Deep Learning Techniques for Breast Cancer Detection in Mammograms

Rhona McCracken

University of St Andrews

Supervised by Dr David Harris-Birtill and Craig Myles

MSci Computer Science Dissertation, University of St Andrews

January 2023

Abstract

Breast Cancer is the most common cancer for females in the UK with approximately 55,000 new cases reported a year in 2016-18 [1]. This project investigates the use of Convolutional Neural Networks and associated Deep Learning Techniques on the classification of benign and malignant lesions in mammogram images for detecting breast cancer. Computer-aided techniques are needed to reduce the repetitive, error-prone workload for radiologists and aid in the detection of cancer to give patients critical treatment as early as possible. The investigations carried out in this project build on the future work suggested by Adam Jaamour in their MSci dissertation on the same topic. The results of the investigations attempt to recreate Jaamour's models and compare the new findings with their work. This project presents a pre-processing pipeline for a combined dataset of CBIS-DDSM [2] and CMMD [3] mammograms informed by a literature review of related work. These two datasets have been combined in a hybrid set since they offer images in two different scanning techniques (film-screen and digital mammography) and patients from two distinct geographical regions (USA and China). This work also investigates the impact of using different network architectures in transfer learning for mammogram classification and hyperparameter tuning techniques. This research selected a best-performing model from the validation set evaluation: a hyperparameter-tuned network using the InceptionV3 base model and combined dataset without augmentation or data preprocessing. In the final results section, this model achieved an accuracy of 63.1% but was outperformed by the original baseline model with a test accuracy of 66.2% - so no improvement from the baseline was demonstrated. However, there were several key findings from the investigation - for example, a larger image size of 500 × 500 pixels was seen to consistently achieve higher accuracy in models than using a lower image resolution of 160 × 160 pixels.

Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated. The main text of this project report is 16628 words long, including project specification and plan. In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

Acknowledgements

I would like to thank both of my supervisors: David Harris-Birtill and Craig Myles for all their encouraging feedback and support throughout the project. David's recommendations for robust research in this domain and Craig's technical help with understanding previous code have been invaluable. Coming towards the end of five years of study in the School of Computer Science at St Andrews I want to show my gratitude for the incredible teaching, supportive staff and welcoming community in this department. Finally, I want to thank all my family and friends for their support as I complete my final year of study.

Ethics

This project aims to investigate the use of different techniques in machine learning to identify breast cancer tumours in images of mammogram scans. The project will use secondary data from The Cancer Imaging Archive: CBIS-DDSM [2] and CMMD [3]. No data will be collected, and no surveys or interviews will be required.

The main ethical issue here is the use of large secondary datasets of patients' medical scans. This data has been anonymised and no attempt will be made to identify any individual patients to maintain confidentiality. The data will be stored on the encrypted Computer Science School servers and all data processing will be on a School GPU. The data is available for public access on The Cancer Imaging Archive. The data will be used to train Convolutional Neural Networks and evaluate the accuracy of these models in identifying breast cancer from the labelled mammography images. The methods used to work with this data will include data cleaning, preprocessing including traditional edge detection and image segmentation techniques and machine learning models.

This work has received full ethical approval from the University of St Andrews, please see the written approval in Appendix B.

Glossary

ANN	Artificial Neural Network
Baseline Model	Our selected initial model used as a control in investigations. Here selected as a MobileNetV2 Transfer Learning model with fine-tuning, 2 fully-connected layers, 1 dropout layer and a Sigmoid classification. (Based on Jaamour's best-performing model [4])
Base Model	Term commonly used in literature to refer to the pre-trained model used for transfer learning e.g. MobileNet, ResNet, VGG-19 etc
Biopsy	Sample of tissue taken from the body for medical diagnosis
CADe	Computer-Aided Detection system
CADx	Computer-Aided Diagnosis system
CBIS-DDSM	Curated Breast Imaging Subset of Digital Database for Screening Mammography (also referred to here as DDSM for brevity)
CLAHE	Contrast Limited Adaptive Histogram Equalisation
CMMD	Chinese Mammography Database
CNN	Convolutional Neural Network - deep learning model with convolutions applied to an image at each layer
ImageNet	[5] large public database of images for training neural networks for computer vision
K-Means Clustering	Unsupervised learning technique to group data into K clusters using a distance metric
Mammogram	Low-energy X-ray commonly used for breast cancer screening
Naive Bayes	Probabilistic classifier applying Bayes Theorem
Random Forest	Randomly initialised decision trees form a collection of classifiers for ensemble learning
ROI	Region of Interest in image analysis
SVM	Support Vector Machines map new data points to points in space so as to maximise the gap in decision boundary between two classes
Transfer learning	Networks pre-trained on general image sets such as ImageNet are fine-tuned for a specific classification problem with specialised data

Contents

Chapter 1

Introduction

1.1 Breast Cancer and Mammography

According to Cancer Research UK; Breast Cancer is the most common cancer for females in the UK with approximately 55,000 new cases reported a year in 2016-18 [1]. Of all reported cases, 23% are thought to be preventable [1]. A common technique for early detection of breast cancer is screening with a low-energy X-ray called a mammogram. Breast screenings are offered to women with a higher risk of breast cancer and the mammograms produced are interpreted by expert radiologists to make a diagnosis. Unfortunately, the chance of human error is high for this repetitive task with 10-30% of cancer cases missed during detection [6]. There is also a risk of false positives which can lead to distrust and anxiety in patients who undergo invasive procedures like biopsies or even unnecessary treatment [7]. With machine learning techniques, Computer-Aided Detection (CADe) or Computer-Aided Diagnosis (CADx) systems could assist radiologists in diagnosis by identifying patterns of lesions in mammogram images or offering full diagnoses based on training knowledge from large datasets.

1.2 Previous Work

This paper presents research based on the future work expansions from a previous MSc project by Adam Jaamour [4]. Jaamour built a baseline Convolution Neural Network to detect breast cancer using two datasets separately: mini-MIAS and CBIS-DDSM and applied a bag-of-tricks approach (his code is open and accessible on Github [8]). The best-performing model from this research had a test accuracy of 67.08% using the DDSM dataset and a transfer learning architecture with MobileNetV2 base model (pre-trained on imagenet).

This research aimed to recreate Jaamour's best-performing model and use it as a baseline to investigate their future work suggestions and additional techniques selected based on a context survey. This included creating a pre-processing pipeline, using data augmentation, changing image size, altering the network architecture and using hyperparameter tuning to find the optimal parameter set. In addition, the investigations used a combined dataset with a proportional number of images from the CBIS-DDSM [2] and CMMD [3] datasets for training, validation and testing.

1.3 Objectives

The initial project aims are set out below.

1.3.1 Primary

1. Literature Review - review existing work on machine learning for cancer detection in medical scans and the proposed techniques for building a new model
2. Basic Convolutional Neural Network model informed by Literature Review - use open source models to create a basic CNN model to form the basis of further research with different techniques
3. Investigate different model architectures informed by Literature Review by altering the basic CNN and report on the differences in results
4. Investigate a selection of pre-processing techniques identified in the Literature Review and report on their impact on results from the basic CNN model

1.3.2 Secondary

1. Extend the basic CNN model using hyperparameter-tuning techniques e.g. Grid Search
2. Report on results from hyperparameter-tuned CNN with selected pre-processing (based on best results from the initial investigation)

1.4 Report Structure

The structure of this paper is as follows: Chapter 2 is a review of related literature, Chapter 3 describes the technology, datasets and methods used and Chapter 4 outlines the investigations carried out with results from validation set evaluation. Final results, discussion, evaluation and conclusions can be found in Chapters 5-7.

Chapter 2

Literature Review

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been chosen for our task since they can learn features directly from images and have outperformed other machine learning techniques in several medical image classification tasks [9].

Convolutional Neural Networks are Deep Learning Networks where convolutions are applied to an input image at each layer of the network to learn features. CNNs are well-designed for image classification tasks since they incorporate the complexity of Deep Learning models and learn image features through layers of applied filters. CNNs have been successful in other medical image analysis tasks, in particular due to their ability to identify features from datasets without the need for manual segmentation from experts [9]. The main limitation of CNNs in medical image analysis is the lack of large labelled datasets - however this problem can be overcome through transfer learning and data augmentation [9] (see Section 2.3).

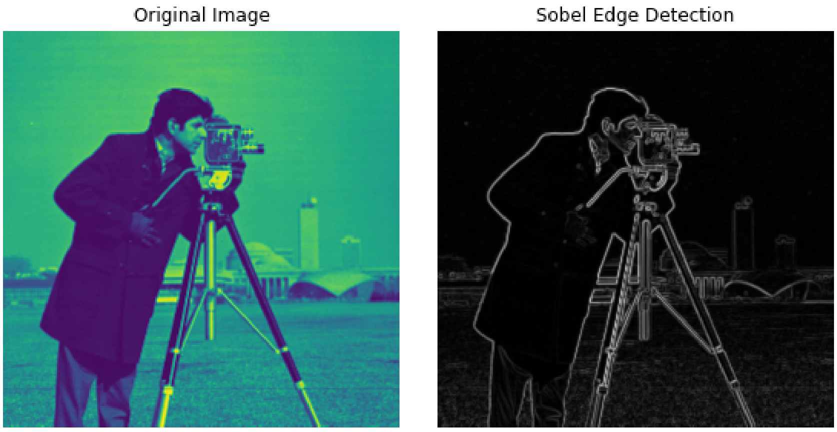
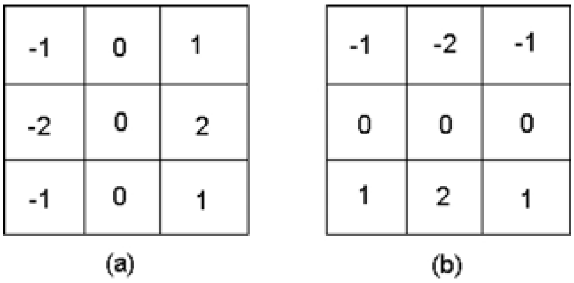


Figure 2.1: Example of Sobel filter applied to a standard image. Figure from [10]

At each convolutional layer, CNNs use kernels (also known as filters) to transform an image, for example a sobel filter could be applied for edge detection (see Figures 2.2 and 2.1). In CNNs the convolutions applied use kernels learnt by the network (the weights between layers in a traditional ANN) to learn features. As the network



-1	0	1
-2	0	2
-1	0	1

(a)

-1	-2	-1
0	0	0
1	2	1

(b)

Figure 2.2: Sobel kernel for edge detection. a) Vertical edges b) Horizontal edges. Figure from [11]

deepens, the high-level features (like edges from a simple sobel filter) are put through more convolutions to produce more detailed features. Every time the filters are applied the output image shrinks and other downsampling such as a max-pooling layer is commonly applied to save memory and reduce training time. Each convolutional layer also applies a Rectified Linear Unit (ReLU) function 2.1 to the output feature map to make negative values zero and add non-linearity [12]:

$$rect(x) = max(0, x) \tag{2.1}$$

At the end of the network the images have been reduced significantly to numeric information representing the combination of features (without spatial information). The final layer of the output is a fully connected layer that uses a Sigmoid or Softmax activation function to produce a probability value used to classify the input image based on these features [13].

2.1.1 Network Architectures

From Table 2.1, some related work uses a combination of traditional machine learning classifiers such as Random Forest, Support Vector Machines, K-means clustering and Naive Bayes [14]. CNNs are often utilised for automatic feature extraction and sometimes for classification [15] [16] [12] [14]. Often in mammography classification, transfer learning is used to fine-tune a pretrained CNN for the specific problem domain: such as AlexNet or VGG16 models pretrained on ImageNet([5]) [15] [16] [17] [14].

Complex pipelines can be observed in the literature, often to overcome the problem of a lack of a large labelled dataset in the domain suitable for deep learning. Bai et al. describe a “Feature Fusion Siamese Network” [15]. Siamese networks can learn from a small sample set of data, making them ideal for medical imaging tasks where data is difficult to label in large quantities and is not always available publicly [15]. Siamese networks contain two identical subnetworks with the same architecture that learn the same weights [15] [18]. In [15] two images are fed as input to the network: one from the current year and one from the previous year for the same patient. The siamese network uses a distance learning network to evaluate the level of similarity between these two images by learning the differences between features. These differences are used to identify whether the patient has developed cancerous masses or microcalcifications since their last scan. Miller et al. use a strong augmentation-based self-supervised learning (SSL) model [19]. Similar to the siamese network, this model tries to compensate for a lack of large labelled datasets by using a combination of labelled and unlabelled data. SSL models try to learn the labels for unknown data through prediction: based on similarities with the labelled data. Different SSL methods were used in [19] but for each method, the whole mammogram image was first split into patches for pre-training an encoder. This was due to mammograms being very large and features of interest (e.g. lesions) being localised to small areas in the images. The pre-trained encoder was then applied to the whole image for classification. This SSL model with finetuning on DDSM and CMMD (without patch labels) outperformed the supervised model [19], showing that this type of learning has been used successfully within the domain.

In [12] a deep learning algorithm known as 'You Only Look Once' (YOLO) was used to create a pipeline for mammogram feature extraction and lesion classification. YOLO algorithms are often used in computer vision since they identify objects in an image in real-time [20]. YOLO algorithms split an input image into a grid and use regression problems to identify the bounding boxes of objects in the image. These algorithms create a classification probability score for each grid cell and these are combined to identify the class of each object captured by a bounding box [12]. Since YOLO completes this full object detection process in a single pass of the CNN it is very fast and uses context of the whole image to make accurate object-level classifications.

A Stacked Denoising Autoencoder can be used to pre-train a network for feature extraction using unsupervised learning. SDAs move data around creating noise and attempt to select the important features from this noise to perform classification. The word 'stacked' comes from training each layer in turn and then freezing that layer to train the next layer - stacking the denoising autoencoder layers together [21]. After pretraining, networks are often fine-tuned with supervised learning. Resampling with a Markov Chain allows the SDA to try different combinations of the data until the best performing model is found [22].

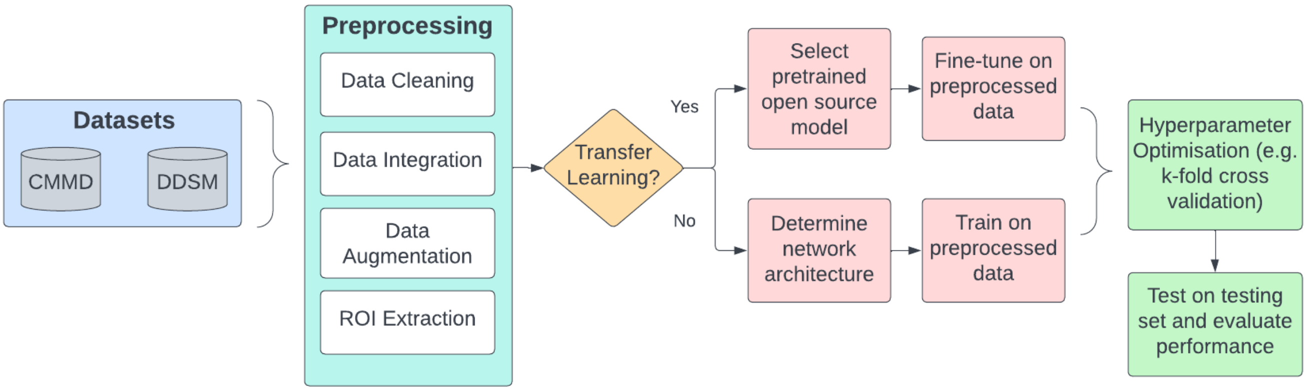

Datasets
CMMD
DDSM
Preprocessing
Data Cleaning
Data Integration
Data Augmentation
ROI Extraction
Transfer Learning?
Yes
No
Select pretrained open source model
Fine-tune on preprocessed data
Determine network architecture
Train on preprocessed data
Hyperparameter Optimisation (e.g. k-fold cross validation)
Test on testing set and evaluate performance

Figure 2.3: Basic machine learning pipeline of stages required in machine learning for cancer detection, including preprocessing steps

2.2 Datasets

2.2.1 CMMD

The Chinese Mammography Database (CMMD) contains 2212 scans (excluding a part of the dataset - explained in Chapter 3) from 1775 patients in China with benign or malignant tumors confirmed by biopsy [3]. The dataset consists of images from full-field digital mammography - the technology that has replaced analogue film-screen mammography globally [23]. Mediolateral Oblique (MLO) and Craniocaudal (CC) projections are provided for each patient [24] so it is important to split train, test and validation sets by patient rather than by scan to avoid data leakage.

2.2.2 CBIS-DDSM

The Curated Breast Imaging Subset of the Digital Database for Screening Mammography (CBIS-DDSM) contains scanned film-screen mammography images. These scans have been chosen from the original DDSM dataset where lesions could be clearly identified and labelled by a radiographer. The images were converted to the modern medical image format DICOM and modified to include segmentation, bounding boxes and diagnosis. The dataset comprises a total of 3047 images from 1566 patients, split into train and test sets.

2.2.3 Combined Datasets

Table 2.1 provides a comparison of studies using both CMMD and CBIS-DDSM datasets together and studies using these datasets individually. There are far fewer studies using CMMD than DDSM since the CMMD dataset is much more recent. Only 4 papers were found where CMMD and DDSM were used in conjunction. This is most likely due to the recent publication of the CMMD dataset and the difficulty processing images from each dataset to standardise the input sizes and features. For example the DDSM images require more preprocessing to remove artifacts such as handwritten radiographer labels from analogue scans [25] but CMMD images are taken from digital scans without any of these artifacts (confirmed through extensive sampling of CMMD images). See Figure 2.4 for comparison of DDSM and CMMD scans.

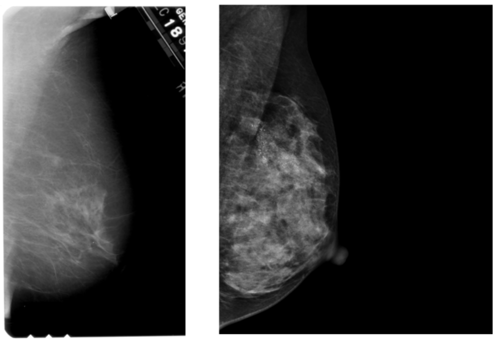


Figure 2.4: Left: CBIS-DDSM scan with visible radiographer artefacts, Right: CMMD digital scan

Paper	DDSM	CMMD	Architecture	Data Cleaning	Data Integration	Data Augmentation	ROI Extraction	Evaluation
[15]	✓	✓	Siamese CNN (pairs of previous and current year scans from private UCHC dataset). Transfer learning with VGG, ResNet pretrained with ImageNet	✓ min-max normalisation, background and labels removed, CLAHE applied	✓ images resized	✓ rotations of 90, 180 and 270°	✓ cropping to largest region of interest	0.92 accuracy, AUC 0.95, specificity 0.91, precision of 0.91, sensitivity 0.939
[19]	✓	✓	Strong augmentation-based self-supervised learning (SSL). Patch set grids created with 50% overlap	✗	✓ patches with 20% background discarded	✓ random crop, brightness, contrast and gamma shift, gaussian blur, histogram equalisation, sharpening, high-pass filter	✗	
[16]	✓	✓	Transfer Learning with pretrained AlexNet, VGG16, VGG19 trialed with and without frozen layers	✓ remove noise and background with binary filter. Wiener filter and CLAHE to enhance images.	✗	✓ random transformations	✓ ROI extracted from background with binary filter	Allowing layers to train gave better performance. VGG16 gave 100% test accuracy, sensitivity and specificity. AlexNet had 99.5%, 99% and 100% respectively

Paper	DDSM	CMMD	Architecture	Data Cleaning	Data Integration	Data Augmentation	ROI Extraction	Evaluation
[17]	✓(train and test)	✓(test only)	“meta-repository” of models tested on 7 datasets, trained on DDSM and fine-tuned on INbreast	✗	✗	✗	✗	Too many to list, DDSM performed better than CMMD in testing
[12]	✓	✗	Novel Regional CNN: You Only Look Once (YOLO), 24 conv layers, max pooling and 2 fully connected layers. Splits input image into $n \times n$ overlapping cells to classify entire image at once	✓ Otsu thersholding, 2D Gaussian low pass filter and Normalised Thickness Profile to remove background. Images resized for uniformity.	✗	✓ rotation of 90, 180 and 270 °	✗	Detection accuracy 99.70%, classification accuracy 97.00%. Sensitivity for benign: 100%, specificity for malignant: 94%. Detected masses in pectoral muscle and dense tissue (known challenge)
[24]	✗	✓	Stacked Denoising Autoencoder (SAE) compared with SVM benchmark, 10-fold hyperparameter tuning in 3 scenarios (microcalcifications alone, masses alone and masses and microcalcifications together)	✗	✗	✓ rotation, scaling and transformation	✓ computerised segmentation approach based on clustering from [26]	Micro-calcifications alone: 0.873 accuracy for SAE and 0.858 for SVM. Masses alone: 0.613 for both SAE and SVM. Combined microcalcifications and masses accuracies: 0.897 for SAE and 0.858 for SVM

Paper	DDSM	CMMD	Architecture	Data Cleaning	Data Integration	Data Augmentation	ROI Extraction	Evaluation
[14]	✗	✓	Regions of Interest Calcifications (ROIC) extracted with image analysis. Handracfted and CNN-learnt features used independently and in combination. CNN for learning features (same convolutional layers as AlexNet) pre-trained on ImageNet and fine-tuned on CMMD. Different classifiers compared.	✗	✓ to combine manual and CNN features Canonical Correlation Analysis (CCA) was used	✓ rotations of 0, 45, 90 and 135° to ROICs	✓ traditional image processing for ROIC segmentation - morphological top-hat filtering with spherical structure, binarised with otsu thresholding and dilated with disk structure. Max-connected region selected as ROIC	CNN features alone outperformed manual features alone with accuracies of 0.8768 and 0.8667 respectively. Combined manual and CNN features gave best performance overall with filtered features showing greatest performance improvement. Highest accuracy achieved by CNN features filtered by morphological features (0.8859)

Table 2.1: Comparison of related work on data pre-processing techniques using CMMD and DDSM datasets

2.3 Preprocessing Techniques

Preprocessing on image datasets can refer to the data cleaning and resizing done to allow two datasets to be combined successfully or to highlight important features more clearly. Preprocessing also includes data augmentation for artificially increasing the size of the dataset and identification of regions of interest to crop images and speed up training time.

2.3.1 Data Cleaning

Data cleaning aims to improve the quality of images by reducing noise, filling in missing values and removing outliers [27]. One common example of image preprocessing was referred to as ‘centering' in a 2012 paper on AlexNet; where the mean pixel of an image is subtracted from each pixel in each channel as a form of normalisation [28]. An alternative normalisation scales pixel values into a specific range, for example, the range [-1,1] for GoogLeNet Inception [29].

Preprocessing in mammography imaging often includes noise reduction and background removal. [15] applied min-max normalisation and removed the majority of the black background by calculating the maximum width of the breast area across all images and cropping the images to this maximum width plus some small margin. In [16], the background noise was reduced by applying a binary filter and then a Wiener filter. [12] applied a complex process to remove the background noise - techniques can be seen in Table 2.1, row 5.

In Table 2.1, the papers using CMMD alone did not apply any data cleaning techniques suggesting that these digital scans may contain fewer artifacts and background noise than the CBIS-DDSM.

Another preprocessing technique is contrast enhancement, assumed to be particularly useful in DDSM scans where there is lower contrast between the breast tissue and the background compared to the CMMD scans. Lbachir et al [25] used the Contrast Limited Adaptive Histogram Equalisation (CLAHE) technique, for contrast enhancement, on DDSM images (see Figure 2.5). This technique was chosen after achieving the best Peak Signal-To-Noise Ratio (PSNR) in a comparative study, by the same authors, of five contrast enhancement methods for mammogram images. CLAHE achieved a PSNR of 76.04 dB and was found to have an advantage over the other methods since it can avoid over-enhancement from the user selecting a maximum “clip limit value”. This technique has also been used in Bokade and Shah's model outlined in [31].

Another important data cleaning task for the DDSM dataset is the removal of artefacts in the scans that should not be included in the classification, for example radiographers' labels. Lbachir et al. describe a method of removing these labels from the images with traditional image processing techniques including thresholding to binarise the image and exploiting morphological properties to select only the breast area (largest solid area in the scans) [25]. See Figure 2.6. This technique could also be used to remove the background completely with additional edge detection filtering.

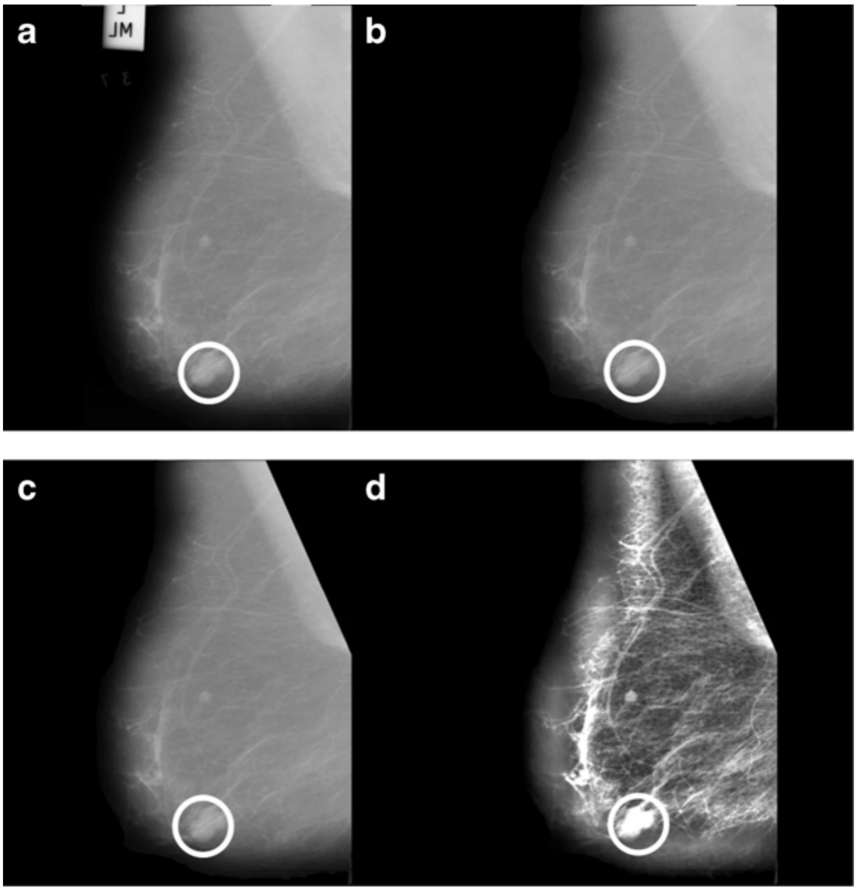


Figure 2.5: Junior et al Preprocessing: (a) Original image, (b) Image after background removal (removed annotation tag), (c) Image after pectoral muscle removal and (d) Image after contrast enhance with CLAHE and contrast stretching [30]

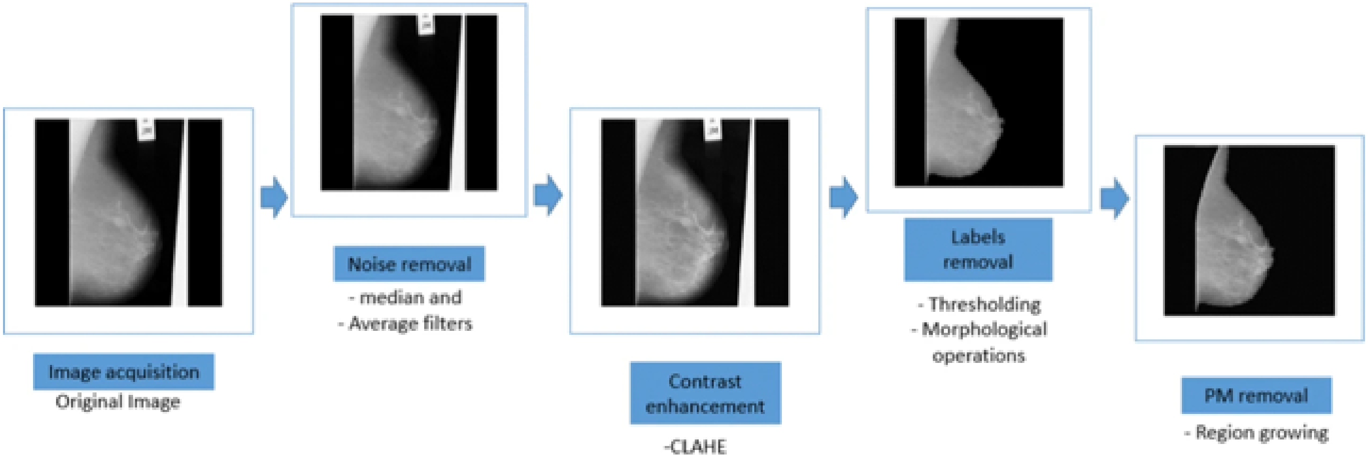


Figure 2.6: Figure from Lbachir et al Preprocessing Flow Chart [25]

2.3.2 Data Integration

The images are of different quality and size in DDSM and CMMD datasets. In some studies that combine both datasets in training, additional preprocessing is carried out to make the images more similar in size and visible features [15]. In [19] patch set grids were used for training an SSL model and these were taken with 50% overlap from both image sets. In [25] the pectoral muscle was removed from DDSM images to allow the model to focus on learning lesions in the breast tissue. This technique could also be employed for data integration to make the feature set the same across both datasets since the CMMD images do not always include the pectoral muscle (see Figure 2.4 for a comparison of a DDSM and CMMD image).

2.3.3 Data Augmentation

After preparing the data and splitting it into train and test sets, Data Augmentation can be used to increase the size of the dataset. This is very useful since deep learning performs best when trained on very large datasets and it can be difficult to create large enough medical datasets with ground truth images labelled by radiologists. Data Augmentation involves applying different transformations to the images in a dataset to create additional images that are flipped or colour-altered and difficult to match to the original images. In order to prevent data leakage - where the model may gain an advantage if it sees the flipped pair of a training image in the test set - Data Augmentation must take place after splitting the dataset into train and test sets.

Commonly in CNN preprocessing the images are flipped horizontally [28] [32] but any combination of flips and rotations could be used. Another Data Augmentation technique is colour or brightness alterations - brightness, colour and contrast can all be changed to produce different images for training. This could be a random colour shift as in the VGG model [32] or calculated colour changes based on Principal Component Analysis as in AlexNet [28]. GoogLeNet [29] and AlexNet [28] both also applied some brightness level changes.

Another approach to Data Augmentation is to take randomly selected cropped sections of images (often referred to as patches) with fixed aspect ratios as in GoogLeNet [29]. The model is trained on all the patches and the prediction scores from the Softmax layer are averaged across the patches for the original image. A similar approach can be taken on the testing data as in AlexNet [28]. Another approach called multiscaling used in VGG [32] and GoogLeNet [29] for test data involves resizing the images to different scales, cropping where necessary, and averaging the predictions as before.

Data Augmentation has been used in studies on mammography classification including [31] where images were augmented with random rotation between 0 and 45 degrees and a horizontal flip was applied (see Figure 2.7 for examples). Data Augmentation has been seen to improve the accuracy of a U-Net model used for mammogram classification. Zeiser and Costa et al. achieved 85.96% accuracy for a 5-depth network with augmentation in comparison to 70.26% accuracy for the same 5-depth model without augmentation [33]. The augmentation applied included horizontal flips, zoomed sections and ROI extraction in 256x256 pixel patches [33].

2.3.4 Regions of Interest

Other studies on mammography classification that use other machine learning techniques such as Random Forest Classifiers [31] and Support Vector Machines [25] carry out extensive preprocessing to identify ROIs to aid the classification. This

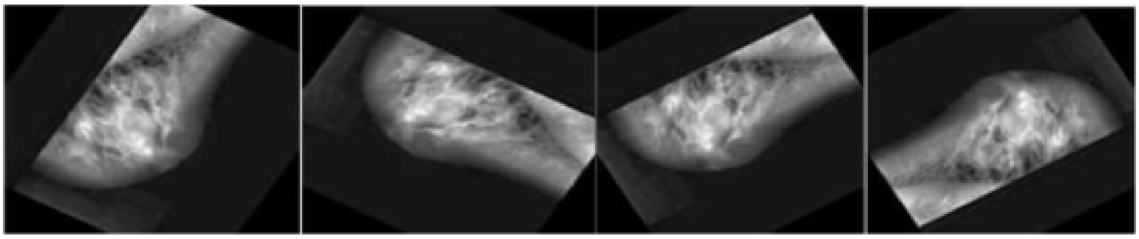


Figure 2.7: Figure from Bokade and Shah et al. Data Augmentation Example [31]

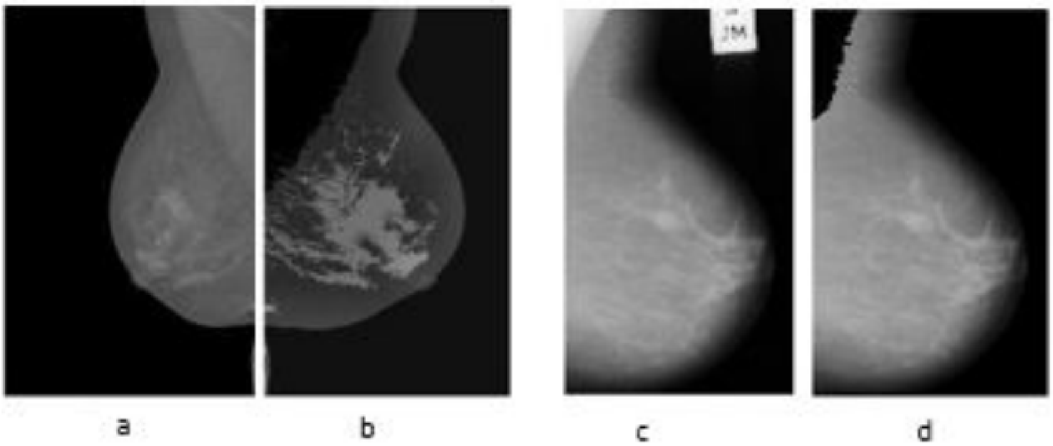


Figure 2.8: Figure from Lbachir et al Pectoral Muscle Removal: (a) and (c) original dataset images from INbreast and MIAS respectively, (b) and (d) images after preprocessing with pectoral muscle removed [34]

preprocessing involves Otsu's thresholding, k-means clustering and histogram-based investigations to find suspicious lesions. Since CNNs are designed to identify these features independently through training - this level of detailed preprocessing will not be required. However, one additional step was included in [25] to use region growing to remove the pectoral muscle from DDSM and MIAS mammogram scans before applying machine learning. This leaves only the breast tissue present in the image as the most important part for the model to learn on (see Figure 2.8). This technique could be particularly useful here, not only to save training time on a cropped image but also to create a better match between the information present in a DDSM image and a CMMD image. In the CMMD images, the pectoral muscle is generally not visible so this data would not be learnt in training.

2.4 Hyperparameter-Tuning Techniques

2.4.1 Transfer Learning

Transfer Learning is a technique employed to make use of a previously trained model for a similar classification task. Often this is particularly useful when datasets for a problem domain are limited since the model has already been pre-trained on a more general image set. The method uses a standard model trained on a general dataset (such as ImageNet [5]) or one for a related specific task. The trained network is then fine-tuned on the dataset in the desired domain to fit the new classification problem with some alterations to the network architecture and parameters.

The alterations made to the pre-trained network should be chosen to help the model classify specific details of the new problem while maintaining the valuable information learnt in its pretraining. In a CNN the first layers in the network will learn general features such as edges and textures. The deeper into the network the more detailed the composite features become. To change the classification task it can be beneficial to remove one or more of the final layers in the model and 'freeze' the first layers to preserve the learnt weights [35]. This means the weights in these frozen layers will not be updated through backpropagation during training. The removed layers should be replaced with new layers for classifying the new problem and trained on the new dataset to set the weights in these non-frozen layers.

Often a second training stage is applied where some or all of the layers of the network are unfrozen and the whole model is trained on the new dataset with a smaller learning rate (to avoid changing previously learnt weights too much) - this is known as fine-tuning.

Paper	Dataset and Model	Random Search	Grid Search	Bayesian Optimisation	Genetic Algorithm	Hyperparameters Tuned	Advantages	Disadvantages
[36]	Transfer learning with pretrained Inception-v3 model. Fine-tuned on 1000 mammograms from DDSM.	✗	✓	✗	✗	Batch Size (8-16), Learning Rate (0.01-0.001), Loss Function (categorical cross entropy, MSE), layers frozen (50,172,249). Optimisers (Rmsprop, SGD, Adadelta)	Many accuracy and loss results allow best parameter combination to be selected and find which are most influential.	Time consuming (brute force search of all permutations of the range of hyperparameter values). Requires selection of suitable value ranges.
[37]	Deep CNN - preprocessing, LeNet-5 model with training, feature extraction and classification. DDSM images used for fine-tuning.	✗	✓	✗	✗	Training: learning rate, mini batch-size, number of epochs, momentum coefficients Model: pooling method, kernel size, number of filters per layer, size of pooling region, number of neurons in fully connected layer, activation function.	Reduce training time and improve model accuracy	Range of values to test limited by available compute power

Paper	Dataset and Model	Random Search	Grid Search	Bayesian Optimisation	Genetic Algorithm	Hyperparameters Tuned	Advantages	Disadvantages
[38]	CBIS-DDSM, classifiers: naive Bayes and support vector machine. AutoML TPOT pipeline for hyperparameter tuning	✗	✓- control	✗	✓	Grid search for SVM, NB and Multi-Layer Perceptron e.g. params for SVM: C, gamma, kernel. Genetic algorithm with same 4 models.	Higher accuracy from genetic algorithm compared to grid search. More complex selection procedures lead to less time in tuning hyperparameters.	More complex model requires more training time - grid search takes less time to configure.
[39]	"Ensemble-based deep transfer learning with classifiers namely optimizable k-nearest neighbours, optimizable naive Bayes, optimizable ensemble and optimizable support vector machine" [39]. Feature extraction prior to classification with transfer learning on pretrained CNNs. MIAS (322 images), INbreast (410 images)	✗	✗	✓	✗	Multiclass method, box constraint level, kernel function, standardise data.	Bayes Optimised models had high accuracy e.g. 99.689% for MIAS and 98.883% for INbreast for SVM classifier. Optimal hyperparameters identified at iteration 17 for MIAS and 5 for INbreast - fast convergence informed by probabilistic model.	No comparison without the use of Bayesian Optimisation or another method so difficult to determine impact of this technique.

Paper	Dataset and Model	Random Search	Grid Search	Bayesian Optimisation	Genetic Algorithm	Hyperparameters Tuned	Advantages	Disadvantages
[40]	1070 images from DDSM. Pretrained Inception-v3 CNN for classification via transfer learning (fine-tuned with DDSM). Two models with different architectures (e.g. layers, inputs and number of neurons in final layer). RandomSearch and GridSearch combined - GridSearch for further improvement	✓	✓	✗	✗	Batch size (8,16), number of layers to freeze (50,120,249), learning rate (0.01, 0.001), optimizers (Adadelta, RMSProp, SGD), loss function (CCE and MSE)	Random Search can be applied to find initial combination of parameters quickly. GridSearch used to fine-tune in first pass on all parameters, then second pass focusing only on learning rate, loss function and layers to freeze (higher impact).	Somewhat based in trial and error to find suitable ranges for parameters in grid search.

Table 2.2: Comparative table of hyperparameter optimisation techniques in related work on mammogram lesion classification.

2.4.2 Hyperparameter Optimisation

As well as freezing layers in a pre-trained network and training specific model layers with a new dataset, there are additional techniques that can be used to find the best-performing set of hyperparameters for a model. Hyperparameters alter the learning performance of a model but are generally selected manually before training. Unlike other model parameters, hyperparameters are not altered during the training process but are chosen by the programmer. Selecting the best hyperparameters often requires trial and error so techniques including GridSearch, RandomSearch, Bayesian Optimisation and Genetic Algorithms can be used to search for the best values. These techniques can be applied with k-fold cross-validation of the model to average the performance metrics across the validation set.

GridSearch involves choosing ranges of hyperparameter values predicted to do well based on similar research and iterating over them to select the best combination. An $n \times d$ grid is constructed where n is the set of hyperparameters and d is the range of values. Search tries each combination of values in the grid to identify the best performing set [41] [42]. The range of values is only limited by the computing power needed to perform search.

RandomSearch includes initialising a pool of hyperparameter values and randomly selecting sets of values. This approach can be time-consuming but it can give rise to unexpected combinations of values that perform better than grid search for the specific model [41] [42].

Unlike RandomSearch and GridSearch, Bayesian Optimisation methods are informed by previous iterations to select new hyperparameter values. In Bayesian Optimisation, a “surrogate” function is constructed as a probabilistic model mimicking the true complex objective function of the model. As new hyperparameter sets are tested the results are used to update the “surrogate” function and improve convergence towards the real objective function [39]. This allows the optimiser to make closer hyperparameter value selections in fewer iterations than Random or Grid Search [43].

Genetic Algorithms can also be used for hyperparameter optimisation and will make informed selections of hyperparameter sets based on previous scoring. Genetic Algorithms mimic natural evolution, employing a “survival of the fittest” approach to selecting features [42]. A scoring mechanism, such as minimising a loss function, is used to evaluate the fittest model at each generation and different combinations of parameter values are inherited or added through mutation. This introduces randomness to the model but also allows high-performing parameter values to be sustained throughout the evolution [44].

2.4.3 Related Work in Mammography

Table 2.2 offers a summary of related work with different hyperparameter tuning techniques in machine learning mammography studies. The studies with network architectures most similar to that proposed here (a CNN with transfer learning and preprocessing) used GridSearch and RandomSearch ([36] [37] [40]). In general, GridSearch was more commonly used for fine-tuning although in one paper [40] RandomSearch was used first to select an initial parameter set and GridSearch was applied afterwards for additional fine-tuning of specific parameters. GridSearch has shown success in this type of classification problem - selecting the best-performing set of parameters in pre-specified ranges in [36] for the SGD optimiser improved accuracy from 0.87 (for worst performing parameter set) to 0.97 (for highest performing set). The main limitations of GridSearch are the need to select sensible value ranges for each hyperparameter and the computational cost of searching every combination of parameters in those given ranges.

The papers focused on more complex hyperparameter tuning algorithms also implemented a more complex pipeline. For example, [39] performed "ensemble-based learning" with CNN deep learning for feature extraction, bayesian optimisation for hyperparameter tuning and a selection of classifiers including k-nearest neighbour and SVM. This type of complex model produced high test set accuracy (99.689% for MIAS dataset) on the selected highest-performing classifier (SVM) however it was difficult to attribute this performance impact to the hyperparameter tuning when there was no comparison with other methods. Siti Fairuz Mat Radzi et al. used an AutoML implementation referred to as TPOT (tree-based pipeline optimization tool) for hyperparameter tuning with a Genetic Algorithm but they also tested model selection and compared the results on different classifiers: Naive Bayes, Support Vector Machine and Multi-Layer Artificial Neural Network [38]. Here the experimental setup gave a clear conclusion that the parameter set chosen by the Genetic Algorithm had a higher accuracy than the set chosen by grid search. Both Genetic Algorithms and Bayesian Optimisation showed promising results in terms of high accuracy in the chosen model (0.923 for Genetic Algorithm in [38] and 0.997 for Bayesian Optimisation [39]).

Although Genetic and Bayesian hyperparameter tuning algorithms may find higher-performing parameter sets more quickly, the pipelines used to implement these are often more complex and need to be compared to GridSearch as a control to understand the real performance benefit here. GridSearch has demonstrated good results in mammography classification and remains a sensible choice of hyperparameter optimisation when ranges of values for hyperparameters can be selected from prior research. Random Search can be seen to find good combinations of hyperparameters faster than GridSearch but may not find the optimal combinations.

Chapter 3

Methodology

3.1 Approach

The initial approach to the investigation was to attempt to recreate Adam Jaamour's results from his MSc dissertation on classifying mammogram cases [4]. Jaamour was a previous student of this project's supervisor and wrote a Masters thesis with a well-structured pipeline that could be recreated as a baseline model for further investigation (code openly available: [8]). Due to the time constraints of a Masters project, it was useful to have a starting point from previous research to build new investigations with support from a shared supervisor.

For clarity, a brief note on the use of the terms "baseline" and "base model" is necessary for the rest of this report. The term "baseline" is used here to refer to the complete initial model selected to recreate Jaamour's model [4] that was used as a control for comparison with all other trials in the investigations. The "base model" is a common term in literature to refer to the pre-trained model in Transfer Learning e.g. MobileNet, ResNet, VGG-19 etc.

The first step was to build a basic machine learning pipeline with a similar structure to Jaamour's and fine-tune it on the CBIS-DDSM dataset [2] for training and evaluation. This model formed a baseline for the investigations following the reported best-performing model from Jaamour's research: namely, a MobileNetV2 transfer learning model, pre-trained on imagenet [5] and fine-tuned with DDSM [4]. By focusing on DDSM alone initially, the model could be tweaked and improved to better match Jaamour's model since he used the same dataset. The next step was to combine DDSM and CMMD [3] data to form a hybrid train, validate and test sets with proportional distributions of each dataset. The transfer learning model described above was used to gather baseline accuracies with the datasets for comparison in further investigation.

The main goal of this research was to investigate the impact of Jaamour's cited future work considerations: including a more complex pre-processing pipeline and hyperparameter-tuning such as grid search [4]. The objectives outlined in the Introduction, Chapter 1, were used as a basis for the investigations introduced later in this chapter. First to build a basic CNN to gather baseline results from DDSM and both datasets combined, then an investigation of whether introducing pre-processing techniques can improve the accuracy of the models. Further investigation looked into the use of different network architectures and hyper-parameter tuning techniques. The combination of best results from these investigations were proposed as a new best-performing model and compared with Jaamour's results as a follow-up to their future work suggestions.

3.2 Pipeline

The pipeline requires both the CBIS-DDSM and CMMD datasets to be converted from DICOM to PNG files and stored in appropriate directory structures to allow the training, testing (and for CMMD validation) sets to be created using tensorflow. This directory structure typically contains a top-level directory representing the set (e.g. Train) and sub-directories for each class (e.g. Benign and Malignant). From this structure tensorflow's `image_dataset_from_directory` function can create a Dataset object with images and labels. Scripts for moving images into this directory structure, converting DICOM to PNG and splitting into train/test/validate can be found in the *data_preparation* folder in *src*. See the Usage guide in Appendix A for a detailed description of these scripts, the desired directory structure and program arguments. The project code is openly accessible on Github: `https://github.com/RAMcCracken/CS5199_Breast_Cancer_Detection_Project`.

The pipeline can run with either dataset alone but for the investigations, it combines both CBIS-DDSM and CMMD datasets into hybrid training, testing and validation sets with an equal proportion of each dataset. Selected pre-processing techniques are applied to the input image datasets such as data cleaning, augmentation, integration and basic ROI extraction. Transfer learning was selected based on Jaamour's research (to allow comparison of final results) [4] and because of its popular use discussed in the Literature Review, Chapter 2. The transfer learning models were pre-trained on Imagenet [5]: providing weights for a generalised computer vision task that could be re-used for classifying mammogram images with appropriate fine-tuning. The initial hyper-parameters were selected to match Jaamour's best performing model. Training consisted of 100 epochs with batch size of 32 while all but the new fully-connected layers were frozen and a further 50 epochs of fine-tuning with a smaller learning rate and all layers unfrozen. The initial learning rate was 1×10^-4 and the fine-tuning learning rate was 1×10^-5 (see Table 3.1 for a summary of these chosen parameters). After fine-tuning, the models could be evaluated on an unseen test set or on the validation set (during investigations) and performance metrics were calculated to allow for comparison among models. See Figure 3.1 for a diagram of the pipeline.

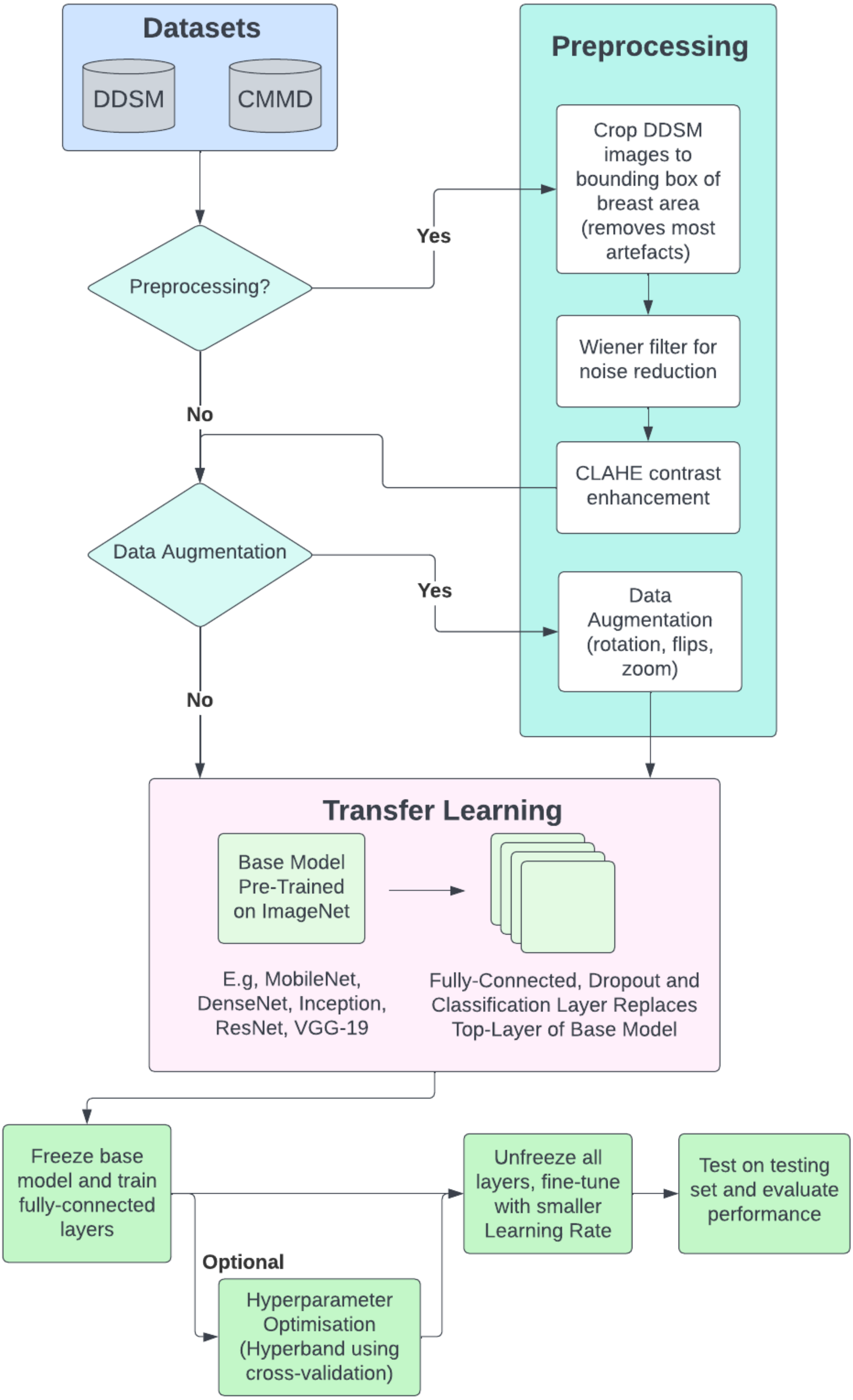

Datasets
DDSM
CMMD
Preprocessing
Crop DDSM images to bounding box of breast area (removes most artefacts)
Wiener filter for noise reduction
CLAHE contrast enhancement
Data Augmentation (rotation, flips, zoom)
Preprocessing?
Yes
No
Data Augmentation
Yes
No
Transfer Learning
Base Model Pre-Trained on ImageNet
E.g, MobileNet, DenseNet, Inception, ResNet, VGG-19
Fully-Connected, Dropout and Classification Layer Replaces Top-Layer of Base Model
Freeze base model and train fully-connected layers
Optional
Hyperparameter Optimisation (Hyperband using cross-validation)
Unfreeze all layers, fine-tune with smaller Learning Rate
Test on testing set and evaluate performance

Figure 3.1: Flowchart of pipeline used in the following investigations - including branching for optional steps.

3.3 Datasets

3.3.1 CBIS-DDSM

The CBIS-DDSM [2] dataset comes with a pre-determined train/test split where the test set makes up 20% of the data. A validation set was generated by taking 20% of the training set. CBIS-DDSM contains 3047 mammography scans from 1566 patients with mediolateral-oblique and craniocaudal projections for each patient (see an example of these projections in Figure 3.2). This dataset uses an older modality of mammography storing digitised versions of film scans since it is a modified version of the original DDSM dataset collected from an American population [45]. The dataset contains images of masses and calcifications separately and combined. For the following investigations, all images were used rather than separating mass and calcification images.

3.3.2 CMMD

The CMMD dataset does not have a predetermined train/validation/test split so scripts were written as part of the pipeline to create a separation of this data stratified by patient. The dataset contains 3,728 studies from 1,775 patients and was collected in 2016 [3]. Each patient has a minimum of two samples in the dataset (images are provided in the mediolateral-oblique and craniocaudal projections). Patients may also have 4 samples if scans were taken from both the left and right breast. This dataset contains scans from full-field digital mammography on a Chinese population. The second part of the dataset (patient IDs of the form D2-XXXX) contains 4 images per patient and with an unclear classification. The D2 section of the dataset was removed from the model for the basic pipeline since its true labels could not be confirmed from online research. This left a total of 2212 images for training and testing.

3.3.3 Combined

In the investigations, CMMD and CBIS-DDSM datasets were combined to form train/validate and test sets with an equal distribution from each dataset. This approach was selected to produce a model that could generalise to datasets from two geographical regions (American and Chinese populations) and two different imaging types (digital and film-screen mammography). The benefit of this approach is that the model is likely to perform better for both datasets in the testing stage since it has encountered both image types and regions in training and testing. The alternative would be to train on only one dataset and test on another, for example training on DDSM and testing on CMMD, but this was unlikely to generalise well. The main drawback of this approach is that taking data from the same datasets in the test phase will not test the model on very different new data. However, the test set remained unseen during training and validation and the pipeline ensures the avoidance of data leakage when constructing these sets by stratifying by patient. In practice, it is very difficult to build a model with enough varied data to test it on truly new data with accurate results.

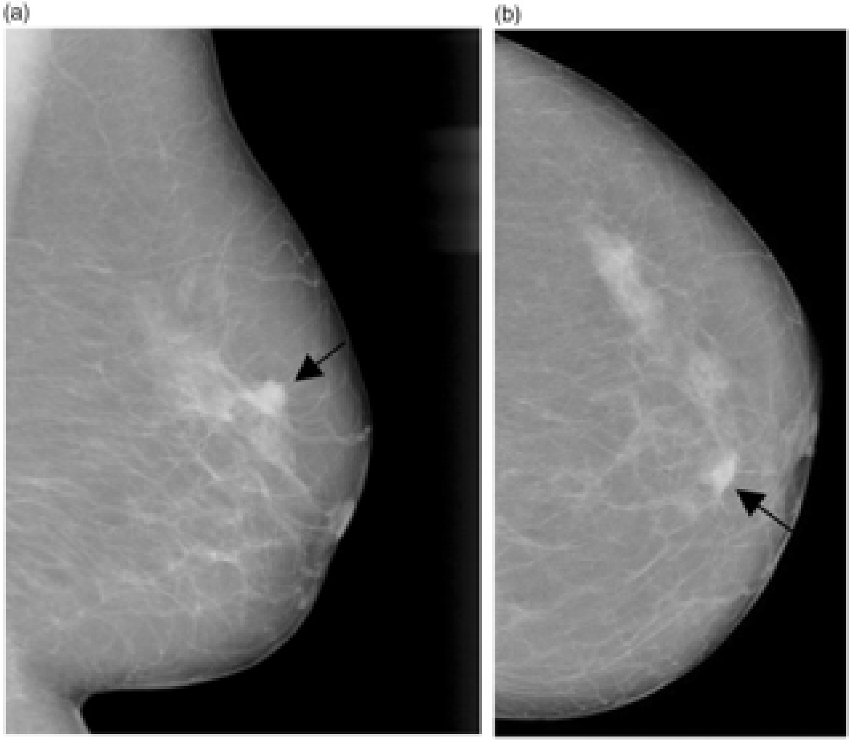


Figure 3.2: (a) Mediolateral-Oblique (b) Craniocaudal projections of a mammogram showing a mass. From Fujiwara et al. [46]

3.4 Technology

To develop the pipeline: Tensorflow and Keras were used in Python for handling the large datasets and implementing CNN models. The code was developed and run in a GPU-accelerated Docker container supporting the most recent version of Tensorflow at time of writing (2.8.3). A University-provided GPU was used for the development and running of this program (referred to throughout later sections as the "original GPU machine" since a second machine was used later with more GPU cores (known as the "DGX machine"). This additional machine was allocated by the University to allow models to be trained with larger image sizes in a more reasonable time frame.

3.5 Performance Metrics

During investigation, the test set was concealed completely and the validation set was used to gauge the performance of the models. For each trial: graphs of validation accuracy and loss (Binary Cross Entropy loss function) were generated for the frozen and unfrozen (fine-tuning) phases of training. At the end of training, the model was evaluated on the validation set to produce performance metrics: accuracy, precision, recall and f1-score - defined in the sections below. Sci-kit learn `sklearn.metrics.classification_report` was used to provide a summary of class-based metrics. A confusion matrix was also generated for each trial to show the class balance of predictions.

In the equations in this section: TP represents True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

3.5.1 Accuracy

Accuracy measures the number of correct predictions out of all predictions made and is a simple measure of performance in a neural network. This metric can be supplemented by a confusion matrix to understand the class balance in the predictions made (e.g. not all correct predictions skewed to one class). See equation 3.1, Gupta et al. [47]

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{3.1}$$

3.5.2 Precision

Precision determines the accuracy of positive results (in this case Malignant cases). See equation 3.2, Gupta et al. [47].

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

3.5.3 Recall

Recall determines the number of positive (Malignant) cases correctly predicted of all cases that should have been predicted as positive. This metric is especially important in this domain since false negatives lead to missed cancer cases so we need to ensure recall is as high as possible. See equation 3.3, Gupta et al. [47].

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

3.5.4 F1-Score

F1-Score gives a harmonic mean of precision and recall. This metric aims to be high when precision and recall are both high and is a useful combination of these two metrics to find a balance between them. See equation 3.4, Gupta et al. [47].

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{3.4}$$

3.6 Hyperparameter Choices

The initial selection of hyperparameters formed the baseline for all further investigations. Since the investigations build on the work of Jaamour [4], many of the hyperparameters were chosen to recreate his best-performing model to later find future improvements. A list of hyperparameters with a justification of their choice and comparison with Jaamour's can be found in table 3.1.

Hyperparameter	Value	Justification
Learning Rate (Initial)	1×10^{-4}	Taken from Jaamour's best model. Other learning rates investigated in Pre-processing investigation
Learning Rate (Fine-Tuning)	1×10^{-5}	From Jaamour's best model, very small LR for slow adjustments in training the unfrozen network
Batch Size	32	Jaamour used batch size 2 with large image sizes (512 × 512). Larger batch size of 32 acceptable here since training is faster with smaller image size choice. In literature [12] uses batch size of 64, [37] uses batch size 32 and [15] uses batch size 16 so the chosen batch size is the centre of this range.
Max No. Epochs (Frozen)	100	Matches Jaamour's model, allows early stopping.
Max No. Epochs (Unfrozen)	50	Matches Jaamour's model, allows early stopping.
Base Model	Mobile Net v2	Jaamour's best model used MobileNet - other models investigated.
Fully-Connected Layers	Dropout (rate 0.2), Relu (512 units), Relu (32 units), Sigmoid output	Architecture matches Jaamour's best model, additional dropout layers were added and dropout rate increased to 0.4 in an attempt to reduce overfitting but no improvement was seen in loss and accuracy. ReLu is a common activation function used in deep networks for non-linearity [48]
Augmentations	Horizontal, Vertical flips, rotations of 90, 180 and 270°	Two of the papers referenced in the literature review applied augmentation by rotation of 90, 180 and 270°[15][12]. Horizontal and vertical flips were chosen as sensible because one creates an artificial 'left' and 'right' view and the other inverts features without distorting them.
Image Size	160 × 160	Jaamour used images of 512 × 512 the image size was reduced for faster training but larger sized images were investigated on a machine with more GPU cores
Loss Function	Binary Cross Entropy	Matches Jaamour's model, standard categorical loss function for binary output
Optimizer	Adam	Matches Jaamour's model

Table 3.1: Hyperparameter Choices in Base Model. Jaamour's code can be found on Github [8]

Chapter 4

Validation Set Investigations

In this section the investigations carried out on the validation dataset will be described with an outline of the findings. The investigations covered are: reducing overfitting in the Baseline CNN in Section 4.1, an investigation of a data pre-processing pipeline and data augmentation in Section 4.2, exploration of different base models in 4.3 Network Architectures and finally Section 4.4 Hyper-Parameter Tuning. These investigations build on one another such that the final models tuned in the Hyper-Parameter Tuning section were the best-performing models after the previous investigations.

4.1 Baseline CNN

The basic CNN pipeline was built as a baseline to get initial results using DDSM alone and the combined dataset. The basic model was designed to replicate Jaamour's best-performing model [8]. The CNN architecture consisted of a MobileNetV2 transfer learning base model [49] with pre-trained imagenet weights, fine-tuned on our training dataset. The classification layers were excluded from the MobileNet model and fully-connected layers for breast-cancer classification were added. These were flatten, dropout and dense layers with a ReLu activation function and a final layer Sigmoid function to make binary classification predictions. During training, the base layers were frozen and the fully-connected layers were trained using the pre-trained weights of the network. In the second training phase, all the layers were unfrozen and a smaller learning rate was used to fine-tune all the layers of the network on the given dataset. In both training phases, cross-validation was applied and the early stopping technique was used to stop learning when validation loss did not improve over a specified number of epochs (known as 'patience'). Another technique was also applied to reduce the learning rate by a small factor on a plateau - when validation loss did not improve after half of the 'patience' number of epochs.

From the validation set results, the baseline model overfits for the combined CMMD and DDSM dataset with a disparity of approximately 15% in training and validation accuracies with an image size of 500 × 500 pixels (see Figure 4.1). The validation accuracy of the baseline model with combined dataset and an image size of 160×160 pixels was 82.3% and for an image size of 500 × 500 pixels was 83.7%. The model performs very well on DDSM alone with a validation accuracy of 99.4% with image size 500 × 500 pixels (see graph in Figure 4.2). This validation result significantly outperforms Jaamour's result of 67.08% testing accuracy [4] although our accuracy is likely to decrease on the unseen test set for our model. The model does not perform particularly well for CMMD alone due to the lack of data used.

Since the model reaches a higher validation accuracy with DDSM alone than with the combined dataset - this suggests that the combined dataset may be introducing too much variation for the model to identify similar visual features. The two datasets: DDSM and CMMD are different imaging modalities (one film and one digital) so the images have different visual features which may confuse the model. It is also likely that biological or anatomical differences exist between the datasets from two geographical populations, American and Chinese, which could create a lot of variation.

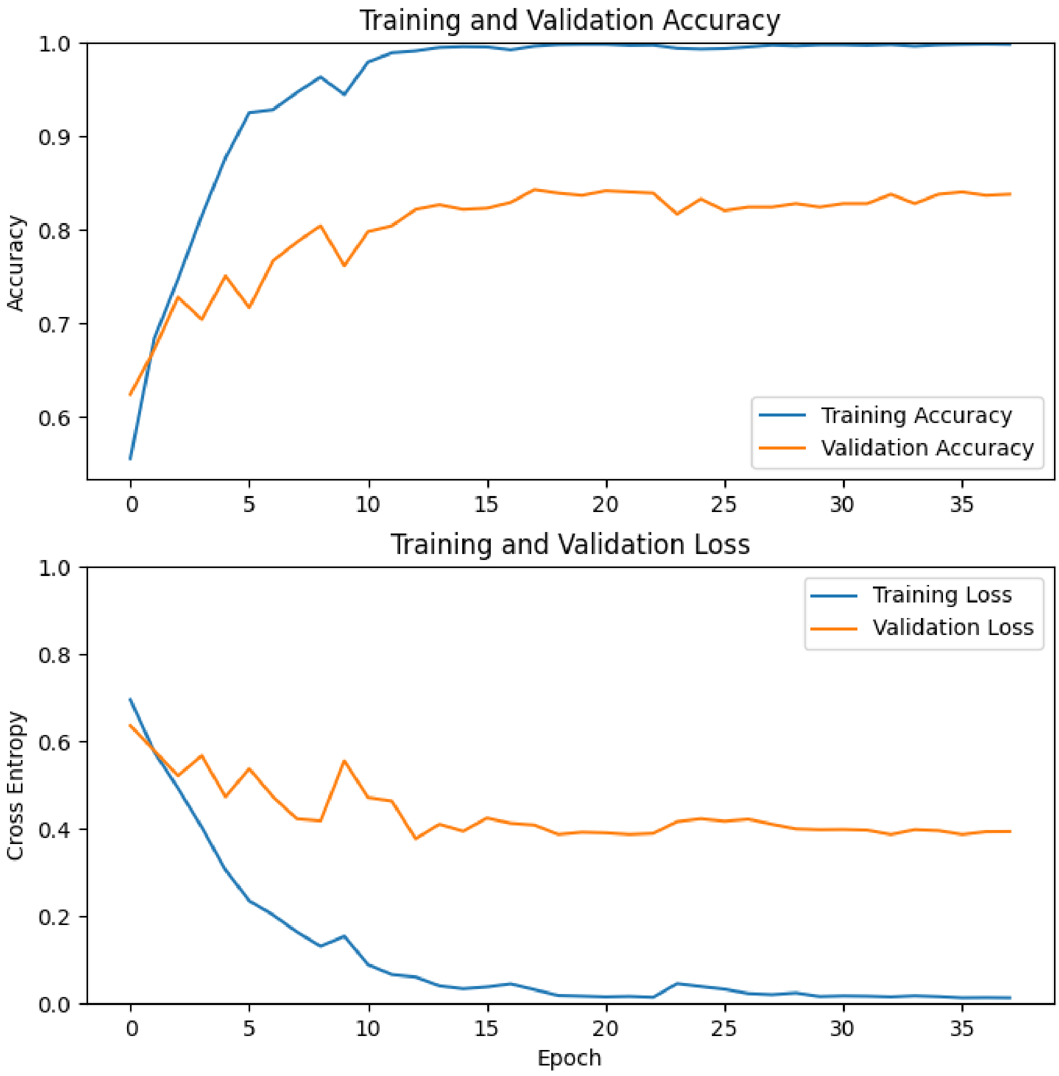


Figure 4.1: Baseline Model - trained on combined dataset using image size of 500 × 500 pixels, learning rate of 0.0001

Since the model overfits for the combined dataset using the same parameters as Jaamour [4], some techniques were applied to attempt to generalise the model and reduce the difference between validation and training accuracy. The first of these techniques was to increase the dropout rate in the dropout layer from 0.2 to 0.4. The second approach was to add one additional dropout layer in the full-connected layers of the model (with both dropout layers using a dropout rate of 0.4). Both of these techniques aimed to increase random data loss during training to prevent the model from fitting too closely to the training data. A final technique was to apply data augmentation to the baseline model to increase the number of training samples and aim to generalise the training fit. Table 4.1 gives a summary of validation accuracy after these attempts for an image size of 160 × 160 (on the original GPU machine) and 500 × 500 pixels (on the DGX 8-core GPU). The original baseline was seen to have the best validation accuracy. In the accuracy graphs, there was little change to training accuracy except for the attempt with data augmentation which lowered the overall accuracy of the model and did not reduce overfitting. See Appendix C.1 for the original baseline compared with augmented data.

Experiment	**160 × 160 pixels**	**500 × 500 pixels**
Baseline, Dropout Rate 0.2	82.3	**83.8**
Baseline with Augmentation	71.6	78.1
Baseline, Dropout Rate 0.4	80.7	82.0
Baseline, Additional Dropout Layers, Dropout Rate 0.4	79.4	83.4

Table 4.1: Summary of Validation Accuracies (%) for Baseline Models with Different Image Sizes

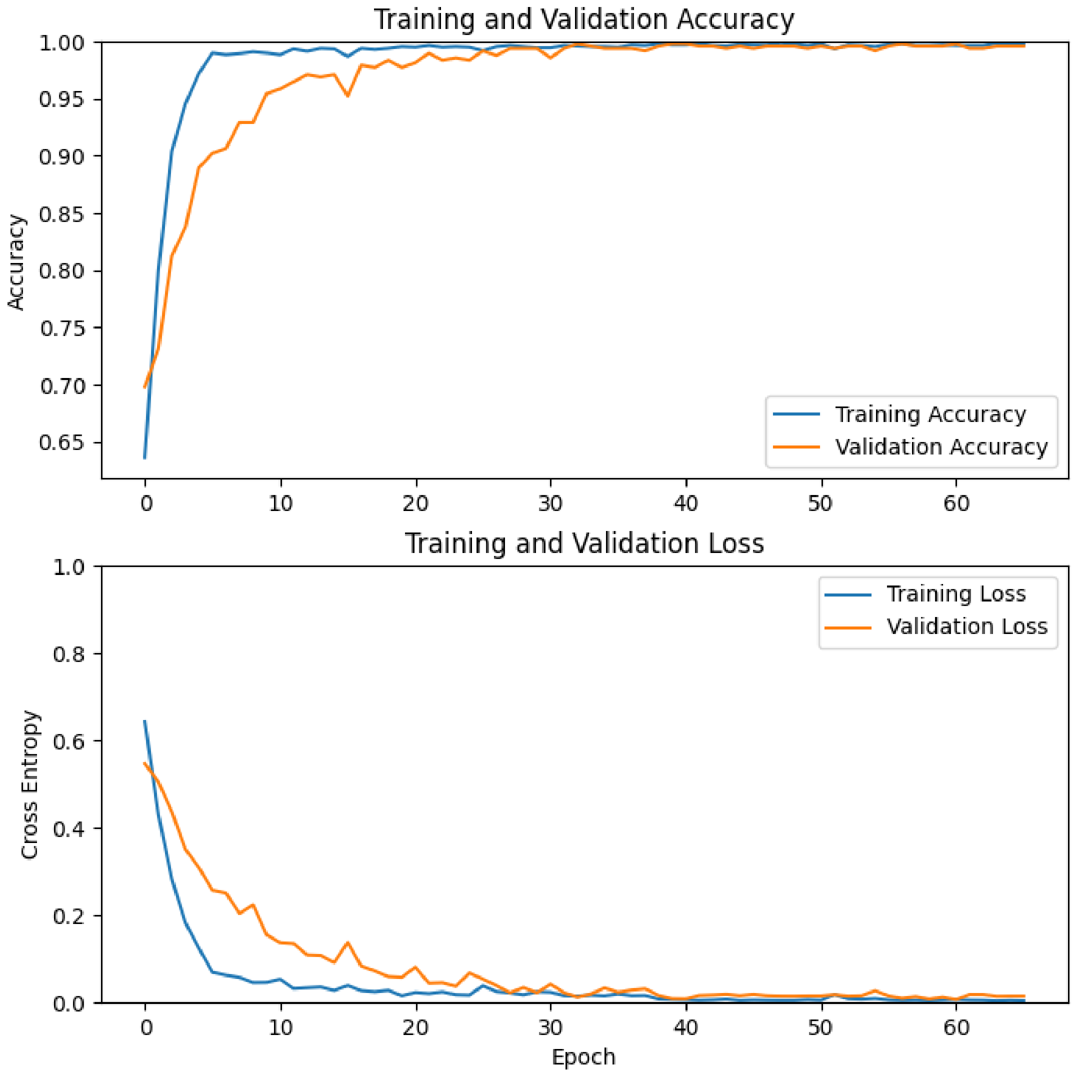


Figure 4.2: Baseline Model - trained on DDSM dataset using image size of 500 × 500 pixels, learning rate of 0.0001

4.2 Pre-Processing

4.2.1 Experimental Setup

The pre-processing investigation applies techniques commonly seen in mammography machine learning investigations; as identified in the Literature Review, Chapter 2. These techniques include: data cleaning to remove artefacts in the DDSM images, reduction of background noise and application of contrast enhancement and data integration to ensure images from both datasets are the same shape and size. Data augmentation was applied to increase the size of the dataset. Crude ROI selection was used to crop images to the bounding box of the breast area, using image processing techniques to identify this area in each image. Removal of the pectoral muscle was referenced as a technique in the literature review, Figure 2.8 however this was too difficult to apply in this short investigation given the time constraints. To compare the results of different pre-processing techniques, the baseline model and all hyperparameters described above were kept the same throughout this investigation. This allowed a comparison of the accuracy of the model before and after different pre-processing techniques with the combined dataset. The pre-processing pipeline created can be seen in Figure 4.3.

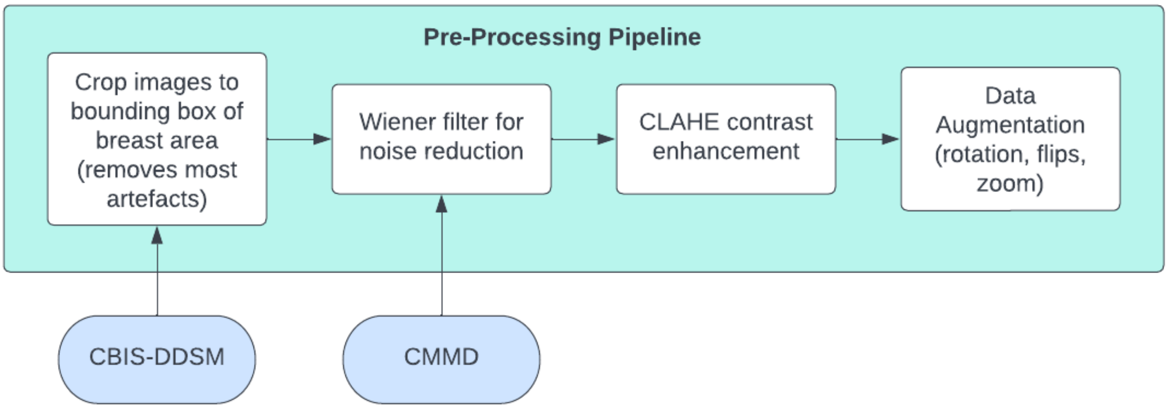

Pre-Processing Pipeline
Crop images to bounding box of breast area (removes most artefacts)
Wiener filter for noise reduction
CLAHE contrast enhancement
Data Augmentation (rotation, flips, zoom)
CBIS-DDSM
CMMD

Figure 4.3: Initial Full Pre-Processing Pipeline

The pre-processing pipeline only applied image cropping to DDSM images (not CMMD) as a simple approach to remove the majority of the image background and artefacts on the film scans (e.g. radiographers' labels). Since the CMMD images are digital scans they have far less visible background noise and no artefacts such as handwritten labels. To implement this cropping - traditional image processing techniques including edge detection and open-cv contour detection were used to draw an approximate outline of the breast tissue in the images. A heuristic was used to easily select this contour from other edges in the image as the longest edge (since this outlined area is the largest shape in the image). To improve the effectiveness of this technique a small border was cropped away from each image before edge detection to remove artefacts at the image boundaries. Once the breast area had been identified the bounding box of the contour was found. The image was cropped to the bounding box area plus a small border (to allow for imperfect edge detection and avoid loss of important data). Figure 4.4 shows the steps of this process and a successful crop removing a textual artefact in the mammogram scan. This method was not always effective since sometimes artefacts could fall partially within the bounding box before cropping - however random sampling was carried out on the dataset and the majority of images seen were cropped successfully.

The following pre-processing steps were applied to both the DDSM and CMMD images (see Figures 4.5 and 4.6). First, a 3 × 3 Wiener filter with a signal-to-noise ratio (SNR) of 0.4 was applied as a noise reduction technique. Following this, a CLAHE contrast enhancement algorithm with a clip limit of 5 was applied to highlight the lesions in the mammograms. Finally, data augmentation was added to the model (in Tensorflow augmentation can happen at train time to use GPU acceleration). The augmentation layers applied included horizontal and vertical

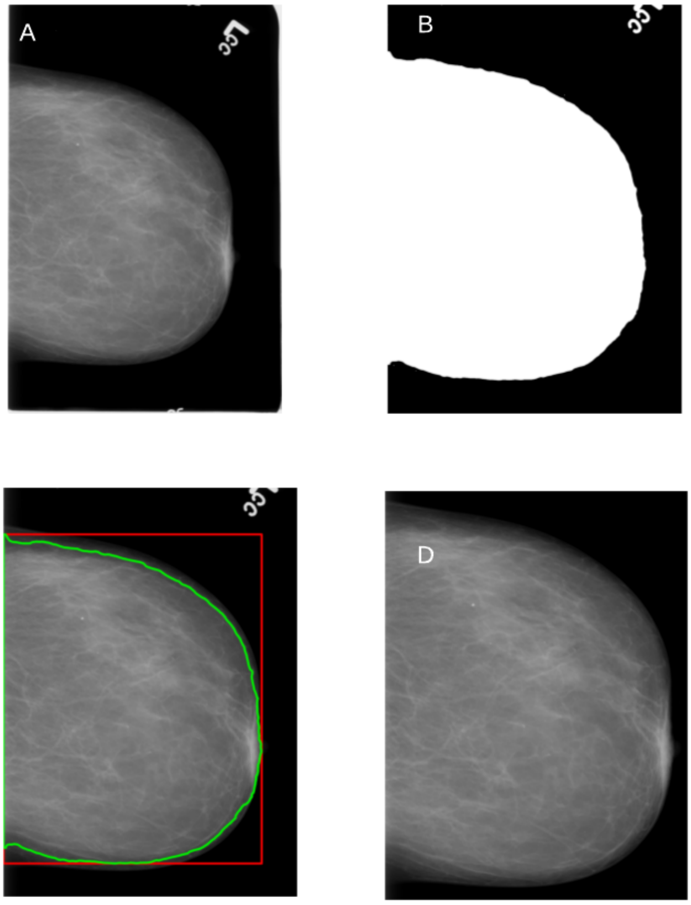


Figure 4.4: Process of cropping a DDSM image: A) original image, B) Otsu threshold applied, C) breast area edge identified and bounding box, D) final cropped image

flips, and random rotation of 90, 180 or 270 degrees.

4.2.2 Investigation Trials

The first trial of the investigation was to run the baseline model with all the set hyperparameters on the pre-processed dataset (i.e. images with the full pipeline applied) and with data augmentation. The hypothesis of this investigation was that learning would improve since the images highlight the important features more clearly (at least to the human eye). Unexpectedly, the final accuracy was significantly lower on the validation set (61.6%) compared to the baseline accuracy (82.3%), with training accuracy finishing close to 80% on the pre-processed data.

The second unexpected result was that the model using the pre-processed data ran significantly faster than the baseline model, despite the addition of data augmentation. The pre-processed model took approximately 30 minutes to complete, although often stopped early in both training phases through Early Stopping. The baseline model took closer to 12 hours to finish. The reason for this disparity in runtime

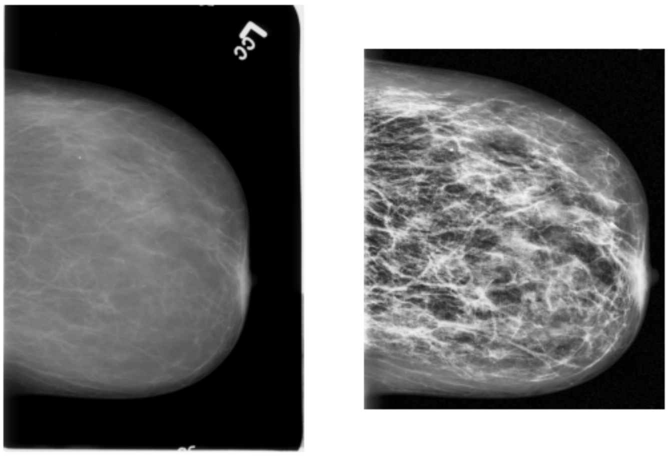


Figure 4.5: Comparison of original DDSM image (left) and image after pre-processing pipeline (right)

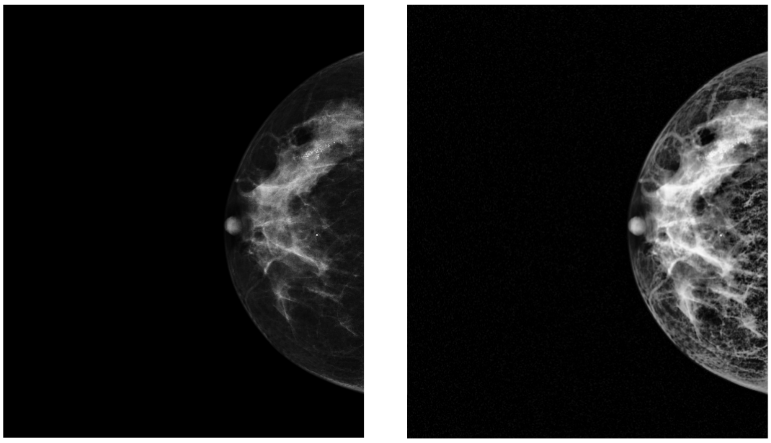


Figure 4.6: Comparison of original CMMD image (left) and image after pre-processing pipeline (right)

is still unclear. It was assumed that it took less time for the program to read in the pre-processed dataset compared to the original dataset since the DDSM images were smaller from cropping. It may also have been the case that a poor choice of model triggered very early stopping for the pre-processed attempt when it wasn't learning successfully.

Following this initial attempt, the pre-processed data was used again with the same model but without data augmentation. Interestingly, this gave a much closer accuracy to that of the baseline model (73.9%) indicating that the data augmentation may not be useful in this model. At this stage of the investigation, the code was altered multiple times to try different types of augmentation (such as cropped, zoomed, rotated and sheared images) but very little difference was seen in the accuracy outcome. The reason for this may be that the augmentation added more significant differences in features than would naturally be present in the images. For example, the model may be learning about the lesion features based on the angle in their appearance where augmented rotations might have altered the appearance of lesions too much compared to the general population.

Castro et al. investigated an approach to rotational augmentation in CNNs for medical image classification where the filters were rotated at convolutional layers instead of rotating the input image [50]. This paper describes the common choice of using 90° intervals for rotations to avoid information loss at corners of the image - as used in our research. Castro et al. demonstrate how weights in the convolutional layers can be rotated in the CNN architecture as augmentation in training giving higher test accuracy than input image rotations for data affected by orientation. Although this paper reported a slight decrease in accuracy for the CBIS-DDSM dataset with the new method - this was applied to small patches of lesions [50]. In our research, weight rotation could have a greater effect on full-size DDSM images, since breast scans are always taken at the same orientation, so this method could be investigated further in future.

In the second trial of the investigation: different learning rates were applied to the model using the pre-processed data with and without augmentation for comparison. The range of learning rates reported is 0.001, 0.0001 (initial choice) and 0.00001, although other smaller and larger learning rates were also tested in practice but gave very poor results. The validation results showed that the learning rate had very little difference on the final accuracy (see Table 4.2). In the investigation without data augmentation, LR of 0.0001 was the best choice and with data augmentation, LR of 0.001 gave a slightly higher accuracy but the graph shows that the model's training accuracy performs worse (see graphs in appendix C.2).

Learning Rate	With Augmentation	Without Augmentation
0.001	**62.2**	64.0
0.0001	61.6	**73.9**
0.00001	62.0	73.2

Table 4.2: Validation Accuracies (%) for Pre-Processing with Different Learning Rates (Image Size: 160 × 160 pixels)

In the third trial of the investigation, a new machine (referred to as DGX) was used with 8 GPU cores, allowing the models to be trained more quickly and larger image sizes to be used. A larger image size of 500 × 500 pixels was chosen since many of the DDSM images were seen to be cropped to approximately 500 × 600 pixels through random sampling. In some cases, the aspect ratio of images changed in resizing but images were never cropped so information was not lost. On the new machine, the same investigations were carried out again with the larger image size. In all the trials, significant improvement can be seen in the learning and final accuracies with the larger images compared to the smaller 160 × 160 pixel images (compare tables 4.2 and 4.3). This indicates that the initial use of such small image sizes may have caused a loss of important information and made it more difficult for the model to learn the features needed for accurate classification. Small features such as calcifications appear as tiny specs on the images but their appearance and formation can be a key indicator of cancer, so these features needed to be preserved when resizing images. See an example comparison of features seen in 500×500 pixel images and features visible in 160×160 pixel images in Appendix D.

The same investigation with different learning rates as above was repeated with the larger image sizes and the results are shown in Table 4.3.

Learning Rate	With Augmentation	Without Augmentation
0.001	51.4	79.1
0.0001	76.8	83.3
0.00001	74.2	**83.5**

Table 4.3: Validation Accuracies (%) for Pre-Processing with Different Learning Rates (Image Size: 500×500 pixels)

To extend this investigation, the two best model hyperparameter combinations from the original GPU with image size 160×160 pixels and the new DGX machine (image size 500×500 pixels) were used with a higher number of fine-tuning epochs to allow them a longer learning time. Originally all the experiments had 50 fine-tuning epochs and this was increased to 100 epochs. The models were trained on the pre-processing set with and without data augmentation and with learning rates of 0.0001 and 0.00001 on both machines (see tables 4.4 and 4.5 for comparison).

Learning Rate	With Augmentation	Without Augmentation
0.0001	62.2	78.1
0.00001	64.5	**79.2**

Table 4.4: Validation Accuracies (%) for Pre-Processing with 100 Fine-Tuning Epochs for GPU machine, Image Size: 160×160 pixels

Learning Rate	With Augmentation	Without Augmentation
0.0001	77.8	81.4
0.00001	69.8	**83.5**

Table 4.5: Validation Accuracies (%) for Pre-Processing with 100 Fine-Tuning Epochs for DGX machine, Image Size: 500×500 pixels

From these investigations, most of the models did not use many additional epochs in fine-tuning, usually stopping early at less than 60 epochs in total. For example, early stopping was triggered before 60 epochs in the models without data augmentation on the DGX machine, using an image size of 500×500 pixels. This indicates that the model had finished learning and did not need additional epochs since the validation accuracy had levelled off (see graph 4.7 for example of the highest performing model with increased fine-tuning epochs).

Overall, the best-performing model found in the pre-processing investigation was using a learning rate of 0.00001, batch size of 32, image size of 500×500 pixels (on DGX machine) and without data augmentation, giving validation accuracy: 83.5%. This accuracy was also obtained with additional fine-tuning epochs (see Table 4.5)

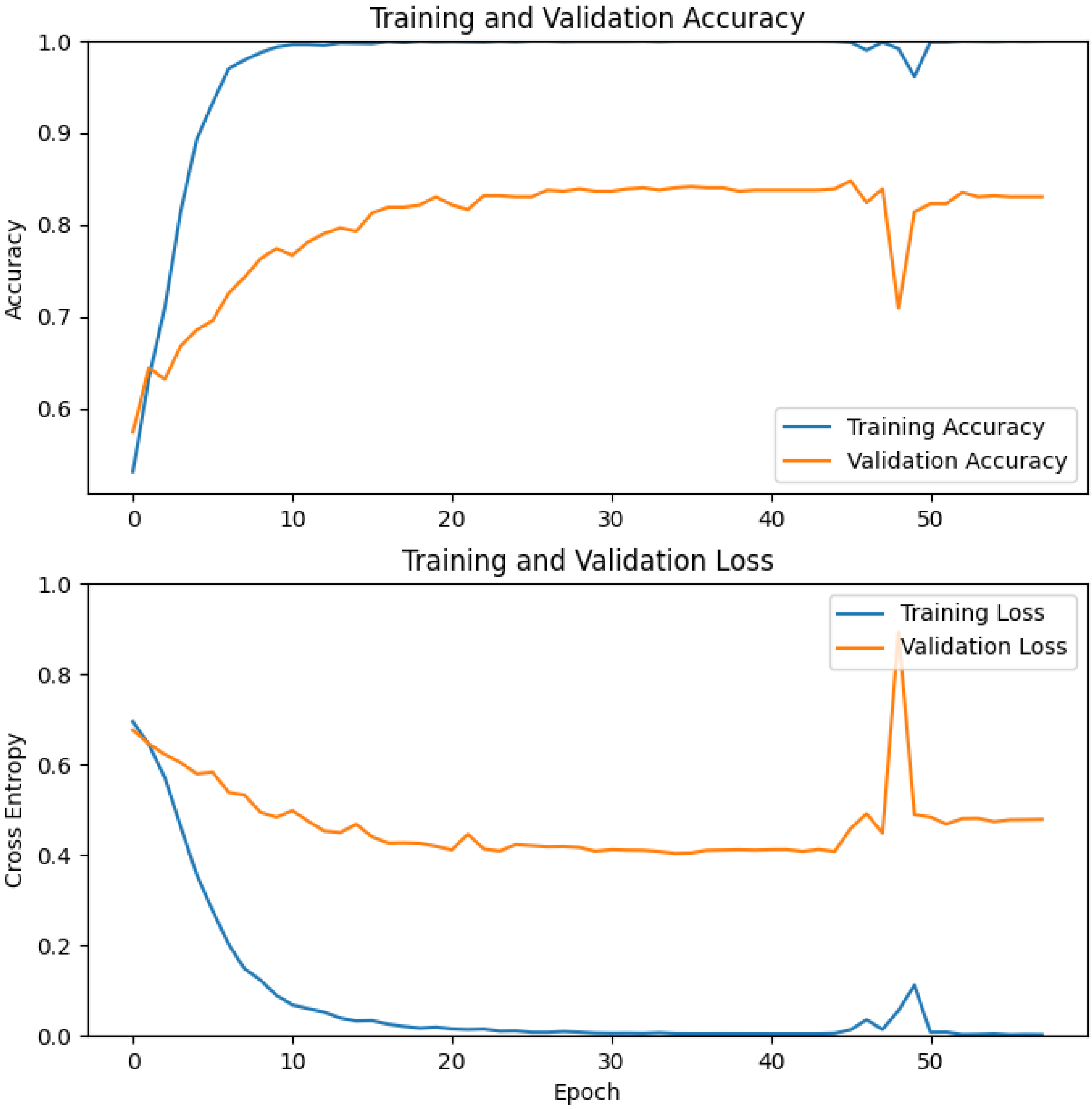


Figure 4.7: Validation Accuracy for Preprocessing Dataset Without Data Augmentation, LR of 0.00001 and Image Size: 500 × 500 pixels on DGX Machine with 100 Fine-Tuning Epochs Allowed

however this indicates that the additional epochs were not necessary since the model does not use the additional learning time to improve in accuracy. This accuracy was slightly lower than the best baseline accuracy of 83.8% indicating that the pre-processing investigation did not offer any improvement in performance. The trials using the larger image size on the machine with more GPU cores consistently outperformed the trials with the smaller image size on the slower machine. Data augmentation did not improve the validation accuracy results of any trials. The confusion matrix of the best-performing model shows the class distribution in Figure 4.8.

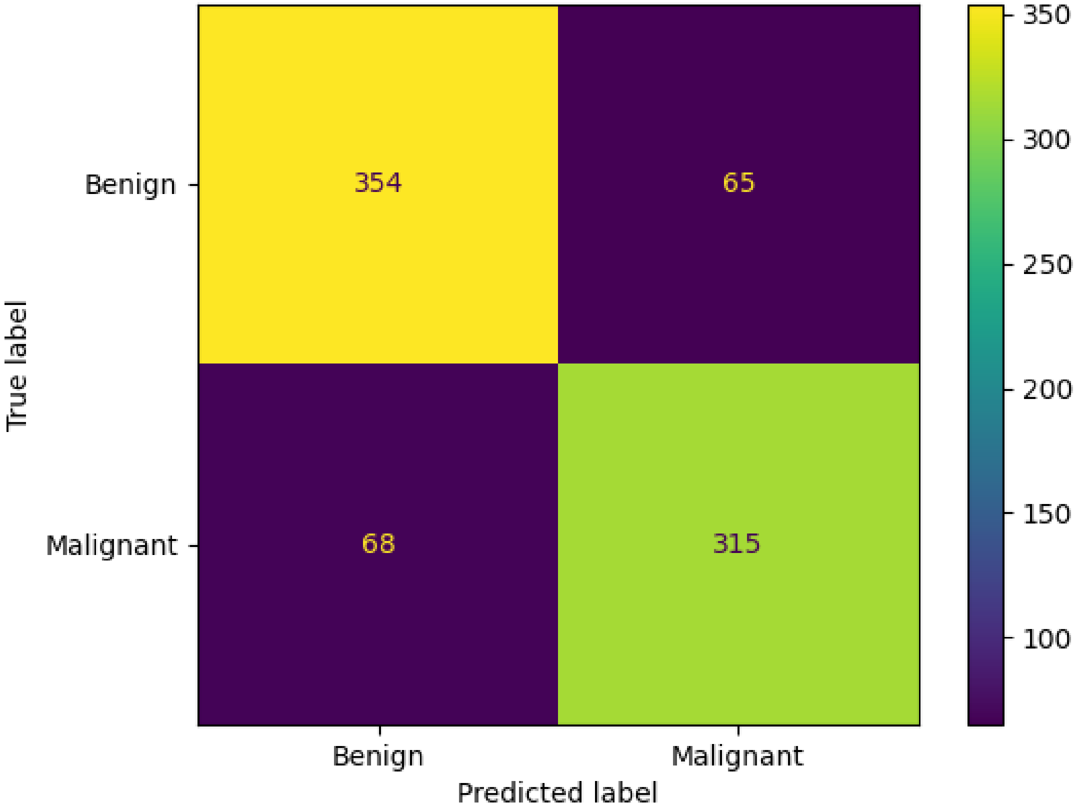


Figure 4.8: Confusion Matrix for Preprocessing Dataset Without Data Augmentation, LR of 0.00001 and Image Size: 500 × 500 pixels on DGX Machine. Right Axis: number of predicted samples (in validation dataset)

4.3 Network Architectures

Model Architecture	Justification
MobileNetV3	Used in Jaamour's Investigation
DenseNet121	Used in Jaamour's Investigation
InceptionV3	Referenced in Literature Review used in [36], [40]
ResNet50	Referenced in Literature Review [15]
VGG-19	Referenced in Literature Review [15]

Table 4.6: Justification of Choice of Model Architectures for Investigation. Jaamour's code is openly available on Github [8]

The base models used in transfer learning were altered to investigate the impact on accuracy in the best-performing models so far throughout the research in this paper. The transfer learning models investigated were: MobileNetV3, DenseNet121, InceptionV3, ResNet50 and VGG-19 and the motivation for this choice is given in Table 4.6.

The first phase of the investigation compared the different models with an original learning rate of 0.0001 on pre-processed data with both image sizes (500 × 500 and

160 × 160 pixels). For these experiments, data augmentation was not used since it gave lower validation accuracy results in the previous investigation. The results of this comparison can be seen in Table 4.7.

In the second phase of the investigation, the 500 × 500 pixel image size was fixed (since this gave overall higher results than 160 × 160 pixel image size). A comparison of the two best learning rates for pre-processing: 0.0001 and 0.00001 are given alongside trials with the baseline model with a learning rate of 0.0001 in Table 4.8.

Model Architecture	160 × 160 Pixels	500 × 500 Pixels
MobileNetV3	73.9	**83.3**
DenseNet121	73.1	81.9
InceptionV3	67.3	81.0
ResNet50	**81.9**	80.8
VGG-19	68.5	71.9

Table 4.7: Validation Accuracies(%) for Preprocessing with Different Models and Learning Rate of 0.0001 (No Data Augmentation)

Model Architecture	Baseline, LR 0.0001	Preprocessing, LR 0.0001	Preprocessing, LR 0.00001
MobileNetV3	83.8	83.3	83.5
DenseNet121	82.7	81.9	**83.8**
InceptionV3	**85.2**	81.0	82.2
ResNet50	82.5	80.8	81.8
VGG-19	69.6	71.9	68.0

Table 4.8: Validation Accuracies(%) with Different Models for 500 × 500 Pixel Images(No Data Augmentation)

From these results we can see that the model with the highest accuracy was the baseline with InceptionV3, the learning rate of 0.0001 and image size 500 × 500 pixels. This model achieved a validation accuracy of 85.2% compared to the best-performing baseline accuracy with MobileNetV3 (original choice): 83.8%. For pre-processed data, the best-performing model had a validation accuracy of 83.8% and used DenseNet with a learning rate of 0.00001 and image size 500 × 500 pixels.

This confirms the findings of the previous investigation that pre-processing has not improved the validation results of the initial chosen baseline model.

4.4 Hyperparameter-Tuning

Different hyperparameter-tuning techniques were considered in the Literature Review 2. In this investigation - hyperparameter tuning was applied to search for the best model hyperparameters for the best-performing models found in the previous investigations.

The hyperparameter tuning was implemented using `keras-tuners` [51] with the Hyperband tuner (an optimised version of Random Search) [52]. Random Search was a common choice in the papers summarised in the Fine-Tuning section of the literature review 2.2 because it is simple to implement and reduces search time compared to grid search since it doesn't check every single combination of hyperparameters in the given ranges. Random search is still suboptimal when it chooses obviously poor value combinations so Hyperband is an optimisation applied to Random Search to check the whole hyperparameter space with a small number of epochs first and only train for longer on parameters with higher validation accuracy. This method is an iterative process that can still take a long time depending on how many Hyperband iterations are specified. Although Hyperband doesn't always get to the most optimal results like Bayesian Optimisation - because it doesn't learn from the history of choices made - the Hyperband tuner used by Keras has been optimised to improve performance. Bayesian Optimisation can take a long time and often needs to be stopped early due to a lack of time or resources.

The hyperparameter ranges investigated here are specified in Table 4.9. These hyperparameters were selected because they hadn't been investigated individually in previous investigations (with exception of Learning Rate - included as it was thought to be a particularly influential parameter). The ranges were selected as sensible from the Literature Review, Chapter 2 and Jaamour's investigations [4]. Some papers in the Literature Review also altered the loss function during hyperparameter optimisation. This was not included here since this model is a binary classifier and the best loss function is Binary Cross Entropy - the other papers compared Mean-Squared Error with Categorical Cross Entropy loss for multi-class classification.

For this investigation, the fine-tuning learning rate was decreased to $1e-6$ from $1e-5$ since the hyperparameter tuning happened before fine-tuning and graphs were showing a drop in accuracy after this tuning when fine-tuning started. This was tweaked during validation set testing so that the fine-tuning would have less impact on the model optimised during frozen training.

Hyperparameter tuning trials were run on the 3 high-performing models selected from the previous investigations. These were: Preprocessing on 500×500 pixel images using DenseNet and without data augmentation (original validation accuracy of 83.8%), preprocessing on 160×160 pixel images using ResNet50 and no data augmentation (original validation accuracy 81.9%) and the baseline model with InceptionV3 on 500×500 pixels and without data augmentation (original validation accuracy 85.2%). Hyperband was run for one iteration on each of these trials to tune the hyperparameters before fine-tuning for the fully-connected layers. The validation accuracy results are given in Table 4.10. This table gives the accuracy of the model tuned before fine-tuning was applied (to demonstrate the improved accuracy from search) and the final accuracy with fine-tuning. In all three cases, the final accuracy went down after fine-tuning which may indicate that fine-tuning was not useful in this task since it doesn't improve the learning of the model.

From Table 4.10 we can see that Hyperband found an improvement in the accuracy of the model for the second trial (preprocessing for 160×160 pixel images with ResNet50) from 81.9% before tuning to 83.3% after. For the final trial, final validation accuracy also improved after hyperparameter tuning validation accuracy: with an increase from 85.2% to 86.2%. The first trial with preprocessing on 500×500 pixel images using DenseNet search found a worse-performing set of hyperparame-

Hyperparameter	Range	Justification
Number of Hidden Layers	1 - 3	Jaamour's model used 2 fully-connected layers - testing either side of this
Number of Units per Layer	128 - 512, step size: 32	[37] uses 128 and 256 layers in tuning, Jaamour uses 512 so this was the minimum and maximum bounds for the range
Dropout Rate (single layer)	0 - 0.4, step size: 0.1	0.2 dropout was used in Jaamour's model - aim to reduce overfitting by increasing the range
Learning Rate (frozen)	0.001, 0.0001, 0.00001	[36] and [40] varied learning rate in hyperparameter tuning. From previous trials learning rates larger and smaller than this were found to give significantly worse results and were discarded.
Optimizer	Adam, SGD, Adadelta, RMSProp	[36] and [40] both trialed the latter three options and Adam optimizer was used in our baseline model.

Table 4.9: Hyperparameters Tuned in Fully-Connected Layers with Ranges Selected and Justification of Choices. For reference, Jaamour's code can be found on Github [8]

ters than with the initial investigation.

It was unexpected that the fine-tuning would bring down the final accuracy of the models - it could be excluded in future experiments or incorporated in the evaluation of each trial in search to optimise the final accuracy. Some papers discussed in the literature review also altered the number of layers unfrozen for fine-tuning [36] [40] - so this hyperparameter could also be tuned in further investigations.

Another important aspect in selecting a hyperparameter optimisation technique is efficiency. The runtimes have been included for each of the 3 trials using Hyperband optimisation. It was assumed that the main impact on efficiency was the training time of the models and the image size since the search algorithm performed the same number of experiments (254) for each trial. The final tuned accuracy increased by 1% for the third trial compared to before hyperparameter tuning - taking over 38 hours to find this improvement. This is a lot of time for a small improvement because the baseline model takes longer to load the dataset and train than the models using preprocessed data. This is because the images are larger (none have been cropped like DDSM). Both of the trials using 500×500 pixel images took significantly longer

Trial	Tuned Validation Accuracy (Before Fine-Tuning)	Final Validation Accuracy	Run Time	Machine
Preprocessing, DenseNet, 500 × 500 pixels, No Augmentation	84.3%	83.2%	14hr 15m 01s	DGX GPU, 8 cores
Preprocessing, ResNet50, 160 × 160 pixels, No Augmentation	83.8%	82.3%	7hr 13m 51s	Single-core GPU
Baseline, InceptionV3, 500 × 500 pixels, No Augmentation	**87.4%**	**86.2%**	38hr 9m 42s	DGX GPU, 8 cores

Table 4.10: Validation results from trials using Hyperband hyperparameter tuning on best models so far. Fine-tuning learning rate: 0.000001

than the trial with 160 × 160 pixel images even though these trials ran on a faster machine. For the small improvements seen here - the efficiency of Hyperband search may not make it the best choice of optimisation.

Table 4.11 outlines the chosen hyperparameters for the fully-connected layers in the 3 trials. In all three trials, the learning rate was selected as 0.00001 and the optimiser was chosen as Adam. Adam was used as the default optimiser in the previous investigations but the default learning rate was 0.0001. None of the trials were found to use more than one dropout layer and the dropout rate was 0.2 in the two trials that included a dropout layer. This is similar to the baseline model we started with - where a single dropout layer with a dropout rate of 0.2 was included before the hidden layers. Two of the trials chose to use 2 hidden layers and the final trial used 1 hidden layer. The baseline model used 2 hidden layers with 512 units in the first layer and 32 in the second. the layers chosen by Hyperband had fewer units in the first layer than the default and more units in the second layer.

Trial	Hidden Layers	Dropout Layers	Learning Rate	Optimiser
Preprocessing, DenseNet, 500 × 500 pixels, No Augmentation	2 layers, Layer1: 352 units, Layer2: 224 units	no dropout	0.00001	Adam
Preprocessing, ResNet50, 160 × 160 pixels, No Augmentation	2 layers, Layer1: 384 units, Layer2: 480 units	dropout layer after second hidden layer, dropout rate: 0.2	0.00001	Adam
Baseline, InceptionV3, 500 × 500 pixels, No Augmentation	1 layer, Layer1: 480 units	dropout layer with dropout rate: 0.2	0.00001	Adam

Table 4.11: Validation results from trials using Hyperband hyperparameter tuning on best models so far.

4.5 Best Validation Results

Overall, the best-performing models for each of the investigations in this chapter are given in Table 4.12 along with their validation accuracies.

Investigation	Model	Validation Accuracy
Baseline	500 × 500 pixels, LR 0.0001, 0.2 Dropout Rate, MobileNetV2	83.8%
Preprocessing	500 × 500 pixels, LR 0.00001, No Augmentation, MobileNetV2	83.5%
Network Architectures	Baseline (no preprocessing), 500 × 500 pixels, LR 0.0001, InceptionV3	85.2%
Hyperparameter Tuning	Baseline (no preprocessing), 500 × 500 pixels, LR 0.00001, InceptionV3, one hidden layer after base model with 480 units	**86.2%**

Table 4.12: Best models selected from investigations using validation dataset. All models use default hyperparameter values from Table 3.1 unless otherwise specified. All models selected based on validation set results not test set.

Chapter 5

Test Set Results

The following results were gathered on the unseen test dataset after all the validation set investigations described in Chapter 4 were completed. This test set combined both datasets and was separated from the training and validation sets before running the program. The test set was constructed from CMMD and DDSM stratifying by patient to avoid data leakage and with a fixed random seed so that the same set could be generated again. The data splitting was contained in a separate program from the model training such that the split happened once and the images were stored in train and test directories.

This chapter is structured in the same way as Chapter 4, with sections for each investigation carried out. The best models were selected from each investigation using the validation set and have been outlined in the final section of Chapter 4. Following the selection of these best models - each was tested on the test set to evaluate their success on unseen data. These test results are presented in this chapter to determine whether the proposed best models could generalise to unseen data.

The class balance of the test set is weighted more towards Benign samples than Malignant with 565 and 444 images respectively.

5.1 Baseline Results

5.1.1 Model Trained on DDSM Only

As specified in the objectives: the investigations were performed on a combined dataset of samples from DDSM and CMMD for both training and testing. However, the baseline model was originally selected to recreate Jaamour's model for the DDSM dataset and build on his research [4]. See Section 3.1 for justification of this choice of research to recreate. To get an insight into the success of matching the baseline model to Jaamour's design - the model was trained and tested on the DDSM dataset alone. The validation results for this model were given in Section 4.1 - the validation accuracy was **99.4%** and very little overfitting was observed in Figure 4.2.

From Table 5.2 the model using DDSM alone for training and testing achieves a higher validation and testing accuracy (99.4% and 67.2% respectively) than the

	Precision	**Recall**	**F1-Score**	**No. Samples**
Benign (class 0)	73%	71%	72%	363
Malignant (class 1)	59%	61%	60%	246
Macro Avg	66%	66%	66%	609
Weighted Avg	67%	67%	67%	609

Table 5.1: Baseline DDSM Dataset with LR 0.0001 and image size 500 × 500 pixels - Test Set Results Class Balance Summary

Metric	**Validation Results**	**Test Results**
Accuracy	99.4%	**67.2**
Final Loss	0.0151	1.34
AUC	99.4%	66.2%
Testing Runtime	-	24.0s

Table 5.2: Baseline DDSM Dataset with LR 0.0001 and image size 500 × 500 pixels - Test Set Results Summary (Rounding to 3 significant figures)

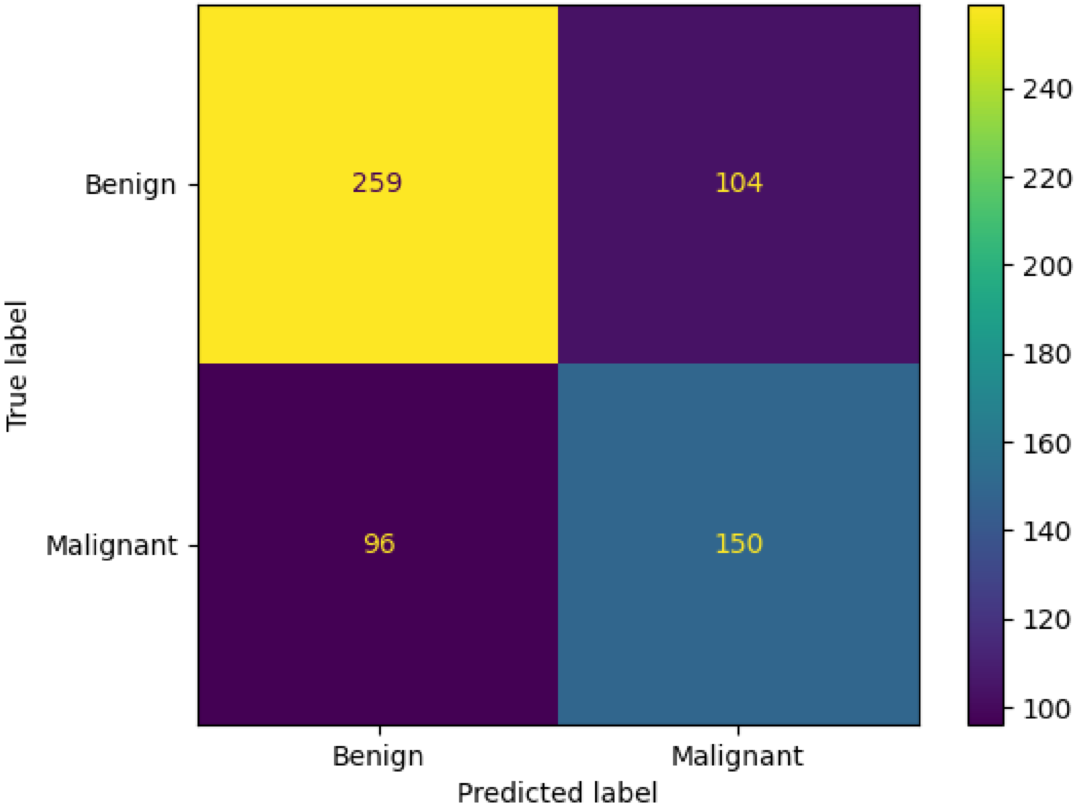


Figure 5.1: Baseline DDSM Dataset with LR 0.0001 and image size 500 × 500 pixels - Confusion Matrix. Right Axis: number of predicted samples (in test dataset)

baseline combined model (results in Table 5.4). This model exhibits overfitting from the validation to testing phase (also seen in all other investigations) although there was very little overfitting seen in the train to validate stage (in Chapter 4). On Jaamour's best-performing model, used to design the baseline model for this research, with the DDSM dataset the testing accuracy was 67.08% [4]. The results achieved with DDSM alone for our model demonstrate that the method to recreate Jaamour's best results was successful. Our test accuracy of 67.2% was slightly higher than Jaamour's (67.08%) [4] but the model was kept as similar as possible with only a slight difference in image size (500 × 500 pixels in ours and 512 × 512 pixels in Jaamour's).

5.1.2 Best Baseline Model

The highest performing Baseline model in Investigation 4.1 was designed to match Jaamour's model [4] with original hyperparameters described in Table 4.9 and image size of 500×500 pixels. This model achieved a validation accuracy of 83.8% although it was seen to overfit compared to the training accuracy. The results of testing this model on the combined test dataset are given in Tables 5.3 and 5.4. In this domain - it is particularly important that malignant cases are correctly identified since misclassified malignant samples lead to missed diagnoses of cancer. Looking at the precision score for the benign class and the recall score for the malignant class gives an indication of how successful the model was at classifying malignant cases. In this case, we can see that precision is high for benign cases (71%), which shows our model is better at classifying the benign cases as correctly benign (i.e. not misclassifying many malignant cases). In addition, the recall is high for the malignant case (64%) which indicates that the malignant cases are being classified as correctly malignant, however, recall is higher for the benign class (68%) so the model classifies benign samples more accurately than malignant. This issue might have been improved by balancing the training dataset with class weights or data augmentation since the malignant samples made up a smaller proportion of the dataset (1859 samples were malignant compared with 2069 benign samples in the combined training dataset). This can be seen visually in the confusion matrix in Figure 5.2.

	Precision	**Recall**	**F1-Score**	**No. Samples**
Benign (class 0)	71%	68%	69%	565
Malignant (class 1)	61%	64%	63%	444
Macro Avg	**66%**	**66%**	**66%**	1009
Weighted Avg	67%	66%	66%	1009

Table 5.3: Baseline Combined Dataset with LR 0.0001 and image size 500 × 500 pixels - Test Set Results Class Balance Summary

5.1.3 Further Investigation Splitting Test Set

The large drop in accuracy from 83.8% to 66.2% for the baseline model was unexpected overfitting (see Table 5.4. The assumption made was that the combination of CMMD and DDSM images in the dataset meant that the model may not generalise

Metric	**Validation Results**	**Test Results**
Accuracy	83.8%	**66.2%**
Final Loss	0.389	1.07
AUC	83.5%	66.2%
Testing Runtime	-	23.9s

Table 5.4: Baseline Combined Dataset with LR 0.0001 and image size 500 × 500 pixels - Test Set Results Summary (Rounding to 3 significant figures)

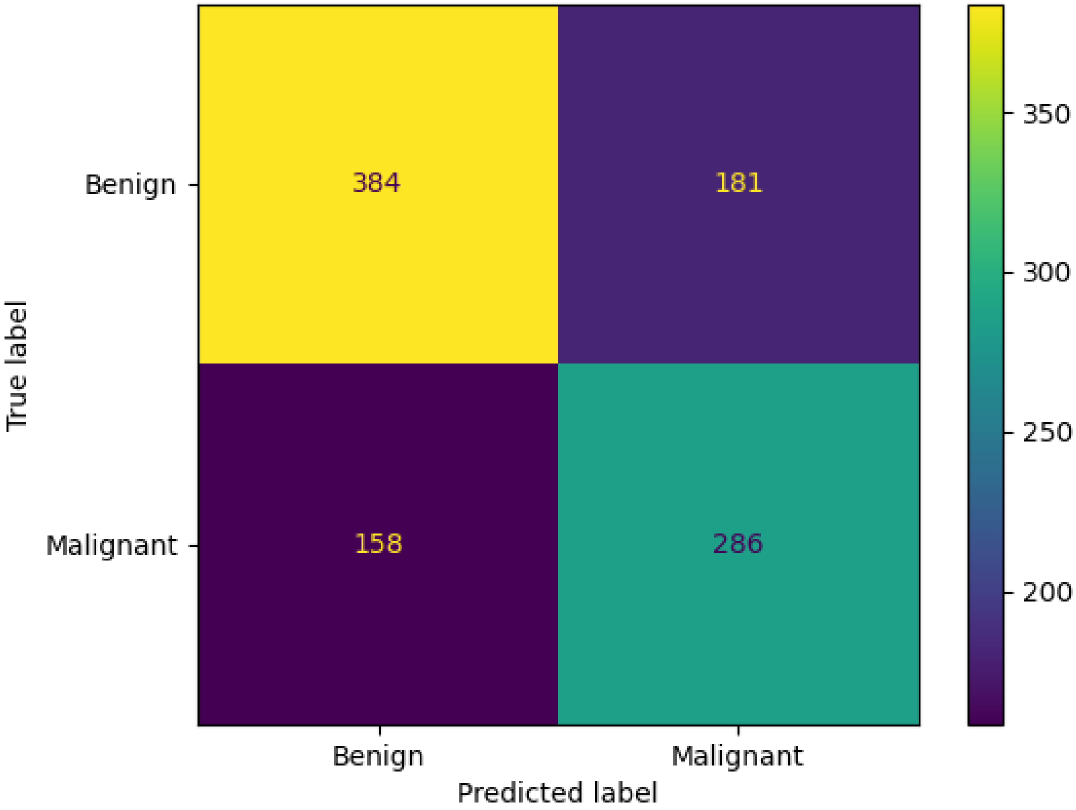


Figure 5.2: Baseline Combined Dataset with LR 0.0001 and image size 500 × 500 pixels - Confusion Matrix. Right Axis: number of predicted samples (in test dataset)

well to both types of images. To investigate this assumption and aim to find the cause of overfitting from the validation to the test set - the model trained with the combined dataset was tested on the DDSM and CMMD parts of the test set separately. Since DDSM images make up a larger portion of the combined dataset (2438 out of 4250 training and validation images i.e 57.4%), it was hypothesised that the model might be more fine-tuned to DDSM images than CMMD and therefore achieve higher accuracy on DDSM test images alone than CMMD images.

The results of this extended investigation shown in Table 5.5 confirm that this is the case - the model performs with higher test accuracy for the DDSM subset than the CMMD subset (but not significantly more). This may be a contributing factor towards the overfitting seen from validation to testing and indicates that the visual features in the images from each dataset may be too different to combine into a single training set without augmentation or further preprocessing. Since there is a lack of new large mammography datasets using modern digital mammography techniques it is challenging to construct a large dataset for training. This makes it difficult to identify whether the important visual differences in the images in DDSM and CMMD are due to the imaging modality or the different geographical locations of patients (America and China).

From confusion matrices for each subset in Figures 5.3 and 5.4 both these subsets are correctly classifying samples in both classes (rather than classifying all samples as one class or classifying randomly).

Metric	**CBIS-DDSM Results**	**CMMD Results**
Accuracy	**70.8%**	**59.8%**
Final Loss	1.24	0.996
AUC	69.7%	59.8%
Macro Avg Precision	70%	60%
Macro Avg Recall	70%	60%
Number of Samples	609	400
Testing Runtime	26.1s	17.4s

Table 5.5: Baseline Combined Dataset with LR 0.0001 and image size 500×500 pixels - Test Subsets: CBIS-DDSM and CMMD Test Set Results Separated (Rounding to 3 significant figures)

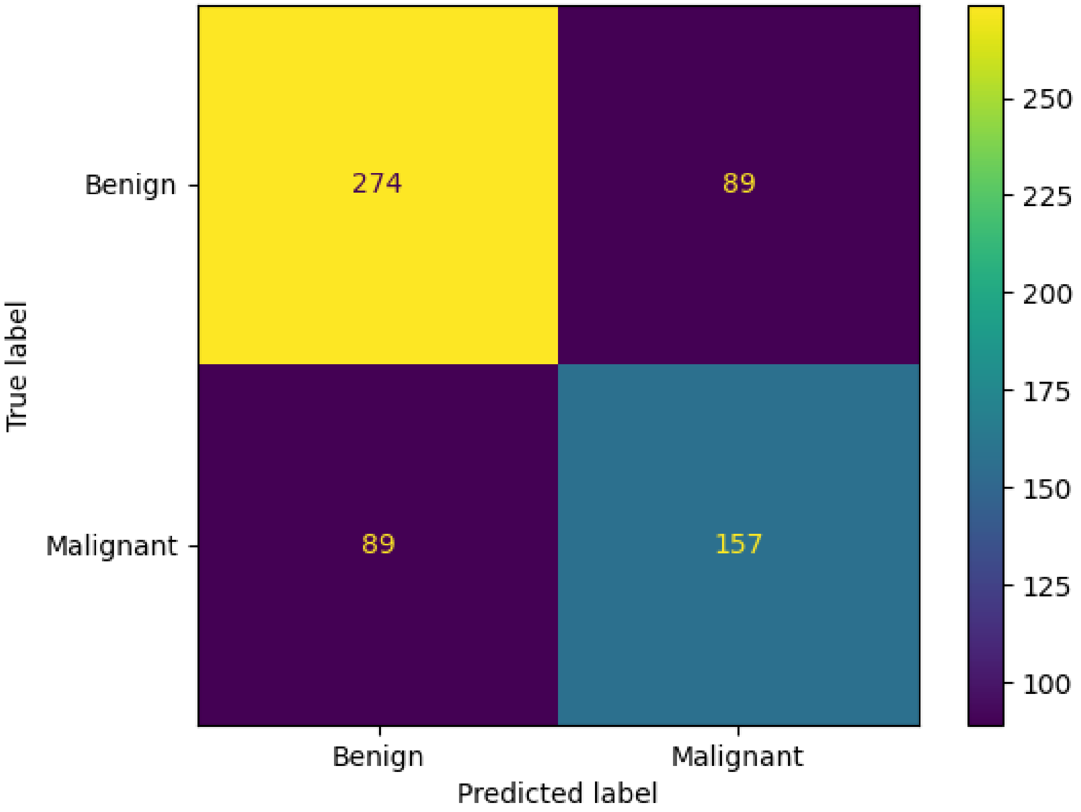


Figure 5.3: Baseline Combined Dataset with LR 0.0001 and image size 500 × 500 pixels on CBIS-DDSM Test Subset - Confusion Matrix. Right Axis: number of predicted samples (in test dataset)

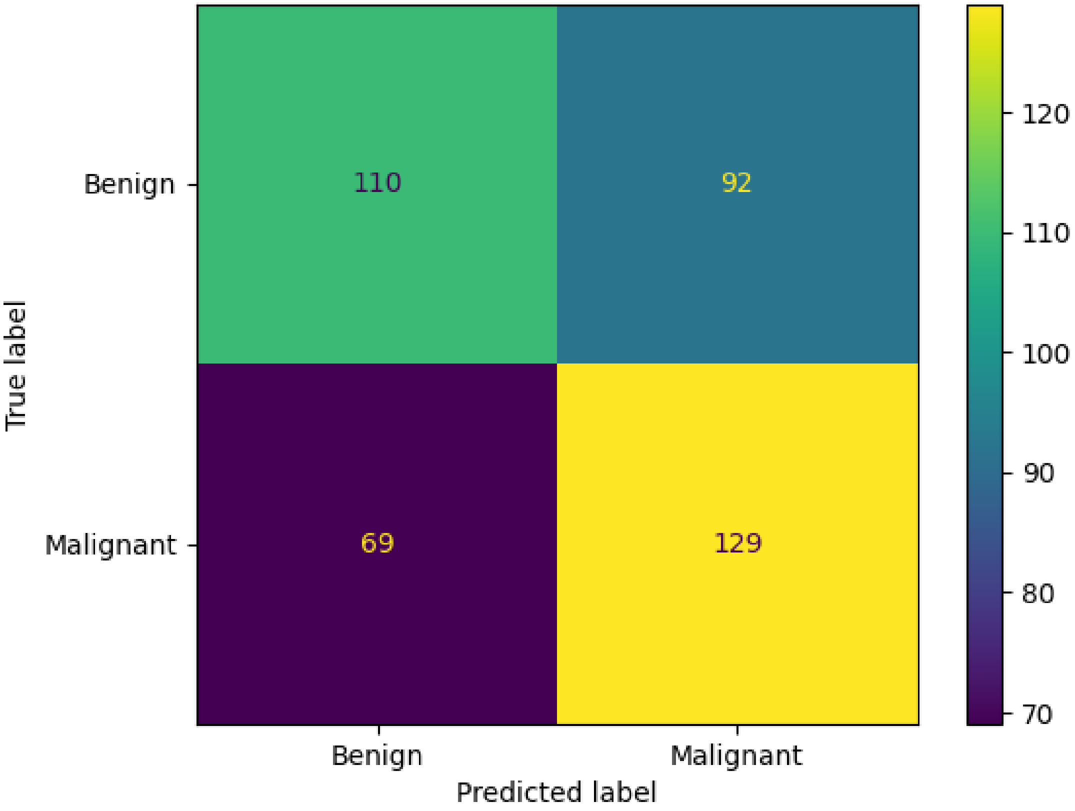


Figure 5.4: Baseline Combined Dataset with LR 0.0001 and image size 500 × 500 pixels on CMMD Test Subset - Confusion Matrix. Right Axis: number of predicted samples (in test dataset)

5.2 Preprocessing Investigation

The results of the preprocessing investigation are given in Tables 5.6 and 5.7. This model performed more poorly in terms of test accuracy (59.7%) than the baseline model (66.2%). This was expected since this model achieved the lowest validation accuracies of all the investigations (83.5%). The recall for malignant cases was low for this model (57%) indicating that the model has a higher risk of missing cancer diagnoses. The confusion matrix in Figure 5.5 shows this visually. Looking at the leading diagonal in this figure demonstrates that the preprocessing model is still classifying more samples correctly in both classes than not - rather than classifying randomly.

	Precision	**Recall**	**F1-Score**	**No. Samples**
Benign (class 0)	65%	62%	63%	565
Malignant (class 1)	54%	57%	55%	444
Macro Avg	**59%**	**59%**	**59%**	1009
Weighted Avg	60%	60%	60%	1009

Table 5.6: Preprocessed combined dataset (without data augmentation) with LR 0.00001 and image size 500 × 500 pixels - Test Set Results Class Balance Summary

Metric	**Validation Results**	**Test Results**
Accuracy	83.5%	**59.7%**
Final Loss	0.443	1.12
AUC	83.4%	59.4%
Testing Runtime	-	10.7s

Table 5.7: Preprocessed combined dataset (without data augmentation), MobileNetV3 base model with LR 0.00001 and image size 500 × 500 pixels - Test Set Results Summary (Rounding to 3 significant figures)

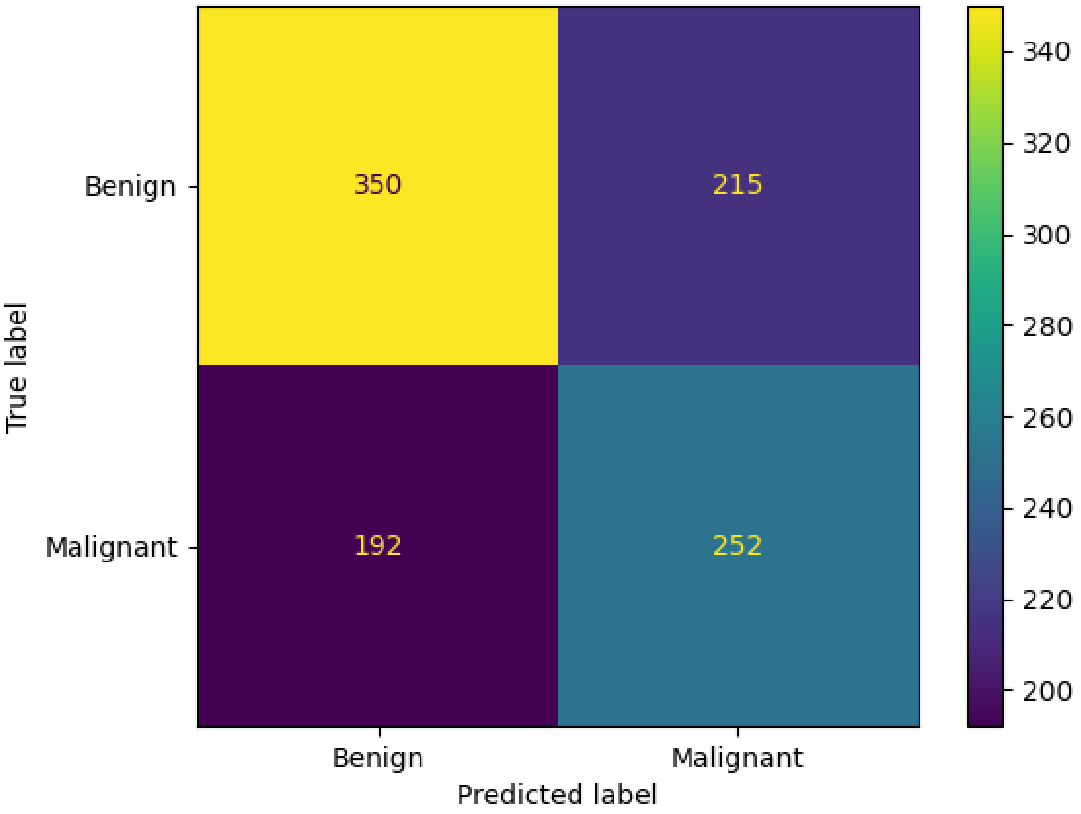


Figure 5.5: Preprocessed combined dataset (without data augmentation), MobileNetV3 base model with LR 0.00001 and image size 500 × 500 pixels - Confusion Matrix. Right Axis: number of predicted samples (in test dataset)

5.3 Network Architecture Investigation

The model selected to use the baseline dataset, InceptionV3 base model and image size 500 × 500 pixels was expected to perform better than the best baseline model and preprocessing model based on the validation accuracies. Unexpectedly this model performed worse than the baseline (66.2%) with a test accuracy of 60.6% and only slightly higher than the preprocessing model (59.7% accuracy). The model accuracy showed a drop from 85.2% validation accuracy to 60.6% on the test set (see Tables 5.8 and 5.9). Similarly to the other investigations the model performed with lower recall for malignant than benign - which is a problem for this domain where detecting all cancer cases is very important. The precision of the benign class (65%) is higher than precision of the malignant class (55%) which indicates that benign cases are predicted with higher accuracy and therefore fewer truly malignant cases are misclassified as benign. Figure 5.6 shows the confusion matrix of the number of samples classified as benign and malignant for this model.

	Precision	**Recall**	**F1-Score**	**No. Samples**
Benign (class 0)	65%	63%	64%	565
Malignant (class 1)	55%	57%	56%	444
Macro Avg	**60%**	**60%**	**60%**	1009
Weighted Avg	61%	61%	61%	1009

Table 5.8: Baseline combined dataset with LR 0.0001 and InceptionV3 base model and image size 500 × 500 pixels - Test Set Results Class Balance Summary

Metric	**Validation Results**	**Test Results**
Accuracy	85.2%	**60.6%**
Final Loss	0.430	1.212
AUC	85.1%	60.2%
Testing Runtime	-	24.5s

Table 5.9: Baseline combined dataset with LR 0.0001 and InceptionV3 base model and image size 500×500 pixels - Test Set Results Summary (Rounding to 3 significant figures)

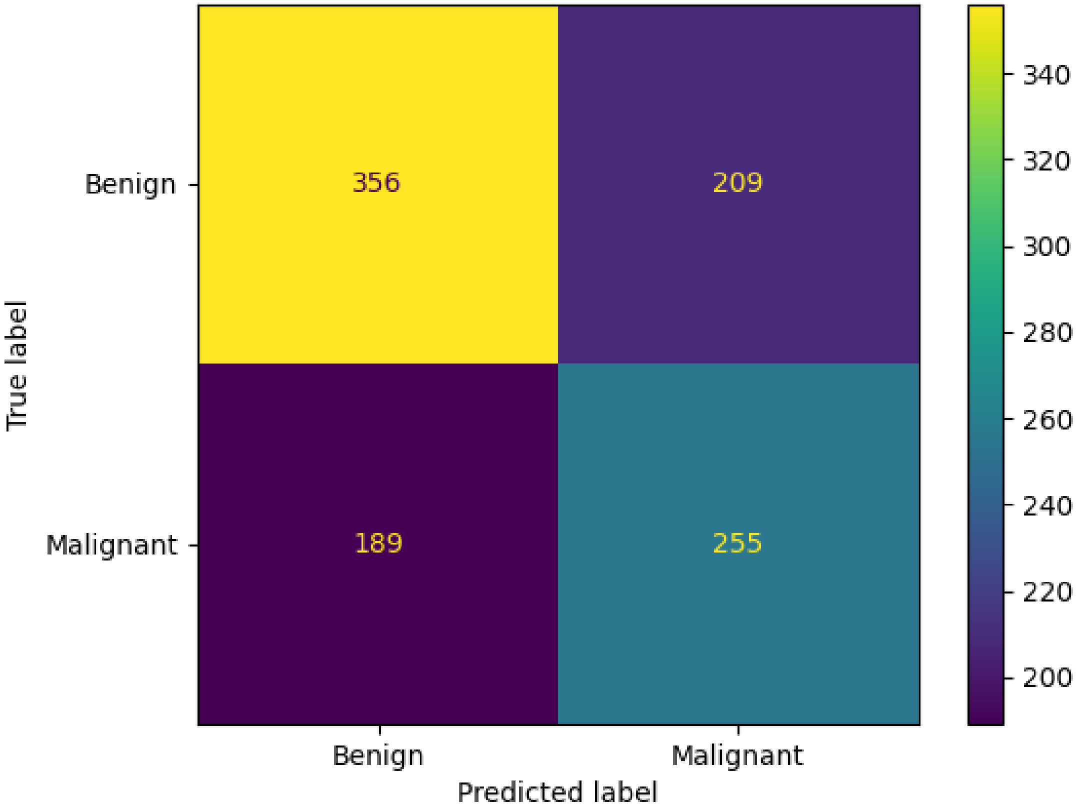


Figure 5.6: Baseline combined dataset with LR 0.0001 and InceptionV3 base model and image size 500×500 pixels - Confusion Matrix. Right Axis: number of predicted samples (in test dataset)

5.4 Hyperparameter-Tuned Model

The hyperparameter-tuned model using the baseline combined dataset and InceptionV3 base model was selected as the best-performing model in the investigations in Chapter 4 with a validation accuracy of 86.2%. However, on the test set this model was not found to perform as well as the baseline model - the hyperparameter-tuned model achieved only 63.1% accuracy compared with 66.2% for the baseline. See Tables 5.10 and 5.11 for a summary of results. Since the improvements were made to the baseline to reach this model - the lower test accuracy indicates that the original model selected based on Jaamour's research [4] may have been the best selection. The methods used to trial different preprocessing techniques and choice of hyperparameters are all valid in trying to improve a model - although here they may not have had a big impact on finding a better-suited model for the problem. The baseline model did achieve the highest test accuracy of 66.2% - however, this initial choice of base model was from Jaamour's investigation using the DDSM dataset alone [4] and may not be the best starting point for the combined dataset and further investigation. In our testing, we could see that for 500 × 500 pixel images - the same baseline model trained and tested on DDSM alone outperformed all other investigations with the combined dataset. Since the DDSM dataset made up a larger proportion of the images in the training and testing sets than CMMD - the baseline model may perform as well as it does because of the DDSM images (while CMMD images are reducing the accuracy of the model).

	Precision	**Recall**	**F1-Score**	**No. Samples**
Benign (class 0)	68%	65%	66%	565
Malignant (class 1)	58%	61%	59%	444
Macro Avg	63%	63%	63%	1009
Weighted Avg	63%	63%	63%	1009

Table 5.10: Hyperparameter-tuned combined dataset (without preprocessing) with InceptionV3 base model and image size 500 × 500 pixels - Test Set Results Class Balance Summary

Metric	**Validation Results**	**Test Results**
Accuracy	86.2%	**63.1%**
Final Loss	0.389%	1.08
AUC	51.5%	62.9%
Testing Runtime	-	19.3%

Table 5.11: Hyperparameter-tuned combined dataset (without preprocessing) with InceptionV3 base model and image size 500 × 500 pixels - Test Set Results Summary (Rounding to 3 significant figures)

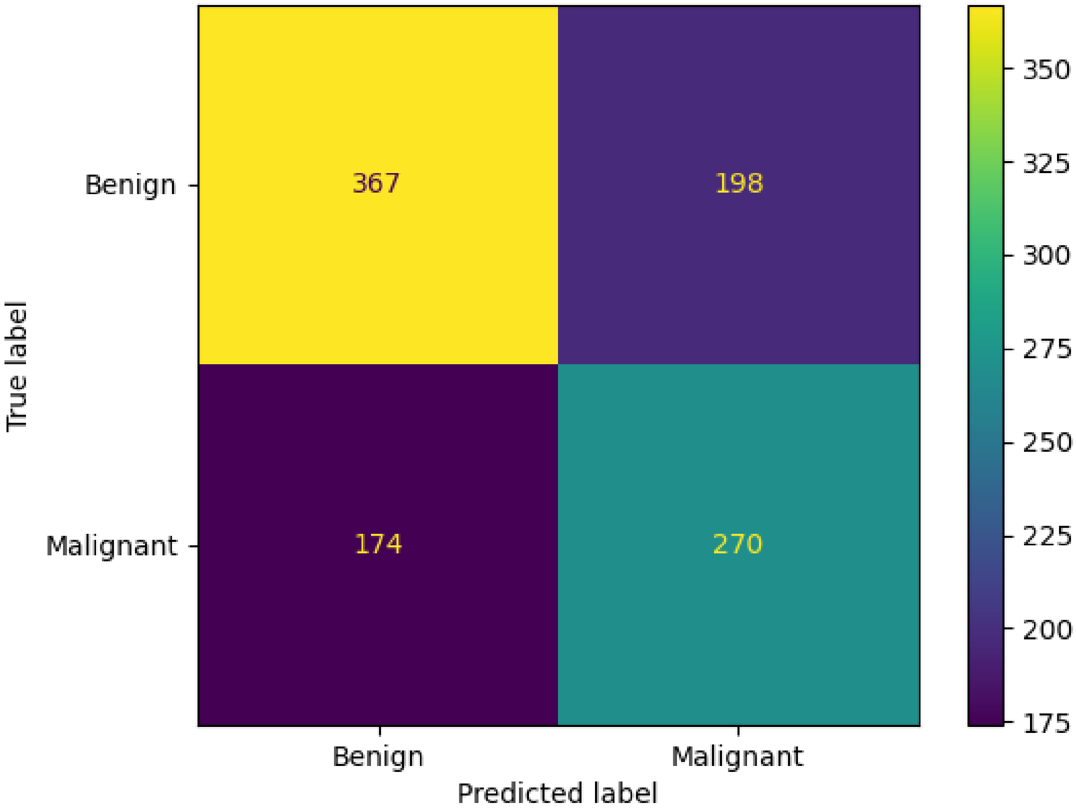


Figure 5.7: Hyperparameter-tuned combined dataset (without preprocessing) with InceptionV3 base model and image size 500 × 500 pixel - Confusion Matrix. Right Axis: number of predicted samples (in test dataset)

Chapter 6

Discussion And Evaluation

6.1 Best Performing Model

The best-performing model identified in the investigations in Chapter 4 was the hyperparameter-tuned model with InceptionV3 base model on the combined dataset without preprocessing and on 500 × 500 pixel images. This model had a validation accuracy of 86.2% but failed to achieve the highest test accuracy results with 63.1% accuracy on the test set. The baseline model achieved the highest test set accuracy for the combined dataset: 66.2% despite a significantly lower validation accuracy of 83.8%. The highest performing model in validation and training (used to demonstrate successful recreation of Jaamour's model [4]) was the baseline model trained and tested on DDSM dataset alone - giving validation accuracy of 99.4% but showing overfitting for the final test accuracy of 67.2%. This demonstrates that the model was well suited for the DDSM dataset and further investigation testing the baseline model, trained with combined data, on the DDSM and CMMD subsets confirmed a higher accuracy on the DDSM subset (70.8%) than the CMMD subset (59.8%). This matches the hypothesis that overfitting seen in the models from all 4 investigations could have been caused by the model being tuned to the DDSM dataset more than CMMD. Since the model was originally designed for the DDSM dataset (from Jaamour's research [4]) and DDSM images made up a larger proportion of the training data this might exacerbate the problem further.

6.2 Critical Appraisal

In this research, Adam Jaamour's best-performing model was used as a baseline for investigating their future work recommendations and further trials [4]. This was selected because Jaamour was a previous student of this project's supervisor and wrote a Masters thesis with a well-structured pipeline that could be recreated as a baseline model for further investigation. Since a Masters thesis is time-constrained it was useful to have a starting point from previous research to build new investigations with support from a shared supervisor. An extension of Jaamour's research (improving on a noted limitation in their work) [4] was to combine the DDSM and CMMD datasets for training and testing to create a dataset with a mixed demographic (mammogram scans taken in America and China) and two medical imaging modalities: film-screen mammography and full-field digital mammography. One key finding during the investigations was that using a larger image size consistently improved the accuracy of classification in all investigations (this was also performed on a faster machine).

The use of transfer learning and CNN models in this research is an approach commonly used in related literature (e.g. [9] [15] [16]). CNNs have been successful in a wide range of medical image analysis tasks in recent years as highlighted in this review of the domain from 2018 [9] which concluded that CNNs have shown a greater ability to accurately classify medical images than other deep learning methods. The main disadvantage of using CNNs for medical imaging is the lack of labelled clinical data publicly available, however common approaches for tackling this problem are transfer learning and data augmentation [9] - which were both trialled in the investigations in this project.

Due to the limitation of time in a Masters thesis, this project did not propose entirely novel techniques - but the techniques and results can be compared with similar work in the literature. In a paper from 2022, considered as an example of using the DDSM and CMMD datasets combined with preprocessing for training, Jun Bai et al. developed a Feature-Fusion Siamese CNN [15]. This research applied complex techniques to create two twin networks (with identical structure and weights) to make a distance comparison of features in old and new images from the same patient; to identify similarities and dissimilarities between the pairs of images. The Siamese network used the ImageNet-trained [5] ResNet model as a backbone to identify features and the paper also gives results for two baseline (non-siamese) transfer learning CNNs using ResNet and VGG base models. The techniques and hyperparameters used in creating the baseline models were very similar to the research carried out in this thesis - whereby ResNet and VGG base models were trialled with two additional hidden fully-connected layers, a dropout rate of 0.2, the Adam optimiser and sigmoid classification function. Differing from our baseline model, this investigation used L1 and L2 regularisation to reduce overfitting in the fully-connected layers, larger image size of 1024×1024 pixels and batch size of 16. The test results obtained for the baseline model in [15] were 82% accuracy for VGG and 86% accuracy for ResNet which significantly outperformed our baseline test accuracy of 66.2%. Although the results achieved here are lower than the cited literature, the techniques used were very similar - giving confidence in the method trialled. With larger image sizes, more data or further attempts to reduce overfitting - the models investigated here are likely to have performed better. Overall [15] found a significantly higher final accuracy of 92% with their novel siamese network approach - however, this relied on obtaining private mammogram data for current and past scans per patient which would not have been possible within the scope of this project.

Another similar approach was seen in a second paper from 2022, where Transfer Learning was applied with AlexNet and VGG base models [16]. Our research used some similar preprocessing techniques including the application of a Wiener filter for noise reduction, CLAHE method for contrast enhancement and coarse ROI extraction with binary thresholding to identify the breast area (although this was multiplied with the original image to remove background noise in [16]). In addition, this model applied data augmentations including horizontal flips, zooming and pre-trained augmentations from VGG. [16] combined DDSM and CMMD for training, validation and testing (as used here) although they took a balanced number of samples from each to avoid giving the model an advantage on one dataset. The final accuracies achieved were extremely high: 99.5% test accuracy for AlexNet and 100% test accuracy for VGG-16 [16]. These accuracies significantly outperformed our results - possibly due to a more successful application of data augmentation (not specified in detail) or the size of images used in training.

Another paper [19] also split both DDSM and CMMD into training, validation and testing datasets for an augmentation-based self-supervised learning (SSL) model. However, the datasets were not combined (as in this research), the training was performed on DDSM dataset and later the pre-trained DDSM model was also transferred onto the model trained with CMMD. This model made use of a large number of augmentations such as random cropping (more suited to patch sets), brightness and contrast changes, gamma shift, histogram equalisation and other image filters to add data in support of self-supervised learning [19]. This investigation took a very different approach to the research presented here: taking a patch set of each mammogram for self-supervised training and then extending the trained knowledge to the whole, labelled images. The best-performing model achieved 73.4% AUC (for DDSM) [19] - compared to our best model's AUC result of 66.2%.

The first investigation in section 4.2 applied a pre-processing pipeline (with coarse ROI cropping, noise reduction, contrast enhancement and data augmentation) to the combined dataset to investigate the effect on model accuracy. One key result of this investigation was that data augmentation had a negative effect on validation accuracy - indicating that the augmentations applied introduced too much variation from the general population of images. Secondly, the pre-processing pipeline (without data augmentation) failed to improve on the accuracy of the baseline model - despite the hypothesis that removing unwanted artefacts in the images and enhancing the visual features would make classification more accurate. The preprocessing techniques applied in Chapter 4 were taken from examples in the literature where they were successful. Lbachir et al. have produced several papers investigating these techniques - demonstrating the improvement from applying a preprocessing step to a model [53] and identifying the best techniques (specifically for segmentation of mammograms). These techniques include noise removal with mean and median filters, contrast enhancement (CLAHE was selected as the best method), label suppression in DDSM and pectoral muscle removal [25]. The label suppression technique was slightly different to the research presented in this paper - our images were cropped to the breast area and in [25] the breast area was selected with binary thresholding and morphological operations were applied to images to remove labels and background. For noise reduction, our research applied a Wiener filter as in [16] instead of median and mean filters.

Data augmentations commonly used in literature included: horizontal flips [17] [33] and rotational augmentations (of 90° intervals) [15] [12] [24] [14]. Unlike Zeiser and Costa et al. who achieved 85.96% accuracy for a 5-depth network with augmentation in comparison to 70.26% accuracy for the same 5-depth model without augmentation [33], our results did not show any improvement with data augmentation applied. However, no rotational augmentations or vertical flips were applied in [33], only horizontal flips, zooming and small patch ROI cropping. The augmentations applied in this research were likely to have created too much variation in the dataset -

however, we see that this method has been applied successfully in other papers. It was noted that none of the papers cited used vertical flip augmentations (as used in our research) which may indicate that, since the vertical orientation of mammograms is always the same, it was not found to be a useful augmentation in this domain.

The final investigation aimed to improve the validation accuracy of the best-performing models from the preceding investigations by applying Hyperband hyperparameter tuning. This led to the selection of the best-performing model from the investigations with a validation accuracy of 86.2%. This best-performing model did not achieve the highest test accuracy in the final results section with 63.1%. This model was the baseline model using the InceptionV3 base model, tuned to have one fully connected layer with 480 hidden units, one dropout layer with a dropout rate of 0.2, a learning rate of 0.00001 and Adam optimiser.

Investigation of hyperparameter tuning techniques led to related work in mammography classification. Grid search is common but was not selected here because it can take a lot of time to search through every combination of hyperparameters. [36] and [37] both used grid search for mammogram classification - the first compared 72 different models (with an approximate training time of 12 hours for our baseline model - this search would have taken 36 days to complete). Soriano et al. take a different approach and apply random search before grid search to find an initial model and reduce the parameter search space for tuning the hyperparameters further [40]. Our method used an optimised version of random search: Hyperband from Keras-Tuners [51] which showed an improvement in accuracy in all but one of the trials. Grid search was not applied after random search in this research due to time limitations. [38] built a Tree-Based Pipeline Optimisation Tool (TPOT) using a genetic algorithm for hyperparameter optimisation. This study showed that TPOT consistently outperformed grid search for different classifiers [38]. Genetic Algorithms take a long time to train (many generations and a large population set are needed for effective search) so this research recommends considering the trade-off between the training efficiency and the relatively small performance improvement (several percentage point increase in accuracy) [38]. Bayesian Optimisation is another technique used in the domain - Chakravarthy et al. showed very successful models with accuracy over 98% using Support Vector Machines with Ensemble-Based Transfer Learning and Bayesian Optimisation [39]. This method was not selected due to time constraints but could be considered for future work.

Moya et al. concluded that the most influential hyperparameters in transfer learning for multi-class classification of mammograms were the optimiser and loss function, closely followed by the number of layers frozen in fine-tuning. This research also found that varying learning rates and batch sizes had very little effect on the model performance[36]. In our research binary classification was used so the loss function was not changed from Binary Cross Entropy. We did not include fine-tuning in the hyperparameter search or tune the number of layers unfrozen in this training phase. This parameter was tuned in multiple papers ([36, 40]) and could be included as future work for this investigation.

6.3 Limitations

The main limitation of this investigation was the time constraints for trialling different models before carrying out the experiments with a fixed set of parameters. Some techniques used in other papers identified in the literature review could not be completed due to these time constraints. For example, the removal of the pectoral muscle as a step in the pre-processing pipeline is suggested in Lbachir et al's Preprocessing Method [34]. The image processing techniques required to accurately segment and remove the pectoral muscle from the rest of the image was outwith the scope of the project and could not be completed in the time frame.

Another challenge was a limitation of computing power and time needed to train with larger image sizes. The preprocessed DDSM images were cropped to a smaller size (around 500 × 600 pixels) but the original images were much larger and could have been used in training without being resized as much.

6.4 Future Work

This research aimed to trial different preprocessing and hyperparameter tuning techniques to try and improve on a previously designed baseline model from Jaamour's research [4]. The pipeline created for the preprocessing investigation (including coarse cropping to the region of interest in DDSM images, wiener filter for noise reduction, CLAHE contrast enhancement and then data augmentation) was unsuccessful in producing higher accuracy results than the baseline. Future work could be to try more techniques for data integration for combining the CMMD and DDSM datasets since they have different visual features. This might include applying different filters and noise reduction to make the features more visually similar between the two datasets or resizing and cropping images to include the same region of interest without background. See a comparison of DDSM and CMMD features in Figures 4.5 and 4.6 in Chapter 4 - including the size of the breast area, transparency and noise and visible lesion patterns. In addition, future research should try different augmentations with smaller changes from the original images such as smaller angle rotations. The data augmentation here was unsuccessful and this is assumed to be due to model learning features based on position and rotation which was affected by augmentations varying too much from the general population. Castro et al. [50] offer a novel method of applying rotational augmentations to filters and weights in the CNN instead of to the input image directly for medical imaging. This method could be investigated for full-size DDSM and CMMD images - as it was only trialled on patches in the paper [50].

In the literature review, hyperparameter tuning techniques were compared in mammography classification research, see Table 2.2. In the final investigation in Chapter 4 only one technique was chosen: Hyperband tuning. In the future, it would be interesting to compare different hyperparameter tuning techniques from the literature review in their effectiveness at improving a baseline model and their relative time efficiency. In particular, Bayesian Optimisation produced good results for tuning a model for breast cancer detection in research by Chakravarthy et al. [39]. Hyperband tuning could also be investigated further by fine-tuning the model as part of the search and searching for longer by adding more iterations of the Hyperband algorithm. These techniques could also be used on more hyperparameters such as the number of layers to unfreeze in fine-tuning (see Table 2.2 for more examples from related literature).

6.5 Evaluation

This thesis met all the primary and secondary objectives identified in the Introduction, Chapter 1. See Table 6.1 for a summary of how the objectives were met in this research. The primary objectives included constructing a comprehensive literature review to give context to the related work in this area of research. Informed by this research and specifically the previous work by Adam Jaamour - a baseline CNN model was constructed to recreate Jaamour's best-performing model [4].

This baseline CNN was trained and evaluated on a combined dataset using DDSM and CMMD images. This research is new to the domain comparable with only two similar studies using different architectures: [15], [19]. The investigations carried out using this baseline model as a control were rigorous and well-informed from previous related work. Although the results cannot compete with current published research papers - we discovered some important key findings. The first key finding was that using a larger image size (500 × 500 pixels) consistently outperformed a smaller image size (160 × 160 pixels) in terms of model accuracy. The second discovery was that combining DDSM and CMMD datasets (which contain scans from two different mammogram modalities and two different geographical populations) presented challenges for the model's ability to classify cancer from visual features. Some parts of the investigations were found to be unsuccessful: such as the use of data augmentation and preprocessing. From these results, we discovered that vertical flips may not be a sensible augmentation for mammograms (which are always captured with the same orientation) and neither are 90° rotations on the input images (despite the benefit of not losing corners of the image). Another key finding was that the models were performing better on the DDSM images than the CMMD images - indicating that it could be beneficial to balance the number of images from each dataset (even if this resulted in a smaller overall number of images) or to find a model that was well-suited to both datasets separately before combining them. The final important finding from applying hyperparameter-tuning to the model was that fine-tuning should be excluded completely at this stage (since it worsened results) or included within the hyperparameter search (potentially with tuning for the number of layers to freeze) so the overall model accuracy could be improved. It should be considered that this would take more time in the search process due to longer training - so this may be a trade-off between accuracy and efficiency.

Overall this research yielded some new and interesting results about the use of a combined CMMD and DDSM dataset for transfer learning - which could be used by future researchers in developing new models. This research has contributed to a vital area of medical imagining analysis that could one day help to build a robust cancer detection system for early diagnosis of this disease. The work here is open and reproducible - the code for recreating all of the investigations can be found on GitHub: `https://github.com/RAMcCracken/CS5199_Breast_Cancer_Detection_Project`

Objective	**Work Completed**
P1: Literature Review	Literature review in Chapter 2 covers the domain background, network architecture, datasets of interest, preprocessing techniques and methods for hyperparameter tuning
P2: Basic CNN	Informed by Literature Review and previous work by Adam Jaamour [4] - a baseline CNN model was constructed to recreate Jaamour's best performing model
P3: Investigate Model Architectures	In Section 4.3: 5 different base models were investigated for transfer learning: MobileNetV3, DenseNet121, InceptionV3, ResNet50, VGG-19. Notably; Jaamour's baseline model [4] was improved with the InceptionV3 base model instead of MobileNetV2.
P4: Investigate Pre-Processing Techniques	In Section 4.2: a preprocessing pipeline was constructed (with common techniques from related work) and trialled with and without selected data augmentations
S1: Apply Hyperparameter-Tuning Techniques	In Section 4.4: Hyperband tuning was used on the best-performing models from the previous trials. Hyperparameters were selected from commonly tuned parameters in Table 2.2
S2: Report Results of Best Model	All the results from the investigations were evaluated and reported for the validation set in Chapter 4. The best-performing models were identified and evaluated on the unseen test set in Chapter 5

Table 6.1: Evaluation of Meeting Original Objectives. Note: 'P' is a Primary objective and 'S' is a Secondary objective

Chapter 7

Conclusion

The aim of this thesis was to build a neural network to classify mammograms as benign or malignant for breast cancer detection. In future, these techniques could be used to assist radiologists in making diagnoses that increase the rate of early detection of breast cancer for patients and enable vital early treatment.

A convolutional neural network was constructed using transfer learning (with a base model trained on Image Net [5] weights) and several investigations were carried out to trial different deep learning techniques. First, a baseline model was constructed to replicate the best-performing model from Adam Jaamour's MSci thesis (a previous student of this project's supervisor) [4]. This baseline model (and all subsequent investigations) were applied to a combined dataset of images from CBIS-DDSM and CMMD datasets. The first investigation applied an image preprocessing pipeline with coarse cropping to the breast area in DDSM images, noise reduction and contrast enhancement filters. The trials were also applied with and without data augmentation and for different learning rates and image sizes. One of the key findings in this investigation was that increasing the image size from 160 × 160 pixels to 500 × 500 pixels (and running on a faster machine) consistently improved the accuracy of classification in all trials. Another key finding was that image preprocessing did not improve the final accuracy results and neither did data augmentation with flips and 90° rotations. Different base models for transfer learning were investigated: MobileNetV3, DenseNet121, InceptionV3, ResNet50 and VGG-19. The best base model found for the baseline model (without preprocessing/data augmentation) was InceptionV3 and the best base model for the model with preprocessing was DenseNet121. The final investigation applied Hyperband hyperparameter-tuning to the model before fine-tuning. The overall best-performing model in the validation result evaluation was found to be the baseline model with the InceptionV3 base model (after hyperparameter tuning). In the final test results (evaluating models on an unseen dataset) the best-performing model achieved the second-highest accuracy of 63.1% and the original baseline achieved the highest test accuracy of 66.2%.

This work has contributed to an important research area discovering the techniques needed to build computer-based diagnosis systems for breast cancer. As the most common cancer affecting women in the UK, thousands of new cases are reported yearly [1]. Screening programs in the NHS and abroad have created a large number of scans to be diagnosed regularly by radiologists and in addition, there is a

European recommendation for double-reading to reduce errors [54]. Artificial intelligence would be an incredible tool to make this repetitive task less error-prone. With enough data and enough research on this problem - CNNs for medical diagnosis could save lives in this field in the future. This thesis contributes a comprehensive literature review and rigorous investigations to this field of research such that future academics can build on the findings in this paper: re-using techniques that were successful and avoiding the techniques that were not. In particular, this paper offers one of very few analyses on the use of a combined dataset from two geographical populations and two different imaging modalities. Future work on this combined dataset could remove bias towards a single population in computer-aided breast cancer diagnosis, saving more patients' lives.

Bibliography

[1] Cancer Research UK. *Breast Cancer Statistics.* URL: https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/breast-cancer#heading-Zero. (accessed: 15.09.22).

[2] The Cancer Imaging Archive. *CBIS-DDSM.* URL: https://wiki.cancerimagingarchive.net/display/Public/CBIS-DDSM#22516629cf2ec23796854d91bc86c4ae2e499baa. (accessed: 15.09.22).

[3] The Cancer Imaging Archive. *The Chinese Mammography Database (CMMD).* URL: https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70230508. (accessed: 15.09.22).

[4] Adam Jaamour, David Harris-Birtill, and Lewis McMillan. "Breast Cancer Detection in Mammograms using Deep Learning Techniques". 2020. URL: https://studres.cs.st-andrews.ac.uk/Library/ProjectLibrary/cs5098/2020/agj6-Final_report.pdf.

[5] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition.* Ieee. 2009, pp. 248–255.

[6] Horsch M. Elter A. "CADx of mammographic masses and clustered microcalcifications: a review". In: *Medical Physics* 36.6 (2009), pp. 2052–2068. DOI: 10.1118/1.3121511.

[7] Diane M Novy et al. "Percutaneous core biopsy of the breast: correlates of anxiety". In: *Academic radiology* 8.6 (2001), pp. 467–472.

[8] Adam Jaamour and Craig Myles. *Breast Cancer Detection in Mammograms Using Deep Learning Techniques.* 2020 [Online]. URL: https://doi.org/10.5281/zenodo.3985051.

[9] A. Qayyum S.M. Anwar M. Majid. "Medical Image Analysis using Convolutional Neural Networks: A Review". In: *J Med Syst* 42 (2018), p. 226. DOI: https://doi.org/10.1007/s10916-018-1088-1.

[10] Aaron Dougherty. *Magic of the Sobel Operator.* URL: https://towardsdatascience.com/magic-of-the-sobel-operator-bbbcb15af20d. (accessed: 28.01.23).

[11] K Santle Camilus, VK Govindan, and PS Sathidevi. "Pectoral muscle identification in mammograms". In: *Journal of applied clinical medical physics* 12.3 (2011), pp. 215–230.

[12] Mohammed A Al-Masni et al. “Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system”. In: *Computer methods and programs in biomedicine* 157 (2018), pp. 85–94.

[13] IBM Cloud Education. *What Are Convolutional Neural Networks.* 2020. URL: `https://www.ibm.com/cloud/learn/convolutional-neural-networks`. (accessed: 29/09/22).

[14] Hongmin Cai et al. “Breast microcalcification diagnosis using deep convolutional neural network from digital mammograms”. In: *Computational and mathematical methods in medicine* 2019 (2019).

[15] Jun Bai et al. “Feature fusion Siamese network for breast cancer detection comparing current and prior mammograms”. In: *Medical Physics* 49.6 (2022), pp. 3654–3669.

[16] Saida Sarra Boudouh and Mustapha Bouakkaz. “Breast Cancer: Breast Tumor Detection Using Deep Transfer Learning Techniques in Mammogram Images”. In: *2022 International Conference on Computer Science and Software Engineering (CSASE)*. IEEE. 2022, pp. 289–294.

[17] Benjamin Stadnick et al. “Meta-repository of screening mammography classifiers”. In: *arXiv preprint arXiv:2108.04800* (2021).

[18] Sean Benhur J. *A friendly introduction to Siamese Networks.* URL: `https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942`. (accessed: 10.10.22).

[19] John D Miller et al. “Self-Supervised Deep Learning to Enhance Breast Cancer Detection on Screening Mammography”. In: *arXiv preprint arXiv:2203.08812* (2022).

[20] Section. *Introduction to YOLO Algorithm for Object Detection.* URL: `https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/#:~:text=YOLO%5C%20is%5C%20an%5C%20algorithm%5C%20that,%5C%2C%5C%20parking%5C%20meters%5C%2C%5C%20and%5C%20animals.`. (accessed: 11.10.22).

[21] Bjorn Schuller Hesam Sagha Nicholas Cummins. “Stacked denoising autoencoders for sentiment analysis: a review”. In: *WIRES Data Mining and Knowledge Discovery* 7.5 (2017). DOI: `https://doi.org/10.1002/widm.1212`.

[22] Chris Nicholson. *Denoising Autoencoders.* URL: `https://wiki.pathmind.com/denoising-autoencoder#:~:text=A%5C%20stacked%5C%20denoising%5C%20autoencoder%5C%20is,as%5C%20input%5C%20is%5C%20fed%5C%20through.`. (accessed: 11.10.22).

[23] Rachel Farber et al. “Impact of full-field digital mammography versus film-screen mammography in population screening: a meta-analysis”. In: *JNCI: Journal of the National Cancer Institute* 113.1 (2021), pp. 16–26.

[24] Jinhua Wang et al. “Discrimination of breast cancer with microcalcifications on mammography by deep learning”. In: *Scientific reports* 6.1 (2016), pp. 1–9.

[25] S. Tallal I. A. Lbachir I. Daoudi. "Automatic Computer-Aided Diagnosis System for Mass Detection and Classification in Mammography". In: *Multimedia Tools and Applications* 80 (2020), pp. 9493–9525. DOI: `https://doi.org/10.1007/s11042-020-09991-3`.

[26] Yuan-Zhi Shao et al. "Characterizing the clustered microcalcifications on mammograms to predict the pathological classification and grading: a mathematical modeling approach". In: *Journal of digital imaging* 24.5 (2011), pp. 764–771.

[27] Pragati Baheti. *A Simple Guide to Data Preprocessing in Machine Learning.* 2022. URL: `https://www.v7labs.com/blog/data-preprocessing-guide`. (accessed: 29/09/22).

[28] G. E. Hinton A. Krizhevsky I. Sutskever. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems* 25 (2012). URL: `https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

[29] Christian Szegedy et al. "Going Deeper with Convolutions". In: *CoRR* abs/1409.4842 (2014). arXiv: `1409.4842`. URL: `http://arxiv.org/abs/1409.4842`.

[30] Geraldo Braz Junior et al. "Breast cancer detection in mammography using spatial diversity, geostatistics, and concave geometry". In: *Multimedia Tools and Applications* 78.10 (2019), pp. 13005–13031.

[31] Aarti Bokade and Ankit Shah. "Breast Mass Classification with Deep Transfer Feature Extractor Model and Random Forest Classifier". In: *2021 International Conference on Recent Trends on Electronics, Information, Communication and Technology (RTEICT).* 2021, pp. 634–641. DOI: `10.1109/RTEICT52294.2021.9573909`.

[32] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). arXiv: `1409.1556`. URL: `https://arxiv.org/abs/1409.1556`.

[33] T. Zonta F.A. Zeiser C.A. da Costa. "Segmentation of Masses on Mammograms Using Data Augmentation and Deep Learning". In: *J Digit Imaging* 33 (2020), pp. 858–868. DOI: `https://doi.org/10.1007/s10278-020-00330-4`.

[34] Ilhame Ait Lbachir et al. "A New Mammogram Preprocessing Method for Computer-Aided Diagnosis Systems". In: *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA).* 2017, pp. 166–171. DOI: `10.1109/AICCSA.2017.40`.

[35] DeepLizard. *Fine-Tuning A Neural Network Explained.* 2017. URL: `https://deeplizard.com/learn/video/5T-iXNNiwIs#:~:text=Fine%5C%2Dtuning%5C%20is%5C%20a%5C%20way,perform%5C%20a%5C%20second%5C%20similar%5C%20task.`. (accessed: 05/10/22).

[36] Edison Moya et al. "Multi-category Classification of Mammograms by Using Convolutional Neural Networks". In: *2017 International Conference on Information Systems and Computer Science (INCISCOS).* 2017, pp. 133–140. DOI: `10.1109/INCISCOS.2017.56`.

[37] D Saranyaraj, M Manikandan, and S Maheswari. "A deep convolutional neural network for the early detection of breast carcinoma with respect to hyperparameter tuning". In: *Multimedia Tools and Applications* 79.15-16 (2020), pp. 11013–11038.

[38] Siti Fairuz Mat Radzi et al. "Hyperparameter Tuning and Pipeline Optimization via Grid Search Method and Tree-Based AutoML in Breast Cancer Prediction". In: *Journal of Personalized Medicine* 11.10 (2021), p. 978.

[39] SR Sannasi Chakravarthy and Harikumar Rajaguru. "Deep-features with Bayesian optimized classifiers for the breast cancer diagnosis". In: *International Journal of Imaging Systems and Technology* 31.4 (2021), pp. 1861–1881.

[40] Danny Soriano et al. "Mammogram classification schemes by using convolutional neural networks". In: *International Conference on Technology Trends.* Springer. 2017, pp. 71–85.

[41] Jason Brownlee. *Hyperparameter Optimization With Random Search and Grid Search.* 2020. URL: `https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/`. (accessed: 05/10/22).

[42] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. "SVM parameter optimization using grid search and genetic algorithm to improve classification performance". In: *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 14.4 (2016), pp. 1502–1509.

[43] Will Koehrsen. *A Conceptual Explanation of Bayesian Hyperparameter Optimisation for Machine Learning.* 2018. URL: `https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f`. (accessed: 05/10/22).

[44] Sourabh Mehta. *How to use genetic algorithm for hyperparameter tuning of ML models?* 2022. URL: `https://analyticsindiamag.com/how-to-use-genetic-algorithm-for-hyperparameter-tuning-of-ml-models/`. (accessed: 06/10/22).

[45] Rebecca Sawyer Lee et al. "A curated mammography data set for use in computer-aided detection and diagnosis research". In: *Scientific data* 4.1 (2017), pp. 1–9.

[46] Keishi Fujiwara, Ichiro Maeda, and Hidefumi Mimura. "Granular cell tumor of the breast mimicking malignancy: a case report with a literature review". In: *Acta radiologica open* 7.12 (2018), p. 2058460118816537.

[47] P. Gupta and N.K. Sehgal. *Introduction to Machine Learning in the Cloud with Python: Concepts and Practices.* Springer International Publishing, 2021, pp. 68–69. ISBN: 9783030712709. URL: `https://books.google.co.uk/books?id=t-grEAAAQBAJ`.

[48] Wang Hao et al. "The role of activation function in cnn". In: *2020 2nd International Conference on Information Technology and Computer Application (ITCA).* IEEE. 2020, pp. 429–432.

[49] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2018, pp. 4510–4520.

[50] Eduardo Castro, Jose Costa Pereira, and Jaime S. Cardoso. “Weight Rotation as a Regularization Strategy in Convolutional Neural Networks”. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019, pp. 2106–2110. DOI: 10.1109/EMBC.2019.8856448.

[51] *Keras Tuner Library*. URL: https://keras.io/keras%5C_tuner/. (accessed: 25.01.23).

[52] Lisha Li et al. “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization”. In: *Journal of Machine Learning Research* 18.185 (2018), pp. 1–52. URL: http://jmlr.org/papers/v18/16-558.html.

[53] Ilhame Ait lbachir et al. “A Survey on Segmentation Techniques of Mammogram Images”. In: *Advances in Ubiquitous Networking 2*. Springer Singapore, 2017, pp. 545–556. ISBN: 978-981-10-1627-1. DOI: 10.1007/978-981-10-1627-1_43.

[54] Kristina Lång et al. “Identifying normal mammograms in a large screening population using artificial intelligence”. In: *European radiology* 31 (2021), pp. 1687–1692.

Appendix A

Usage

The code for this thesis is open and publicly accessible on Github: `https://github.com/RAMcCracken/CS5199_Breast_Cancer_Detection_Project`.

A.1 Setup

On University of St Andrews, School of Computer Science GPUs - the code should be run inside a Docker container built with the latest Tensorflow image from `https://catalog.ngc.nvidia.com/orgs/nvidia/containers/tensorflow/tags`. Instructions for setting up Docker can be found on the school's systems wiki: `https://systems.wiki.cs.st-andrews.ac.uk/index.php/Docker`. The `requirements.txt` file included in the source code on Github can be used to build the Docker container to install all the necessary Python libraries.

The data has not been included in the source code since it is around 300 GB. The CMMD dataset and CBIS-DDSM dataset can be downloaded using The Cancer Imaging Archive data retriever tool: `https://wiki.cancerimagingarchive.net/display/NBIA/Downloading+TCIA+Images`.

The downloaded data should be stored in a `/data/` folder inside the top-level project directory (outside src). Some paths need to be changed in the code to point to this new directory (please refer to the README on GitHub for detailed instructions).

There are several scripts included in the `src/data_preparation` directory for preparing the CSVs of labels for our use, loading the images into TRAIN, VALIDATE and TEST directories and converting those images from the medical imaging DICOM format to PNGs. This allows the use of Tensorflow and Keras' `image_dataset_from_directory` function to pull a labelled dataset from the directory structure (where the images are stored in sub-folders for each class). See Figure A.1 for an outline of CBIS-DDSM structure.

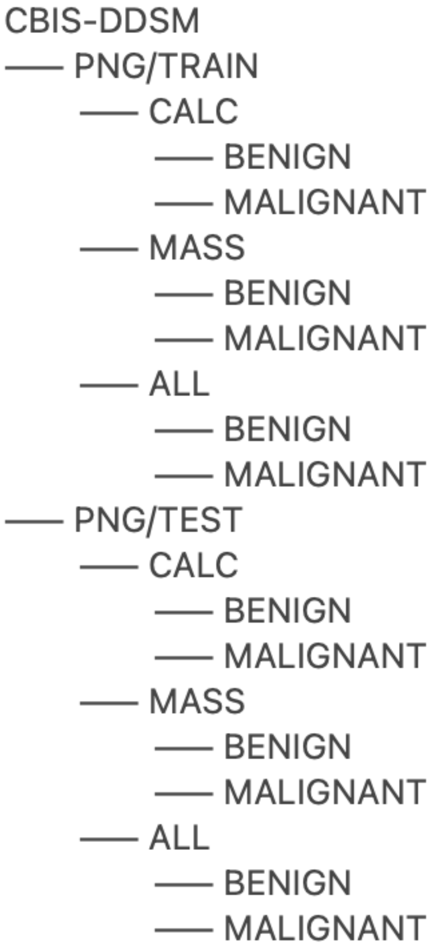
For the DDSM dataset:

1. Change paths in files in `src/data_preparation` to point to where the data has been downloaded

2. In `data_preparation` First run main method in `CBIS_DDSM_csv_combined.py` to generate correctly formatted CSVs
3. Run the `create_ddsm_pngs.py` to store images in the correct directory structure and convert them to PNGs

For the CMMD dataset:

1. Change paths in files in `src/data_preparation` to point to where the data has been downloaded
2. In `data_preparation` First run main method in `CMMD_preparation.py` to rearrange the labelled CSV for easier processing and set file paths
3. Run the `create_data_pngs.py` to split images into TRAIN/VALIDATE/TEST sets stratified by patient, store images in the correct directory structure (according to class) and convert them to PNGs



```
CBIS-DDSM
—— PNG/TRAIN
        —— CALC
                —— BENIGN
                —— MALIGNANT
        —— MASS
                —— BENIGN
                —— MALIGNANT
        —— ALL
                —— BENIGN
                —— MALIGNANT
—— PNG/TEST
        —— CALC
                —— BENIGN
                —— MALIGNANT
        —— MASS
                —— BENIGN
                —— MALIGNANT
        —— ALL
                —— BENIGN
                —— MALIGNANT
```

Figure A.1: Directory structure storing CBIS-DDSM images after data preparation. CMMD dataset also has a VALIDATION folder in the same structure, omitted for DDSM since the validation set is taken as a small subset of the training set on loading the data. Note: Mass and Calcification separation was not needed for this project but included for future extensions to this work.

Following data preparation - the pre-processing pipeline can also be applied to the dataset. This will save images in a similar directory structure as above in a separate folder: PNG-PREPROC. Please ensure you have sufficient storage space available to store both the original and pre-processed data on your machine.

1. Alter paths in `data_preprocessing/shared_preprocessing.py` to point to the location of separated data for DDSM and CMMD datasets (see above)
2. Call `run_pipeline()` for DDSM and CMMD separately, to create all the preprocessed images and store them in directories

Note: data augmentation is not applied at this stage to save space - augmentation is applied at training time and can be optionally included or excluded with a runtime flag (-a in command line arguments).

A.2 Usage

Before running - please create empty directories `output` and `saved_models` in the pipeline directory.

```
python main.py -d <dataset> -t -p -a -hy -lr <learning rate>
-m <model name> -s <image size> -n <name> -b <batch size>
- e <number fine-tune epochs>
```

- $-d$: select dataset for training/testing from "DDSM", "CMMD" or "BOTH, required parameter
- $-t$: test-mode flag, sets the program to run in test-mode when included
- $-p$: preprocessing flag, sets the program to use the pre-processed dataset when included (note Preprocessing Set Up from above must be completed first)
- $-a$: data augmentation flag, sets the program to trigger pre-specified data augmentations when included (horizontal and vertical flips, 90-degree rotations)
- $-hy$: hyperparameter-tuning flag, sets the program to run a hyperparameter tuning search with Hyperband when included (tunes the number of fully-connected layers after the base model and number of neurons in each layer, number of dropout layers and dropout rate in each layer, optimizer and learning rate)
- $-lr$: set the learning rate, defaults to 0.0001 for training while the base model is frozen (fine-tuning LR is 0.00001)
- $-m$: select a base model trained on imagenet for transfer learning: "MobileNet" (default), "VGG", "ResNet", "Inception", "DenseNet"
- $-s$: set image size: typically 160 (for 160 x 160 pixels) or 500 (500 x 500 pixels)
- $-n$: name of this trial (can be any value, should be entered the same for train or test-mode - no need to include .h5 at the end)
- $-b$: set the batch size, default 32

- $-e$: set the number of training epochs for fine-tuning, defaults to 50

A.3 Examples

Train the baseline model:

```
python main.py -d BOTH -lr 0.0001 -m MobileNet -s 500
-n "baseline-model" -b 32
```

Test the baseline mode:

```
python main.py -d BOTH -t -lr 0.0001 -s 500 -n "baseline-model" -b 32
```

Appendix B

Ethics Approval

University of St Andrews | FOUNDED 1413

School of Computer Science Ethics Committee

01 November 2022

Dear Rhona,

Thank you for submitting your ethical application which was considered at the School Ethics Committee.

The School of Computer Science Ethics Committee, acting on behalf of the University Teaching and Research Ethics Committee (UTREC), has approved this application:

Approval Code:	CS16535	**Approved on:**	01.11.22	**Approval Expiry:**	01.11.27
Project Title:	Deep Learning Techniques for Breast Cancer Detection in Mammograms				
Researcher(s):	Rhona McCracken				
Supervisor(s):	Dr David Harris-Birtill				

The following supporting documents are also acknowledged and approved:

1. Application Form

Approval is awarded for 5 years, see the approval expiry data above.

If your project has not commenced within 2 years of approval, you must submit a new and updated ethical application to your School Ethics Committee.

If you are unable to complete your research by the approval expiry date you must request an extension to the approval period. You can write to your School Ethics Committee who may grant a discretionary extension of up to 6 months. For longer extensions, or for any other changes, you must submit an ethical amendment application.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the School Ethics Committee.

Approval is given on the following conditions:

- that you conduct your research in line with:
  - the details provided in your ethical application
  - the University's Principles of Good Research Conduct
  - the conditions of any funding associated with your work
- that you obtain all applicable additional documents (see the 'additional documents' webpage for guidance) before research commences.

You should retain this approval letter with your study paperwork.

Yours sincerely,

Wendy Boyter

SEC Administrator

University of
St Andrews

University Teaching and Research Ethics Committee (UTREC)

Standard/Proportionate Review Filter

This form requires use of Microsoft Word desktop version (available via IT Services)

Standard/proportionate review filter

Please complete the filter questions - these determine whether your application will undergo standard review or proportionate review by your School ethics committee. If you are unsure which responses to select, please contact your School ethics committee.

For more information on the review process please visit the Ethical review application webpage.

Filter questions	**Yes**	**No**
Will your research involve participants from any of the following groups: • Children under 16 years of age (18 in England) • Protected adults • NHS patients or staff • Individuals engaged in criminal activity • Individuals in custody, care homes, or other residential institutions • Individuals impacted by a traumatic event such as war, displacement, acts of terrorism, abuse, discrimination, crime, disasters, life-changing illness or injury, bereavement • Individuals where there is any doubt over their capacity for freely given consent such as through cognitive impairment, language barriers, legal status, terminal illness. • Any other individuals where the researcher or SEC identifies a vulnerability that cannot be satisfactorily mitigated.	☐	☒
Will your research involve sensitive topics such as: • Criminal activity • Traumatic experiences like those detailed above • Self-identity i.e. gender, national, ethnic or racial identity • Body image • Mood or mental health conditions	☐	☒
Will your research involve collection, creation or inference of special category data. Special category data is identifiable data that is also: • personal data revealing racial or ethnic origin • personal data revealing political opinions • personal data revealing religious or philosophical beliefs • personal data revealing trade union membership • data concerning health • data concerning a person's sex life or sexual orientation • genetic data • biometric data (where this is used for identification)	☐	☒
Will your research involve collection, creation or inference of any other personal, confidential or sensitive data where you feel this might cause distress or that could cause harm should this data be intercepted?	☐	☒
Is there a risk that the research may result in participants becoming distressed? (For remote research, consider that this may be harder to monitor and whether participants will be able to access support)	☐	☒
Will your research involve the use of deception, the withholding of any information about the aims of the research or anything other than total transparency over your role as a researcher?	☐	☒
If you answered **YES** to **ANY** of the above, your application will undergo standard review by your SEC.		
If you answered **NO** to **ALL** of the above, your application will undergo proportionate review by your SEC.		

Appendix C

Graphs

C.1 Baseline Graphs

The graphs below show a comparison between the original baseline, with parameters specified in Figure 3.1, and the same model with data augmentation applied. Both models used image sizes of 500 pixels on the DGX machine.

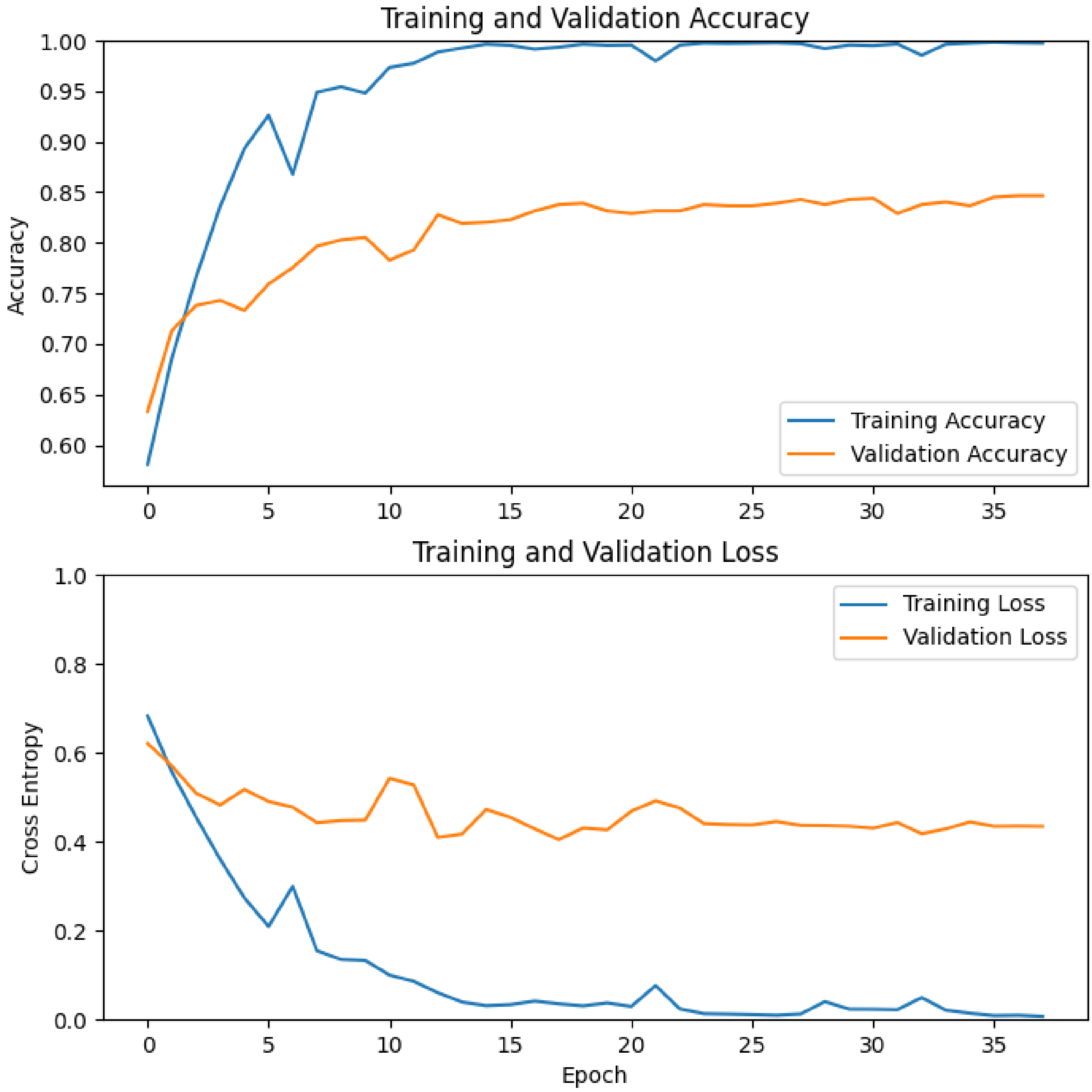


Figure C.1: Fine-Tuned Training of Original Baseline Model with 500 pixel images

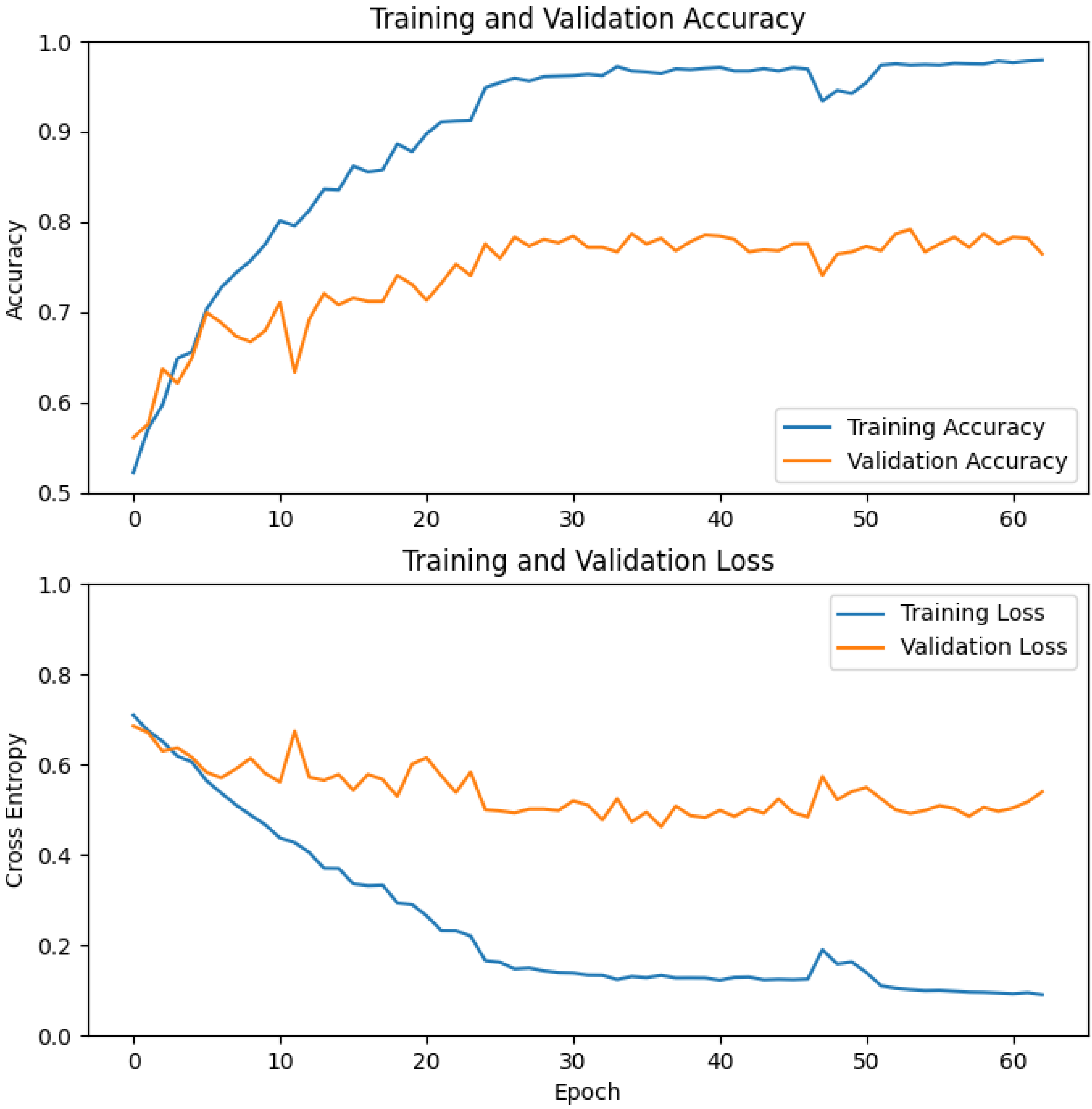


Figure C.2: Fine-Tuned Training of Baseline Model with Data Augmentation with 500 pixel images

C.2 Pre-Processing Learning Rate Investigation

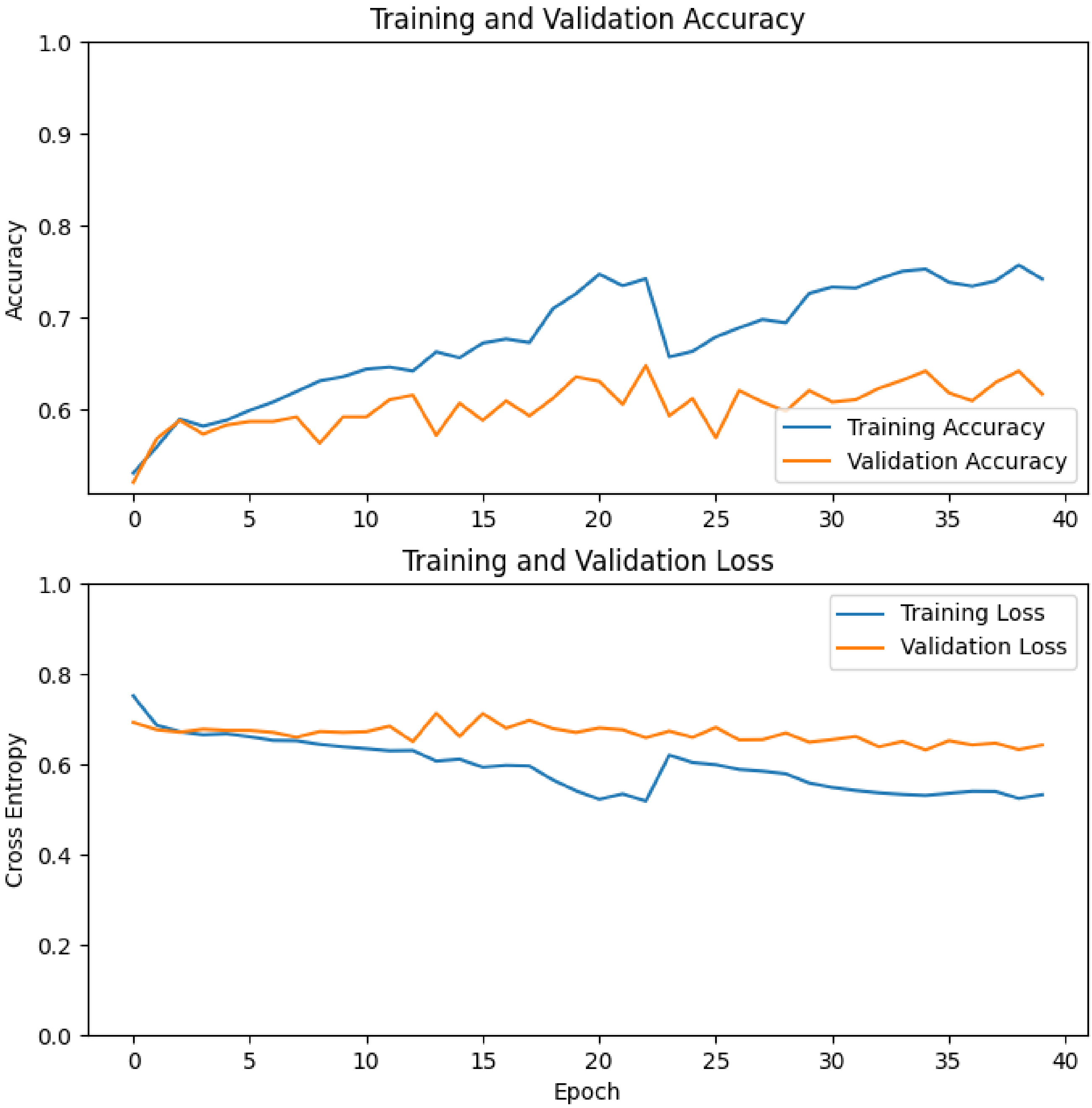


Figure C.3: Pre-Processing Investigation with Augmentation and Image Size of 160 × 160, Learning Rate of 0.001

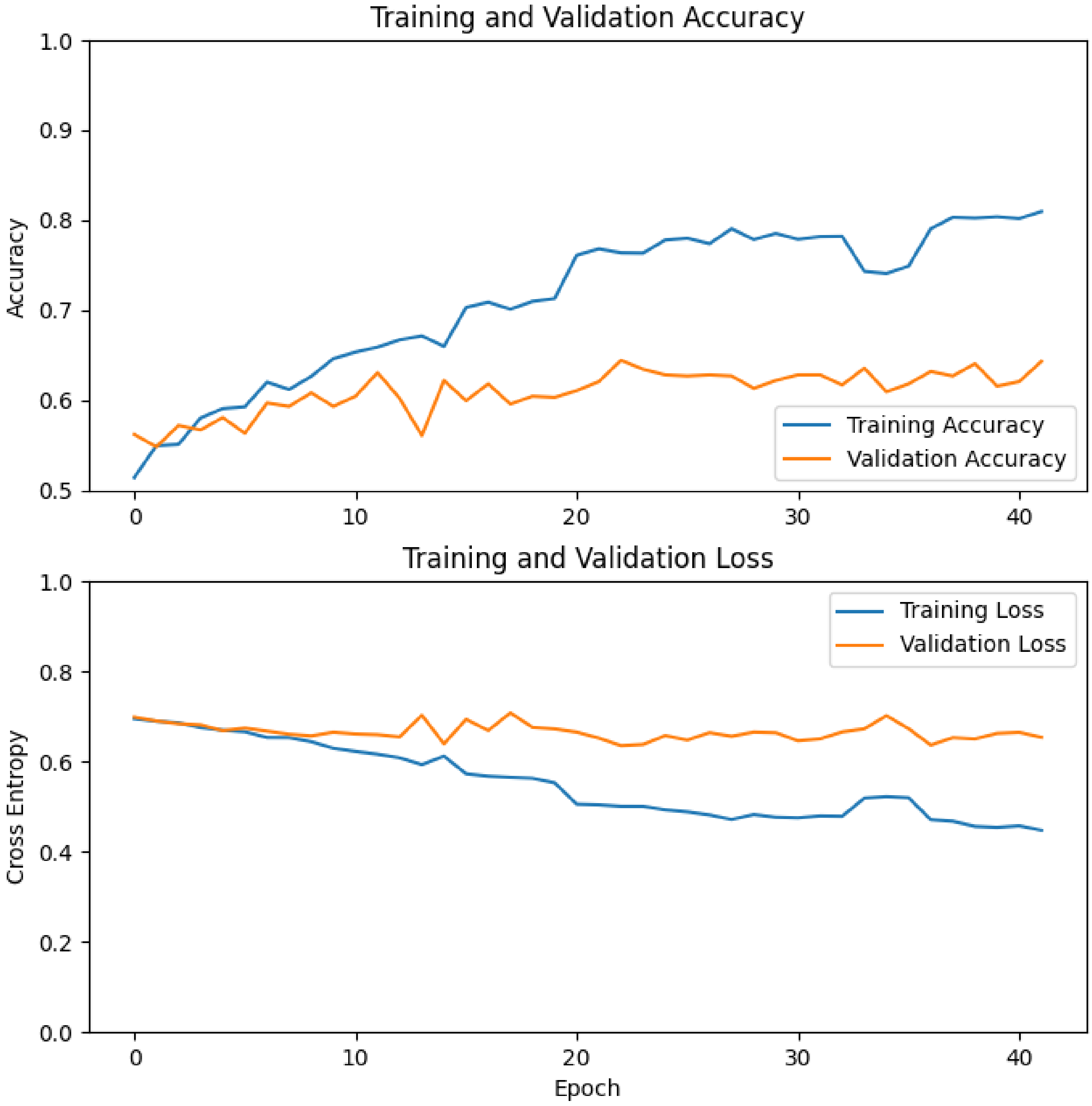


Figure C.4: Pre-Processing Investigation with Augmentation and Image Size of 160 × 160, Learning Rate of 0.0001

Appendix D

Image Sizes

The following images are resized CMMD mammograms to 500 × 500 pixels and 160 × 160 pixels. The size in this report may not be exactly this size but the difference in scale between the two is proportional to the real dimensions. This shows the difference in visibility of features in the two images.

89

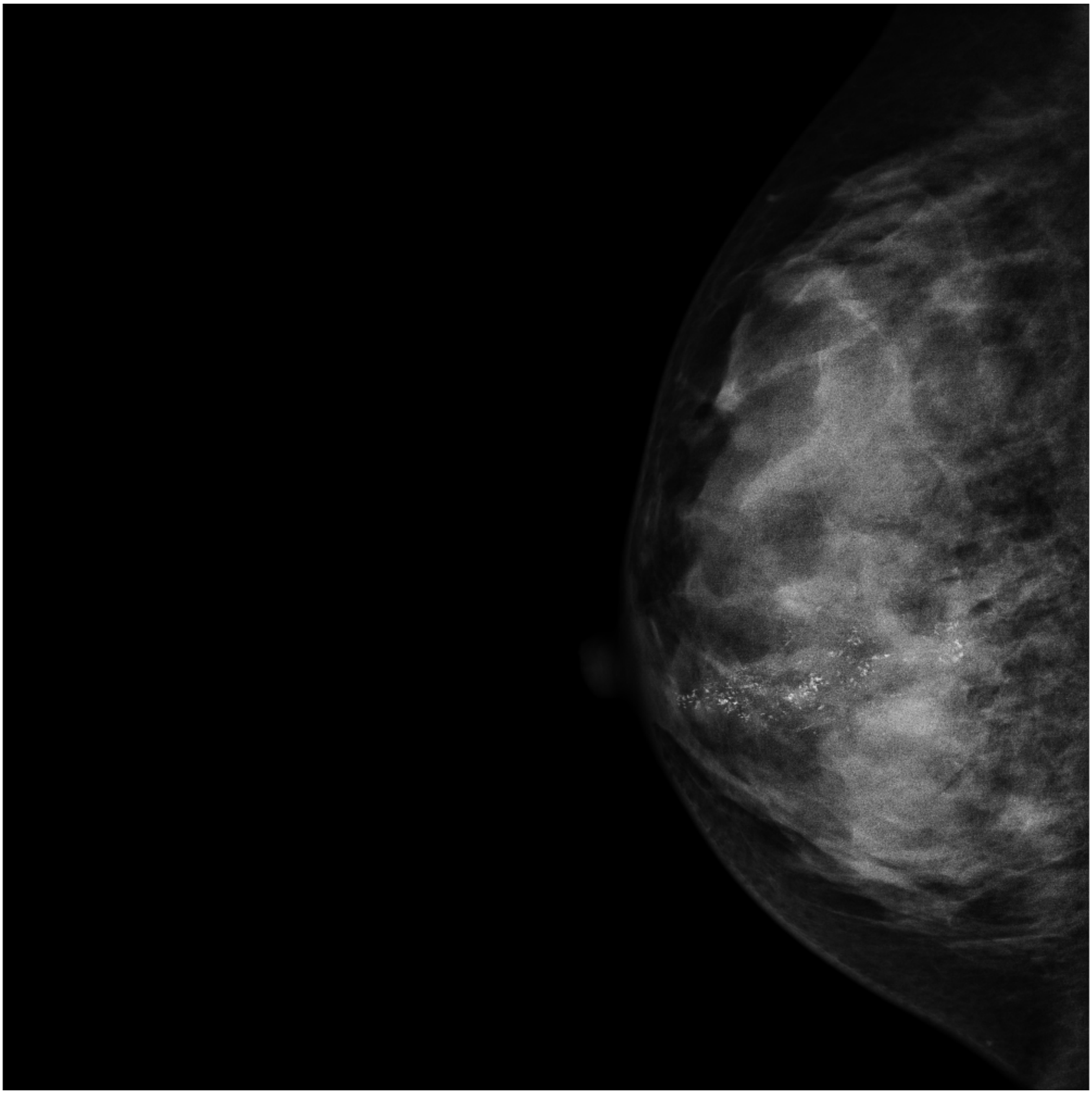


Figure D.1: Malignant CMMD with Calcifications Image Resized to 500 × 500 pixels

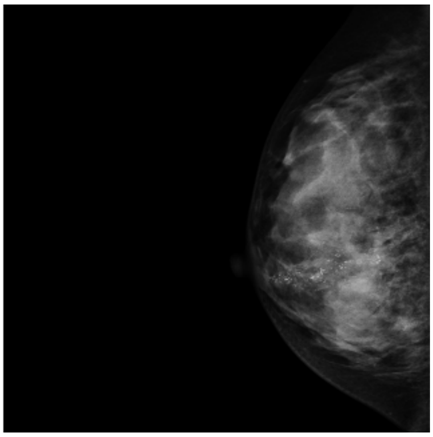


Figure D.2: Malignant CMMD Image with Calcifications Resized to 160×160 pixels