THE UNIVERSITY *of* ADELAIDE

ENG 4001 Research Project
Final Report

COVID or flu? That's the question!

Group: Abbott-2022s1-EEE-UG-13148

Prepared by:
Andreas Kotsanis (a1720576)
Nivin Jose Kovukunnel (a1749677)
Mohammad Shafaie (a1748929)

Supervisors:
Professor Derek Abbott
Dr Mohsen Dorraki

24 October 2022 Faculty of Science, Engineering and Technology Pages: 50/55

Executive Summary

The rampant spread of COVID-19 has caused many testing centres to become overwhelmed, suffering from long patient queue times as well as a delayed turnaround time between testing and receiving the result. To reduce the impact on testing and healthcare centres, research must be undertaken to investigate novel techniques for COVID-19 diagnosis. One technique, which is the focus of this report, is to employ deep learning models for automated detection of COVID-19 using a patient's chest X-ray image. In particular, the aim of this project is to construct various deep learning models to determine whether patients with COVID-19 can be differentiated from those with viral pneumonia and those that are completely healthy.

As part of the project's scope, each model was analysed for its efficiency, relevancy to image classification, interpretability, and framework support availability. This project uses the data from the COVID-19 Radiography Database to train, validate and test the models implemented. This database contains 15153 chest X-ray images, collated from various studies across several Asian Universities. Each image is attributed to one of the following classes: COVID-19, viral pneumonia and normal.

This project consists of three objectives. The first objective is to perform image pre-processing and data augmentation to prepare the dataset for the deep learning models. The following pre-processing techniques were implemented: pixel normalisation, region of interest extraction, contrast limited adaptive histogram equalisation and denoising filter. Several data augmentation techniques have also been implemented such as rotating, shifting and flipping the images. The second objective is to design and implement various deep learning models to perform three-way classification of the chest X-ray images. Four deep learning convolutional neural network architectures were implemented for chest X-ray image classification. The models include three pre-trained architectures (VGG-16, Inception-V3 and DenseNet-121) and one custom made convolutional neural network architecture that was specifically designed for this project. The final objective is to evaluate each model using a series of performance metrics such as accuracy, precision, recall and F1-score on both the pre-processed and unprocessed (original) datasets. According to the results, the best performing model was DenseNet-121 which obtained F1-scores of 88% and 91% for the unprocessed and pre-processed datasets respectively. The performance of this model was verified using an Explainable Artificial Intelligence technique known as Grad-CAM. From analysis, it was determined that the pre-processed dataset increased the performance, reliability and the overall 'correctness' of the ML models.

This project was subject to several limitations including missing key patient descriptors such as age, gender, contamination period, medical history. Furthermore, the COVID-19 strain type could not be retrieved from the metadata. Sample size was another limitation in this project, as only a subset of the original database was used in order to retain a balanced dataset. In terms of future work, the authors recommend further exploring the effect of each pre-processing technique individually. In addition to this, future studies can also explore other novel deep learning techniques such as Vision Transformers for chest X-ray classification.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Meaning
AHE	Adaptive Histogram Equalisation
AI	Artificial Intelligence
AUC	Area Under the Curve
CIR	Contrast Improvement Ratio
CLAHE	Contrast Limited Adaptive Histogram Equalisation
COVID-19	Coronavirus 2019
CNN	Convolutional Neural Network
CT	Computer Tomography
CXR	Chest X-Ray
DenseNet	Dense Connected Convolutional Networks
GGO	Ground Glass Opacity
Grad-CAM	Gradient-weighted Class Activation Mapping
HE	Histogram Equalisation
HOG	Histogram-Oriented Gradient
ML	Machine Learning
MobileNet	Mobile Network
RAT	Rapid Antigen Testing
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
RT-PCR	Reverse transcription-polymerase chain reaction
SVM	Support Vector Machines
TL	Transfer Learning
VGG	Visual Geometry Group
WHO	World Health Organisation
XAI	Explainable Artificial Intelligence

Authorship Declaration

We wish to confirm that this report has no known conflict of interests and that there was no financial support provided towards this work that could have influenced its outcome.

We confirm that this manuscript has been written, edited, and approved by all named authors and that there were no other persons involved in the making of this document. We confirm that each author equally contributed to each section/chapter of the report in terms of both content and overall review. The order that the authors are listed in this document is purely alphabetical and is in no way an indication of the work and effort of any individual author.

The authors would like to give a special thanks to Professor Derek Abbott and Dr Mohsen Dorraki for their guidance and feedback throughout this project.

Signed by all authors as follows:

Andreas Kotsanis

Nivin Jose Kovukunnel

Mohammad Shafaie

Chapter 1: Introduction

1.1 Project Overview

The development of new methods to detect the Coronavirus Disease 2019 (COVID-19) is crucial to increase the testing accessibility of COVID-19. The focus of this project, 'COVID or flu? That's the question!' is to explore advanced image classification techniques to correctly classify chest X-ray (CXR) images into one of the following categories: COVID-19, normal and viral pneumonia.

1.2 Background

According to World Health Organisation (WHO), there has been a total of approximately 623.8 million COVID-19 cases and 6.5 million deaths, as of October 21, 2022 [1]. The COVID-19 virus has caused an ongoing global pandemic and with newer variants being developed rapidly, the world is struggling to adapt. The first case of the COVID-19 virus was reported in December 2019, in Wuhan, Hubei Province, China, from where it began to transmit rapidly to the rest of the countries around the world [2].

For the diagnosis of COVID-19, various methods are used with the most common method being the Reverse transcription-polymerase chain reaction (RT-PCR) testing [3]. Although RT-PCR tests can be cost-effective, patients can expect a delay in testing and receiving results, especially during an outbreak. Numerous studies also concluded that RT-PCR testing has low sensitivity during the early stages of the infection, contributing to false-negative results [4]–[6]. Chest imaging using X-rays and computer tomography (CT) scans are protocols currently practiced by healthcare centres to patients, that show strong respiratory symptoms [7]. Contrast to other popular methods such as RT-PCR testing and Rapid Antigen Testing (RAT), the process of using CXR imaging is very simple, fast and provides greater accuracy due to its high sensitivity during the early stages of the infection [8]. Viral pneumonia is still one of the leading causes of death [9]. According to WHO, chest imaging using X-rays is the best method for diagnosing pneumonia [10].

Over the recent years, there has been a significant development in the areas of Artificial Intelligence (AI) and Machine Learning (ML). With increasing computational power and growing amount of quality available data, various ML methods have already shown good performance for medical imaging diagnosis [11]. However, there are still areas of improvement in the analysis, as it requires proficiency and incorporates a diverse range of techniques to improve, accelerate and generate an accurate diagnosis. Several studies have shown that deep learning methods, more specifically, Convolutional Neural Networks (CNNs), have achieved better performance on image classification problems in comparison to other traditional ML models [12]–[14]. In this project, existing studies will be used as a guide to verify the work being conducted. Furthermore, this project will also focus on exploring deep learning models such as CNNs to perform accurate CXR image classification.

1.3 Motivation

The main motivation behind this project is to develop a new method for diagnosing COVID-19 which can serve as a viable alternative to patients who require high-accuracy testing with a quick turnaround time. In this project, advanced image classification techniques will be implemented to accurately differentiate normal patients from those with COVID-19. In addition to this, it is also important to be able to distinguish patients with COVID-19 and those with viral pneumonia. In comparison to viral pneumonia, COVID-19 is highly transmissible and can display little to no symptoms, especially during the incubation period. Therefore, it is also important to differentiate patients with COVID-19 and viral pneumonia, to help contain the spread of the COVID-19 virus and to assign appropriate medical treatments and measures.

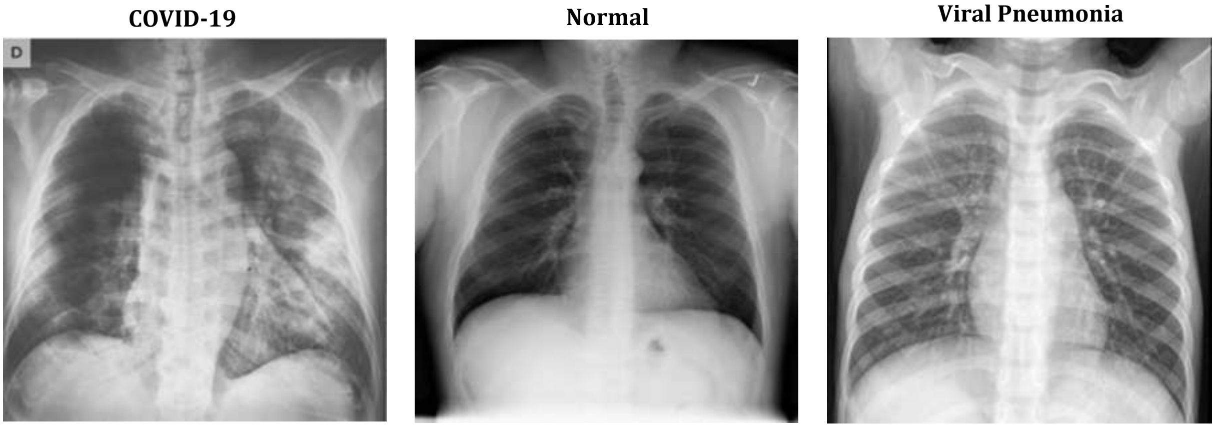


Figure 1.1: Examples of chest X-ray images for COVID-19, normal and viral pneumonia patients from the COVID-19 Radiography Database [15].

Both COVID-19 and viral pneumonia diseases display similar symptoms such as coughing, fever and shortness of breath. To an untrained eye, both diseases show similar characteristics in CXR images, as illustrated in Figure 1.1. As such, differentiating patients with COVID-19 from those with viral pneumonia using CXR images can be tedious, even for expert radiologists. For more information on CXR radiography, refer to Appendix A. Therefore, this project provides healthcare centres with an alternative method using automated ML models to detect COVID-19 and viral pneumonia diseases using CXR images. This process is also automated which has the additional benefit of lessening the burden and stress on healthcare workers, especially during outbreaks when there is a significant influx of patients that need to undergo testing.

1.4 Aims and Scope

The aim of this project is to explore advanced techniques for image classification to determine whether CXR images from patients with COVID-19 can be differentiated from those with normal lungs or viral pneumonia. This project will involve designing and constructing ML models, that can extract specific features from CXR images and then learn from it, to perform accurate classification.

The scope of this project involves identifying and selecting numerous effective ML methods that can be used to perform classification on CXR images. The ML model's efficiency, relevancy to image classification problems, interpretability, and framework support availability, will all be analysed for selection. This project will require an analysis of CXR images from normal patients as well as those that have been diagnosed with COVID-19 and viral pneumonia. There is no need to perform experiments or contact healthcare centres for CXR data as there exists numerous online datasets containing CXR images of COVID-19, viral pneumonia, and normal patients, that are free and publicly available.

This project will use data from the COVID-19 Radiography Database [15], which was created from various research across several Asian universities. The journal articles, 'Can AI Help in Screening Viral and COVID-19 Pneumonia?' by Chowdhury *et al.* [16] and 'Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images' by Rahman *et al.* [17] have provided a 'Data Availability Statement,' allowing public use of their data. The dataset constitutes of 3,616 COVID-19, 1,345 viral pneumonia and 10,192 normal CXR images.

1.5 Objectives

The technical objectives for this project are summarised in Table 1.1. This table also includes the specific tasks that will need to be undertaken for each objective as well as the expected deliverables/outcomes.

Table 1.1: List of project objectives and key specifications/outcomes.

#	Objective description	Specifications	Deliverables / outcomes
1.	Perform image pre-processing and data augmentation on the dataset provided to prepare the dataset for training.	• Resize images to 224 × 224 pixel size. • Pixel value range will be normalised to the range 0 to 1. • Perform advanced pre-processing techniques for contrast enhancement. and denoising the CXR images. • Additional data will be generated via data augmentation.	• All CXR images have been pre-processed and are ready for training the ML models. • Additional CXR images generated via data augmentation to increase robustness of ML models.
2.	Design and construct ML models to differentiate between COVID-19 and viral pneumonia in CXR images.	• 80% of the dataset will be used to train and cross-validate the ML models. The remaining 20% will be used for testing and final evaluation. • Various CNN architectures will be trained to perform classification on CXR images. • Cross-validation will be performed using K-Fold cross-validation.	• Multiple ML models capable of classifying COVID-19, viral pneumonia and normal CXR images.
3.	Evaluate ML models to determine the best performing model.	• A test set consisting of CXR images that were not used during training or validation will be used to obtain an unbiased performance of the ML models. • Several evaluation metrics will be used to evaluate models. • F1-score will be used as the primary evaluation metric to select the best performing ML model.	• Accuracy, precision, recall and F1-score quantities determined for each ML model. • Selection of best performing ML model that can accurately classify COVID-19, viral pneumonia and normal CXR images.

1.6 Document Overview

The next chapter in this document is the literature review. This chapter will concisely summarise the previous research that has been undertaken regarding CXR image classification for COVID-19 detection. Particular emphasis will be placed on the different methods and techniques that these studies implemented as well as the overall performance of the ML models. This chapter will then be proceeded by the methodology that was used to execute this project. This will include a summary of the data preparation, pre-processing and augmentation techniques that were used as well as a description of the various CNN architectures that have been implemented. The Results chapter consists of an evaluation of the CNN architectures that have been implemented on unprocessed and pre-processed data using performance metrics such as accuracy, precision, recall and F1-score. The subsequent chapter will discuss the performance of the ML models evaluated on both unprocessed and pre-processed data and the best performing model will be compared to results from previous studies. Moreover, a review will be performed on the project objectives and the outcomes for each objective will be summarised. The limitations of this project will be addressed with reference to the methodology and the results obtained. This report will conclude with a Recommendation for Future Works chapter which will provide several recommendations to extend and improve the project.

Chapter 2: Literature Review

2.1 Introduction

Ever since the emergence of the global COVID-19 pandemic, numerous studies have attempted to detect traces of COVID-19 in CXR images using a wide range of image classification techniques. Although a large number of these studies have already demonstrated a high classification accuracy for CXR images, there is limited research comparing these different deep learning techniques for the same dataset. As such, the aim of this literature review is to collate the results from these various studies and identify the techniques and/or models that yielded the best results for CXR image classification.

2.2 Findings

This review consists of two sections. In the first section, an analysis is performed looking into the data pre-processing techniques that have been used to prepare CXR images for the deep learning models. The second section will then draw upon literature regarding various image classification techniques that have previously been used to classify CXR images.

2.2.1 Data Preparation

The main techniques that have been used by earlier studies to prepare the CXR data can be allocated into one of the following categories: data pre-processing and data augmentation. In this case, data pre-processing refers to a group of techniques that are used to improve the quality and/or contrast of a CXR image. On the other hand, data augmentation refers to a group of techniques that are used to artificially increase the size of a dataset by creating modified copies of existing images.

2.2.1.1 Data Pre-processing

In this case, data pre-processing refers to several techniques that are used to enhance the quality of the dataset that is used to train the ML model. One of these techniques involves adjusting the contrast on the CXR images to make features encompassing the lungs more prominent. In a study conducted by Reynaldi *et al.* [18], the contrast in CXR images was adjusted by applying a Contrast Limited Adaptive Histogram Equalisation (CLAHE) filter. This filter partitions the image into similarly sized non-overlapping regions and then performs Adaptive Histogram Equalisation (AHE) on each section [19]. As part of their results, the researchers of this study [18] compared the accuracy of an ML model with and without using the CLAHE filter. The results from this experiment showed increased accuracy, sensitivity, and specificity when the CLAHE filter was applied. These results are supported by a separate study conducted by Umri *et al.* [20], in which the accuracy of the CNN model was increased by 1% after the CLAHE filter was used to pre-process the dataset. The CLAHE filter can also be used in conjunction with other pre-processing techniques such as intensity normalisation to remove low or high frequencies from the CXR images [21].

Contrast enhancement can also be performed by means of mathematical morphology [22]. Mathematical morphology involves creating a structural element from the original image. This structural element consists of binary numbers (1s and 0s), which is then overlayed with the original image to suppress some of the complex background tissue commonly found in medical CXR images [23]. An example of contrast enhancement using mathematical morphology was used successfully implemented in a study by Sarki *et al.* [22] to prepare a CXR dataset for an ML model. The effectiveness of contrast enhancement via mathematical morphology was demonstrated in a study by Kimori [23], in which this technique was compared against several other popular contrast enhancement methods, including CLAHE. The results showed a higher contrast improvement ratio (CIR) for the mathematical morphology technique over the other methods that were considered in this paper [23].

Region of Interest (ROI) extraction is another technique that can be applied to the CXR images during the data pre-processing stage. This technique is used to crop down the CXR images to eliminate irrelevant information such as the patient's arms, neck, stomach etc., which may otherwise affect how the ML model classifies the images. This technique was applied by Tabik *et al.* [24] during their investigation into how deep learning models can be used to detect COVID-19 from a set of CXR images. To accomplish this, without manually cropping the images, a U-Net segmentation model [25] was employed to isolate the lungs in the CXR images.

A slightly less common data pre-processing technique is wavelet decomposition. This technique was used by Singh *et al.* [26] to convert the CXR images from the spatial domain to the frequency domain, from which multiresolution analysis can then be performed. This technique offers several advantages such as reducing the background noise that plagues CXR images while also improving the image contrast in the process [27].

2.2.1.2 Data Augmentation

Data augmentation is a technique that is used to expand the number of images contained in a dataset by making copies of the existing data with some slight alterations. This technique has been used in various studies [24], [28], [29] with the aim of increasing the ML model performance by subjecting it to a wider and richer dataset. Data augmentation was used by Li *et al.* [29], when attempting to classify COVID-19 CXR images using a novel neural network model. In this study, the authors noted that the number of COVID-19 CXR images (400) were far fewer than those in the viral pneumonia and normal classes (7000 and 10000 respectively). To avoid the introduction of biases in the ML model, the COVID-19 dataset was expanded twofold via data augmentation [29]. A similar process was used by Mishra *et al.* [30] to expand a limited dataset of 1800 CXR to over 15000 images. The data augmentation techniques that were implemented in these studies include sheering, zooming, shifting, rotating and (horizontally) flipping the CXR images [29], [30]. One way to employ these techniques is to use python libraries, such as OpenCV, as described in the study conducted by Hernandez *et al.* [31]. Unlike previous studies [29], [30], the study conducted by Hernandez *et al.* [31] also included Gaussian noise as a data augmentation technique. This was done with the aim of increasing the robustness of the ML model to poor CXR image quality. When altering these images, the authors in studies [29], [30] noted that they had to be careful to not create images that were drastically different than those in the original dataset. Thus, the authors opted to only used one data augmentation technique at a time for each particular

CXR image. In addition to this, the degree that these images were rotated/shifted was limited to a range of +/- 10% [29], [30].

2.2.2 Image Classification Techniques

The two main image classification techniques that will be discussed in this section are CNNs and tradition computer vision techniques. CNNs are deep learning algorithms specific to image classification. It employs end-to-end learning which requires the algorithm to determine an underlying pattern within a group of images so that they may be correctly classified. There are various architectures for CNNs. The main architectures that will be discussed in this literature review are: ResNet, VGG, DenseNet and MobileNet. On the other hand, with traditional computer vision techniques, the user is responsible for extracting the features of interest from these images manually. These extracted features are then used to train an ML model so that these images may be accurately classified.

2.2.2.1 Convolutional Neural Networks

ResNet is one of the most common CNN architectures for image classification. This architecture is renowned for its quick training despite consisting of network layers that are considerably deeper than many other typical CNN architectures [32]. A variation of this architecture, ResNet-101, was used by Reynaldi *et al.* [18] to identify CXR images for normal patients and those with COVID-19. This model was able to accurately classify these images with a reported accuracy of 99.61%, a recall of 99.62% and a specificity of 99.6% [18]. These results are supported by a separate, yet similar, study which used Support Vector Machines (SVMs) alongside the ResNet-50 architecture to categorise CXR images as either COVID-19, pneumonia or normal [33]. In this study, Narin [33] reported an overall accuracy, recall and specificity of 94.86%, 96.04% and 96.62% respectively using a SVM Cubic model.

The VGG architecture is another popular CNN model that has been used to classify CXR images. Umri *et al.* [20] used the VGG-16 architecture (consisting of 16 convolutional layers) to achieve up to a 99% training accuracy and a 97% validation accuracy when classifying CXR images into COVID-19 and normal classes. In a similar study, Sarki *et al.* [22] used a pre-trained VGG-16 model to achieve a 100% accuracy for binary classification (COVID-19/normal) and an 87.5% accuracy for tertiary classification (COVID-19/pneumonia/normal). Alam *et al.* [34] used VGG-19 architecture (with 19 convolutional layers) to a 99.49% accuracy. This study also implemented individual feature extraction techniques such as histogram-oriented gradient (HOG) which is believed to have increased the accuracy of the model [34].

DenseNet is a CNN architecture which consists of many convolutional layers but makes use of concatenation to reduce the number of hyperparameters required [35]. Chaudhary *et al.* [36] implemented a pre-trained DenseNet architecture to detect the presence of COVID-19 in CT scans. This was done in a two-stage classification framework which yielded an overall accuracy of 89.3% for a three-class classification: COVID-19, pneumonia and normal. Similar results were also obtained by Albahli, Ayub and Shiraz [37] who used a pre-trained DenseNet architecture to obtain an accuracy of 92% using the same three-class classification identified earlier. Montalbo [35] was able to boost the performance of the DenseNet architecture by using a lightweight model with partial layer freezing and feature extraction to classify CXR images. With these techniques, Montalbo [35] achieved an accuracy of 99.84%, a precision of 99%, a recall of 100% and a F1-

score of 99%. Bohmrah and Kaur [38] used CXR images to train several variations of DenseNet architectures including DenseNet-121, DenseNet-169 and DenseNet-201. The best performing model was DenseNet-201, which achieved an accuracy of 95.2% [38]. The authors also experimentally determined that DenseNet architectures produced better results using the RMSprop optimiser over the more commonly used Adam and Adamax optimisers [38].

MobileNet is a CNN architecture designed for mobile and vision-based embedded applications [39]. Mohammadi *et al.* [40] achieved an accuracy of 99.1% when using the MobileNet architecture to perform binary classification of CXR images (COVID-19 or normal). A modified version of this model was used by Tangudu, Kakarla and Venkateswarlu [41] to perform the same binary classification as above. This model used a residual separable convolutional block in conjunction with a pre-trained MobileNet model [41]. Overall, this model achieved a 99% accuracy for two publicly available datasets (COVID5K and COVIDRD) [41]. Another modified MobileNet model was also used by Jia, Lam and Xu [42] to perform a 5-way classification for CXR images (COVID-19, tuberculosis, viral pneumonia, bacterial pneumonia and normal). The modified MobileNet model was based on the architecture of MobileNet-V3_Small and aimed to overcome some of the issues that were present in the original model such as vanishing gradients and overfitting [42]. This modified MobileNet architecture produced similar results to that which was used in [41], with an overall accuracy of 99.6% for the 5-way classification of CXR images [42].

Some comparative studies have also been conducted to determine the best performing CNN architecture for COVID-19 diagnosis via CXR images. El Asnaoui and Chawki [21] measured the accuracy of several CNN architectures such as VGG-16, VGG-19, DensNet-101, Inception_ResNet-V2, Inception-V3, ResNet-50 and MobileNet-V2. The models with the highest accuracies were reported to be Inception_ResNet-V2 (92.18%) and DensNet101 (88.09%) [21]. On the other hand, studies performed by Jabber *et al.* [43] and Akter *et al.* [44] have demonstrated that a modified version of the MobileNet-V2 architecture can outperform other CNN models including DenseNet-121 and the ResNet-50V2. In a separate study, Shorfuzzaman and Masud [45] conducted a comparative study with several major CNN architectures including VGG-16, ResNet-50V2, MobileNet and DenseNet. The authors concluded that ResNet-50V2 was the best performing CNN model in terms of accuracy, precision, sensitivity, specificity, and F1-score [45]. This was closely followed by the MobileNet architecture which managed to outscore ResNet50-V2 in the Area Under Curve (AUC) metric [45]. These results are supported by Hernandez *et al.* [31] who performed a similar study where they compared their own custom CNN model to the likes of ResNet-50, VGG-16 and DenseNet-121. Once again, the ResNet-50 architecture outscored all the other models (including the custom CNN model) for all the performance metrics including accuracy, precision, recall and the F1-score [31]. Santoso and Pernomo [46] also created their own CNN model which they used to compare against the ResNet-50, Inception-V3 and Xception architectures. In this case, the custom CNN model outperformed the other models in this study [46]. However, the researchers noted that the computational time for their custom model exceeded that of the other models. Another comparative study was performed by Rahaman *et al.* [47] using 15 pre-trained CNN models to determine which one could best classify CXR images into normal, pneumonia and COVID-19 categories. In this case, the VGG-19 architecture was found to outperform the other models in this study [47].

2.2.2.2 CNNs vs Traditional Computer Vision Techniques

Ever since deep learning techniques became mainstream in digital image processing, traditional computer vision techniques have become obsolete [12]. This can be seen in a study performed by Hedjazi, Kourbane and Genc [14], in which a comparison is drawn between CNN and classical ML methods. In this case, the CNN model (AlexNet) was shown to have a validation accuracy of at least 86.75%, thus outperforming the traditional ML method with a maximum validation accuracy of 83% [14]. This result was consistent for the two datasets that were considered in this study, the smaller of which contained approximately 6000 images and the larger containing nearly 300,000 [14]. On the other hand, O'Mahoney *et al.* [12] notes that deep learning methods (such as CNNs) can sometimes be unnecessary as traditional computer vision techniques can sometimes classify images with fewer lines of code and greater efficiency. Another advantage for traditional computer vision techniques is that the algorithms are not class-specific and can thus be used to detect features from any image in the training dataset [12]. This differs from CNNs which are required to learn the features of each class of images separately. As such, a poorly constructed training dataset will likely result in a subpar performing CNN model. That being said, the major advantage of CNNs is the elimination of having to manually perform feature extraction, which is perhaps the most time-consuming process in traditional ML methods [14]. In another study, López-Cabrera *et al.* [48] investigated the limitations of deep learning methods for CXR image classification and compared those to traditional ML methods. In this paper the authors suspected that the patterns extracted via deep learning approaches can be subtle and often overlap with other viral pneumonias [48]. Furthermore, results from their study showed that many CNN models suffered from shortcut learning, in which the model would use simple characteristics to classify the image as opposed to learning and capturing the true essence of the underlying data. This was observed when the authors noted that a large portion of the region outside the lungs was being used to classify the CXR image [48]. This result is highly irrational and can be partially attributed to improper image pre-processing. In this case, the authors recommend using traditional ML methods which they believe are better at generalising CXR image classification for new/unseen datasets [48].

2.2.3 Explainable Artificial Intelligence

Since ML models effectively function as black boxes, research is also being undertaken to understand how these models classify data. This area of research, known as Explainable Artificial Intelligence (XAI), has been used in various CXR classification studies [22], [42] - [45]. A popular XAI approach is Gradient-weighted Class Activation Mapping (Grad-CAM), which adds a heatmap over an image showing the key features that the ML model used for classification. Grad-CAM has been used by in a study by Sarki *et al.* [22], in which the ML model was shown to classify images based on irrelevant information unrelated to the lungs. In this case, the ML model was shown to focus on the edge of the CXR, the shoulder joints and even on random text/numerals that can be found on the corner of CXR images. Grad-CAM has also been used to verify the performance of some ML models. This was shown in studies performed by Jia, Lam and Xu [42] and Akter *et al.* [44], where the ML models classified the CXR images by primarily looking at the lung region. Despite the existence of other XAI approaches, such as SHAP and LIME, Grad-CAM appears to be the most popular technique for CXR verification.

2.2.4 Comparison of CNN models

A summary of the main literature discussed in Section 2.2.2.1 can be found in Table 2.1 below. It should be noted here that the data size column corresponds to the size of the original data, i.e., prior to performing any data augmentation. Most the literature considered here implemented CNN architectures in conjunction with Transfer Learning (TL). Furthermore, for comparison-based studies which considered more than one CNN architecture, the results for the best performing model were selected for display. These studies are indicated by an asterisk (*).

Table 2.1: Summary of literature that performed X-ray classification using CNN models.

Study	Classes	Data size	Data Preparation Techniques	Model(s)	Performance Metrics	Average Performance (%)
Reynaldi *et al.* [18]	COVID-19 Normal	1281 1281	CLAHE	ResNet-101 + TL	Accuracy Recall Specificity	99.61 99.62 99.60
Narin [33]	COVID-19 Pneumonia Normal	219 1345 1341	N/A	ResNet-50 + SVM Cubic + TL	Accuracy Recall Specificity	94.86 96.04 96.62
Umri *et al.* [20]	COVID-19 Normal	100 100	CLAHE	VGG-16 + TL	Accuracy	97
Sarki *et al.* [22]	COVID-19 Pneumonia Normal	296 3875 1341	Math. Morphology	VGG-16 + TL	Accuracy Recall Specificity	87.50 96.43 100
Alam *et al.* [34]	COVID-19 Normal	1979 3111	ROI Extraction	VGG-19 + TL	Accuracy Recall Specificity	99.49 93.65 95.7
Chaudhary *et al.* [36]	COVID-19 Pneumonia Normal	171 60 76	N/A	DenseNet-121 + TL	Accuracy	89.3
Albahli, Ayub and Shiraz [37]	COVID-19 Pneumonia Normal	590 6057 8851	Data Augmen.	DenseNet-512 + TL	Accuracy Recall Specificity	92 85 99
Montalbo [35]	COVID Pneumonia Normal	1281 4657 3270	N/A	Custom DenseNet + TL	Accuracy Precision Recall F1-score	97.99 98.38 98.15 98.26
Bohmrah and Kaur [38]	COVID-19 Pneumonia Normal	111 70 70	N/A	DenseNet-201 + TL	Accuracy Precision Recall F1-score	86 92 91 91
Mohammadi *et al.* [40]	COVID-19 Normal	181 364	Data Augmen.	MobileNet + TL	Accuracy Precision Recall F1-score	99.1 100 98.0 99.0
Tangudu, Kakarla and Venkateswarlu [41]	COVID-19 Normal	1341 1200	N/A	MobileNet + TL	Accuracy Recall Specificity	99.65 99.65 100.0

Jia, Lam and Xu [42]	COVID-19 Tuberculosis B. Pneumonia V. Pneumonia Normal	1770 1436 1700 1345 1341	N/A	MobileNet	Accuracy Recall Precision	95.0 95.0 95.3
El Asnaoui and Chawki [21]*	Coronavirus B. Pneumonia Normal	1724 2780 1583	CLAHE	Inception_ ResNet-V2	Accuracy Recall Specificity Precision F1-score	92.18 92.11 96.06 92.38 92.07
Jabber *et al.* [43]*	COVID-19 Non-COVID	500 6000	N/A	MobileNet	Accuracy Recall Specificity Precision F1-score	98.6 87.8 99.3 87.8 87.8
Akter *et al.* [44]*	COVID-19 Normal	3616 10192	Data Augmen. CLAHE	MobileNet + TL	Accuracy Recall Specificity Precision F1-score	98 98 97 97 97
Shorfuzzaman and Masud [45]*	COVID-19 Pneumonia Normal	226 226 226	Data Augmen.	ResNet-50V2	Accuracy Recall Specificity Precision F1-score	98.15 98.26 98.89 97.87 98.06
Hernandez *et al.* [31]*	COVID-19 Pneumonia Normal	1234 4576 16627	N/A	ResNet-50 + TL	Accuracy Recall Precision F1-score	90.63 91.67 90.00 90.72
Santoso and Pernomo [46]*	COVID-19 Pneumonia Normal	206 206 206	N/A	FCovNet	Accuracy Recall	99.55 96.67
Rahaman *et al.* [47]*	COVID-19 Pneumonia Normal	260 300 300	Data Augmen.	VGG-19	Accuracy Recall Precision F1-score	89.3 89.7 90.8 89.6

2.3 Review Conclusions

Referring to Table 2.1, a large amount of research has already been conducted on the classification of CXR images using CNN models. Although various performance metrics were used, the accuracy metric is the only one that is common to all these studies. As such, this will be the major performance metric that will be used to compare the results from these studies.

Firstly, by examining the classes column, it is evident that the studies that performed binary classification (i.e., studies that only consisted of two classes) had on average a higher accuracy value. This result is expected as it is relatively easy to differentiate a normal patient from one that contains COVID-19. However, when additional classes are added, such as viral pneumonia, the model may struggle to differentiate between the various illnesses (especially if these diseases display similar CXR characteristics). Out of the seven studies that performed binary classification, the minimum testing accuracy (97%) corresponded to the study conducted by Umri *et al.* [20]. Despite this impressive result, this accuracy is slightly lower compared to the rest of the binary classification studies, which contained an average performance accuracy of 99.08%. This is likely attributed to the low data size that was used by Umri *et al.* [20], which only contained 100 CXR images for each class. Due to this small data size, the VGG-16 model was likely inadequately trained compared to the models implemented in some of these other studies which used thousands of CXR images for training. In this case, the researchers of this study could have benefitted from implementing data augmentation to increase the training data sample size. This technique was used by Mohammadi *et al.* [40] and who was able to achieve an accuracy of 99.1%, despite the fact that the original data size only contained approximately 500 CXR images. However, it should be noted that these two studies used different CNN architectures and thus their results cannot be fully compared.

On the other hand, the average accuracy for the studies that performed a three-way classification is approximately 92.5%. Overall, the ResNet architecture consistently displayed a high-performance accuracy with respect to the other studies. This can be seen in studies by El Asnaoui and Chawki [21], Hernandez *et al.* [31], Narin [33] and Shorfuzzaman and Masud [45] which all used the ResNet architecture to obtain classification accuracies of 92.18%, 90.63%, 94.86% and 98.15% respectively. In one particular study by Montalbo [35], the custom DenseNet architecture was shown to perform reasonably well, with an overall accuracy of 97.99%. However, when considering the other studies that used DenseNet, this architecture had a considerably lower average accuracy of approximately 89%. In addition to this, the DenseNet architecture is infamous for its long training times and larger number of hyperparameters needed [35]. As such this model is likely less suited to performing image classification compared to some of the other architectures considered here. Similarly, the VGG architecture also had a lower-than-average performance compared to ResNet with an average classification accuracy of 88.4% [22], [47].

The CLAHE filter was implemented in several studies by Reynaldi *et al.* [18], Umri *et al.* [20], El Asnaoui and Chawki [21], and Akter *et al.* [44]. These studies used this filter to improve the contrast of the CXR and to highlight small features that might otherwise be missed by the CNN model when performing image classification. The studies that implemented this filter achieved an average accuracy of over 95% despite the various CNN architectures and data sizes used. An interesting data pre-processing technique that was used by Alam *et al.* [34] is ROI extraction.

ROI extraction was used to remove any unnecessary labels (such as numbers and letters) that commonly appear on the edges of CXR images. This process encourages the CNN model to focus in on the lung region of the CXR image and thus lessens the likelihood of the model using shortcut learning when classifying the image [48]. Unfortunately, ROI extraction appears to be overlooked in most of the studies that were considered in this literature review. This pre-process technique can be quite tedious to implement (especially for datasets with thousands of images), which is likely why many researchers have elected to skip this process entirely. Finally, it is worth noting that a large number of these studies implemented CNNs via TL. TL was used to reduce the training time required by reusing the model hyperparameters from a different (albeit similar) task.

In summary, upon reviewing the previous literature, the following techniques will be considered for implementation in our project:

1. Data augmentation to increase/balance out the data size between the different classes.
2. Image contrast enhancement via a CLAHE filter.
3. ROI extraction to remove unwanted labels and to focus in on the lung region.
4. Implementation of various CNN architectures including such as VGG, InceptionNet and DenseNet.
5. Implementation of Grad-CAM to verify the correctness of the ML model's prediction.
6. An evaluation of various models using performance metrics such as accuracy, recall, F1-score.

Chapter 3: Methods

3.1 Introduction

This chapter recounts the methodology that was used to classify the CXR images using ML models. The first section describes the techniques that were used to prepare the data. These techniques are categorised into the following subsections: data selection, pre-processing, and augmentation. This chapter will then discuss the various CNN architectures that were implemented to classify the CXR images. This includes a custom CNN model that was designed from scratch and several pre-trained CNN architectures such as VGG-16, Inception-V3 and DenseNet-121. Thereafter, an XAI approach, Grad-CAM, used to interpret and verify the correctness of the predictions produced by the ML models will be discussed. An overview of the methodology above is presented in Figure 3.1.

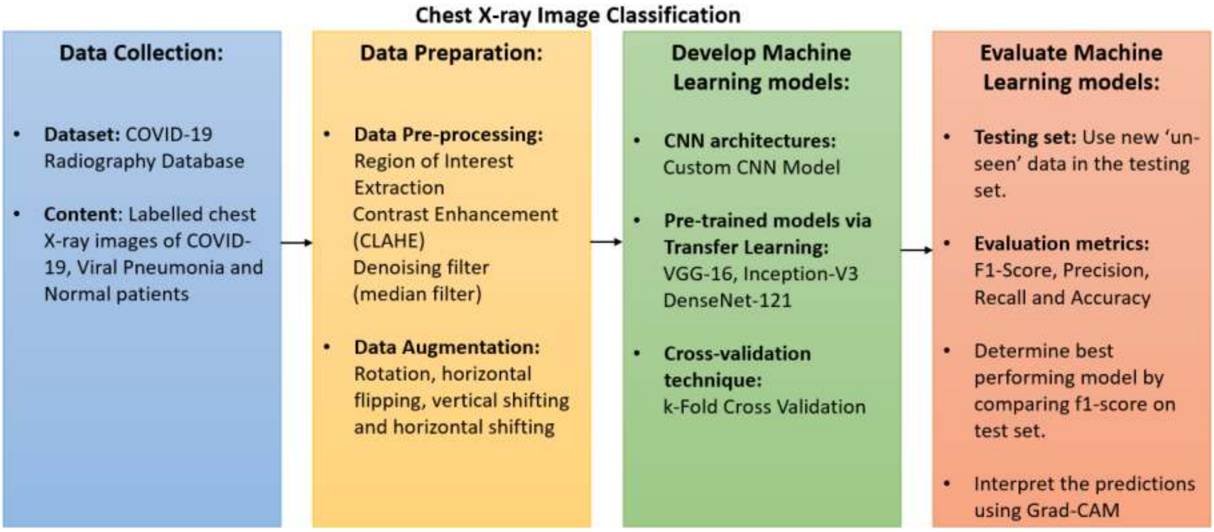

Chest X-ray Image Classification
Data Collection:
Dataset: COVID-19 Radiography Database
Content: Labelled chest X-ray images of COVID-19, Viral Pneumonia and Normal patients
Data Preparation:
Data Pre-processing: Region of Interest Extraction Contrast Enhancement (CLAHE) Denoising filter (median filter)
Data Augmentation: Rotation, horizontal flipping, vertical shifting and horizontal shifting
Develop Machine Learning models:
CNN architectures: Custom CNN Model
Pre-trained models via Transfer Learning: VGG-16, Inception-V3 DenseNet-121
Cross-validation technique: k-Fold Cross Validation
Evaluate Machine Learning models:
Testing set: Use new 'un-seen' data in the testing set.
Evaluation metrics: F1-Score, Precision, Recall and Accuracy
Determine best performing model by comparing f1-score on test set.
Interpret the predictions using Grad-CAM

Figure 3.1: Diagram showing the methodology used in this project.

3.2 Data Selection

The ML experiments were performed on the COVID-19 Radiography database [15]. For the COVID-19 Radiography database, a data imbalance exists, as shown in Figure 3.2, which may cause false classification performances [49]. Furthermore, to minimise the training time required for the dataset, a random sample of 1,345 CXR images was selected for each class. Since the viral pneumonia class contained the least amount of CXR images (1,345), this sample size was used to randomly select the CXR images from the COVID-19 and normal classes to help even out the distribution.

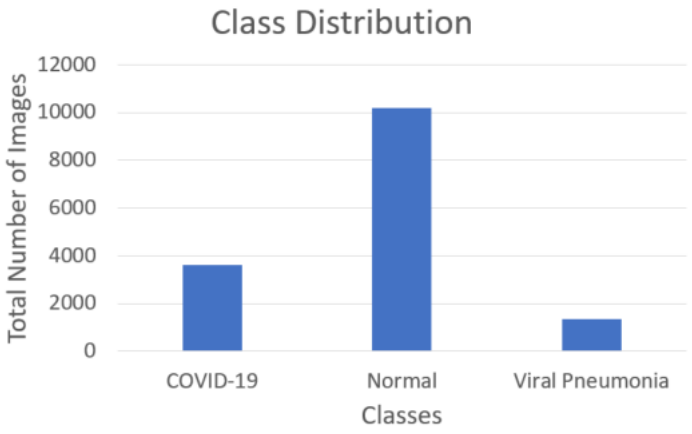


Figure 3.2: Bar graph showing image distribution between classes for the dataset.

3.3 Data Pre-processing

Based on the literature review that was conducted in section 2, several studies reported better ML model performance for data that had undergone pre-processing [18], [20]. In this report, the performance of the ML models will be evaluated using the original dataset as well as the pre-processed dataset. The pre-processing techniques that were implemented include pixel normalisation, region of interest extraction, CLAHE and a denoising filter.

3.3.1 Pixel Normalisation

The CXR images present in the dataset were in PNG format, with sizes of the images fixed to 299 × 299 pixels. For performing the experiments, the image size of 224 × 224 was selected, as this was the required input size for all of the TL models used in this project. The bilinear interpolation method was used to compress (i.e., down sample) the images to the target size. For further information on bilinear interpolation refer to section 3.3.2.2. Pixel normalisation is crucial as it reduces the computational time of the deep learning models [50]. The learning process of the deep learning models can be significantly slower, when the inputs hold large integer values. The CXR images in the datasets were normalised, such that the pixel value ranged from 0 to 1. This was performed by multiplying every pixel value with a multiplication factor. The dataset contained 8-bit RGB colour images, and the pixel values ranged from 0 to 255. Therefore, the multiplication factor can be computed using the following expression:

$$\text{Multiplication factor} = \frac{\text{pixel value} - \text{min (pixel value)}}{\text{max(pixel value)} - \text{min(pixel value)}} = \frac{\text{pixel value}}{255}. \qquad \textit{Eq. (3.1)}$$

3.3.2 Region of Interest Extraction

ROI extraction is crucial to ensure that the ML model focuses on the important aspects of the CXR image, which in this case is the lung region. As a result, each CXR image was cropped to remove unnecessary information such as the shoulder joints and stomach. Since this study consisted of over 3,000 CXR images, it was not feasible not manually crop each CXR image. This process was automated by overlaying each CXR image with its own binary mask, which outlined the lungs. To generate the binary mask for each CXR image, the U-Net CNN model was employed similar to the study conducted by Tabik *et al.* [24]. This process is outlined below in section 3.3.2.1 Lung Segmentation. Once the binary masks were obtained, one could either use these masks to isolate only the lungs for each CXR image, or alternatively, one could use these masks to create an outer box region encompassing all the information around the lungs. For this study, the latter option was used to remove any possibility of the ML models classifying the images based on their shape of the lungs. This method that was used to create this outer box region using the complementary binary masks is described in section 3.3.2.2 Outer Box Region. The ROI extraction process employed is illustrated in Figure 3.3.

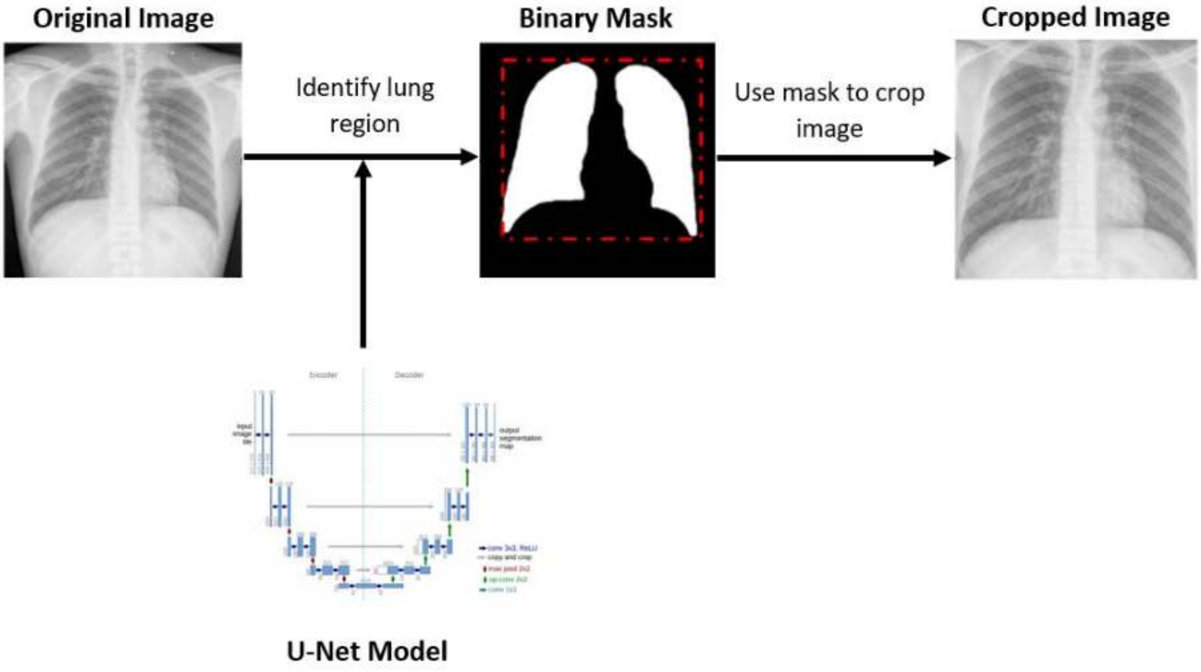


Figure 3.3: Flow diagram depicting the region of interest extraction process.

3.3.2.1 Lung Segmentation

The first stage in the ROI extraction involves performing lung segmentation to retain the binary masks associated with each CXR image. Lung segmentation of the CXR images can be difficult due to various reasons including, non-pathological changes, pathological changes, and foreign body coverage [51]. Non-pathological changes consider the variation in the shape and size of the lung due to varying age, gender, and heart size. Pathological changes consider the variation in the intensity value regarding opacity, which becomes a high value as a result of a severe lung disease. Foreign body coverage considers the obstruction of the lung field caused by the patient's clothes or medical equipment (e.g., pacemaker) [51]. Lung segmentation can be performed manually by radiologists using manual delineation techniques to extract the lung component of the CXR image. However, this can be a tedious task which is prone to human errors. In a real-world scenario where there is a huge influx of patients, it is not efficient to perform manual lung segmentation, as this procedure can be extremely time inefficient. Therefore, it is important to perform lung segmentation of CXR images to retrieve binary masks using an automatic method that is highly reliable.

The CNN architecture, U-Net, originally designed and commonly used for medical segmentation, was used to perform lung segmentation. The input to the U-Net model is the CXR images of size 256 × 256 × 1. Note that, this is the default size of the CXR images provided by the COVID-19 Radiography database. The output of the U-Net model is a binary mask of the lung component of the CXR. The size of the output image is the same as the size of the input image, which is 256 × 256 × 1. Figure 3.4 provides an illustration of the expected input and output of the U-Net model for lung segmentation.

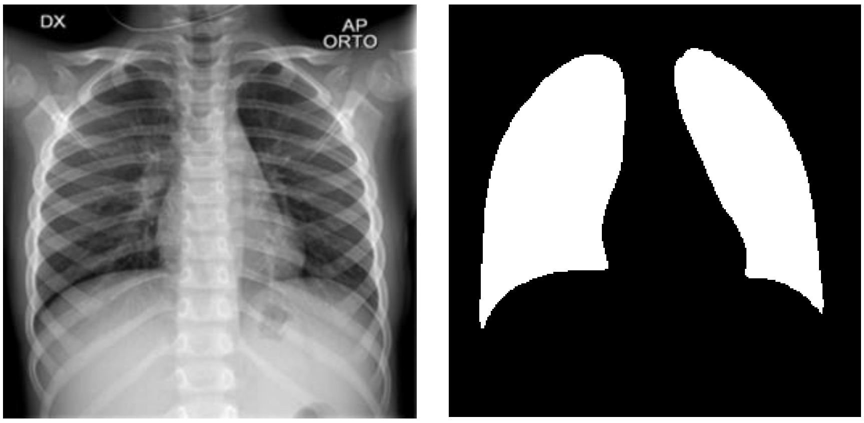


Figure 3.4: Input chest X-ray image (left) and output of the U-Net model (right).

For the U-Net model to learn distinct patterns and generate reliable and accurate binary masks, it is crucial to pass in binary masks for training purposes. The COVID-19 Radiography database does provide labelled binary masks which can be used for training the U-Net model. Once the U-Net model has been trained using labelled binary masks, this model can be used to predict binary masks for new CXR images that were not used for training. This is highly desirable, as in a real-world scenario, the trained U-Net model can be used for generating accurate and reliable binary mask for a new patient's CXR image within a relatively short timeframe.

All the CXR images present in the database can be used for training, validating, and testing purposes. The main objective in this ML experiment is to generate binary masks for the CXR images. Therefore, it is not important to consider the data imbalance problem associated with the dataset. As the U-Net model is to be used for generating binary masks for un-seen CXR images, it is important that the model is trained with a large wide-range of CXR images, which allows the U-net model to achieve good generalisation. Table 3.1 summarises the distribution of the train, validation, and test sets used for lung segmentation. The training set consisted of 60% of the entire data. The validation set comprised of 20% of the entire data and the remaining 20% were used for testing the trained U-Net model.

Table 3.1: Distribution of chest X-ray images between the classes.

Set	**Count**	**Percentage**
Train	9,091	60%
Validation	3,031	20%
Test	3,031	20%
Total	15,153	100%

The naming of the U-Net model is derived from the symmetric shape (which has a “U” shape) followed by the model. The model, as illustrated in Figure 3.5, comprises of two main networks: an encoder network and a decoder network. The encoder network, also referred as the contraction path, is used to capture information of the image that is passed in. The decoder network, also referred as the expansion path, is used to generate the segmentation output utilising the encoded information [52].

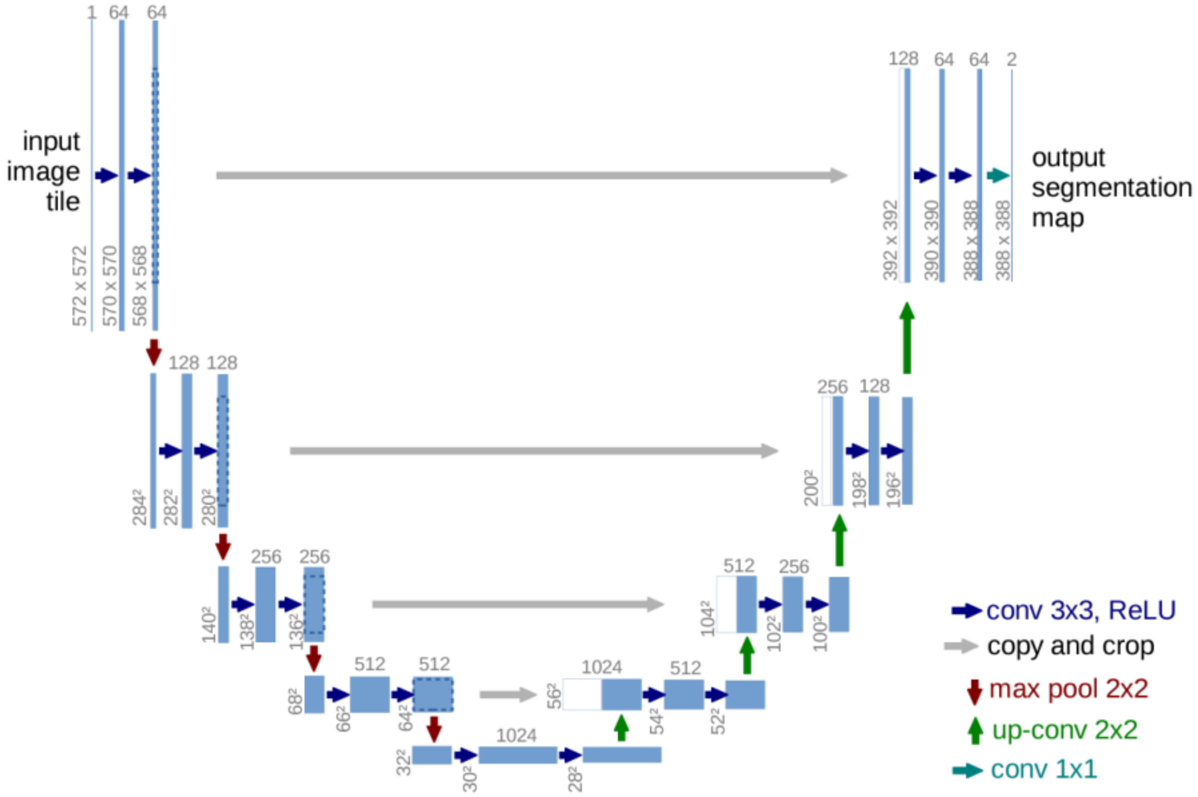


Figure 3.5: Diagram showing architecture of the U-Net model [52].

The encoder follows the architecture of a CNN. It comprises of two 3 × 3 convolutional layers where each layer is followed by a Rectified Linear Unit (ReLU) and a 2 × 2 max pooling layer for down sampling purposes. The input image is encoded into feature maps at multiple different stages [52]. The decoder network comprises of up sampling the feature map at each level, which is followed by transposed convolutional layer of size 2 × 2. Concatenation is then applied with the corresponding feature map from the encoder. Thereafter, two 3 × 3 convolutional layers are used where each layer is followed by a ReLU. Finally, a 1 × 1 convolutional layer is applied to map the channels to the expected number of classes [53].

Figure 3.6 shows several examples of predicted binary masks by the U-Net model compared with the ground truth (Actual). From visualisation, it can be observed that for the selected CXR images, the predicted binary masks closely follow the ground truth. In this project, the ground truth binary masks for the CXR images were used for obtaining the outer box region.

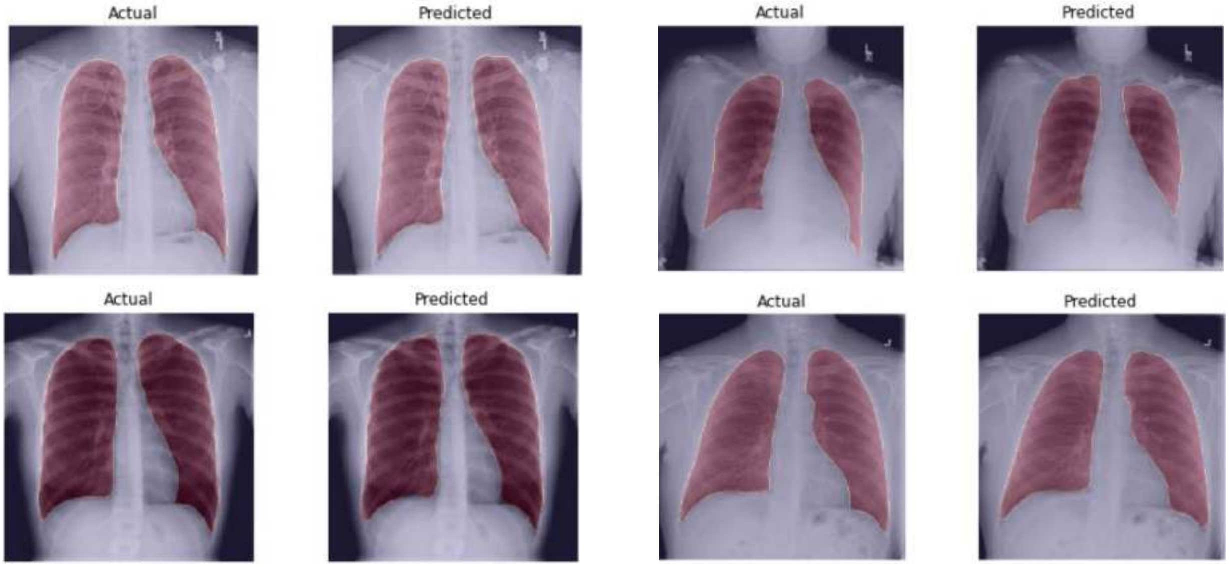
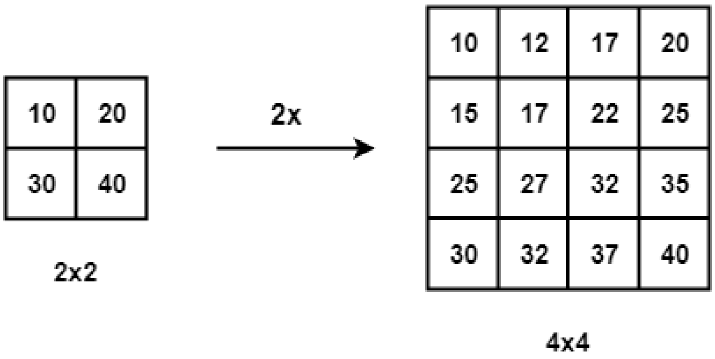


Figure 3.6: Comparison of lung region with the predicted lung segmentation from the U-Net model.

3.3.2.2 Outer Box Region

The binary mask for each CXR image were then imported into MATLAB to extract an outer box region surrounding the lungs. To achieve this, the binary mask was used to determine the leftmost, rightmost, upper, and lower bounds of the lung region. These bounds were then used to form a box and crop the CXR image. However, to pass these images into the CNN model each cropped CXR image must have the same size. In this case, bilinear interpolation was used to resize the cropped CXR images into a fixed size of 224 × 224. This was done via the in-built ‘imresize’ MATLAB function with the ‘bilinear’ interpolation method specified as one of its inputs.

Bilinear interpolation is the two-dimensional extension of linear interpolation. Linear interpolation is an averaging technique that is used to estimate the value of an unknown point which resides between two other known points [54]. In the context of this study, interpolation was used to add or remove pixels from the cropped images using the values of the neighbouring pixels in that area. As an example, consider the use of bilinear interpolation to upsize a 2 × 2-pixel image to a 4 × 4 image as shown in Figure 3.7 below. In addition to this, an example of a cropped CXR image before and after bilinear interpolation is shown in Figure 3.8.



10	20
30	40

10	12	17	20
15	17	22	25
25	27	32	35
30	32	37	40

Figure 3.7: Example of bilinear interpolation being used for upsizing [54].

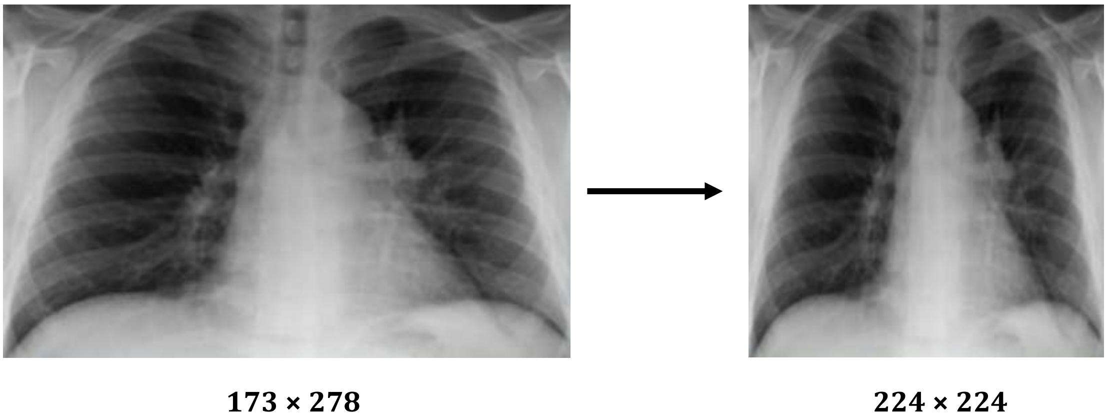


Figure 3.8: Cropped CXR image (left) after performing bilinear interpolation to obtain the required output size (right).

3.3.3 CLAHE

Image and contrast enhancement is a process used to improve the quality of the input CXR images with low contrast and its features via adjusting the image's pixel intensity level. This process has been proven to be very useful in the pre-processing phase of medical imaging analysis, as it aids in improving the classification performance of the ML models [55]. A common technique used for contrast enhancement is Histogram Equalisation (HE), which itself contains many different variations that function on the same basic principle as HE but with modifications. These different variations include AHE and CLAHE [56]. From the literature review, CLAHE was identified as the most promising technique for contrast enhancement on CXR images. However, it is crucial to develop fundamental understanding of HE and AHE, as CLAHE builds on these techniques.

The HE technique is used primarily because of its simplicity and effectiveness on any image [55]. It produces an equal and even spread of the image's intensity range so that the low contrast and dark parts of the image will be increased. The most common intensity values are spread out, achieving a stretched intensity range of the image. As a result, the image would be brighter and not previously available hidden details would be revealed [57]. Hence, the image's histogram would be more spread out and more linear as illustrated in Figure 3.9.

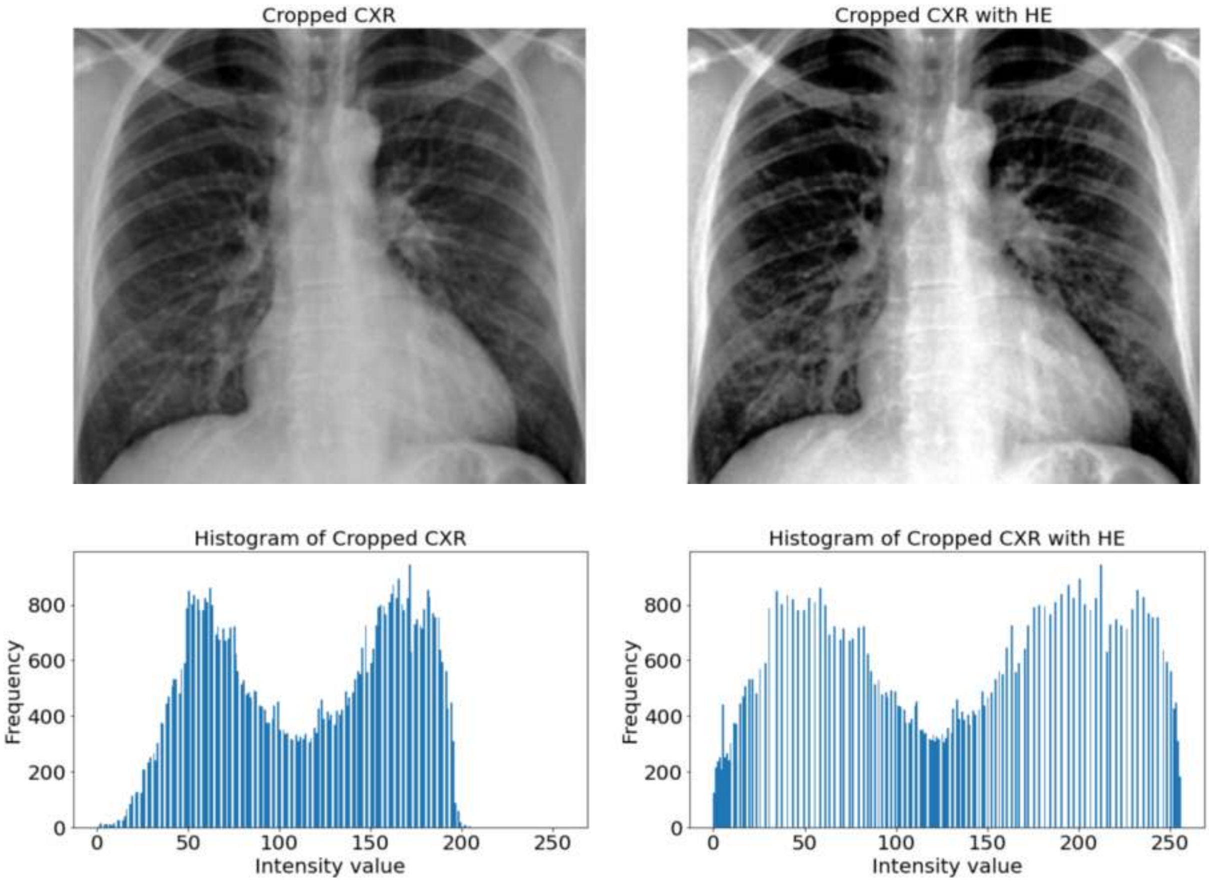


Figure 3.9: Cropped CXR image without HE (left) and with HE (right) and their respective image histograms.

As HE affects the global contrast of the image, it can produce unintended results on some images which may lead to images becoming less smooth and excessively lighter or darker at some regions [58]. To overcome this, AHE is an alternative method that can be used to improve the contrast by computing the HE in different regions of the image known as tiles, so that the pixel intensity level is better adjusted. By applying this method, the image would have better local contrast with more details near edges of each computed tile [57]. Figure 3.10 illustrates a comparison of the methods, HE and AHE and its effect on the image histograms.

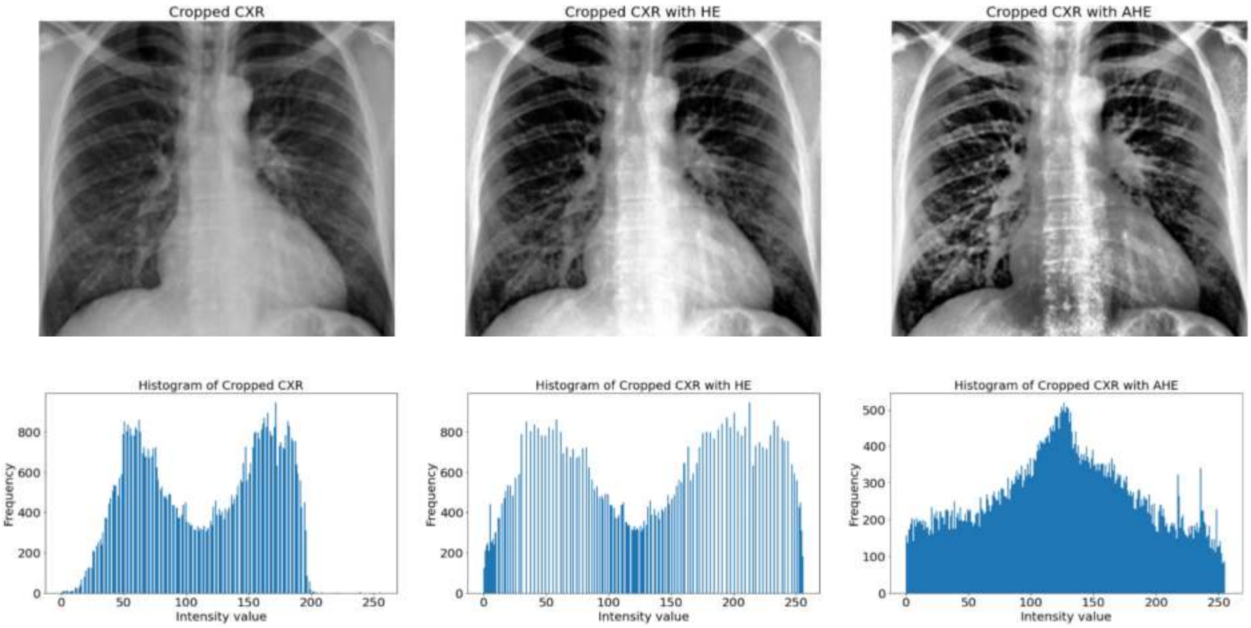


Figure 3.10: Comparison of HE and AHE effects on a cropped CXR image and their respective image histograms.

A limitation of the AHE method is that it can introduce amplification to the small noise present in each tile and creates an unrealistic image at times. To prevent this from occurring, a more novel technique, CLAHE is used [57]. This technique is developed based on AHE but unlike AHE, it applies a contrast limit which would eliminate the over-amplification.

The application of CLAHE on an image involves the following series of steps [55], [56]:

1. The input image first gets split into different sections which are called tiles and are not overlapped with each other.
2. Then for each section or tile, a histogram is created.
3. There are two key parameters that define and separate CLAHE from other techniques
    a. Clipping Limit, which is a pre-defined and selectable value that would be the limit set for the contrast.
    b. The tile size, which determines how many tiles are to be inside an image.
4. For each tile, the centre point would be selected as the sample point/pixel and after applying CLAHE, the neighbouring points inside that tile will have the same effect on them.

After completion of all steps, the final histogram will be a clipped histogram which has a maximum allowable contrast limit to ensure that the noise level is not amplified. Through trial and error, a clipping limit value of 10 and a tile size of 8 × 8 was determined, as it led to producing the best results for the CLAHE technique. In conclusion, CLAHE is a very popular method in medical image analysis and even though its computational time is greater than the other techniques, it produces the most promising result which is more detailed and smoother as illustrated in Figure 3.11.

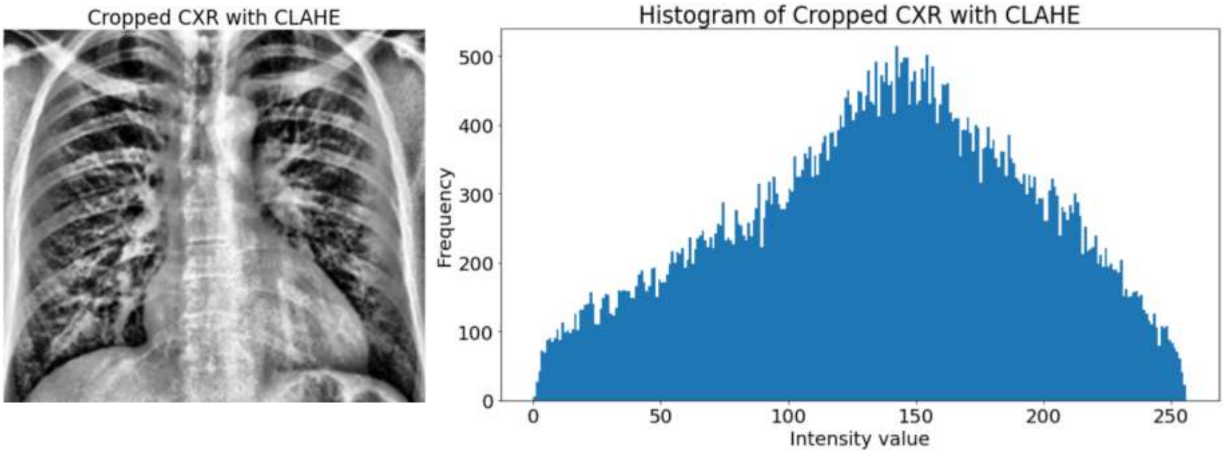


Figure 3.11: CLAHE applied on a cropped CXR image and its respective image histogram.

3.3.4 Denoising Filter

The application of denoising filters to reduce noise from the input CXR images is an essential part of the pre-processing stage. This is because CXR images present in the dataset may contain corrupted pixel values which can degrade the quality of those images. The main function of a noise filter is to reduce any noise and disturbance while preserving the quality and edge boundaries [59].

The median noise filter is one of the most effective and widely used technique for noise reduction. When applying the median filter on an image, if a corrupted pixel is located, that pixel will get replaced by a median value which is computed from all the surrounding pixel values. The surrounding pixels would follow a pattern known as window and be in a block size named window size which is pre-selected. Thus, within an image there are other windows which then altogether would cover the entire image.

The median filtering process starts by arranging the pixel numbers present inside each window in ascending order. Thereafter, if there is a corrupt pixel value inside a window, it will be replaced by the median value of that window [60], [61], as shown in Figure 3.12 [62]. In the case where the window size is an even value, the average of the two median numbers will be taken [60], [61].

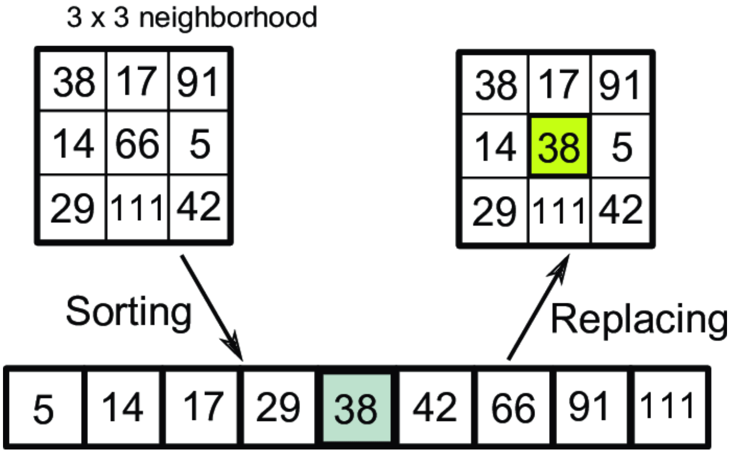


Figure 3.12: Median filtering process using 3 × 3 window size [62].

This series of tasks will be repeated for all the other remaining corrupted pixels and after it is completed, the final output will be a smooth and clear image that would aid in enhancing the classification performance. Figure 3.13 illustrates the effect of applying median filter onto a CXR image.

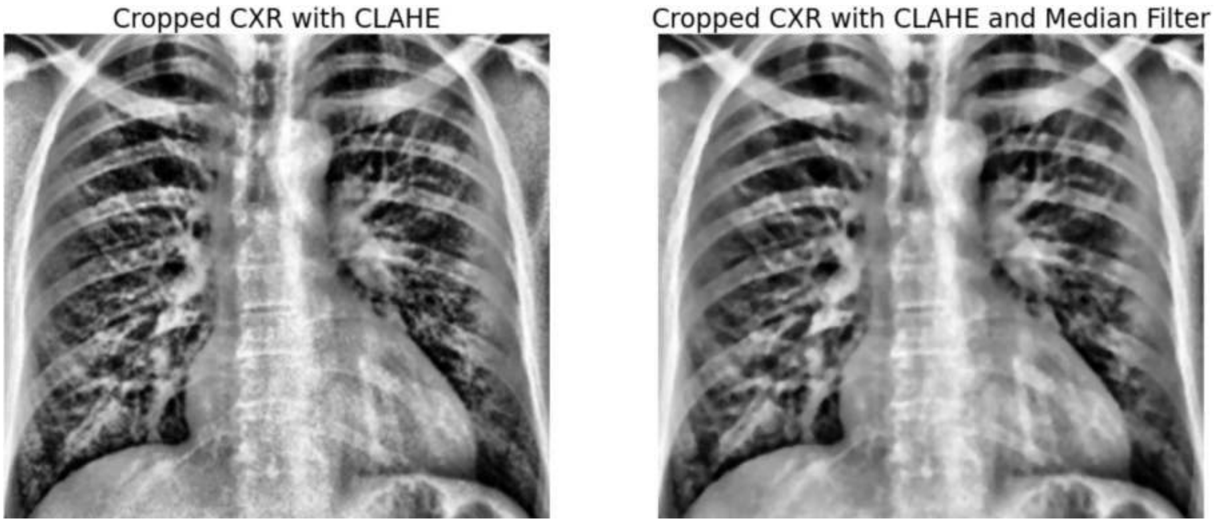


Figure 3.13: Comparison of original test image (left) and median filtered image (right).

3.4 Data Augmentation

Several studies have shown that data augmentation have been effective in increasing the performances in medical image classification problems [63], [64]. Data augmentation will generate additional images with slight variations using existing data. Data augmentation was performed on the training set to increase the training set size and to reduce overfitting. Overfitting occurs when the ML model performs too well on the training set but, is unable to generalise to the testing set which contains unseen data. As the ML models are exposed to a larger and a wider range of data during training process, the models are being forced to generalise.

Table 3.2 provides a summary of the data augmentation methods applied on the training set. For this project, the following data augmentation techniques were used: rotation, horizontal shifting, vertical shifting, and horizontal flipping. For rotation augmentation, each image was randomly rotated within a maximum range of ±20°. Horizontal and vertical shift augmentation were responsible for shifting the CXR image pixels horizontally or vertically by a certain fraction. In horizontal flip augmentation, the pixel columns in the CXR images were reversed (i.e., the image was flipped horizontally). An example of some of the augmented CXR images is provided in Figure 3.14.

Table 3.2: Data augmentation techniques applied on the training datasets.

Type of Augmentation	Value
Rotation range	20°
Width shift range	10%
Height shift range	10%
Horizontal flip	True

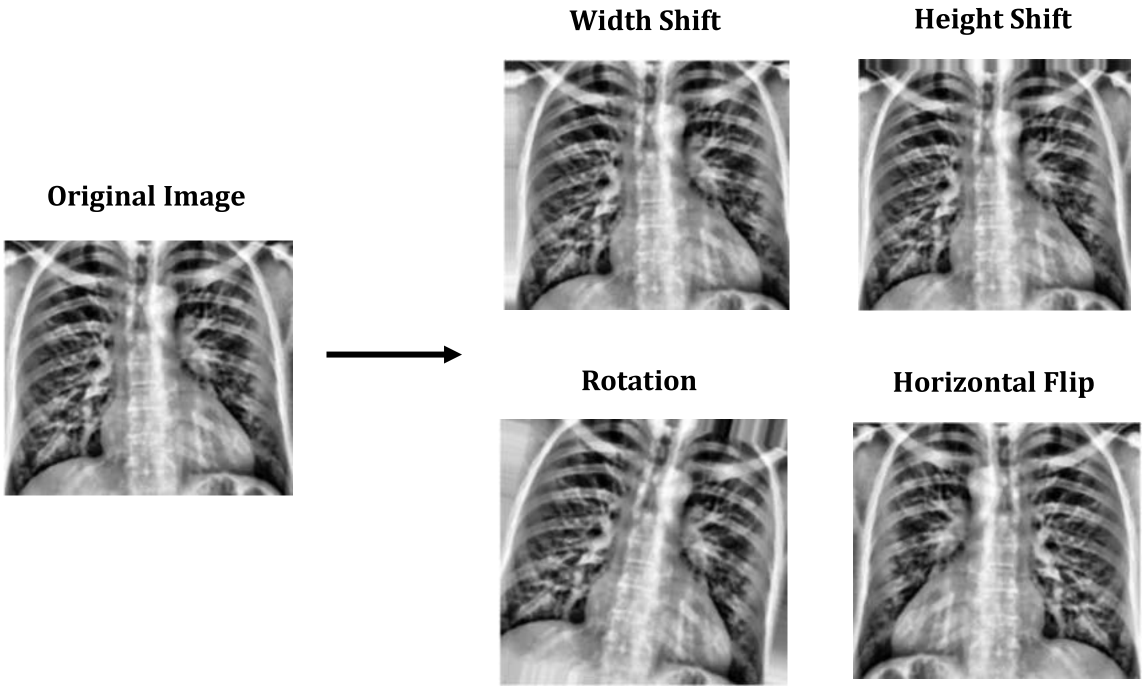


Figure 3.14: Examples of data augmentation.

3.5 Image Classification

According to a study conducted by Litjens *et al.* [65], [66], CNNs are one of the most common techniques for medical image analysis. The is likely due to the CNNs distinctive ability to extract and retain the complex features of the input images. A CNN architecture comprises of multiple sequential convolutional and pooling layers. A CNN's main aim is to learn and retain patterns of images associated with each class, while reducing the dimensionality of the image. For more information on the configuration of the CNN layers, refer to Appendix B. During this project, a custom CNN model was designed and implemented from scratch to perform image classification. The CNN model was implemented using the Keras library in the Python programming language. Several pre-trained CNN models were also implemented via TL using the Keras library.

3.5.1 Custom CNN Model

A diagram of the custom CNN architecture that was designed and implemented for CXR image classification is shown in Figure 3.15. The custom CNN architecture consists of 3 convolutional layers. This architecture takes in an input image of size 224 × 224. For each convolutional layer, the ReLU activation function is used. After each convolutional layer, a 2 × 2 max pooling layer is applied to reduce the dimensionality of the image. The first convolutional layer uses 16 3 × 3 kernel filters. The second convolutional layer uses 32 3 × 3 kernel filters. The final convolutional layer uses 64 3 × 3 kernel filters. To perform classification on the output obtained from the convolutional layers, multiple dense layers are used. The first dense layer consists of 512 neurons with the activation function being ReLU. The second dense layer consist of 3 neurons with a softmax activation function.

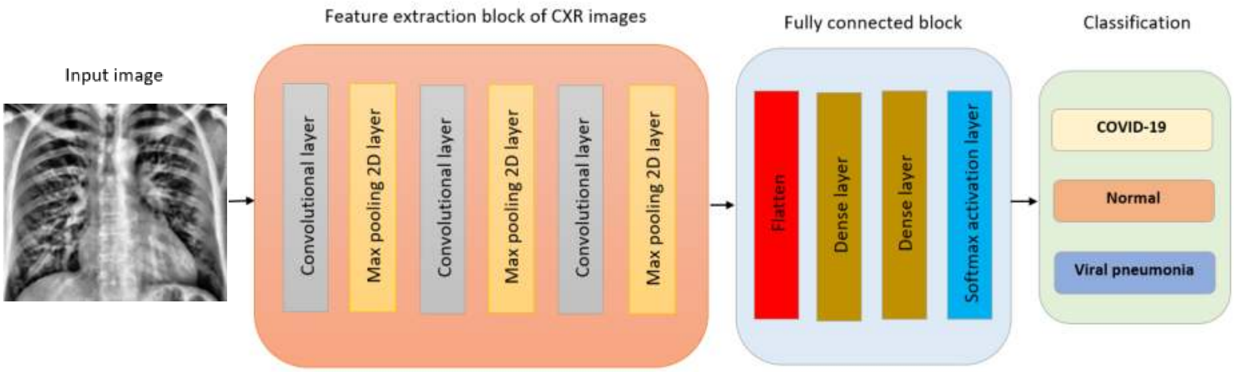


Figure 3.15: Architecture of the custom CNN model implemented.

3.5.2 CNN models with Transfer Learning

TL is a popular ML technique that uses pre-trained models (often with the same weights) to solve a new problem. TL is a well-researched technique in ML and is generally shown to result in increased model performance while minimising training time, overfitting and the number of parameters required [67].

This section provides a brief overview of the TL methods used in this project. The pre-trained models along with the pre-trained weights, made available in Keras Applications have been utilised for this project. For this project, the models, VGG-16, Inception-V3, and DenseNet-121 with pre-trained weights on the ImageNet database have been used. The ImageNet database is a large-scale hierarchical image database, consisting of 3.2 million images with over 5247 classes [68]. A study by Alexander Ke *et al.* [69], found that pre-trained models on ImageNet dataset yielded a significant boost in performance compared to other architectures for CXR interpretation.

3.5.2.1 VGG-16

The VGG-16 architecture consists of 16 convolutional layers. Similar to the proposed CNN architecture above, VGG-16 is also designed for input images of size 224 × 224 × 3. The first two layers start off by making use of 64 kernel filters of size 3 × 3 [70]. The subsequent convolutional layers make use of progressively more filters from 128 (for layers 3 and 4) to 256 (for layers 5 to 7) and finally to 512 (for layers 8 to 13) [70]. Convolutional layers 2, 4, 7, 10 and 13 also make use of max pooling which is used to select the most prominent features contained in those layers [71]. Dense layers consisting of 4096 neurons are then used for the 14th and 15th convolutional layers. These layers are then fed into a dense softmax layer which also utilises ReLU activation to classify these images. In total, this model consists of approximately 140 million hyperparameters [71].

3.5.2.2 Inception-V3

The Inception-V3 architecture is made up of 42 convolutional layers. Like VGG-16, Inception-V3 also makes use of max pooling layers throughout the model and terminates with a softmax layer for image classification [72]. Where these architectures differ however, is that Inception-V3 employs three inception-based modules which serve as a “multi-level function extractor” [22]. These modules compute convolutions of sizes 1 × 1, 3 × 3 and 5 × 5 for the same network layer, which are all then concatenated via a filter before being passed onto the next network layer [22].

3.5.2.3 DenseNet-121

In a DenseNet architecture, each convolutional layer is connected to each other layer via a Densely Connected Convolutional Network [73]. In this study, the DenseNet-121 model is used which contains 117 convolutional layers, 3 transition layers and 1 classification layer. This architecture consists of four DenseBlocks consisting of 6, 12, 24 and 16 dense layers respectively [74]. Each dense layer is made up of numerous convolutions of 1 × 1 and 3 × 3. Between each DenseBlock is a transition layer which consists of 1 × 1 convolution followed by a 2 × 2 average pool [75]. The softmax function is once again used to perform image classification in the final layer of this architecture [73] .

3.6 Evaluation Metrics

Evaluation metrics are used to measure the quality of the ML models. For classification type problems, the evaluation metrics, accuracy, recall, precision, and F1-score are commonly used [76].

Accuracy can be defined as the number of correct predictions over the total number of predictions. The accuracy score can be defined using the following expression:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \qquad \textit{Eq. (3.2)}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives [76]. The accuracy score should not be used alone to evaluate a model's performance as it can produce misleading results on an imbalanced dataset. Therefore, it is important to consider other evaluation metrics.

Recall is another evaluation metric used to measure the ML model's ability to determine positive cases. Recall can be defined as the number of positive class predictions over the combined number of true positives and false negatives [76]. Recall be defined using the following expression:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \qquad \textit{Eq. (3.3)}$$

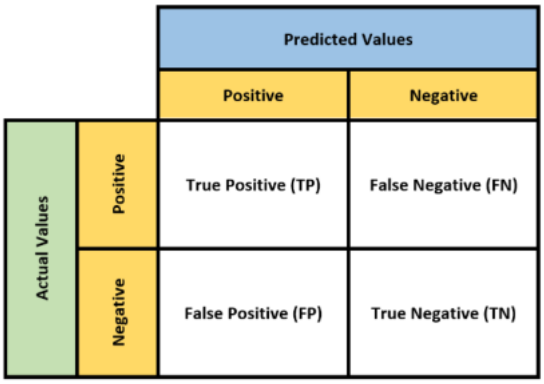
Precision is an evaluation metric that measures the number of true positives over the total number of positive predictions [76]. Mathematically, it can be expressed using the definition below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \qquad \textit{Eq. (3.4)}$$

An F1-score is an evaluation metric that considers both recall and precision. Therefore, F1-score can be defined to be the harmonic mean of recall and precision [76]. It can be defined mathematically using the expression shown below,

$$\text{F1} - \text{score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \qquad \textit{Eq. (3.5)}$$

A confusion matrix allows to visualise the performance of the ML model on a test dataset. Figure 3.16 shows the format of a confusion matrix. For each class, a confusion matrix provides information about the number of true positives, true negatives, false negatives, and false positives [76]. Using the confusion matrix, the accuracy, recall, precision, and F1-scores can be computed.



		Predicted Values	
		Positive	Negative
Actual Values	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 3.16: Layout of a confusion matrix.

Although at first glance, the accuracy score might seem like a good metric to evaluate the performance of each ML model, this metric can be easily misinterpreted. For example, it is possible for the ML model to correctly classify all (or most of) the COVID-19 CXR images but to also incorrectly (or falsely) classify normal and viral pneumonia CXR images as COVID-19 too. In this case, the accuracy may be high for the COVID-19 class but it would not necessarily mean that the ML model is capable of distinguishing COVID-19 from the normal and viral Pneumonia classes. On the other hand, the recall and precision metrics take false positive and false negative classifications into account. These two metrics are then used to calculate the F1-score which is a less biased metric that can be used to evaluate the ML models. For this reason, the F1-score will serve as the primary evaluation metric for comparing the ML models and determining the best performing model.

3.7 Cross-Validation

Cross-validation is a resampling procedure often employed for measuring a ML model's ability to generalise on unseen data. Cross-validation uses different partitions of the data to train and validate a model on different iterations. A simple and a commonly used cross-validation method is the holdout cross-validation technique, which splits the entire data into independent training and validation sets. A limitation of this technique is that the performance of the ML models is highly dependent on which samples end up in the training and validation sets. Therefore, the performance may vary depending on how the samples are distributed in the training and validation sets [77]. Using this technique, the ML models will also produce poor performance if the dataset size is not sufficiently large.

To address these issues, the K-Fold cross-validation technique is used for the validation phase. In this technique, the CXR images from the dataset are randomly split into K smaller subsets. Thereafter, the hold-out technique is repeated K number of times. At each iteration, a single subset is used for validation purposes and the remaining K-1 subsets are used for training the ML models. After a single iteration, the evaluation score on the validation set is retained. This process is iterated until each unique subset has been used as the validation set. After the cross-validation procedure is completed and the evaluation score for each K iteration is recorded, the average of all these evaluation scores is computed to determine the final validation performance of the ML models [78], [79]. Through this technique, every subset has the opportunity to be used as the validation set as well as being used as part of the training set. The performance of the ML models is validated on multiple subsets of the entire data. Therefore, K-Fold cross-validation aids in determining a more accurate estimate of ML model prediction.

For training and validating the ML models using K-Fold cross-validation, 80% of the total data was used. Therefore, a total of 1076 CXR images were used for training and validating purposes. For the K parameter involved in K-Fold cross-validation, a value of 4 was set, as it allowed an even split of data. This meant that, the training set was split into 4 smaller subsets, where each subset contained a total of 269 images. For each of the 4 iterations, the ML models were trained using 3 of the subsets as training set, and the remaining subset was used to validate the performances of the ML models. Therefore, for each iteration of the K-Fold cross-validation, 807 CXR images would be used to train the ML models and 269 CXR images would be used to validate the ML models. A visualisation of this procedure is shown in Figure 3.17.

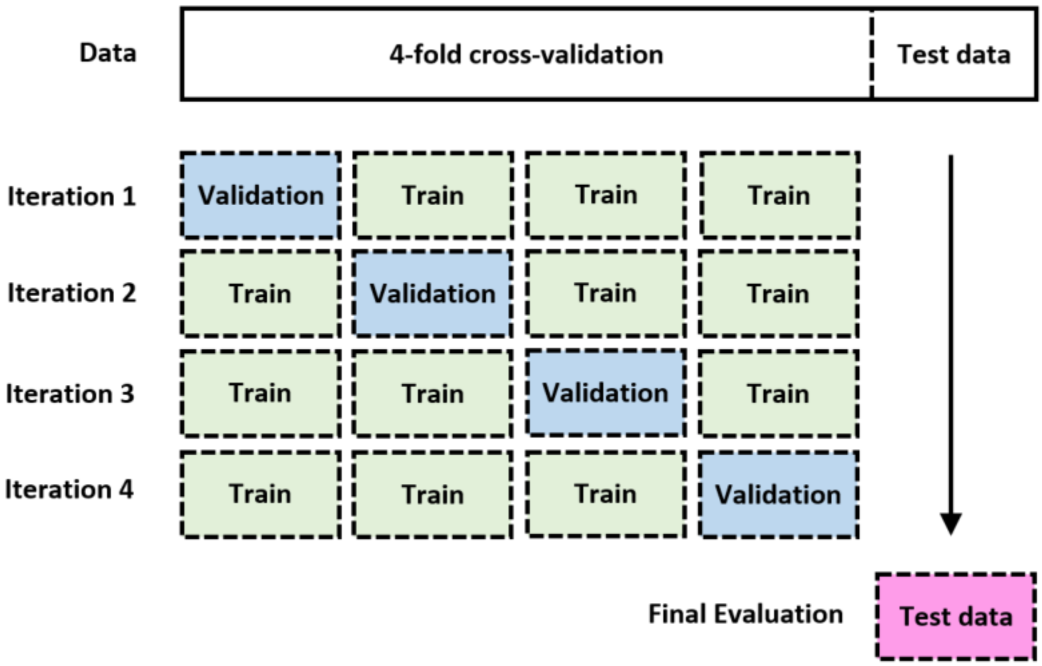


Figure 3.17: Illustration of K-Fold cross-validation.

In order to obtain an unbiased performance, the test set which contained 'un-seen' data was used for the final evaluation of the ML models. The test set contained a total of 807 CXR images, which equates to 269 images for each class. The distribution of the train, validation, and test sets for the dataset is summarised in Table 3.3.

Table 3.3: Distribution of the training, validation, and testing sets.

Set	**Count**			**Percentage**
	COVID-19	**Normal**	**Viral Pneumonia**	
Train (for each iteration)	807	807	807	60%
Validation (for each iteration)	269	269	269	20%
Test	269	269	269	20%
Total	1345	1345	1345	100%

3.8 Verification

Over the recent years, XAI approaches have gathered the attention of many researchers in the AI field. XAI approaches can provide insight into the key features that are used by ML models to classify data [80]. In this project, XAI approaches are used to determine whether the ML models are using the correct features (i.e., the lungs) for image classification. It should be noted that the XAI approaches do not influence the results obtained from the ML models and that this approach was only implemented to verify that the relevant CXR features are used by the ML models for image classification.

A popular XAI approach among CNN models is Grad-CAM. This technique makes use of the spatial information that is contained within the convolutional feature maps to determine which neurons were used for image classification. The Grad-CAM technique then produces a heatmap highlighting the position of these neurons on the image. There are three steps involved in implementing the Grad-CAM technique [81]. First, the gradient of the class score, y^C, is computed with respect to the feature map activations A^k of a convolutional layer, i.e., $\frac{\partial y^C}{\partial A^k}$. It should be noted that the class score term, y_C, refers to the raw output of the neural network prior to the final softmax layer which converts this value to a probability [80], [81]. For a 2D image, there are k convolutional feature maps, each with a height m and width n. In the second step, each neuron is assigned a weighting factor by taking the global average pool along the width and height dimensions (in pixels) of the feature map A^k, i.e.,

$$\beta_k^C = \frac{1}{N}\sum_j^n \sum_i^m \frac{\partial y^C}{\partial {A^k}_{ij}}, \qquad \textit{Eq. (3.6)}$$

where N is the total number of pixels in the feature map [80], [81]. In the final step, the Grad-CAM heatmap is generated by taking the weighted sum of the feature map activations A^k and the neuron weighting factor β_k^C. This is then passed into a ReLU function,

$$G^C_{Grad-CAM} = \mathrm{ReLU}\left(\sum_k \beta_k^C A^k\right). \qquad \textit{Eq. (3.7)}$$

3.9 Summary

In summary, four ML methods were used for CXR classification. These models consisted of three pre-trained CNN architectures (VGG-16, InceptionNet-V3 and DenseNet-121) and one custom-made CNN model. Data augmentation was employed to increase the training set size and to reduce overfitting of ML models. The ML models implemented were cross-validated using the technique, K-Fold cross-validation. The performance of each model was evaluated separately on both the unprocessed and pre-processed datasets using a series of evaluation metrics. For the pre-processed dataset, a variety of pre-processing techniques including ROI extraction, CLAHE and median filter were used to improve the quality of the CXR images.

Chapter 4: Results

4.1 Introduction

This chapter will summarise the major results obtained from this project. In this study, the CNN architectures will be evaluated on a test set according to the following performance metrics: accuracy, precision, recall and F1-score. The F1-score was selected as the primary evaluation metric to compare the performances of the ML models. To help visualise the performance of these ML models, a series of confusion matrices were also created.

4.2 Unprocessed Data

This subsection summarises the performance of the CNN models when the unprocessed dataset was used for training and evaluation. The performance of the models is evaluated using the set of performance metrics stated above, as well as through a set of confusion matrices which visualises the classification performance of the ML models.

4.2.1 Evaluation of CNN Architectures

Table 4.1 show the performance evaluation of the ML models on the unprocessed data. The tables include the precision, recall and F1-score values obtained for each class and the overall accuracy of the ML models when evaluated on the testing set.

Table 4.1: Performance evaluation of ML models for unprocessed data.

Model	**Classes**	**Precision**	**Recall**	**F1-score**	**Average Testing Accuracy (%)**
Custom CNN	COVID-19	0.78	0.74	0.76	**81**
	Normal	0.79	0.70	0.74	
	Viral Pneumonia	0.85	0.98	0.91	
	Average	**0.81**	**0.81**	**0.81**	
VGG-16	COVID-19	0.99	0.98	0.98	**88**
	Normal	0.76	0.95	0.84	
	Viral Pneumonia	0.94	0.71	0.81	
	Average	**0.90**	**0.88**	**0.89**	
Inception-V3	COVID-19	0.92	0.97	0.94	**82**
	Normal	0.73	0.82	0.77	
	Viral Pneumonia	0.81	0.67	0.73	
	Average	**0.82**	**0.82**	**0.82**	
DenseNet-121	COVID-19	0.98	0.95	0.97	**88**
	Normal	0.77	0.94	0.85	
	Viral Pneumonia	0.93	0.75	0.83	
	Average	**0.89**	**0.88**	**0.88**	

According to Table 4.1, VGG-16 and DenseNet-121 tied for the highest average testing accuracy (88%) for the three-way classification of COVID-19, viral pneumonia and normal CXR images. These models also produced the highest average values for precision, recall and F1-score across all the three classes. Both models, DenseNet-121 and VGG-16, obtained a near perfect score (95%+) for precision, recall and F1-score for the COVID-19 class. In particular, the high F1-scores indicate that, the DenseNet-121 and VGG-16 models produced a low number of false positives and low number of false negatives for the COVID-19 class. Overall, both models were successful in detecting the CXR images of COVID-19 patients and differentiating them from normal and viral pneumonia patients.

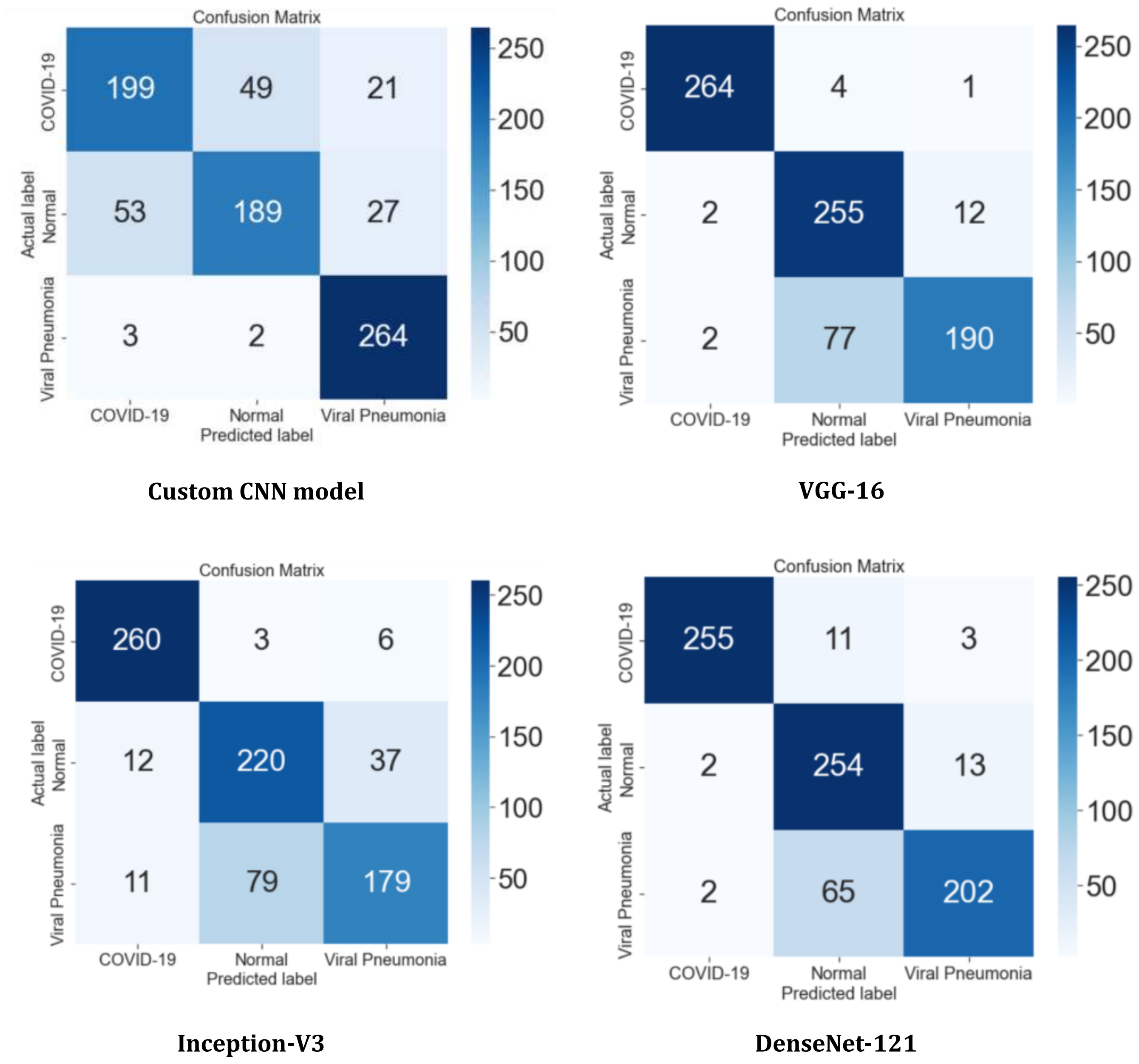
For the normal class, DenseNet-121 and VGG-16 obtained precision scores of 77% and 76% respectively. This indicates that, both models produced a moderate number of false positives. Both models produced high scores for recall, indicating a low number of false negatives produced by the models for the normal class. This indicates that, for the CXR images that belonged to the normal class, both models were highly successful in correctly identifying the CXR images of normal patients. The DenseNet-121 and VGG-16 models produced decent results for the F1-score for the normal class. This implies that the models, although not highly successful, are still capable of identifying CXR images of normal patients and differentiating them from patients with COVID-19 and viral pneumonia.

Both DenseNet-121 and VGG-16 models produced high scores (92%+) for precision for the viral pneumonia class. This suggests that the models produced a low number of false positives. This means that the models do not incorrectly identify CXR images as belonging to viral pneumonia class when they are not actually CXR images of viral pneumonia patients. The models obtained low scores for recall, indicating a high number of false negatives produced by the models. This means that the models are not highly capable of identifying all CXR images of viral pneumonia patients.

The Inception-V3 and custom CNN models produced similar results for testing accuracy, 82% and 81%, respectively. Both Inception-V3 and custom CNN models also produced an average precision, average recall, and average F1-score of 82% and 81%, respectively. In this case, the Inception-V3 model slightly outperformed the custom CNN model (by 1%) in terms of average testing accuracy, average precision, average recall, and average F1-score. Despite this, when taking a closer look at the metrics for the individual classes it is evident that the custom CNN model largely outperformed all the other models in terms of recall and F1-score for the viral pneumonia class. As such, it appears that the custom CNN model is better at correctly identifying CXR images with viral pneumonia and differentiating them from the other classes. The Inception-V3 and the remaining TL models, still outperformed the custom CNN model in terms of identifying CXR images of COVID-19 patients. This is evident by the high scores of precision, recall and F1-scores obtained for the TL models in comparison with the custom CNN model. Nevertheless, due to their sub-par performance, evident by their low average F1-score values, both Inception-V3 and custom CNN models were considered to be less effective at classifying CXR images for the three classes.

4.2.2 Confusion Matrices

Figure 4.1 depicts the confusion matrices obtained from the ML models when evaluated on the unprocessed testing dataset. These confusion matrices offer a better look at how each ML model classified the CXR images.



Custom CNN model

VGG-16

Inception-V3

DenseNet-121

Figure 4.1: Confusion matrices for unprocessed dataset.

From Figure 4.1, the VGG-16, Inception-V3 and DenseNet-121 models were relatively successful in correctly classifying CXR images with COVID-19 due to the low number of false negatives i.e., the number of COVID-19 CXR images that were incorrectly (or falsely) classified as either normal or viral pneumonia. This was reflected earlier in Table 4.1 with these models obtaining high values for the recall metric for the COVID-19 class. These three models were also capable of distinguishing the normal and viral pneumonia CXR images from COVID-19 due to the low number of false positives i.e., the number of non-COVID-19 CXR images that were incorrectly classified as COVID-19. This observation corresponded to the high precision values for the COVID-19 class for the three models. On the other hand, the custom model struggled to both correctly classify COVID-19 CXR images and to distinguish them from the other classes. In particular, the custom CNN model had a hard time distinguishing the COVID-19 from normal class as it incorrectly classified 49 COVID-19 CXR images as normal and another 53 normal CXRs as COVID-19. This resulted in the lower values for recall and precision for these two classes as shown in Table 4.1 above.

The VGG-16, Inception-V3 and DenseNet-121 models also struggled to differentiate the normal CXR images from the viral pneumonia class and vice versa. This is evident by the large number of false negatives obtained for the viral pneumonia class for all three models and the large number of false positives obtained for the viral pneumonia class for the Inception-V3 model. In contrast to this, the custom CNN model had an easier time separating and correctly classifying the viral pneumonia CXR images. However, this model appears to be heavily biased towards the viral pneumonia class as many normal and COVID-19 CXR's were incorrectly classified as viral pneumonia.

4.3 Pre-processed Data

This subsection summarises the performance of the CNN models when the pre-processed dataset was used for training and evaluation. The performance of the models is evaluated using the same set of performance metrics (precision, recall, F1-score, and testing accuracy), as well as through a set of confusion matrices which illustrates the distribution of the CXR images according to how they were classified by the ML models.

4.3.1 Evaluation of CNN Architectures

Table 4.2 summarises the performance metrics for each ML model when the pre-processed dataset was used. This table contains the individual precision, recall and F1-score for each class and ML model. The average precision, recall, F1-score, and testing accuracy for each ML model is also included.

Table 4.2: Performance evaluation of ML models for pre-processed data.

Model	**Classes**	**Precision**	**Recall**	**F1-score**	**Average Testing Accuracy (%)**
Custom CNN	COVID-19	1.00	0.88	0.93	**86**
	Normal	0.75	0.92	0.83	
	Viral Pneumonia	0.87	0.77	0.82	
	Average	**0.87**	**0.86**	**0.86**	
VGG-16	COVID-19	0.99	0.93	0.96	**88**
	Normal	0.77	0.96	0.86	
	Viral Pneumonia	0.93	0.77	0.84	
	Average	**0.90**	**0.89**	**0.89**	
Inception-V3	COVID-19	0.89	0.80	0.84	**87**
	Normal	0.83	0.81	0.82	
	Viral Pneumonia	0.88	0.99	0.93	
	Average	**0.87**	**0.87**	**0.86**	
DenseNet-121	COVID-19	0.89	0.90	0.90	**91**
	Normal	0.84	0.89	0.86	
	Viral Pneumonia	1.00	0.92	0.96	
	Average	**0.91**	**0.90**	**0.91**	

According to Table 4.2 above, the best performing model on the pre-processed dataset is DenseNet-121. This model obtained the highest average testing accuracy (91%) and average F1-score (91%). The next best performing model was VGG-16, with an average testing accuracy of 88% and an average F1-score of 89%. This was closely followed by Inception-V3 and finally the custom CNN model which had average testing accuracies of 87% and 86% respectively, and an average F1-score of 86% for both models.

For identifying CXR images of patients with COVID-19, the VGG-16 model produced the best performance as it obtained the highest F1-score of 96%. This meant that for the VGG-16 model, a low number of false positives and false negatives was obtained for the COVID-19 class. Hence, the VGG-16 model was the best at being able to successfully identify CXR images of COVID-19 patients and differentiate them from normal and viral pneumonia patients.

The DenseNet-121 and VGG-16 models outperformed all the other models in detecting CXR images of normal patients and differentiating them from other classes. This is because, both models obtained the highest F1-score of 86%. This was closely followed by the custom CNN and Inception-V3 models, which obtained F1-scores of 83% and 82%, respectively.

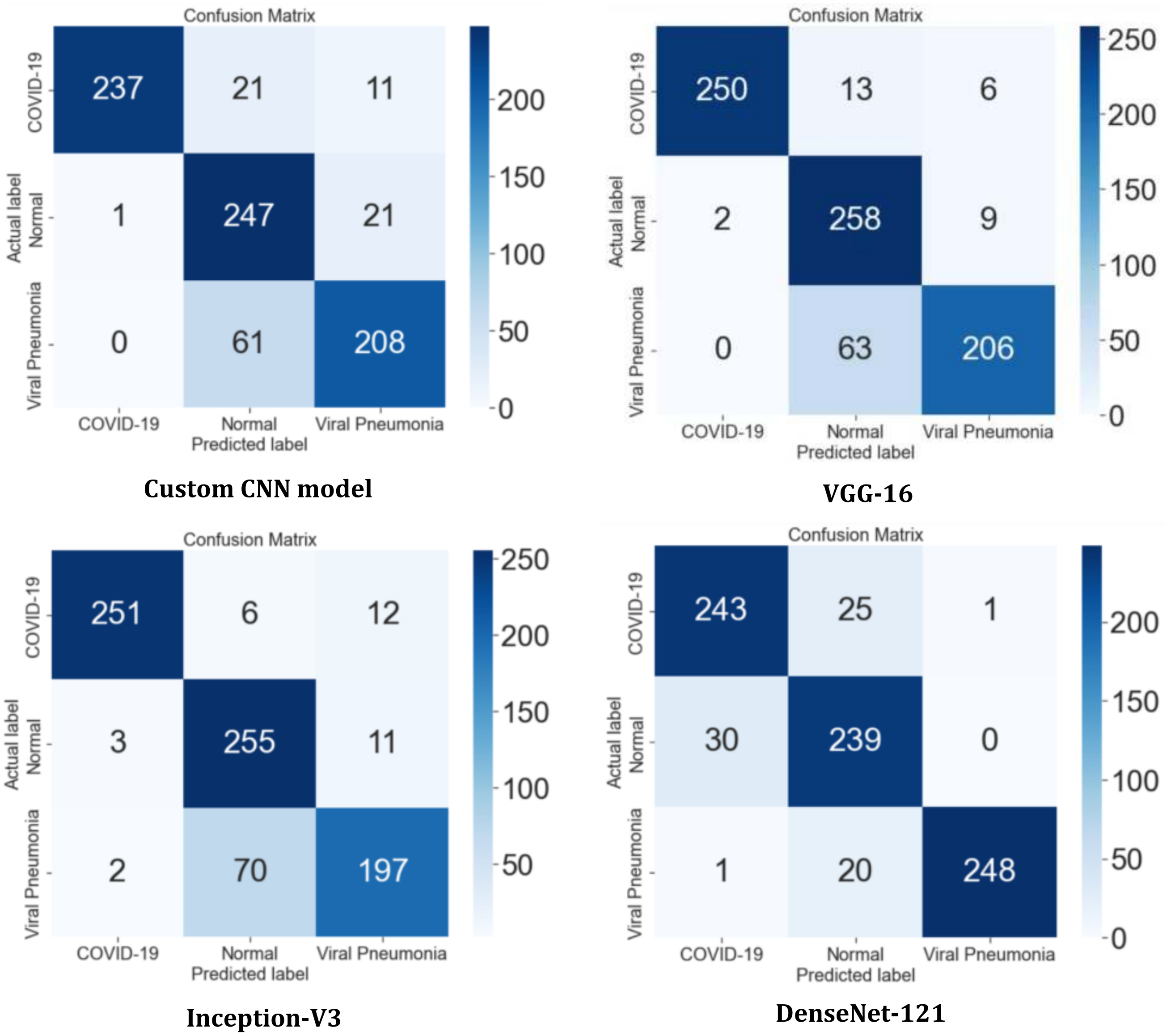
For the viral pneumonia class, the DenseNet-121 model outperformed all the other models, as it obtained the highest F1-score of 96%, closely followed by the Inception-V3 model which obtained 93%. Both custom CNN and VGG-16 models showed struggle in detecting CXR images of viral pneumonia patients and differentiating them from COVID-19 and normal patients. This is evident from the lower F1-score values obtained by these two models for the viral pneumonia class.

Overall, the DenseNet-121 model demonstrated the best ability in detecting CXR images of normal and viral pneumonia patients and differentiating them from other classes, as it obtained the highest F1-score values for those two classes. For the COVID-19 class, although the model only obtained the third highest F1-score value (closely behind the VGG-16 and custom CNN models), an F1-score of 90% still indicates that the model can successfully identify CXR images of COVID-19 patients and differentiate them from normal and viral pneumonia patients. Therefore, the DensetNet-121 model was selected to be the best performing model.

When comparing to the unprocessed dataset, all the ML models showed improvements in average testing accuracy and average F1-score. The custom CNN and Inception-V3 models showed the largest improvement in average testing accuracy by an increase of 5%. Similarly, there was an improvement in the average F1-score for custom CNN and Inception-V3 models by 5% and 4%, respectively. The performance of the VGG-16 model did not improve or worsen, as it obtained the same average testing accuracy and average F1-score of 88%, for both unprocessed and pre-processed data. The best performing model, DenseNet-121, showed improved average testing accuracy and average F1-score, as it increased by 3% for both metrics. Therefore, it can be stated that pre-processing the data enhances the classification performance of ML models. In comparison with the unprocessed dataset, for each model, there was also less variation in the recall and precision metrics across the three classes. In addition to this, the average recall and precision metrics only differed by at most 1% for each model. This implies that these ML models were less biased to one particular class as was the case with the unprocessed dataset.

4.3.2 Confusion Matrices

The confusion matrices for each ML model when the pre-processed dataset was used is shown in Figure 4.2.



Custom CNN model

VGG-16

Inception-V3

DenseNet-121

Figure 4.2: Confusion matrices for the pre-processed dataset.

Comparing the results in Figure 4.2 above to the confusion matrices in section 4.2.2, the DenseNet-121 model showed considerable improvement when the pre-processed data was used. In this case, the model was better able to distinguish the viral pneumonia and normal CXR images, as evident from the reduced number of false negatives in the viral pneumonia class (from 67 to 21) and the presence of only one false positive. However, this weakened the model's ability to detect and distinguish the COVID-19 CXR images from the other classes as evident by the larger number of false positives and false negatives, as well as the reduced recall and precision metrics for the COVID-19 class as seen in Table 4.2 above. Despite this, there was still a net increase in the classicisation performance for the DenseNet-121 model when the pre-processed data was used.

The other models displayed similar characteristics, with most of them exhibiting a large number of false negatives for the viral pneumonia class. This behaviour was also present in the unprocessed dataset for the Inception-V3 and VGG-16 models. As such, the pre-processing techniques were less effective on the Inception-V3 and VGG-16 models. According to Tables 4.1 and 4.2, the custom CNN model showed the greatest improvement when the pre-processed data was used, with both the average testing accuracy and average F1-score increasing by 5%. This is further evident by the confusion matrix for the custom CNN model which showed an increase in the number of true positives for the COVID-19 and normal classes. Therefore, with the pre-processed data, the custom CNN model is more capable of correctly identifying CXR images of COVID-19 and normal patients.

4.4 Summary

The results demonstrate that both the designed and pre-trained CNN models can successfully extract important features from the CXR images to perform classification. The custom CNN model implemented produced the worst performance for CXR classification on both the unprocessed and pre-processed dataset. The DenseNet-121 and VGG-16 models produced the highest average testing accuracy and average F1-score on the unprocessed dataset. However, when the pre-processed data was used to train and test the ML models, the DenseNet-121 model outperformed the VGG-16 model. Overall, the ML models performed better on the pre-processed data, with the average F1-score increasing by up to 5%. Overall, the DenseNet-121 model produced the best performance in classifying CXR images of patients.

Chapter 5: Discussion

5.1 Introduction

In this section, the performance of the ML models was evaluated for both the unprocessed and pre-processed datasets. The results for the best performing model from this study was also compared to the results from the previous studies listed in the literature review. The initial project objectives were also reviewed and the outcomes for each objective summarised. Finally, the limitations in this study and their impact on the results were discussed.

5.2 Unprocessed vs Pre-Processed Results

The average F1-Score for each ML model on both the unprocessed and pre-processed datasets is summarised in Figure 5.1 below.

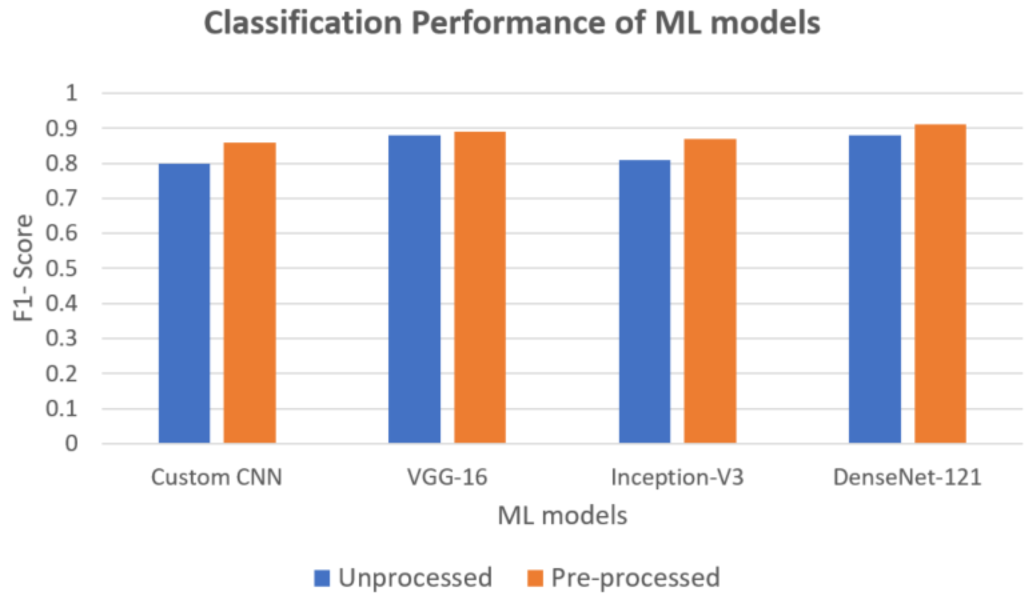


Figure 5.1: Classification Performance for ML models on the unprocessed and pre-processed datasets.

Overall, a higher average F1-score was obtained on the pre-processed dataset for all the ML models considered in this study. The best performing models for the unprocessed datasets were VGG-16 and DenseNet-121, with both obtaining an 88% for the average F1-score. For the pre-processed dataset, the best performing model was DenseNet-121, which obtained an F1-score of 91%. The performance of the DenseNet-121 is likely attributed to its large number of convolutional layers (121), which exceeded the convolutional layers used by the other models in this study. Although, increasing the number of convolutional layers does not necessarily result in better classification performance, it does help the model to extract and learn more complex features from the images. Since CXR images contain a substantial amount of information, the ML models will likely need to extract very specific features for accurate classification. It should also be noted that the DenseNet-121 model also yielded the longest computational time, which adds further emphasis to this model's higher complexity compared to the other ML models in this study.

Figure 5.2 depicts a sample of CXR images that are overlayed with a heatmap that was generated using the Grad-CAM XAI approach. Although, a model's performance can be measured using evaluation metrics such as F1-score, it is important to develop insight on what regions on the image the model focuses on for prediction. This is because, even if the evaluation metrics produce a high score, the model could be learning other patterns not related to the disease if the model uses information from other regions. The Grad-CAM approach highlights key regions in the CXR image that were used for ML model predictions. These heatmaps were generated for the best performing ML model, DenseNet-121, for the unprocessed dataset.

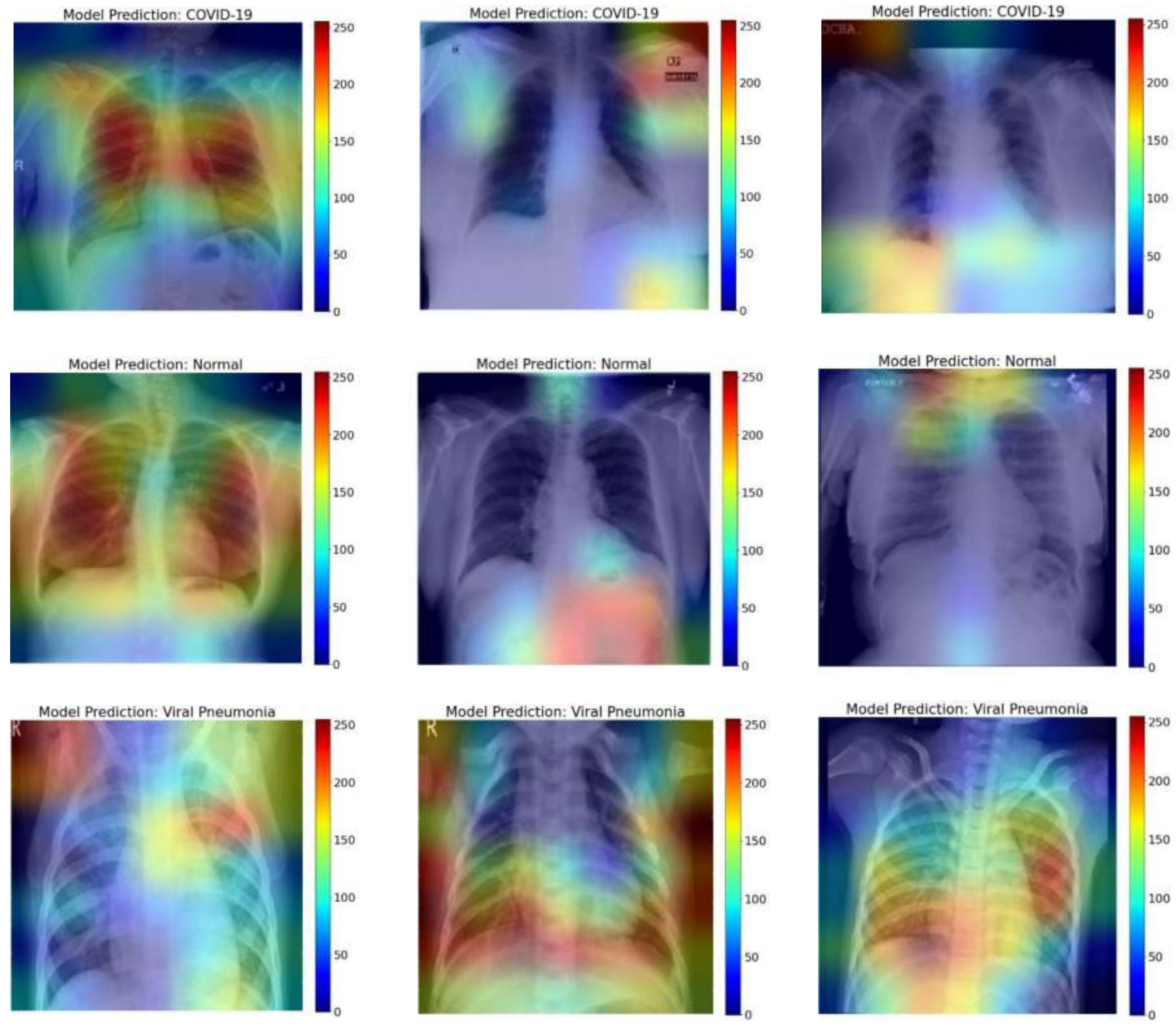

Model Prediction: COVID-19
Model Prediction: COVID-19
Model Prediction: COVID-19
Model Prediction: Normal
Model Prediction: Normal
Model Prediction: Normal
Model Prediction: Viral Pneumonia
Model Prediction: Viral Pneumonia
Model Prediction: Viral Pneumonia
250
200
150
100
50
0

Figure 5.2: Key features used for CXR classification for the DenseNet-121 model for the unprocessed dataset.

Referring to Figure 5.2, there are multiple instances where the ML model classified CXRs based on irrelevant information such as random text, or even the shoulder and abdominal regions. This was especially prominent in CXRs that were taken using a larger scanning window, which made the lungs appear smaller in the image. This suggests that the ML model is not classifying the CXRs according to features related to COVID-19 or viral pneumonia but is instead learning other patterns pertaining to this particular dataset. That being said, the ML model did use the lung region for CXRs classification where the lungs were more prominent in the image. Overall, it is evident that the ML model is not as effective in using the lungs for CXR classification when the lungs are not the primary focus of the image. Although the F1-score obtained on the unprocessed dataset for the DenseNet-121 model was high (88%), it can be stated that model is not highly reliable and is not suitable for practical world applications, as the model did not use prominent lung features for classification.

The Grad-CAM heatmaps for the pre-processed dataset for the DenseNet-121 model are depicted in Figure 5.3.

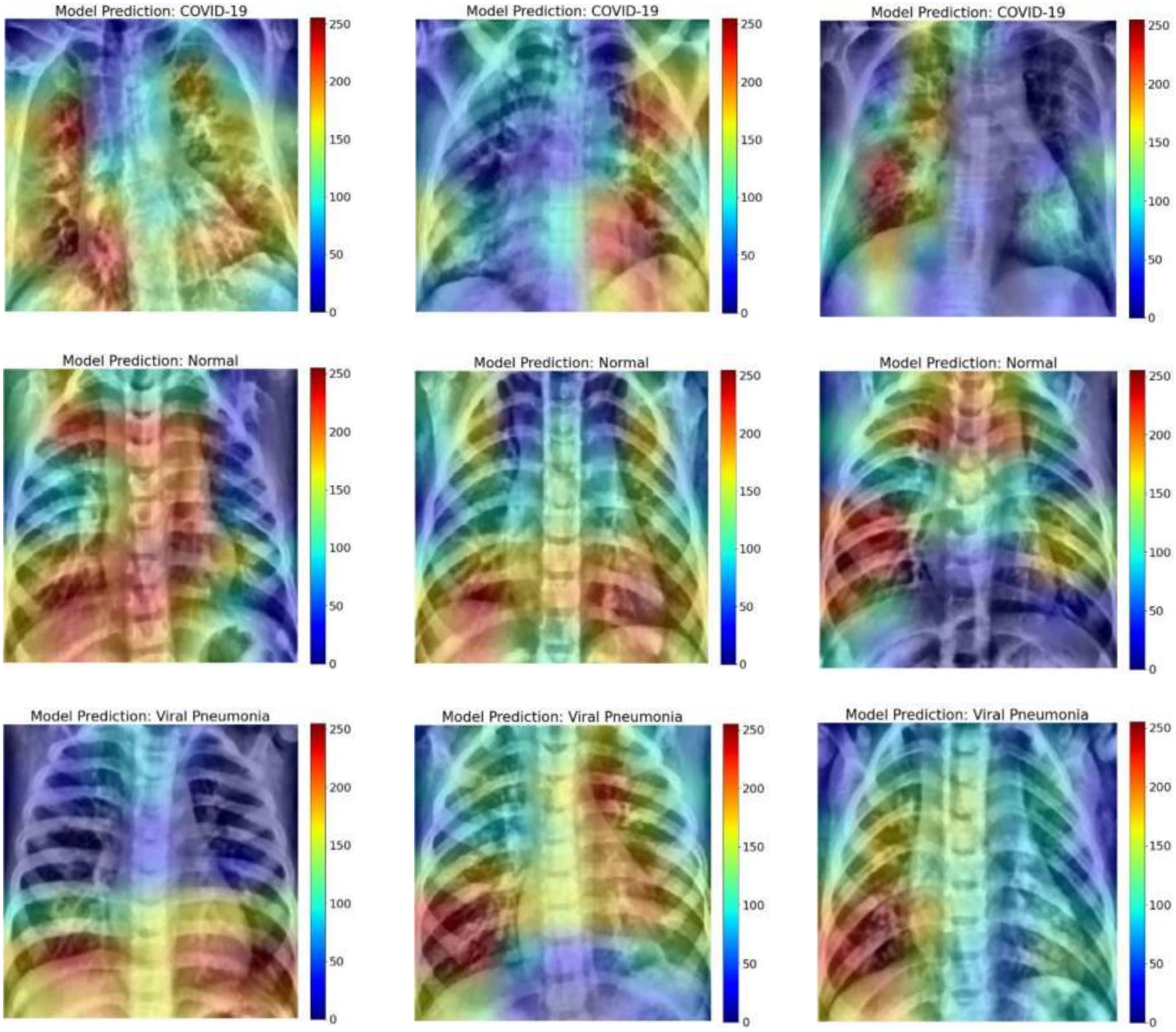

Model Prediction: COVID-19
Model Prediction: COVID-19
Model Prediction: COVID-19
Model Prediction: Normal
Model Prediction: Normal
Model Prediction: Normal
Model Prediction: Viral Pneumonia
Model Prediction: Viral Pneumonia
Model Prediction: Viral Pneumonia
250
200
150
100
50
0

Figure 5.3: Key features used for CXR classification for the DenseNet-121 model for the pre-processed dataset.

From Figure 5.3, the ML model appears to focus more on the lungs for CXR classification for the pre-processed dataset when compared to the unprocessed dataset. Furthermore, it appears that certain sections of the lungs (such as the lower portions) are more frequently used for classification than others. This could indicate that the ML model is better at identifying and using certain features pertaining to the lungs for CXR classification. In fact, one of the key features of COVID-19 in CXR images is the presence of Ground Class Opacity (GGO) in the lower region of the lungs. For more information on GGO refer to Appendix B. As such, not only has the inclusion of pre-processing techniques improved the performance of the ML models, but it has also increased the reliability and 'correctness' of the ML models for CXR classification.

5.3 Comparison with Previous Studies

The best performing model in this project was DenseNet-121 for the pre-processed dataset which obtained an average testing accuracy and F1-score of 91%. According to the results from the literature review, this model appears to slightly underperform compared to the models from previous 3-way classification studies. These studies typically reported accuracies and F1-scores greater than 95% despite minimal data pre-processing [33], [35]. However, the vast majority of these studies have used an imbalanced dataset with considerably fewer COVID-19 CXRs than those in the normal and viral pneumonia classes [22], [33], [37]. This may suggest that the models are better suited to differentiating viral pneumonia and normal CXRs but are likely not adequately trained to detect COVID-19 CXRs. For the few 3-way classification studies that used a balanced dataset, each class consisted of only a couple hundred CXRs [45]– [47]. In these studies, the ML models likely suffered from low input data variation which would likely make ML models less effective at making predictions on a wider population. As such, this project has a distinct advantage over previous studies as it considers the need for a balanced dataset (to minimise bias) while also consisting of a moderate number of CXRs (to increase input data variation and stabilise the model). It should be noted that the studies that performed a binary classification are not suitable for comparison to the results in this study. This is due to the reduced complexity involved when performing binary classification as opposed to 3-way classification. This is reflected in the literature review where several binary classification studies reported accuracies and F1-scores over 99%.

This study also implemented more pre-processing techniques than any other study in the literature review. Although the exact effect of these techniques on the results is unknown, some of these techniques, namely ROI extraction, help the ML model to focus on specific features relevant to CXR classification. From the list of studies in the literature review, only Alam *et al.* [34] considered ROI extraction in their methodology. This technique in particular is crucial for CXR classification as it removes unnecessary information which may otherwise have an undesirable influence on the predictions made by the ML model.

5.4 Meeting Project Objectives

All the project objectives listed in Table 1.1 were satisfied. These objectives were carefully planned at the start of the project to ensure that they were attainable, measurable and that they could be completed in a timely manner to support the project goal. In the following subsections, each objective is explored to determine how they contributed to the success of the overall project.

5.4.1 Objective 1

The first objective of the project was to develop a series of processing techniques that can be applied on the CXR images to aid the ML models with classification. There were two main tasks involved in this process which are data pre-processing and data augmentation. With data pre-processing, the main goal is to enhance the quality of the input images. Initially, the pre-processing process only focused on pixel normalisation of CXR images. However, over the duration of the project, additional advanced pre-processing techniques including ROI extraction, CLAHE and denoising filter were employed to improve the image quality, with ROI extraction expected to be most important pre-processing technique addition.

Bilinear interpolation to resize the CXR images to the size of 224 × 224 pixels, allowed the project to meet the required input size for TL models in classification stage. Furthermore, ROI extraction was very crucial to the project's pre-processing phase, as the original input CXR images contained unwanted texts which hindered the ML model's ability to correctly identify features and patterns pertaining to CXR classification. In this project, ROI extraction was successfully implemented to filter out unwanted information on CXRs. As a result, only the prominent lung area information of the CXR images were passed into the ML models for classification for the pre-processed dataset.

Another element of this objective was to increase the training data set size for the purpose of minimising the effect of the overfitting. Using various data augmentation methods including rotation and width/height shifts greatly assisted to successfully achieve this objective. Not only was this objective successfully achieved, but it also improved the classification performance of the ML models using the new pre-processed dataset.

5.4.2 Objective 2

In the second objective, the main task was to develop ML models that could perform accurate CXR classification. The models developed should be capable of accurately identifying CXRs belonging to a class and differentiating them from other classes. From the literature review, CNN architectures were deemed to be the most effective ML technique for image classification. Therefore, to successfully achieve this project objective, the team placed prominence on CNN models for accurate CXR classification. A custom CNN model was developed from scratch to better understand the inner workings of a CNN architecture. In general, these CNN models required an input image of size 224 × 224 as well as an appropriately selected kernel size for each layer.

The use of pre-trained CNN models via TL was shown to be effective for CXR classification in many of the literature studies analysed. The implementation of pre-trained models, VGG-16, Inception-V3 and DenseNet-121, assisted greatly in improving the classification performance which is evident from the results obtained for this project. The pre-trained models via TL exploit knowledge they attained from their previous problem to perform classification on a new problem. By doing so, it allows the pre-trained models to quickly and accurately extract relevant spatial features specific to the new problem. By using pre-trained models, there is also a reduction in the computational time of the training process which otherwise would have been longer with custom CNN models. This characteristic of short duration for training was highly desired as models with a long duration of training time required a greater number of resources and attention.

Upon successfully developing these CNN models, cross-validation was performed to improve the ML model's accuracy and generalisability on unseen data. In other words, overfitting of these ML models was greatly reduced by the use of cross-validation. Initially, the team aimed to perform hold-out cross-validation, which takes up less computational time and is straightforward to implement. However, since only a subset of the entire data was considered for this project, applying hold-out cross-validation would result in greater variance for the training and validation set performances. This means that, the model would show good performance on the training set, but poor on the validation set as the model will struggle to generalise on data that is not exposed to the model during the training process.

In order to reduce overfitting of the ML models and increase generalisability to unseen data, an alternative approach, known as "K-Fold cross-validation", was explored and implemented. Even though this method required a longer computational time, it improved the overall classification performance of the ML model, by reducing biasness and variance. Overall, this objective was successfully achieved, as multiple ML models capable of accurately classifying CXRs of COVID-19, normal and viral pneumonia patients were developed.

5.4.3 Objective 3

The third objective revolved around the evaluation and testing of the ML models that were implemented in the second objective. These models were evaluated using a test set which contained unseen data that was not included in the training and validation datasets. To determine the performance of each ML model, several evaluation metrics were considered including accuracy, precision, recall and F1-score. The values for the performance metrics for each model were produced and are listed in Tables 4.1 and 4.2. According to these tables, the DenseNet-121 model outperformed all the other ML models that were considered in this study, as it obtained the highest F1-scores on both unprocessed and pre-processed datasets. With this final objective fulfilled, the overall aim of this project, which was to explore advanced image classification techniques for classification of COVID-19, normal and viral pneumonia patients, was successfully achieved.

5.5 Limitations

This project was subjected to several limitations that impacted the quality and generalisability of the results. The major limitations of this study are mainly the small dataset size and missing metadata. These limitations and their impact on the project's results are discussed in the following subsections.

5.5.1 Sample Size

The COVID-19 Radiograph database that was used in this project consisted of 3616 COVID-19, 1,345 viral pneumonia and 10,192 normal CXR images. Although a larger dataset often leads to greater generalisation, the effect of biasing may still be prevalent due to a data imbalance between the classes of the dataset. To obtain an even distribution of data, a random sample of 1,345 CXR images were selected from each class. Although this sample size is larger than most of the studies that were considered in the literature review, it is still considered insufficient for the models to properly learn the characteristics of the image. However, the authors believe that the trade-off between sample size and balanced dataset was necessary to reduce the likelihood of the ML model learning one class better than the others.

5.5.2 Missing Metadata

Both datasets were missing key metadata such as COVID-19 strain type (Delta, Omicron etc.), as well as some patient descriptors such as a patient's age, gender, contamination period and medical history. This metadata could help to further generalise the results obtained in this study.

For example, different COVID-19 strains are known to affect the lungs differently. One of these variants (Omicron) causes less damage to the lungs compared to other variants such as Delta [82]. Since the variant type was not included in the CXR metadata, the performance of these ML models could not be generalised for different COVID-19 strains. Another limitation could be the patient's sex, as the male and female immune systems differ in the way that they treat infectious diseases [83]. A patient's age could also influence the interpretation of the results in this study. Government health officials reported that the severity of COVID-19 increases with the patient's age [84]. For example, older patients are more likely to suffer from other respiratory diseases which may also be visible in CXR images. These images can cause the ML models to produce misleading classification performances. Similarly, a patient's medical history can also limit the interpretability of our results. For example, CXR images from patients with a history of lung disease can affect the learning and classification of the ML models. Another important limitation is contamination period of COVID-19. For example, during the incubation period the patients may be asymptomatic and thus there may be no distinguishable COVID-19 features in the CXR image [85]. In this case, the ML model may misclassify a COVID-19 CXR image as normal.

5.5.3 Model Hyperparameter Ambiguity

Newly created custom ML models require some degree of parameter tuning to optimise their classification performance. Certain values such as model learning rate, number of epochs and regulariser intensity can be difficult to determine without sufficient knowledge regarding the fundamental theory that constitutes ML models. As such, the custom CNN model that was designed for this study likely requires further refinement and optimisation. This is a limitation of this particular model, which likely resulted in its poor classification performance when compared to the other pre-trained models.

5.5.4 Fault Diagnosis

With the ML models developed, fault diagnosis and adjusting these faults for improvement proved to be quite challenging. In fact, this is a fundamental limitation associated with all ML models. This is due to the complex algorithm structures involved in ML models which can make locating an error a very tedious task. For example, the DenseNet-121 model that was used in this project has 121 convolutional layers each with different number of parameters, which made it difficult to detect and correct any faults in a timely manner as it would require a thorough investigation of the different layers.

5.5.5 Prediction Reliability

Despite the effort that was undertaken to increase the variability of our dataset, the models that were implemented in this study could still suffer from bias when used on a wider and more general population. This suggests that the verification performed in the testing stage is not completely reliable and that some variation in the results is to be expected when the models are used on different datasets. Unfortunately, due to the statistical nature of ML, the models implemented do not provide context on why a particular decision/classification was made other than pattern recognition (which could vary drastically between datasets).

5.6 Summary

In summary, the Grad-CAM XAI approach has shown that the pre-processed dataset resulted in improved ML model reliability and ‘correctness’. When compared to previous studies, this project was shown to have many advantages. The dataset in this project was considered to be better optimised than those used in previous studies due to its balanced classes and moderate size. This project also implemented ROI extraction, a key pre-processing technique that was omitted in many previous studies. However, this project was not without its own limitations. One of these limitations was the reduced sample size which was necessary to achieve a balanced dataset. Another limitation that was identified was the lack of key metadata. This includes the strain of the COVID-19 virus as well as several other patient descriptors that could provide insight into the patients' health at the time when the CXR images were taken. With these important factors missing, the resulting obscure dataset may in turn lead to subpar and/or inaccurate ML model performance.

Chapter 6: Recommendations for Future Work

The authors recommend future studies to investigate the performance of these CNN models for a larger (but still balanced) CXR dataset. It is hoped that a larger (and possibly more diverse) dataset will better train the CNN models to detect and differentiate COVID-19 from normal CXRs and those with viral pneumonia. It is also recommended that future studies explore CXR datasets with existing metadata pertaining to the COVID-19 strains. Different strains of COVID-19 such as the Delta variant, are known to have a greater impact on the lungs than others, such as the Omicron variant. In this case, if a dataset contains a mixture of both variants, the training of the ML model will be compromised as some COVID-19 characteristic features for one variant may be missing in another. As such, it is best to separate the COVID-19 CXR images for the different variants where possible.

Another important area of future research is the investigation of the pre-processing techniques for CXR images. In this study, several pre-processing techniques were implemented, which when combined, improved the classification performance of the ML models. However, the effect of each pre-processing technique on its own was unknown. In this case, the authors recommend investigating each pre-processing technique on its own and to compare it to the unprocessed dataset to determine which the techniques were most crucial in the pre-processing stage. The hyperparameters of some of these pre-processing techniques such as CLAHE and the median filter could also be explored to determine the most suitable settings for CXR image classification.

In this study, only CNN models were used for image classification. Other ML classification techniques, such as Vision Transformers, could also be investigated and compared to the performance of the CNN models that were considered in this study. One could also explore other CNN architectures such as ResNet, EfficientNet and even some variants of the DenseNet, InceptionNet and VGG architectures that contain a larger number of convolutional layers. In this case, better computational hardware is recommended to reduce the turnaround for the architectures with higher complexity.

To better understand the inner workings of these ML models, the authors highly recommend implementing and exploring XAI approaches. Although these techniques do not affect the results obtained from the ML model, they are instrumental in verifying the performance of the ML models and helping researchers understand how the model classifies data. Some other XAI approaches that could be explored include SHAP (Shapley Additive explanations), DeepSHAP, DEEPLift, CXplain and LIME.

Chapter 7: Conclusion

Overall, the aim of this project was achieved, with several indicators suggesting that ML models could one day be a viable method for CXR classification and COVID-19 detection. Despite the limitations, the ML models implemented were able to achieve satisfactory results on both unprocessed and pre-processed datasets. The pre-trained models, VGG-16, Inception-V3 and DenseNet-121 produced a higher classification accuracy compared to the custom CNN model that was designed for this project. The DenseNet-121 model outperformed all the other models that were implemented, as it produced the highest F1-score on the testing sets for the unprocessed dataset (88%) and the pre-processed dataset (91%). Furthermore, after implementing several pre-processing techniques, the classification performance of each model increased by up to 5%. In this study, ROI extraction is expected to be the most significant pre-processing technique that was implemented as it reduces the likelihood of the ML models undergoing shortcut learning i.e., learning and classifying CXRs based on features that are irrelevant to the problem domain. In terms of future work, the authors highly recommend exploring each pre-processing technique individually to determine which ones were responsible for the observed increase in CXR classification. Furthermore, the authors also recommend future studies to explore other deep learning techniques such as Vision Transformers which have been shown to yield impressive results for the purposes of image classification.

References

[1] WHO, “WHO Coronavirus (COVID-19) Dashboard,” 13 May 2022. [Online]. Available: https://covid19.who.int/. [Accessed 16 May 2022].

[2] P. Zhou, X.-L. Yang, X.-G. Wang, B. Hu, L. Zhang, W. Zhang *et al*., “A pneumonia outbreak associated with a new coronavirus of probable bat origin,” *Nature,* vol. 579, no. 7798, pp. 270-273, 2020.

[3] Z. C. Brooks and S. Das, “COVID-19 Testing: Impact of Prevalence, Sensitivity, and Specificity on Patient Risk and Cost,” *American journal of clinical pathology,* vol. 154, no. 5, pp. 575-584, 2020.

[4] J. Hellewell, T. W. Russell, R. Beale, G. Kelly, C. Houlihan, E. Nastouli *et al*., “Estimating the effectiveness of routine asymptomatic PCR testing at different frequencies for the detection of SARS-CoV-2 infections,” *BMC medicine,* vol. 19, no. 1, p. 106, 2021.

[5] S. Mallett, A. J. Allen, S. Graziadio, S. A. Taylor, N. S. Sakai, K. Green *et al.*, “At what times during infection is SARS-CoV-2 detectable and no longer detectable using RT-PCR-based tests? A systematic review of individual participant data,” *BMC medicine,* vol. 18, no. 1, p. 346, 2020.

[6] D. Jarrom, L. Elston, J. Washington, M. Prettyjohns, K. Cann, S. Myles *et al.*, “Effectiveness of tests to detect the presence of SARS-CoV-2 virus, and antibodies to SARS-CoV-2, to inform COVID-19 diagnosis: a rapid systematic review,” *BMJ evidence-based medicine,* vol. 27, no. 1, pp. 33-45, 2020.

[7] M. Elsharkawy, A. Sharafeldeen, F. Taher, A. Shalaby, A. Soliman, A. Mahmoud *et al*., “Early assessment of lung function in coronavirus patients using invariant markers from chest X-rays images,” *Scientific reports,* vol. 11, no. 1, p. 12095, 2021.

[8] X. Meng and Y. Liu, “Chest Imaging Tests versus RT-PCR Testing for COVID-19 Pneumonia: There Is No Best, Only a Better Fit,” *Radiology,* vol. 297, no. 3, p. 345, 2020.

[9] “Leading Causes of Death,” Centres for Disease Control and Prevention, 13 January 2022. [Online]. Available: https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm. [Accessed 16 May 2022].

[10] WHO, Standardization of Interpretation of Chest Radiographs for the Diagnosis of Pneumonia in Children/World Health Organization Pneumonia Vaccine Trial Investigators' Group, Switzerland: WHO, 2001.

[11] A. Anaya-Isaza, L. Mera-Jiménez and M. Zequera-Diaz, “An overview of deep learning in medical imaging,” *Informatics in medicine unlocked,* vol. 26, p. 100723, 2021.

[12] N. O’Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova *et al.*, “Deep Learning vs. Traditional Computer Vision,” *Advances in Computer Vision,* pp. 128-144, 24 April 2019.

[13] S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image,” *Journal of big data,* vol. 6, no. 1, pp. 1-18, 2019.

[14] M. A. Hedjazi, I. Kourbane and Y. Genc, “On identifying leaves: A comparison of CNN with classical ML methods,” in *2017 25th Signal Processing and Communications Applications Conference (SIU)*, Antalya, Turkey, 2017.

[15] T. Rahman, C. Muhammad and A. Khandakar, "COVID-19 Radiography Database," Kaggle, 1 March 2022. [Online]. Available: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database. [Accessed 30 June 2022].

[16] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub *et al.*, "Can AI Help in Screening Viral and COVID-19 Pneumonia?," *IEEE Access,* vol. 8, pp. 132665-132676, 2020.

[17] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. Abul Kashem *et al.*, "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," *Computers in biology and medicine,* vol. 132, p. 104319, 2021.

[18] D. Reynaldi D.S., B. S. Negara, S. Sanjaya and E. Satria, "COVID-19 Classification for Chest X-Ray Images using Deep Learning and Resnet-101," in *2021 International Congress of Advanced Technology and Engineering*, Taiz, Yemen, 2021.

[19] B. Soni and P. Mathur, "An Improved Image Dehazing Technique using CLAHE and Guided Filter," in *2020 7th International Conference on Signal Processing and Integrated Networks*, Noida, India, 2020.

[20] B. K. Umri, M. W. Akhyari and K. Kusrini, "Detection of Covid-19 in Chest X-ray Image using CLAHE and Convolutional Neural Network," in *2020 2nd International Conference on Cybernetics and Intelligent System*, Manado, Indonesia, 2020.

[21] K. El Asnaoui and Y. Chawki, "Using X-ray iamges and deep learning for automated detection of coronavirus disease," *Journal of biomolecular structure & dynamics,* vol. 39, no. 10, pp. 3615-3626, 2021.

[22] R. Sarki, K. Ahmed, H. Wang, Y. Zhang and K. Wang, "Automated detection of COVID-19 through convolutional neural network using chest x-ray images," *PloS one,* vol. 17, no. 1, p. 262052, 2022.

[23] Y. Kimori, "Mathematical morphology-based approach to the enhancement of morphological features in medical images," *Journal of clinical bioinformatic,* vol. 1, no. 1, p. 33, 2011.

[24] S. Tabik, A. Gomez-Rios, J. L. Martin-Rodriguez, I. Sevillano-Garcia, M. Rey-Area, D. Charte *et al.*, "COVIDGR Dataset and COVID-SDNet Methodology for Predicting COVID-19 Based on Chest X-Ray Images," *IEEE journal of biomedical and health informatics,* vol. 24, no. 12, pp. 3595-3605, 2020.

[25] E. Mineo, "Kaggle," 2018. [Online]. Available: https://www.kaggle.com/code/eduardomineo/u-net-lung-segmentation-montgomery-shenzhen/notebook. [Accessed 16 5 2022].

[26] K. K. Singh and A. Singh, "Diagnosis of COVID-19 from chest X-ray images using wavelets-based depthwise convolution network," *Big Data Mining and Analytics,* vol. 4, no. 2, pp. 84-93, 2021.

[27] Q. Hou and W. Li, "An improved enhancement self-daptive algorithm for X-ray digital image based on wavelet decomposition," *2010 3rd International Congress on Image and Signal Processing,* vol. 3, pp. 1041-1044, 2010.

[28] D. Haritha, M. K. Pranathi and M. Reethika, "COVID Detection from Chest X-rays with DeepLearning: CheXNet," in *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, Patna, India, 2020.

[29] J. Li, D. Zhang, Q. Liu, R. Bu and Q. Wei, “COVID-GATNet: A Deep Learning Framework for Screening of COVID-19 from Chest X-Ray Images,” in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2020.

[30] M. Mishra, V. Parashar and R. Shimpi, “Development and evaluation of an AI System for early detection of Covid-19 pneumonia using X-ray (Student Consortium),” in *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, New Delhi, India, 2020.

[31] D. Hernandez, R. Pereira and P. Georgevia, “COVID-19 detection through X-Ray chest images,” in *2020 International Conference Automatics and Informatics (ICAI)*, Varna, Bulgaria, 2020.

[32] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016.

[33] A. Narin, “Detection of Covid-19 Patients with Convolutional Neural Network Based Features on Multi-class X-ray Chest Images,” in *2020 Medical Technologies Congress*, Antalya, Turkey, 2020.

[34] N.-A. Alam, M. Ahsan, M. A. Based, J. Haider and M. Kowalski, “COVID-19 Detection from Chest X-ray Using Feature Fusion and Deep Learning,” *Sensors,* vol. 21, no. 4, p. 1480, 2021.

[35] F. J. P. Montalbo, “Diagnosing Covid-19 chest x-rays with a lightweight truncated DenseNet with partial layer freezing and feature fusion,” *Biomedical signal processing and control,* vol. 68, p. 102583, 2021.

[36] S. Chaudhary, S. Sadbhawna, V. Jakhetiya, B. N. Subudhi, U. Baid and S. C. Guntuku, “Detecting Covid-19 and Community Acquired Pneumonia Using Chest CT Scan Images With Deep Learning,” in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, ON, Canada, 2021.

[37] S. Albahli, N. Ayub and M. Shiraz, “Coronavirus disease (COVID-19) detection using X-ray images and enhanced DenseNet,” *Applied soft computing,* vol. 110, p. 107645, 2021.

[38] M. K. Bohmrah and H. Kaur, “Classification of Covid-19 patients using efficient fine-tuned deep learning DenseNet model,” *Global Transitions Proceedings,* vol. 2, no. 2, pp. 476-483, 2021.

[39] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang and T. Weyand, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *Computer Vision and Pattern Recognition,* vol. 1, pp. 1-9, 2017.

[40] R. Mohammadi, M. Salehi, H. Ghaffari, A. A. Rohani and R. Reiazi, “Transfer Learning-Based Automatic Detection of Coronavirus Disease 2019 (COVID-19) from Chest X-ray Images,” *Journal of biomedical physics and engineering,* vol. 10, no. 5, pp. 559-568, 2020.

[41] V. S. K. Tangudu, J. Kakarla and I. B. Venkateswarlu, “COVID-19 detection from chest X-ray using MobileNet and residual seperable convolutional block,” *Soft Computing,* vol. 26, no. 5, pp. 2197-2208, 2022.

[42] G. Jia, H.-K. Lam and Y. Xu, “Classification of COVID-19 chest X-Ray and CT images using a type of dynamic CNN modification method,” *Computers in biology and medicine,* vol. 134, p. 104425, 2021.

[43] B. Jabber, J. Lingampalli, C. Z. Basha and A. Krishna, “Detection of Covid-19 Patients using Chest X-ray images with Convolution Neural Network and Mobile Net,” in *2020 3rd International Conference on Intelligent Sustainable Systems*, Thoothukudi, India, 2021.

[44] S. Akter, F. M. J. M. Shamrat, S. Chakraborty, A. Karim and S. Azam, "COVID-19 Detection Using Deep Learning Algorithm on Chest X-ray Images," *Biology,* vol. 10, no. 11, p. 1174, 2021.

[45] M. Shorfuzzaman and M. Masud, "On the Detection of COVID-19 from Chest X-Ray Images Using CNN-Based Transfer Learning," *Computers, materials & continua,* vol. 64, no. 3, pp. 1359-1381, 2020.

[46] F. Y. Santoso and H. D. Purnomo, "A Modified Deep Convolutional Network for COVID-19 detection based on chest X-ray images," in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems*, Yogyakarta, Indonesia, 2020.

[47] M. M. Rahaman, C. Li, Y. Yao, F. Kulwa, M. A. Rahman, Q. Wang *et al.*, "Identification of COVID-19 samples from chest X-Ray images using deep learning: A comparison of transfer learning approaches," *Journal of X-ray science and technology,* vol. 28, no. 5, pp. 821-839, 2020.

[48] J. D. López-Cabrera, R. Orozco-Morales, J. A. Portal-Díaz, O. Lovelle-Enríquez and M. Pérez-Díaz, "Current limitations to identify covid-19 using artificial intelligence with chest x-ray imaging (part ii). The shortcut learning problem," *Health and technology,* vol. 11, no. 6, pp. 1331-1345, 2021.

[49] A. Fernandez, S. C. N. V. del Rio and F. Herrera, "An insight into imbalanced Big Data classification: outcomes and challenges," *Complex & Intelligent Systems,* vol. 3, no. 2, pp. 105-120, 2017.

[50] P. Sane and R. Agrawal, "Pixel normalization from numeric data as input to neural networks: For machine learning and image processing," in *2017 International Conference on Wireless Communications, Signal Processing and Networking*, Chennai, India, 2017.

[51] W. Liu, J. Luo, Y. Yang, W. Wang, J. Deng and L. Yu, "Automatic lung segmentation in chest X-ray images using improved U-Net," *Nature,* vol. 12, no. 1, p. 8649, 2022.

[52] "How U-Net works?," ArcGIS Developers, [Online]. Available: https://developers.arcgis.com/python/guide/how-unet-works/. [Accessed 10 October 2022].

[53] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Lecture Notes in Computer Science,* vol. 9351, pp. 234-241, 2015.

[54] "Image Processing - Bilinear Interpolation," The AI Learner, 29 December 2018. [Online]. Available: https://theailearner.com/2018/12/29/image-processing-bilinear-interpolation/. [Accessed 10 October 2022].

[55] M. Sahani, S. K. Rout, L. M. Satpathy and A. Patra, "Design of an embedded system with modified contrast limited adaptive histogram equalisation technique for real-time image enhancement," in *International Conference on Communications and Signal Processing (ICCSP)*, India, 2015.

[56] Z. Xu, X. Liu and N. Ji, "Fog Removal from Color Images using Contrast Limited Adaptive Histogram Equalization," in *2009 2nd International Congress on Image and Signal Processing*, China, 2009.

[57] S. Muniyappan, A. Allirani and S. Saraswathi, "A novel approach for image enhancement by using contrast limited adaptive histogram equalization method," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, India, 2013.

[58] R. Senaratne, “CLAHE and Thresholding in Python,” Towards Data Science, 4 July 2020. [Online]. Available: https://towardsdatascience.com/clahe-and-thresholding-in-python-3bf690303e40. [Accessed 2 October 2022].

[59] R. Kumudham, Dhanalakshmi, A. G. Swaminathan and V. Rajendran, “Comparison of the performance metrics of median filter and wavelet filter when applied on SONAR images for denoising,” in *2016 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*, India, 2016.

[60] G. George, R. M. Oommen, S. Shelly, S. S. Philipose and A. M. Varghese, “A Survey on Various Median Filtering Techniques For Removal of Impulse Noise From Digital Image,” in *2018 Conference on Emerging Devices and Smart Systems (ICEDSS)*, India, 2018.

[61] L. Tan and J. Jiang, Digital Signal Processing: Fundmentals and Applications, United Kingdom: Academic Press, 2019.

[62] N. Marturi, “Vision and visual servoing for nanomanipulation and nanocharacterization in scanning electron microscope,” Micro and Nanotechnologies/Microelectronics, 2013.

[63] T. Araújo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy *et al.*, “Classification of breast cancer histology using Convolutional Neural Networks,” *PloS one,* vol. 12, no. 6, p. 177544, 2017.

[64] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger and H. Greenspan, “Synthetic data augmentation using GAN for improved liver lesion classification,” in *2018 IEEE 15th International Symposium on Biomedical Imaging*, DC, USA, 2018.

[65] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian *et al.*, “A survey on deep learning in medical image analysis,” *Medical Image Analysis,* vol. 42, pp. 60-88, 2017.

[66] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian *et al.*, “A survey on deep learning in medical image analysis,” *Medical image analysis,* vol. 42, pp. 60-88, 2017.

[67] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE transactions on knowledge and data engineering,* vol. 22, no. 10, pp. 1345-1359, 2010.

[68] J. Deng, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, FL, USA, 2009.

[69] A. Ke, W. Ellsworth, O. Banerjee, A. Y. Ng and P. Rajpurkar, “CheXtransfer: performance and parameter efficiency of ImageNet models for chest X-Ray interpretation,” in *2021 Proceedings of the Conference of Health, Inference, and Learning*, NY, USA, 2021.

[70] A. Thite, “Introduction to VGG16 | What is VGG16?,” Great Learning Team, 1 October 2021. [Online]. Available: https://www.mygreatlearning.com/blog/introduction-to-vgg16/. [Accessed 22 May 2022].

[71] M. Hassan, “VGG16 - Convolutional Network for Classification and Detection,” Neurohive, 20 November 2018. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/. [Accessed 22 May 2022].

[72] X. Xia, C. Xu and B. Nan, “Inception-v3 for flower classification,” in *2017 2nd International Conference on Image, Vision and Computing*, Chengdu, 2017.

[73] A. Arora, “DenseNet Architecture Explained with PyTorch Implementation from TorchVision,” GitHub, 2 August 2020. [Online]. Available: https://amaarora.github.io/2020/08/02/densenets.html. [Accessed 22 May 2022].

[74] A. Ahmed, “Architecture of DenseNet-121,” OpenGenus, [Online]. Available: https://iq.opengenus.org/architecture-of-densenet121/#:~:text=In%20short%2C%20DenseNet%2D121%20has%20120%20Convolutions%20and%204%20AvgPool,use%20features%20extracted%20early%20on.. [Accessed 22 May 2022].

[75] E. T. Hastui, A. Bustamam, P. Anki, R. Amalia and A. Salma, “Performance of True Transfer Learning using CNN DenseNet121 for COVID-19 Detection from Chest X-Ray Images,” in *2021 IEEE International Conference on Health, Instrumentation & Measurement, and Natural Sciences*, Medan, Indonesia, 2021.

[76] R. Agarwal, “The 5 Classification Evaluation metrics every Data Scientist must know,” Towards Data Science, 18 September 2019. [Online]. Available: https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226. [Accessed 1 October 2022].

[77] E. Allibhai, “Hold-out vs. Cross-validation in Machine Learning,” Medium, 3 October 2018. [Online]. Available: https://medium.com/@eijaz/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f. [Accessed 3 October 2022].

[78] P. Tamilarasi and R. U. Rani, “Diagnosis of Crime Rate against Women using k-fold Cross Validation through Machine Learning,” in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, India, 2020.

[79] S. Yadav and S. Sanyam, “Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification,” in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, India, 2016.

[80] M. A. Khan, M. Azhar, K. Ibrar, A. Alqahtani, S. Alsubai *et al.,* “COVID-19 Classification from Chest X-Ray Images: A Framework of Deep Explainable Artificial Intelligence,” *Computational intelligence and neuroscience,* vol. 2022, p. 4254631, 2022.

[81] R. Draelos, “Grad-CAM: Visual Explanations from Deep Networks,” Glass Box Medicine, 29 May 2020. [Online]. Available: https://glassboxmedicine.com/2020/05/29/grad-cam-visual-explanations-from-deep-networks/. [Accessed 15 October 2022].

[82] CDC, “What you need to know about variants,” Centers for Disease Control and Prevention, 26 April 2022. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/variants/about-variants.html#:~:text=The%20Omicron%20variant%20causes%20more,of%20all%20variants%2C%20including%20Omicron. [Accessed 25 May 2022].

[83] B. G. M., “Coronavirus: Why Men are More Vulnerable to Covid-19 Than Women?,” *SN Compr Clin Med,* vol. 2, no. 7, pp. 874-876, 2020.

[84] D. o. Health, “Coronavirus (COVID-19) advice for older people and carers,” Australian Government, 30 May 2022. [Online]. Available: https://www.health.gov.au/node/18602/coronavirus-covid-19-advice-for-older-people-and-carers#:~:text=More%20information-,Older%20people%20are%20at%20risk,or%20a%20weakened%20immune%20system. [Accessed 25 May 2022].

[85] L. Maragakis, “Coronavirus Diagnosis: What should I expect?,” Health, 24 January 2022. [Online]. Available: https://www.hopkinsmedicine.org/health/conditions-and-

diseases/coronavirus/diagnosed-with-covid-19-what-to-expect . [Accessed 25 May 2022].

[86] RadiologyInfo, "Chest X-ray," RadiologyInfo, 15 June 2020. [Online]. Available: https://www.radiologyinfo.org/en/info/chestrad . [Accessed 15 May 2022].

[87] D. Weatherspoon, "Chest X-Ray," Healthline, 18 November 2018. [Online]. Available: https://www.healthline.com/health/chest-x-ray. [Accessed 15 May 2022].

[88] "X-Ray Examinations," Better Health, 6 February 2019. [Online]. Available: https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/x-ray-examinations. [Accessed 15 May 2022].

[89] "X-Ray," Your Practice Online, [Online]. Available: https://www.ypo.education/medical-tests/x-ray-t291/video/. [Accessed 2022 April 30].

[90] "X-ray," NHS, 20 April 2022. [Online]. Available: https://www.nhs.uk/conditions/x-ray/ . [Accessed 15 May 2022].

[91] L. A. Rousan, E. Elobeid, M. Karrar and Y. Khader, "Chest X-ray findings and temporal lung changes in patients with COVID-19 pneumonia," *BMC Pulmonary Medicine,* vol. 20, no. 1, p. 245, 2020.

[92] R. Yasin and W. Gouda, "Chest X-ray findings monitoring COVID-19 disease course and severity," *Egyptian Journal of Radiology and Nuclear Medicine,* vol. 51, no. 1, pp. 1-18, 2020.

[93] A. Cattamanchi, "What is ground glass opacity," Medical News Today, 29 March 2021. [Online]. Available: https://www.medicalnewstoday.com/articles/ground-glass-opacity . [Accessed 15 May 2022].

[94] C. Bao, X. Liu, H. Zhang, Y. Li and J. Liu, "Coronavirus Disease 2019 (COVID-19) CT Findings: A Systematic Review and Meta-analysis," *Journal of the American College of Radiology,* vol. 17, no. 6, pp. 701-709, 2020.

[95] "Machine Learning," IBM Cloud Education, 15 July 2020. [Online]. Available: https://www.ibm.com/cloud/learn/machine-learning . [Accessed 16 May 2022].

[96] S. Brown, "Machine Learning, Explained," MIT Management, 21 April 2021. [Online]. Available: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained. [Accessed 16 May 2022].

[97] "What is Machine Learning (ML)?," Berkeley, 26 June 2020. [Online]. Available: https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/. [Accessed 16 May 2022].

[98] A. Bajaj, "Performance Metrics in Machine Learning," Neptune Blog, 18 March 2022. [Online]. Available: https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide . [Accessed 16 May 2022].

[99] "Convolutional Neural Network (CNN) in Machine Learning," GeeksforGeeks, 28 Decemeber 2020. [Online]. Available: https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/ . [Accessed 16 May 2022].

[100] S. Saha, "A comprehensive guide to convolutional neural networks - the ELI5 way," Towards Data Science, 16 December 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 . [Accessed 16 May 2022].

[101] J. Brownlee, “How do Convolutional Layers Work in Deep Learning Neural Networks?,” Machine Learning Mastery, 17 April 2020. [Online]. Available: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/ . [Accessed 16 May 2022].

[102] “What is pooling in a convolutional neural network (CNN): Pooling layers explained,” Programmathically, 5 December 2021. [Online]. Available: https://programmathically.com/what-is-pooling-in-a-convolutional-neural-network-cnn-pooling-layers-explained/ . [Accessed 16 May 2022].

[103] S. Saxena, “Introduction to Softmax for Neural Network,” Analytics Data, 5 April 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/ . [Accessed 16 May 2022].

[104] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong and M. Ghassemi, “COVID-19 Image Data Collection: Prospective Predictions are the Future,” *Journal of Machine Learning for Biomedical Imaging,* vol. 2, pp. 1-38, 2020.

[105] A. Vabalas, E. Gowen, E. Poliakoff, A. J. Casson and E. Hernandez-Lemus, “Machine learning algorithm validation with a limited sample size,” *PloS One,* vol. 14, no. 11, p. 224365, 2019.

Appendices

This section of the report contains some supplementary background information related to the project which has been included for the reader's benefit.

Appendix A: Chest X-ray Radiography

CXR is one the imaging techniques that is used to look inside the body, specifically the chest area as the ROI. Hence, this technique is useful for analysing the heart, lungs, and chest area for medical diagnosis and treatment purposes [86]. This imaging techniques uses small amount of ionising radiation to produce these images [87]. In small dosages, this radiation is considered to be non-lethal, and is thus not likely to cause any serious health complications for the patient [88].

A CXR image is typically undertaken in a special room purposefully equipped with a portable X-ray camera supported by a metallic frame. Prior to proceeding with the examination, the patient will be asked by the technician to remain in front of X-ray film so when the radiation beam passes through the body, it can be captured and recorded [88], as shown in Figure A.1 [89].

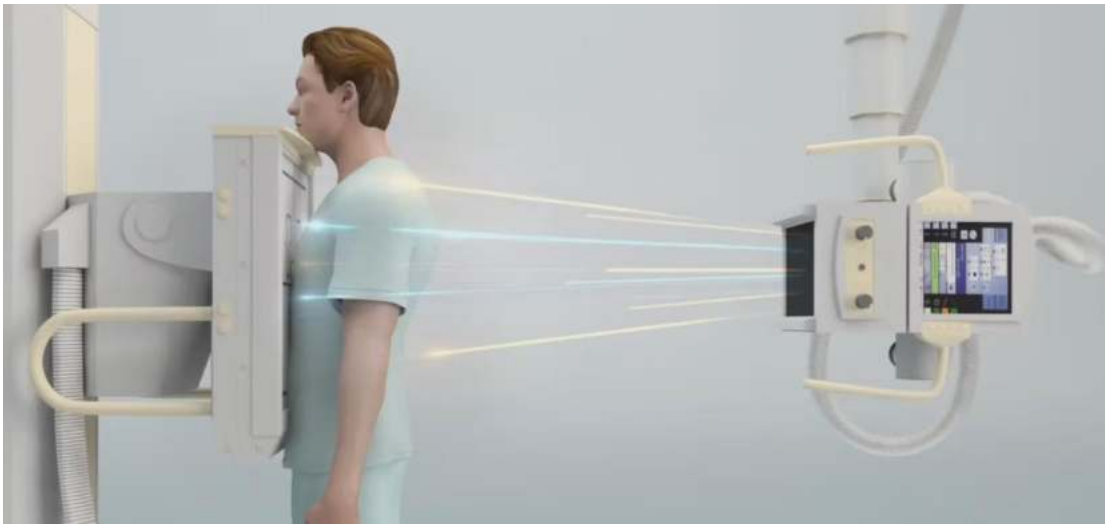


Figure A.1: Depiction of patient undergoing CXR imaging [89].

If the patient does not remain still during this procedure, the film will appear to be blurry and the resulting CXR will not be sufficient for the radiologists to correctly report on the patient's condition [87].

A CXR is a 2D image where body parts in the chest area appear in different colours, depending on the intensity of the radiation passing through them [86]. For instance, more radiation will pass through soft body tissue compared to dense structures such as bones. Thus, the X-ray will show the dense structures as a bright white colour, soft tissues as a grey colour and any air inside the lungs as a dark colour [90]. After the X-ray examination is completed, the patient may go home, and the X-ray image would be sent to a radiologist so a technical analysis report would be prepared for the patient's general practician [90].

CXR imaging has several advantages when compared to other medical imaging techniques. The use of X-ray imaging is prevalent inside emergency rooms as it is significantly faster and more convenient to use [86]. More medical centres and clinics intend to equip their place with X-ray machines as it is considered to be a cost-effective machine for medical diagnosis [86].

For patients with COVID-19 symptoms, their CXR images can show signs of abnormalities with the prominent ones being the presence of Ground Glass Opacity (GGO) in the lower region of the lungs called, "Lower Lobes" [91] and consolidation of the lungs [92].

GGO refers to the grey area that can be present on CXR images and in the case of lung organs, it can be described as a sign of an unusual condition for the lungs since healthy lungs should appear as a black colour on the CXR images [93]. GGO can be caused due to many reasons that can compromise the normal function of the lungs [93]. One of the main causes of GGO is due to an increased density in the lungs which can be a result of fluid build-up, inflammation, or tissue damage in the lung region.

However, the scale of the size of GGO depends on various important factors such as the severity of disease and at what period the image is taken [20], [94]. In the case of a common COVID-19 disease, GGO may start to develop from the first few days when the symptoms are shown and reaches the peak level from day 5 to day 10 since the first symptoms [91], [94]. In the case of a severe COVID-19 patient, GGO would compromise the whole lobe and lungs in general.

As discussed, one of the other common features in the CXR images of patients with COVID-19, is the consolidation of the lungs. On a CXR, this can be seen as a white area with similar features to GGO. However, in this case, the blood vessels are concealed in the X-ray image [91].

Appendix B: Machine Learning Techniques

ML is a field of computer and data science which uses advanced algorithms and a large dataset, to develop and train a model, so that a machine can learn, think, and understand like a human without the need of any pre-programming [95].

ML runs in different arrangements called supervised, unsupervised, semi-supervised and reinforcement learning. For this project, the focus is placed on supervised ML methods, where the ML model is trained on a dataset and learns the patterns associated with the data [96]. To achieve this, the dataset should be separated into two sets: training set (labelled) and testing set (unlabelled). From the training set, the model would be able to learn the pattern present in the data [96]. The model can then be evaluated using the testing dataset to measure the model's effectiveness in learning and classifying images [97]. Several evaluation metrics, such as precision, recall, accuracy, and F1-score, can be used to measure the model's performance [98].

In this project, emphasis is placed on deep learning image classification techniques, such as CNNs. CNN is a unique technique where convolutional layers are used to classify data such as image portrays, as seen in Figure B.1. The first convolutional layer is responsible for extracting the high-level features associated with the images, such as major shape structures. A pooling layer is then used to further reduce the size of the image [99]. The purpose of this reduction in size is to reduce computation power to process the data. These two layers together subsequently make up a single layer of a CNN. There can typically be multiple copies of this single layer in the case where image data has significantly more complex patterns. However, as more layers are added, there is an increased demand on the computational power required to process the image [100].

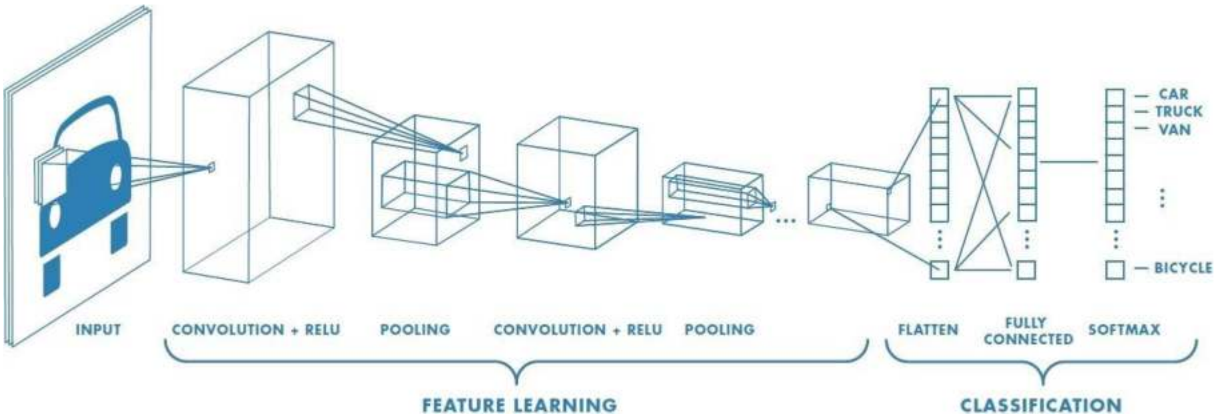


Figure B.1: Diagram depicting the layers of a CNN model [100].

The learning process for a CNN model starts with a convolutional layer that can be described as the main feature of the CNN modelling technique. This simply runs a linear operation of multiplication using the input function (I) and a set of weight function called kernel or filter (f) to produce a function map or feature map $F(t)$ from the operation [22] [101].

$$F(t) = (I * f)(t) \qquad \textit{Eq. (B.1)}$$

Given the input value is one dimensional, and t only takes inputs that are integers, the following equation can be constructed [22]:

$$F(t) = \sum_a I(a) \cdot f(t-a) \qquad \textit{Eq. (B.2)}$$

If the given input value is two-dimensional, the variables would be $I(m, n)$ and $f(a, b)$ and the equation can be written as the following [22]:

$$F(t) = \sum_a \sum_b I(a, b) \cdot f(m-a, n-b) \qquad \textit{Eq. (B.3)}$$

This equation can be re-expressed by switching the filter using the commutative law [22]:

$$F(t) = \sum_a \sum_b I(m-a, n-b) \cdot f(a, b) \qquad \textit{Eq. (B.4)}$$

However, in neural networks, the cross-correlation function is applied instead, which has the same form as the convolution equation above but without flipping the filter variable [22].

$$F(t) = \sum_a \sum_b I(m+a, n+b) \cdot f(a, b) \qquad \textit{Eq. (B.5)}$$

Once the feature map is produced, each feature is passed through the next layer called ReLU which is defined as an activation function. This function aims to convert the negative input values to zero so that the training of the ML model on the input data can be more efficient and faster [22], [101].

The final layer in this technique is called max pooling, whose function is to extract the maximum pixel value or element in the feature map input from the previous layer [102]. Once the learning process is finished, classification of the input data is initiated by having a connected layer which takes the flattened output from the max-pooling layer to calculate the probability values for classification purposes [22], [100].

The classification stage also consists of an activation function called softmax. This function is typically used for classification problems that deal with multiple classes (i.e., 2 or more classes). This function takes the input from the output of the previous layer to perform the final probability calculation [22], [100]. The equation for this function can be expressed as [103]:

$$\text{softmax function } (Z_i) = \frac{e\ (Z_i)}{\sum_j e\ (Z_j)} \qquad \textit{Eq. (B.6)}$$

In Eq. B.6, Z represents the neuron values from the output of the previous layer which passes through an exponential function and is normalised by dividing it by the sum of the neuron values (Z).