$ accuracy. In contrast, the coarser-grain partitions only perform $69.4 \pm 0.8\%$ accuracy. The fine-grain partitions from the three-class classification perform and $55.4 \pm 1.7\%$ accuracy, while the coarser-grain only perform $48.9 \pm 1.9\%$ accuracy. To recover the fine-grained the 757-classes classification results into nine-class and three-class, Esteva et al.(2017) sum the probabilities from the fine-grained descendants.

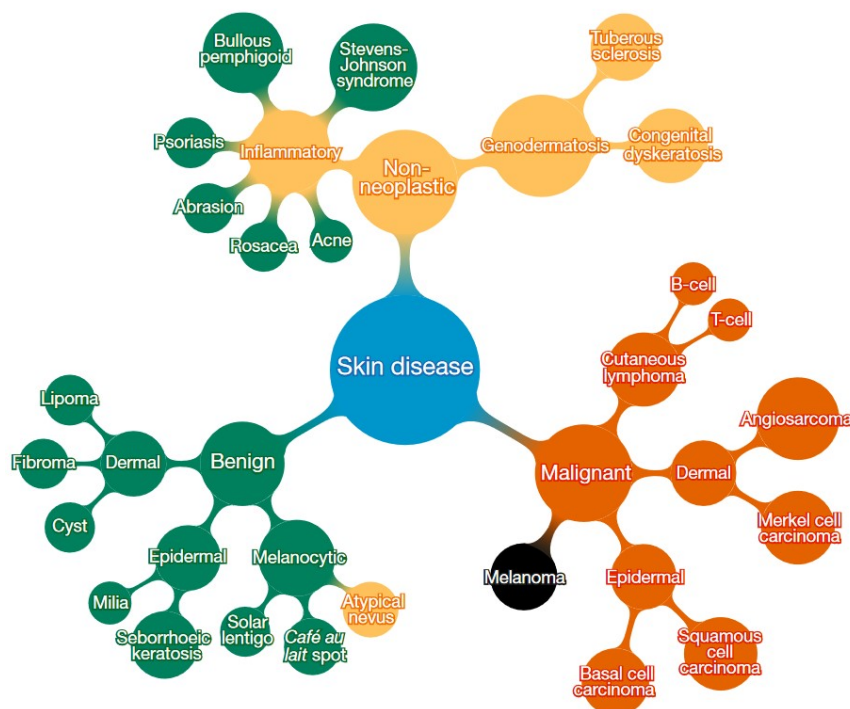


Figure 2.4: The Taxonomy of Skin Cancer Lesions [49]

Menegola et al. (2020) [168] used transfer learning, resulting from multi-way analysis of variance (anova) from several “factorial” experiments (factors/variables: network model, dataset, resolution, augmentation, normalisation, segmentation, training length, svm layer and weight transfer learning). Previously [111], Menegola et al. (2017) used transfer learning from models that were pre-trained from different image domains (between related domain—e.g. retinopathy and skin—versus unrelated domain—e.g. ImageNet and Skin). The experiments’ settings in [168] and [111] are different, but the objectives are the same: to provide evidence of the effectiveness of transfer learning.

The work of Kowsar et al. (2020) [94] performs hierarchical training to classify the severity of Celiac Disease. The private dataset contains three parent classes of bowel enteropathies: environmental enteropathy, celiac disease, and normal. Celiac disease has four fine-grained sub-classes: type I, type IIIa, type IIIb and type IIIc. Hence, Kowsar et al. designed two types of networks: the parent CNN with three softmax node-outputs and the child CNN with four softmax node-outputs.

Before being fed into the hierarchical training, the original images were pre-processed into 1000 x 1000 image patches. The original images were fed into an autoencoder to obtain the image patches. The bottleneck layer is in-between the middle of the decode-encode layers; the patches were taken from features in the bottleneck layer. The K-mean clustering was applied to the features; the clustering removed unused background patches. Only the valuable patches that contain meaningful information are fed to the network. Both the parent CNN and the child CNN were fed with patches rather than the original images.

Kowsar et al. [94] define the custom CNN networks for baseline. They argue that the existing pre-built networks (e.g. ResNet, Alexnet) are only capable of handling small images 250 x 250 and not suitable for handling 1000 x 1000 input. The hierarchical images classification (HMIC) system obtains $88.61 \pm 0.37\%$ F1-Score for non-whole slide classification, $90.89 \pm 0.38\%$ F1-Score for whole slide classification

Zhang et al. (2020) [186] proposed using active learning with CNN to address the imbalanced dataset problem. The work argues that an imbalanced problem may exist in the medical dataset due to the expensive cost of the labelling process.

Since there is a process to label the unlabelled dataset during training actively, some also mention “active learning” as “semi-supervised learning”. The term “active learning” means interventions from domain experts during the training process.

The proposed active-learning [186] compute similarity scores from feature vector inputs. The algorithm [186] decides manual labelling is necessary for two types of the unlabelled images—first, the unlabelled images with high similarity scores to the labelled. Second, the unlabelled images with high dissimilarity scores to the labelled, contrasting to the first. The first is assuming that similar images are under the same classes. They need manual labelling from domain experts. The second is under the assumption that highly dissimilar images are novel classes; also, they need manual labelling from domain experts.

The work [186] results in a 94% average-recall value with the Endoscopy dataset, and 92% average-recall value with the Caltech-256 dataset. The results were achieved with only 5.6% unlabelled examples from the Endoscopy dataset, and 7.5% unlabelled examples from the Caltech-256 dataset need manual labelling by experts.

The work of Galdran et al. (2018) [56] proposes improvements over Zhang et al.’s work (2018) [187], namely MixUp [187]. The early idea of MixUp [187] was to provide synthetic augmentations based on the interpolation algorithm. Initially, MixUp aimed to provide regularisation in order to prevent overfitting. Galdran et al. argue that the classical MixUp approach ignores the class distribution, resulting in underfitting for the minority class [56]. Therefore the balanced-MixUp [56] improves the MixUp approach by enhancing the sampling strategy into a more balanced majority-minority example in the training distribution.

The performance evaluation [56] was performed through quadratic-kappa metric and Matthews Correlation Coefficient, respectively, for the Diabetic Retinopathy grading’s - the Eyepacs dataset [8] and the Gastro-Intestinal images - the Hyper Kvasir dataset [10]. The quadratic-kappa metric results are 80.78 and 91.15 under ResNext-50 architecture, respectively, for the Eyepacs and Hyper Kvasir datasets.

The work of Liu et al. (2019) [102] does not use medical datasets. However, as it might help medical images research, makes it reasonably sufficient to mention

in the thesis. Liu et al. [102] propose a novel neural network architecture for 3D convolutions and uses 3D images to feed the network. 3D images occupy large GPU's memory, similar to high-resolution medical images.

The voxel-based and point-based models are the prominent architectures to perform convolution for 3D images. The first uses volumetric convolution, whereas the latter uses point-based convolution. The voxel-based has good memory locality but occupies memory cubically. On the other hand, the point-based is memory savvy but prone to irregular access behaviour.

Liu et al. [102] propose a point-voxel CNN (PVCNN) that integrates volumetric convolution with the point-based representation. Therefore, the PVCNN is both more space-savvy and enhances the small latency to memory access. The evaluation datasets were obtained from the case of indoor space segmentation using the S3DIS dataset [23] and 3D object detection using the KITTI dataset [55]. The indoor segmentation results in 14x memory speedup and 10x memory reduction. The 3D object detection results in 1.8x memory speedup and 1.4x memory reduction.

A review paper from Wang et al. (2020) [172] provides details of the deep neural network's transfer learning. Transfer learning is the method of applying previous knowledge from the source domain to the destination domain, where there exists relevant tasks between the source and the targeting domain. The manuscript classifies transfer learning into four types. The first is instance-based, the second is feature-based, the third is parameter-based, and the fourth is relational knowledge. The instance-based mainly is to amplify the subset of matching or the similar source domain's data into the targeting domain. The action would be effective when using the source's data directly without adjustments. However, the availability of a similar subset of source data may not always exist to match the destination domain.

The feature-based mainly transforms the source data into the feature spaces that match the targeting domain. The feature transformation will narrow the source and target domain gap. The parameter-based is mainly to fine-tune and use the parameters in the source domain as initialization to the targeting domain. Hence the targeting domain is learning to solve its problem by adjusting the source domain's parameter. The relational knowledge transfer-learning mainly occurs

when there are many compatible data points between the source domain and the targeting domain. Hence the source domain and the destination domain share some logical relationship intrinsically.

2.2.4 The DenseNet-121 Network

DenseNet-121 is popular to perform classification [173] with some other methods [26, 64, 173, 180] that use ResNet [71]. DenseNet [80] and ResNet [71] utilise different skip-connection approaches to pass features from previous layers to later layers. ResNet [71] performs a summation of features for the skip connections while DenseNet [80] performs concatenation from features. After the input layer, DenseNet utilises 7 x 7 convolution in a stride 2 mode and uses 3 x 3 max pooling in stride 2 mode. Then, it concatenates features in the first dense block.

Table 2.4: The Layer Comparison DenseNet-121 and ChexNet

Layers	Output Size	DenseNet - 121	ChexNet
	112 x 112 56 x 56	7x7 CONV stride 2 MAX POOL stride 2	7x7 CONV stride 2 MAX POOL stride 2
Dense block (1)	56 x 56	1 x 1 CONV x 6 3 x 3 CONV	1 x 1 CONV x 6 3 x 3 CONV
Transition (1)	56 x 56 28 x 28	1 x 1 CONV 2 x 2 AVG POOL stride 2	1 x 1 CONV 2 x 2 AVG POOL stride 2
Dense block (2)	28 x 28	1 x 1 CONV X 12 3 x 3 CONV	1 x 1 CONV X 12 3 x 3 CONV
Transition (2)	28 x 28 14 x 14	1 x 1 CONV 2 x 2 AVG POOL stride 2	1 x 1 CONV 2 x 2 AVG POOL stride 2
Dense block (2)	14 x 14	1 x 1 CONV x 24 3 x 3 CONV	1 x 1 CONV x 24 3 x 3 CONV
Transition (3)	14 x 14 7 x 7	1 x 1 CONV 2 x 2 AVG POOL stride 2	1 x 1 CONV 2 x 2 AVG POOL stride 2
Dense block (4)	7 x 7	1 x 1 CONV x 16 3 x 3 CONV	1 x 1 CONV x 16 3 x 3 CONV
Classification layer	1 x 1	7 x7 GLOBAL AVG POOL 1000D SOFTMAX	7 x7 GLOBAL AVG POOL 14D SIGMOID

There are four dense blocks in DenseNet; each dense block consists of at least six consecutives of a 1 x 1 convolution layer, followed by a 3 x 3 convolution layer. The numbers of these consecutive 1 x 1 and 3 x 3 layers in dense blocks depend on the types of DenseNet which are either 121,169,201 or 264 layered DenseNet. However, all DenseNet configurations have four dense blocks, and the differences

are only in the number of consecutive convolution layers within a dense block. The concatenated features from a dense block in DenseNet are then downsampled through a transition layer.

The transition layer consists of a 1×1 convolutional layer and a 2×2 average-pool layer in stride 2 mode. A dense block in DenseNet is followed by a transition layer consecutively. ChexNet by Rajpurkar et al. [136] initiates the popularity of DenseNet-121 as the backbone network to perform the chest X-ray classification. ChexNet [136] consists of the sigmoid functions in the last layer. ChexNet changes the output dimension of the final classification layer of DenseNet-121 from 1,024 dimension of softmax output into 14 dimensions of sigmoid functions. The changes from 1,024 to 14 nodes reflects the number of classification labels in the chest X-ray dataset [142]. Table 2.4 depicts the layer differences between ChexNet [136] and DenseNet [80].

2.2.5 EfficientNet

Table 2.5: **The EfficientNet-B0 Layer [161]**

Stage i	Operator \mathcal{F}_i	Resolution $\mathcal{H}_i \times \mathcal{W}_i$	Channels \mathcal{C}_i	Layers \mathcal{L}_i
1	Conv 3x3	224x224	32	1
2	MBCConv1, k3x3	112x112	16	1
3	MBCConv6, k3x3	112x112	24	2
4	MBCConv6, k5x5	56x56	40	2
5	MBCConv6, k3x3	28X28	80	3
6	MBCConv6, k5x5	28X28	112	3
7	MBCConv6, k5x5	14x14	192	4
8	MBCConv6, k3x3	7x7	320	1
9	Conv 1x1 & Pooling & FC	7x7	1280	1

The recent work from Tan et al. [161] introduced EfficientNet. It proposed a formulation to perform grid search among three prominent aspects of the deep-network's architecture: depth, width and input resolution. The depth defines the number of layers, the width defines the number of nodes for each layer and the input resolution defines the size of the input images. The compound scaling from those

three components is then composed into different architectures from EfficientNet-B0 into EfficientNet-B7. The networks use the mobile inverted bottleneck layers, similar to [143, 160]. The layers are then concatenated to a squeeze-excitation layer [78]. The ReLu6 function is capped at the magnitude of 6; it was used in MobileNetV2 [143]. However, EfficientNet replaces the use of ReLu6 with Swish. Equation 2.1 shows the difference among the ordinary ReLu function, the ReLu6 [97] and the Swish activation function:

$$\begin{aligned} ReLu(x) &= \max(0, x) \\ ReLu6(x) &= \min(\max(0, x), 6) \\ Swish(x) &= x ReLu(x) \end{aligned} \tag{2.1}$$

The layers of EfficientNet-B0 are depicted in Table 2.5. The further scaling of EfficientNets B0 into B7 are then defined by the grid-search formula, as reported in [161]. After the input layer, EfficientNet uses a 3 x 3 spatial convolutional layer in stride 2 mode; then, it uses MBConv1, the linear bottleneck and inverted residual layer [143]. After the MBconv1 layer, the network has six consecutive MBConv6 layers with various 3 x 3 and 5 x 5 kernels, as listed in Table 2.5. Each MBConv6 has three consecutive layers consisting of a 1 x 1 convolutional layer, a 3 x 3 or 5 x 5 depth-wise convolutional layer and another 1 x 1 convolutional layer. Each MBConv1 has two consecutive layers consisting of a 3 x 3 depth-wise convolutional layer and another 1 x 1 convolutional layer. The final layer consists of 1 x 1 convolutional, the global average pooling and 1,280 nodes of a fully connected layer. Following the previous modification of DenseNet-121 into the specific implementation of the chest X-ray [173] classification problem, we also modify the final output layer from 1,280 nodes into 14 nodes.

2.2.6 The Conclusion: Traditional Classifiers Versus Neural Networks

The traditional classifiers, such as an SVM, a decision tree or a logistic regression, require feature engineering to perform classification. The better features chosen during feature engineering will produce more accurate classification performance.

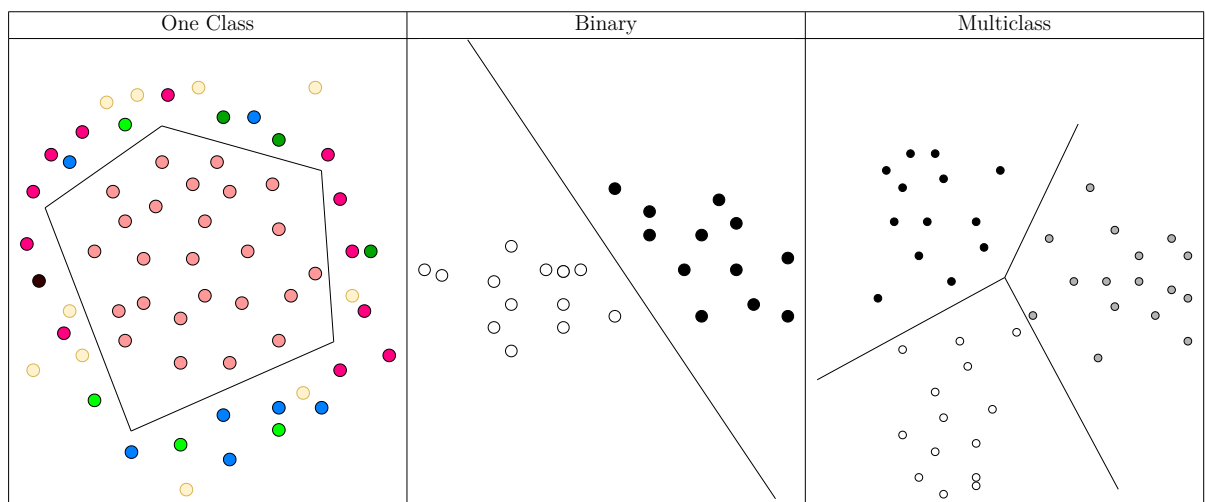
However, it is also the primary disadvantage of traditional classifiers; the incorrect features will not satisfy classification performances. Further, the trials to select the appropriate features may be time-consuming. Despite the necessity of feature engineering, a neural network has the advantage of performing end-to-end training to output a final classification's predictions. The removal of feature engineering in the neural network will reduce the risk of incorrect features for classification. The feature engineering in the traditional classifier is complex; the resulting features do not always deliver good classification performance.

Conversely, the traditional classifier, such as a decision tree, has better explainability. The classification process of a decision-tree classifier is more interpretable than that of the neural network type. Also, the decision tree supports feature importance, which sorts the rank of features being used during classification.

2.3 The Binary, Multiclass and One-Class Classification Overview

Speaking of the characteristics of a classifier, the decision boundary to separate the distinct training patterns is illustrated in Table 2.6. This study refers to the term “training-patterns” as the patterns present in the training process. The re-

Table 2.6: **Decision Boundary**



search presented in this thesis defines the task of classification in terms of quantity

from the training patterns into four categories: binary classification, multiclass classification, one-class classification and multilabel classification. The multilabel problem exists when there is a training example that belongs to more than one training pattern [109]. However, the multilabel classification, in most cases, converts into binary classification or multiclass classification problem. Therefore, in Section 2.3, this study discusses only three prominent types of classification.

2.3.1 Binary Classification

Binary classification occurs when there are two training patterns present during training. The binary classification objective is to assign a training example to one of two categories [47].

Some popular algorithms suitable for performing binary classification include SVM, logistic regression, multilayer perceptron/neural network, decision tree and Naïve Bayes. Prominent advantages of a binary classification task are:

1. The binary classification uses fewer labels than multiclass classification. In the case of an equal number of examples between multiclass and binary classification tasks; the binary classification task requires fewer computing resources because of this reason [124].
2. Binary classification produces a more complex decision boundary than one-class classification tasks [90].
3. Several binary classifications can be extended into a multiclass classification to solve more complex problems [156].
4. Binary classifier is suitable for both linear and non-linear classification problems.

The primary disadvantage of the binary classification is when the data suffers from massive imbalance to one of the labels; then, the classifier tends to result in non-representative accuracy outputs [40] because there was only a small fraction of samples from the minority pattern to learn.

2.3.2 Multiclass Classification

Multiclass classification occurs when more than two training patterns are available for the training process. Khan et al. (2014) [90] emphasised a constraint that in the multiclass classification, the decision boundary is supported by the presence of training examples from each class.

Sugiyama (2016) [156] proposed that solutions to address the multiclass classification can be in the form of: (i) the decomposition of the multiclass classification problem into several binary classifications; and (ii) the direct method to approach the multiclass problem (e.g.: support vector extension for multiclass problem; the typical procedure is to use the kernel trick to provide the non-linear decision boundary, which can capture multiclass patterns) [156]. Other works [115,134] use the tree-based algorithm as a direct approach to address multiclass classification.

The decomposition of a multiclass problem into several binary classifications is also called “binarization”. “A class binarization is a mapping of a multiclass problem on several two-class problems that allows a derivation of a prediction for the multiclass problem from the predictions of the two-class classifiers. The two-class classifier is usually referred to as base learner” [60]. Generally, there are two types of binarisation scheme. The first is one-versus-rest or one-versus-all, abbreviated as OvR or OvA. Rajpurkar et al. (2017) [136] classified a single class of pneumonia among 14 classes of lung cancer with the OvR strategy. The second is the one-versus-one (OvO) strategy.

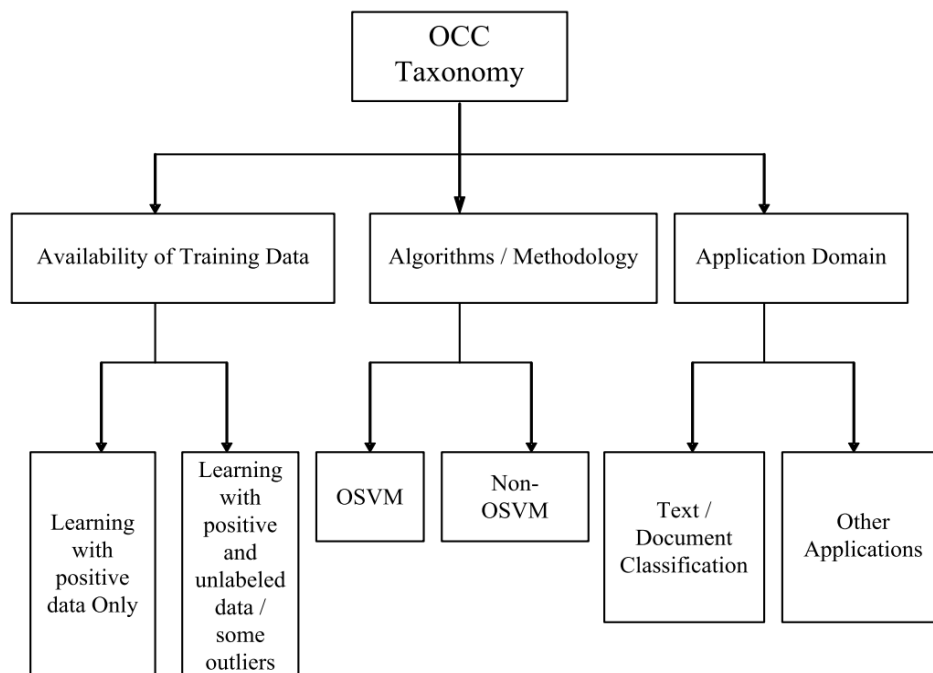
Suppose we have N different training patterns, the binarisation for N multiclass problem will result only in N binary classification for OvR/OvA schemes and $N \cdot (N - 1)/2$ binary classification for the OvO scheme.

In terms of comparison between the binarisation and direct method to address the multiclass classification problem, this study quotes the opinion of Sugiyama (2016) [156] “However, the direct method does not necessarily perform better than the reduction approaches, because multiclass classification problems are usually more complicated than binary classification problems. In practice, the best approach should be selected depending on the target”.

2.3.3 One-Class Classification

The application of one-class classification (OCC) is useful when (i) there is an imbalance in the dataset [58] or (ii) there is a case when the end user “may only be interested in a specific class without considering other” [100]. In the context of medical images classification, a dataset can have multiple labels, but the subject of interest for the classification might be in the specific patterns. The literature [87, 90] suggests that OCC occurs when there is only one training pattern. Other patterns are either absent or available in limited amounts (hence, the other patterns are presented as outliers). Khan et al. (2014) [90] emphasises a constraint: “only one side of the classification boundary can be determined using only positive data (or some negatives)”. Figure 2.5 depicts the taxonomy of the OCC from Khan et al. (2014) [90].

Figure 2.5: The One Class Classification (OCC) Taxonomy [90]



The concept of OCC is principally different from binary classification, multi-class classification. Only one class is present during training [52, 133]. Generally,

there are two procedures to train a one-class classifier [45,90]. The first is to train with only the presence of the regular pattern. The second is with the normal and the other patterns as the outliers. These training procedures then create a final classifier model, which only recognises the present pattern during the training as the standard class, and the other patterns as the outliers.

Another study [52] described three approaches to train an OCC. The first is to train using the majority pattern; the second is to “fine-tune” a pre-trained OCC into the minority pattern. The third is to train the OCCs in both majority and minority and combine the outputs.

Yu et al. (2003) [183] proposed an SVM optimisation algorithm employing a single class support vector mapping convergence (SVMC), using only minimum training examples to maintain the decision boundary and take less training time . The literature [183] reports that SVMC produces better accuracy decision boundaries than regular SVM with fully labelled data and the one-class SVM [106,107] (OSVM). However, the latter OCC work of Yu et al. (2005) [184]—the mapping converge algorithm—provides slightly better accuracy decision boundary than does the SVMC [183].

Several works [67,68,163] have proposed approaches to address the multiclass classification problem with the use of one-class learners. Ban and Abe (2006) [163] proposed the ensemble approach from several OCCs with the adaptation from previous works: support vector domain description (SVDD) [164] and kernel whitening-kernel principal component analysis (KW-KPCA) [165]. The proposed method [163] shows faster training and better generalisation under the appropriate given parameters.

Hadjadji et al. (2014,2017) [67,68] proposed “a dynamic weighted average rule to measure the importance of the used classifiers”. The work [67,68] trains separately several distinct classifiers in the same set. There are three classifiers in the early work of Hadjadji et al. (2014) [67]: one-class nearest neighbour, one-class SVM and auto associative neural network. Later, Hadjadji et al. (2017) [68] used five classifiers with two additional classifiers: one-class K-center and one-class principal component analysis (PCA). The proposed approach [67,68] ensures the best classification performance is chosen among those three classifiers.

Gao et al. (2020) [59] propose an Image Complexity based on One-Class Classification (ICOCC). The work [59] uses perturbation to augment four imbalanced datasets. Gao et al. argue that the perturbation can improve the one-class classification performance. Despite the perturbation behaviour, which is quite similar to ordinary augmentation. Gao et al. [59] note that perturbation and augmentation are pretty different. Perturbation aims to build a multiclass classification task from single-class examples. In contrast, augmentation aims to generalise the classification performance (e.g. reduce the overfitting).

The proposed work [59] evaluates four datasets, namely: The Breast Magnetic Resonance Imaging (MRI) dataset, The Breast Full-Field Digital Mammography (FFDM), The Space-occupying Kidney Lesion (SOKL) and HEP-2 Cell Image dataset. The other competing methods for benchmarks are One-Class SVM (OCSVM), Convolutional Autoencoder OCSVM (COCSVM), Deep Structured Energy-Based Model (DSEBM), Deep Autoencoding Gaussian Mixture Model (DAGMM). The evaluation metrics are AU-ROC and AU-PRC; for AU-PRC, the calculation is performed in two ways. First, the samples for a particular class are considered positive. Second, the samples for all-other classes are considered positive. Results from ICOCC [59] provide improvements over other methods on all the evaluation datasets.

The research presented in this thesis intends to improve the multiclass medical images classification problem with the help of the one-class neural network approach. The approach of Ruff et al. (2018) [141] and Perera and Patel, 2018 [133] proposed OCC using a neural network, but their works are not in medical case problems.

This research trains the network with only positive examples. The reasons for choosing this approach for this study chose are listed below:

1. A one-class classifier with a single pattern training procedure represents a better definition than the one also trained with outlier patterns. Despite the number of outliers examples, the outliers' patterns render the training procedure more similar to binary or multiclass classifiers.
2. This study aims to combine the strength of compactness and descriptiveness, which is similar to [133]. These two concepts represent different perspectives.

Compactness exists when the decision boundary converges to a single pattern. Conversely, descriptiveness exists in the binary or multiclass settings because of the discriminative behaviour among different patterns. A primary distinction of our work with [133] is that this study proposes a custom network architecture.

2.4 The Computational Cost of the Use of Features for Classification Task

Features-based methods refer to the use of salient features from a dataset, rather than using the dataset source in the original forms (e.g., images and text). The main objective in performing features-based classification is mainly to reduce the computational cost. This type of study is essential when the application domain consumes significant computational environment (e.g. large input size), but only limited resources are available. In terms of the importance of low-cost neural network architecture, previous works from Howard et al. [76] and Sandler et al. [143] proposed low FLOPs (floating point operations) architecture compatible for mobile devices.

The advantage of [76, 143] is that those works do not perform the feature-extraction process; they use the end-to-end training model. The key to achieving the low-cost architectures is the grid search of the width factor from the convolutional filter and the input size's grid search into the maximum of 224 x 224 pixels. Thus, this type of approach also comes with trade-offs. Reducing the input size and the width of the convolutional filter also leads to reduced classification accuracy. Hence, this particular method might not be effective when accuracy is critical, such as in medical image classification tasks. A significant advantage of feature extraction is its capability to use the feature vectors from large size input without sacrificing accuracy. The feature extraction will gain a better trade-off in terms of computational resources versus classification accuracy, compared with reducing the input size.

Sarkar et al. [144] extracted features with AlexNet [98] from 640 x 360 reduced pixels, which were initially 1,280 x 720 in size. The work [144] then used the

extracted features to run the traditional SVM algorithm for face detection in several mobile telephone platforms, such as Samsung Galaxy, HTC One and Google Nexus. This study can have different perspectives regarding the computational cost from the classifier model: either from the deployment or production perspectives. For example, Sarkar et al. [122] was more concerned about the aspect of the final classification model deployment into the mobile devices, while Nanni et al. [122] emphasised the requirements of substantial computational resources during the classifier’s training. Nanni et al. [122] combined the feature extraction from the neural network with the handcrafted features to increase accuracy using low computational resources required during the training process.

Understanding the computational-cost perspective is critical and affects the calculation of the total final required resources. It also clarifies a separation between training and testing costs. For example, Howard et al. [76] and Sandler et al. [143] did not consider the architectural-search process in the total sum of required computational resources. Further, Sarkar et al. [144] did not count the backbone network’s training process in the sum of required computational resources for the final model. This study can conclude that these works [76, 143, 144] care more about the deployment computational-cost perspective than the production perspective.

In general, there are three ways to obtain dataset features. The first is to perform feature selection; this method is performed by handcrafting the selected features from the dataset. The easy examples use feature importance from a decision-tree algorithm [89] or a PCA algorithm [151]. The second is to perform traditional feature extraction. The method extracts features from the dataset to acquire the desired features; some of the famous methods are scale-invariant feature transform (SIFT) [121], speeded-up robust features (SURF) [28] and histogram of oriented gradients (HOG) [88]. The third is to perform a deep-network-based feature extraction; the method extracts the features with a neural network’s help. This method is sometimes also mentioned as “deep features”. The first and second methods are more manual with the supervised human intervention process. Conversely, the third is more automatic with the use of pre-trained weights from the neural network.

However, a combined method is widely known as “feature fusion”. This method combines the approaches above. For example, feature fusion is performed by concatenating the features from two processes: the feature selection from the random forest algorithm and the feature extraction from a neural network [63]. Another feature fusion type joins the feature extraction from a neural network and the feature extraction from traditional methods [122].

Another advantage of using a deep-network-based approach is that it requires less expertise to comprehend the application domain [148] than does the conventional features of handcrafting methods. Also, handcrafted features are more problem-specific and not guaranteed to solve other problems because the features were designed by human experts [148].

The neural network backbone to perform feature extraction can be supervised (e.g. convolutional network) or unsupervised (e.g. autoencoder). An example of the applicability of feature-extraction work for the imbalanced classification problem is in [84], in which Jiang et al. (2019) applied the feature extraction from an autoencoder. The work [84] improves the classification problem that a medical dataset suffers from—the intrinsic imbalance classes. The same dataset was used later in other work [99], but a different backbone network was applied to extract the features. Li et al. (2017) [99] barely used the convolutional network for the extraction. In the term comparison of accuracy between “the deep features” and the use of handcrafted features, a deep-network classifier performs better than a traditional classifier [84].

The use of the unsupervised network for feature extraction, such as in [84], extends the requirements of “deep features” to a more minimum domain expert. The statement is closely related to the autoencoder network’s capability to perform the features extraction without the labels from the domain experts. Unlike the supervised network, an autoencoder minimises the reconstruction loss (error) during training rather than minimising the errors from the predicted label.

The research presented in this thesis’s approach is to use “deep features” extracted from supervised network EfficientNet-B3. This is because the original images in the NIH dataset have large size (1,024 x 1,024) and the use of “deep features” significantly reduces the computational resources. Further, this study maintains accuracy through the Bayesian optimisation process.

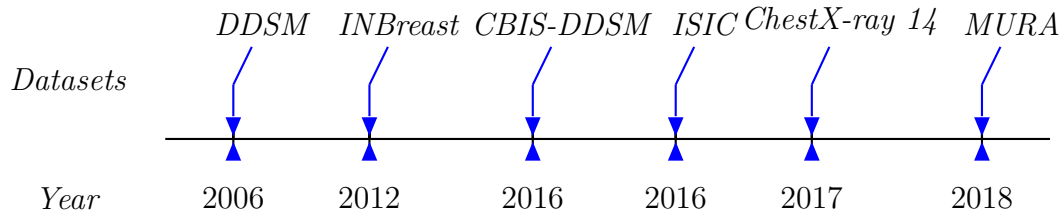


Figure 2.6: The Datasets' Timeline

2.5 The Datasets of Medical Images

A primary reason to nominate candidate datasets listed in Table 2.7 is because these datasets have labels from experts in their particular domains. Hence, the datasets are suitable for the supervised tasks, such as deep-learning methods. The DDSM dataset contains 2,620 patients' cases in high resolution and each case has more than one image. However, Ribli et al. (2017) [140] reported that the dataset is not fit for evaluation purposes; hence, they only used the DDSM dataset for training and used INBreast dataset [119] for testing. This research obtained the INBreast dataset [119] by requesting privately, and the original owner obliged us to cite them in our publications. This dataset has 410 images from 115 cases. CBIS-DDSM is an improvement subset from the DDSM dataset, which has been curated for better CAD purposes.

The international skin imaging collaboration (ISIC) dataset uses photographic images to reduce skin cancer mortality, especially melanoma. The use of images as a skin cancer detection tool aims to reduce the need for biopsies (test tissue taken under a microscope). The ISIC dataset has roughly 23,000 cancer images. NIH Chest-X-Ray 14 has 112,120 frontal views for 14 classes. As mentioned previously, the Chest-X-Ray 14 has an imbalance class problem. A large dataset musculoskeletal radiographs (MURA) of bone X-rays has 40,561 abnormality images. MURA is suitable for binary classification task.

Table 2.7: **The Datasets**

Title	How To Obtain
Database for Screening Mammography (DDSM)	http://marathon.csee.usf.edu/Mammography/Database.html
INbreast: Toward a Full-field Digital Mammographic Database [119]	By request to INBreast dataset owner
International skin imaging collaboration (ISIC) dataset for skin cancer detection	https://www.isic-archive.com
An updated and standardised version dataset from Curated Breast Imaging Subset of Database for Screening Mammography (CBIS-DDSM)	https://wiki.cancerimagingarchive.net/display/Public/CBIS-DDSM
The ChestX-ray14 dataset	https://nihcc.app.box.com/v/ChestXray-NIHCC
A large dataset MURA of bone X-rays	https://stanfordmlgroup.github.io/competitions/mura/
The Glaucoma dataset	The dataset was previously used in a scholarly work [91]

2.6 Discussion

Deep learning has become the most prominent algorithm in medical image analysis. However, its applicability in the real world faces several challenges. These include the requirement of high-computing resources and sufficient labelled data. According to the literature, there are several approaches to tackle the problems. In most cases, the best solutions for each case are specific, and no method works best for all problems.

Chapter 3

Imbalance Classification: The Aggregate Method for Chest Cancer Classification

3.1 Introduction

In this work, a novel focal-loss function is proposed to address the imbalance of positive–negative patterns and tackle the classification correctness in both positive and negative samples when training the neural networks. The performance of the proposed focal-loss function is evaluated by performing chest X-ray classification, which relates to the imbalance data [173].

This research also proposes the use of EfficientNet [161] with progressive image resizing under two-phase training in complement with the proposed loss function. The motivation to use EfficientNet is to inspect the outcome of the proposed loss function into different scaling architecture. The aggregate of the proposed loss function and the two-phase EfficientNet training achieved 2.10% improvement, which is measured with area under receiver-operating-characteristic curve (AU-ROC). Also heatmap visualisation shows that better coverage of disease can be achieved by the proposed aggregate approach compared with the baseline [136].

To achieve fair benchmarks, this research reports several results from various split-set options for the performance evaluation. This research performs three split-set experiments, which aim to provide better evaluation and comprehensive analysis ; the first is by the use of “official” splits from [157]. The second is

under five-folds cross-validation configuration, which was also used in the work of Baltruschat et al. [26]. The last one is via use of identical splits from the public Github page [139, 174].

This research achieves state-of-the-art results for the classification problem of the chest X-ray dataset [173], measured under these three split-set configurations. The dataset [173] for base metrics includes the same training, validation and test splitting set from [129]. This research refers to the split set [157] as “official split”. [157] and has two groundtruth files as labels. They are train_val_list.txt, which consists of 86,524 samples, and test_list.txt, which consists of 25,596 samples. Baltruschat et al. [26] emphasised that different splitting of datasets [173] has a significant impact on classification performance. Since the splitting of training and test data is the same, the benchmark is fair. Figure 2.2 and Table 2.1 show that the class distribution is imbalanced since the positive and negative samples are very different.

This research contributes to the improvement of medical image classification problem and addresses the imbalance problem within the chest X-ray dataset. Also, this research proposes the advancement of the use of state-of-the-art neural net architecture for the final classification performance—The EfficientNet—with two-stage training. The progression of Chapter 3 is the improvement of classification performance over previous studies [26, 64, 136, 173].

3.2 Method

3.2.1 The Existing Weights Function and Network Architecture

Wang et al. [173] and Gündel et al. [64] defined the weights, ω_{k+} and ω_{k-} , of the positive and negative samples for the k -th pattern.

$$\begin{aligned}\omega_{k+} &= \frac{P_k + N_k}{P_k} \\ \omega_{k-} &= \frac{P_k + N_k}{N_k}\end{aligned}\tag{3.1}$$

where P_k and N_k are the numbers of positive and negative samples for the k -th pattern. However, Cui et al. [44] used both ω_{k+} and ω_{k-} equally to develop the

loss function. Lin et al. [101] proposed the focal-loss function:

$$L_{foc}(p) = -\alpha(1-p)^\gamma \log(p). \quad (3.2)$$

p is the prediction.

In Equation 3.2, parameter α attempts to balance the positive–negative samples, while γ is adjusted to release the easy samples and dominate the hard samples; the easy and hard samples are those classified correctly and incorrectly, respectively. Generally, $\gamma \geq 0$; when $\gamma = 0$ focal loss is the same as an ordinary cross-entropy loss [101]. The experimental results showed that easy samples are down-weighted when $\gamma \approx 1$; The samples are further down-weighted when $\gamma > 1$. Determination of α is discussed to demonstrate the impact to the focal-loss function (see Equation 3.2). The parameters chosen as below [44]:

$$\begin{aligned} \beta &= \frac{(N-1)}{N} \\ \alpha_k(\beta) &= \frac{1-\beta}{1-\beta^{n_k}} \\ N(\beta) &= \sum \alpha_k(\beta) \end{aligned} \quad (3.3)$$

where n_k is the number of the k_{th} pattern, and N is the number of samples.

Conceptually, β is used to adjust the significance of the number of samples. $N(\beta)$ is the sum of all α_k -s, which corresponds to the β value for each k -pattern. $N(\beta)$ is used for normalisation with the number of patterns. However, Cui et al. [44] ignored the negative pattern in the weight calculations, which is very common in the medical image classification problem.

3.2.2 Proposed Weights Function and Network Architecture

The normalisation of α_k formulated in Equation 3.4 is used to weight the k^{th} pattern:

$$\tilde{\alpha}_k(\beta) = \frac{C}{N(\beta)} \cdot \alpha_k(\beta) \quad (3.4)$$

where C is the number of class. Although Cui et al. [44] proposed the grid search to determine β based on their formulation, the separable weights of a positive and negative patterns have not been addressed .

This research integrates the separability of positive and negative patterns into the loss-function to improve the classification capability of Cui et al.'s approach. The hypotheses address the importance of both positive and negative pattern weights to improve the end-to-end training.

$$\omega_{k+} = \widetilde{\alpha}_k(\beta) \quad (3.5)$$

where ω_{k+} are the weights for positive samples of the k^{th} pattern. Equation 3.5 is an elaboration point between [44] and the proposed method.

This research deliberately assigns α_k to each sample in k^{th} pattern on the basis of the specified ω_{k+} weights. [44] emphasised the importance of effective samples to define the weights and this research has two types of weights ω_{k+} and ω_{k-} . In the proposed approach, $\widetilde{\alpha}_k(\beta)$ from [44] attempts to determine the weights of only the positively labelled samples, which is given in Equation 3.5. Also, this research determines the weight of the negative patterns:

$$\omega_{k-} = 1 - \omega_{k+} \quad (3.6)$$

Experimental results evaluate the performance of the proposed weights in Equations 3.5 and 3.6 to balance the imbalanced samples.

In the proposed method, the five hyperparameters β are given in Equation 3.7.

$$\begin{aligned} \beta_1 &= 1 - 2.0 \cdot 10^{-6}; \beta_2 = 1 - 2.0 \cdot 10^{-5}; \beta_3 = 1 - 2.0 \cdot 10^{-4}; \\ \beta_4 &= 1 - 7.0 \cdot 10^{-4}; \beta_5 = 1 - 2.0 \cdot 10^{-3} \end{aligned} \quad (3.7)$$

where β_2 is determined by Equation 3.3. The other β -s are determined by the grid-search. With the exception of the β_4 , the grid search was performed by changing the β value with standard deviation of 10 from β_2 . The current value of β_4 was chosen because that magnitude is the median between β_3 and β_5 . Also the results obtained by the proposed method is compared with those obtained by the other six methods, Wang et al. [173], Yao et al. [180], baseline ChexNet [136], weighted binary cross-entropy loss, Balturschat et al. [26] and Gündel et al. [64]. The comparison is depicted in Tables 3.5.A and 3.5.B.

3.2.2.1 The Weighted Cross-Entropy

The formulation for cross-entropy loss [190] with the proposed weight is:

$$L_{bce}(p) = \sum_{k=1}^C \omega_k (-y_{true}^k \log(p))$$

$$\omega_k = \begin{cases} \omega_{k-} & \text{if } y_{true}^k = 0 \\ \omega_{k+} & \text{if } y_{true}^k = 1 \end{cases} \quad (3.8)$$

where y_{true}^k are the groundtruth labels for each sample in pattern k . To perform the experiments in Section 3.4.2, this research sets the $\omega_{k-} = \omega_{k+}$ for a particular case, the case where this research wants to see the outcome from Cui et al.'s [44] formulation adjusted into the dataset [173] classification problem. The cross-entropy loss uses softmax output by default, whereas the binary cross-entropy loss uses sigmoid output.

3.2.2.2 The Weighted Focal Loss

The formulation for focal loss with the proposed weight is:

$$L_{foc}(p) = \sum_{k=1}^C \omega_k (-\alpha (1 - p)^\gamma y_{true}^k \log(p))$$

$$\omega_k = \begin{cases} \omega_{k-} & \text{if } y_{true}^k = 0 \\ \omega_{k+} & \text{if } y_{true}^k = 1 \end{cases} \quad (3.9)$$

The proposed focal loss attempts to weight both the easy–hard samples and the positive–negative patterns, which are not addressed in Cui et al.'s approach [44]. The proposed focal loss also suits the multiclass classification problem [120, 123]. There is no existing focal-loss method that addresses both effective number of samples and positive–negative patterns weighting.

3.2.2.3 Progressive Image Resizing

Progressive image resizing is the procedure to train a single deep-network architecture with incremental input sizes in multiple stages of training. The first stage trains the network with the default image size for the network, followed by the next stage that utilises the bigger size images and the best performance of the pre-trained model from the previous stage. There is no formal definition of the

exact number of steps, but the classification performance will improve to some extent and then become saturated. Then, gain diminishes; this is highly specific to classification problems. This research reports that the third stage of training with progressive image resizing did not improve the performance of the existing chest X-ray classification problem.

Another functionality from the progressive image resizing is to provide another form of augmentation. It (re)trains the model with the augmentations of different sized inputs. Several works [34, 132, 149, 175] mention that augmentation is a proven method to reduce overfitting. This research required the final model to be risk-free from overfitting; two-stage training is the approach to ensure this. In summary, this research performed two-stage training to achieve two aims: to improve classification accuracy and prevent overfitting.

3.2.2.4 The Network Backbone

This research used DenseNet 121 [80] and EfficientNet [161] for the experiments. However, the results in the Tables 3.2, 3.4, 3.5.A and 3.5.B suggest that EfficientNet [161] is a better network to improve classification performances than is DenseNet 121 [80].

3.2.2.5 Baseline

This research reproduces ChexNet [136] based on [42]. The experiments performed by the proposed method and the other methods [26, 64, 173] are based on the training and test split in [157] and are reported in Table 3.5.A and e 3.5.B. However, Rajpurkar et al. [136] never shared the split set with the public. The use of official split [157] resulted in lower performance than reported in Rajpurkar et al. [136]. This research used the ADAM optimiser as in [136] to develop the neural network, of which optimisation converged at epoch 11. Other research also used ADAM [26, 64] and stochastic gradient descent [173].

3.2.2.6 Performance Evaluation

Suppose this research needs a better perspective of algorithm performance; it should apply different metrics to evaluate the results. This research applied the

area under precision-recall curve (AU-PRC) metric for further evaluation; the metric has a different characteristic than AUROC. In terms of baseline, AUROC has a fixed baseline of 0.50 for random classifiers and 1 for the perfect classifier, respectively [43, 69].

In contrast, the AU-PRC baseline is dynamic since it heavily depends on the ratio between positive and negative samples [142]. AU-PRC is more sensitive to data distribution. AU-PRC will have the baseline (0.50) for a random classifier under the equal number of positive and negative samples. When the number of negative samples is 10 times that of positive samples, this baseline will decrease to a smaller number (0.09) [142]. The formulation to calculate the baseline of AU-PRC shown in Equation 3.10 is from the literature [142].

$$\text{baselineAUPRC} = \frac{\text{positives}}{\text{positives} + \text{negatives}} \quad (3.10)$$

Suppose there are two classes with an identical value of AU-PRC (0.50); the interpretation from this result will vary for both classes. The 0.50 AU-PRC is a good result for the class with low positive samples, but it may not be satisfactory for the class with a remarkable number of positive samples.

3.3 Research Contribution and Novelty Statement

The contribution is to propose an approach that can combine a weights calculation algorithm for deep network and the optimisation of training strategy from the state-of-the-art architecture.

3.4 Experiments and Results

3.4.1 Backbone Network Training

Since this experiment uses DenseNet 121 [80] as the primary backbone network, the availability of pre-trained ImageNet can be used for the classification. This research used the pre-trained weights from ImageNet to develop the network. This

research used a single Titan V with 12 Gb GPU memory to develop the network; 24 hours with 25 epochs of training were required. This research also trained the Densenet-121 in a two-phase training cycle and performed progressive image resizing for comparison in Table 3.5.B. Because this research aims to improve overall classification performance, the experiments also modified the architecture of backbone network from DenseNet-121 into EfficientNet [161].

The approach is mainly to expand the performances from the proposed cost-sensitive loss function into better architecture. This research was limited only to the use of the EfficientNet-B0 and EfficientNet-B3 networks for experiments. This is because the use of EfficientNet-B3 in combination with the progressive image resizing method (as discussed in Section 3.7), further input beyond 600 x 600 was not practical. Consecutive EfficientNets training requires extensive computations because of the scaling of the image sizes, the depth and width of the network. Conversely, the approach of progressive image resizing only considers the aspect of image sizes into computational resources; it ignores the depth and width of the network.

To train the EfficientNets, the experiments used the Tesla v100 with 32 Gb of GPU memory. For each network, this research performed the two-phase training procedure with progressive image resizing, as previously discussed. In the first phase, the experiment trained the network with the pre-trained model from ImageNet. In the second phase, the experiments trained the network with the best performing model from the first phase.

The important fine-tune was the size of the image input. In the first phase, this research used the default input size from the network. Then, it doubled the input size in the second phase. This was implemented with size of 224 x 224 in the first stage of EfficientNet-B0 and 448 x 448 in the second stage (EfficientNetB0). Also, it used 300 x 300 in the first stage of EfficientNet-B3 and 600 x 600 in the second stage (EfficientNet-B3). The experiment reduced the batch size to half, from 32 in the first phase to 16 in the second phase. The reduced batch size is mainly to ensure the batched images for each step on each epoch will fit into the GPU's memory boundary.

Their default configuration determines the first phase of the EfficientNets' input sizes. In this case, 224 x 224 is the default input size for EfficientNet-B0 and

200 x 300 is the default input size for EfficientNet-B3. This research assumes that those sizes are the best configuration for each EfficientNet network since the EfficientNet’s creators [161] chose those sizes. The input sizes for the second phase were also assumed that if this research doubled the input size, it would still suit the network quite nicely. The two-phase training with progressive image resizing improved ($\pm 1\%$) the classification outputs between each model’s first and second phases.

3.4.2 Weighted Binary Cross-Entropy with Effective Number of Samples

This experiment is an adoption of Cui et al.’s [44] method for the chest X-ray dataset [173] classification problem. In Cui et al.’s approach [44], the balanced weights between positive and negative was not used; the weights were computed using the effective number of samples. Cui et al. [44] used Equation 3.3 to compute the weights. The research performed this experiment to provide evidence of performances derived from [44] versus that of the proposed approach. This experiment used binary cross-entropy as a loss function and combined the weighting into the loss function. This research required improved comparison because the work [136] of Rajpurkar et al. used binary cross-entropy loss but ignored the importance of the imbalance problem.

This research sets the $\omega_{k-} = \omega_{k+}$ for the implementation of Equation 3.8 for this case, since [44] ignored the balanced positives–negatives. The best performance classification for the model was also achieved on epoch 11, similar to the Section 3.2.2.5 baseline. Comparison results with the other experiments are shown in Tables 3.5.A and 3.5.B. This method performed only slightly better than the baseline, with the 79.24% area under receiver-operating-characteristic (ROC) curve.

3.4.3 Weighted Focal Loss with Positive and Negative Pattern

This experiment uses the loss function [101], which is integrated with the focal loss and proposed weighting. This experiment selected the value of α value on the basis

of [101], which is between [.25,.75]. This experiment discovered that $\alpha = 0.5$ and $\gamma = 1$ is the best of focal-loss hyperparameters for the proposed method. Table 3.1 depicts the γ hyperparameter-tuning process.

Table 3.1: Results from Various γ for Focal Loss

	weighted focal loss $\beta = 0.9998$		
	$\alpha = 0.5$ $\gamma = 1$	$\alpha = 0.5$ $\gamma = 2$	$\alpha = 0.5$ $\gamma = 4$
Atelectasis	0.7777	0.7784	0.7755
Cardiomegaly	0.8925	0.8911	0.8912
Effusion	0.8322	0.8288	0.8318
Infiltration	0.7098	0.7064	0.6989
Mass	0.8262	0.8294	0.8235
Nodule	0.7626	0.7662	0.7551
Pneumonia	0.7311	0.7276	0.7147
Pneumothorax	0.8665	0.8661	0.8624
Consolidation	0.7563	0.7502	0.7526
Edema	0.8460	0.8427	0.8489
Emphysema	0.9211	0.9251	0.9182
Fibrosis	0.8296	0.8295	0.8189
Pleural thickening	0.7783	0.7805	0.7792
Hernia	0.8977	0.9131	0.9292
Average	0.8175	0.8168	0.8143

This research used the rectified ADAM and look-ahead (RANGER) optimiser, which requires a smaller number of training epochs to converge. The optimiser converged at epoch 5. The experiment deliberately assigned the two-stage training to prevent overfitting and to improve performance. This method achieved 82.32% area under ROC curve with two-phase DenseNet-121 and 83.13% with two-phase EfficientNet-B3 under the official split setting. The comparison of the official split setting results with other experiments is shown in Tables 3.5.A and 3.5.B.

The training time took 71 minutes for one epoch in the first phase, with 32 batch size and 180 minutes for one epoch in the second phase, with 16 batch sizes.

The test took 15 minutes with eight batch sizes for the first phase and 27 minutes with eight batch sizes in the second phase.

Table 3.2: **Identical Split Comparison** [139]

Pathology	Third party [139] of Guan et al. [62]	Weighted focal loss $\beta =$ 0.9998 EfficientNet-B3	
		Phase 1	Phase 2
Cardiomegaly	0.9097	0.9137	0.9144
Emphysema	0.8905	0.9471	0.9558
Edema	0.9185	0.9021	0.9071
Hernia	0.9064	0.9357	0.9409
Pneumothorax	0.8794	0.9003	0.9092
Effusion	0.8843	0.8899	0.8923
Mass	0.8707	0.8596	0.8669
Fibrosis	0.8208	0.8526	0.8657
Atelectasis	0.8225	0.8350	0.8397
Consolidation	0.8210	0.8124	0.8208
Pleural thicken.	0.8127	0.8041	0.8136
Nodule	0.7691	0.8043	0.8293
Pneumonia	0.7614	0.7721	0.7703
Infiltration	0.7006	0.7297	0.7363
Average	0.8405	0.8542	0.8616

* This research found the third-party re-implementation [139] reported lower performances than did [62]. Guan et al. [62] did not provide the official code and split sets. The critical classification problems for the dataset [173] is that different splits will lead to different performances [26]

Table 3.2 shows the comparison with the latest research’s outputs; Tables 3.3.A and 3.3.B show the improvement rates. The comparison of five-folds setting results are depicted in Table 3.4.

Table 3.3.A: **Improvement Rate**

Name	Hernia	Pneumonia	Fibrosis	Edema	Emphysema	Cardiomegaly	Pleural Thick.	Pneumothorax
Rate	+3.45%	+0.89%	+4.49%	-1.14%	+6.53%	+0.47%	+0.09%	+2.98%

Table 3.3.B: **Improvement Rate (cont.)**

Consolidation	Mass	Nodule	Atelectasis	Effusion	Infiltration	Average
-0.02%	-0.38%	+6.02%	+1.72%	+0.80%	+3.57%	+2.10%

Table 3.4: **Results from Five-Folds Cross-Validation**

Pathology	Baltruschat et al. [26]	Weighted focal loss $\beta =$ 0.9998 EfficientNet-B3 two-phase training
Cardiomegaly	89.8 \pm 0.8	90.6 \pm 2.4
Emphysema	89.1 \pm 1.2	94.6 \pm 1.2
Edema	88.9 \pm 0.3	90.3 \pm 0.9
Hernia	89.6 \pm 4.4	92 \pm 1.3
Pneumothorax	85.9 \pm 1.1	91.2 \pm 1.2
Effusion	87.3 \pm 0.3	88.5 \pm 0.5
Mass	83.2 \pm 0.3	86.9 \pm 1.1
Fibrosis	78.9 \pm 0.5	82.2 \pm 2.5
Atelectasis	79.1 \pm 0.4	83.3 \pm 0.7
Consolidation	80.0 \pm 0.7	80.9 \pm 0.5
Pleural Thicken.	77.1 \pm 1.3	82.7 \pm 1.3
Nodule	75.8 \pm 1.4	81.7 \pm 1.4
Pneumonia	76.7 \pm 1.5	77 \pm 1.9
Infiltration	70.0 \pm 0.7	72.8 \pm 4.5
Average	82.2 \pm 1.1	85.3 \pm 0.6

* This research performs five-fold cross-validation, mainly aiming to have the most similar setting to the benchmarked work by Baltruschat et al. [26]

Table 3.5.A: Comparison Results with Previous Research under The Official Splits

	Wang et al. [173]	Yao et al. [180]	Baseline reproduce ChexNet [136]	Weighted binary cross entropy loss	Baltruschat et al. [26]
Atelectasis	0.700	0.733	0.7541	0.7625	0.763
Cardiomegaly	0.810	0.856	0.8787	0.8812	0.875
Effusion	0.759	0.806	0.8236	0.8266	0.822
Infiltration	0.661	0.673	0.6928	0.6939	0.694
Mass	0.693	0.777	0.8053	0.8023	0.820
Nodule	0.669	0.724	0.7318	0.7383	0.747
Pneumonia	0.658	0.684	0.6980	0.7019	0.714
Pneumothorax	0.799	0.805	0.8378	0.8344	0.819
Consolidation	0.703	0.711	0.7349	0.7390	0.749
Edema	0.805	0.806	0.8345	0.8305	0.846
Emphysema	0.833	0.842	0.8666	0.8701	0.895
Fibrosis	0.786	0.743	0.7957	0.8040	0.816
Pleural thickening	0.684	0.724	0.7456	0.7502	0.763
Hernia	0.872	0.775	0.8684	0.8589	0.937
Average	0.745	0.761	0.7906	0.7924	0.806

Table 3.5.B: Comparison Results with Previous Research under The Official Splits (cont.)

	Weighted focal loss $\beta = 0.99998$ DenseNet-121	Weighted focal loss $\beta = 0.9998$ DenseNet-121	Weighted focal loss $\beta = 0.9998$ DenseNet-121 two-phase training	Weighted focal loss $\beta = 0.9998$ EfficientNet-B3 two-phase training
Gündel et al. [64]				
0.767	0.7781	0.7777	0.7820	0.7919
0.883	0.8918	0.8925	0.8845	0.8917
0.806	0.8310	0.8322	0.8380	0.8414
0.709	0.7037	0.7098	0.7022	0.7051
0.821	0.8263	0.8262	0.8329	0.8356
0.758	0.7685	0.7626	0.7863	0.8036
0.731	0.7262	0.7311	0.7338	0.7366
0.846	0.8664	0.8665	0.8706	0.8909
0.745	0.7546	0.7563	0.7537	0.7601
0.835	0.8491	0.8460	0.8534	0.8609
0.895	0.9201	0.9211	0.9413	0.9424
0.818	0.8276	0.8296	0.8229	0.8408
0.761	0.7789	0.7783	0.7970	0.8080
0.896	0.9172	0.8977	0.9260	0.9286
0.807	0.8171	0.8175	0.8232	0.8313

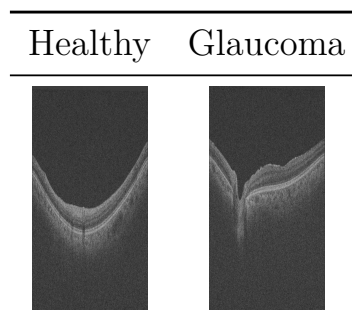
3.4.4 Generalisation of the Weights Formula into the Glaucoma Classification Problem

This research required further evidence that the proposed weight function can be generalised for different cases of medical classification problems. The glaucoma images were obtained from optical coherence tomography. The example of the healthy versus glaucoma-diseased images are shown in Table 3.6. Three layers exist in glaucoma disease; they are internal limiting membrane, inner retinal pigment epithelium (IRPE) and outer aspect of the Bruch's membrane. The difference between healthy and glaucoma image is that the diseased image has dripped contour for the IRPE layer, while the normal image has a smooth surface. The distinctiveness is shown in Table 3.6. The size for each image was 512 x 992 pixels.

This research recognised that the glaucoma dataset has an imbalanced training set; also, the glaucoma case suffers from the binary classification problem, which makes it distinct from the multiclass problem in the chest X-ray dataset [173]. Since there is a problem generalising from multiclass into binary classification, this research can claim the proposed weights function also works well in the binary classification problem.

The glaucoma dataset consists of 254 training samples, 58 validation samples and 58 test samples. This research follows the work of Yamashita et al. [177], using 70:15:15 proportion for training-validation-testing split configuration. This research asserts that the work in [177] shares some similarities with ours in terms of the number of samples in the dataset, the problem it addresses and the chosen method. As aforementioned, the training set suffers from an imbalance problem. There are 182 negative cases and 72 positive glaucoma cases in the training set. Since the number of samples is considered too small to feed an established pre-trained deep-network algorithm, this research decided to construct a custom shallow network to classify the images. During training, the experiment resized the images into 64 x 64.

Since smaller input sizes feeding a neural network will lead to faster training, this research estimated that the size of 64 X 64 would be the smallest size possible to gain the best classification accuracy in the lowest training time. Although the images have the grayscale texture pattern, the proposed network used RGB

Table 3.6: **Glaucoma Cases**

mode for training. The network layers consisted of a Gaussian noise layer after the input layer. After the Gaussian noise layer, this experiment placed another two consecutive blocks of intermediate layers. Each intermediate layer consisted of a 3 x 3 convolution layer with 32 filters, followed by a 2 x 2 average-pool layer. The experiment flattened the outputs from the last intermediate layer; then, the research used a fully connected layer with 16 nodes. For all convolutional and fully connected layers, the experiment used the ReLu activation functions.

This experiment finalised the network with a final classification layer, for which the experiment used a sigmoid activation with two output nodes. The experiment trained the network for 20 epochs with a batch size of 1; the initial learning rate of 0.001 reduced 10 times if the validation loss plateaued after one epoch. The experiment obtained $\beta = 1 - 4.0 \cdot 10^{-3}$ by the use of Equation 3.3. The experiment achieved increased accuracy from the imbalanced weight model, with 76% performance. Then, the weighted binary cross-entropy loss with 81% performance and the highest accuracy was performed by the weighted focal-loss model, with 93% and 100% performance. This increment is listed in Table 3.7. This research concluded that the proposed weight function works well for the binary classification problem.

Table 3.7: **Effectiveness of Weights for Glaucoma Classification**

	Imbalanced binary cross entropy loss	Weighted binary cross entropy loss $\beta = 0.996$	Weighted focal loss $\beta = 0.996$ $\alpha = 0.1$ $\alpha = 0.1$ $\gamma = 0.0$ $\gamma = 0.01$	
Accuracy	76 %	81 %	93 %	100 %

The weights in Table 3.7 are different from those used for the chest X-ray dataset. This is mainly because both datasets have a different number of samples and positive–negative patterns. In this case, the formula to calculate the weights functions outputs differently for both datasets. However, since the glaucoma dataset is relatively small [91], it is faster to perform experiments. The experiments selected the combination of the α and γ value for the focal-loss function arbitrarily into the one that demonstrated the best classification performance.

3.5 The Intuitive Theoretical Background and Evidence from Experiment

Since part of the approach inherits the strength of focal loss [101], and the class-balanced approach [44], this research can obtain further theoretical analysis from the proposed approach intuitively using [101] and [44]. The main distinction of focal loss with binary cross-entropy loss is the existence of α and γ parameter. Cui et al. mentioned “the class-balanced term can be viewed as an explicit way to set α in focal loss based on the effective number of samples” [44]. However, Lin et al. also stated “a common method for addressing class imbalance is to introduce a weighting factor $\alpha \in [1, 0]$ for class 1 and $1 - \alpha$ for *class* – 1” [101]. This research implements these two statements into the elaboration in Equation 3.6.

Table 3.8: **The Improvement of The Proposed Weight Calculation**

Pathology	Focal loss DenseNet-121		Weighted focal loss $\beta =$ 0.9998 DenseNet-121	
	Validation	Test	Validation	Test
Cardiomegaly	0.9155	0.9096	0.9092	0.9090
Emphysema	0.9140	0.9178	0.9056	0.9327
Edema	0.9141	0.8851	0.9147	0.8917
Hernia	0.8614	0.9135	0.9067	0.9404
Pneumothorax	0.8896	0.8663	0.8973	0.8749
Effusion	0.8822	0.8762	0.8792	0.8827
Mass	0.8622	0.8430	0.8655	0.8514
Fibrosis	0.8277	0.8219	0.8313	0.8308
Atelectasis	0.8191	0.8079	0.8228	0.8259
Consolidation	0.8247	0.8007	0.8224	0.8043
Pleural thicken.	0.8219	0.7874	0.8214	0.7910
Nodule	0.7823	0.7751	0.7888	0.7756
Pneumonia	0.7722	0.7504	0.7586	0.7698
Infiltration	0.7061	0.7073	0.7113	0.7166
Average	0.8424	0.8330	0.8453	0.8427

The experiments provide further evidence for the theory. The improvement from the change of the proposed formula is $\pm 1\%$ under the test set, according to Table 3.8. Both experiments were performed with $\alpha = 0.5$ and $\gamma = 1.0$ for the focal-loss parameters. Tables 3.2 and 3.8 use an identical split. The training set consists of 78,468 images, the validation set consists of 11,219 images and the test set consists of 22,433 images. The training took 38 minutes for one epoch with a batch size of 32, and the test took 12 minutes with a batch size of 8.

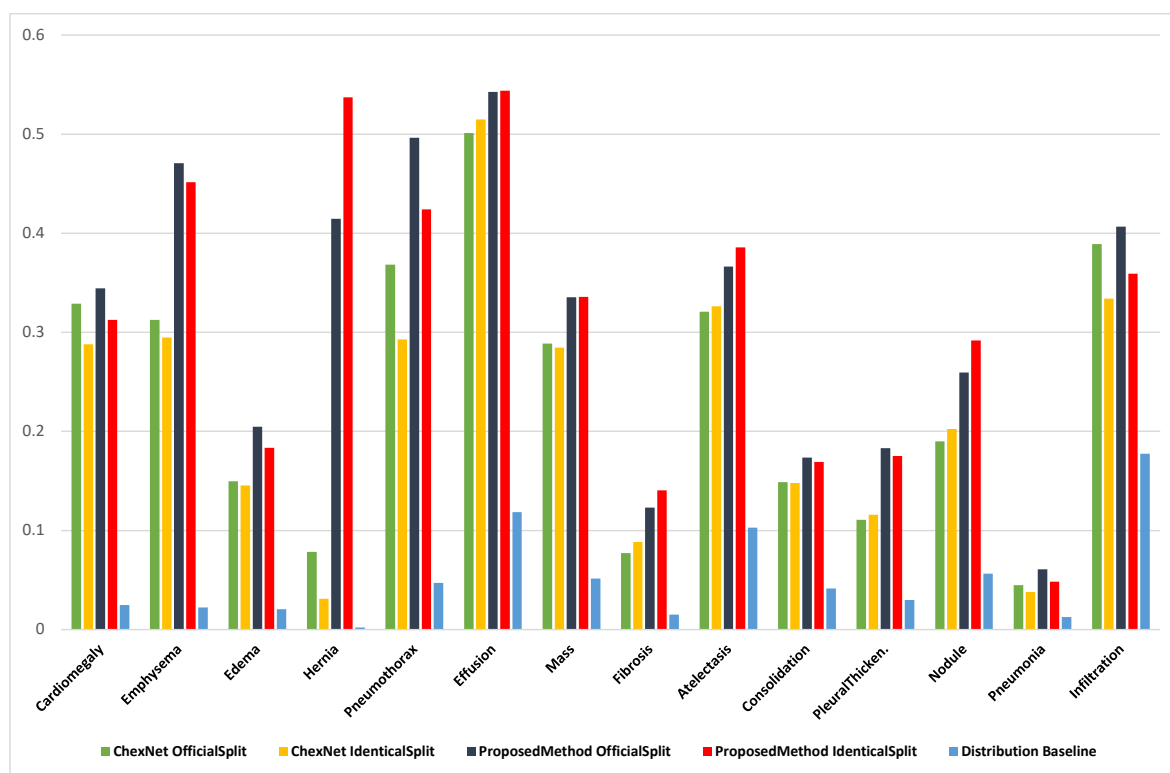
3.6 The Imbalance Metric Evaluation

Table 3.9: The AU-PRC Improvement

Pathology	Reproduce ChexNet [136]		Weighted focal loss $\beta =$ 0.9998 EfficientNet-B3 two-stage		Baseline AU-PRC from Distribution
	Official	Identical [139]	Official	Identical [139]	
Cardiomegaly	0.3288	0.2880	0.3444	0.3127	0.0247
Emphysema	0.3125	0.2948	0.4706	0.4515	0.0224
Edema	0.1497	0.1455	0.2048	0.1835	0.0205
Hernia	0.0785	0.0311	0.4147	0.5372	0.0020
Pneumothorax	0.3683	0.2929	0.4965	0.4242	0.0472
Effusion	0.5012	0.5149	0.5428	0.5439	0.1187
Mass	0.2887	0.2847	0.3355	0.3357	0.0515
Fibrosis	0.0773	0.0886	0.1231	0.1405	0.0150
Atelectasis	0.3208	0.3262	0.3664	0.3859	0.1030
Consolidation	0.1488	0.1479	0.1736	0.1692	0.0416
Pleural thicken.	0.1109	0.1159	0.1831	0.1754	0.0301
Nodule	0.1899	0.2025	0.2595	0.2919	0.0564
Pneumonia	0.0448	0.0381	0.0609	0.0484	0.0127
Infiltration	0.3891	0.3342	0.4067	0.3592	0.1774
Average	0.2364	0.2218	0.3130	0.3114	0.0517

Table 3.9 and Figure 3.1 show the advancement from the proposed method in comparison with previous work [136] and the baseline retrieved from the dataset. This research calculated the baseline of AU-PRC metric directly from the dataset’s distribution of positive and negative samples using Equation 3.10. The bold fonts show the top scores achieved between a same split-set configuration. The hernia has the lowest number of positive samples in the distribution. Despite being in the greatest minority, the proposed algorithm for hernia resulted in a couple of hundred more AU-PRC than the baseline, as shown in Table 3.9 and Figure 3.1.

Figure 3.1: Area Under Precision-Recall Curve



3.7 Third-Phase Training Saturation

To ensure the effectiveness of progressive image resizing, this research needs to ensure the applicable boundary from the method for the dataset. The experiment performed third-phase training with an identical configuration from Table 3.2, except the resized input image changed into 1,024 x 1,024. Table 3.10 show results from the third-phase training; this research concludes that further training would not improve classification performance. Since the method of progressive image resizing is another form of transfer learning, the reason for the lack of improvement for the third-phase training is most likely because the network did not find better features to learn. This research summarises that a 1,024 x 1024-pixel input in the third phase did not provide new features for the network to learn and improve the

classification.

Table 3.10: **Third-Phase Training Results from Table 3.2**

Pathology	Phase 1	Phase 2	Phase 3
Cardiomegaly	0.9137	0.9144	0.9101
Emphysema	0.9471	0.9558	0.9517
Edema	0.9021	0.9071	0.9025
Hernia	0.9357	0.9409	0.9328
Pneumothorax	0.9003	0.9092	0.9079
Effusion	0.8899	0.8923	0.8877
Mass	0.8596	0.8669	0.8580
Fibrosis	0.8526	0.8657	0.8648
Atelectasis	0.8350	0.8397	0.8341
Consolidation	0.8124	0.8208	0.8118
Pleural thicken.	0.8041	0.8136	0.9025
Nodule	0.8043	0.8293	0.8255
Pneumonia	0.7721	0.7703	0.7683
Infiltration	0.7297	0.7363	0.7254
Average	0.8542	0.8616	0.8561

3.8 Discussion

To provide greater insights of the effect from different splits into classification performance, several split sets have been assessed in performance evaluation.

The results in Tables 3.3.A and 3.3.B are the improvements made compared with the most recent research [46]. The individual comparisons for each disease with the latest research [62] are listed in Table 3.2. This research achieved better performances than did the work of Guan et. al [62]. Further, this research proposes technically more simple approaches to achieve the results.

The standard procedure is to follow the “official” splits [157]. This research reports the results in Tables 3.5.A and 3.5.B. To the best of our knowledge, only [26] reported the performance evaluation of a random fivefold cross-validation from the chest X-ray dataset [173]. This research reports the results from the proposed method in Table 3.4.

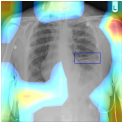
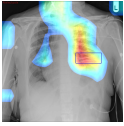
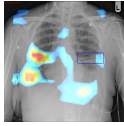
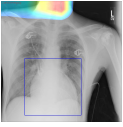
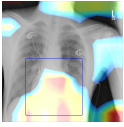
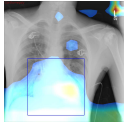
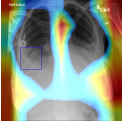
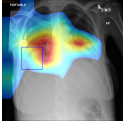
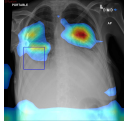
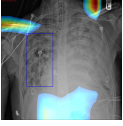
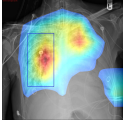
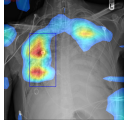


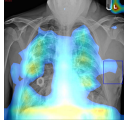
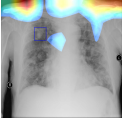
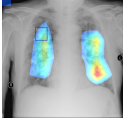
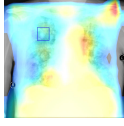
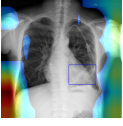
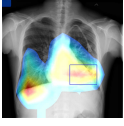
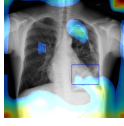
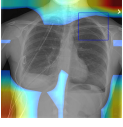
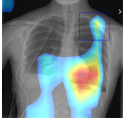
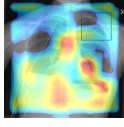
Some split sets are considered “non-standard” settings. These splits are from Github pages. [139] is the third-party re-implementation of [62] and [174] is the third-party re-implementation of [136]. However, after further investigation, [139] and [174] were subject to identical training, validation and testing sets. This research reports the results with the custom sets [139, 174] in Table 3.2.

Since the diversity of split sets is a well-known problem for the dataset’s [173] evaluation, the use cross-validation is a fair method to follow. [26] is the only work that reported performing cross-validation on the dataset [173]. This research achieved better performance in five-folds cross-validation experiment than did Baltruschat et al. [26].

The class-activation-mapping (CAM) method [178, 191] visualises the discriminative features from the deep network’s last layer in the form of heatmap localisation. The more the heatmap visualisation matches the groundtruth bounding-box from the dataset, the network has a better understanding of the images. This research visualises the classification performances with heatmap from the CAM method in Table 3.11.

This research obtained the bounding boxes as the annotation groundtruth for only eight classes, which are available from the file BBox List 2017.csv [157]. The annotations consist of 984 images, and the number of samples for each class is not distributed evenly. Table 3.11 shows that the networks equipped with the proposed method read the area of the disease better than did the baseline.

Table 3.11: The Heatmap from Different Methods and Various Networks

Pathology	Baseline reproduce ChexNet [136]	Weighted focal loss DenseNet-121	Weighted focal loss EfficientNet-B3
Atelectasis			
Cardiomegaly			
Effusion			
Infiltration			
Mass			
Nodule			
Pneumonia			
Pneumothorax			

Chapter 4

Hyperparameters and Network Architectures Learning for Features Classification

4.1 Introduction

This study aims to reduce the computational cost of training deep networks for medical image classification. However, it is a common practice that a deep neural network dedicated to an image classification task requires substantial computational resources and days to train. Hence, this study used feature vectors instead of the original images to feed the network, similar to previous works [63, 122, 144].

This study expected the outcomes to be practical to use within low-computing resources environments. The research applies to improve classification performance within mobile devices, system on chip (SoC) or internet of things (IoT) devices. This study has strong confidence that features will significantly reduce the network floating-point operations (FLOPs); thus, it will decrease training time. The simple logical reasoning for this is that with features for a neural network, the required input for the network is smaller than the use of original images, which reduces the required FLOPs. Further, the classifier network does not require the convolutional layer. Hence, it is computationally cheaper to tune the hyperparameters in the feature spaces [122, 144].

This study also has strong confidence that using extracted features to construct a new neural network architecture will have a lower classification performance than

original images. This study has a particular reason to support this hypothesis: extracting salient useful features from images is an automated process without human intervention. It has a disadvantage; when important features are unintentionally removed during the process, it will reduce classification performance later. This study extracts features from the EfficientNet-B3 network, which were trained in Chapter 3.

The contribution from this work is the findings of smaller FLOPs architectures with comparable classification results. The novelty is to find a computationally low-cost architecture for features classification tasks in the medical domain through Bayesian optimisation. Conversely, previous research [66, 147] focused on the contributions under synthetic datasets and MNIST (Modified National Institute of Standards and Technology). The proposed approach maximises the potential of balanced exploitation–exploration under the weighted expected improvement acquisition function [51, 152]. This study aimed to prove that the approach is suitable for finding a sound trade-off between classification performance and the required cost of computing. In short, the proposed approach attempted to find the most suitable network architecture and hyperparameters to perform a classification task with the help of features from medical images.

The study also provides a further contribution that the proposed method works not only for the Chest X-Ray dataset but also is generalisable to other datasets. Experimental results from the ISIC-2016 dataset support this additional contribution. This study proposed the iteration-partitioning framework to maximise the potential of exploration-exploitation balance from the expected-improvement acquisition function’s weighted version under the constraints of minimal computing resources. Similarly, [182] required HPC (High-Performance Computing) resources to perform this task. Another advantage of the proposed work is that this study only required lower computing resources, less than [182]. Further, the approach is suitable to resume in an altered optimisation process. This study alters the search space in the proposed method.

The progression of Chapter 4 is the necessity of lower computational cost of the final model in comparison with the model in Chapter 3. The advantage of the model in Chapter 4 is more adjustable than in Chapter 3 to the low-cost computing device.

4.2 Method

4.2.1 Research Contributions and Novelty Statement

The first contribution from this research is the comparable classification performance, using minor FLOPs neural-network architecture for the features classification task. Then, the second contribution is generalising the proposed method to various datasets. This study contributes and provides evidence that the proposed Bayesian iteration-partitioning framework works both for (i) the Chest X-Ray dataset and (ii) the skin cancer dataset.

The third contribution is that the proposed method ensures the maximum result has been achieved from the Bayesian-Optimisation exploitation process in the quickest time possible. The trade-off between exploration and exploitation occurs automatically and tend to exhibit stochastic-random behaviour. The proposed method forces a more deterministic Bayesian-Optimisation, ensuring the maximum magnitude of results is achieved whilst minimising the time required. The fourth contribution is that the research provides the proposed method's applicability to mobile devices into implementation. The results are empirically reported and tested both on the Android simulator and the actual device (Samsung Galaxy S8).

The fifth contribution is that all the source codes are shared publicly [127] for further study, ensuring that the works are repeatable and well-documented.

4.2.2 The Existing Computational Cost for Neural Network

A universal tool to measure computational cost is the FLOPs unit. This study used the following definition of FLOPs [83]: "to refer to the number of floating point operations executed". Another measurement unit is the MACCs unit (multiply accumulation operations). One MACC operation is described as one operation of accumulation and one operation of addition in the accumulator [25]. Hence, one MACC is equivalent to two FLOPs [74, 77, 79]¹. Since the real-world application of deep learning uses central processing unit and general purpose-graphical processing

¹The literatures measure MobileNetV1 architecture with different units, MACCs and FLOPs

unit (GP-GPU), which are based on the von Neumann architecture [50, 171], the MACCs and FLOPs calculations also follow the von Neumann model.

An earlier study [162] presented in the NIPS 2018 workshop proposed the use of FLOPs as the objective function to learn sparse neural networks. [162] directly replaced L0 regularisation with a customised FLOPs objective. This FLOPs objective is applied during the training process ². However, this thesis maintained the use of loss function as the objective during the training process and AUROC to measure the classification performance. Further, this study finalised the FLOPs calculation to evaluate the computational cost from the neural network.

The advantage of measuring computational resources with FLOPs is that several significant studies [71, 80, 161, 162] in deep learning have used it to measure the network’s computational cost. [144] reported that different mobile-device platforms result in various execution times for a single face-detection algorithm. [144] used the features extracted from the neural network. This study determined that the use of time to measure the computational cost is not accountable because of the evidence cited in [144].

4.2.3 The Proposed Approach for Reducing Computational Cost

This research performed feature extraction from the Chest X-ray images dataset [173] from an EfficientNet-B3 [161] backbone. This study took the features from the last global average-pool layer to perform experiments. Each feature vector input was 1 x 1,536 in size. This research built a neural network backbone and used extracted features to feed the network. Further, it used Bayesian optimisation to approximate the sets of best hyperparameters and the best architectures for the network with a chosen prior surrogate function [137]. The approximation of a hyperparameter \boldsymbol{x} that outputs the maximum magnitude from an objective function $f(x)$ and uses the search space χ will be written as in Equation 4.1:

$$x^* = \operatorname{argmax}_{\boldsymbol{x} \in \chi} f(x) \tag{4.1}$$

²https://github.com/AMLab-Amsterdam/L0_regularisation

The design of Bayesian optimisation requires two components. First is the prior surrogate function, which defines the assumptions and beliefs over the objective function $f(x)$. The second is the acquisition function, which defines the selection of the next sample points for the objective function $f(x)$. The challenge of Bayesian optimisation is to find the trade-off between exploitation and exploration.

The exploitation aims for the maximum value from objective function $f(x)$ given the value of \mathbf{x} , and exploration aims to gain more variability of x from the search-space χ . The Gaussian process [137] is a well-known standard prior surrogate function for Bayesian optimisation. A Gaussian process can be defined by both its mean function $m(x)$ and the covariance function [137].

A critical property of Gaussian process is its covariance function, also called the kernel $\kappa(\mathbf{x}, \mathbf{x}')$, which measures the distance between \mathbf{x} and \mathbf{x}' . The kernel function estimates the unknown value of $f(x')$ given the preceding $f(x)$; hence, the kernel also measures the similarity between \mathbf{x}' and \mathbf{x} . To achieve a better value of x^* in Equation 4.1, there are several acquisition functions that are available.

In the application, this research used the acquisition function to identify several hyperparameters, such as the optimum number of neurons, the required level of noise, the appropriate learning rate and the critical dropout factors. Since Bayesian optimisation is a continuous stochastic function, the acquisition function α to recommend next x_{k+1}^* from k observations can be written in the form of [104]:

$$x_{k+1}^* = \operatorname{argmax}_{\mathbf{x} \in \chi} \alpha(x; D_k) \Leftrightarrow \mathbf{x}' \tag{4.2}$$

where D_k is the k -th observation from dataset D. The proper configurations of the acquisition function α from Bayesian optimisation and the kernel function from the Gaussian process; $\kappa(\mathbf{x}, \hat{\mathbf{x}})$ will maximise the output from the objective function $f(x)$.

4.2.3.1 The Kernel of Gaussian Process

Equation 4.3 defines a squared exponential kernel that is widely known as the RBF kernel:

$$\kappa(x, \hat{x}) = \exp\left(-\frac{\mathbf{d}(x, \hat{x})^2}{2l^2}\right) \tag{4.3}$$

The Matérn kernel [137] in Equation 4.4 is also the generalisation form of the RBF kernel.

$$\kappa(x, \hat{x}) = \frac{1}{\gamma(\mathbf{v}) 2^{v-1}} \left(\frac{\sqrt{2\mathbf{v}}}{\mathbf{l}} \mathbf{d}(x, \hat{x}) \right)^v K_v \left(\frac{\sqrt{2\mathbf{v}}}{\mathbf{l}} \mathbf{d}(x, \hat{x}) \right) \quad (4.4)$$

where $\mathbf{d}(x, \hat{x})$ is the Euclidean distance between point x and \hat{x} .

The magnitude $\mathbf{l} > 0$ is the length scale parameter; this parameter prescribes the length of maximum extrapolation as \mathbf{l} units away from the data. In the condition of $\mathbf{v} \rightarrow \infty$ the Matérn kernel is identical to an RBF kernel.

4.2.3.2 The Partition of Iterations

This study proposes the partitioning of iterations; this effort is mainly to reduce the required computational resources. The previous work [182] performed search-space partitioning utilising HPC resources. The proposed work has the advantage of lower computing cost. This study still used classical sequential model-based optimisation, while [182] used distributed and parallel computational resources. Another advantage of the proposed approach is that it can alter and supervise the optimisation process. This includes resuming with altered settings of search space.

4.2.3.3 The Acquisition Function

The expected improvement (EI) is an example of the acquisition function for the Bayesian optimisation. Suppose we have the black-box objective function $f(x)$ with the search-space χ and the possible best magnitude achieved is $\hat{f}(x)$; then this study can define the acquisition function $\alpha EI(\mathbf{x} \in \chi)$ in Equation 4.5

$$\begin{aligned} \alpha EI(\mathbf{x} \in \chi) &= \mathbb{E} \left[\max \left(0, \hat{f}(x) - f(x) \right) \right] \\ &= \mathbf{h} \cdot \sigma(x) \cdot \theta(\mathbf{h}) + \sigma(x) \cdot \phi(\mathbf{h}) \end{aligned} \quad (4.5)$$

where $\mathbf{h} = \frac{\mu(x) - \hat{f}(x)}{\sigma(x)}$ with $\mu(x)$ is the mean and $\sigma(x)$ is the standard deviation; also θ is the cumulative distribution function and ϕ is the probability density function.

The value of \mathbf{h} is a utility function that measures the distance of the approximation value to the current best performance value. The search within domain χ will be accomplished whenever the algorithm requires a new hyperparameter

tuning to maximise the outcomes. Another example of acquisition function is the upper confidence bound (UCB), which can be defined in Equation 4.6

$$\alpha UCB(\mathbf{x} \in \mathcal{X}) = \mu(x) + \tau \sigma(x) \quad (4.6)$$

where coefficient τ is used to balance exploration and exploitation. Small τ leads to more exploitation, while large τ provides more exploration from the unknown points. However, τ is only defined as any positive number in the literature; there is no definition of the upper bound from the largest possible number that suits the best exploration. Because of this, the study aimed to have better approach to define the magnitude of exploitation-exploration required to improve the existing problem.

The weighted version [51, 152] of expected improvement [86, 117] defines a better solution to address this problem. The acquisition function [51, 152] can be written in the form of Equation 4.7.

$$\alpha \omega EI(\mathbf{x} \in \mathcal{X}) = \begin{cases} \omega \cdot \mathbf{h} \cdot \sigma(x) \cdot \theta(\mathbf{h}) + (1 - \omega) \cdot \sigma(x) \cdot \phi(\mathbf{h}); & \text{for } \sigma(x) > 0 \\ 0; & \text{for } \sigma(x) = 0 \end{cases} \quad (4.7)$$

This acquisition function defines a coefficient ω to address the exploitation–exploration balance. The ω coefficient does not exist in the conventional EI [86, 117] acquisition function. The value of $\omega \in [0, 1]$ defines the lower and upper bounds that are applicable to use, where ω close to 0 will lead to the intention of exploitation and ω close to 1 will lead to better exploration. Figure 4.1 shows the value of ω to capture the next potential input for the acquisition function; the distribution with small ω tends to focus on the exploitation of specific points.

The study defines a further contribution, to make the proposed work distinct with the common weighted expected improvement [51, 152]; this study performed search-space pruning in Algorithm 1. The proposed search-space pruning **maximises the potential** of the balanced exploration-exploitation process for the problem. The maximisation occurs because this study used the pruned search-space with high exploitation parameter (small ω). The theoretical Figure 4.1 shows that small ω with small input (x-axis) will potentially achieve higher magnitude for the black-box function (y-axis). The pruned search-space is the proposed

approach **to feed the exploitation from the high probable regions** of inputs. The theoretical arguments are supported in Figure 4.1 and Algorithm 1; the evidence is from experiments documented in the Jupyter-notebook code.

Algorithm 1 Iteration Partitioning and Search Space Pruning

input : The Candidate Hyperparameters

output: The Classification Results from The Concatenated Iterations with Altered Settings

*Stage 1: Exploration phase *

initialisation: split $n - partitions$ of $iterations$

while $iteration < (iterations/n)$

 optimise with high exploration *large ω *

 capture best search-space

return **best potential search-space**

end exploration

*Stage 2: Exploitation phase *

initialisation: **pruning the search-space with only best potential**

while $iterations/n < (iterations + 1)$

 optimise with high exploitation *small ω *

return **high-probable best classification**

end exploitation

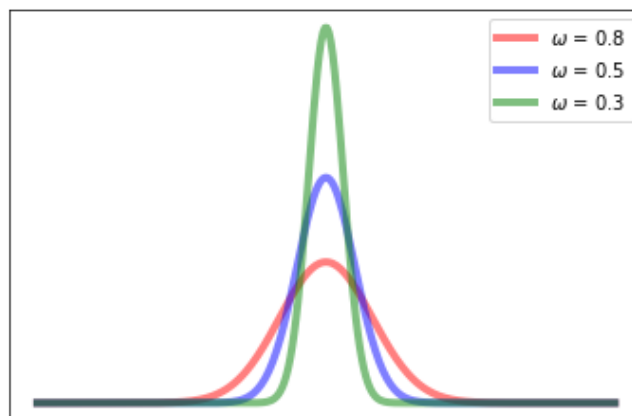


Figure 4.1: The Width of ω and Balance of Exploration-Exploitation

4.2.3.4 Gaussian Noise

The noise injection in the neural network is similar to that used to perform data augmentation [61]; also, it works as another form of regularisation to prevent overfitting. Since the noise layer provides some degree of distraction to the original input signals during the training process, it was intended to make the final model more robust.

The provided Gaussian noise layer within the Keras library [41] is equipped with a tuneable standard deviation hyperparameter. The other way to add noises is to directly augment the noise vectors into the input vectors. Despite the potential classification improvement because of the use of noises-augmentation, this method would not significantly increase the required computational resources..

4.2.4 The Total FLOPs/MACCs Calculation

This study defines the total of required computational resources to perform classification in a neural network architecture as the relation between FLOPs/MACCs and the number of epochs to train, depicted in Equations 4.8 and 4.9:

$$total_{FLOPs} = (2n + 1) \cdot base_{FLOPs} + extraction_{FLOPs} \quad (4.8)$$

$$total_{MACCs} = (2n + 1) \cdot base_{MACCs} + extraction_{MACCs} \quad (4.9)$$

where n is the number of training epoch, $base_{FLOPs}$ and $base_{MACCs}$ are the required computational resources to perform one-time pass-through of the base neural network architecture. The multiplication of $2n$ to calculate the computational resources from the validation phase after the end of each epoch training, then the finalisation ($2n + 1$) is performed to calculate the required computational resources for the test phase. **The extraction FLOPs is one-time pass-through for the original network to perform feature extraction with the pre-trained weights.** This process is similar to that of [144], which also extracted the features from Deep Network then performed feature classification in the mobile platform.

The use of pre-trained weights is common to lift the classification performance; previous works [49, 136] used ImageNet pre-trained weights to tune the network. The use of pre-trained weights reduces the number of FLOPs; thus, it decreases the total computational resources required. Since the primary objective is shifted from ImageNet classification into a new classification task, the process is well-known by the term “transfer-learning”. This term is used because different tasks are assigned; for example, the first task is the ImageNet classification. Then, the weights from the first task are used to train the network and classify other subjects (e.g. chest cancer [136] and skin cancer [49]). However, this study used the pre-trained weights from the Chest X-ray dataset with DenseNet-B3 to extract the features rather than to perform transfer learning.

4.2.5 The Training Epochs and Total FLOPs Correlation

A classification task requires achieving optimal weights from a neural network, which is closely related to the number of epochs required during the training process. The calculation of the required computational resources in both FLOPs and MACCs units are written in Equations 4.10 and 4.11, respectively:

$$training_{FLOPs} = trainingEpochs \cdot base_{FLOPs} \quad (4.10)$$

$$training_{MACCs} = trainingEpochs \cdot base_{MACCs} \quad (4.11)$$

In case a neural network classifier does not apply “deep features” to feed its input; the network is not for application of Equations 4.8 and 4.9, but still requires Equations 4.10 and 4.11.

4.3 Experiments and Results

This experiment designs a neural network “template” in the Keras wrapper [41] of the Tensorflow [21] library to facilitate the hyperparameters learning with Bayesian optimisation. This experiment modifies the Bayesian optimisation implementation from the public repositories [72,125]. The “network-template” basically consists of an input layer followed by a flattened layer. After the flattened layer, this experiment inserted a Gaussian noise layer, aimed to achieve greater robustness. Then, this experiment added a fully connected layer followed by a dropout layer and final classification layer. The hyperparameters from the “template” network were dynamically adjusted from the Bayesian-optimisation algorithm. The “template” generates two types of network: (i) a network without a noise layer and (ii) a network with noise layer.

Table 4.1: Results Comparison with Previous Study under The Official Splits

	ResNet-50 Baltruschat et al. [26]	Variant of DenseNet-121 Gündel et al. [64]	Features B.O Neural Network without noise	Features B.O Neural Network with noise
Cardiomegaly	0.875	0.883	0.8923	0.8919
Emphysema	0.895	0.895	0.9408	0.9399
Edema	0.846	0.835	0.8600	0.8604
Hernia	0.937	0.896	0.9278	0.9204
Pneumothorax	0.819	0.846	0.8902	0.8912
Effusion	0.822	0.806	0.8389	0.8394
Mass	0.820	0.821	0.8344	0.8357
Fibrosis	0.816	0.818	0.8417	0.8423
Atelectasis	0.763	0.767	0.7909	0.7907
Consolidation	0.749	0.745	0.7600	0.7616
Pleural thickening	0.763	0.761	0.8074	0.8072
Nodule	0.747	0.758	0.7979	0.7978
Pneumonia	0.714	0.731	0.7341	0.7330
Infiltration	0.694	0.709	0.7046	0.7033
Average	0.806	0.807	0.8301	0.8296
base MACCs	1,938.98M [159]	1,436.065M [159]	0.635M [166]	0.635M [166]
base FLOPs	3,877.95M [159]	2,872.13M [159]	1.27M [166]	1.27M [166]

This study used the ReLu activation function for the fully connected layer and the sigmoid function for the final classification layer. Each layer’s numeric-type hyperparameters were adjusted dynamically with the Bayesian optimisation algorithm. This study performed two consecutive Bayesian optimisations with weighted EI acquisition functions. For each stage, this study performed seven point searches, then used the history for the next stage of optimisation. The primary differences from the first and second stages is the search spaces and the value of ω coefficient used for the weighted EI. This study performed the vigorous exploration with $\omega = 0.9$ for the first stage. Then, this study resumed a substantial exploitation with $\omega = 0.1$ for the second stage.

Table 4.2: **Identical Split Comparison [139]**

Pathology	Third party [139] of Guan et al. [62]	B.O. Features Neural Network	
		without Noise	with Noise
Cardiomegaly	0.9097	0.9136	0.9091
Emphysema	0.8905	0.9538	0.9506
Edema	0.9185	0.9044	0.9018
Hernia	0.9064	0.9512	0.9384
Pneumothorax	0.8794	0.9100	0.9106
Effusion	0.8843	0.8915	0.8904
Mass	0.8707	0.8662	0.8651
Fibrosis	0.8208	0.8630	0.8583
Atelectasis	0.8225	0.8393	0.8393
Consolidation	0.8210	0.8196	0.8176
Pleural Thicken.	0.8127	0.8094	0.8050
Nodule	0.7691	0.8280	0.8302
Pneumonia	0.7614	0.7676	0.7665
Infiltration	0.7006	0.7343	0.7347
Average	0.8405	0.8608	0.8584
base MACCs	1,938.98M [159]	0.635M [166]	0.635M [166]
base FLOPs	3,877.95M [159]	1.27M [166]	1.27M [166]
extraction MACCs	-	4,275M	4,275M
extraction FLOPs	-	8,550M	8,550M
training MACCs	193,898M	4.445M	4.445M
training FLOPs	387,795M	8.89M	8.89M
total MACCs	193,898M	4,279.445M	4,279.445M
total FLOPs	387,795M	8,558.89M	8,558.89M

* This study found the third-party re-implementation [139] reported lower performances than did [62]. Guan et al. [62] did not provide the official code and split sets. The critical classification problems for the dataset [173] is that different splits will lead to different performances [26]

The search space was pruned for the second stage based on the results from the first stage; this will provide stronger exploitation to the points of interests. The results are shown in Tables 4.2 and 4.1.

This study shows that the proposed approach results in better outcomes than in previous works [26, 62, 64]. This study measures the claim with two pieces of evidence: (i) the classification performance and (ii) the required computational resources of the final architecture.

To calculate the total required computational resources, this study required that some variables be written in Equations 4.8 and 4.9. However, several works [26, 64] do not provide details with the number of the epochs used to train their model to achieve the reported results. This study limits the calculation **only** with the use of the final architectural model in Table 4.1 because of this. Nevertheless, this study can discover more detail about the work of Guan et al. [62] in Table 4.2.

The work [62] mentions the use of ResNet50 in two branches (global and local); for each branch, 50 epochs of training were performed. This study concludes the work of [62], which performed 100 epochs of training of ResNet50 for both branches. Conversely, this research performed only seven epochs with the proposed network, which was optimised by Bayesian optimisation to achieve the reported results. In terms of improved classification performance, the improvement rate in comparison with [62] is depicted in Tables 4.3.A and 4.3.B.

Table 4.3.A: Improvement Rate

Name	Hernia	Pneumonia	Fibrosis	Edema	Emphysema	Cardiomegaly	Pleural Thick.	Pneumothorax
Rate	+4.48%	+0.62%	+4.22%	-1.41%	+6.33%	+0.39%	-0.33%	+3.06%

Table 4.3.B: Improvement Rate (cont.)

Consolidation	Mass	Nodule	Atelectasis	Effusion	Infiltration	Average
-0.14%	-0.45%	+5.89%	+1.68%	+0.72%	+3.37%	+1.81%

To have a more detailed calculation of the required computing resources, this study calculated the real FLOPs from the EfficientNet B3 for both stage one and stage two during training with the use of original input images. This research calculated the one-epoch feature-extraction process, similar to Equation 4.8. One crucial element is that the FLOPs required during training are not identical between the first and second stage; this is because of the different input size. This

study refers to the literature [161] to calculate the required FLOPs for both training stages of EfficientNet-B3. [161] shows the calculation of EfficientNet-B0, which multiplies the required FLOPs into 4.75 times with the use of twice input resolution. This means that 1,800 M FLOPs are required for the first stage, then multiplied into 8,550 M FLOPs for the second stage.

This research takes the required computational resources to perform feature extraction simply because this study needed greater transparency to calculate the overall process. Importantly, the previous works [76, 143, 144] **only** counted the final model computational resources and ignored the preliminary processes considered in the sum of the total final.

4.3.1 The Generalisation of the Method for the Skin Cancer Classification

The 2016 ISIC dataset [65] for the classification task consists of 900 skin lesion images for training and 379 images for testing. This research pre-processed and converted the images into array format with the help of public code [70]. The classification itself is a binary task, which aims to separate malignant cases from benign. The dataset is highly imbalanced, where the training set consists of only 173 malignant cases and the test-set consists of 75 malignant cases. Reasonably, the imbalance problem that affects the official evaluation metric is the average precision (AP) [11].

The proposed approach outperforms the champion of the ISIC 2016 challenge, both in terms of the classification performance and the required computational results, as depicted in Table 4.4. The approach also achieves perfect performance under the specificity metric as depicted in Table 4.5, also results in excellent performance under the sensitivity metric as depicted in Table 4.6. The proposed Bayesian-Optimisation approach in Table 4.4 ,4.5 and 4.6 were under a single training session. However, they represent the superior performance for that particular metric. One primary advantage of the proposed method is the applicability to optimise under specific metrics for each training session, the results in Table 4.4 ,4.5 and 4.6 were under the sensitivity metric optimisation.

A specific model that optimises a particular metric is beneficial for healthcare cases. A model with good sensitivity will accurately classify the cases of the positive disease, and a model with reasonable specificity will correctly identify the healthy cases [158]. Sensitivity emphasises the disease over the healthy, while specificity emphasises the healthy over the disease. Suppose the importance is different between the disease cases and the healthy cases; the proposed model fully support this particular metric optimisation.

Table 4.4: **The Benchmark under Average Precision Metric**

The Approach	Estimated Architectural Computational FLOPs	Average Precision	Notes
Bayesian-Optimisation Framework (ours)	1.5 M	0.727	raw images input (array) without specific / customized Feature Engineering method
ResNet - 50 Training-Optimisation	3,877.95M [159]	0.709	Published in Q1 Journal [33]
CUMED	3,877.95M [159]	0.637	ISIC 2016 Challenge Champion [185] ResNet - 50 variant

Table 4.5: **The Benchmark under Specificity Metric**

The Approach	Estimated Architectural Computational FLOPs	Specificity
Bayesian-Optimisation Framework (ours)	1.5 M	1.0
ARDT-DenseNet	2,872.13 M [159]	0.756 [176]
DenseNet-100	2,872.13 M [159]	0.742 [176]
ResNet-101	7,597.95 M [159]	0.739 [176]
ResNet-50	3,798.98 M [159]	0.714 [176]
GoogleNet	1,566 M [46]	0.689 [176]
VggNet	15,480.10 M [159]	0.678 [176]

Table 4.6: **The Benchmark under Sensitivity Metric**

The Approach	Estimated Architectural Computational FLOPs	Sensitivity
Bayesian-Optimisation Framework (ours)	1.5 M	0.819
ARDT-DenseNet	2,872.13 M [159]	0.816 [176]
DenseNet-100	2,872.13 M [159]	0.812 [176]
ResNet-101	7,597.95 M [159]	0.804 [176]
ResNet-50	3,798.98 M [159]	0.799 [176]
GoogleNet	1,566 M [46]	0.770 [176]
VggNet	15,480.10 M [159]	0.768 [176]

4.3.2 The Applicability into Mobile Device

The research presented in the thesis needed to check whether the hypothesis supporting the method is correct. The research employs an android simulator with an integrated development environment to check the work’s applicability into mobile devices, namely Android studio. Android studio enhances the capability to use Java and Kotlin programming language to develop an Android phone application, supported with different mobile devices settings.

The research exports the trained model in the chapter into the TensorFlow-lite model. Then the research inferences the model to classify random data from the dataset. The result of the inference process with the simulator is shown in Figure 4.2 and 4.3. The setting for the Android’s environment was Android 5.0 (Lollipop) or higher version. The setting makes the application works on a broader scope of devices (to date, roughly applicable to 91 percent of Android devices, according to the Android studio’s statistic). The code in the research to convert the model and run inference in mobile devices is publicly available [127]. The final Android application was tested in a real device Samsung Galaxy S8, and it works as expected.

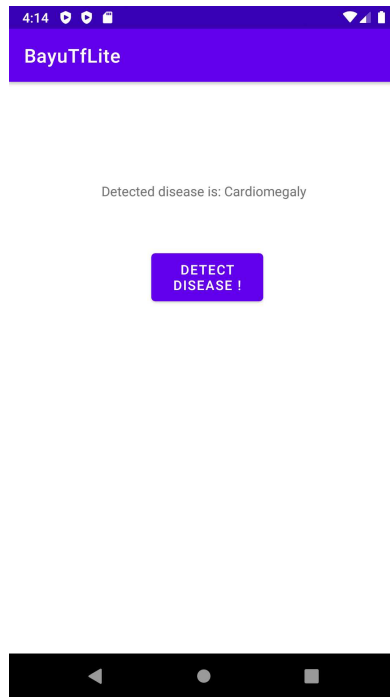


Figure 4.2: The Mobile App Detects Cardiomegaly

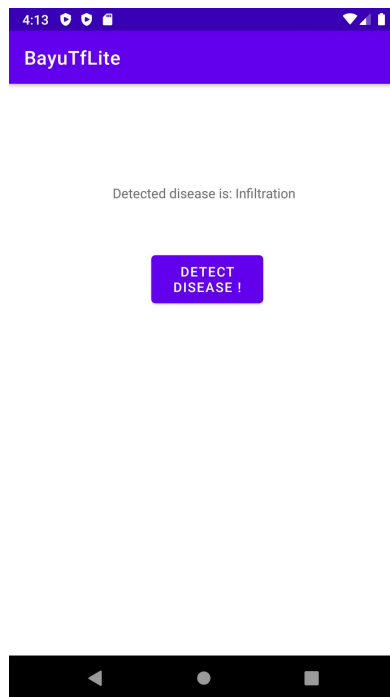


Figure 4.3: The Mobile App Detects Infiltration

4.4 Discussion

This study proposes a framework to produce a lightweight final classification model. It resulted in a final model with a comparable result classification with less computational resources. The prior work [62] has a total of 387,795M FLOPs from Equation 4.8. This study has a total of 8.89M FLOPs from Equation 4.10 of the proposed work.

Additional calculation of this study adds the extra 8,550M FLOPs during the extraction phase from EfficientNet-B3 [161], which makes the proposed final model require 8,558.89M FLOPs. In summary, the proposed model requires 45.30 times fewer FLOPs than [62], as shown in Table 4.2. Further, the proposed base architectures require more minor FLOPs than the works [26,64], as shown in Table 4.1.

Chapter 5

One-Class Classification

5.1 Introduction

The application of OCC is useful when (i) there is an imbalance in the dataset [58] or (ii) there is a case in which the end user “may only be interested in a specific class without considering other” [100]. In the context of medical images classification, a dataset can have multiple labels but the subject of interest for the classification might be the specific patterns.

A one-class classifier’s primary challenge is the lower classification performance achieved than in binary classification; this statement is supported in the work of Krawczyk et al. (2015) [96]: “Our initial assumption was, that OCC won’t be superior to binarization for all cases, and that was confirmed”. One-class classifiers also have a lower performance in comparison with multiclass classifiers [90] because of the lack of labelled training patterns for other classes in the training process.

Perera and Patel (2019) [133] introduced the concept of descriptiveness and the compactness. Descriptiveness is the measurement for differences of features from one pattern compared with other patterns (interclass). Compactness is the measurement for differences of features within samples inherently in one pattern (intraclass) [133]. As previously mentioned, the standard one-class training procedure has a low descriptiveness; the training only consists of one pattern. [133] proposed a novel training procedure with the appropriate loss function to support both descriptiveness and compactness to tackle this problem.

The motivation to propose this particular study was because of existing evidence of some classes with rare samples in the medical images dataset (e.g. the NIH Chest-X-Ray 14). Since the examples are rare, to form a good representation pattern for a binary classifier might be difficult [103]. One solution is to train the majority pattern in one classifier and re-present the minority as outliers [153]. However, in the dataset case (such as the NIH Chest-X-Ray 14), the majority pattern is the negative samples. This condition might pose an overfitting problem for specific classifiers [38], and training with only the target class (the minority pattern) might produce better predictions [103].

This study in Chapter 5 proposes a one-class classifier method that requires low computational resources and has competitive classification results compared to multiclass classification. The main difference with Chapter 4 is the additional inclusion of training examples and not only the number of epochs in calculating the required FLOPs. However, this study in Chapter 5 still uses the FLOPs base features-network listed in Table 4.1 and 4.2. The “template network” is also the identical network from Chapter 4. The progression of Chapter 5 is a more focused classification of the particular disease than the model in Chapter 4, also with the less computational cost for the final model.

5.2 Method

5.2.1 Research Contribution and Novelty Statement

This research’s primary contribution is that it proposes a dual-branch network architecture to train a one-class classifier without the presence of counterexamples. This study can show the advantage of the proposed method to achieve better results than previous studies [26, 62, 64]. It also requires fewer computational costs during training. Other works [67, 68] summarise the best classification performance from several traditional classifiers, which were trained separately. The proposed work summarises the best classification performance from a single neural network with dual branches classifiers trained together.

Naturally, a one-class classifier has a lower performance than other classifier types because of the lack of access to the counterexamples [90, 96]. However, for

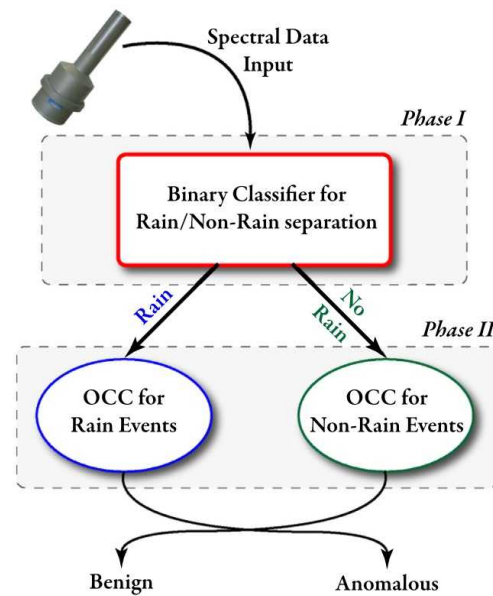


Figure 5.1: Two Tier Learning [29]

this research, the study has access to the counterexamples. However, this study opted not to use the counterexamples to show the advantages of the proposed approach.

5.2.2 The Existing One Class Classification

Bellinger et al. [29] proposed a sub-concept learning method, entitling the method as a “two-tiered multi classifier system”. The dataset [29] is a multilabel classification problem, consisting of rain-benign, rain-anomalous, non-rain-benign and non-rain-anomalous classes. Similarly, the NIH Chest X-Ray [173] dataset consists of 20,796 multilabel examples within the dataset; 18,5% of the total sum of examples.

The approach proposed by Bellinger et al. [29] is a two-phase training method. The first phase separates rain and non-rain with the binary classifier, then the two one-class classifiers perform the classification between benign and anomalous in the second stage. The method [29] is depicted in Figure 5.1. In the second phase, the training proceeds without counterexamples. Krawczyk et al. [96] investigated several one-class classifiers’ accuracy to decompose a multiclass classification sys-

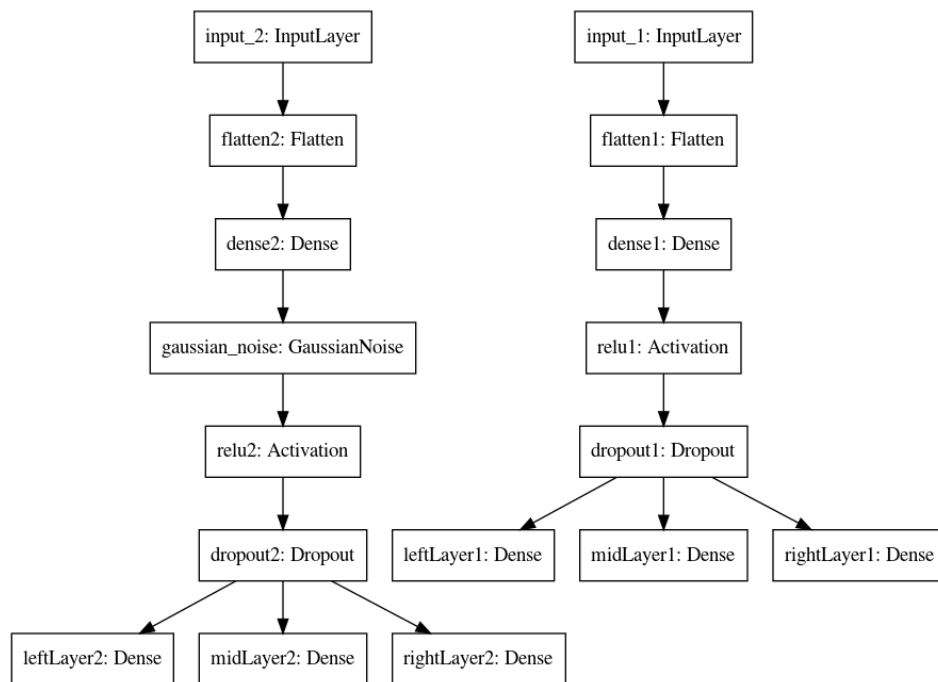


Figure 5.2: A Dual-Branch Network with Six Outputs

tem.

5.2.3 The Proposed Approach for One-Class Classification

Figure 5.2 depicts the proposed network. It is a single network with two main branches. Figure 5.2 is generated automatically from the Keras [41] library. This study used the features from the last global average-pool layer of an EfficientNet-B3 [41] backbone with the chest X-ray 14 dataset [173] to perform experiments. The experiments trained a one-class classifier for each of the 14 classes.

The “two-tiered multi classifier system” [29] did not explicitly address the concept of descriptiveness and compactness [133]. This study also used extracted features for the sub-concept learning process, whereas [29] used the original input images for every phase of the “two-tiered multi classifier system”. Similarly, [95] decomposed a multiclass classification problem from several OCC problems.

Since previous work [133] has emphasised the importance of descriptiveness and compactness in the OCC system. This study intends to undertake the same analysis. However, this study proposes a different approach to address descriptiveness

and compactness in the OCC problem.

This study proposes a network with dual branches; for every branch, this study applies three different loss functions. In the left sigmoid output, this study applied the Huber loss [35, 81, 108, 112]. In the middle output, this study applied focal loss [101], while in the right sigmoid output, this study applied an Euclidean distance loss [155]. All loss functions maximise only the compactness by using single-label training examples. In our understanding, the use of extracted features from the EfficientNet-B3 preserved the descriptiveness.

Equation 5.1 depicts the Huber loss with the threshold. Suppose p is the prediction. The Huber loss output will be the the mean squared error (MSE) for the norm of p smaller or equal than the threshold. It outputs the the average mean error (MAE) for the other value of p .

$$\mathfrak{L}_\delta(p) = \begin{cases} \frac{p^2}{2}, & \text{if } |p| \leq \delta, \\ \delta|p| - \frac{\delta^2}{2}, & \text{otherwise.} \end{cases} \quad (5.1)$$

Equation 5.2 depicts the focal-loss from literature [101].

$$\mathfrak{L}_{foc}(p) = -\alpha (1 - p)^\gamma y_{true}^k \log(p) \quad (5.2)$$

Equation 5.3 depicts the Euclidean distance loss, which measures the distance between the prediction and the ground-truth. Because the proposed work has six loss functions in total, the network also outputs six validation values for each training epoch.

$$\mathfrak{L}_{euclidean}(p) = \sqrt{\sum_{i=1}^n (q_i - y_{true}^i)^2} \quad (5.3)$$

The proposed algorithm uses the sum of those six validation outputs to tune the network's hyperparameters. The advantage of having six output loss functions in two branches is that this study can choose the best performance from six nodes' results.

5.2.4 The Best Outputs from Two Branches

Algorithm 2 The Highest Accuracy for Dual-Branch Network

input : The learning-rate hyperparameters

output: The highest accuracy for dual-branch network

start

initialisation: define grid of learning-rates for n epochs

while $iteration < (n \text{ epochs})$

optimise network with classification with the learning-rate inputs

capture the best classification between two branches

return: choose the best classification between two branches

end

Since the proposed work has six output terminal nodes, each trained model will have six accuracies that correspond to loss functions. In addition, the proposed method also chooses the result that outputs the highest accuracy.

5.2.5 The Fine-Grained Computational-Cost Evaluation

Suppose this study required a fine-grained calculation to factor the number of training data into the computational-cost evaluation. Then, this study can use the path of the classical research by Mizutani and Dreyfus (2001) [116]. The work [116] decomposed the backpropagation algorithm, then performed a fine-grained cost evaluation for each process from forward-pass and backwards-pass in the algorithm.

Table 5.1: **Decomposition of Backpropagation Algorithm** [116]

Backpropagation Algorithm	
Forward Pass	
Process 1	Node - input computation
Process 2	Activation or output evaluation
Process 3	Objective function evaluation
Backward Pass	
Process 4	Node sensitivity
Process 5	Gradient computation
Process 6	Parameter updates

The literature [116] defines the approximate FLOPs in Equation 5.4.

$$Approximate_{FLOPs} = \frac{cost(BackwardPass)}{cost(ForwardPass)} \quad (5.4)$$

where each of forward-pass and backwards-pass consists of three processes with independent FLOPs calculation. The decomposition is depicted in Table 5.1

Table 5.2: **The Notations** [116]

Notation	Description
d	Number of training examples
n	Number of weights parameters
N	Total number of layers, including the input layer
s	A particular layer
P_N	Number of output neurons at the terminal layer N
P_S	Number of neurons at the layer s , excluding the bias
T_s	Cost to evaluate activation node $f^s(\cdot)$
V_s	Cost to evaluate derivative $f^s'(\cdot)$

Suppose d is the number of training examples; the other notation is shown in Table 5.2. This study has the decomposition of backpropagation into six processes. Calculating the computing cost for each process is bound to the d training examples. Equations 5.4 and 5.5 are the mathematical perspectives from which to calculate the cost of a backpropagation algorithm [116].

$$\begin{aligned}
Process_1 &= 2d \sum_{s=2}^N (P_{s-1} + 1) P_s = 2dn & (5.5) \\
Process_2 &= d \sum_{s=2}^N T_s P_s \\
Process_3 &= dP_N + 2dP_N \\
Process_4 &= dP_N + d \sum_{s=1}^{N-1} P_{s+1} (V_{s+1} + 1) + 2d \sum_{s=2}^{N-1} P_s P_{s+1} \\
Process_5 &= 2dn \\
Process_6 &= 2n
\end{aligned}$$

Suppose this study wants to have a fine-grained FLOPs calculation for the common activation functions to determine T_s . Then, this study can analyse it using the mathematical equation from the corresponding functions. This study can evaluate from Equation 5.6 that ReLu only requires one FLOP (one operation), while sigmoid requires four FLOPs (four operations).

$$\begin{aligned}
ReLu(x) &= \max(0, x) & (5.6) \\
Sigmoid(x) &= \frac{1}{1 + e^{-x}}
\end{aligned}$$

This study can have a fine-grained FLOPs calculation for the derivation of sigmoid activation functions to determine V_s . This study can evaluate from Equation 5.7 that the derivation requires 20 FLOPs (20 operations).

$$\begin{aligned}
Sigmoid(x) &= \sigma(x) & (5.7) \\
\sigma'(x) &= \sigma(x) (1 - \sigma(x))
\end{aligned}$$

This study can determine from Equation 5.8 that the derivation of ReLu requires only one FLOP (one operation).

$$\text{ReLU}(x) = \sigma(x) \quad (5.8)$$

$$\sigma'(x) = \begin{cases} 0; & \text{for } x < 0 \\ 1; & \text{for } x > 0 \end{cases}$$

The derivation of ReLU is undefined for $x = 0$

It can also be concluded from Equation 5.5: suppose the study has two identical networks N_1 and N_2 which were trained with the different number of examples d_1 and d_2 . Then the study can formulate the computational cost required during N_1 training with the ratio of examples as shown in Equation 5.9.

$$\text{cost}(N_1) = \frac{d_1}{d_2} \cdot \text{cost}(N_2) \quad (5.9)$$

5.3 Experiment and Results

Table 5.3: Identical Splits Comparison with Previous Work

Pathology	Third party [139] of Guan et al. [62]			The Proposed One-Class Features Classification		
	AUROC	Cost of Examples FLOPs	Train Examples	AUROC	Cost of Examples FLOPs	Train Examples
Cardiomegaly	0.9097	304,295,765.28 M	78,468	0.9151	2,466.75 M	1,950
Emphysema	0.8905	304,295,765.28 M	78,468	0.9539	2,275.735 M	1,799
Edema	0.9185	304,295,765.28 M	78,468	0.9036	2,137.85 M	1,690
Hernia	0.9064	304,295,765.28 M	78,468	0.9434	182.16 M	144
Pneumothorax	0.8794	304,295,765.28 M	78,468	0.9100	4,686.825 M	3,705
Effusion	0.8843	304,295,765.28 M	78,468	0.8917	11,715.165 M	9,261
Mass	0.8707	304,295,765.28 M	78,468	0.8668	5,044.82 M	3,988
Fibrosis	0.8208	304,295,765.28 M	78,468	0.8651	1,464.87 M	1,158
Atelectasis	0.8225	304,295,765.28 M	78,468	0.8398	10,114.94 M	7,996
Consolidation	0.8210	304,295,765.28 M	78,468	0.8196	4,127.695 M	3,263
Pleural thicken.	0.8127	304,295,765.28 M	78,468	0.8135	2,882.935 M	2,279
Nodule	0.7691	304,295,765.28 M	78,468	0.8297	5,534.375 M	4,375
Pneumonia	0.7614	304,295,765.28 M	78,468	0.7694	1,237.17 M	978
Infiltration	0.7006	304,295,765.28 M	78,468	0.7352	17,601.21 M	13,914

The results in this section are under assumptions that the use of batch size equals one. This study performed the experiments to support the preliminary

hypothesis: that the proposed approach can achieve an acceptable classification performance for most training patterns, with fewer training examples and smaller computational cost.

This study alters each pattern’s training and validation set using **only** positive examples, and did not alter anything about the test set. It is precisely an identical normal test set. Hence, this study can have a fair comparison with the multiclass classifier. The study also has a preliminary hypothesis that negative examples are not required since the experiment uses extracted features rather than original images. Each feature is embedded with negative examples since the features are extracted from a trained network. This behaviour would not be available for original images input.

In term of the correlation of the AUROC value with the capability of the expert system itself, the literature [105] defines: “In general, an AUC of 0.5 suggests no discrimination (i.e., ability to diagnose patients with and without the disease or condition based on the test), 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding” [105]. According to the literature [43, 69, 105], the baseline 0.5 is generally known for AUROC metric. The comparison of the proposed approach’s results with the previous works [26, 62, 64] are depicted in Tables 5.3 and 5.4.

Table 5.4: **The Official Splits Comparison with Previous Works**

Pathology	ResNet-50 Baltruschat et al. [26]		Variant of DenseNet-121 Gündel et al. [64]			The Proposed One-Class Features Classification		
	AUROC	Cost of Examples FLOPs	AUROC	Cost of Examples FLOPs	Train Examples	AUROC	Cost of Examples FLOPs	Train Examples
Cardiomegaly	0.875	335,536,611.04 M	0.883	253,181,337.36 M	86,524	0.8933	2,159.355 M	1,707
Emphysema	0.895	335,536,611.04 M	0.895	253,181,337.36 M	86,524	0.9410	1,800.095 M	1,423
Edema	0.846	335,536,611.04 M	0.835	253,181,337.36 M	86,524	0.8604	1,743.17 M	1,378
Hernia	0.937	335,536,611.04 M	0.896	253,181,337.36 M	86,524	0.9253	178.365 M	141
Pneumothorax	0.819	335,536,611.04 M	0.846	253,181,337.36 M	86,524	0.8926	3,335.805 M	2,637
Effusion	0.822	335,536,611.04 M	0.806	253,181,337.36 M	86,524	0.8406	10,953.635 M	8,659
Mass	0.820	335,536,611.04 M	0.821	253,181,337.36 M	86,524	0.8365	5,103.01 M	4,034
Fibrosis	0.816	335,536,611.04 M	0.818	253,181,337.36 M	86,524	0.8375	1,582.515 M	1,251
Atelectasis	0.763	335,536,611.04 M	0.767	253,181,337.36 M	86,524	0.7904	10,474.2 M	8,280
Consolidation	0.749	335,536,611.04 M	0.745	253,181,337.36 M	86,524	0.7616	3,607.78 M	2,852
Pleural thicken.	0.763	335,536,611.04 M	0.761	253,181,337.36 M	86,524	0.8080	2,836.13 M	2,242
Nodule	0.747	335,536,611.04 M	0.758	253,181,337.36 M	86,524	0.7984	5,955.62 M	4,708
Pneumonia	0.714	335,536,611.04 M	0.731	253,181,337.36 M	86,524	0.7350	1,108.14 M	876
Infiltration	0.694	335,536,611.04 M	0.709	253,181,337.36 M	86,524	0.7058	17,434.23 M	13,782

The results in Table 5.3 show that the proposed method has the advantages of lower FLOPs; thus, it requires smaller training examples. A primary difference between the proposed work and [62, 139] is that the proposed approach requires 14 distinct classifiers. In contrast, [62, 139] only requires one classifier. This study of OCC did not perform averages from the results in Table 5.4 because it has an independent classifier for each class. Comparison of this study’s one-class classifiers results with its multiclass classifiers is depicted in Tables 5.5 and 5.6.

This study requires slightly more total FLOPs since it used the multiclass classifiers’ pre-trained weights to obtain more optimal results. However, this effort increased the required FLOPs because this study needed the multiclass’ FLOPs to be considered.

Table 5.5: **Identical Splits Comparison with Multiclass Classification**

Pathology	The Chapter 4 Multiclass Features Classification				The Proposed One-Class Features Classification			
	AUROC without Noise	AUROC with Noise	Cost of Examples FLOPs	Train Examples	AUROC	Cost of Examples FLOPs	Train Examples	Total FLOPs
Cardiomegaly	0.9136	0.9091	99,654.36 M	78,468	0.9151	2,466.75 M	1,950	126,788 M
Emphysema	0.9538	0.9506	99,654.36 M	78,468	0.9539	2,275.735 M	1,799	124,687.445 M
Edema	0.9044	0.9018	99,654.36 M	78,468	0.9036	2,137.85 M	1,690	123,170.71 M
Hernia	0.9512	0.9384	99,654.36 M	78,468	0.9434	182.16 M	144	101,658.12 M
Pneumothorax	0.9100	0.9106	99,654.36 M	78,468	0.9100	4,686.825 M	3,705	151,209.435 M
Effusion	0.8915	0.8904	99,654.36 M	78,468	0.8917	11,715.165 M	9,261	228,521.175 M
Mass	0.8662	0.8651	99,654.36 M	78,468	0.8668	5,044.82 M	3,988	155,147.38 M
Fibrosis	0.8630	0.8583	99,654.36 M	78,468	0.8651	1,464.87 M	1,158	1115,767.93 M
Atelectasis	0.8393	0.8393	99,654.36 M	78,468	0.8398	10,114.94 M	7,996	210,918.7 M
Consolidation	0.8196	0.8176	99,654.36 M	78,468	0.8196	4,127.695 M	3,263	145,059.005 M
Pleural thicken.	0.8094	0.8050	99,654.36 M	78,468	0.8135	2,882.935 M	2,279	131,366.645 M
Nodule	0.8280	0.8302	99,654.36 M	78,468	0.8297	5,534.375 M	4,375	160,532.485 M
Pneumonia	0.7676	0.7665	99,654.36 M	78,468	0.7694	1,237.17 M	978	113,263.23 M
Infiltration	0.7343	0.7347	99,654.36 M	78,468	0.7352	17,601.21 M	13,914	293,267.67 M

This study used Equation 5.9 to calculate the effect of the number of training examples and Equation 4.10 to adjust the epochs’ number into the FLOPs calculation. The use of the number of training examples and the required epoch numbers are taken into account in the FLOPs calculation did not appear in Chapter 4, such as the results in Table 4.2. However, the other benchmarked research presented in Chapter 4 also did not take the number of training examples, and the required epoch numbers are taken into account into the FLOPs calculation

Table 5.6: The Official Splits Comparison with Multiclass Classification

Pathology	The Chapter 4 Multiclass Features Classification				The Proposed One-Class Features Classification			
	AUROC without Noise	AUROC with Noise	Cost of Examples FLOPs	Train Examples	AUROC	Cost of Examples FLOPs	Train Examples	Total FLOPs
Cardiomegaly	0.8923	0.8919	109,542.58 M	86,524	0.8933	2,159.355 M	1,707	133,295.485 M
Emphysema	0.9408	0.9399	109,542.58 M	86,524	0.9410	1,800.095 M	1,423	129,343.625 M
Edema	0.8600	0.8604	109,542.58 M	86,524	0.8579	1,743.17 M	1,378	128,717.45 M
Hernia	0.9278	0.9204	109,542.58 M	86,524	0.9253	178.365 M	141	111,504.595 M
Pneumothorax	0.8902	0.8912	109,542.58 M	86,524	0.8926	3,335.805 M	2,637	146,236.435 M
Effusion	0.8389	0.8394	109,542.58 M	86,524	0.8406	10,953.635 M	8,659	230,032.565 M
Mass	0.8344	0.8357	109,542.58 M	86,524	0.8365	5,103.01 M	4,034	165,675.69 M
Fibrosis	0.8417	0.8423	109,542.58 M	86,524	0.8375	1,582.515 M	1,251	126,950.245 M
Atelectasis	0.7909	0.7907	109,542.58 M	86,524	0.7904	10,474.2 M	8,280	224,758.78 M
Consolidation	0.7600	0.7616	109,542.58 M	86,524	0.7608	3,607.78 M	2,852	149,228.16 M
Pleural thicken.	0.8074	0.8072	109,542.58 M	86,524	0.8080	2,836.13 M	2,242	140,740.01 M
Nodule	0.7979	0.7978	109,542.58 M	86,524	0.7984	5,955.62 M	4,708	175,054.4 M
Pneumonia	0.7341	0.7330	109,542.58 M	86,524	0.7350	1,108.14 M	876	121,732.12 M
Infiltration	0.7046	0.7033	109,542.58 M	86,524	0.7058	17,434.23 M	13,782	301,319.11 M

The total FLOPs columns in Tables 5.5 and 5.6 are the results from the multiclass FLOPs required to obtain the pre-trained weights added with the FLOPs of the number of epochs and training examples in the calculation. In detail, the number of epochs is 11, multiplied by the FLOPs listed in the cost of the examples column.

5.3.1 Effectiveness of the pre-Trained Weights

This study provides other results to reach a conclusion about the effectiveness of the pre-trained weights. Tables 5.8 and 5.7 depict this study’s results in identical networks using random initialization versus pre-trained weights. Using the results depicted in Table 5.8 and 5.7, this study can conclude that the pre-trained weights significantly help the network increase classification performance.

Table 5.7: **Identical Splits Comparison Between from Scratch and Pre-Trained Initialisation**

Pathology	AUROC	
	from scratch	pre-trained
Cardiomegaly	0.8574	0.9151
Emphysema	0.8607	0.9539
Edema	0.8162	0.9036
Hernia	0.8500	0.9434
Pneumothorax	0.7771	0.9100
Effusion	0.8031	0.8917
Mass	0.7565	0.8668
Fibrosis	0.7085	0.8651
Atelectasis	0.6824	0.8398
Consolidation	0.7182	0.8196
Pleural thicken.	0.6462	0.8135
Nodule	0.7027	0.8297
Pneumonia	0.6195	0.7694
Infiltration	0.5982	0.7352

Table 5.8: **The Official Splits Comparison Between from Scratch and Pre-trained Initialisation**

Pathology	AUROC	
	from scratch	pre-trained
Cardiomegaly	0.7776	0.8933
Emphysema	0.8808	0.9410
Edema	0.8126	0.8579
Hernia	0.8738	0.9253
Pneumothorax	0.8316	0.8926
Effusion	0.7429	0.8406
Mass	0.7136	0.8365
Fibrosis	0.7555	0.8375
Atelectasis	0.6332	0.7904
Consolidation	0.6971	0.7608
Pleural thicken.	0.6606	0.8080
Nodule	0.6325	0.7984
Pneumonia	0.6737	0.7350
Infiltration	0.6208	0.7058

5.4 Discussion

The proposed one-class classifier approach provides competitive results compared with traditional multiclass classifiers, which were trained under fully labelled set-

tings. However, the study of more reliable network architecture variations to perform OCC requires further investigation. These experiments show that another essential factor other than the network architecture is the weight initialisation. Similarly, the weight initialisation method for the top layers to address the classification task has been done previously [53, 54]. The proposed work's primary difference is in the previous works [53, 54] use of an autoencoder for the backbone network. The proposed work used a dual-branch network. The previous works [53, 54] did not perform one-class training to address the multiclass classification problem.

The early idea was to extend the descriptiveness and the compactness concepts from [133]. Using one-class training is part of the proposed method to maximise and stretch the compactness without ignoring the descriptiveness. The single pattern used during training will converge into the smaller training loss; this means a smaller distance to the ground truth for that single pattern.

However, since this study's task was to address the multiclass classification problem, it aimed also to have good descriptiveness so that the classifier had an acceptable decision boundary to distinguish the other patterns that were not trained during the one-class pattern training. In this case, this study can summarise that to maintain the descriptiveness, there are two essential factors.

The first is the features being used. this study uses the extracted features from the previously multiclass classification problem, which means the features have preserved some descriptiveness. The second is the weight initialisation; a proper weight initialisation significantly improves classification performance.

The thesis has two significant differences from [133], the first is that this thesis use a decision-based algorithm to determine best outputs from six loss functions, whilst [133] used a customized join loss-function. The second is that this thesis used a customized network, whilst [133] use AlexNet and VGG16.

Chapter 6

Conclusion

This chapter discusses the conclusions of the work in the thesis. Mainly, it summarises the contributions from the work and the future works that have potential to explore and provide the sustainability of the current results.

6.1 Summary of Contributions

Generally, the thesis has three significant contributions. First, it proposes an approach that can combine weights calculation algorithms for deep networks and optimise training strategies from the state-of-the-art architecture. The findings show that the proposed method can significantly improve classification performance. In the initial research phase, the research used AUROC as the primary metric to evaluate the result, mainly because of the similarity with other works. However, one reviewer suggested different metrics to show the work's advantages during the manuscript submission review stages. The research used AU-PRC to fulfil the request. This metric results exceptionally well on the hernia as the most minority class.

This research also verifies the method with a private binary classification case dataset—the glaucoma dataset—as discussed in Section 3.4.4. The glaucoma dataset is available by contacting the previous scholarly work [91] listed in Table 2.7. After verification, this study can conclude that the method performs well on multiclass classification and binary classification problems. The research also shares the chest X-ray classification code in the public repository. In summary,

(1) This research publishes the method with a public dataset in the refereed journal [128]. (2) The research verifies the method with another private dataset. (3) The research shares the code of the public dataset for reproducibility. One important aspect is that this study was able to benchmark the results with different experiment settings, and those results were satisfactory. Hence, it was accepted in a top-tier publication [128]. In summary, the method can achieve state-of-the-art results for the chest X-ray dataset by improving the imbalance problem.

Second, the research proposes two-stage Bayesian optimisation training to perform competitive classification performance using minor FLOPs neural network architecture. The findings show that the proposed method contributes and suits both extracted features input and original images input. This study aims to have a light multiclass classifier suitable for use in low-cost computing environments. The findings show that the method can achieve classification performance competitively with a significant advantage of the smaller computing resources required. This study used FLOPs and MACCs units of measurement to assess the required computational cost. One primary reason is that those two units have been used in the notable works [71, 80, 161, 162]. This research will help implement the chips (SoC) system or the IoT platform under von Neumann architecture. In summary, the method can achieve competitive performances with the bit of computational cost required.

Third, the research proposes a dual-branch network architecture to train one-class classifiers without counterexamples. The method is proper to use when interested in a particular disease pattern rather than all the available patterns in the dataset. The evidence indicates that the proposed one-class method has competitive results compared with regular multiclass classifiers trained in fully labelled settings. There are cases for which the end user needs to diagnose only the diseases. Despite the competitive classification results for most diseases, performing multiclass analysis may not be time-efficient or necessary. The findings show that OCC with the extracted features achieved the best performance when the neural network was initialised with pre-trained weights rather than the random initialisation. In summary, the method can achieve competitive performances without counterexamples during training.

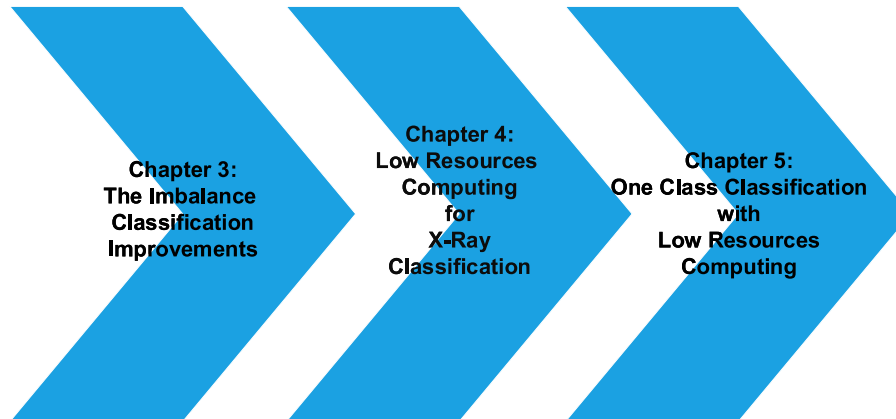


Figure 6.1: The Chapters' Progressions

6.2 The Progressions from Chapters

Figure 6.1 depicts the research's progression for each chapter. In Chapter 3, the research has successfully presented the method to improve classification performance. The method generalises well from the Chest X-Ray dataset to the Glaucoma dataset. However, the final model was still quite computationally expensive to run on low-cost computing devices. In Chapter 4, the research evolves into feature-based classification, mainly to reduce the computational cost. The method also generalises well from the Chest X-Ray dataset into the Skin cancer dataset. The model has been tested on the Android simulator and an actual Android mobile phone. The test results show that it can classify diseases well on low-computing platforms. In Chapter 5, the research focuses on single-class classifiers for each disease. The outcome for this research is the advantage of low-cost computing capability, but with the superior performance for each particular disease, compared to the features classification in Chapter 4.

6.3 Future Works

Several aspects may improve the work in the thesis, which are currently beyond our time budget. This study suggests future works involving process automation, the multimodality of the data sources and the real-time expert system.

Prominent future work is automating the whole process into one integrated system. Each of our approaches currently works partially as different parts of the system. A future work that incorporates every process into an integrated modular backbone will provide better applicability of the system. The implementation is challenging because it may integrate large GPU clusters and SoC or the IoT technology.

This research also encourages the study of medical images classification using the multimodal dataset; the term multimodal refers to using different data sources (e.g. images and text). The existing research has tended to use the dataset in a single type of data-source fashion (e.g. fully labelled images setting). Further text analysis into medical images dataset to improve the classification performance is a promising research direction. The text-based dataset requires a smaller computational cost than does an image-based dataset. The simple reasoning for this is that the text-based machine representation is the word embeddings composed of binary numbers (zero and one). Conversely, the image-based machine representation is floating numbers that require more computational cost in the arithmetic logic unit. Hence, the extensive study in the direction will also extend the potentials to reduce the overall training cost for an image classification algorithm.

The other aspect that needs further discussion is the real-time expert system. This researcher was aware that the machine-learning approaches require the training process to transfer knowledge from the expert into the artificial intelligence system. However, the training process itself is a bottleneck, especially when considering the consuming time to train the enormous image training data in the deep-learning platform. A more simplified architecture to artificial intelligence (AI) is required to update its knowledge in a real-time fashion. Hence, we can have a more responsive system to the actual condition in the field of healthcare services.

Appendix A

The Code Listing

The code from the proposed algorithms are listed below.

The code snippets from the file: createOneClassTrainLabel.ipynb. The modification of the training labels into a single training label for each class is depicted in Figure A.1.

```
In [5]: import pandas as pd
c = len(train_onlyCols)
for i in range(c):
    cleanIndices = []
    name = train_onlyCols[i]
    myStr = "select * from train_only where train_only."+name+" == 1"
    print(myStr)
    theIndices = pysql(myStr)
    theIndices.to_csv("oneClasslabel/"+name+".csv", index=False, sep=',')
```



```
select * from train_only where train_only.Infiltration == 1
select * from train_only where train_only.Mass == 1
select * from train_only where train_only.Nodule == 1
select * from train_only where train_only.Pneumonia == 1
select * from train_only where train_only.Pneumothorax == 1
select * from train_only where train_only.Consolidation == 1
select * from train_only where train_only.Edema == 1
select * from train_only where train_only.Emphysema == 1
select * from train_only where train_only.Fibrosis == 1
select * from train_only where train_only.Pleural_Thickening == 1
select * from train_only where train_only.Hernia == 1
```

Figure A.1: Training Label Modification for One Class

The code snippets from the file: util.py. The modification of the weighted-expected-improvement acquisition function is depicted in Figure A.2.


```

@staticmethod
def _weightedei(x, gp, y_max, xi, omega):
    with warnings.catch_warnings():
        warnings.simplefilter("ignore")
        mean, std = gp.predict(x, return_std=True)

    z = (mean - y_max - xi)/std
    return omega*(mean - y_max - xi) * norm.cdf(z) + (1 - omega)*std * norm.pdf(z)

```

Figure A.2: Weighted EI Acquisition Function

The code snippets from the file: calculateModel.ipynb. The calculation of a model is depicted in Figure A.3.

```

In [4]: model.summary()

```

Layer (type)	Output Shape	Param #
flat1 (Flatten)	(None, 1536)	0
dense1 (Dense)	(None, 410)	630170
relu1 (Activation)	(None, 410)	0
dropout1 (Dropout)	(None, 410)	0
OUTPUT (Dense)	(None, 14)	5754

```

Total params: 635,924
Trainable params: 635,924
Non-trainable params: 0

```

```

In [5]: from keras_flops import get_flops
flops = get_flops(model, batch_size=1)

```

```

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\de_def_name (from tensorflow.python.framework.graph_util_impl) is deprecated
Instructions for updating:
Use `tf.compat.v1.graph_util.tensor_shape_from_node_def_name`

```

```

In [6]: print(f"FLOPS: {flops / 10 ** 6:.03} M")

```

```

FLOPS: 1.27 M

```

Figure A.3: Calculate FLOPs from The Model

Appendix B

The List of Repositories

The list of public repositories which were used.

- <https://github.com/fchollet/keras>
- <https://github.com/brucechou1983/chexnet-keras>
- <https://github.com/ien001/ag-cnn>
- <https://github.com/arnoweng/chexnet>
- The code for Chapter 3: <https://github.com/bayu-ladom-ipok/weOpen>
- https://github.com/AMLab-Amsterdam/L0_regularisation
- <https://github.com/fmfn/bayesianoptimization>
- https://github.com/jeffheaton/t81_558_deep_learning/
- <https://github.com/tokusumi/keras-flops>
- The code for Chapter 4: <https://github.com/bayu-ladom-ipok/weOpenBayesianOpt>
- The code for Chapter 5: <https://github.com/bayu-ladom-ipok/weOpenOneClass>

Bibliography

- [1] Atelectasis - <https://www.nhlbi.nih.gov/health-topics/atelectasis>.
- [2] Causes of death, australia, 2018 — australian bureau of statistics. <https://www.abs.gov.au/statistics/health/causes-death/causes-death-australia/2018>. (Accessed on 12/06/2021).
- [3] Chronic obstructive pulmonary disease (copd), impact - australian institute of health and welfare. <https://www.aihw.gov.au/reports/asthma-other-chronic-respiratory-conditions/copd-chronic-obstructive-pulmonary-disease/contents/deaths>. (Accessed on 12/06/2021).
- [4] Chronic obstructive pulmonary disease (copd), impact - australian institute of health and welfare. <https://www.aihw.gov.au/reports/chronic-respiratory-conditions/copd/contents/impact>. (Accessed on 12/06/2021).
- [5] Chronic respiratory conditions - australian institute of health and welfare. <https://www.aihw.gov.au/reports/australias-health/chronic-respiratory-conditions>. (Accessed on 12/06/2021).
- [6] Chronic respiratory conditions - <https://www.aihw.gov.au/reports/australias-health/chronic-respiratory-conditions>.
- [7] Chronic respiratory diseases in australia: their prevalence, consequences and prevention, summary - australian institute of health and welfare. <https://www.aihw.gov.au/reports/chronic-respiratory-conditions/>

- [chronic-respiratory-diseases-australia/contents/summary](#). (Accessed on 12/06/2021).
- [8] Diabetic retinopathy detection — kaggle. <https://www.kaggle.com/c/diabetic-retinopathy-detection>. (Accessed on 12/10/2021).
- [9] Emphysema - <https://www.healthdirect.gov.au/emphysema>.
- [10] Endotect 2020. <https://endotect.com/>. (Accessed on 12/10/2021).
- [11] Isic challenge. <https://challenge.isic-archive.com/landing/2016/39/>. (Accessed on 12/01/2021).
- [12] Pleural effusion - <https://www.healthdirect.gov.au/pleural-effusion>.
- [13] Pneumonia - <https://www.healthdirect.gov.au/pneumonia>.
- [14] Pneumonia - <https://www.nhlbi.nih.gov/health-topics/pneumonia>.
- [15] Pulmonary edema: Medlineplus medical encyclopedia - <https://medlineplus.gov/ency/article/000140.htm>.
- [16] The global economic burden of asthma and chronic obstructive pulmonary disease. *The International Journal of Tuberculosis and Lung Disease*, 20(1):11–23, 2016.
- [17] Collapsed lung — atelectasis — pneumothorax - <https://medlineplus.gov/collapsedlung.html>, Oct 2020.
- [18] The lungs - <https://www.health.qld.gov.au/news-events/podcast/my-amazing-body-the-lungs>, Oct 2020.
- [19] Faststats - asthma - <https://www.cdc.gov/nchs/fastats/asthma.htm>, Apr 2021.
- [20] Lung disease: Medlineplus medical encyclopedia - <https://medlineplus.gov/ency/article/000066.htm>, Feb 2021.

- [21] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.
- [22] Hina Amin. Cardiomegaly - <https://www.ncbi.nlm.nih.gov/books/nbk542296/>, Nov 2020.
- [23] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.
- [24] Jay E. Aronson. Expert systems. In Hossein Bidgoli, editor, *Encyclopedia of Information Systems*, pages 277–289. Elsevier, New York, 2003.
- [25] Peter J. Ashenden. Case study: A pipelined multiplier accumulator. In *The Designer's Guide to VHDL*, pages 337–354. Elsevier, 2008.
- [26] Ivo M. Baltruschat, Hannes Nickisch, Michael Grass, Tobias Knopp, and Axel Saalbach. Comparison of deep learning approaches for multi-label chest x-ray classification. *Scientific Reports*, 9(1), April 2019.
- [27] Gopi Battineni, Nalini Chintalapudi, and Francesco Amenta. Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (svm). *Informatics in Medicine Unlocked*, 16:100200, 2019.
- [28] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.
- [29] Colin Bellinger, Shiven Sharma, and Nathalie Japkowicz. One-class classification – from theory to practice: A case-study in radioactive threat detection. *Expert Systems with Applications*, 108:223 – 232, 2018.

- [30] Eta S. Berner and Mark L. Graber. Overconfidence as a cause of diagnostic error in medicine. *The American Journal of Medicine*, 121(5):S2–S23, May 2008.
- [31] Siddharth Bhatia, Yash Sinha, and Lavika Goel. Lung cancer detection: A deep learning approach. In Jagdish Chand Bansal, Kedar Nath Das, Atulya Nagar, Kusum Deep, and Akshay Kumar Ojha, editors, *Soft Computing for Problem Solving - SocProS 2017, Volume 2, Bhubaneswar, India, December 23-24, 2017*, volume 817 of *Advances in Intelligent Systems and Computing*, pages 699–705. Springer, 2017.
- [32] I. K. Bindoff, P. C. Tenni, G. M. Peterson, B. H. Kang, and S. L. Jackson. Development of an intelligent decision support system for medication review. *Journal of Clinical Pharmacy and Therapeutics*, 32(1):81–88, February 2007.
- [33] Titus J. Brinker, Achim Hekler, Alexander H. Enk, and Christof von Kalle. Enhanced classifier training to improve precision of a convolutional neural network to identify images of skin lesions. *PLOS ONE*, 14(6):e0218713, June 2019.
- [34] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 1565–1576, 2019.
- [35] Jacopo Cavazza and Vittorio Murino. Active regression with adaptive huber loss. *CoRR*, abs/1606.01568, 2016.
- [36] The U.S. Census Bureau, Apr 2021.
- [37] Abhishek Chaturvedi, Prabhakar Rajiah, Alexander Croake, Sachin Saboo, and Apeksha Chaturvedi. Imaging of thoracic hernias: types and complications. *Insights into Imaging*, 9(6):989–1005, November 2018.

- [38] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor.*, 6(1):1–6, 2004.
- [39] Ching-Chin Chern, Yu-Jen Chen, and Bo Hsiao. Decision tree-based classifier in providing telehealth service. *BMC Medical Informatics and Decision Making*, 19(1), May 2019.
- [40] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), January 2020.
- [41] Francois Chollet et al. Keras - <https://github.com/fchollet/keras>, 2015.
- [42] Bruce Chou. This project is a tool to build chexnet-like models, written in keras:<https://github.com/brucechou1983/chexnet-keras>, Mar 2018.
- [43] Corinna Cortes and Mehryar Mohri. AUC optimization vs. error rate minimization. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 313–320. MIT Press, 2003.
- [44] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9268–9277, 2019.
- [45] Xueqing Deng, Wenkai Li, Xiaoping Liu, Qinghua Guo, and Shawn Newsam. One-class remote sensing classification: one-class vs. binary classifiers. *International Journal of Remote Sensing*, 39(6):1890–1910, 2018.
- [46] Xuanyi Dong, Junshi Huang, Yi Yang, and Shuicheng Yan. More is less: A more complicated network with less inference complexity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5840–5848, 2017.

- [47] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification, 2nd Edition*. Wiley, 2001.
- [48] Hoda Ahmed Galal Elsayed and Liyakathunisa Syed. An automatic early risk classification of hard coronary heart diseases using framingham scoring model. In Hani Hamdan, Djallel Eddine Boubiche, Homero Toral-Cruz, Sedat Akleylek, and Hamid Mcheick, editors, *Proceedings of the Second International Conference on Internet of things and Cloud Computing, ICC 2017, Cambridge, United Kingdom, March 22-23, 2017*, pages 141:1–141:8. ACM, 2017.
- [49] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.
- [50] Xin Feng, Youni Jiang, Xuejiao Yang, Ming Du, and Xin Li. Computer vision algorithms and hardware implementations: A survey. *Integration*, 69:309–320, 2019.
- [51] Zhiwei Feng, Qingbin Zhang, Qingfu Zhang, Qiangang Tang, Tao Yang, and Yang Ma. A multiobjective optimization based framework to balance the global exploration and local exploitation in expensive optimization. *J. Glob. Optim.*, 61(4):677–694, 2015.
- [52] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from Imbalanced Data Sets*. Springer International Publishing, 2018.
- [53] M. F. Ferreira, R. Camacho, and L. F. Teixeira. Autoencoders as weight initialization of deep classification networks applied to papillary thyroid carcinoma. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 629–632, 2018.
- [54] Mafalda Falcão Ferreira, Rui Camacho, and Luís F. Teixeira. Using autoencoders as a weight initialization method on deep neural networks for disease

- detection. *BMC Medical Informatics and Decision Making*, 20(S5), August 2020.
- [55] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
- [56] Adrian Galdran, Gustavo Carneiro, and Miguel Ángel González Ballester. Balanced-mixup for highly imbalanced medical image classification. In Marleen de Bruijne, Philippe C. Cattin, Stéphane Cotin, Nicolas Padoy, Stefanie Speidel, Yefeng Zheng, and Caroline Essert, editors, *Medical Image Computing and Computer Assisted Intervention - MICCAI 2021 - 24th International Conference, Strasbourg, France, September 27 - October 1, 2021, Proceedings, Part V*, volume 12905 of *Lecture Notes in Computer Science*, pages 323–333. Springer, 2021.
- [57] Inés M. Galván, Josà M. Valls, Nicolas Lecomte, and Pedro Isasi. A lazy approach for machine learning algorithms. In *IFIP Advances in Information and Communication Technology*, pages 517–522. Springer US, 2009.
- [58] Long Gao, Lu Yang, Dooman Arefan, and Shandong Wu. One-class classification for highly imbalanced medical image data. In Po-Hao Chen and Thomas M. Deserno, editors, *Medical Imaging 2020: Imaging Informatics for Healthcare, Research, and Applications*, volume 11318, pages 342 – 347. International Society for Optics and Photonics, SPIE, 2020.
- [59] Long Gao, Lei Zhang, Chang Liu, and Shandong Wu. Handling imbalanced medical image data: A deep-learning-based one-class classification approach. *Artificial Intelligence in Medicine*, 108:101935, 2020.
- [60] Nicolás García-Pedrajas and Domingo Ortiz-Boyer. Improving multiclass pattern recognition by the combination of two strategies. *IEEE transactions on pattern analysis and machine intelligence*, 28:1001–6, 07 2006.
- [61] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

- [62] Qingji Guan, Yaping Huang, Zhun Zhong, Zhedong Zheng, Liang Zheng, and Yi Yang. Thorax disease classification with attention guided convolutional neural network. *Pattern Recognition Letters*, 131:38 – 45, 2020.
- [63] O. Guehairia, A. Ouamane, F. Dornaika, and A. Taleb-Ahmed. Feature fusion via deep random forest for facial age estimation. *Neural Networks*, 130:238–252, October 2020.
- [64] Sebastian Gündel, Sasa Grbic, Bogdan Georgescu, Siqi Liu, Andreas Maier, and Dorin Comaniciu. Learning to recognize abnormalities in chest x-rays with location-aware dense networks. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 23rd Iberoamerican Congress, CIARP 2018, Madrid, Spain, November 19-22, 2018, Proceedings*, pages 757–765, 2018.
- [65] David A. Gutman, Noel C. F. Codella, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Nabin K. Mishra, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC). *CoRR*, abs/1605.01397, 2016.
- [66] Huong Ha, Santu Rana, Sunil Gupta, Thanh Nguyen, Hung Tran-The, and Svetha Venkatesh. Bayesian optimization with unknown search space. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 11795–11804. Curran Associates, Inc., 2019.
- [67] Bilal Hadjadji, Youcef Chibani, and Yasmine Guerbai. Multiple one-class classifier combination for multi-class classification. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 2832–2837. IEEE Computer Society, 2014.
- [68] Bilal Hadjadji, Youcef Chibani, and Yasmine Guerbai. Combining diverse one-class classifiers by means of dynamic weighted average for multi-class pattern classification. *Intell. Data Anal.*, 21(3):515–535, 2017.

- [69] J A Hanley and B J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, April 1982.
- [70] hazibzunair. adversarial-lesions/data_preprocess_isic2016.py at master · hasibzunair/adversarial-lesions · github. https://github.com/hasibzunair/adversarial-lesions/blob/master/isic2016_scripts/data_preprocess_isic2016.py, March 2020. (Accessed on 01/05/2022).
- [71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.
- [72] Jeff Heaton. Applications of deep neural networks, 2020.
- [73] Juerg Hodler, Rahel A. Kubik-Huch, and Gustav K. von Schulthess, editors. *Diseases of the Chest, Breast, Heart and Vessels 2019-2022*. Springer International Publishing, 2019.
- [74] Syed Mohammad Minhaz Hossain, Kaushik Deb, Pranab Kumar Dhar, and Takeshi Koshiba. Plant leaf disease recognition using depth-wise separable convolution-based models. *Symmetry*, 13(3), 2021.
- [75] Fujun Hou and Evangelos Triantaphyllou. An iterative approach for achieving consensus when ranking a finite set of alternatives by a group of experts. *Eur. J. Oper. Res.*, 275(2):570–579, 2019.
- [76] A. Howard, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le. Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society.
- [77] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

- [78] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [79] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions, 2018.
- [80] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269. IEEE Computer Society, 2017.
- [81] Peter J. Huber. Robust estimation of a location parameter. In *Springer Series in Statistics*, pages 492–518. Springer New York, 1992.
- [82] S. S. Jaipurkar, W. Jie, Z. Zeng, T. S. Gee, B. Veeravalli, and M. Chua. Automated classification using end-to-end deep learning. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 706–709, July 2018.
- [83] Jim Jeffers, James Reinders, and Avinash Sodani. Chapter 10 - vectorization advisor. In Jim Jeffers, James Reinders, and Avinash Sodani, editors, *Intel Xeon Phi Processor High Performance Programming (Second Edition)*, pages 213–250. Morgan Kaufmann, Boston, second edition edition, 2016.
- [84] Jing Jiang, Huaifeng Zhang, Dechang Pi, and Chenglong Dai. A novel multi-module neural network system for imbalanced heartbeats classification. *Expert Systems with Applications: X*, 1:100003, 2019.
- [85] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, Mar 2019.
- [86] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.*, 13(4):455–492, 1998.

- [87] Piotr Juszczak. *Learning to recognise : a study on one-class classification and active learning*. PhD thesis, Delft University of Technology, Netherlands, 2006.
- [88] Ryoji Kadota, Hiroki Sugano, Masayuki Hiromoto, Hiroyuki Ochi, Ryusuke Miyamoto, and Yukihiro Nakamura. Hardware architecture for HOG feature extraction. In Jeng-Shyang Pan, Yen-Wei Chen, and Lakhmi C. Jain, editors, *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2009), Kyoto, Japan, 12-14 September, 2009, Proceedings*, pages 1330–1333. IEEE Computer Society, 2009.
- [89] Jalil Kazemitabar, Arash Amini, Adam Bloniarz, and Ameet S Talwalkar. Variable importance using decision trees. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 426–435. Curran Associates, Inc., 2017.
- [90] Shehroz S. Khan and Michael G. Madden. One-class classification: taxonomy of study and review of techniques. *Knowl. Eng. Rev.*, 29(3):345–374, 2014.
- [91] Wai Ginn Khong. Surface edge detection using convolutional neural network application: Image classifications of healthy and glaucoma-diseased eyes. Master’s thesis, Curtin University - School Of Electrical Engineering, Computing And Mathematical Sciences, 2019.
- [92] Peter Kotschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulò. Deep neural decision forests. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1467–1475, 2015.
- [93] Hiral Kotadiya and Darshana Patel. Review of medical image classification techniques. In Xin-She Yang, Simon Sherratt, Nilanjan Dey, and Amit Joshi, editors, *Third International Congress on Information and Communication Technology*, pages 361–369, Singapore, 2019. Springer Singapore.
- [94] Kamran Kowsari, Rasoul Sali, Lubaina Ehsan, William Adorno, Asad Ali, Sean Moore, Beatrice Amadi, Paul Kelly, Sana Syed, and Donald Brown.

- Hmic: Hierarchical medical image classification, a deep learning approach. *Information*, 11(6), 2020.
- [95] Bartosz Krawczyk, Mikel Galar, Michał Woźniak, Humberto Bustince, and Francisco Herrera. Dynamic ensemble selection for multi-class classification with one-class classifiers. *Pattern Recognition*, 83:34 – 51, 2018.
- [96] Bartosz Krawczyk, Michał Woźniak, and Francisco Herrera. On the usefulness of one-class classifier ensembles for decomposition of multi-class problems. *Pattern Recognition*, 48(12):3969–3982, December 2015.
- [97] Alex Krizhevsky. Convolutional deep belief networks on cifar-10, 2010.
- [98] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.
- [99] D. Li, J. Zhang, Q. Zhang, and X. Wei. Classification of ecg signals based on 1d convolution neural network. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6, 2017.
- [100] W. Li, Q. Guo, and C. Elkan. A positive and unlabeled learning algorithm for one-class classification of remote-sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(2):717–725, 2011.
- [101] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. pages 2999–3007, 10 2017.
- [102] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel CNN for efficient 3d deep learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 963–973, 2019.

- [103] Sebastián Maldonado and Claudio Montecinos. Robust classification of imbalanced data using one-class and two-class svm-based multiclassifiers. *Intell. Data Anal.*, 18(1):95–112, 2014.
- [104] Gustavo Malkomes and Roman Garnett. Automating bayesian optimization with bayesian optimization. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5988–5997, 2018.
- [105] Jayawant N. Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315 – 1316, 2010.
- [106] Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *J. Mach. Learn. Res.*, 2:139–154, 2001.
- [107] Larry M. Manevitz and Malik Yousef. One-class svms for document classification. *J. Mach. Learn. Res.*, 2:139–154, March 2002.
- [108] R. Matsuoka, S. Ono, and M. Okuda. Transformed-domain robust multiple-exposure blending with huber loss. *IEEE Access*, 7:162282–162296, 2019.
- [109] Andrew Maxwell, Runzhi Li, Bei Yang, Heng Weng, Aihua Ou, Huixiao Hong, Zhaoxian Zhou, Ping Gong, and Chaoyang Zhang. Deep learning architectures for multi-label classification of intelligent health risk prediction. *BMC Bioinformatics*, 18(S14), December 2017.
- [110] Mike May. Eight ways machine learning is assisting medicine. *Nature Medicine*, 27(1):2–3, January 2021.
- [111] A. Menegola, M. Fornaciali, R. Pires, F. V. Bittencourt, S. Avila, and E. Valle. Knowledge transfer for melanoma screening with deep learning. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 297–300, April 2017.

- [112] Gregory P. Meyer. An alternative probabilistic interpretation of the huber loss, 2020.
- [113] Randolph A. Miller, Harry E. Pople, and Jack D. Myers. Internist-i, an experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine*, 307(8):468–476, August 1982.
- [114] E. Miranda, M. Aryuni, and E. Irwansyah. A survey of medical image classification techniques. In *2016 International Conference on Information Management and Technology (ICIMTech)*, pages 56–61, 2016.
- [115] S. Mirjalili, S. H. Sardouie, and N. Samiee. A novel algorithm based on decision trees in multiclass classification. In *2018 25th National and 3rd International Iranian Conference on Biomedical Engineering (ICBME)*, pages 1–6, 2018.
- [116] E. Mizutani and S.E. Dreyfus. On complexity analysis of supervised MLP-learning for algorithmic comparisons. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*. IEEE.
- [117] Jonas Mockus. *The Bayesian Approach to Local Optimization*, pages 125–156. Springer Netherlands, Dordrecht, 1989.
- [118] G.B. Moody and R.G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.
- [119] Inês C. Moreira, Igor Amaral, Inês Domingues, António Cardoso, Maria João Cardoso, and Jaime S. Cardoso. Inbreast: Toward a full-field digital mammographic database. *Academic Radiology*, 19(2):236 – 248, 2012.
- [120] Mulyanto Mulyanto, Muhamad Faisal, Setya Widyawan Prakosa, and Jenq-Shiou Leu. Effectiveness of focal loss for minority classification in network intrusion detection systems. *Symmetry*, 13(1), 2021.
- [121] M. E. Munich, P. Pirjanian, E. Di Bernardo, L. Goncalves, N. Karlsson, and D. Lowe. Sift-ing through features with vipr. *IEEE Robotics Automation Magazine*, 13(3):72–77, 2006.

- [122] L. Nanni, S. Brahnem, S. Ghidoni, and A. Lumini. Bioimage classification with handcrafted and learned features. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(3):874–885, 2019.
- [123] Keisuke Nemoto, Ryuhei Hamaguchi, Tomoyuki Imaizumi, and Shuhei Hikosaka. Classification of rare building change using cnn with multi-class focal loss. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 4663–4666, 2018.
- [124] A. Nguyen, D. Moore, I. McCowan, and M. Courage. Multi-class classification of cancer stages from free-text histology reports using support vector machines. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5140–5143, 2007.
- [125] Fernando Nogueira. Bayesian Optimization: Open source constrained global optimization tool for Python - <https://github.com/fmfn/bayesianoptimization>, 2014–.
- [126] Bayu A. Nugroho. An aggregate method for thorax diseases classification - <https://arxiv.org/abs/2008.03008>, 2020.
- [127] Bayu Adhi Nugroho. Github - bayu-ladom-ipok/weopenbayesianopt: weopenbayesianopt. <https://github.com/bayu-ladom-ipok/weOpenBayesianOpt>. (Accessed on 12/23/2021).
- [128] Bayu Adhi Nugroho. An aggregate method for thorax diseases classification. *Scientific Reports*, 11(1), February 2021.
- [129] Department of Health amp; Human Services. Emphysema - <https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/emphysema>, Nov 2014.
- [130] Ademola S. Ojo, Simon A. Balogun, Oyeronke T. Williams, and Olusegun S. Ojo. Pulmonary fibrosis in COVID-19 survivors: Predictive factors and risk reduction strategies. *Pulmonary Medicine*, 2020:1–10, August 2020.

- [131] Jefferson Tales Oliva and João Luís Garcia Rosa. Differentiation between normal and epileptic EEG using k-nearest-neighbors technique. In *Lecture Notes in Computer Science*, pages 149–160. Springer International Publishing, 2016.
- [132] Xi Peng, Zhiqiang Tang, Fei Yang, Rogério Schmidt Feris, and Dimitris N. Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation. *CoRR*, abs/1805.09707, 2018.
- [133] P. Perera and V. M. Patel. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019.
- [134] Kemal Polat and Salih Güneş. A novel hybrid intelligent method based on c4.5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Systems with Applications*, 36(2, Part 1):1587–1592, 2009.
- [135] Pranav Rajpurkar, Jeremy Irvin, Robyn L. Ball, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis P. Langlotz, Bhavik N. Patel, Kristen W. Yeom, Katie Shpanskaya, Francis G. Blankenberg, Jayne Seekins, Timothy J. Amrhein, David A. Mong, Safwan S. Halabi, Evan J. Zucker, Andrew Y. Ng, and Matthew P. Lungren. Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. *PLOS Medicine*, 15(11):e1002686, nov 2018.
- [136] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *CoRR*, abs/1711.05225, 2017.
- [137] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

- [138] Murali Ravuri, Anitha Kannan, Geoffrey J. Tso, and Xavier Amatriain. Learning from the experts: From expert systems to machine-learned diagnosis models. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 3rd Machine Learning for Healthcare Conference*, volume 85 of *Proceedings of Machine Learning Research*, pages 227–243, Palo Alto, California, 17–18 Aug 2018. PMLR.
- [139] Ian Ren. This is a reimplementation of ag-cnn - <https://github.com/ien001/ag-cnn>, Nov 2019.
- [140] Dezso Ribli, Anna Horváth, Zsuzsa Unger, Péter Pollner, and István Csabai. Detecting and classifying lesions in mammograms with deep learning. *CoRR*, abs/1707.08401, 2017.
- [141] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4393–4402, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.
- [142] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):1–21, 03 2015.
- [143] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520. IEEE Computer Society, 2018.
- [144] S. Sarkar, V. M. Patel, and R. Chellappa. Deep feature-based face detection on mobile devices. In *2016 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–8, 2016.
- [145] Chiara Scelfo, Chiara Longo, Marina Aiello, Giuseppina Bertorelli, Ernesto Crisafulli, and Alfredo Chetta. Pulmonary hernia: Case report and review of the literature. *Respirology Case Reports*, 6(8):e00354, October 2018.

- [146] Emily Seto, Kevin J. Leonard, Joseph A. Cafazzo, Jan Barnsley, Caterina Masino, and Heather J. Ross. Developing healthcare rule-based expert systems: Case study of a heart failure telemonitoring system. *International Journal of Medical Informatics*, 81(8):556–565, August 2012.
- [147] Bobak Shahriari, Alexandre Bouchard-Côté, and Nando de Freitas. Unbounded bayesian optimization via regularization. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 1168–1176. JMLR.org, 2016.
- [148] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, June 2017.
- [149] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *J. Big Data*, 6:60, 2019.
- [150] Edward H. Shortliffe and Bruce G. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3):351–379, 1975.
- [151] I. Siegert, R. Böck, A. Wendemuth, and B. Vlasenko. Exploring dataset similarities using pca-based feature selection. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 387–393, 2015.
- [152] Andras Sobester, Stephen J. Leary, and Andy J. Keane. On the design of optimization strategies based on global response surface approximation models. *J. Glob. Optim.*, 33(1):31–59, 2005.
- [153] D. Sotiropoulos, C. Giannoulis, and G. A. Tsihrintzis. A comparative study of one-class classifiers in machine learning problems with extreme class imbalance. In *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, pages 362–364, 2014.

- [154] Paolo Spagnolo, Elisabetta Balestro, Stefano Aliberti, Elisabetta Coccinelli, Davide Biondini, Giovanni Della Casa, Nicola Sverzellati, and Toby M Maher. Pulmonary fibrosis secondary to COVID-19: a call to arms? *The Lancet Respiratory Medicine*, 8(8):750–752, August 2020.
- [155] Trudie Strauss and Michael Johan von Maltitz. Generalising ward’s method for use with manhattan distances. *PLOS ONE*, 12(1):1–21, 01 2017.
- [156] Masashi Sugiyama. *Introduction to Statistical Machine Learning*. Elsevier, 2016.
- [157] Ronald Summers. <https://nihcc.app.box.com/v/chestxray-nihcc>, Sep 2017.
- [158] Amelia Swift, Roberta Heale, and Alison Twycross. What are sensitivity and specificity? *Evidence-Based Nursing*, 23(1):2–4, 2020.
- [159] Oleg Sémary. <https://pypi.org/project/tensorflowcv/>, Dec 2019.
- [160] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2820–2828. Computer Vision Foundation / IEEE, 2019.
- [161] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.
- [162] Raphael Tang, Ashutosh Adhikari, and Jimmy Lin. Flops as a direct optimization objective for learning sparse neural networks. *CoRR*, abs/1811.03060, 2018.
- [163] Tao Ban and S. Abe. Implementing multi-class classifiers by one-class classification methods. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 327–332, 2006.

- [164] David M. J. Tax and Robert P. W. Duin. Support vector domain description. *Pattern Recognit. Lett.*, 20(11-13):1191–1199, 1999.
- [165] David M. J. Tax and Piotr Juszczak. Kernel whitening for one-class classification. In Seong-Whan Lee and Alessandro Verri, editors, *Pattern Recognition with Support Vector Machines, First International Workshop, SVM 2002, Niagara Falls, Canada, August 10, 2002, Proceedings*, volume 2388 of *Lecture Notes in Computer Science*, pages 40–52. Springer, 2002.
- [166] Tokusumi. keras-flops - <https://github.com/tokusumi/keras-flops>, Aug 2020.
- [167] Hamed Valizadegan, Quang Nguyen, and Milos Hauskrecht. Learning classification models from multiple experts. *Journal of Biomedical Informatics*, 46(6):1125–1135, 2013. Special Section: Social Media Environments.
- [168] Eduardo Valle, Michel Fornaciali, Afonso Menegola, Julia Tavares, Flávia Vasques Bittencourt, Lin Tzy Li, and Sandra Avila. Data, depth, and design: Learning reliable models for skin lesion analysis. *Neurocomputing*, 383:303–313, 2020.
- [169] Bram van Ginneken. Fifty years of computer analysis in chest imaging: rule-based, machine learning, deep learning. *Radiological Physics and Technology*, 10(1):23–32, Mar 2017.
- [170] B. Venkataramanaiah and J. Kamala. ECG signal processing and KNN classifier-based abnormality detection by VH-doctor for remote cardiac healthcare monitoring. *Soft Computing*, 24(22):17457–17466, July 2020.
- [171] Dani Voitsechov and Yoav Etsion. Control flow coalescing on a hybrid dataflow/von neumann GPGPU. In *Proceedings of the 48th International Symposium on Microarchitecture*. ACM, December 2015.
- [172] Jian Wang, Hengde Zhu, Shui-Hua Wang, and Yu-Dong Zhang. A review of deep learning on medical image analysis. *Mobile Networks and Applications*, 26(1):351–380, November 2020.

- [173] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3462–3471, 2017.
- [174] Xinyu Weng, Nan Zhuang, Jingjing Tian, and Yingcheng Liu. A pytorch reimplementation of chexnet:<https://github.com/arnoweng/chexnet>, Dec 2017.
- [175] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. Understanding data augmentation for classification: When to warp? In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6, 2016.
- [176] Jing Wu, Wei Hu, Yuan Wen, Wenli Tu, and Xiaoming Liu. Skin lesion classification using densely connected convolutional networks with attention residual learning. *Sensors*, 20(24), 2020.
- [177] Rikiya Yamashita, Amber Mittendorf, Zhe Zhu, Kathryn J. Fowler, Cynthia S. Santillan, Claude B. Sirlin, Mustafa R. Bashir, and Richard K. G. Do. Deep convolutional neural network applied to the liver imaging reporting and data system (LI-RADS) version 2014 category classification: a pilot study. *Abdominal Radiology*, 45(1):24–35, November 2019.
- [178] Wenjie Yang, Houjing Huang, Zhang Zhang, Xiaotang Chen, Kaiqi Huang, and Shu Zhang. Towards rich feature discovery with class activation maps augmentation for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1389–1398, 2019.
- [179] Yongxin Yang, Irene Garcia Morillo, and Timothy M. Hospedales. Deep neural decision trees. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, 2018.

- [180] Li Yao, Jordan Prosky, Eric Poblenz, Ben Covington, and Kevin Lyman. Weakly supervised medical diagnosis and localization from multiple resolutions. *CoRR*, abs/1803.07703, 2018.
- [181] Son Youn-Jung, Kim Hong-Gee, Kim Eung-Hee, Choi Sangsup, and Lee Soo-Kyoung. Application of support vector machine for prediction of medication adherence in heart failure patients. *Healthc Inform Res*, 16(4):253–259, 2010.
- [182] M. Todd Young, Jacob D. Hinkle, Ramakrishnan Kannan, and Arvind Ramanathan. Distributed bayesian optimization of deep reinforcement learning algorithms. *J. Parallel Distributed Comput.*, 139:43–52, 2020.
- [183] Hwanjo Yu. SVMC: single-class classification with support vector machines. In Georg Gottlob and Toby Walsh, editors, *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 567–574. Morgan Kaufmann, 2003.
- [184] Hwanjo Yu. Single-class classification with mapping convergence. *Mach. Learn.*, 61(1-3):49–69, 2005.
- [185] Lequan Yu, Hao Chen, Qi Dou, Jing Qin, and Pheng-Ann Heng. Automated melanoma recognition in dermoscopy images via very deep residual networks. *IEEE Transactions on Medical Imaging*, 36(4):994–1004, 2017.
- [186] Chuanhai Zhang, Wallapak Tavanapong, Gavin Kijkul, Johnny Wong, Piet C. de Groen, and JungHwan Oh. Similarity-based active learning for image classification under class imbalance. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1422–1427, 2018.
- [187] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- [188] Jianshu Zhang, Jun Du, Yongxin Yang, Yi-Zhe Song, Si Wei, and Lirong Dai. A tree-structured decoder for image-to-markup generation. In Hal Daumé III

- and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11076–11085. PMLR, 13–18 Jul 2020.
- [189] Ran Zhang, Xin Tie, Zhihua Qi, Nicholas B. Bevins, Chengzhu Zhang, Dalton Griner, Thomas K. Song, Jeffrey D. Nadig, Mark L. Schiebler, John W. Garrett, Ke Li, Scott B. Reeder, and Guang-Hong Chen. Diagnosis of coronavirus disease 2019 pneumonia by using chest radiography: Value of artificial intelligence. *Radiology*, 298(2):E88–E97, February 2021.
- [190] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 8792–8802, 2018.
- [191] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, June 2016.