



# **ANALYSIS AND CLASSIFICATION OF MAMMOGRAM IMAGES USING DEEP LEARNING TECHNIQUES**

**Prashant Agarwal**

**Student ID: 19089198**

**Supervised by: Professor Conor Ryan**

Submitted to the University of Limerick, August 2020 in partial fulfilment

Of the requirements for the degree of Master of Science in Software Engineering

## **Abstract**

Breast cancer is the most common type of cancer in women and the reason for maximum deaths after lung cancer. Mammogram images obtained from Mammography is the most effective and frequently used technique for early detection of breast cancer. It uses low-energy X-rays to examine the human breast for diagnosis and screening. Although it's an effective technique, detection of cancerous tissue by radiologists at an earlier stage is a non-trivial task because in early stage its very small and it requires biopsy to confirm. And as of today, it is a known fact that early screening of cancer increases the chance of survival of patient's multiple times. So, some automated computer aided detection system is required to act as double reader to radiologist.

The advancement of deep learning techniques over a last decade has given a good result for object recognition and created the curiosity to apply these techniques over Mammogram images, study the behavior and evaluate their results.

In this Dissertation, we propose convolution neural network (CNN) which is a type of deep learning technique to classify cancerous Mammogram images, study and evaluate their results over full breast image and segmented breast image which is annotated by the radiologist. The VGG16 and RESNET50 CNN architecture models are trained and multiple classification metrics are generated to evaluate their results. Among all models that are trained VGG16 model gives best AUC value of 0.82.

To carry out all these experiments, Mammogram image processing techniques are applied to remove noise, artifact and pectoral muscle from original Mammogram image.

## **Declaration**

**Title:** ANALYSIS AND CLASSIFICATION OF MAMMOGRAM  
IMAGES USING DEEP LEARNING TECHNIQUES

**Author:** Prashant Agarwal

**Award:** Master of Science

**Supervisor:** Professor Conor Ryan

I hereby declare that this thesis is entirely my own work and does not contain material previously published by any other author, except where due reference or acknowledgment has been made. Furthermore, I declare that it has not previously been submitted for any other academic award.

---

Prashant Agarwal

August 2020

## **Acknowledgement**

Firstly, I thank God for giving me the opportunity to pursue my academic goals from University of Limerick

I would like to thank my supervisor, Professor Conor Ryan for guiding me throughout my journey of dissertation work. Many times, I find myself struck at dead-end and he was the one who showed me the path out of it. His criticism and feedback always helped me to improve my skills and contributed to the success of this dissertation.

To my parents – Mr. Pradeep Agarwal and Mrs. Sunayna Agarwal for their continuous support, love and guidance and a sibling – Surbhi Agarwal for motivating me to pursue my goals.

A special thanks to Dr Arjun Agarwal for helping me understand medical literature and Paras Agarwal for guiding me throughout my dissertation work.

Thanks to my friends -Shashank, Akankshha, Ria, and Kavya for their unconditional support and love.

Finally, I would like to appreciate the lecturers of the CSIS department for all their support throughout the entire master's course.

## Contents

CHAPTER1- INTRODUCTION .....	8
1.1 Background .....	8
1.2 Mammography .....	9
1.3 Research Problem .....	10
1.4 Outline.....	12
2.1 Noise Removal:.....	13
2.2 Mammogram Image Label Removal and Breast Contour Extraction.....	14
2.3 Pectoral Muscle Segmentation.....	15
2.4 Feature Extraction and Classification .....	16
CHAPTER 3- IMPLEMENTATION, CLASSIFICATION AND EVALUATION .....	18
3.1 Implementation .....	18
3.1.1 Programming Language and Software Libraries .....	18
3.1.2 Cloud Platform.....	18
3.2 Classification.....	18
3.3 Evaluation Techniques.....	25
CHAPTER 3- MAMMOGRAM DATA COLLECTION AND PREPROCESSING .....	28
3.1 Data Collection: .....	28
3.2 Data preparation and cleaning.....	29
3.3 Mammogram Image Preprocessing .....	29
3.4 Segmentation.....	34
3.5 Summary:.....	36
CHAPTER 4- ANALYSIS .....	37
4.1 Experiments .....	37
4.1.2. Segmented Images .....	37
4.1.3. Full Images.....	48
4.2 Summary .....	55
CHAPTER 5 : LIMITATIONS, CONCLUSION and FUTUREWORK .....	56
5.1 Limitations Encountered .....	56
5.2 Discussion on Research Questions .....	57
5.3 Future Work.....	58
References.....	59

## List of Figures

Figure 1: CC and MLO Mammogram. ....	10
Figure 2: Neuron Structure .....	19
Figure 3: MultiLayer Neuron Structure .....	20
Figure 4: VGG network architecture .....	23
Figure 5 : RESNET neural network architecture .....	24
Figure 6: Sample ROC curve.....	26
Figure 7: Mammogram segmentation .....	28
Figure 8: Extracted Metadata.....	29
Figure 9 : Histogram Equalization Mammogram Image .....	30
Figure 10: Artifact Original Mammogram image.....	31
Figure 11 : Mammogram Image After Thresholding .....	32
Figure 12: Artifact removed Mammogram Mask .....	33
Figure 13: Pectoral MuscleRemoval.....	34
Figure 14: Mammogram image segmentation example.....	35

## List of Tables

Table 1 : Sample Confusion Matrix.....	25
Table 2: Segmented images AUC, Loss v/s Epoch .....	38
Table 3 : Segmented Images Test ROC-AUC Curve .....	40
Table 4: Segmented Images Confusion Matrix .....	43
Table 5 : Segmented Images Classification Metrics.....	46
Table 6: Full Images AUC, Loss v/s Epoch .....	49
Table 7: Full Images ROC-AUC V/S Epoch.....	50
Table 8 : Full Images Confusion Matrix.....	52
Table 9 : Full Images Classification Metrics .....	54
Table 10: Research Question 2 Comparison Table.....	57
Table 11: Research Question 3 Comparison Matrix.....	58

## List of Abbreviations

CC	Cranial-caudal
MLO	Medio-lateral
ROC	Receiver Operating Characteristics



# CHAPTER1- INTRODUCTION

## 1.1 Background

Cancer is one of the most leading cause of death and occurs as the result of mutation or abnormal changes in the gene responsible for regulating the growth of cells. The cells replace themselves through an orderly process of cell growth which results in healthy new cells taking over as old cells die out. Mutation can turn on certain genes and turn off others, which changes the cell's ability to keep dividing without control and forms a tumor [1].

A Tumor can be benign or malignant, Benign tumors are noncancerous while malignant tumors are cancerous. Benign tumors spread slowly and don't spread to other parts of the body while malignant tumors can spread beyond the other parts of the body. Cancer is the disease that can affect almost any tissue and organ but in women breast cancer is the most common type of cancer and accounts for a large proportion of deaths. According to American cancer society "Breast cancer is a group of diseases in which cells in breast tissue change and divide uncontrolled, typically resulting in a lump or mass. Most breast cancers begin in the lobules (milk glands) or in the ducts that connect the lobules to the nipple" [1].

It is the most common type of cancer and the reason for the greatest number of deaths after lung cancer. Approximately 1 in 8 women will be diagnosed and one in 39 women will die from breast cancer [2]. According to the world health organization, breast cancer impacts nearly 2.1 million women each year and accounts for 627,000 deaths, which is 15% of all cancer-related deaths. Breast cancer rates are more in developed countries, but rates are increasing in developing countries as well [3].

Breast cancer is identified by the lump but generally, it is painless when it is small and easily treatable. So early detection is the only option to detect the tumor and hence increase the chance of recovery and bring down the mortality rate. There are two early detection strategies, first is early diagnosis and second is screening. Early diagnosis focuses on providing resources for early effective diagnosis service while screening consists of screening women before any symptoms actually appear. Today mammography, ultrasound, and MRI are the available tools for screening of breast cancer but till now mammogram is the best tool among them. It uses low energy x-rays

for screening breast tissue. It has been shown to reduce mortality by approximately 20% but confirming cancer in it is a non-trivial task. Denser breasts are harder to diagnose as they have low contrast between cancerous lump and background. Most of the lump seen on mammograms is benign (non-cancerous) but to confirm that the tissue of that region has to be obtained using biopsy further Image quality and human error also aid the complexity of discriminating cancerous from noncancerous lumps.

So, Radiologists try to increase specificity and sensitivity of mammograms by double reading the mammogram by different radiologists, but it increases the cost to the patient which results in women less inclined to participate in the process.

So, to automate the detection of cancer some automated CAD (computer-aided design) systems are required which will help radiologists in detecting cancer. Although the final decision is taken by radiologists, CAD systems act as a double reader [1]. CAdE and CAdx are two types of systems in which CAdE stands for computer-aided detection in which these systems help radiologists in locating and identifying abnormalities and leaving the interpretation to the doctor while CAdx stands for Computer-Aided Diagnosis system in which it acts as a second decision.

According to the findings of Elizabeth Wende Breast Clinic, Rochester, New York “Computer-aided detection (CAD) of breast cancer using digitized mammograms could have detected malignancies at least 1 year earlier than film assessment by radiologists alone” [4].

Advancement in machine learning mainly in deep neural networks attracted researchers from the medical imaging field due to their automatic feature extraction and representation learning ability. Improvement in computation ability is another important factor that enabled training large neural networks. Detection and classification of cancerous and non-cancerous tumors are difficult even for deep neural networks because the size of Mammograms is typically 4000\*3000 but the cancerous region of interest (ROI) can be as small as 100\*100. So usually these classification tasks are limited to classifying manually annotated ROIs.

## **1.2 Mammography**

It's a dedicated low-dose x-ray technique based on breast screening. It aids in early detection and diagnosis in women. Digital mammogram also called full-field digital mammography (FFDM) is a mammography system in which the x-ray film is replaced by electronics that convert x-rays into

mammographic pictures of the breast. These systems are similar to those found in digital cameras and their efficiency enables better pictures with a lower radiation dose. These images of the breast are transferred to a computer for review by the radiologist and for long term storage. Images in mammograms are taken from two views namely ‘Cranial-caudal (CC)’ and ‘Medio-lateral (MLO)’. CC view is a view from above while the ‘MLO’ view is the view from the side at an angle. For better understanding please refer Figure 1.

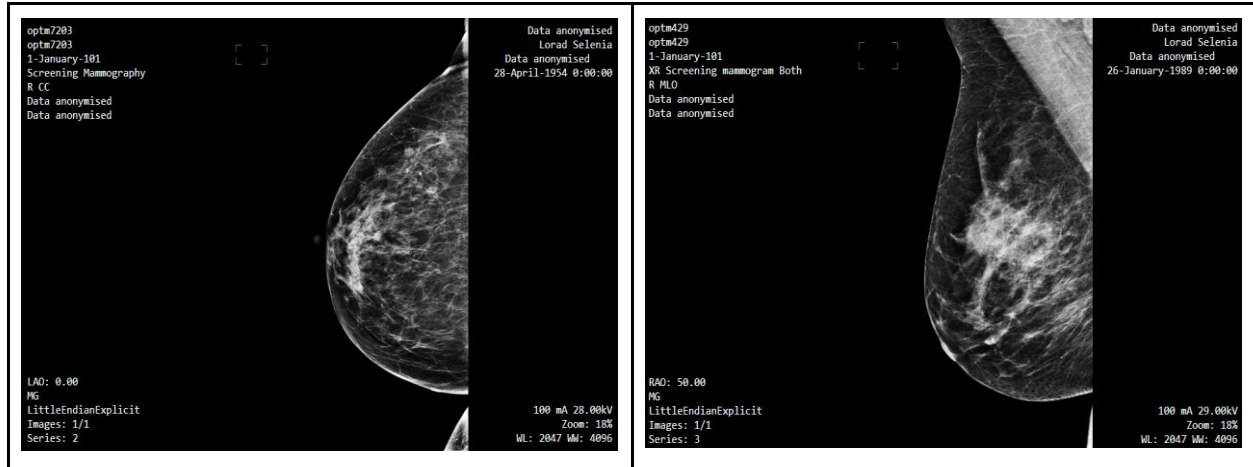


Figure 1: CC and MLO Mammogram. On the left is the ‘Cranial-caudal (CC)’ and on the right ‘Medio-lateral (MLO)’

### 1.3 Research Problem

There has been a lot of development in the field of deep learning in the recent past which can be validated by its ability to solve complex problems like speech to text and object recognition.

Seeing these developments, this dissertation looks to apply deep learning techniques for the classification of Cancerous Mammogram images and evaluate the classification using multiple classification evaluation metrics. To achieve this following task are performed

1. Collection of images.
2. Apply data and image pre-processing techniques and prepare a dataset.
3. Apply deep learning techniques to build classifiers.
4. Evaluate the performance of the classifier on new images using multiple classification metrics

### 1.3.1 Research Questions

**1. Does transfer learning using models trained on an ‘ImageNet’ dataset improve mammogram deep learning cancer detection results?**

Deep learning models require training on large amounts of data and to get the mammogram dataset of that size is difficult so in this research study, we will find if fine-tuning pre-trained models on ‘ImageNet’ dataset improves results for cancer detection.

**2. How much does the size of the mammogram input image affect deep learning model results?**

The size of the original mammogram is large typically around 4000\*3000 and training a deep learning model on images of this size will require high computation power and hence increases the cost. This research will help in knowing how deep learning model results change with change in size of input mammogram image.

**3. Between ‘CC’ and ‘MLO’, which view gives better deep learning results?**

As we know that during Mammogram screening, we capture images using two views namely ‘CC’ and ‘MLO’. This research will help in ascertaining as to which view captures cancerous cells in a more efficient manner.

### 1.3.2 Methodology

This is quantitative research based on data analysis. Multiple experiments will be conducted from the same data source but with different data samples. These experiments would include training and creation of multiple deep learning models, each of these models will be evaluated using multiple evaluation metrics like AUC, Precision, Recall/Sensitivity, Accuracy, and Specificity. The primary data used for this study is from OPTIMAM Medical Image Database (OMI-DB), a collection of NHS Breast Screening program images from multiple breast screening centers across the UK. This dataset contains images from multiple mammogram images.

## 1.4 Outline

This section gives the common view of what to expect from this dissertation work and the way it is designed. This introductory chapter covers background and introduction to breast cancer, its screening using mammogram images, and the scope of deep learning in the classification of cancerous mammogram images by outlining research questions. The next chapter talks through the literature review where through discussion on this research study is done. The third chapter talks of technical infrastructure used to carry out experiments, types of classification models, and its evaluation. Chapter four includes discussion about data and its pre-processing techniques. The fifth chapter covers the analysis of experimentations performed and their evaluation results. At Last, the sixth chapter begins with limitations and problems faced during experimentation and answers the research questions and ends with the scope for future work.

## **CHAPTER 2 - LITERATURE REVIEW**

This chapter covers the history and state of the art research done to unearth the full potential of CAD systems for the early detection of breast cancer. The two most important parts before classification are mammogram image processing and feature extraction.

Preprocessing is done before we extract features because original mammogram images contain labels and may contain noise that needs to be removed. Thus the main aim of preprocessing is to improve the quality of the image: this includes noise (random variation of brightness or color information in images) removal, labels removal, breast contour segmentation to remove extra unwanted part from the image and the suppression of pectoral muscle because it appears with the same density as most abnormal regions in a mammogram.

### **2.1 Noise Removal:**

D.Narain Ponraj et al. discussed the advantages and disadvantages of various types of filters for denoising the image. Denoising by low pass filters reduces the noise in the image but also blurs the image, this effect can confuse the classifier in understanding small or low-density cancers and larger lesions, particularly in dense breast tissue. Spatial and frequency domain filters used as a tool for image enhancement. Below is the list of some famous filters and their usage [5].

#### **Mean filter**

Mean filter replaces the mean intensity values of neighborhood pixels, by doing so it reduces the local variance around a cell. It is optimum for removing Gaussian noise (noise following normal distribution in the time domain) but since mammogram images may contain noise other than Gaussian noise, so this technique is not good for noise removal in mammograms.

#### **Adaptive mean filter (AMF)**

AMF filters are useful if we want to reduce the blurring effect. It adapts the value of the image locally based on image statistics such as mean, variance, and spatial correlation to effectively detect and preserve edges and features.

#### **Histogram Equalization**

A histogram is a graphical representation of the number of pixels for each intensity value present in an image and histogram equalization is the spreading out of the most frequent intensity values in an image. By doing histogram equalization, the overall contrast of an image is improved.

### Adaptive Histogram Equalization (AHE)

This method differs from the normal histogram equalization in a way that it calculates multiple histograms corresponding to different regions and uses this to redistribute the intensity value of the image. This makes it more useful in enhancing the definition of the edges in each region.

### Contrast Limited Adaptive Equalization (CLAHE)

CLAHE works in the same way as AHE but the main difference is that it prevents over-amplification of noise signals by imposing limiting characteristic, which is done by clipping the histogram at a predefined value.

## 2.2 Mammogram Image Label Removal and Breast Contour Extraction

**H. Mirzaalian et al.** proposed the algorithm for extracting breast contour [6]. His process starts with histogram equalization which is necessary to make the algorithm work similarly on all images. Then he proposed one mask for convolution with the image as shown below

$$MASK = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

This was useful for the type of image they used in their work, but nothing is mentioned that it will work for all types of mammogram images.

**Jawad Nagi et al.** implemented the label removal and breast contour segmentation by following the steps given below [7]:

1. They converted the grayscale image to binary [0,1] format by applying the thresholding method. In this method, all the pixels above a fixed threshold are converted to 255 and others were replaced with 0 as a pixel value.

2. After the conversion, he performed morphological operations such as dilation, and erosion.

“In morphological operations, each pixel in the output image is based on the comparison of the corresponding pixel in the input image with its neighbors” [21]. Combination of dilation and erosion is used to remove small objects from an image and smooth the border of the objects.

3. All objects present in the image are then labeled using “bwlab” function in MATLAB.  
This function does the labeling of connected components in the 2-D image.
4. The area of all regions is calculated and the area with the largest region is extracted.

## **2.3 Pectoral Muscle Segmentation**

For Pectoral Muscle Segmentation, most of the methods are based on global contrast or intensity based. Raba et al. presented the contrast-based method for segmentation, but this method comes with the problem of over segmentation when the contrast between muscle and breast tissue is very blur [8].

Other popular methods are based on region-based approaches, generally it works by examining the neighboring pixels of stating points and partition the image into multiple regions. Jawad Nagi et al. proposed a method for pectoral muscle segmentation using Seeded Region Growing (SRG) [8]. In this method, the seeds (grid points selected to agglomerate the surrounding control volume) is injected and image is divided into regions with a property that each connected component of the region meets exactly one seed point. Makandar et al. also suggested a method using the seeded Region Growing method (SRG) [9]. This seed point is selected automatically based on the orientation of the breast but the main problem with the seeded approach is the mammogram images are large so computation cost is very high.

Boss et al. used a thresholding technique to get a binary image and applied an 8- neighborhood connected component algorithm [10]. This algorithm is used to compute connected components in a graph or regions in an image. Two pixels belong to the same regions if there exists a direct path between them. It comes with two types of connectivity with 4 and 8 neighbors. His work was effective but doesn't consider the orientation of the mammogram.

Sheba et al. tried to remove pectoral muscle based on bounding boxes. They suppose pectoral muscle to be on the upper left corner or upper right corner. They first find the coordinates of the bounding box of the breast contour which includes the coordinates of the contour then they extracted the pectoral muscle using a cropping triangle of length one-third of the width of the



bounding box [11]. This method is not so powerful because the area of pectoral muscle is not fixed, sometimes it will more or less.

Toshpulatov et al. used a watershed algorithm to remove pectoral muscle. He tested the algorithm on 188 images taken from inBreast database [12]. Out of 188 mammogram images, 132 images were segmented correctly, 26 gave acceptable segmentation and 32 gave a bad performance. This is because the intensity of pectoral muscle and breast boundaries are very similar.

## **2.4 Feature Extraction and Classification**

There has been vast research before and after the advent of deep learning for breast mass classification.

Conor Ryan et al. achieved a TPR of 100% which uses GP at its core. They generated textual features and feed through PCA to get important features and apply GP to generate classifiers. They used the research by Tabar [13] to assume that both breasts from the same patient have the same textual characteristics and made a hypothesis that breasts that differ textually may contain suspicious regions [14].

Li Shen et al. proposed end to end training approach in which lesion annotations are required only in the initial training phase and later on we can work on image-level annotations. The main work of their study is to use two datasets. One dataset has marked lesions, they train CNN on this data set at patch level and use the weights of this model to train another model on another dataset. This way we can avoid the reliance on lesion annotations. He tested on CBIS-DDSM dataset and achieved AUC of 0.91 (sensitivity 86.1%, specificity:0.1%) [15].

William Lotter et al. trained multi-scale CNN, first they trained CNN on segmentation masks of lesions and then used these learned weights to initialize a model trained on whole weight. They achieved 0.92 AUCROC on DDSM dataset [16].

Dina A. Ragab et al. used two approaches for finding the region of interest, first is manual based while the second approach uses the technique of threshold and region based. They used Alexnet for feature extraction and they used Support Vector Machine (SVM) as a classifier after the last

fully connected layer. They achieved AUC of 0.88 for both approaches i.e. samples obtained manually and same for threshold and region [17].

Krzystof et al. studied the impact of training set size and image size on prediction accuracy using deep neural networks. They observed that good performance can only be achieved if we used the same resolution image and accuracy can increase if we use a larger training dataset [18].

Hiba Chougrad et al. used three datasets DDSM, BCDR, 'Inbreast' dataset to train three models VGG16, RESNET50, INCEPTIONV3 using random initializer, and pre-trained weights but he used accuracy as metrics. Accuracy does not give a good evaluation when we have an imbalance dataset so we cannot accurately compare their method [19].

Thijs et al. in his work made a comparison between manually designed feature sets and CNN. They trained both systems on 45,000 images and results show CNN outperformed the traditional system. They also confirmed that several manually designed features can give small improvements. They took CNN +Contrast, texture, topology, location, context individually with CNN and all with CNN. They observed that CNN+All features outperform individual trained CNN and CNN with one feature [20].

## **CHAPTER 3 - IMPLEMENTATION, CLASSIFICATION AND EVALUATION**

This chapter focuses on the Implementation, Classification, and Evaluation part of our dissertation work. The implementation section covers the programming language, software libraries, and cloud platform used for experimentation. The main aim of this research study is the classification of the cancerous and non-cancerous mammogram images, so classification is the most important part of this chapter and talks about neural networks that are used in this dissertation work. To measure the performance of classifiers, one or more evaluation metrics need to be applied so in the section of Evaluation, classification metrics are described which are used to evaluate our classifiers.

### **3.1 Implementation**

#### **3.1.1 Programming Language and Software Libraries**

The programming language to be utilized for executing the Mammogram classification task is Python. Its developing fame in the field of machine learning and deep learning because of its capacity to deal with complex numerical calculations and easy to use. The Python programming language has a wide range of machine learning libraries that makes it easier to handle complex mathematical operations when applied to datasets regardless of size. Some of the libraries that will be using are NumPy, Pandas, Matplotlib, OpenCV, and TensorFlow which are all open source libraries. OpenCV is the open-source machine learning and computer vision software library that we used extensively in image preprocessing tasks. For training deep neural networks, we will be using keras API, it is a high-level API of TensorFlow:” an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning” [21][22].

#### **3.1.2 Cloud Platform**

We will be using the Google Cloud Platform (GCP) and Google Colab for all the preprocessing and classification tasks [23][24].

### **3.2 Classification**

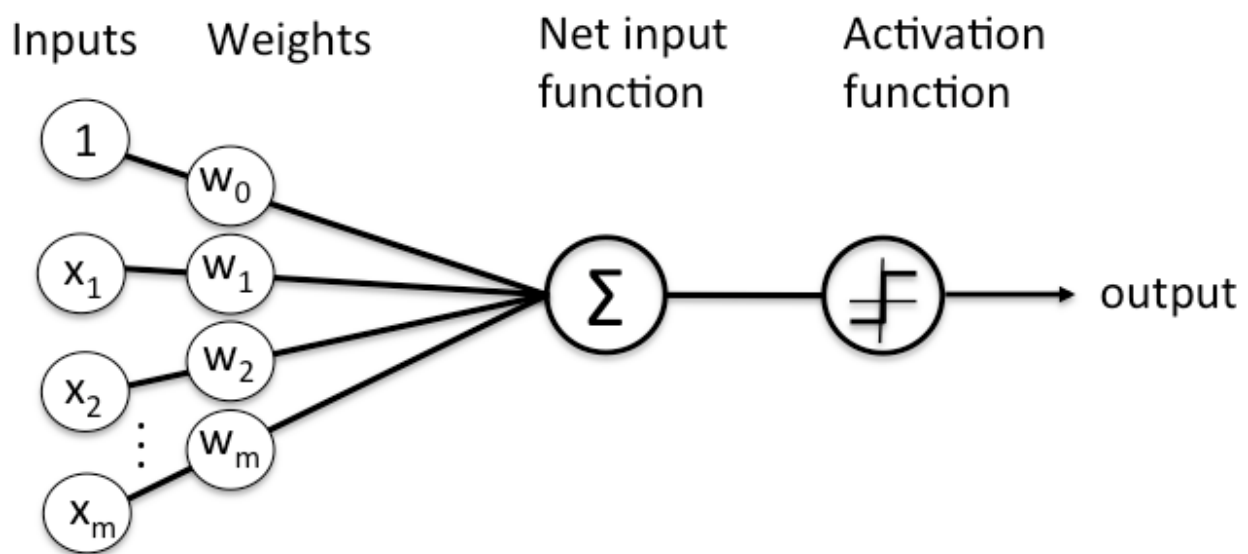
For classification tasks, we will be using VGG16 and RESNET50 Convolution neural networks (CNN). Before understanding what, these are, let’s first discuss neural networks and CNN.

### 3.2.1 Neural Networks

Neural networks were first proposed in 1944 by two University of Chicago researchers namely Warren McCullough and Walter Pitts. The idea behind neural networks is to simulate the structure of lots of densely interconnected brain cells so your software can learn things, recognize patterns, and make decisions like humans do. The good thing is that we don't program it to learn specifically, it learns all by itself [25].

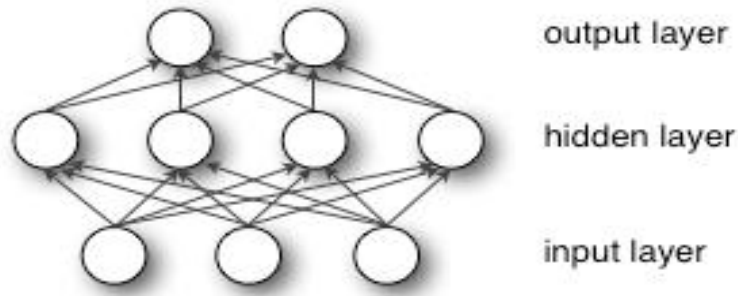
Neural networks consist of multiple layers and layers are made up of nodes.

The below figure shows the basic structure of neurons.



**Figure 2: Neuron Structure [25]**

In the above figure  $w_0, w_1, \dots, w_m$  are weights,  $x_1, x_2, x_3, \dots, x_m$  are inputs and activation function is the mathematical equation that determines the output of the neural network. If we combine this basic neuron in the form of multiple layers, we will get a multilayer neural network as shown in the figure given below.



**Figure 3: Multi-Layer Neuron Structure [25]**

As mentioned in the above figure lowest most layer is the input layer and the topmost layer contains the output layer and the middle layer is hidden layer but there can be more than one hidden layer.

### **3.2.1.1 Neural Networks Model Training Parameters**

#### **Weight Initialization and Transfer learning**

Neural networks weight initialization is an important step to properly train our model because whole classification results depend upon neural network weights. Suppose if we initialize all weight with the same value then every neuron will be trained similarly and will compute the same thing. So, it's better if we do random initialization of weights. We can initialize the weights using the second technique i.e. transfer learning. In transfer learning, we first train a base neural network model on a very large dataset e.g. ImageNet which contains 1.2 million images with 1000 categories and then uses this model as an initialization for the target model to be trained on the target dataset. In this dissertation, work models are trained using both random initialization and using transfer learning from neural network models trained on the ImageNet dataset.

#### **Training, Validation and Test dataset**

The training dataset is the sample of data used to train the model. The model learns from this sample of data. The validation dataset is used to evaluate the performance of a model fit on the training dataset while tuning model parameters. The test dataset is used to evaluate neural network models after training and parameter tuning is done and the model is ready to predict unseen data. In this dissertation work, all the models are trained with 80-20-20 rule that means first, the whole dataset is divided into training and test dataset with the ratio of 80% and 20 % respectively further, the

training dataset is divided into training and validation dataset with the ratio of 80% and 20% respectively [26].

### **Epoch and Batch Size**

One Epoch is counted when the entire dataset is passed forward and backward through the neural network once. Usually, we train models by more than one Epoch because updating weights optimally in a single pass is not sufficient. The right number of epochs is only decided by monitoring the behavior of loss and classification matrix values. For example, if validation loss (loss calculated on validation dataset) starts increasing after some value of epoch then we can infer that we need to stop our training to further epochs. As the whole data set can't be passed at once, so we divide the whole dataset into batches. Let's say the size of our dataset is 100 and the batch size is 5, so there will be 20 iterations in each epoch [27].

### **Cross-Entropy Loss**

It measures the performance of a classification model whose output is a probability value and lies between 0-1. In binary classification it can be calculated using the following equation:

$$\text{Loss} = -(y \log(p) + (1-y) \log(1-p))$$

In the above equation  $y$  is a binary indicator (0,1),  $p$  is predicted probability, and  $\log$  is the natural log. All losses calculated in this work are cross-entropy Losses.

#### **3.2.1.2 Convolution Neural Network (CNN/ ConvNets)**

CNN is a type of neural network which is designed especially for images as input. Unlike the regular neural networks, the layers of a ConvNet have neurons arranged in 3 dimensions: width, height, depth.

Following are the layers used to build neural networks:

#### **Convolution layer**

The convolution layer in CNN applies filters to input and creates output feature maps.

. Every filter consists of the width, height, and depth which extends the full depth of the input image. For example, if convNet has size  $3 \times 3 \times 1$ , it means ConvNets has 3-pixel width, 3-pixel height, and 1-pixel depth which is the same as the depth of the image. During training, we slide each filter across the width and height of the input image and produce 2 dimensional out. The sliding operation can be controlled by setting stride parameters, the default stride in two

dimensions is (1,1) for height and width movement. Now if we have 12 filters, each of them will produce a separate 2-dimensional output, we will stack them across the depth dimension and produce the output.

### **Pooling layer**

The pooling layer between neural networks is used to reduce the spatial size of the output of the convolution layer to reduce the number of computation parameters in the network. This is also used to avoid overfitting. There are two types of pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

### **Fully Connected Layer**

It takes the output of the previous layer, flattens them, and turns them into a single vector that can be used as input for the next stage.

#### **3.2.1.2.1 VGG**

Karen Simonyan and Andrew Zisserman in the year 2014 proposed VGG network in the paper “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. The main concept of inventing the model is to simplify the complexity of earlier models like Alexnet by using each convolution of 3\*3 with stride =1, same padding, and max pool size of 2\* 2 and stride of 2 [28].

The below figure shows the 5 VGG configurations named A-E, among all these VGG networks, this dissertation work uses VGG16 network.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

**Figure 4: VGG network architecture [29]**

VGG16 network includes the following layers:

- 1.Convolution layer of 3\*3 kernel with 64 filters
- 2.Convolution layer of 3\*3 kernel with 64 filters + Max pooling of size 2\*2, stride 2\*2
- 3.Convolution layer of 3\*3 kernel with 128 filters
- 4.Convolution layer of 3\*3 kernel with 128 filters + Max pooling of size 2\*2, stride 2\*2
- 5.Convolution layer of 3\*3 kernel with 256 filters
- 6.Convolution layer of 3\*3 kernel with 256 filters
- 7.Convolution layer of 3\*3 kernel with 256 filters + Max pooling of size 2\*2, stride 2\*2
- 8.Convolution layer of 3\*3 kernel with 512 filters
- 9.Convolution layer of 3\*3 kernel with 512 filters
- 10.Convolution layer of 3\*3 kernel with 512 filters+ Max pooling of size 2\*2, stride 2\*2



11. Convolution layer of 3\*3 kernel with 512 filters
12. Convolution layer of 3\*3 kernel with 512 filters
13. Convolution layer of 3\*3 kernel with 512 filters + Max pooling of size 2\*2, stride 2\*2
14. Fully connected with 4096 nodes
15. Fully connected with 4096 nodes
16. Output layer with SoftMax activation with 1000 nodes

Since our problem statement is the two-class classification (cancerous or noncancerous) so, neural network models used in this dissertation work will use 2 nodes in the output layer.

### 3.2.1.2.1 Residual Networks (RESNET)

RESNET models were designed to utilize the benefits of very deep neural networks. The below figure shows the available RESNET architecture with 18-layers, 34 layers, 50 layers, 101 layers, and 152 layers.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

ResNet Architectures

**Figure 5 : RESNET neural network architecture [30]**

Out of all the above RESNET architecture, we will be using RESNET50 architecture, it is a convolution neural network that is 50 layers deep and follows the following architecture.

1. The first layer is a convolution layer with a kernel size of 7\*7 and 64 different with a stride of 2.
2. The next layer is the max-pooling layer with a stride of 2.

3. In the next convolution layers there is  $1 \times 1, 64$  kernel following this a  $3 \times 3, 64$  kernel and at last  $1 \times 1, 256$  kernels. This architecture is repeated 3 times which gives 9 layers in this step.
4. Next convolution layers we see  $1 \times 1, 128$  kernel following this a  $3 \times 3, 128$  and lastly  $1 \times 1, 512$ . This architecture is repeated 4 times which gives us 12 layers
5. After that, there is a kernel of  $1 \times 1, 256$  and two more kernels with  $3 \times 3, 256$  and  $1 \times 1, 1024$  and this is repeated 6 times giving us 18 layers.
6. And then again, a  $1 \times 1, 512$  kernel with two more of  $3 \times 3, 512$  and  $1 \times 1, 2048$  and this Was repeated 3 times giving us a total of 9 layers.
7. After that, we do an average pool and end it with a fully connected layer containing 1000 nodes and at the end a SoftMax function so this gives us 1 layer.

Like VGG16, in RESNET50 architecture also we will be using the last layer with 2 nodes instead of 1000 nodes.

### 3.3 Evaluation Techniques

#### 3.3.1 Confusion Matrix

In our work, we are classifying Mammogram images into cancerous and non-cancerous. In this dissertation work “Cancerous” is taken as positive class and “Non-Cancerous” as negative class.

We can summarize the cancer prediction by using a  $2 \times 2$  confusion matrix.

A confusion matrix is an  $N \times N$  table that summarizes how successful classification model predictions were. On one axis we take the label that the model predicted and on the other axis is the actual label. Here ‘N’ represents the number of classes. In our case  $N=2$ , so our confusion matrix will look like the table given below [31].

	Non-Cancerous (Predicted)	Cancerous (Predicted)
Non-Cancerous (Actual)	TN	FP
Cancerous (Actual)	FN	TP

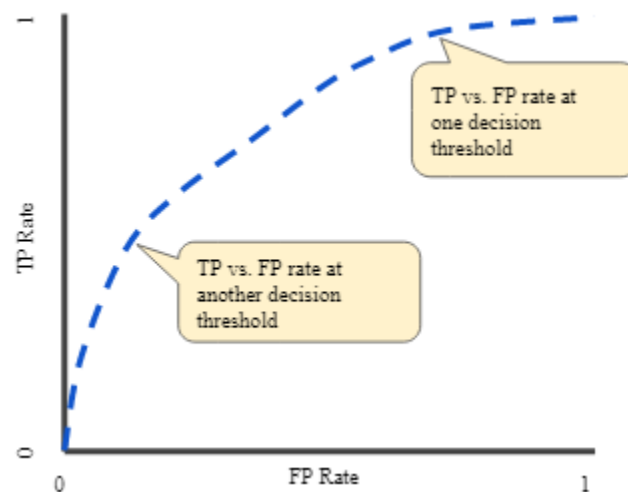
**Table 1 : Sample Confusion Matrix**

A true positive (TP) is an outcome that the model correctly predicts the positive class. Similarly, true negative (TN) is an outcome where the model correctly predicts negative class.

A false positive (FP) is an outcome that the model predicts as positive, but its actual label is negative. Similarly, false negative (FN) is an outcome where the model incorrectly predicts the negative class.

### 3.3.2 Receiver Operating Characteristic Curve (ROC Curve)

A ROC curve is the graph showing the performance of the classification model at all classification thresholds [31]. The below figure shows the typical ROC curve.



**Figure 6: Sample ROC curve [31]**

It is a plot of true positive rate (TPR) v/s false positive rate (FPR) at different classification thresholds.

This plot can be useful in evaluating models by comparing the area under the ROC curve. This area is commonly known as AUC (Area under the ROC curve).

AUC provides an aggregate measure of performance across all possible classification thresholds. Its value ranges from 0-1, 0 if all predictions made by the model are wrong and 1 if the model makes 100% right predictions.

### 3.3.3 Precision

Precision simply means “what proportion of positive identifications was actually correct?” or it is the measure of the ability of a system to make good predictions.

It is defined as  $P = TP / (TP + FP)$

Where TP is “number of true positives”, FP is “number of false positives” and P is precision.

The high precision value indicates high accuracy.

### **3.3.4 Recall or Sensitivity**

Recall tries to answer the question “what proportion of actual positives was identified correctly” or it calculates the proportion of correctly predicted positive images among all images that are known to be positive.

$$R = TP / (TP + FN)$$

Where TP is “number of true positives”, FP is “number of false positives” and R is Recall.

### **3.3.5 Specificity**

It is the ratio of how much correctly classified as negative to how much actually they were.

$$\text{Specificity} = TN / (FP + TN)$$

Where TN is “number of true negative”, FP is “number of false positive”

### **3.3.6 Accuracy**

Accuracy is one of the metrics for evaluating classification models. It is the fraction of the number of right predictions made by our model.

$$\text{Accuracy} = \text{Number of correct predictions} / \text{Total number of predictions.}$$

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Where TP=True positives, TN=True negatives, FP=False positive, and FN=False negative.

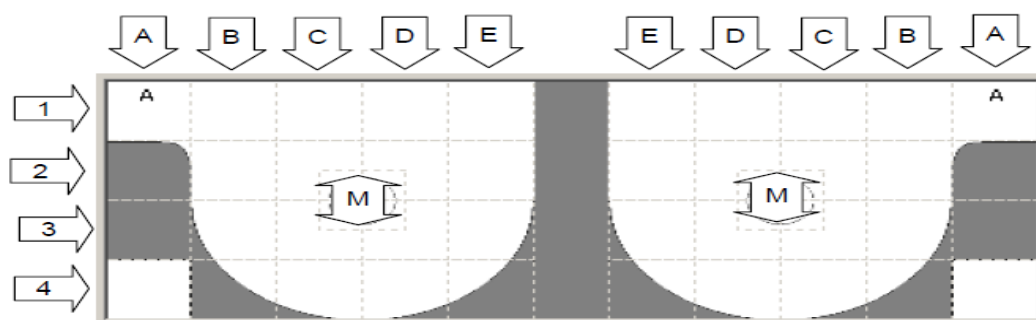
## CHAPTER 3 - MAMMOGRAM DATA COLLECTION AND PREPROCESSING

### 3.1 Data Collection:

In this research work, Mammogram images are collected from OPTIMUM Medical Imaging Database (OMI-DB). It collects NHS Breast Screening Program images from multiple breast screening centers across the UK. The data we collected is 161GB which contains 24079 images. we opted for this dataset because it contains images from multiple machines, training deep learning models on this dataset will give more accurate results [32].

The data contains two directories, one contains images in the form of DICOM image format and another contains metadata in the form of JSON. Each client directory contains one or more directories depending upon the number of studies for that client and each study ID contains data pertaining to four images i.e. ‘cranio-caudal (CC)’ and ‘Medio-Lateral oblique (MLO)’ view images for both breasts.

The data also contains information about ground truth regions of interest made by expert radiologists. It contains Lesion information which includes Lesion Position and classification. The lesion position is a code that is like a spreadsheet cell address as shown in the figure below. The columns are A to E and the rows are 1 to 4 with ‘M’ as a middle cell (nipple).



Examples:

RA1	Right axilla
LA1	Left axilla
RA4	Right bottom corner (position not specified)
LA4	Left bottom corner (position not specified)
RE1	Right upper inner cell
LE1	Left upper inner cell
RM	Right middle (areola/nipple)
LM	Left middle (areola/nipple)

**Figure 7: Mammogram segmentation [32]**

The classification information contains in the form of four labels as follows:

**N:** Normal

**M:** Malignant

**B:** Benign

**CI:** Interval Cancer

### 3.2 Data preparation and cleaning

This step is very important before image processing is executed because it will clean the metadata information and the images that are broken or corrupted.

Some of the problems encountered with the data are mentioned below:

1. We found some mismatch between images and their “Patient Label” column. In the below figure we have taken the example of four images from the PatientId ‘optm6414’, for this patient, Lesion of type ‘Malignant(M)’ was detected by the radiologist in the left breast but no lesion was detected in the right breast but ‘PatientLabel’ was same for all four images.so we have assigned ‘Normal(N)’ value for images corresponding to the right breast (Lateral = ‘R’).

ImgID	StudyID	PatientID	Lateral	LesionPositio	LesionPositionRight	PatientLab
1.2.826.0.1.3680043.9.3218.1.1.154173992.1314.1517665228179.50.0	1.2.826.0.1.3680043.9.3218.1.1.154173992.1314.1517665228179.48.0	optm6414	R	LD2	NULL	M
1.2.826.0.1.3680043.9.3218.1.1.154173992.1314.1517665228179.54.0	1.2.826.0.1.3680043.9.3218.1.1.154173992.1314.1517665228179.48.0	optm6414	L	LD2	NULL	M
1.2.826.0.1.3680043.9.3218.1.1.154173992.1314.1517665228179.58.0	1.2.826.0.1.3680043.9.3218.1.1.154173992.1314.1517665228179.48.0	optm6414	R	LD2	NULL	M
1.2.826.0.1.3680043.9.3218.1.1.154173992.1314.1517665228179.62.0	1.2.826.0.1.3680043.9.3218.1.1.154173992.1314.1517665228179.48.0	optm6414	L	LD2	NULL	M

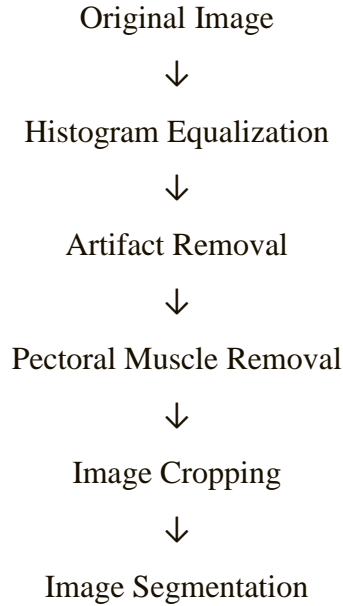
**Figure 8: Extracted Metadata**

2. Some of the images were found to be corrupted so all these images are removed from the dataset.

3. In our work, we focused on two-class classification only i.e. cancerous and non-cancerous. So, we have taken ‘Malignant’ and ‘Interval Cancer’ as cancerous while ‘Benign’ and ‘Normal’ as non-cancerous.

### 3.3 Mammogram Image Preprocessing

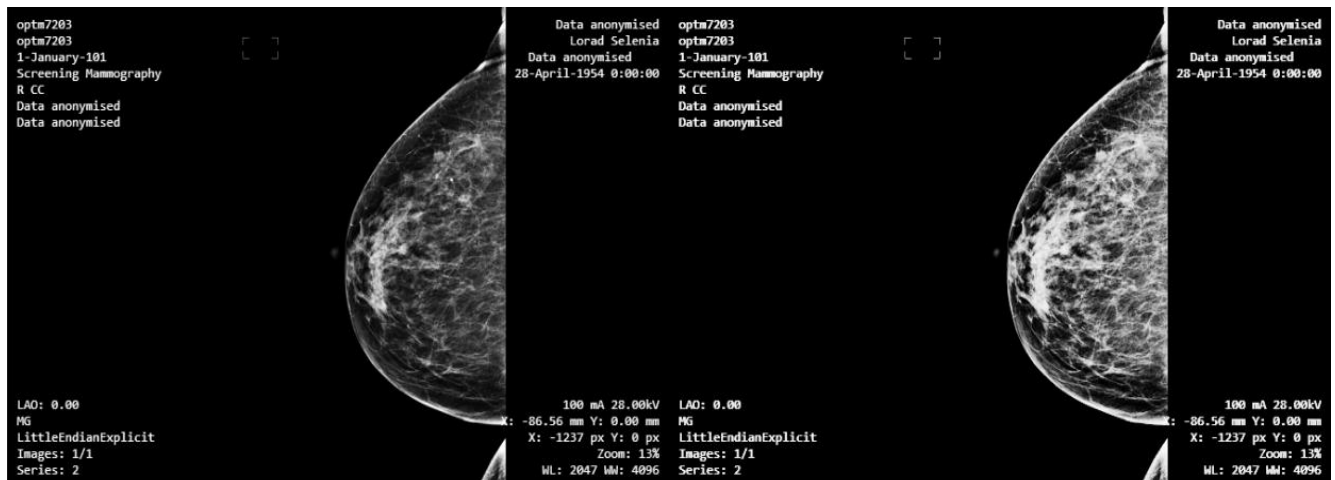
Mammogram Preprocessing is the most important part before we train our deep learning models. Preprocessing involves artifacts removal, enhance the quality of an image, pectoral muscle removal and segment the region of interest We follow the process in the following order:



### 3.3.1 Histogram Equalization

It is the technique used to improve contrast in an image. This can be achieved by effectively spreading out the most frequent intensity values. To apply the histogram equalization technique, we used the “CLAHE” algorithm discussed in the chapter of “Literature Review”.

The below figure shows the Mammogram image before and after applying histogram equalization function.

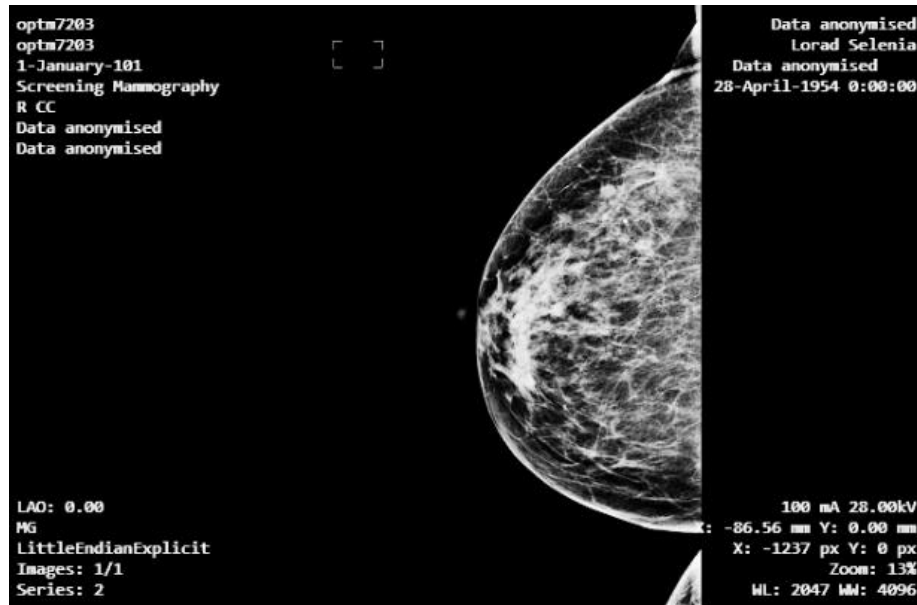


**Figure 9 : Histogram Equalization Mammogram Image**

On the left is the original Mammogram image and on the right is mammogram image obtained after histogram equalization.

### 3.1.2 Artifact Removal and Breast Contour Extraction

As we can see, the below mammogram image contains artifacts therefore first, we need to remove these artifacts before we jump to another part of preprocessing.

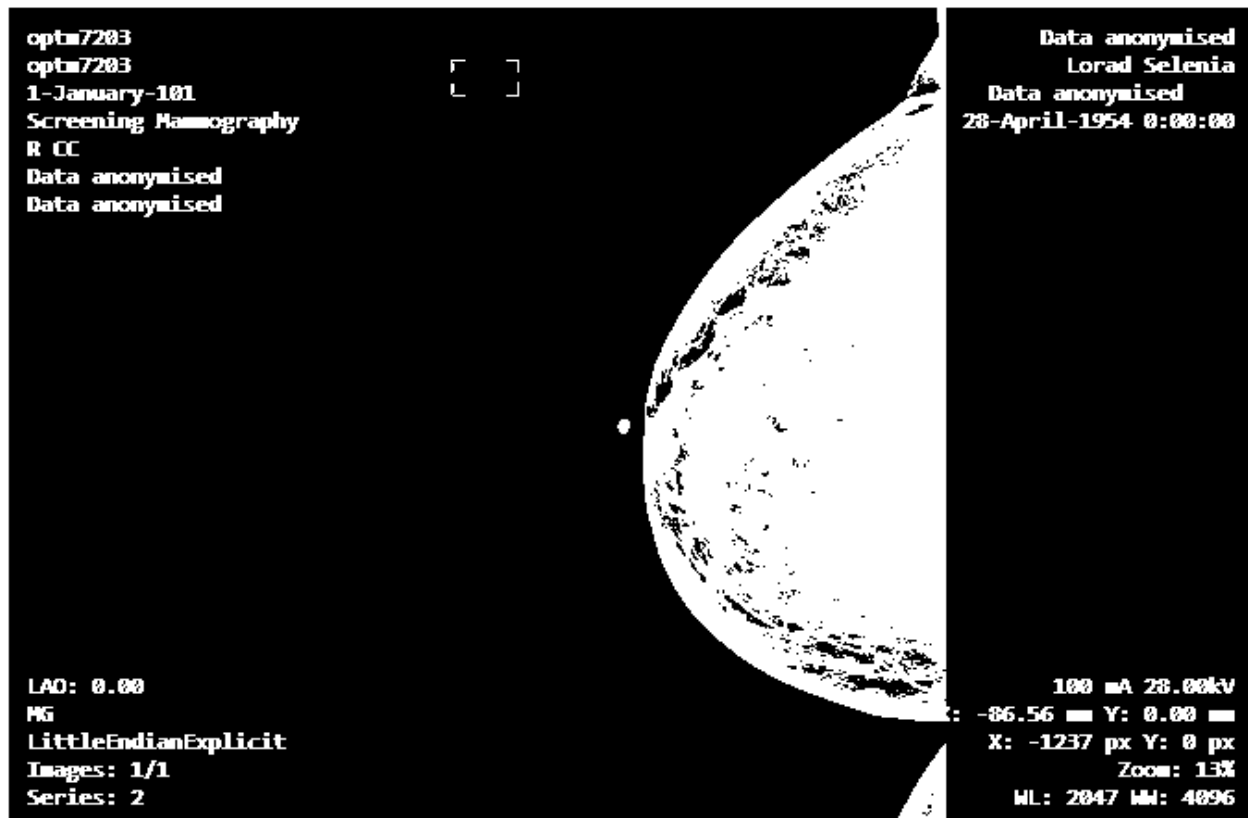


**Figure 10: Artifact Original Mammogram image**

As we can see in Figure 10 the breast part comprises of the largest part in the mammogram image, so we can utilize a simple technique to extract the foreground image which is having the maximum area. To do this first we have to convert an image to a binary image by applying thresholding algorithm. This algorithm works by first selecting threshold value(T) by parameter tuning(applying multiple threshold values on an image and selecting the best one), in our work it comes out to be 18, then all the pixels having the value more than 18 will be assigned pixel value as 255 (white) and all the values below that will be assigned as 0 (black).

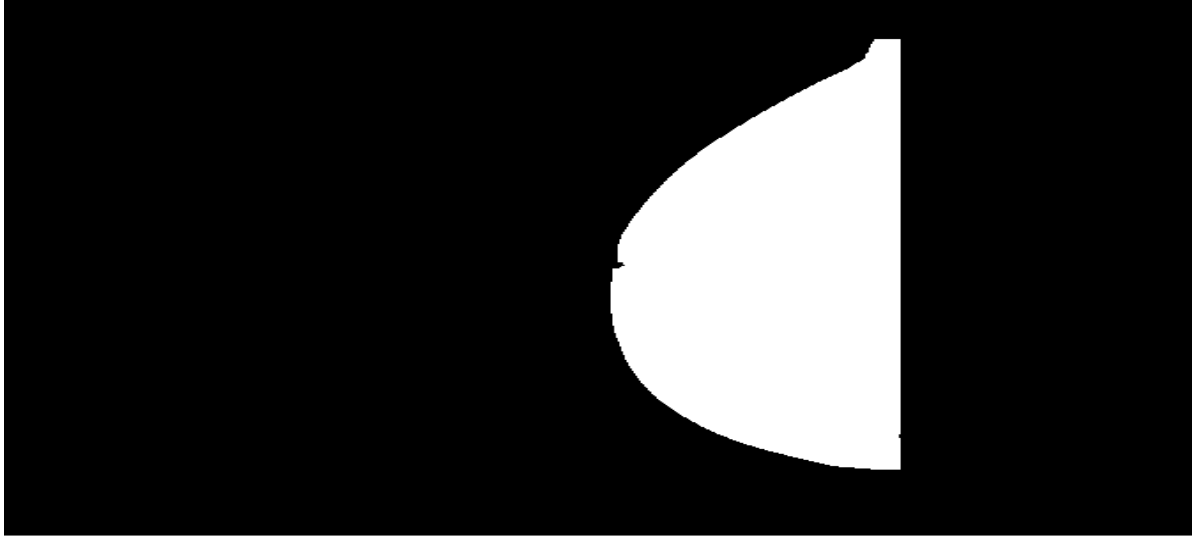
We applied this technique to the Figure 10 and obtained image which is shown in figure 11.





**Figure 11 : Mammogram Image After Thresholding**

After obtaining the binary image as shown in Figure 11, we filled the black color portion inside the breast contour with the white color pixel value i.e. 255 and used the “connected component algorithm”. This algorithm computes the connected component labeled image of the binary image and also produces the statistical output for each labeled image using 4 or 8 connectivity. Using statistical output, label for region having largest area is found and largest mask is extracted as shown in Figure 12.



**Figure 12: Artifact removed Mammogram Mask**

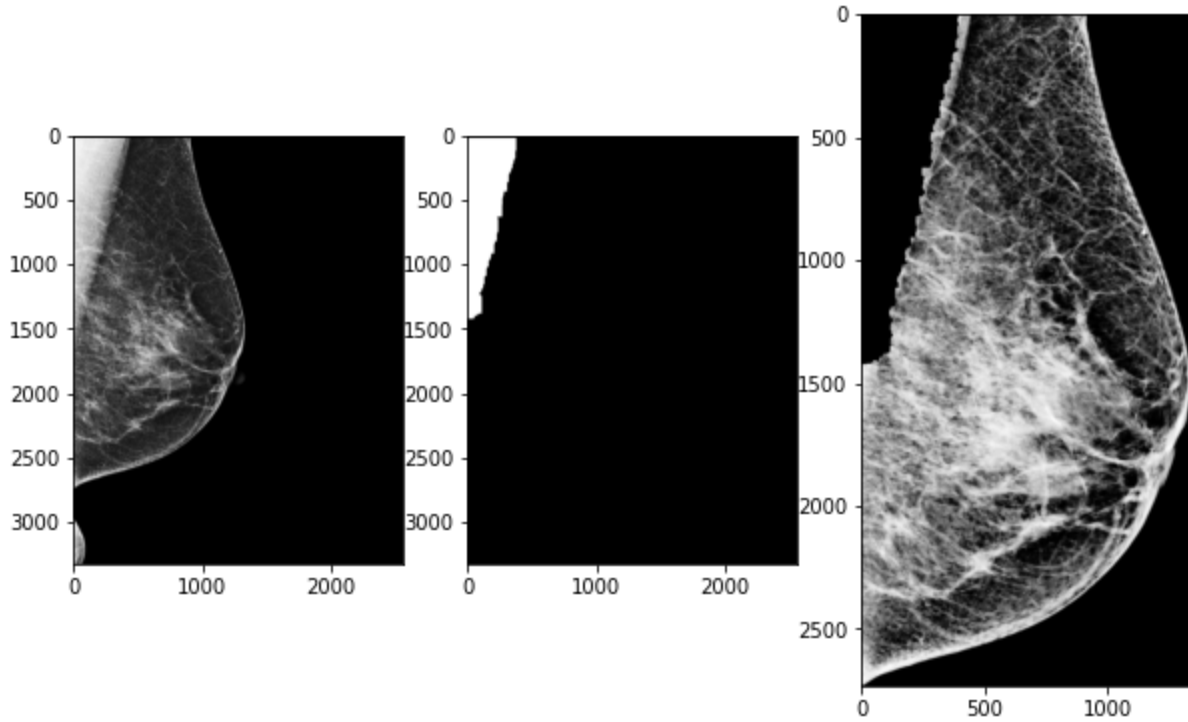
After obtaining the above mask we can then find bounding coordinates of the image which is nonblack i.e. pixel value  $> 0$  and use these coordinates to extract the bounding box of the original image.

### **3.1.3 Pectoral Muscle Removal**

Pectoral muscle is the muscle which connects the front of human chest with the bones of upper arm or shoulder and its removal is an essential part of preprocessing because malignant tissue is a bright dense part in the mammogram image and so is pectoral muscle. Therefore, if we don't remove pectoral muscle neural network models might get confused.

Since pectoral muscle is the brightest part of the image, so we can utilize this property to extract and remove it. To perform this, we can use thresholding algorithm and “connected component” algorithm as we used in the section “Artifact Removal and Brest Contour Extraction” but here we use a higher value of thresholding because we want to extract the brightest part.

Below is the figure which shows step by step process of removing pectoral muscle.



**Figure 13: Pectoral Muscle Removal:** On the left is the MLO view image, while in the middle is the pectoral muscle mask. On the right is the same image after pectoral muscle extraction and cropping.

### 3.4 Segmentation

The Mammogram images are of high resolution 3328\*4096 (the images we took for our study) and training neural networks on the high-resolution images is computationally expensive and if we resize the image then we might lose some information. Best way to come out of both limitations is to segment the images according to the techniques shown in Figure 7. Taking the reference from there, segmentation technique is applied on mammogram images to create segmented images dataset. Figure 14 shows the segmented mammogram image with total 16 segments. Each of these segments were segmented and dataset of 19674 images was created with size of each image was (224,224) pixels.

In the image below, Alphabets are taken as longitude and Numbers as latitude. In our process, we ignored 'A' segments because 'A' represents the axilla which is not there in the mammogram image.

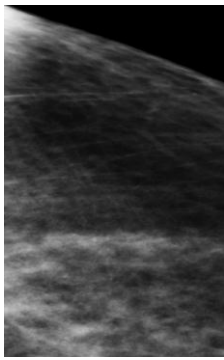
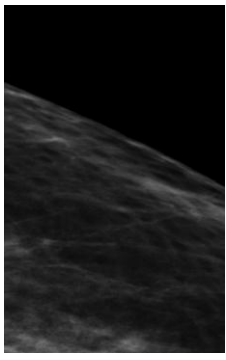
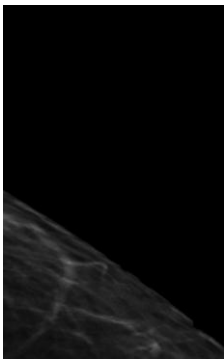
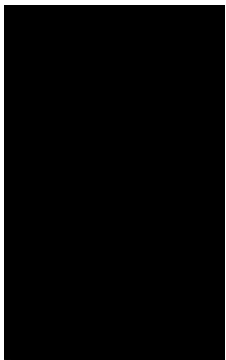
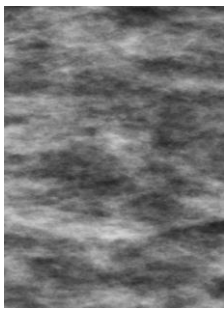
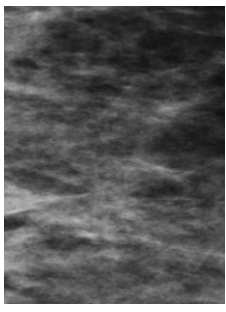
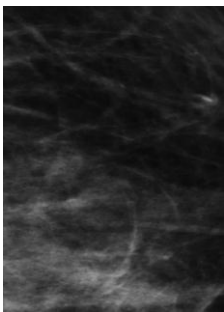
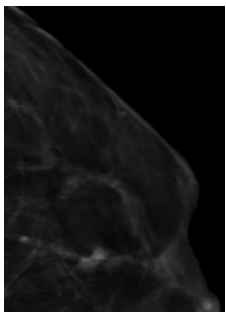
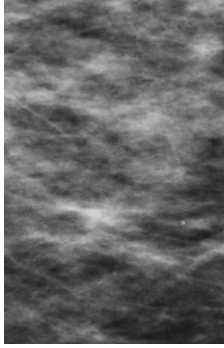
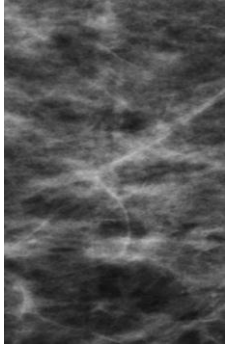
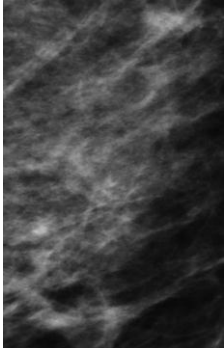
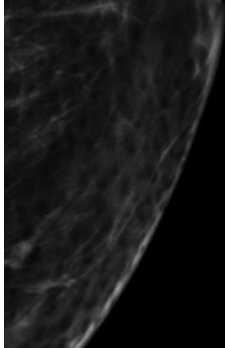
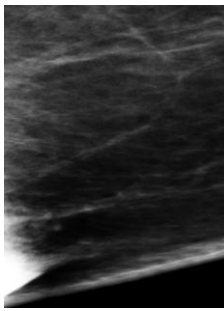
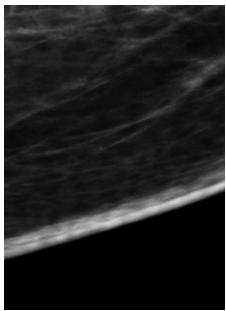
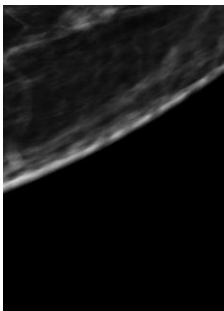

	B	C	D	E
1				
2				
3				
4				

Figure 14: Mammogram image segmentation example

### **3.5 Summary:**

In this chapter, we discussed data collection, data cleaning, and image preprocessing of the Mammogram images. We collected around 24079 images from OPTIMUM Medical Imaging Database (OMI-DB) along with metadata files. Once we had data, we followed the data cleaning process and removed corrupted or broken images before proceeding with the image processing part.

In the image processing section, we discussed various steps like histogram equalization, artifact removal, pectoral muscle removal, image cropping, and image segmentation. We talked about the importance of each part and how to apply them on mammogram image. In the next chapter, we will discuss the deep learning neural networks which we will train on images that we obtained after following the steps mentioned in this chapter.

## CHAPTER 4- ANALYSIS

Examination of results will give more understanding about deep learning and performance of it in the classification of cancerous Mammogram images. We trained VGG16 and RESNET50 neural network models on segmented and full images. We segmented the images as discussed in the chapter of “Mammogram data collection and preprocessing” and for full images we use images that contains full breast contour which are obtained in the preprocessing step just before the segmentation technique is applied.

We will analyze our training of deep learning models using graphs for AUC v/s Epoch and Categorical Cross entropy loss v/s Epoch. These graphs will give a clear view of how well our training was done. Furthermore, we will use the classification examination metrics referenced in the chapter “Implementation and Evaluation” to compare and evaluate our models.

### 4.1 Experiments

#### 4.1.2. Segmented Images

16 images from the whole image were segmented as we had discussed in the chapter “Mammogram data collection and preprocessing” and models were trained using VGG16 and RESNET50 architecture with the following 6 combinations.

##### **Image size- (224,224) Pixels**

VGG16	(Random initialization of neural network layers weights)
VGG16+Transfer learning	(Weights initialized from models trained on ImageNet dataset)
RESNET50	(Random initialization of neural networks layers weights)
RESNET50+Transfer learning	(Weights initialized from models trained on ImageNet dataset)

##### **Image size- (50,50) Pixels**

VGG16	(Random initialization of neural networks layers weights)
RESNET50	(Random initialization of neural networks layers weights)

Note: (224,224) and (50,50) Pixels, signify the size of each segment that we segmented from full image

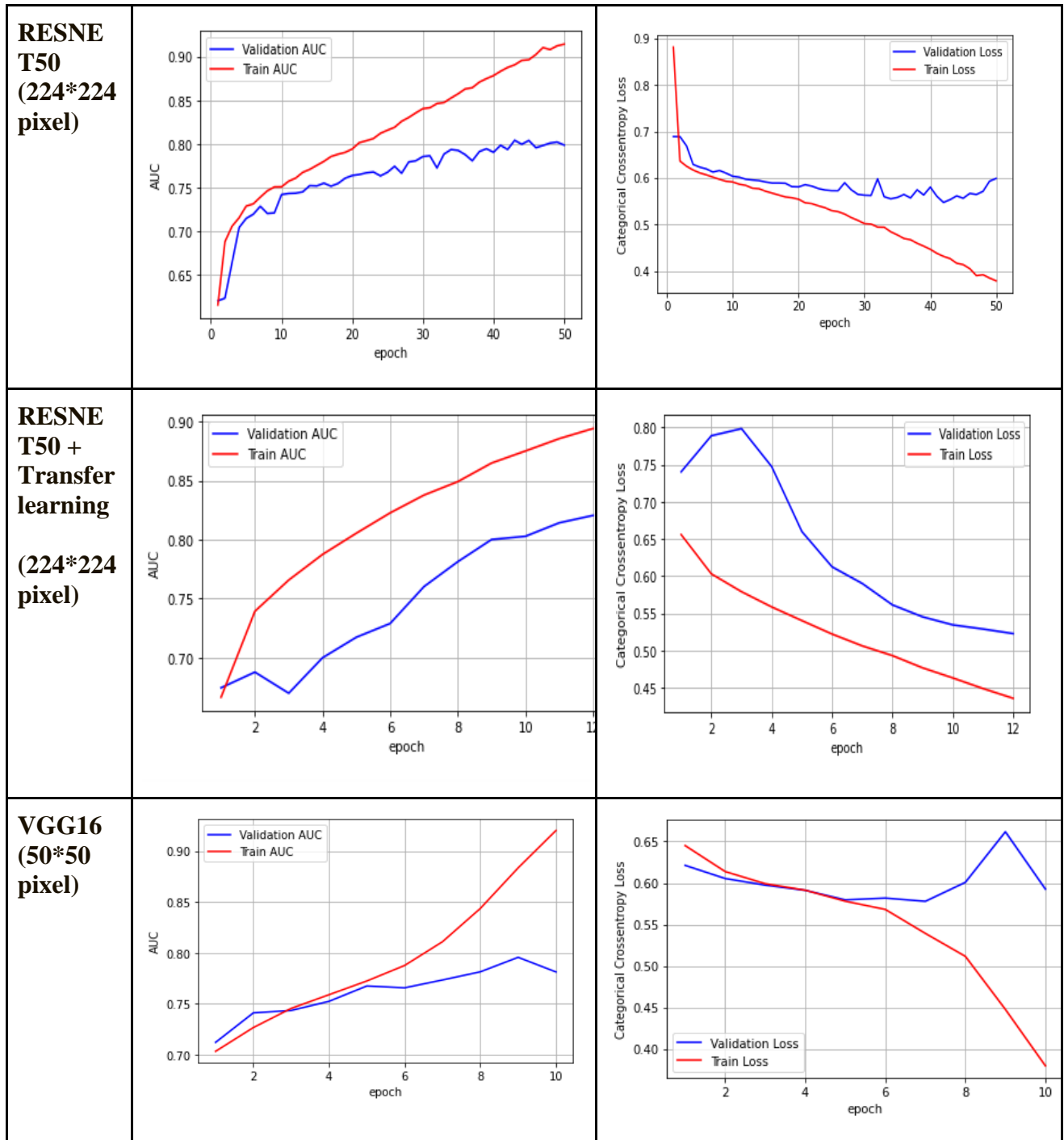
We trained neural network models on two different image sizes i.e. (224,224) and (50,50), and used different neural network weight initialization to see the effect of image size on neural network performance as well as that of transfer learning from models train on ‘ImageNet’ data set.

For experimentation, we took total of 19674 segments, out of these 10000 are non-cancerous (negative) and 9674 are cancerous (positive) segments.

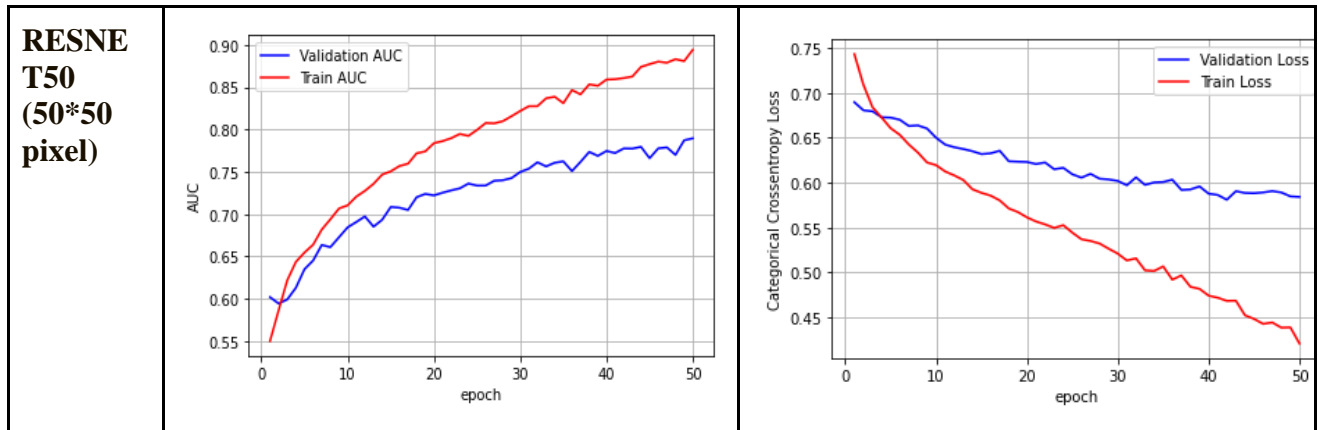
### Training AUC, Loss V/S Epoch graph

All six models which we discussed in the last section are trained with multiple epochs and the below graph shows the trend of AUC and Cross-Entropy loss with each epoch. Analyzing these graphs will give a representation of the training of neural network models.

	Train AUC, Validation AUC/Epoch graph	Train loss, Validation loss/Epoch graph
<b>VGG16 (224*224 pixel)</b>		
<b>VGG16+ Transfer learning (224*224 pixel)</b>		





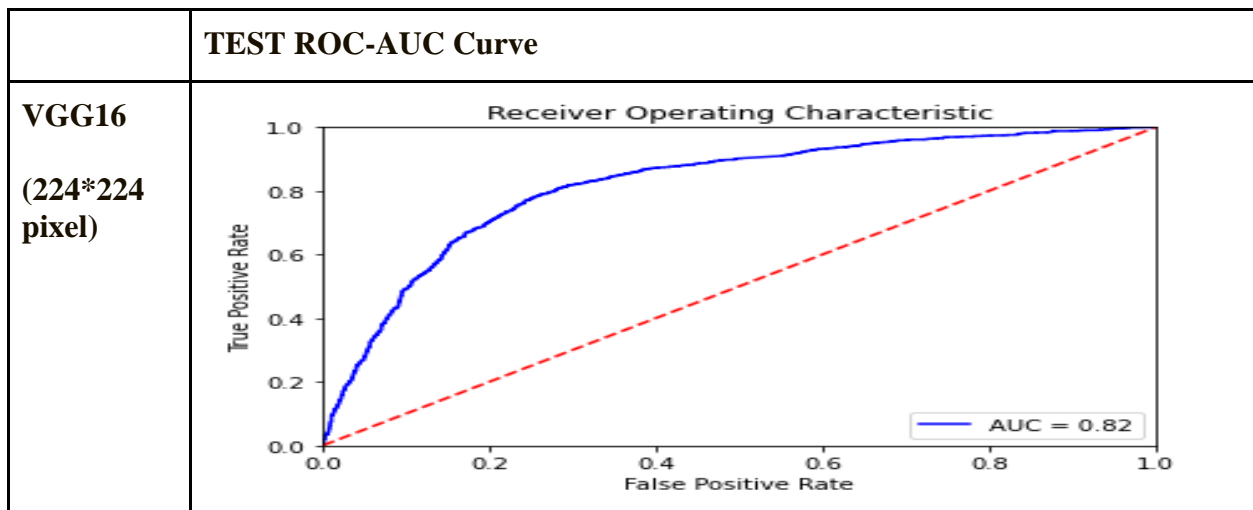


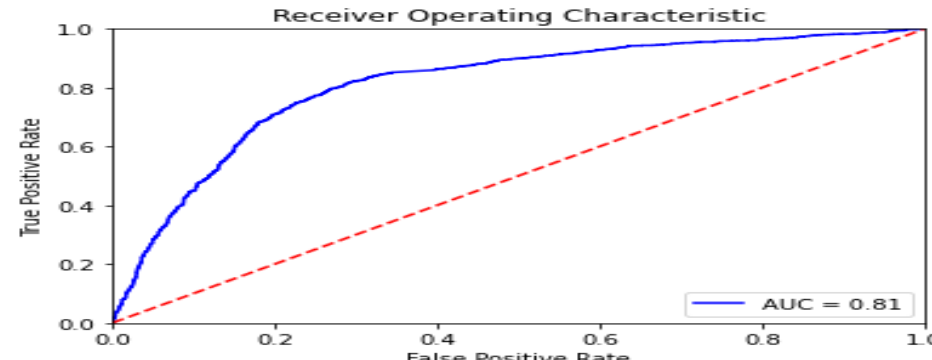
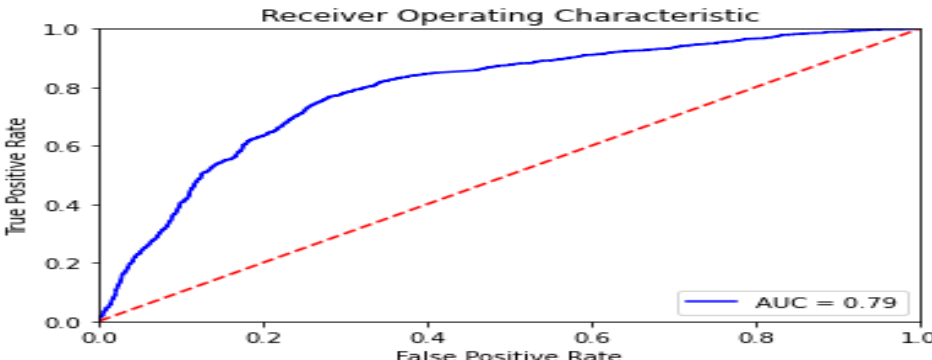
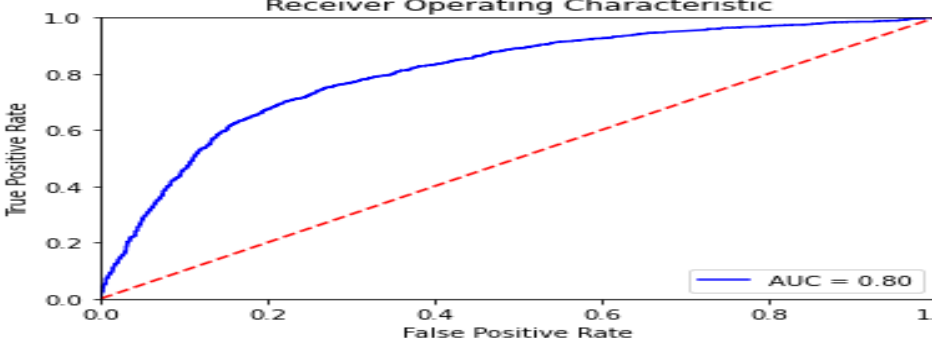
**Table 2: Segmented images AUC, Loss v/s Epoch**

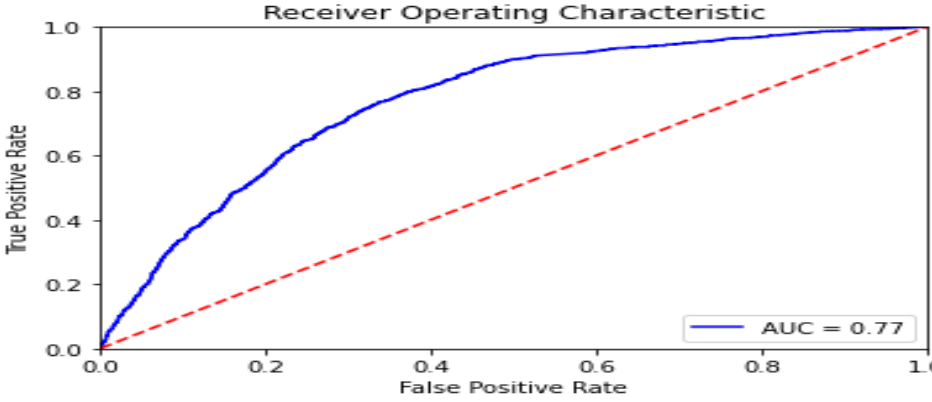
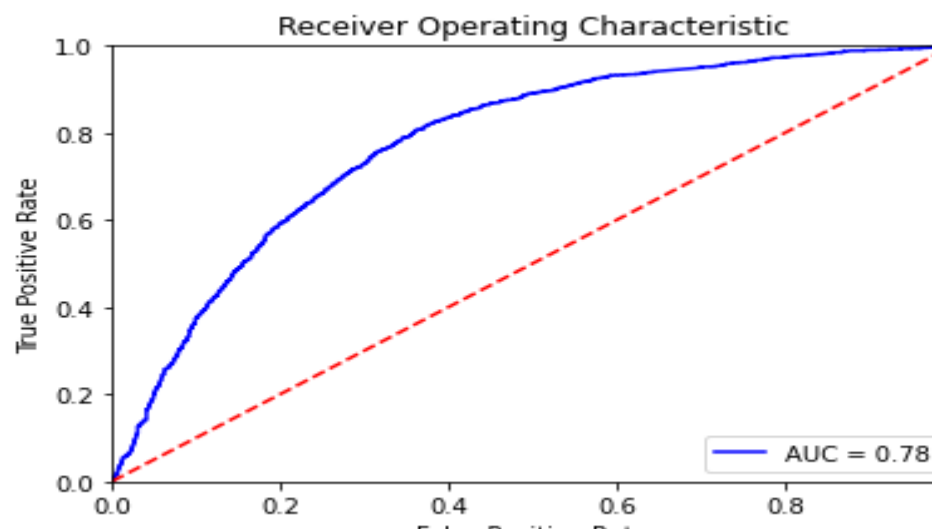
**Discussion:** From the above graphs saw that the first two VGG16 models, VGG16 (224\*224 pixel) and VGG + Transfer learning(224\*224 pixel)) started overfitting after 10 epoch of training as the validation AUC curve started flattening and validation loss started increasing after 10 epochs while the third one(VGG16(50\*50 pixel)) just started overfitting after the 9th epoch. For RESNET50 (224\*224 pixel) model, training for 50 epochs was perfect as after that only loss started increasing and trend RESNET50+Transfer learning (224\* 224 pixels) model. From these results, we conclude that the size of the epoch which we took for our experimentation was good, if we have trained these models over more number of epochs than the epoch size we took in our experimentation, we would have got an overfitted model.

### TEST ROC-AUC Graph

The below table shows the ROC-AUC curve generated by all six models on the segmented images Test dataset. AUC value for each graph is given in below-left corner of each graph.



<p><b>VGG16+</b> <b>Transfer learning</b>  (224*224 pixel)</p>	 <p>Receiver Operating Characteristic</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>AUC = 0.81</p>
<p><b>RESNET50</b>  (224*224 pixel)</p>	 <p>Receiver Operating Characteristic</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>AUC = 0.79</p>
<p><b>RESNET50+</b> <b>Transfer Learning</b>  (224*224 pixel)</p>	 <p>Receiver Operating Characteristic</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>AUC = 0.80</p>

<b>VGG16</b> <b>(50*50</b> <b>pixel)</b>	 <p>Receiver Operating Characteristic</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>AUC = 0.77</p>
<b>RESNET50</b> <b>(50*50</b> <b>pixel)</b>	 <p>Receiver Operating Characteristic</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>AUC = 0.78</p>

**Table 3 : Segmented Images Test ROC-AUC Curve**

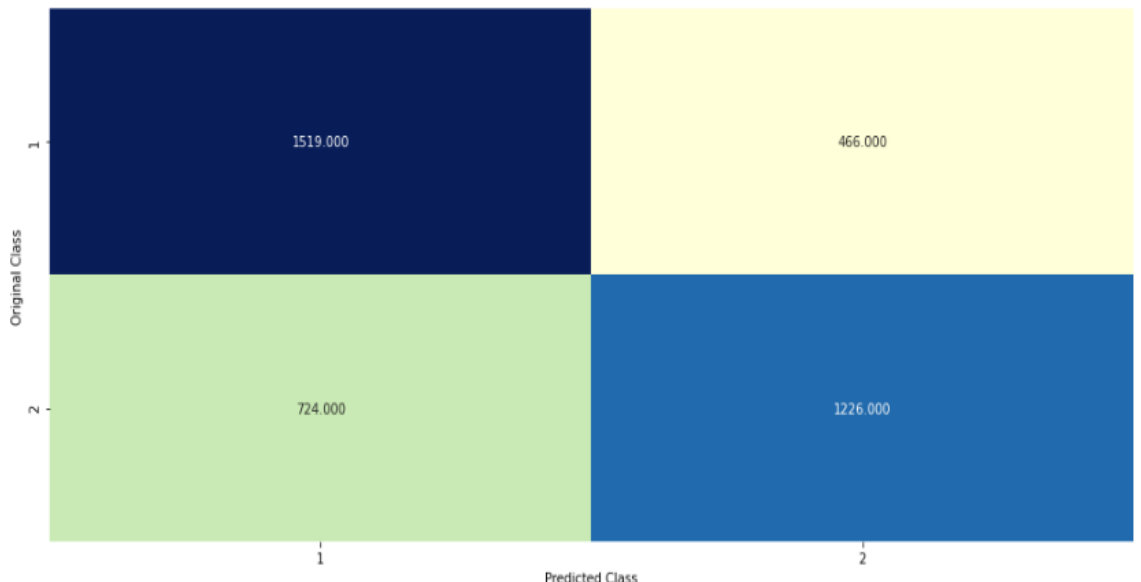
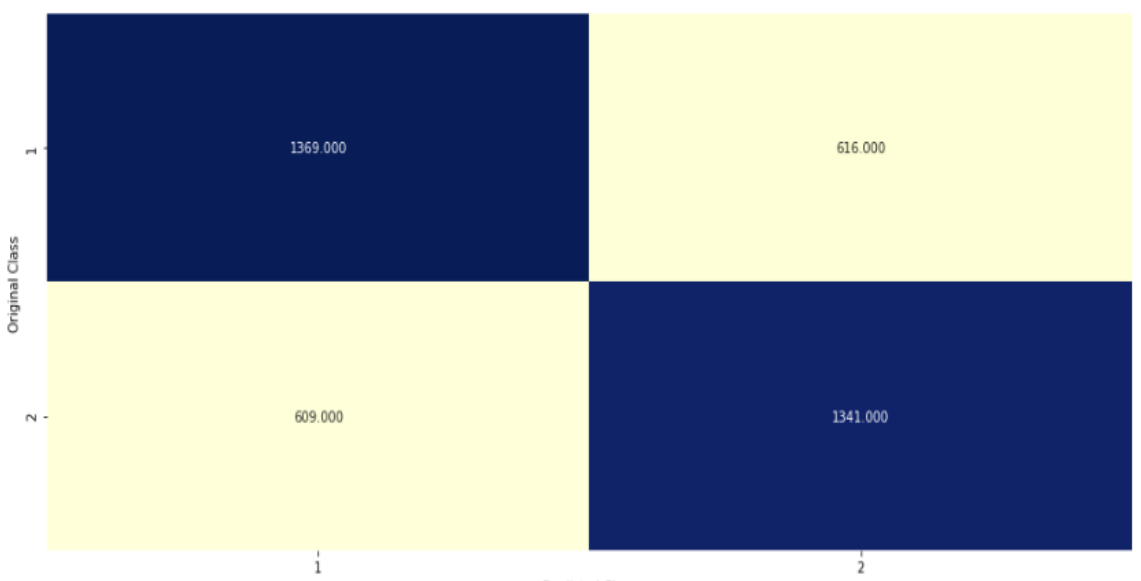
**Discussion:** AUC value makes it easy to compare the models making better predictions, from the above figures we can see that VGG16 (224\*224 pixel) performed best with an AUC value of 0.82 and VGG16 (50\*50 pixel) model performed worst with an AUC value of 0.77 but the difference in AUC values of some models like RESNET50 (224\*224 pixel) and RESNET50+ Transfer Learning (224\*224 pixel) is just 0.01, so for reaching to the conclusion that which model performed better, we have to look for other metrics like confusion matrix.

## Confusion Matrix

The below table shows the confusion matrix from predictions made on the Test dataset by models trained on the segmented image dataset. In the below figures ‘1’ signify non-cancerous and ‘2’ signify cancerous.

	Confusion Matrix									
<b>VGG16 (224*224 pixel)</b>	<div><div>----- Confusion matrix -----</div><table><tr><th>Original Class \ Predicted Class</th><th>1</th><th>2</th></tr><tr><th>1</th><td>1353.000</td><td>632.000</td></tr><tr><th>2</th><td>334.000</td><td>1616.000</td></tr></table></div>	Original Class \ Predicted Class	1	2	1	1353.000	632.000	2	334.000	1616.000
Original Class \ Predicted Class	1	2								
1	1353.000	632.000								
2	334.000	1616.000								
<b>VGG16+ Transfer Learning (224*224 pixel)</b>	<div><div>----- Confusion matrix -----</div><table><tr><th>Original Class \ Predicted Class</th><th>1</th><th>2</th></tr><tr><th>1</th><td>1601.000</td><td>384.000</td></tr><tr><th>2</th><td>592.000</td><td>1358.000</td></tr></table></div>	Original Class \ Predicted Class	1	2	1	1601.000	384.000	2	592.000	1358.000
Original Class \ Predicted Class	1	2								
1	1601.000	384.000								
2	592.000	1358.000								

<div>RESNET 50</div> <div>(224*224 pixel)</div>	<div>----- Confusion matrix -----</div> <div><div><div>1</div><div>2</div></div><div><div>1</div><div>2</div></div></div> <div>Original Class</div> <div>Predicted Class</div> <table><tr><td>1</td><td>1571.000</td><td>414.000</td></tr><tr><td>2</td><td>691.000</td><td>1259.000</td></tr></table>	1	1571.000	414.000	2	691.000	1259.000
1	1571.000	414.000					
2	691.000	1259.000					
<div>Resnet50 +transfer Learning</div> <div>(224*224 pixel)</div>	<div>----- Confusion matrix -----</div> <div><div><div>1</div><div>2</div></div><div><div>1</div><div>2</div></div></div> <div>Original Class</div> <div>Predicted Class</div> <table><tr><td>1</td><td>1479.000</td><td>506.000</td></tr><tr><td>2</td><td>527.000</td><td>1423.000</td></tr></table>	1	1479.000	506.000	2	527.000	1423.000
1	1479.000	506.000					
2	527.000	1423.000					

<b>VGG16</b> <b>(50*50</b> <b>pixel)</b>	<div>----- Confusion matrix -----</div>  <table><tr><th>Original Class \ Predicted Class</th><th>1</th><th>2</th></tr><tr><th>1</th><td>1519.000</td><td>466.000</td></tr><tr><th>2</th><td>724.000</td><td>1226.000</td></tr></table>	Original Class \ Predicted Class	1	2	1	1519.000	466.000	2	724.000	1226.000
Original Class \ Predicted Class	1	2								
1	1519.000	466.000								
2	724.000	1226.000								
<b>RESNET</b> <b>50</b> <b>(50*50</b> <b>pixel)</b>	<div>----- Confusion matrix -----</div>  <table><tr><th>Original Class \ Predicted Class</th><th>1</th><th>2</th></tr><tr><th>1</th><td>1369.000</td><td>616.000</td></tr><tr><th>2</th><td>609.000</td><td>1341.000</td></tr></table>	Original Class \ Predicted Class	1	2	1	1369.000	616.000	2	609.000	1341.000
Original Class \ Predicted Class	1	2								
1	1369.000	616.000								
2	609.000	1341.000								

**Table 4: Segmented Images Confusion Matrix**

**Discussion:** From the above confusion matrix, we can see that for VGG16 model “VGG16( 224 \* 224 pixels)” gives better prediction results for class 2 (cancerous) while the “VGG16+Transfer Learning (224\*224 pixel)” and “VGG16(50\*50 pixel)” gives better results for class1 (non-cancerous).

For RESNET50 models, “RESNET50(224\*224 pixel)” gives better prediction results for class 1 while “RESNET50+Transfer Learning (224\*224 pixel))” and “RESNET50(50\*50 pixel)” give balanced results between the two classes which show that the models are not biased towards any particular class.

In cancer detection Cost of False Negatives (cancerous but predicted as non-cancerous) is high so we can say that model which gives least number of false negatives performed better than others, from the above graph results we can note that among all the models “VGG16 (224 \* 224 pixels)” predicted the least number of false negatives i.e. 332 so we can conclude that this model outperformed other models in detecting cancer.

### Classification Metrics

The table below summarizes AUC, Precision, Recall, Accuracy, Specificity for all six models trained on segmented images.

	<b>TEST - AUC</b>	<b>Precision</b>	<b>Recall/Sensitivity</b>	<b>Accuracy</b>	<b>Specificity</b>
<b>VGG16(224*224 pixel)</b>	0.82	0.71	0.82	75.4%	0.68
<b>VGG16+Transfer learning (224*224 pixel)</b>	0.81	0.780	0.696	75.1%	0.80
<b>RESNET50(224*224 pixel)</b>	0.79	0.753	0.646	71.9%	0.791
<b>RESNET50+Transfer learning (224*224 pixel)</b>	0.80	0.738	0.730	73.7%	0.745
<b>VGG16 (50*50 pixel)</b>	0.77	0.725	0.629	69.7%	0.765

<b>RESNET50 (50*50 pixel)</b>	0.78	0.685	0.688	68.8%	0.689
-----------------------------------	------	-------	-------	-------	-------

**Table 5 : Segmented Images Classification Metrics**

**Discussion:** From the above table, we can see that Test-AUC and accuracy for all models other than “VGG16(50\*50 pixel)” and “RESNET50(50\*50 pixel)” are very close to each other but Precision, Recall/Sensitivity, and Specificity differ.

”VGG16(224\*224 pixel)” gives high Recall but low Specificity and Precision, this means it gives good result of true positives (originally cancerous and predicted cancerous) but also gives some false positives (originally non-cancerous but predicted cancerous) which decrease the value for Specificity and Precision. “VGG16+Transfer learning (224\*224 pixel)” and “VGG16(50\*50 pixel)” gives high specificity and Precision but low recall that is because these models give a large number of false negatives (Originally cancerous but predicted non-cancerous).

“RESNET50(224\*224 pixel)” give high Precision and Specificity compared to Recall but in “RESNET50+Transfer learning (224\*224 pixel)”, all three values Precision, Recall and Specificity are balanced.

From the above classification metric table, we can infer some of the findings, that are mentioned below:

1. In the Mammogram classification problem that we are solving, high sensitivity matters the most so we can say that the” VGG16(224\*224 pixel)” model with the sensitivity value of 0.82 performs best than other models.
2. Models trained using transfer learning from models trained on 'ImageNet' dataset doesn't perform better in mammogram classification problem. This can be confirmed from the results that” VGG16(224\*224 pixel)” trained using random initializer performed best than other models like “VGG16+Transfer learning (224\*224 pixel)” which are trained using transfer learning.
- 3.VGG16(50\*50 pixel) and RESNET50(50\*50) have lower AUC and Accuracy than their corresponding models trained on (224\*224) pixels, this shows that image size does affect neural network deep learning results.



### 4.1.3. Full Images

Getting a large dataset with marked annotations by radiologists is difficult to get so training on full images and analyzing the test results will help to know the effectiveness of deep learning neural networks on full images without marked annotations. Apart from this research study, analysis of results for models trained on multiple datasets which contain images from either both views i.e. ‘CC’ and ‘MLO’ or ‘Only CC’ or ‘Only MLO’ will help in knowing the effect of views in detecting cancer using deep learning techniques.

In this section total 3 models are trained using random initialization of weights with following combinations:

1. VGG16      (‘CC’+’MLO’)
2. VGG16      (‘Only CC’)
3. VGG16      (‘Only MLO’)

VGG16 (‘CC’+’MLO’) model is trained on dataset that contains total 17022 images from both views. VGG16 (‘Only CC’) model is trained on dataset that contains total 8266 images from CC view only. VGG16 (‘Only MLO’) models is trained on dataset that contains total 8048 images from only MLO view.

Note: All experiments for full images are performed on image size of (224,224).

## Training AUC, Loss V/S Epoch graph

All models for full images are trained with multiple epochs and the below graph shows the trend of AUC and Cross-Entropy loss with each epoch. Analyzing these graphs will give a representation of the training of neural network models.

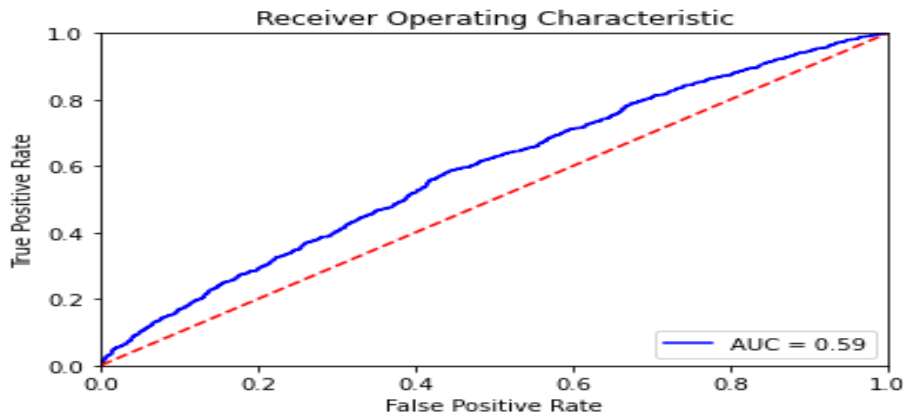
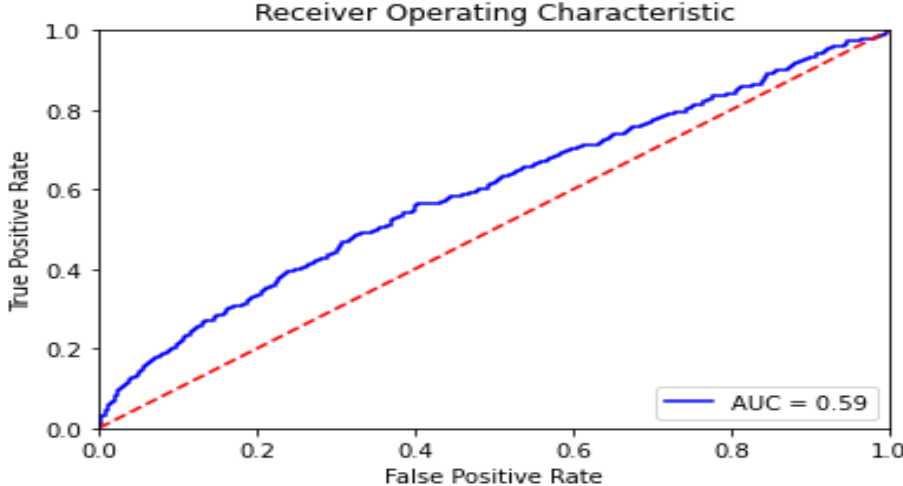
	Train AUC, Validation AUC/epoch graph	Train loss, Validation loss/epoch graph																																																																														
<b>VGG16 (CC+MLO )</b>	<table border="1"> <caption>VGG16 (CC+MLO) AUC Data</caption> <thead> <tr> <th>epoch</th> <th>Train AUC</th> <th>Validation AUC</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.55</td><td>0.57</td></tr> <tr><td>2</td><td>0.57</td><td>0.59</td></tr> <tr><td>3</td><td>0.58</td><td>0.60</td></tr> <tr><td>4</td><td>0.59</td><td>0.61</td></tr> <tr><td>5</td><td>0.60</td><td>0.60</td></tr> <tr><td>6</td><td>0.61</td><td>0.61</td></tr> <tr><td>7</td><td>0.62</td><td>0.61</td></tr> <tr><td>8</td><td>0.63</td><td>0.61</td></tr> <tr><td>9</td><td>0.64</td><td>0.62</td></tr> <tr><td>10</td><td>0.66</td><td>0.62</td></tr> </tbody> </table>	epoch	Train AUC	Validation AUC	1	0.55	0.57	2	0.57	0.59	3	0.58	0.60	4	0.59	0.61	5	0.60	0.60	6	0.61	0.61	7	0.62	0.61	8	0.63	0.61	9	0.64	0.62	10	0.66	0.62	<table border="1"> <caption>VGG16 (CC+MLO) Loss Data</caption> <thead> <tr> <th>epoch</th> <th>Train Loss</th> <th>Validation Loss</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.69</td><td>0.68</td></tr> <tr><td>2</td><td>0.68</td><td>0.70</td></tr> <tr><td>3</td><td>0.68</td><td>0.68</td></tr> <tr><td>4</td><td>0.68</td><td>0.68</td></tr> <tr><td>5</td><td>0.68</td><td>0.68</td></tr> <tr><td>6</td><td>0.68</td><td>0.68</td></tr> <tr><td>7</td><td>0.68</td><td>0.68</td></tr> <tr><td>8</td><td>0.67</td><td>0.68</td></tr> <tr><td>9</td><td>0.66</td><td>0.67</td></tr> <tr><td>10</td><td>0.65</td><td>0.68</td></tr> </tbody> </table>	epoch	Train Loss	Validation Loss	1	0.69	0.68	2	0.68	0.70	3	0.68	0.68	4	0.68	0.68	5	0.68	0.68	6	0.68	0.68	7	0.68	0.68	8	0.67	0.68	9	0.66	0.67	10	0.65	0.68												
epoch	Train AUC	Validation AUC																																																																														
1	0.55	0.57																																																																														
2	0.57	0.59																																																																														
3	0.58	0.60																																																																														
4	0.59	0.61																																																																														
5	0.60	0.60																																																																														
6	0.61	0.61																																																																														
7	0.62	0.61																																																																														
8	0.63	0.61																																																																														
9	0.64	0.62																																																																														
10	0.66	0.62																																																																														
epoch	Train Loss	Validation Loss																																																																														
1	0.69	0.68																																																																														
2	0.68	0.70																																																																														
3	0.68	0.68																																																																														
4	0.68	0.68																																																																														
5	0.68	0.68																																																																														
6	0.68	0.68																																																																														
7	0.68	0.68																																																																														
8	0.67	0.68																																																																														
9	0.66	0.67																																																																														
10	0.65	0.68																																																																														
<b>VGG (ONLY CC)</b>	<table border="1"> <caption>VGG (ONLY CC) AUC Data</caption> <thead> <tr> <th>epoch</th> <th>Train AUC</th> <th>Validation AUC</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.55</td><td>0.55</td></tr> <tr><td>2</td><td>0.58</td><td>0.56</td></tr> <tr><td>3</td><td>0.60</td><td>0.58</td></tr> <tr><td>4</td><td>0.61</td><td>0.58</td></tr> <tr><td>5</td><td>0.62</td><td>0.58</td></tr> <tr><td>6</td><td>0.63</td><td>0.60</td></tr> <tr><td>7</td><td>0.65</td><td>0.59</td></tr> <tr><td>8</td><td>0.66</td><td>0.61</td></tr> <tr><td>9</td><td>0.68</td><td>0.61</td></tr> <tr><td>10</td><td>0.70</td><td>0.60</td></tr> <tr><td>11</td><td>0.72</td><td>0.60</td></tr> <tr><td>12</td><td>0.75</td><td>0.60</td></tr> </tbody> </table>	epoch	Train AUC	Validation AUC	1	0.55	0.55	2	0.58	0.56	3	0.60	0.58	4	0.61	0.58	5	0.62	0.58	6	0.63	0.60	7	0.65	0.59	8	0.66	0.61	9	0.68	0.61	10	0.70	0.60	11	0.72	0.60	12	0.75	0.60	<table border="1"> <caption>VGG (ONLY CC) Loss Data</caption> <thead> <tr> <th>epoch</th> <th>Train Loss</th> <th>Validation Loss</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.69</td><td>0.69</td></tr> <tr><td>2</td><td>0.68</td><td>0.69</td></tr> <tr><td>3</td><td>0.68</td><td>0.68</td></tr> <tr><td>4</td><td>0.67</td><td>0.68</td></tr> <tr><td>5</td><td>0.67</td><td>0.68</td></tr> <tr><td>6</td><td>0.67</td><td>0.68</td></tr> <tr><td>7</td><td>0.66</td><td>0.68</td></tr> <tr><td>8</td><td>0.66</td><td>0.68</td></tr> <tr><td>9</td><td>0.64</td><td>0.68</td></tr> <tr><td>10</td><td>0.63</td><td>0.68</td></tr> <tr><td>11</td><td>0.61</td><td>0.69</td></tr> <tr><td>12</td><td>0.59</td><td>0.69</td></tr> </tbody> </table>	epoch	Train Loss	Validation Loss	1	0.69	0.69	2	0.68	0.69	3	0.68	0.68	4	0.67	0.68	5	0.67	0.68	6	0.67	0.68	7	0.66	0.68	8	0.66	0.68	9	0.64	0.68	10	0.63	0.68	11	0.61	0.69	12	0.59	0.69
epoch	Train AUC	Validation AUC																																																																														
1	0.55	0.55																																																																														
2	0.58	0.56																																																																														
3	0.60	0.58																																																																														
4	0.61	0.58																																																																														
5	0.62	0.58																																																																														
6	0.63	0.60																																																																														
7	0.65	0.59																																																																														
8	0.66	0.61																																																																														
9	0.68	0.61																																																																														
10	0.70	0.60																																																																														
11	0.72	0.60																																																																														
12	0.75	0.60																																																																														
epoch	Train Loss	Validation Loss																																																																														
1	0.69	0.69																																																																														
2	0.68	0.69																																																																														
3	0.68	0.68																																																																														
4	0.67	0.68																																																																														
5	0.67	0.68																																																																														
6	0.67	0.68																																																																														
7	0.66	0.68																																																																														
8	0.66	0.68																																																																														
9	0.64	0.68																																																																														
10	0.63	0.68																																																																														
11	0.61	0.69																																																																														
12	0.59	0.69																																																																														
<b>VGG (Only MLO)</b>	<table border="1"> <caption>VGG (Only MLO) AUC Data</caption> <thead> <tr> <th>epoch</th> <th>Train AUC</th> <th>Validation AUC</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.56</td><td>0.57</td></tr> <tr><td>2</td><td>0.57</td><td>0.57</td></tr> <tr><td>3</td><td>0.58</td><td>0.56</td></tr> <tr><td>4</td><td>0.59</td><td>0.58</td></tr> <tr><td>5</td><td>0.60</td><td>0.57</td></tr> <tr><td>6</td><td>0.61</td><td>0.58</td></tr> <tr><td>7</td><td>0.63</td><td>0.57</td></tr> <tr><td>8</td><td>0.64</td><td>0.57</td></tr> <tr><td>9</td><td>0.65</td><td>0.57</td></tr> <tr><td>10</td><td>0.67</td><td>0.56</td></tr> <tr><td>11</td><td>0.69</td><td>0.58</td></tr> <tr><td>12</td><td>0.70</td><td>0.59</td></tr> </tbody> </table>	epoch	Train AUC	Validation AUC	1	0.56	0.57	2	0.57	0.57	3	0.58	0.56	4	0.59	0.58	5	0.60	0.57	6	0.61	0.58	7	0.63	0.57	8	0.64	0.57	9	0.65	0.57	10	0.67	0.56	11	0.69	0.58	12	0.70	0.59	<table border="1"> <caption>VGG (Only MLO) Loss Data</caption> <thead> <tr> <th>epoch</th> <th>Train Loss</th> <th>Validation Loss</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.69</td><td>0.69</td></tr> <tr><td>2</td><td>0.68</td><td>0.69</td></tr> <tr><td>3</td><td>0.68</td><td>0.70</td></tr> <tr><td>4</td><td>0.68</td><td>0.69</td></tr> <tr><td>5</td><td>0.68</td><td>0.68</td></tr> <tr><td>6</td><td>0.67</td><td>0.69</td></tr> <tr><td>7</td><td>0.67</td><td>0.68</td></tr> <tr><td>8</td><td>0.67</td><td>0.68</td></tr> <tr><td>9</td><td>0.66</td><td>0.71</td></tr> <tr><td>10</td><td>0.66</td><td>0.70</td></tr> <tr><td>11</td><td>0.65</td><td>0.69</td></tr> <tr><td>12</td><td>0.65</td><td>0.71</td></tr> </tbody> </table>	epoch	Train Loss	Validation Loss	1	0.69	0.69	2	0.68	0.69	3	0.68	0.70	4	0.68	0.69	5	0.68	0.68	6	0.67	0.69	7	0.67	0.68	8	0.67	0.68	9	0.66	0.71	10	0.66	0.70	11	0.65	0.69	12	0.65	0.71
epoch	Train AUC	Validation AUC																																																																														
1	0.56	0.57																																																																														
2	0.57	0.57																																																																														
3	0.58	0.56																																																																														
4	0.59	0.58																																																																														
5	0.60	0.57																																																																														
6	0.61	0.58																																																																														
7	0.63	0.57																																																																														
8	0.64	0.57																																																																														
9	0.65	0.57																																																																														
10	0.67	0.56																																																																														
11	0.69	0.58																																																																														
12	0.70	0.59																																																																														
epoch	Train Loss	Validation Loss																																																																														
1	0.69	0.69																																																																														
2	0.68	0.69																																																																														
3	0.68	0.70																																																																														
4	0.68	0.69																																																																														
5	0.68	0.68																																																																														
6	0.67	0.69																																																																														
7	0.67	0.68																																																																														
8	0.67	0.68																																																																														
9	0.66	0.71																																																																														
10	0.66	0.70																																																																														
11	0.65	0.69																																																																														
12	0.65	0.71																																																																														

**Table 6: Full Images AUC, Loss v/s Epoch**

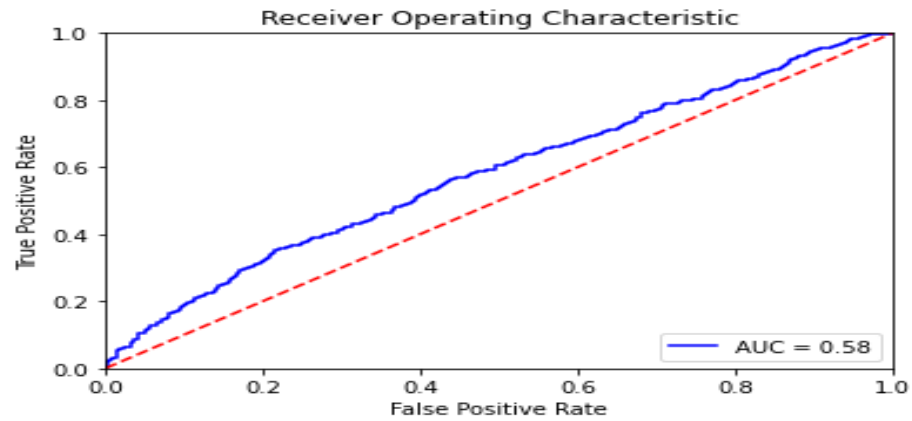
**Discussion:** From the above models we can see that “VGG16(CC + MLO)” model is trained perfectly up to 10 epochs as after 9 epochs gap between training and validation AUC started increasing. “VGG16(Only CC)” and “VGG (Only MLO)” model started overfitting after 10 epochs as we can observe validation loss started increasing significantly and validation AUC curve started flattening. So, from above analysis, we can observe that if we have trained our models for a greater number of epochs then we may get overfitted model.

### TEST ROC-AUC Graph

The below table shows the ROC-AUC curve for all models trained on full mammogram images. AUC value for each graph is given in below-left corner of each graph.

	TEST ROC-AUC Curve
<b>VGG16 (CC+MLO)</b>	 <p>Receiver Operating Characteristic</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>AUC = 0.59</p>
<b>VGG16(ON LY CC)</b>	 <p>Receiver Operating Characteristic</p> <p>True Positive Rate</p> <p>False Positive Rate</p> <p>AUC = 0.59</p>

**VGG16  
(Only  
MLO)**



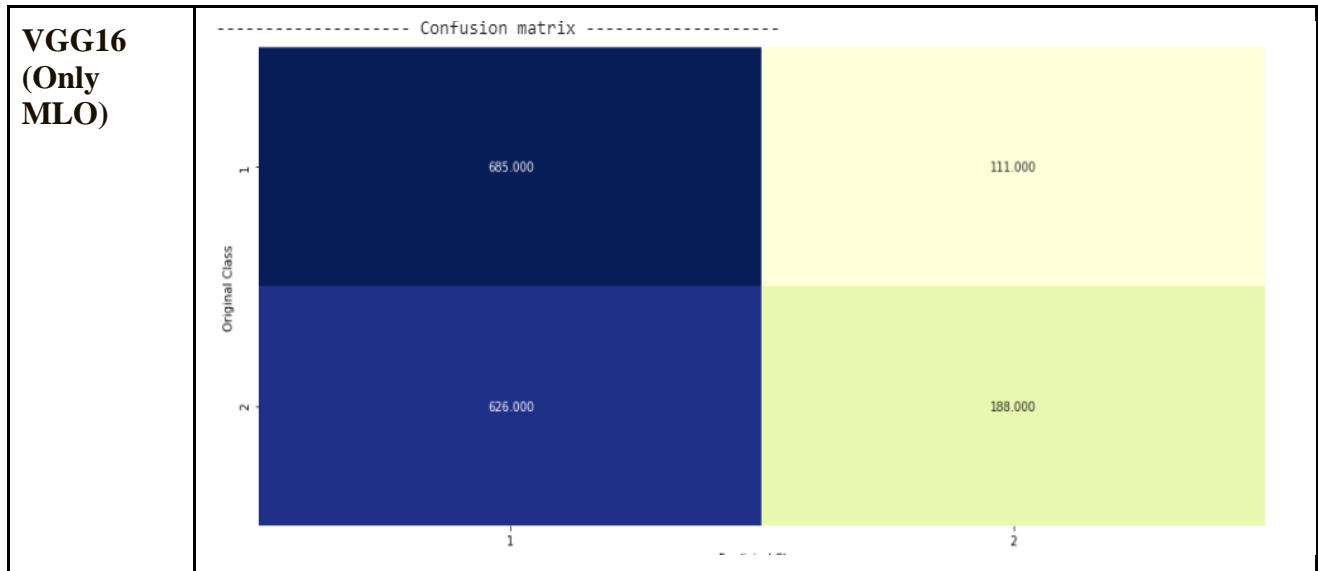
**Table 7: Full Images ROC-AUC V/S Epoch**

**Discussion:** AUC value helps in comparing model making better prediction, but all three above graphs shows approximately same AUC value so comparing models based on AUC values will be difficult, so we need to see confusion matrix and classification metrics for detailed observations.

## Confusion Matrix

The below table shows the confusion matrix from predictions made on the Test dataset by models trained on the full image dataset. In the below figures '1' signify non-cancerous and '2' signify cancerous.

	Confusion Matrix									
VGG16 (CC+MLO )	<div><div>----- Confusion matrix -----</div><table><tr><th>Original Class \ Predicted Class</th><th>1</th><th>2</th></tr><tr><th>1</th><td>396.000</td><td>1291.000</td></tr><tr><th>2</th><td>252.000</td><td>1466.000</td></tr></table></div>	Original Class \ Predicted Class	1	2	1	396.000	1291.000	2	252.000	1466.000
Original Class \ Predicted Class	1	2								
1	396.000	1291.000								
2	252.000	1466.000								
VGG16 (Only CC)	<div><div>----- Confusion matrix -----</div><table><tr><th>Original Class \ Predicted Class</th><th>1</th><th>2</th></tr><tr><th>1</th><td>587.000</td><td>228.000</td></tr><tr><th>2</th><td>483.000</td><td>356.000</td></tr></table></div>	Original Class \ Predicted Class	1	2	1	587.000	228.000	2	483.000	356.000
Original Class \ Predicted Class	1	2								
1	587.000	228.000								
2	483.000	356.000								



**Table 8 : Full Images Confusion Matrix**

**Discussion:** For VGG model trained on both ‘CC’ and ‘MLO’ views, the confusion matrix shows that models are little bit biased towards class ‘2’(cancerous) because number of false positive (originally non-cancerous but predicted cancerous) are large.

For models trained on ‘only CC’ and ‘Only MLO’, they look a little biased towards class 1(non-cancerous) due to large number of false negative (originally cancerous but predicted non-cancerous). Results from these models shows that deep learning model trained on full image with image size of (224,224) pixel is not doing very good job in mammogram classification.

## Classification Metrics

The below table summarizes AUC, Precision, Recall, Accuracy, Specificity for VGG160 neural network architectures. The models were trained using random initialization for images that include ‘CC’ and ‘MLO’, ‘Only CC’, and ‘Only MLO’ images.

	Test AUC	Precision	Recall/Sensitivity	Accuracy	Specificity
<b>VGG16 (CC+MLO)</b>	0.59	0.532	0.853	54.68%	0.234
<b>VGG16 (Only CC)</b>	0.59	0.610	0.424	57.01%	0.7202
<b>VGG16 (Only MLO)</b>	0.58	0.629	0.231	54.2%	0.8605

**Table 9 : Full Images Classification Metrics**

From the above table, we can see that Test-AUC and Accuracy for all models do not have much difference so it is difficult to compare models using these metrics, but we can compare models using Precision, Recall, Specificity. For models trained on (CC+MLO) views, VGG16 model give very low Specificity. This also means that these models give large numbers of false cancerous results which indicate that our model is biased towards class 2(Cancerous).

For models trained on ‘Only CC’ and ‘Only MLO’ give high Precision and Specificity but low value of Recall that is because these models give a large number of false negatives (Originally cancerous but predicted non-cancerous).

As AUC and Accuracy value for models trained on ‘Only CC’ and ‘Only MLO’ don’t show much difference so we can say that among ‘CC’ and ‘MLO’ does not give better results than others on deep learning classification.

## 4.2 Summary

In this chapter, we discussed the training and results for both segmented and full images trained on VGG16 and RESNET50 neural network architecture. For segmented images, we trained four models with an image size of (224,224) pixels with a random initializer and transfer learning weight initialization. The resulted AUC value for these models is around 0.8 and results show that transfer learning using weights from 'ImageNet' does not improve AUC for models trained on Mammograms. Two models trained on the image size of (50,50) segmented images gives AUC and accuracy values which are less than the results we get on models trained on (224,224) pixels. From these results, we can say that image size does affect Mammogram deep learning results. For full images, we train six models on VGG16 neural network architecture. The models are trained on three different datasets which contain images from both views, Only CC and Only MLO. The result shows that a particular view doesn't give better results than any other view. In the next chapter, we will see what limitations and problems we faced in our work and also possible future work.



## **CHAPTER 5 - LIMITATIONS, CONCLUSION and FUTUREWORK**

Throughout the journey of this dissertation, we encountered many difficulties and some of them are discussed in this chapter. The examination from the past can help reach a conclusion and answer the research questions that were asked in the beginning. In addition to this, we have also discussed the future scope of this work.

### **5.1 Limitations Encountered**

Over the span of Dissertation work, few limitations were encountered that prevented the ability to carry out experiments that we thought before the start of the thesis. Some of these limitations were solved either through code optimization or changing our experimentation platform. These limitations do not create a bias for the results that we obtained but the opportunity to work and train deep learning models on large training data sets was lost. It would be fascinating to know what problems and limitations we encountered and how we solved it.

#### **Limited RAM**

Initially, we planned to pre-process the data and train our deep learning models on a trial version of Google Cloud Platform (GCP), So when python files tested on a smaller dataset executed in GCP to process around 161 GB of data, limited memory errors were encountered. So, we processed the data in batches but training a larger neural network like RESNET50 with a large image size would not be possible.

#### **Graphics Processing Unit (GPU) Availability**

GPUs are very useful for faster processing and training neural networks on large training data set. At the beginning of the dissertation, we presumed that we will get GPU on GCP but when we started our work on GCP we came to know that we don't have the availability of GPUs in the GCP trial version. Preprocessing data on GCP without GPU would have taken 10 days to complete so we solved this problem by optimizing our code. We optimized cropping operation in our preprocessing part which was taking  $O(n^2)$  time complexity to linear time complexity.

## 5.2 Discussion on Research Questions

### 1. Does transfer learning using models trained on an ‘ImageNet’ dataset improve mammogram deep learning cancer detection results?

We used segmented images to train VGG16 and RESNET50 models with random and transfer learning and we got to know that models trained using random initializer gives AUC value as 0.82,0.79 respectively while both models trained with transfer learning gives AUC values as 0.81, 0.80 respectively. These values don’t show much difference in their performance, so we can say that results from transfer learning using models trained on ‘ImageNet’ dataset doesn’t improve deep learning cancer detection results.

### 2. How much does the size of the mammogram input image affect deep learning model results.

We used segmented images to train VGG16 and RESNET50 models with images of size (224,224) and (50,50) pixels. We got the following results:

	AUC	Accuracy (%)
VGG16 (224,224 PIXELS)	0.82	75.4
RESNET50 (224,224 PIXELS)	0.79	71.9
VGG16 (50,50 PIXELS)	0.77	69.7
RESNET50 (50,50 PIXELS)	0.78	68.8

**Table 10: Research Question 2 Comparison Table**

VGG16 AND RESNET50 model trained on (224,224) pixel image size shows higher value of AUC and accuracy than models trained on (50,50) pixel image size. So, we can say that Mammogram input image does affect deep learning model results.

### 3. Between ‘CC’ and ‘MLO’, which view gives better deep learning results?

We conducted experimentation for images with both views, only ‘CC’ and only ‘MLO’ images and got the following results.

	Test AUC	Accuracy
VGG16 (Only CC)	0.59	57.01%
VGG16(Only MLO)	0.58	54.2%

**Table 11: Research Question 3 Comparison Matrix**

From the above results we can see that the test AUC and Accuracy for VGG16 model trained on dataset containing ‘Only CC’ images perform better than VGG16 model trained on dataset containing ‘Only MLO’. So, we can conclude that between ‘CC’ and ‘MLO’, ‘CC’ view gives better deep learning results.

### 5.3 Future Work

In accordance with this dissertation work, we saw segmented images gives better results than full images but finding large radiologist annotated images dataset is difficult to get and we know that neural network works better with a large amount of data, therefore, future work can be taken to see if transfer learning from models trained on annotated segments to models train on full images improve specificity and sensitivity results than we obtained in this dissertation.

## References

1. Breastcancer.org. 2020. What Is Breast Cancer? | Breastcancer.Org. [online] Available at: <[https://www.breastcancer.org/symptoms/understand\\_bc/what\\_is\\_bc?gclid=Cj0KCQjw6uT4BRD5ARIsADwJQ1\\_egCsQKStx0SRBif7e49pJjEuCx3s6TDig2ujzqRJiSmfuEz\\_BP4oaAlGXEALw\\_wcB](https://www.breastcancer.org/symptoms/understand_bc/what_is_bc?gclid=Cj0KCQjw6uT4BRD5ARIsADwJQ1_egCsQKStx0SRBif7e49pJjEuCx3s6TDig2ujzqRJiSmfuEz_BP4oaAlGXEALw_wcB)> [Accessed 25 August 2020].
2. Cancer Network. 2020. CAD System Boosts Early Detection Of Breast Cancer. [online] Available at: <<https://www.cancernetwork.com/view/cad-system-boosts-early-detection-breast-cancer>> [Accessed 25 August 2020].
3. World Health Organization. 2020. Breast Cancer. [online] Available at: <<https://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/#:~:text=Various%20methods%20have%20been%20evaluated,%25%20in%20high%2Dresource%20settings.>> [Accessed 25 August 2020].
4. Cancer Network. 2020. CAD System Boosts Early Detection Of Breast Cancer. [online] Available at: <<https://www.cancernetwork.com/view/cad-system-boosts-early-detection-breast-cancer>> [Accessed 25 August 2020].
5. Ponraj, D & Jenifer, M & Poongodi, P. & Manoharan, Samuel. (2011). A Survey on the Preprocessing Techniques of Mammogram for the Detection of Breast Cancer. Journal of Emerging Trends in Computing and Information Sciences. 2.
6. Mirzaalian, Hengameh & Ahmadzadeh, M.R. & Sadri, Saeed & Jafari, Mehdi. (2007). Pre-processing Algorithms on Digital Mammograms. 118-121.
7. J. Nagi, S. Abdul Kareem, F. Nagi and S. Khaleel Ahmed, "Automated breast profile segmentation for ROI detection using digital mammograms," 2010 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES), Kuala Lumpur, 2010, pp. 87-92, doi: 10.1109/IECBES.2010.5742205.
8. Raba D., Oliver A., Martí J., Peracaula M., Espunya J. (2005) Breast Segmentation with Pectoral Muscle Suppression on Digital Mammograms. In: Marques J.S., Pérez de la Blanca N., Pina P. (eds) Pattern Recognition and Image Analysis. IbPRIA 2005. Lecture Notes in

Computer Science, vol 3523. Springer, Berlin, Heidelberg.  
[https://doi.org/10.1007/11492542\\_58](https://doi.org/10.1007/11492542_58) (pp. 471–478).

9. Mughal B, Muhammad N, Sharif M, Rehman A, Saba T (2018) Removal of pectoral muscle based on topographic map and shape-shifting silhouette. BMC Cancer 18:1–14.  
<https://doi.org/10.1186/s12885-018-4638-5>

10. Boss RSC, Thangavel K, Daniel D, Arul P (2013) Automatic mammogram image breast region extraction and removal of pectoral muscle. Int J Sci Eng Res 4(5):1–8

11. Sheba, K. and Gladston Raj, S., 2018. An approach for automatic lesion detection in mammograms. Cogent Engineering, 5(1).

12. Rahimeto, S., Debelee, T., Yohannes, D. and Schwenker, F., 2019. Automatic pectoral muscle removal in mammograms. Evolving Systems,

13. Tot, T., Tabar, L., Dean, P.B.: The pressing need for better histologic mammographic correlation of the many variations in normal breast anatomy. Virchows Archiv 437(4) (October 2000) 338–344

14. Ryan C., Krawiec K., O'Reilly UM., Fitzgerald J., Medernach D. (2014) Building a Stage 1 Computer Aided Detector for Breast Cancer Using Genetic Programming. In: Nicolau M. et al. (eds) Genetic Programming. EuroGP 2014. Lecture Notes in Computer Science, vol 8599. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-662-44303-3\\_14](https://doi.org/10.1007/978-3-662-44303-3_14)

15. Shen, L., Margolies, L.R., Rothstein, J.H. et al. Deep Learning to Improve Breast Cancer Detection on Screening Mammography. Sci Rep 9, 12495 (2019). <https://doi.org/10.1038/s41598-019-48995-4>

16. Lotter. 2017. A Multi-Scale CNN and Curriculum Learning Strategy for Mammogram Classification. [online] Available at: < <https://arxiv.org/abs/1707.06978v>> [Accessed 26 August 2020].

17. Ragab, D., Sharkas, M., Marshall, S. and Ren, J., 2019. Breast cancer detection using deep convolutional neural networks and support vector machines. PeerJ, 7, p.e6201.

18. Geras, K.J., Wolfson, S., Shen, Y., Wu, N., Kim, S., Kim, E., Heacock, L., Parikh, U., Moy, L. and Cho, K., 2017. High-resolution breast cancer screening with multi-view deep convolutional neural networks. arXiv preprint arXiv:1703.07047.

19. Chougrad, H., Zouaki, H. and Alheyane, O., 2018. Deep Convolutional Neural Networks for breast cancer screening. *Computer Methods and Programs in Biomedicine*, 157, pp.19-30.
20. Kooi, T., Litjens, G., van Ginneken, B., Gubern-Mérida, A., Sánchez, C., Mann, R., den Heeten, A. and Karssemeijer, N., 2017. Large scale deep learning for computer aided detection of mammographic lesions. *Medical Image Analysis*, 35, pp.303-312.
21. Courses, A., I, C., II, C., PyTorch, D., Partnership, D., Sponsorship, G. and Kit, M., 2020. Opencv. [online] Opencv.org. Available at: <<https://opencv.org/>> [Accessed 26 August 2020].
22. Team, K., 2020. Keras Documentation: About Keras. [online] Keras.io. Available at: <<https://keras.io/about/>> [Accessed 26 August 2020].
23. Colab.research.google.com. 2020. Google Colaboratory. [online] Available at: <<https://colab.research.google.com/notebooks/intro.ipynb>> [Accessed 26 August 2020].
24. Google Cloud. 2020. Cloud Computing Services | Google Cloud. [online] Available at: <<https://cloud.google.com/>> [Accessed 26 August 2020].
25. MIT News | Massachusetts Institute of Technology. 2020. Explained: Neural Networks. [online] Available at: <<https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>> [Accessed 26 August 2020].
26. Brownlee, J., 2020. What Is The Difference Between Test And Validation Datasets?. [online] Machine Learning Mastery. Available at: <<https://machinelearningmastery.com/difference-test-validation-datasets/>> [Accessed 26 August 2020].
27. 2020. [online] Available at: <<https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9>> [Accessed 26 August 2020].
28. Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
29. Neurohive.io. 2020. VGG16 - Convolutional Network For Classification And Detection. [online] Available at: <<https://neurohive.io/en/popular-networks/vgg16/>> [Accessed 26 August 2020].
30. Medium. 2020. Understanding And Implementing Architectures Of Resnet And Resnext For State-Of-The-Art Image.... [online] Available at: <<https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624>> [Accessed 26 August 2020].

31. Google Developers. 2020. Classification: True Vs. False And Positive Vs. Negative. [online] Available at: <<https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative>> [Accessed 26 August 2020].
32. Medphys.royalsurrey.nhs.uk. 2020. Collection | OMI-DB. [online] Available at: <<https://medphys.royalsurrey.nhs.uk/omidb/about-omi-db/collection/>> [Accessed 26 August 2020].

## **Appendix A – Link to GitHub Repository**

<https://github.com/prashanaga/Classification-of-Mammogram-images-using-Deep-Learning-Techniques.git>