Multi-Label Breast Cancer Screening Using EfficientNetB3 Feature Extractor

Dedefo Bona Buta

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2021

Adama, Ethiopia

Multi-Label Breast Cancer Screening Using EfficientNetB3 Feature Extractor

By: Dedefo Bona Buta

Advisor: Prof. Yun Koo Chung (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2021

Adama, Ethiopia

DECLARATION

I hereby declare that this Master Thesis entitled ***"Multi-Label Breast Cancer Screening Using EfficientNetB3 Feature Extractor"*** is my original work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation

Dedefo Bona Buta _____ _____

Name of the student Signature Date

RECOMMENDATION

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled ***"Multi-Label Breast Cancer Screening Using EfficientNetB3 Feature Extractor"*** prepared under my/our guidance by **Dedefo Bona** submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science Engineering. Therefore, I/we recommend the submission of revised version of the thesis to the department following the applicable procedures.

Prof. Yun Koo Chung (Ph.D.) _____ _____

Major Advisor Signature Date

_____ _____ _____

Co-Advisor Signature Date

APPROVAL PAGE

I/we, the advisors of the thesis entitled ***"Multi-Label Breast Cancer Screening Using EfficientNetB3 Feature Extractor"*** and developed by **Dedefo Bona**, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Prof. Yun Koo Chung (Ph.D.)	_____	_____
Major Advisor	Signature	Date
_____	_____	_____
Co-Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Dedefo Bona have read and evaluated the thesis entitled ***"Multi-Label Breast Cancer Screening Using EfficientNetB3 Feature Extractor"*** and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science Engineering.

_____	_____	_____
Chairperson	Signature	Date
_____	_____	_____
Internal Examiner	Signature	Date
_____	_____	_____
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____	_____	_____
Department Head	Signature	Date
_____	_____	_____
School Dean	Signature	Date
_____	_____	_____
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

Praise to **Allah**, Lord of the worlds. First and foremost, I thanks to **Allah** Almighty for all of his favor and means on me. And peace and blessing be upon prophet **Muhammed**.

Next, I want to express special thanks to our SIG leader and my advisor **Prof. Yun Kon Chung**. I appreciate his being on time to attend every schedule on our SIG meeting for progress reports and comments on presentations as well as resource support for training. This way I was inspired to work hard and participate in a presentation with updating my progress in every meeting.

I also want to express my gratitude to **Dr. Worku Jifara**, our SIG member, for his support starting from proposal guidance to thesis works. Furthermore, his initiation for me to investigate deep learning environments in detail for my future career is also unforgettable.

Again, I want to express my gratitude to all my family members, especially, my father for his successful responsibility by educating and supporting all his children including me. And my brother Abdela Bona being inspiring me with his valuable idea and financial support.

Finally, I thank Mr. Anteneh Tilaye for his deep review of my work, my SIG members for peaceful interaction and valuable discussion, and all my CSE classmates.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SNIP CODES

LIST OF ACRONYMS/ABBREVIATIONS

ANN	Artificial Neural Network
BI-RADS	Breast Imaging Reporting and Data System
CAD	Computer Aided Diagnosis
CBIS	Curated Breast Imaging Subset
CNNs	Convolutional Neural Networks
CPU	Central Processing Unit
DDSM	Digital Database for Screening Mammography
DICOM	Digital Imaging and Communications in Medicine
DL	Deep Learning
GCN	Global Contrast Normalization
GPU	Graphic Processing Unit
KNN	K-Nearest Neighbor
MIAS	Mammographic Image Analysis Society
ML-BCS	Multi-Label Breast Cancer Screening
ONN	Optimized Neural Network
RAM	Random Access Memory
ReLU	Rectified Linear Unit
ROI	Region-of-Interest
SVM	Support Vector Machine

LIST OF SYMBOLS

D: Dataset

xi: all images in a dataset

yi: all labels associated with images' dataset

D^d: Dataset domain

R^L: Dataset Mapped to Label identified domain

BI: BI-RADS I

BII: BI-RADS II

BIII: BI-RADS III

BIV: BI-RADS IV

Calc: Calcification

3D: 3-dimension

4D: 4-dimension

y_n: Ground truth labels

$\hat{y}_n$: $Predicted\ labels$

F_{tr}: Transformation function

U: Transformed features

δ: ReLU function

σ: Sigmoid function

ABSTRACT

Breast cancer is the first new case occurring and the fifth death-causing cancer disease in 2020, which collapses the families as women are the pillar of the family members. 3-dimension image acquisition technology increases treatment methods and decreases the rate of breast cancer death. Furthermore, single-label breast cancer classifications are deployed to solve the problem. However, it cannot convey more information about mammogram images. In the area where the shortage of datasets is there as in the medical image, data augmentation and transfer learning is the best solution. A Multi-Label Breast Cancer Screening is a deep learning model using a transfer learning model that gives information on finding, density, and pathology of patients' breast mammograms to the radiologists, aiming for life-saving and supporting the family. The existing model shows low accuracy on feature extraction and more layers with a very large number of nodes in a layer in the classification part. In this study, the best feature extractor for this particular work is investigated and identified as efficientnetb3, and Optimized Neural Network is used for the classification part. The proposed model outperforms the previous work in all evaluation metrics with a 13.25% f1_score, 53% hamming loss, 36.7% coverage error, and 12.5 an exact match. In addition, the number of parameters decreased from 134 million to 20 million which comes from the optimizing of classification part of the model. This work concludes that the efficientnetb3 feature extractor with Optimized Neural Network in the classification part has made the best improvement over the existing model in terms of evaluation metrics and network performances.

Keywords: *Breast Cancer, Efficientnetb3, Screening, Mammogram, Transfer Learning, Multi-Label*

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

Cancer is a large group of diseases characterized by cells that grow uncontrollably, invade the nearest organ, and/or spread beyond their usual limits. It can begin in any organ or tissue in the body. A malignant tumor is another common name for cancer. Breast, lung, colorectal, prostate, and stomach are the top-5 organ affected by cancer diseases(Sung et al., 2021).

Throughout the world, 19.3 million new cancer cases are assessed and nearly 10.0 million cancer deaths took place in 2020. Breast cancer in women has replaced lung cancer as new cases analyzed cancer from the top-5 lung, colon, prostate, and stomach cancers(Sung et al., 2021). Lung cancer stayed the highest cause of cancer deaths from colon, liver, stomach, and female breast cancers, and the percentage is given in figure 1.1 graphically. Death rates have reduced since 1989 and these decreases are assumed to be the outcome of treatment advances, earlier detection by screening, and enhanced awareness. Early detection of breast cancer is to be the best way to save lives and decrease healthcare costs over time(Siegel et al., 2017). Figure 1.1 shows this graphically.

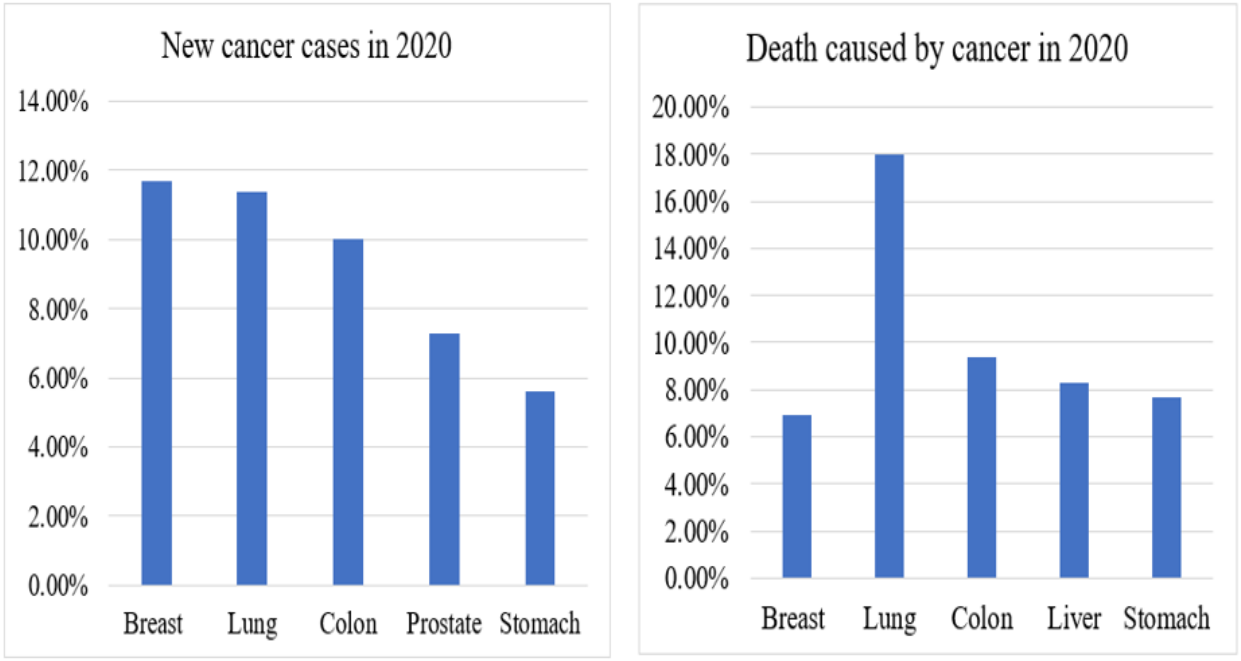


Figure 1.1.Cancer new cases and death rates per top-5 organs

Image acquisition and interpreting technologies to detect and diagnose breast cancer continue to advance aiming to give a better diagnosis and less dangerous option for patients(Narod et al., 2015). Mammography[1] is the major cause of breast cancer mortality reduction, although there is a potential limitation to the procedure(Heywang-Köbrunner et al., 2011). In fact, reading a mammogram correctly is challenging for most radiologists. In a survey(Whang et al., 2013), error in diagnosis was the major cause of misconduct accommodate against radiologists. Many of such cases started from the image quality or the radiologist's unsuccess to diagnose breast cancer on mammograms(Whang et al., 2013). To increase the rate of accurate diagnoses, lesions that contain a 2% probability of being malignant are advocated for a biopsy(Sickles, n.d.). However, malignant biopsies detected are only 15 to 30%(Sickles, n.d.). Pain, fear, anxiety, direct financial losses, indirect costs lost for work lost, and complexity risks are the risks caused by benign biopsies(Review et al., 2006).

Most of the available works depend on single-label screening, in which every mammogram is thought to contain only one label from the given mammogram characteristics. Moreover, the study(Jiao et al., 2016) focused on the masses in a mammogram to diagnosis patients. In other studies (Karahaliou et al., 2008; Wang et al., 2016), researchers used micro-calcification to screen breast cancer. While authors(Arevalo et al., 2016; Berment et al., 2014; Chougrad et al., 2018; Jiang et al., 2014; Jiao et al., 2016; H. Li et al., 2017) depended on the morphology of the contour and shape of the breast mass lesion in mammograms images, assuming it is the most differing criteria among benign and malignant masses. Other works in(Gastounioti et al., 2016; Vachon et al., 2007; Winkel et al., 2016) researchers depended on the mammographic density and parenchyma patterns for breast cancer risk measurement.

These methods, although working well, it has difficult to raise the relationship that exists among the different annotations to help to give a full diagnosis. Single-label screening is the foundation for multi-label screening which aims to forecast the set of important labels for a given input. The inputs used to train such a model for this type of problem have several labels. The multi-class and multi-label classification differ in terms of output label space. labels are not thought to be mutually

[1] Mammography is an X-ray picture of the breast and help physician for detecting early signs of breast cancer.

exclusive in multi-class whereas it is independent in multi-label. (Madjarov et al., 2012; Zhang & Zhou, 2014).

In this, works efficientnetb3 as feature extraction will be used along with less layered and small nodes in a fully connected neural network. This will be done by empirical selection of transferable model from Keras application on a fully annotated dataset for Multi-Label Breast Cancer Screening (ML-BCS).

1.2. The Motivation of Study

Mammography is one of the popular modalities in breast imaging. However, there is some limitation to this imaging system. First, the scarcity of radiologists' manpower to interpret the mammogram images in health care centers in our country. Second, the present radiologists are challenged to read the mammogram image to interpret it accurately. To solve the problem of mammogram reading and interpreting the Computer-Aided Diagnosis (CAD) system using Deep Learning (DL) plays a crucial role in medical image analysis.

The Convolutional Neural Networks (CNNs) are popular and accepted DL models for the task of image classification since it provides high accuracy and impressive results compared with other models. It was specially designed for use on two-dimensional data, such as medical images (Allaouzi & Ben Ahmed, 2019). There are many types of research done to use the CAD system on breast cancer single-label screening by deep learning. However, no more multi-label classification was done on breast cancer screening. Multi-label classification gives more detailed information about a single image such as its density, finding, and pathology. To benefit from its advantage, this study was inspired to explore the earlier work to explore the multi-label classification.

1.3. Statement of Problem

Problem formulation: Assume $D = \{(x_i, y_i)\}_{i=1}^{N}$ to be dataset, the aim is a multi-label classification, in which the input is a Region-Of-Interest (ROI) image $x \in D^d$ with $d = r\ x\ r$ and the output is the set of labels $y \in Y = \{0, 1\}^L$.

Every ROI is associated with multiclass labels $L = \{1, \ldots, L\}$, in this case ($L = 8$) and the predefined set of class labels are {1: BI-RADS I, 2: BI-RADS II, 3: BI-RADS III, 4: BI-RADS IV, 5: Mass, 6: Calcification, 7: Benign, 8: Malignant}. For an image x the target is the label set $y = [y_1, \ldots, y_L] \in y_j$ where $y_j = 1$ if the j^{th} label is applicable to x; else $y_j = 0$. The aim is to train a multi-label

prediction model $f(x)$ that transforms a given unlabeled image to the L-dimensional label space y corresponding to the confidence scores:

$$F(x): D^d \rightarrow R^L. \quad \ldots \quad (1.1)$$

F(x) is a multi-label classification and it can predict more than one label at a time for one image. Four labels are from the density subcategory and one label with the highest probability is selected. Again, two labels are from finding subcategories and one label with higher confidence is selected. Finally, two labels are from pathology and one with a higher probability is selected. In (Chougrad et al., 2020) use the VGG16 model to solve the multi-label classification. There is a problem of misclassification was found. In this study, a transferable learning model with less competitional cost and high accuracy based on the empirical experiment for multi-label breast cancer screening.

1.4. Research Question

This study aims to answer the following important research questions.

1. Which transferable learning model solves the multi-label breast cancer screening accurately?
2. Which fine-tuning method for the classification part of the model gives better performance for multi-label breast cancer screening?

1.5. Objective

1.5.1. General Objective

The general objective of this study is to develop Multi-Label Breast Cancer Screening by transfer learning feature extractor with fully connected neural networks.

1.5.2. Specific Objective

To achieve the general objective, these specific objectives will be addressed by the study.

- Finding a robust feature extraction algorithm for multi-label breast cancer screening.
- Investigating the optimized neural network for the classification part of multi-label breast cancer screening.
- Modeling multi-label breast cancer screening with transfer learning.
- Preparing breast cancer image dataset for training and testing.
- Training the prepared dataset on the identified model.
- Testing the trained model with a test set to fine-tune parameters.

1.6. Scope and Limitation of Study

1.6.1. Scope of the Study

This research focuses on developing a multi-label breast cancer classification. The study aims, by using transferable deep CNN, to classify breast cancer to multi-label classification to give a more accurate report to the radiologist.

1.6.2. Limitation of the Study

This study does not incorporate imaging modalities other than mammography. The mammogram dataset which is not public or does not give full access is additionally excluded for this works. Furthermore, in the dataset ROI is explicitly given with label description. Consequently, image segmentation is not included in training the model by focusing mainly on the accuracy of the model. But, to help the doctors to find the area of the tumor, image segmentation is included while using the model.

1.7. Beneficiaries of the study

The result of this study is used in the health care center for screening women's mammogram images to check breast cancer risk. Detecting and diagnosing breast cancer is saving the life as women are the pillar of the family. Breast cancer treatment is an easy and high chance of cure when it is identified in an early stage. So, it is recommended for women to start screening at the age of 30 years and one's per year. The specific individual benefited from this work are:

- Radiologists: he/she can benefit from it by serving patients accurately and effectively.
- Woman: her cancer status is identified early so that she can be confident or treated in a controllable state.
- Families: since a woman is the pillar of families, they can lead a comfortable life without fear confidently or follow the treatment properly after the presence of cancer is identified through the screening.

1.8. Organization of the Thesis

The rest of this thesis is organized as follows:

Chapter Two: discusses the background literature and related works regarding breast cancer screening. This chapter also explores the theoretical framework of DL and classification.

Chapter Three: enlighten the research methodology, including different methods and techniques used to develop the solution and select the appropriate one. Data collection method, design tools, feature extraction, and evaluation methods are also discussed

Chapter Four: will cover points about the proposed model in detail. Model Architecture and pseudocode for implementation, learning functions, and mathematical correspondent descriptions have been made.

Chapter Five: Explains how the desired proposed solution is implemented. and the working environment setup, efficientnetb4, efficientnetb3, vgg16, densenet121, and efficientnetb3 implementation with training implementation described using snip code.

Chapter Six: The obtained testing result from the ML-BCS model is presented and compared with the other related work to show the best judgment.

Chapter Seven: Finalizes the research and recommends directions for possible future work.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

Cancer can be defined as a disease state that causes cells to stop responding to normal stimuli. Abnormal cells that grow and spread without limit can cause tumors. Tumors that have to unfold beyond the tissue where they developed and are growing into the surrounding healthy tissues are considered invasive. Among women, breast cancer is the highest form of cancer and it is the top-5 cancer death-causing, next to lung, colon, liver, and stomach(Sung et al., 2021).

The breast is an organ of the human body and belongs to the exocrine glands, which indicates it can produce breast milk. It is constructed primarily of fat, connective tissue, and hundreds of small structures called lobes. Each lobe contains many cavities called alveoli. These cavities are connected with milk-releasing epithelial cells and circled by myoepithelial cells. Myoepithelial cells extract milk from the lobes into thin tubes called ducts. Pipes are connected in a network. These pipes are connected to form a larger pipe, which ends at the nipple. Depending on the affected part of this anatomy of the breast part the breast mammograms are annotated as density, mass, or calcification.

2.1.1. Breast Density

Mammography density is a consistent and high probability chance of breast cancer in multiple populations and different ages. Currently, this attribute has been added to the existing breast cancer prediction model, and the discrimination accuracy has been improved by adding this model, albeit with a slight increase. Through verification, these models can replace the existing Gail model for clinical risk assessment. However, the absolute risk estimates derived from these improved models still have limited ability to characterize the likelihood that an individual will develop cancer(Vachon et al., 2007).

2.1.2. Breast Mass

In mammography, the mass is defined as the space occupied by the lesion, which is seeable in two different views, and contour characteristics. Contrary to mass, density asymmetry corresponds to the appearance of local asymmetry of the breast, without a defined contour. For this reason, in addition to the external front and rear views and oblique views, it is also recommended to use

outline views. This is also useful for distinguishing the overlay image of the breast from the actual mass(Berment et al., 2014; Jiang et al., 2014).

The classification of characteristics of the mass plays a critical role in the diagnosis of breast cancer. Existing CAD methods benefit greatly from low or intermediate-level features, which are not good at simulating the actual diagnostic process, increasing the difficulty of improving classification performance. The identification of benign and malignant breast masses is one of the main contents of mammography diagnosis(Jiao et al., 2016).

2.1.3. Breast Calcification

Calcifications are small calcium deposits that appear as tiny bright spots on mammograms. They are characterized by their type and distribution characteristics. It is interested only because it is sometimes developed nearest to breast cancer. This can be a ductal carcinoma at an early stage in situ or to real invasive cancer. Furthermore, calcifications are characterized by architectural distortion with no obvious visible quality. This includes peaks radiating from one point and substantial shrinkage or distortion of substantial edges(Sampat et al., 2005).

2.2. Mammography

Mammography may be the most discussed and argued recent medical procedure, but there is no disagreement on these two points. First, it has a better combination of sensitivity and specificity tools for screening breast cancer. Secondly, according to current practice, mammography is better. It is captured in two standard views. figure 2.1 shows a mediolateral oblique view.

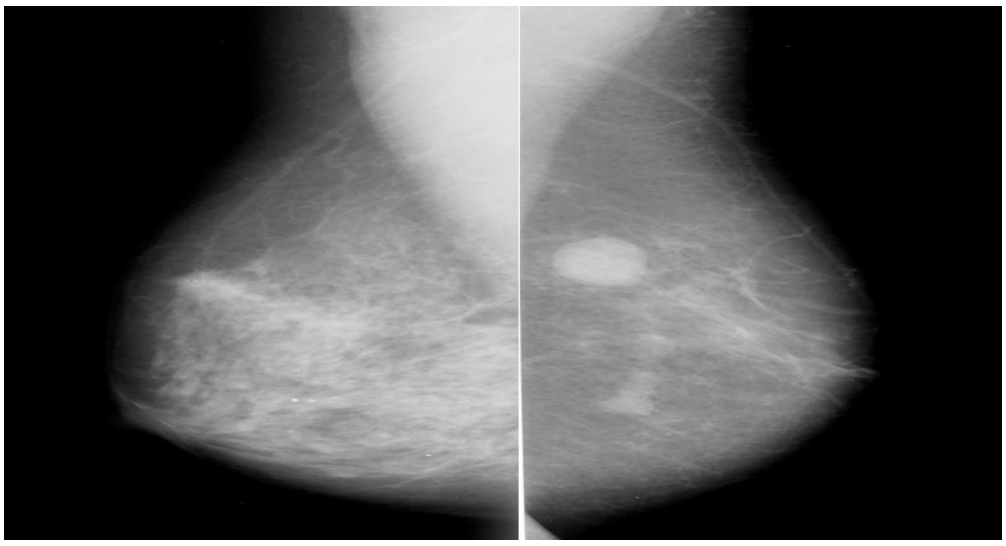


Figure 2.1. Mediolateral oblique mammogram.

Mammogram has two main uses, screening and diagnosis; and it is an X-ray-based method suitable for women without signs or symptoms of breast disease, for the early detection of breast cancer. The second view is craniocaudal and is shown in figure 2.2.

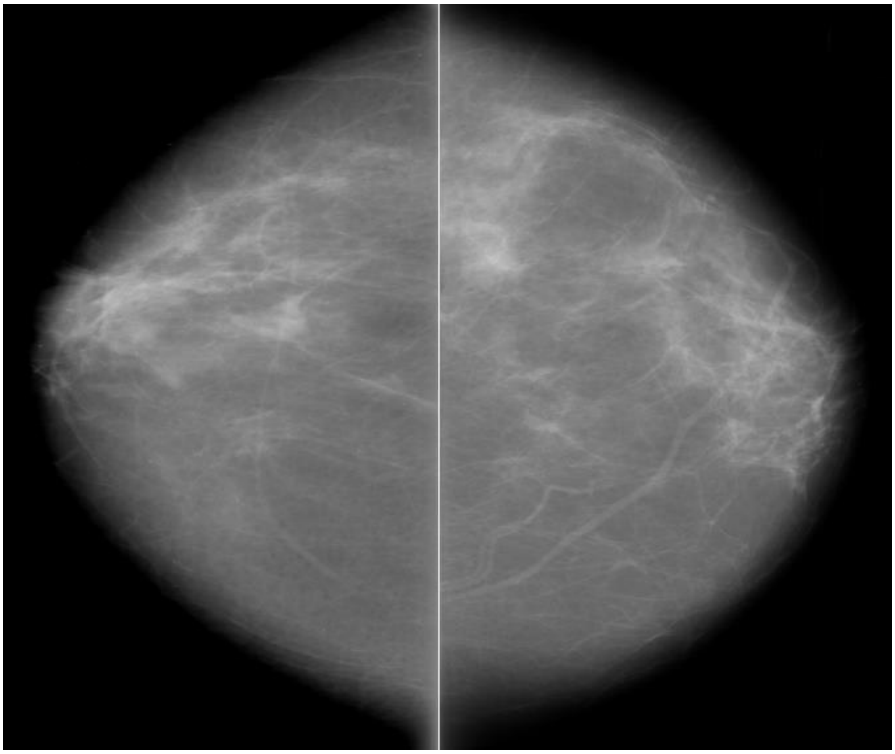


Figure 2.2. Craniocaudal mammogram

In contrast, diagnostic mammograms (also known as problem-solving mammograms) use the same X-ray-based technique but are tailored by the radiologist to a specific patient's signs or symptoms. The procedure is designed to diagnose previously seen signs of breast disease in women or to determine whether a woman with a personal history of breast cancer or biopsy-confirmed breast disease develops breast cancer(Diagnosis, 2005).

2.3. Medical Image Analysis

The medical image is essential to exploit the amount of data and explore and display information contained appropriately for specific medical tasks. This field gets a lot of attention and improvable accuracy after the development of DL algorithms. The familiar medical image analysis is detecting normal or affected X-ray images and identifying abnormality types and severities in radiology. In the 1980s CAD systems were developed and applied to the clinical system, although it has many false positives, it is earlier medical image analysis(Kim et al., 2019).

Unlike general image processing which has one primary purpose (such as enhancing image aesthetics or creative art), the sole purpose of medical image processing is to raise the readability of the rendered content. This may involve enhancing the image itself to increase the perception of certain features and extract information automatically(Ritter et al., 2011).

One of the radiologists' most important duties is to make an effective differential diagnosis with each patient's medical images. This may range from assessing the existence or non-existence of disease to determining the type of malignancy(Kim et al., 2019). Examining the medical image is a challenging and time-consuming task for physicians. To reduce the challenge and save time for another task for the physician the CAD system is developed to analyze the medical image since 1980(Kim et al., 2019). However, the deployed CAD system at that time produces more false positives than physicians, resulting in increased assessment time and needless biopsies (Kim et al., 2019).

The advancement in computer vision has changed and improved the inspiration of medical image usage to improve image acquisition, disease detection, treatment, and prediction. It may use texture, shape, contour, and prior example, as well as contextual information from an image series, to produce 3D and 4D data that aids human comprehension. The automatic processing of 3D medical images was greatly enhanced diagnosis and therapy. This automation produces a slew of new research questions in the fields of computer vision, graphics, and robotics. As a result, in 1991 the output of medical images was projected to be worth 8 billion (US dollars) due to computer vision applied to 3D imagery(Ayache, 1995).

Image processing, image feature analysis, and data classification using methods such as artificial neural networks (ANN) are examples of computer algorithms known as Artificial Intelligence(AI) (Shiraishi et al., 2011; Stoitsis et al., 2006). An ANN is a nonlinear computing model that imitates

the processing of information by a biological neural network(Shiraishi et al., 2011; Stoitsis et al., 2006).

It is generalized that computerized medical image processing combined with AI can be used in clinical works and help to increase diagnosis efficiency(Stoitsis et al., 2006). The architectures of ANNs are similar to the ones shown in Figure 2.3.

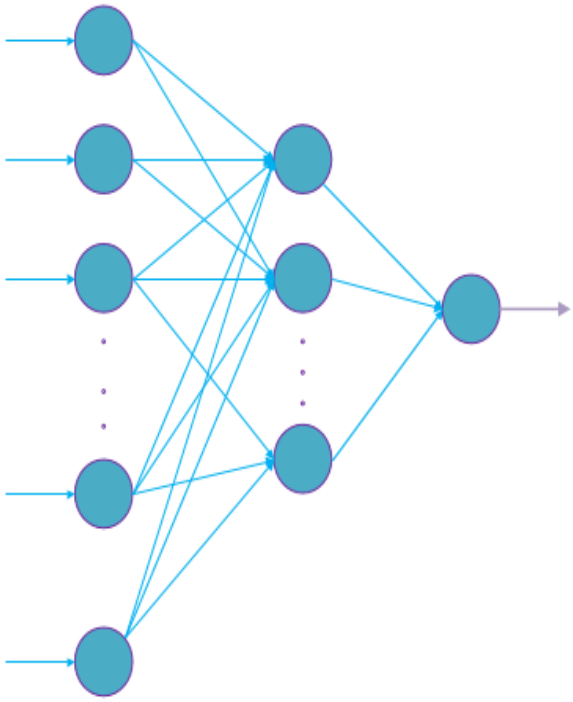


Figure 2.3 Schematic description of feed-forward ANN with 3 layers

Machine learning algorithms have been commonly used in CAD systems to learn a concept from examined cases to assist medical experts in making a diagnosis(M. Li & Zhou, 2007). A large number of diagnosed samples must be found to learn a well-performed hypothesis. Since artifacts like lesions and organs cannot be adequately interpreted by a simple equation, medical pattern recognition requires "learning from examples"(Suzuki, 2013). Despite the ease with which the samples can be obtained during routine medical exams, it is normally difficult for medical experts to make a diagnosis for any of the samples collected(M. Li & Zhou, 2007).

2.4. Deep Learning

Deep learning algorithms are a subfield of machine learning algorithms, that aim at extracting multiple features of distributed representations. It helps computational models that stack multiple

layers of processing to learn the distribution of data with multiple levels of abstraction. In addition to this, it is making major improvements in giving solutions to problems faced by the artificial intelligence community (Lecun et al., 2015).

DL fingerprints are CNN architecture that has two components: feature extraction and classification. A convolution tool, pooling, and stride that helps for separating and identifying various features of the image such as patterns, lines, and edge for analysis are called feature extraction. On the other hand, classification is a fully connected NN that processes the output from the feature extracted and predicts the class of the image(Kim et al., 2019).

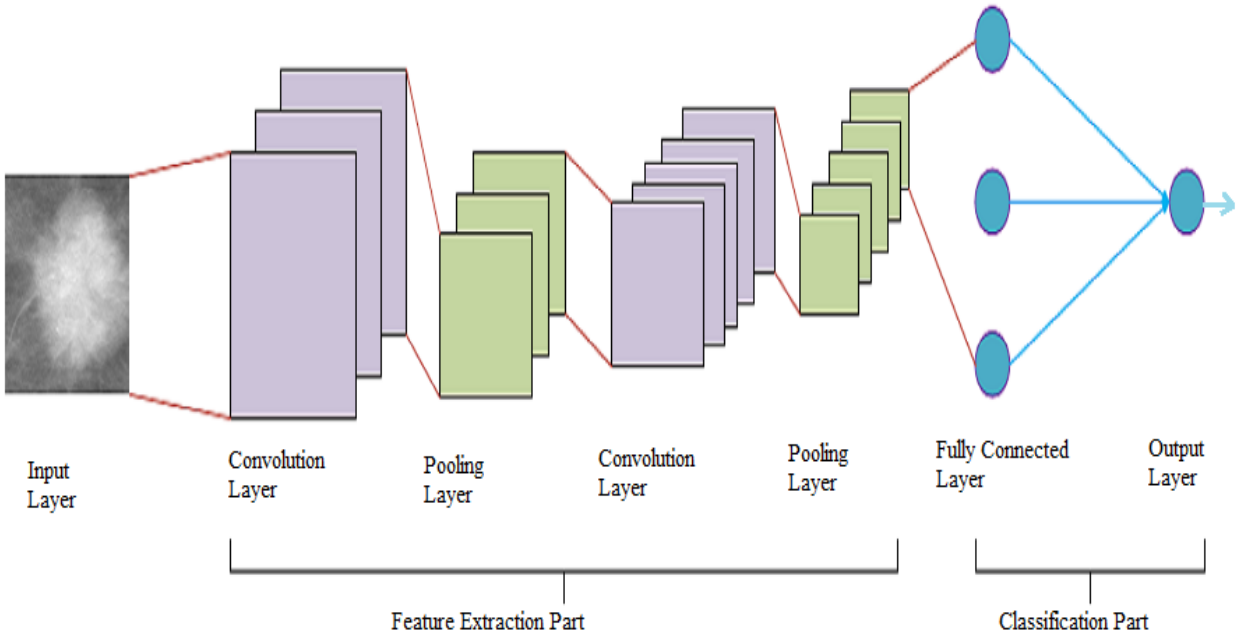


Figure 2.4. The basic architecture of DL algorithms

2.5. Major category for screening

Density-based screening: mammographic breast density, along with age, remains a major predictor factor for screening mammography(Gastounioti et al., 2016; Vachon et al., 2007; Winkel

et al., 2016). American College of Radiologists gives mammographic density and volumetric density using full-field digital mammography(D'Orsi, 1996).

Micro-calcification-based screening: the objective is to train classifying breast cancer depending on the content of micro-calcification found in the breast(Karahaliou et al., 2008),(Wang et al., 2016). Micro-calcifications are the dethronement of calcium oxalate and calcium phosphate in the breast tissue. In (Karahaliou et al., 2008), the co-occurrence of wavelet coefficient and first-order statistics of wavelet coefficient matrices features are extracted from sub-images surrounding ROIs tissue. The power of each attribute set in specifying malignant from benign tissue is found using a probabilistic NN.

Mass-based Screening: Characteristic of mass gives a crucial role in the screening of breast cancer(Arevalo et al., 2016; Berment et al., 2014; Chougrad et al., 2018; Jiang et al., 2014; Jiao et al., 2016; H. Li et al., 2017). In the shape of breast mass in mammography, there is very critical information to identify the pathology of a breast mass. RMS slope is the new contour descriptor. In(H. Li et al., 2017), KNN, SVM, and ANN architecture are used to screen benign breast mass from the malignant mass.

2.6. Related Works

In this section, a review of research works done on the areas of breast cancer screening since 2017 are presented.

The researchers, in(H. Li et al., 2017) come up with new techniques to convert the 2D contour of breast mass in mammography to a 1D signature. The techniques can express the contour features as well as the regularity of breast mass. Four local features within the new contour descriptor are extracted from subsections. The benign mass has a soft form that translates into a simple signature, while the malignant tumor has a toothed outline leading to an approximate signature. The contours of each anomaly are extracted by connecting the point expressed with the series code in the overlap files. The method in which a 2D outline is transformed into a 1D signature can describe the characteristic of the entire contour. RMS slope describes the new contour. KNN, SVM, and ANN classifiers are used to identify benign breast mass and malignant mass. However, the models used are classical machine learning which needs a hand-engineered feature.

In (Chougrad et al., 2018) works three most impressive CNN architectures proposed at that time: VGG16, ResNet50, and Inception v3(Sangeetha & Prasad, 2006; Simonyan & Zisserman, 2015; Szegedy et al., 2016) trained on ImageNet(Fei-Fei et al., 2010) was used. The study investigates the benefit of transfer learning rather than random initialization for every model and finds effective fine-tuning methods. The primary aim of that work is to develop state-of-the-art CNNs and to use transfer learning, to build a powerful mass lesion screening tool that can help the radiologist get support to give more accurate diagnoses. The first constructed layers of a CNN learn generic characteristics and can be performed more as edge detectors, which must be useful for many activities, but the following levels become progressively more specific for the details of the labels contained in the dataset. Accordingly, since the images of the massive member lesion are very different from images of imagenet, they propose to fine-tune models to regulate the characteristics of the Conv2D blocks and make them more recipients. So, they optimized weights of the pre-trained networks using the new series of images restarting backpropagation unfrozen layers.

Other researchers(Nawaz et al., 2018), develop a DL method based on a CNN model classification for multi-class breast cancer. The objective of the proposed approach is to identify the breast tumors as benign or malignant but the study predicts the subclass of the tumors as Fibroadenoma, Lobular carcinoma, etc. The work results on histopathological images on the BreakHis dataset and showed that the DenseNet CNN model attained high processing efficiencies in the multi-class screening of breast cancer tasks when compared with state-of-the-art models. In addition, the proposed model aims to treat high-resolution images generally used for histopathological breast cancer screening. The densenet121 model is modified to extract entirely global characteristics of histological images and use them in the training process using transfer learning. However, the study was concerned mainly with the effects of different resolutions on the accuracy of the models.

The work proposes an "end-to-end" approach(Shen, 2017) in which a fully annotated dataset with ROI information is used and a model identifies local image patches using pre-trained models. In this study, lesion annotations are expected only in the first phase of training and succeeding stages expect limited to image-level labels, solving the reliance on hardly available lesion annotations. To get this, the heatmap having a dimension of $u \times u \times c$ is needed. Then next to the heatmap many layers are added to transform outputs and connect with the final classification output of the image. Adding a Conv2D layer to the top of the patch classifier's outputs converts the entire patch

classifier into a filter and expands its receptive field. The CNN methods VGG16 and ResNet50 are used to classify mammograms to get good performance compared to the model before them. But this study worked on how to annotate the breast cancer lesion automatically by deep learning.

Furthermore study (Chougrad et al., 2020), proposes the joint learning of the tasks using multi-label image classification. Additionally, researchers introduce a different fine-tuning method to apply a pre-trained model, that takes benefits of the full image representation learning while adapting the pre-trained weight to the task. so, they propose a customized label decision scheme, adapted to the multi-label translation problem, which forecasts the optimal trust for each visual concept. But this study was used the label powerset which was computationally expensive to transform the multi-label classification to single-label classification and correlated each label.

A novel classification approach DeepBC for identifying the breast mammogram pathology, depending on the deep CNNs is proposed by (Wenzhong et al., 2020). DeepBC is the sequential integration of Inception, ResNet, and AlexNet to extracted features from mammograms and classifies mammograms as malignant or benign tissues. DeepBC model is intended and proposed to reliably and accurately solve the classification of breast cancer. It extracts not only the features of low-level but also the characteristics of the representative, which is suitable for classifiers with discriminatory analysis capabilities. However, this study has low accuracy and is inefficient in terms of time and computation.

Table 2.1. Summary of related works

No	Authors	Method and Techniques	Gaps	Evaluation metrics
1	H. Li (H. Li et al., 2017).	RMS slope is used for contour descriptors. KNN, SVM, and ANN are used for classification.	It depended on the ML method and masses contour for classification	✓ Accuracy
2	Chougrad (Chougrad et al., 2018).	VGG16, ResNet50, and Inception V3 architectures are used for classification.	It was depending on masses for classification and did not include multi-label classification.	✓ Accuracy ✓ AUC
3	Nawaz (Nawaz et al., 2018)	DenseNet CNN model is used in this paper.	multi-label transformation or algorithm adoption problem.	✓ Accuracy
4	Shen (Shen, 2017).	The models used in this work is GVV16 and ResNet50	This word is concerned with solving the annotation problem.	✓ AUC

5	Wenzhong (Wenzhong et al., 2020).	A novel method DeepBC which is the integration of Inception, ResNet, and AlexNet.	It uses three networks sequentially and it is inefficient in terms of time and computing cost. It is also a single-label classification problem.	✓ Accuracy ✓ F1_score
6	Chougrad (Chougrad et al., 2020) (baseline paper)	VGG16 model from scratch and its pre-trained with fine-tuning is used. Powerset problem transformation is applied.	Uses 3 layers of fully connected NN which is computationally costive. Uses a powerset that was computationally expensive.	✓ F1_score ✓ mAP ✓ convergence ✓ hamming_loss ✓ ranking_loss ✓ Exact_match

Table 2.1 shows models and techniques used, gaps that exist, and evaluation metrics of the model. the last paper is the baseline of this study. The feature extractor and quantity of NN layer and nodes used are identified as a gap. Additionally, problem transformation implemented is expensive in terms of calculation handled.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Chapter Overview

The description of the medical image dataset is shortly briefed. Next, a detailed explanation of the dataset used in this work is given. Feature extraction and classification parts of the model are included. Development tools and evaluation methods are discussed in the last part of this chapter. Figure 3.1 shows the process of acquiring to utilize the dataset.

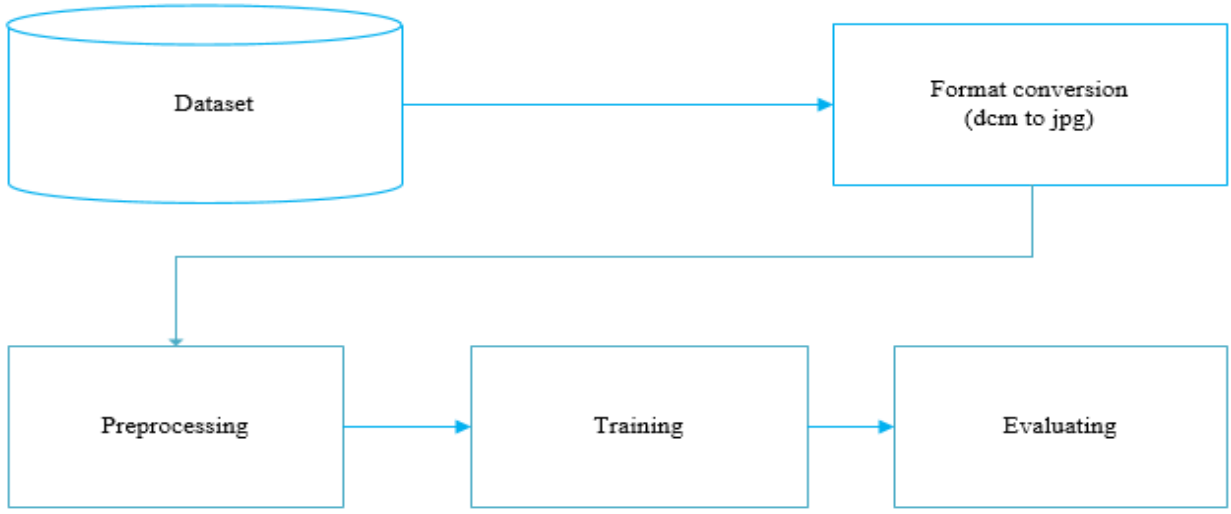


Figure 3.1. Dataset utilizing process

3.2. Medical Dataset

Medical image has specific standard file extension to store, compress, retrieve and put as a dataset and make it available online. This file extension is .dcm and is known as Digital Image Communication in Medicine. This is what makes the medical image dataset more difficult to use directly in the CNN model. Before using such a dataset, it must be converted to JPG/PNG file format.

3.3. CBIS-DDSM Dataset

CBIS-DDSM: **(**Curated Breast Imaging Subset of DDSM)(Lee et al., 2017) a modified and improved version of the Digital Database for Screening Mammography (DDSM) is the dataset that was annotated properly for multi-label image classification. The images have been compressed and available in DICOM format. Modified ROI, mask for segmentation and pathologic diagnosis are given within the mammograms. ROIs contain pixel-level annotations, pathology labeling:

benign or malignant, finding types: masses or calcifications, and ACR class labeling (density labeling) are updated within the dataset. Because of proper multi-label annotation, updated ROI, mask for segmentation, and a large number of images it contains this dataset is selected for this work.

This dataset is publicly available online on the website at: https://wiki.cancerimagingarchive.net/display/Public/CBIS-DDSM and when downloaded on a local hard disk its size is 152GB. It has full mammogram images, cropped images, masks, and labels description of an image in excel .csv file format. Each image file in this dataset is separated by an individual folder and it is merged with many folders.

For instance, C:\CBIS-DDSM\CalcTrain\Calc-Training_P_00005_RIGHT_CC\08-07-2016-DDSM-23157\1.000000-full mammogram images-38548\1.1.dcm, is path of one image file in this dataset.

To make this dataset ready for use it was converted to JPG and this process is done in two popular ways. The first method is using python code to convert from DICOM file format to JPG file format. In this method, one must access all image file path directories to load an image and give the name that is the same as an excel file which is the label description of the same image and save the image in another folder. The second method is using an online converter. In this case, the same procedure is followed but its drawback is its running time. Because of the short time processing, the former procedure is applied, which means python code, to convert the image to make it accessible for training and testing.

The dataset contains train and test set separately including finding as mass and calcification categories. Tables 3.1 explain the numbers of the train and test set with finding categories.

Table 3.1. Training and testing set statistics in number

Finding type	Training	Testing	Total
Calcification	1,226	284	1,510
Mass	1,232	361	1,593
Total	2,458	645	3,103

The co-occurrence matrix shows two important pieces of information about the dataset. Formerly, the diagonal line shows the total number of that label in the dataset. This can help to know the distribution of each label with every image in the dataset. The second information conveyed by the co-occurrence matrix is the relationship between different labels. For example, the mass or calcification with density BII and BIII are more probably benign than malignant. This is illustrated by the percent of orange darkness of the co-occurrence matrix. In figure 3.2. shows training and testing set label co-occurrence matrix of the CBIS-DDSM on the left and right side respectively.

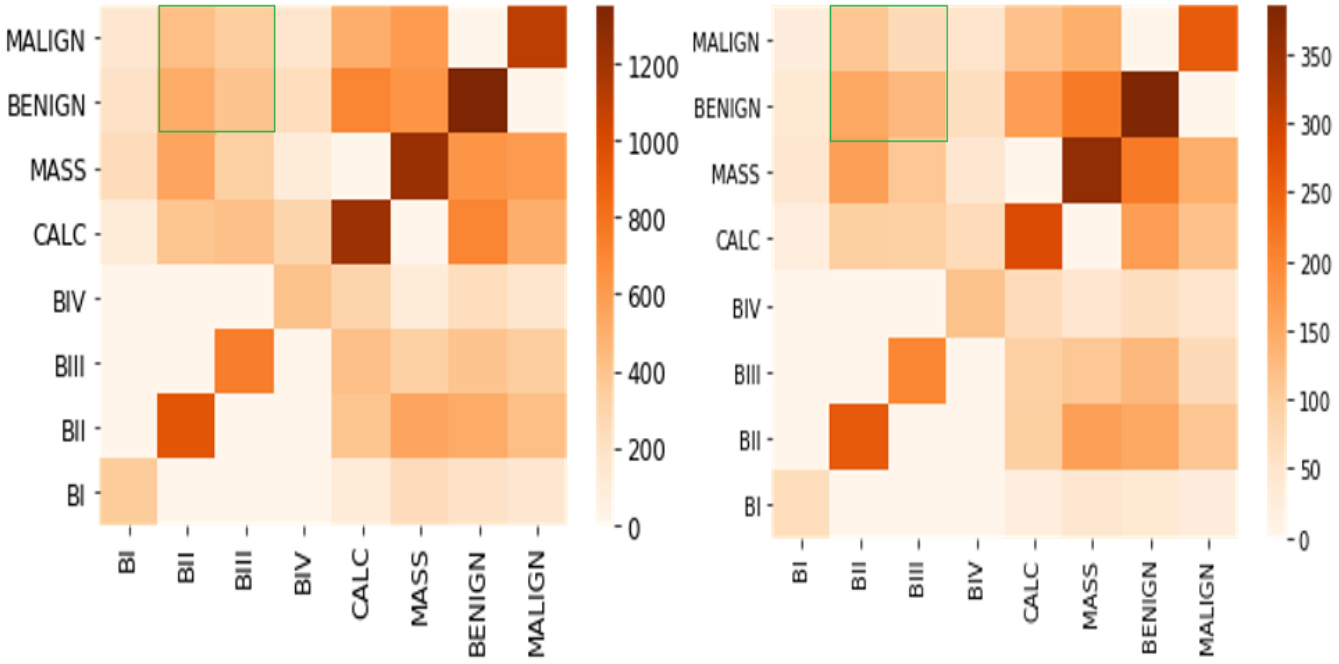


Figure 3.2. Co-occurrence training and testing set label matrix of CBIS-DDSM

There are 8 different labels in the dataset which are subdivided into 3 subcategories. Those are density, finding, and pathology. The density subcategory includes BI – BIV labels. This distribution is summed up to a total number of train or test sets. As seen from the bar chart in figure 3.3, the density subcategory is the distribution that imbalance data has been found in. Finding is another subcategory of label that is described by mass and calcification found in breast mammograms. The training set or testing set can be in the mass label category or calcification label category. Pathology is the last and important subcategory label of the dataset. Because the

other two subcategories are needed as the supporting element to categorize the image into its corresponding pathological category.

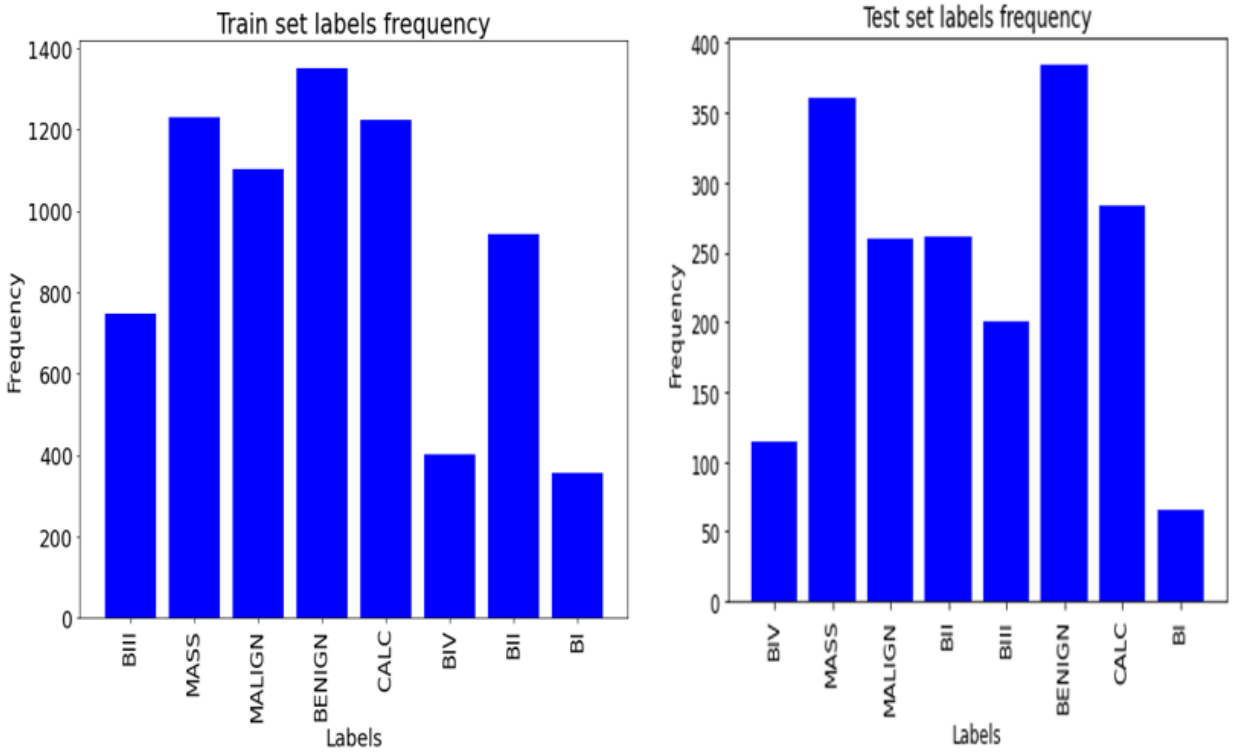


Figure 3.3. The train and test set label frequency of CBIS-DDSM

Table 3.2 shows the distribution of the labels with their subcategories in the dataset. One image can be expressed with three labels one from each subcategory. For instance, an image may have BI density, Mass finding, and Benign pathology.

Table 3.2. Distribution of labels in the dataset

Subcategory	Density				Finding		Pathology	
Labels	BI	BII	BIII	BIV	Mass	Calc	Benign	Malign
Training set	353	944	748	403	1232	1226	1350	1103
Testing set	66	261	200	114	360	283	384	259

Figures 3.4 shows the sample ROIs image mammogram with its corresponding labels.

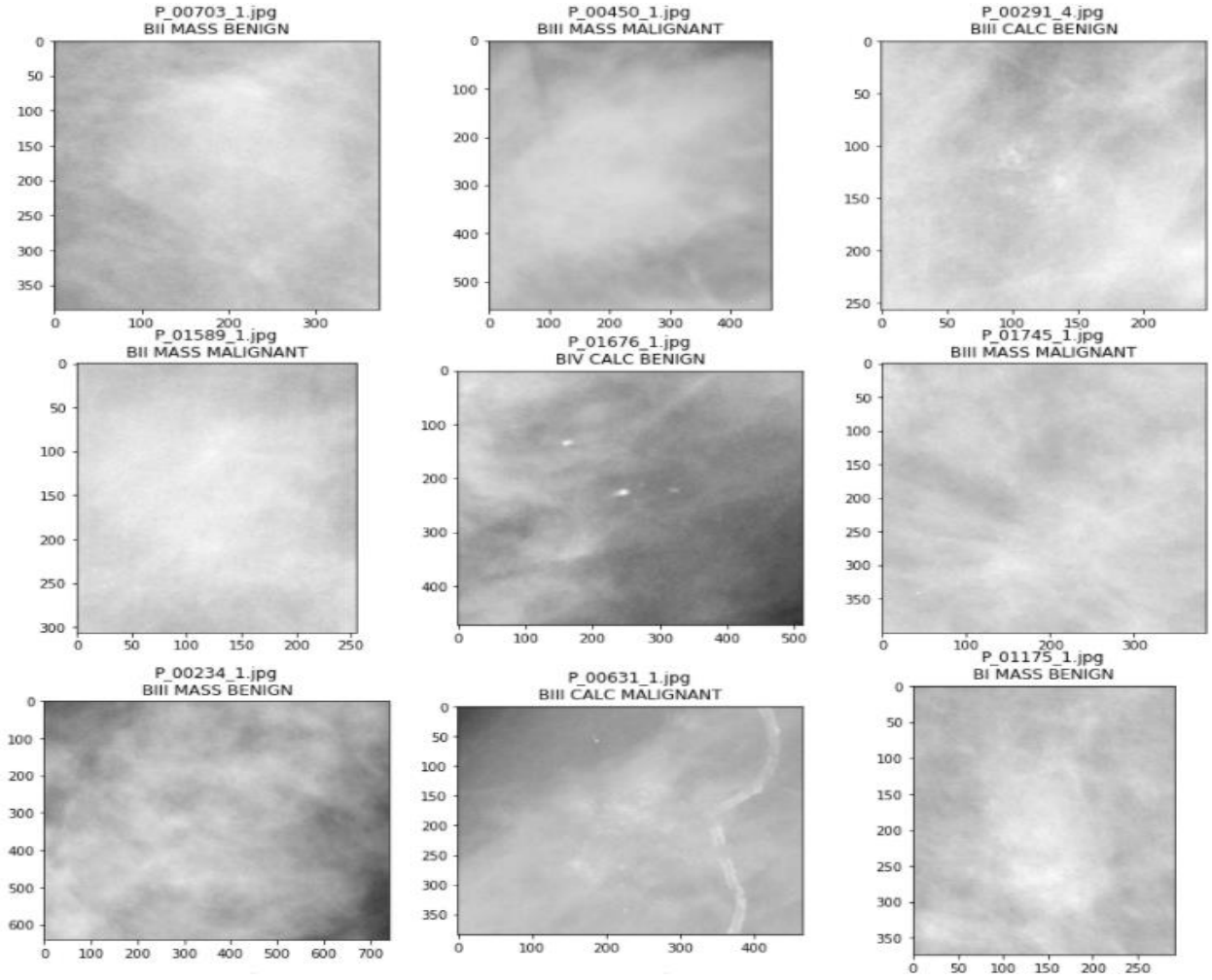


Figure 3.4. ROI mammogram image of CBIS-DDSM with different resolution

The name of an image in a mammogram is given like p_00005_1 and named as patient id. One patient has at most four mammograms. Two mammograms are right and left breast in craniocaudal (CC) views whereas, two mammograms are right and left breast in mediolateral oblique (MLO) views. CC and MLO are the standard views for screening mammography(Lee et al., 2017). To indicate these four images of one patient in naming the last underscore from 1-4 numbers are used. Table 3.3 shows the sample label description in a CSV file.

Table 3.3. Sample labels description of a dataset in one hot vector in CSV format

patient_id	BI	BII	BIII	BIV	CALC	MASS	BENIGN	MALIGN
p_00005_1	0	0	1	0	1	0	0	1
p_00005_2	0	0	1	0	1	0	0	1
p_00007_1	0	0	0	1	1	0	1	0
p_00007_2	0	0	0	1	1	0	1	0
p_00008_1	1	0	0	0	1	0	1	0
p_00008_2	1	0	0	0	1	0	1	0
p_00008_3	1	0	0	0	1	0	1	0
p_00008_4	1	0	0	0	1	0	1	0
p_00010_1	0	0	1	0	1	0	1	0
p_00010_2	0	0	1	0	1	0	1	0
p_00011_1	0	0	1	0	1	0	1	0
p_00011_2	0	0	1	0	1	0	1	0
p_00012_1	0	1	0	0	1	0	0	1
p_00012_2	0	1	0	0	1	0	0	1
p_00013_1	0	0	0	1	1	0	1	0
p_00014_1	0	0	0	1	1	0	0	1
p_00014_2	0	0	0	1	1	0	0	1
p_00016_1	0	0	0	1	1	0	0	1
p_00016_2	0	0	0	1	1	0	0	1
p_00019_1	0	0	0	1	1	0	1	0

3.4. Feature Extraction

The empirical experiment was conducted on models found on the Keras application which has online available pre-trained models, to select the best feature extractor for this work. Those models are 26 and they are inception, vgg, resnet, densenet, mobilenet, nasnet, efficientnet, and their variants.

Accordingly, efficientb3 was chosen confirming that it is leading in terms of accuracy, and having optimal time and computational efficiency. Figure 3.5 shows a comparison of efficientnet network families with other networks.

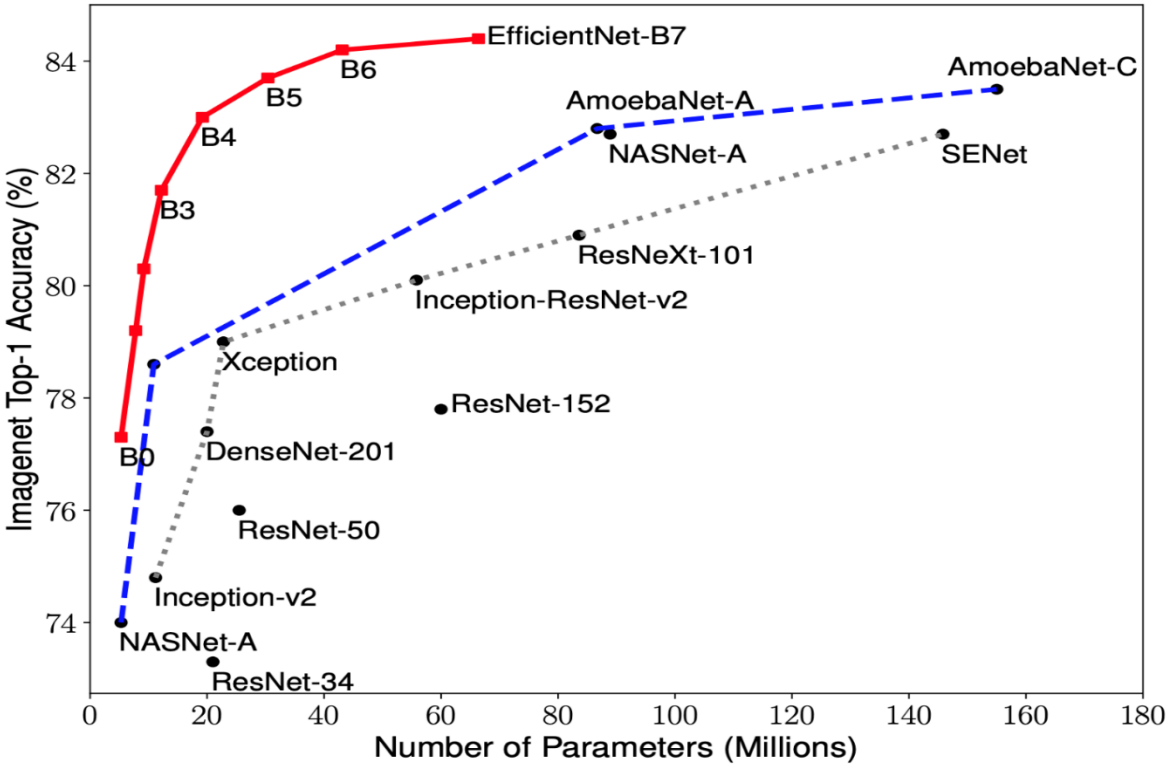


Figure 3.5. EfficientNet comparison with other models

Adopted from: (Tan & Le, 2019)

In (Basha et al., 2020) researchers explain the relationship between the feature extraction part of the network with a fully connected layer of classification parts. According to this study, when the feature extraction part of the model is very deep fully connected part of the network will be less. Moreover, they explain this relationship is also affected by the nature of the dataset used by the model. Studying those methods this work applies fully connected NN with a small number of layers with minimum nodes are identified.

3.5. Developments Tools

For this research, many types of development tools are used to design and implement the proposed thesis work. The development tool section gives a description and justification of these development tools.

3.5.1. Hardware Tools

Hardware tools that will be used for the implementation of this research are hard disk: for the storage of the datasets, GPU: for increasing the computation for an efficient training purpose, CPU: for testing the models, and RAM: for effectively train the model with CPU and GPU cooperatively.

3.5.2. Software Tools

Software tools will have used for the implementation of this study to write the code, debug, visualize results, and document the research process for reporting. Software tools are programming tools and libraries this work will utilize are:

Tensorflow: - It is a machine learning platform to build a model and deploy it in client environments like other real-world applications. It supports machine learning, deep learning, and flexible numerical computation. It is easier than PyTorch in designing networks. Mathlab uses more memory to run training and it is difficult to handle this in the limited resources.

Jupyter Notebook: An interactive web-based application that helps configure, load python API, and write python code.

Keras**:** is user-friendly, has easy extensibility supports modularity**,** and works with Python as a high-level framework or API. It is more attractive to use when compared to Tensorflow.

Python: - Python programming with anaconda cooperation will use for the implementation of the study

Flask: - it is a python module as a web framework used for testing mammograms.

3.6. Evaluation Methods

The performance measurement of the multi-label classification is differing from the single-label classification. The evaluation metrics used for this works are f1_score, hamming loss, coverage error, and exact match.

a. F1_Score(f)

Fl_score is the average of precision and recall. It is a balanced measure between the positive predictive value (precision) and the ratio of correctly predicted positive values to the actual positive values (recall). The higher Fl indicates better performance of the learning algorithm.

$$F1\_score = \frac{1}{N}\sum_{n=1}^{N}\frac{2|\hat{y}_n \cap y_n|}{|\hat{y}_n| + |y_n|} \quad \ldots \quad (3.3)$$

b. Hamming_Loss(f)

The Hamming loss is the result of the wrong labels divided by the sum number of labels. Thus, a smaller value indicates better performance.

$$Hamming\_loss(f) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{L}|\hat{y}_n \Delta y_n|, \quad \ldots \quad (3.4)$$

Where Δ is the symmetric difference between two label sets.

c. Coverage error(f)

Coverage error is defined as the mean number of labels to investigate on the ordered list of predicted numbers (by their ranks) to cover all reference labels. A smaller value indicates is better.

$$Coverage\_error(f) = \frac{1}{N}\sum_{n=1}^{N} max_{y=y_n} r(x_n, y) - 1 \quad \ldots \quad (3.5)$$

d. Exact_Match(f)

The Exact match is a severe metric for multi-label classification since it does not include the additional notion of being partly correct.

$$Exact\_Match(f) = \sum_{n=1}^{N} I\,|\hat{y}_n == y_n| \quad \ldots \quad (3.6)$$

CHAPTER 4

4. PROPOSED TRANSFER LEARNING MODEL FOR ML-BCS

4.1. Introduction

This chapter explains the detail of the proposed solution for multi-label breast cancer classification. Image preprocessing is illustrated first. Secondly, feature extraction and classification parts of the architecture are discussed. Thirdly, the image segmentation model is explained. Lastly, activation and loss functions for multi-label classification are explored. Figure 4.1 explains the process flow of the proposed solution.

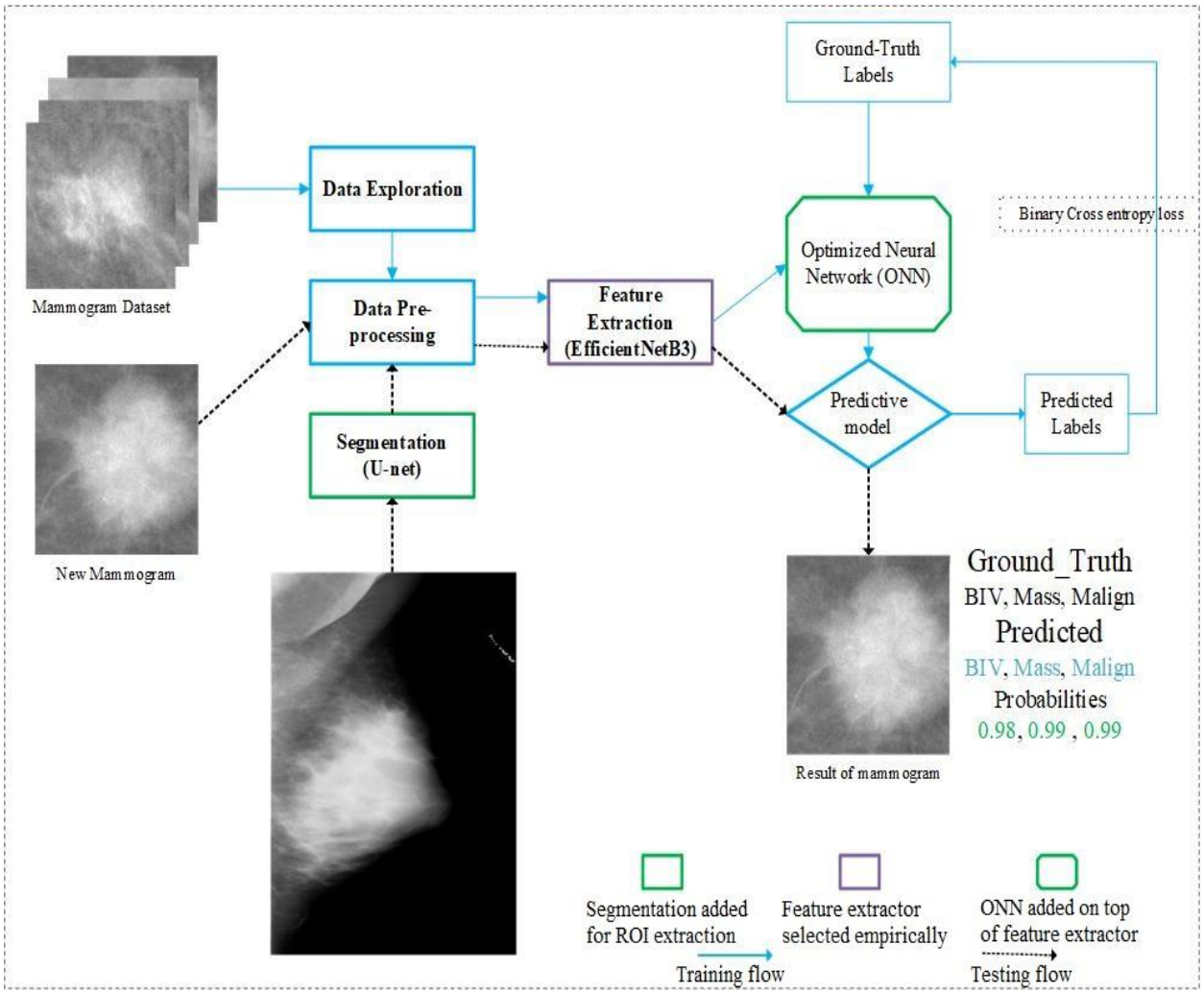


Figure 4.1. Process flow of the proposed system

4.2. Preprocessing Techniques

Preprocessing, after acquiring or collecting a dataset, is the basic process in image classification. To increase the efficiency and accuracy of a CAD system in medical image analysis, preprocessing is a compulsory step while building a dataset. The image format conversion is the first preprocessing task in this works. The ROI extraction, GCN, and Data augmentation are the preprocessing techniques explained here.

A. Image Format Conversion

As explained in section 3.3 in detail, the medical image format conversion is the crucial and first step to work with medical datasets. Table 4.1 shows the image format conversion pseudo code used in this works.

Table 4.1. image conversion pseudo code

Pseudocode: image conversion from dcm to jpg file
Input: path to **S**ource Directory (SD), Target Directory (TD) **Output**: jpg image saved to a target directory **Procedure**: ImgCorversion(SD,TD) 1. Read the image file which is dcm extension 2. Convert the image format from dcm file to jpg file 3. Name the image file as in excel .csv label description name 4. Save the convert image file in .jpg format to a target directory 5. Repeat 1-4 steps for all images

B. ROI extraction

The radiographic image, when captured, contains much amount of useless information, which is unimportant during classification. Therefore, extracting the ROI from the image is curtail preprocessing to focus on important information for classification. The location and surroundings of the lesions must be highlighted by medical imaging radiologists.

CBIS-DDSM dataset has included updated ROIs to guarantee the exact match of ROIs and labels information for multi-label classification. To reduce the risk of mismapping of ROIs and corresponding labels information the same ROIs are taken from the dataset as it is.

Figure 4.2. shows the radiologist mark of ROIs of non-dense first row and dense second row of mammogram images. As it has been seen from the image identifying and marking ROIs of dense mammograms f & h, and it is difficult for copping than that of non-dense mammograms b & d.

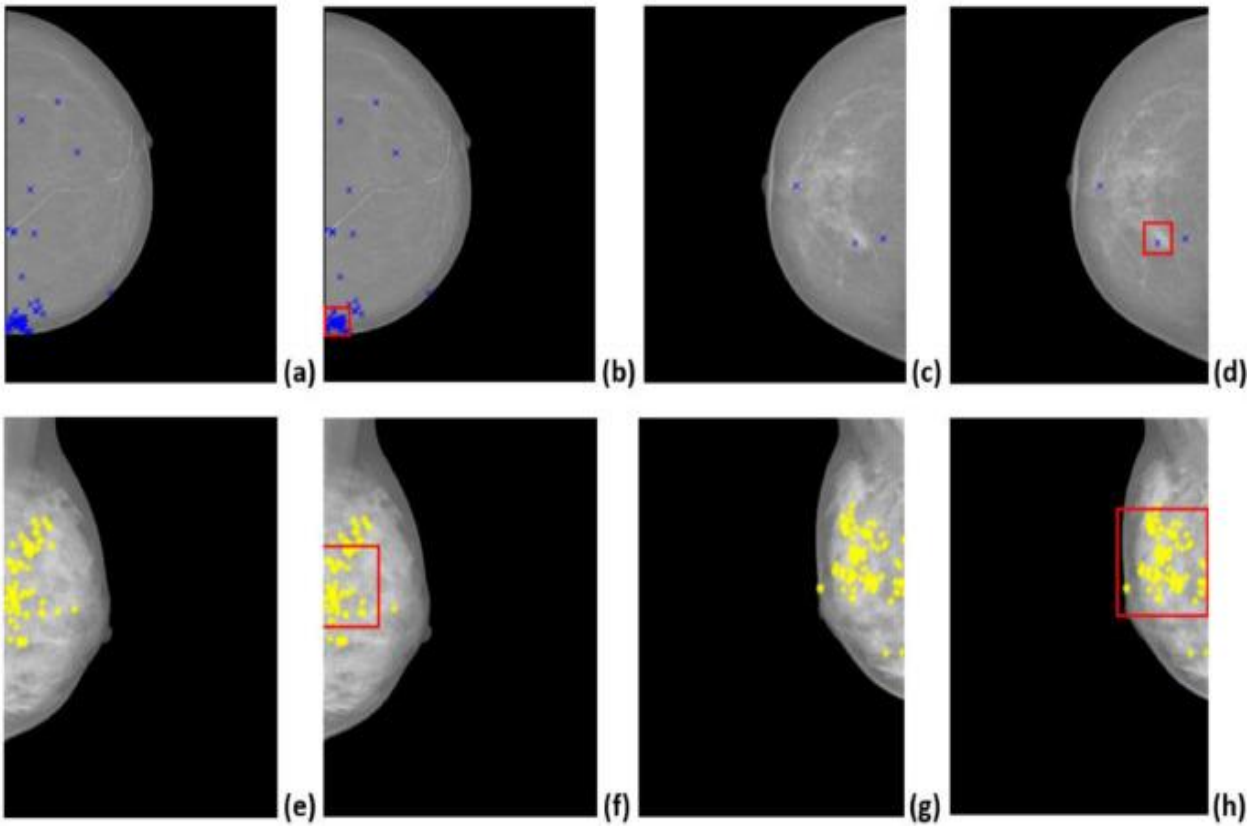


Figure 4.2. ROI from the given images

Adopted from: (Chougrad et al., 2020)

C. Global Contrast Normalization (GCN)

GCN is a common step in medical image classification. It deals with outside sources of data difference such as illumination levels, the different scanners used in the digitalization action, and the effect of this on pixel values. GCN subtracts the mean of intensities after finding it from each pixel of the image. Pseudocode for GCN is given in table 4.2.

Table 4.2. Global contrast normalization pseudo code

Pseudocode: Global Contract Normalization

Input: the numpy array of the input image

Output: Global Contrast Normalized numpy array of image

Procedure: GCN (numpy array of image)

1. Read numpy array of an input image
2. Calculate mean of numpy array (pixel level)
3. Subtract mean from each pixel of the numpy array
4. Output the normalized numpy array to the next layer
5. Repeat 1-4 steps for all images

D. Data augmentation

Deep learning models perform better with a large dataset. The well-known method to make our dataset bigger is data augmentation or jittering. It can raise the amount of the dataset to 10 times or more, which helps to prevent overfitting with very little data. The approach helps to build simpler and robust models that can generalize better.

Data augmentation during training time helps the network to look for a new augmented image in a different iteration. In this work rescale, width, and height shift with 0.2 range, horizontal and vertical flip of data augmentation method is used. Pseudocode for data augmentation is illustrated in table 4.3.

Table 4.3. Pseudocode for data augmentation

Pseudocode: D**ata Augmentation**
Input: matrix of the image in the training process **Output**: Augmented image in each iteration to feed the network **Procedure**: ImageDataGenerator(matrix of image) 1. Read numpy array of an input image 2. Rescale the image by dividing 1 to 255 3. Apply 0.2 range of width shift 4. Apply 0.2 range of height shift 5. Apply horizontal flips to the image 6. Apply vertical flips to the image 7. do the same for every image in the dataset

4.3. Feature Extraction

Feature extraction is an important step in the building of any pattern classification and aims at the extraction of the relevant information that identifies each class. Moreover, it is the process to retrieve the most discriminative data from the raw data. Furthermore, it is finding the set of parameters that define the shape of an image in small dimensions, precisely and uniquely. The goal of feature extraction is to extract a set of features, which increases the acceptance rate with the least amount of instances, and to generate an alike feature set for different images. At this step, important features are filtered from images to form feature vectors. These feature vectors are then used by a fully connected network to classify the input images. It becomes simpler for the network to classify among different labels by these features as it permits fairly easy to distinguish.

Recently, the development of CNNs established impressive efficiency as they grew in depth; producing efficient and high-performance networks. Among CNN architectures this work investigates them empirically and focuses on four architectures.

4.3.1. EfficientNetB3 Architecture

EfficientNetB3 is the feature extractor architecture investigated in this work from empirical experiments. It has 384 layers and is subdivided into blocks. In addition, each block is composed of basic CNN elements like convolution, pooling, batch normalization, and ReLU. Two types of convolutions Conv2D and DepthwiseConv2D are there in it. Three types of pooling which max, average, and global average pooling are multiple pooling categories in efficientnetb3. In addition, the squeeze and excite module is part of efficientnetb3.

It takes 224 x 224 x 3 as input and applies 0-padding. Conv2D is computed followed by pooling operation. 112 x 112 x 40 are convolved and squeeze operation take place giving 1 x 1 x 40 to focus on channel relationship. Excite operation reconstructs spatial resolution from channel information gained with skip connection. Again, convolution is applied on 112 x 112 x 40 and gives 112 x 112 x 24 by setting padding “same” and feed to the second block succeeding to it. BN and ReLU activation take place after each convolution. This is the operation in the first block of efficientnetb3 architecture. It has seven major blocks and these blocks are again subdivided into small subdivisions. This increases the depth of the networks to filters important feature vectors.

Moreover, EfficientNetB3 has 12,320,535 parameters. The output of the last block is 7 x 7 latent information with 1536 filters and after the flattening function, the feature vector extracted is 7 x 7 x 1536 giving 75,264 shapes. Figure 4.3 shows efficientnetb3 with ONN architecture.

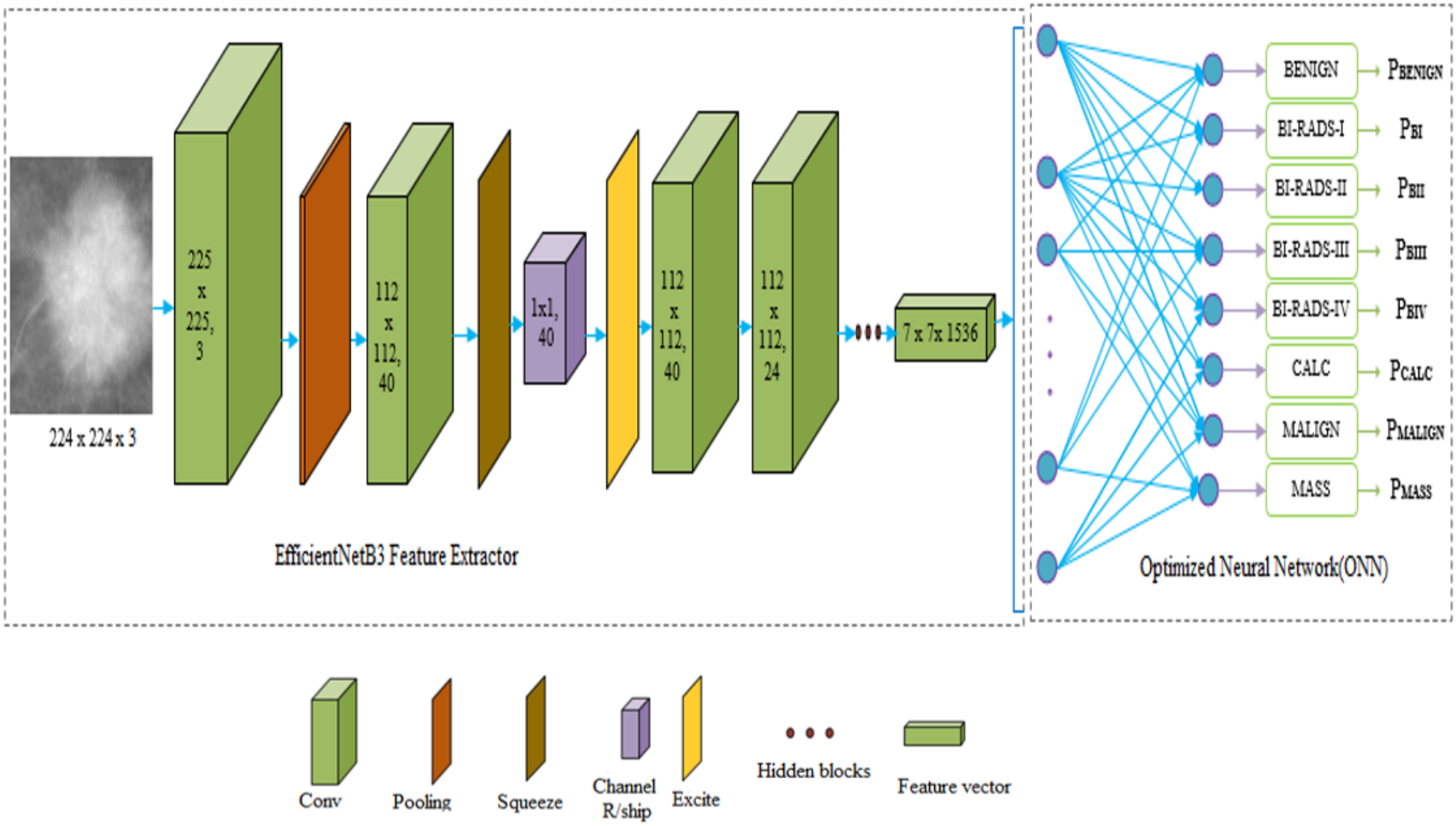


Figure 4.3. EfficientNetB3 architecture with ONN

Conv2D extract feature depending on spatial and channel of the image while squeeze and excitation look for dependencies between the channels. Global average pooling is applied to squeeze and multiplication of the squeezed information is done for exciting back to channel-wise.

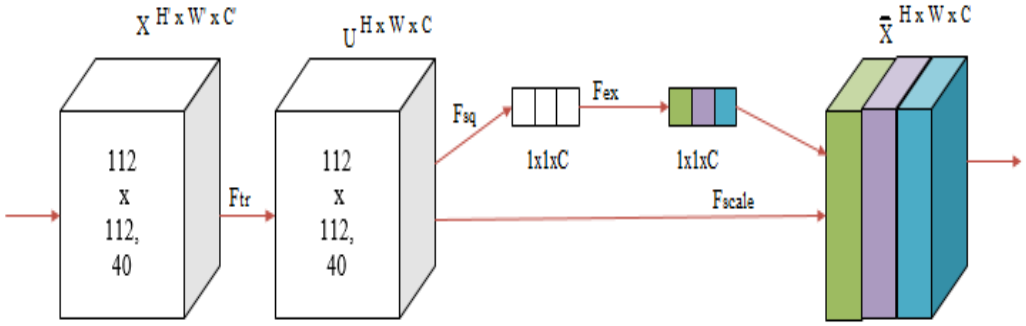


Figure 4.4. Squeeze and excite block

The computation take place in squeeze and excite block is firstly to transform from X input to U output. It is computed by:

$$F_{tr}(X) => U => u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x^s \quad \ldots \quad (4.1)$$

Where * is convolution, $v_c = [\, v_c^1, v_c^2, \ldots v_c^{C'}\,]$ is learned feature, $X = [x^1, x^2 \ldots x^{C'}]$, $u_c \in R^{H \times W}$, v_c^s is a 2D spatial kernel denoting a single channel of v_c.

The Squeeze function is:

$$F_{sq}(u_c) = z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \quad \ldots \quad (4.2)$$

Where z_c is input specific descriptor

The formula for Excitation function is:

$$F_{ex}(z, W) = s = \sigma(W_2 \delta(W_1 z)) \quad \ldots \quad (4.3)$$

Where δ is ReLU activation and σ is sigmoid activation function, $W_1 \in R^{\frac{C}{r} \times C}$, $W_2 \in R^{C \times \frac{C}{r}}$, r is a compression ratio

The reconstruction channel-wise is given by computing this formula.

$$\tilde{X} = F_{scale}(u_c, s_c) \quad \ldots \quad (4.4)$$

where $\bar{X} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_C]$, F_{scale} shows scalar s_c and the feature map u_c channel-wise multiplication.

Optimized Neural Network

Optimized Neural Network (ONN) is a classification part added to the feature extractor to get the best results. It increases the network accuracy. ONN has 128 nodes in a layer next to the feature vector and 8 nodes in the output layer. The order of labels is the same as the order given in this network when the probability of each label is predicted in an array.

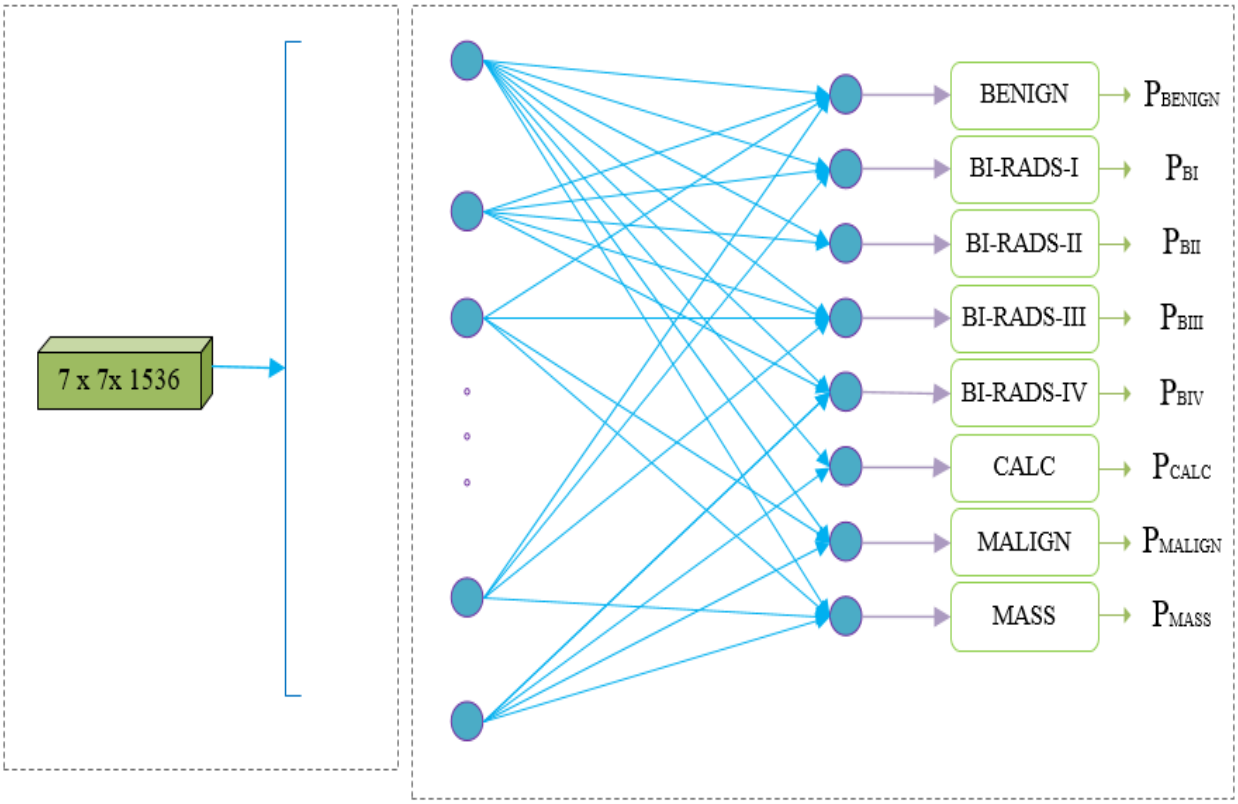


Figure 4.5. ONN

4.3.2. EfficientNetB4 Architecture

EfficientNetB4 is one from a family of EfficientNet. It has some differentiating points from EfficientNetB3. Those are it has a greater number of layers, input image resolution, and numbers of parameters compared to efficientb3. 474, 331 x 331, and 19,466,823 are the values of each attribute respectively. This architecture is used for the case of comparison.

4.3.3. VGG16 Architecture

VGG16 is the CNN architecture used by the baseline paper(Chougrad et al., 2020) for multi-label classification. This model raises the accuracy of the network, at that time, by increasing the depth of networks with very small (3 x 3) convolution filters. It has a stack of 16 layers. The network takes 224 x 224 x 3 RGB images that mean width, height, and channels respectively, and passes to layer one and two which are conv1 & 2 (Conv2D) with the same resolution and 64 filters. After these layers, maxpooling2D followed by two layers with 112 x 112 and 128 filters are added. The other maxpooling2D and three layers with 56 x 56 and 256 filters are added to the network. Again, three layers with 28 x 28 and 512 filters are pushed on the network. Moreover, maxpooling2D and three layers 14 x 14 with 512 filters and the last maxpooling2D added on the network. Figure 4.6 describes this information.

Figure 4.6. VGG architecture.

The convolution stride is set to 1 pixel to scan through all image matrices for capturing important features. The spatial padding of convolution layers is also used to preserve spatial resolution, by fixing padding to “same”. Max-pooling is applied to reduce the spatial resolution of the image by downsizing it by half. It is performed over 2 x 2-pixel window setting stride to 2. After all hidden layers rectified linear unit (ReLU) activation functions are applied.

During training, the convolution performed as dot product is counted as the weight of parameters measure to compare with other networks. VGG16 has 14,714,688 parameters in the feature extraction part and 134M parameters with fully connected networks.

4.3.4. DenseNet121 Architecture

DenseNet(Huang et al., 2017) is a CNN architecture that connects every preceding layer to succeeding layers in feed-forward propagation. The core idea of DenseNet is to guarantee maximum information flow among layers in the network by connecting every layer, that has the same feature-map sizes, directly with each other. Moreover, DenseNet121 architecture has 427 layers which are subdivided into a stack of dense blocks followed by transition layers. A dense block contains a series of building blocks. Each of them consists of two convolutions followed by batch normalization and ReLU activations. The transition layers like pooling and linear transformation with activation function are found between these dense blocks. Furthermore, each building block generates a determined number of feature vectors and which is known as growth rate, and controls the amount of new information that layers can transmit. Table 4.4 shows a detailed explanation of DenseNet-121 architecture in tabular form.

Table 4.4. DenseNet121 architecture

Layer	Output Shape	Kernel size, pooling size, and stride shape
Input	224 x 224, 3	7 *x* 7 conv, stride 2, padding 3
Convolution	112 *x* 112	7 *x* 7 conv, stride 2
Pooling	56 *x* 56	3 *x* 3 max pool, stride 2
Dense-Block (1)	56 *x* 56	$\begin{bmatrix} 1 & x & 1\ conv \\ 3 & x & 3\ conv \end{bmatrix}$ *x* 6
Transition Layer (1)	56 *x* 56	1 *x* 1 conv
	28 *x* 28	2 *x* 2 average pool, stride 2
Dense-Block (2)	28 *x* 28	$\begin{bmatrix} 1 & x & 1\ conv \\ 3 & x & 3\ conv \end{bmatrix}$ *x* 12
Transition Layer (2)	28 *x* 28	1 *x* 1 conv
	14 *x* 14	2 *x* 2 average pool, stride 2
Dense-Block (3)	14 *x* 14	$\begin{bmatrix} 1 & x & 1\ conv \\ 3 & x & 3\ conv \end{bmatrix}$ *x* 24
Transition Layer (3)	14 *x* 14	1 *x* 1 conv
	7 *x* 7	2 *x* 2 average pool, stride 2
Dense-Block (4)	7 *x* 7	$\begin{bmatrix} 1 & x & 1\ conv \\ 3 & x & 3\ conv \end{bmatrix}$ *x* 16
Transition Layer	7 *x* 7 *x 1024*	Flatten()

The model architecture of densenet121 is composed of two major components. The feature extraction is densenet121 itself and classification part. Figure 4.4 shows the high-level abstraction of the model architecture. However, densenet121 is used for comparison in this works.

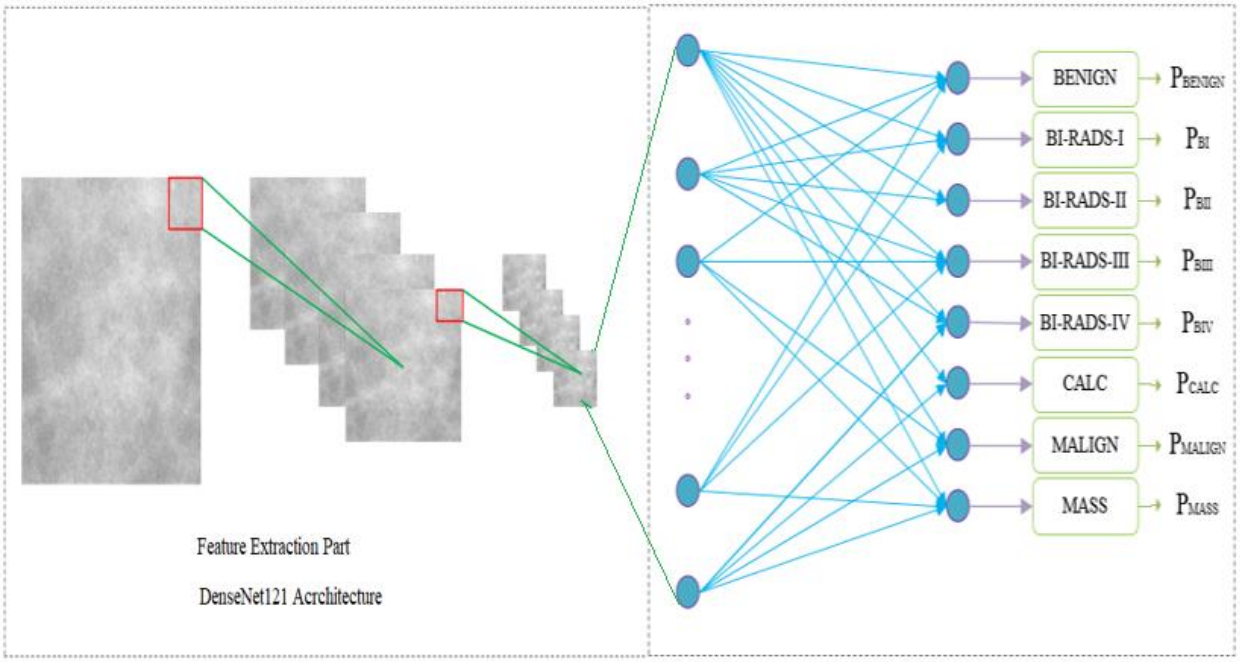


Figure 4.7. A model with densenet121 feature extraction

4.4. Image Segmentation Model

Medical image segmentation shows a significant place in CAD systems for different applications. The huge expense and growth of medical image modality inspire researchers to implement new medical imaging algorithms. Image segmentation is viewed as the most important process because it extracts ROI through a semi-automatic or automated process. It divides the image into various regions according to the specified description, such as body organ/tissue, segmentation for boundary detection, tumor detection/segmentation, and quality inspection in medical applications.

The model used in this work for image segmentation is u-net(Navab et al., 2015). Densenet121 pre-trained weight is used in the decoder part. This can increase the accuracy of the model in extracting ROI. However, this model is not used in model training to focus on full annotated ROI and label description to reduce the limitation that comes from ROI driven from segmentation. But image segmentation is applying while using the model after training. This is to help physicians to identify ROI in mammograms to be cropped for screening.

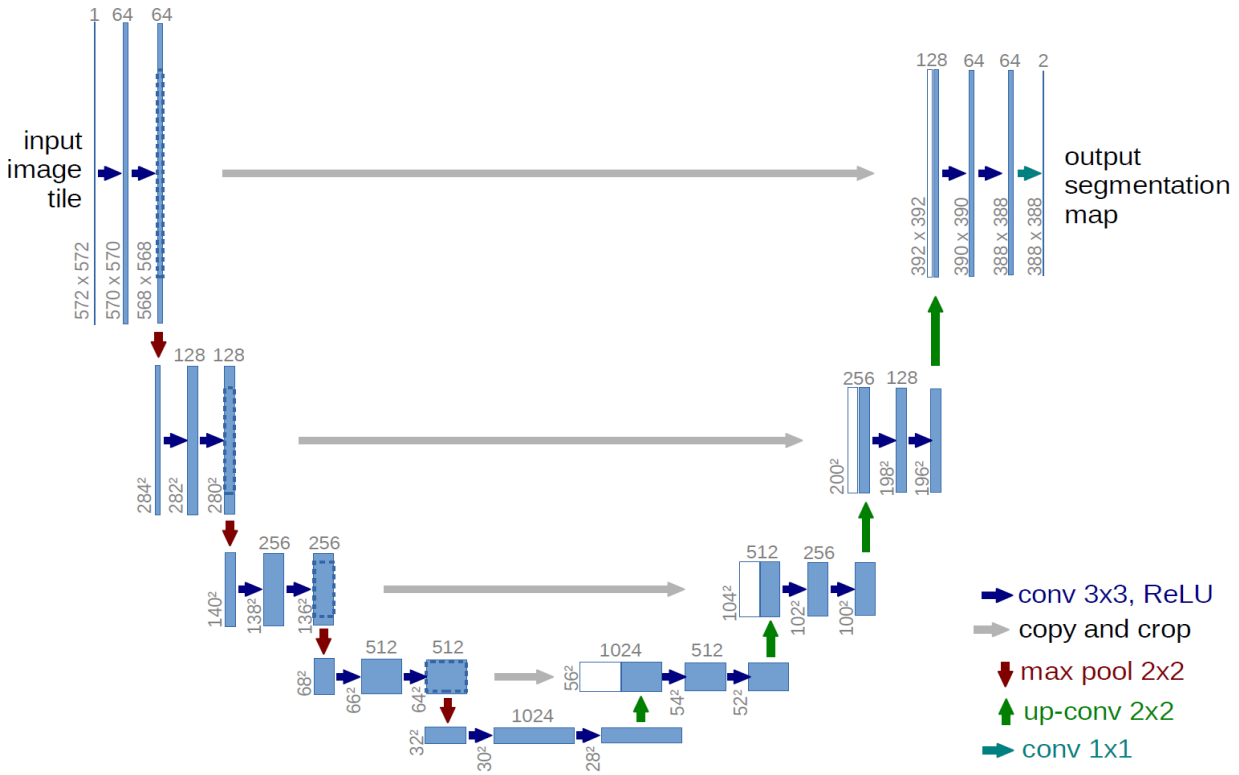


Figure 4.8. The architecture of the u-net

Adopted from: (Navab et al., 2015)

4.5. Activation and Loss Function

An activation function is formulated to handle the nature of non-linearity in the training process. Several activation functions are there and ReLU, and sigmoid are applied in this works.

4.5.1. ReLU activation function

The ReLU is a piecewise linear function. If the input is positive, it will go straight out, otherwise, it will go zero. It is a popular activation function for several types of CNN because the models that use it are easier to train and typically perform better.

$$f(x) = max\ (x,\ 0) \qquad \ldots \qquad (4.5)$$

Where:

x is input value

4.5.2. Sigmoid Activation Function

The sigmoid activation function is a mathematical function with a recognizable "S" shaped curve. It is used for logistic regression and the implementation of basic neural networks. If the classifier is to solve a problem with more than one correct answer, then the Sigmoid function is the correct

choice. It is independently applied to each label of the original output. There are many common sigmoid functions, such as logistic functions, hyperbolic tangent, and arctangent. What makes the sigmoid activation function differ from the softmax activation function is that each label is independent and not summed up to 1. Its formula is:

$$f(x) = \frac{1}{1 + e^{-x}} \quad \ldots \quad (4.6)$$

Where:

x → input vector

e → constant number

4.5.3. Binary Cross-Entropy

The loss gives the understanding of how poor the model performed in each period and how well it performed in a specific data sample. Binary cross-entropy is a loss function used for binary classification tasks. Its formula is:

$$binary\_cross\_entropy = -yilog\widehat{y_i} - (1 - yi)log\ (1 - \widehat{y_i}) \quad \ldots \quad (4.7)$$

Where:

y_i → label

$\widehat{y_i}$ → prediction

CHAPTER 5

5. IMPLEMENTATION OF PROPOSED SOLUTION

5.1. Introduction

In this chapter, the working environment for the proposed solution including hardware utilities and software configuration are addressed. In addition to that, the implementation of the proposed method is given with snippets code and explanation.

5.2. Working Environment

The working environment includes the hardware and software facilities used to succeed in this work and got fruitful results.

5.2.1. Hardware Utilities

In the implementation of this works, a desktop is used for training the models, and for the diagram design and for testing the models another desktop and laptops are utilized.

- The desktop used for training has the following capabilities:
  - Graphics Card Upgrade: GeForce RTX 2070 Super 8 GB RAM
  - Processor: Intel(R) Core i5 dell-3090 CPU, 4 Core(s), 4 Logical Processor(s)
  - Hard-Disk: configured terabyte and added one terabyte for storage.
  - SSD: configured SSD is 500GB.
  - Installed Memory: Physical Memory (RAM) 4.00 GB added 3 RAM, totally configured 16.00 GB RAM.
  - The power supply of GPU:

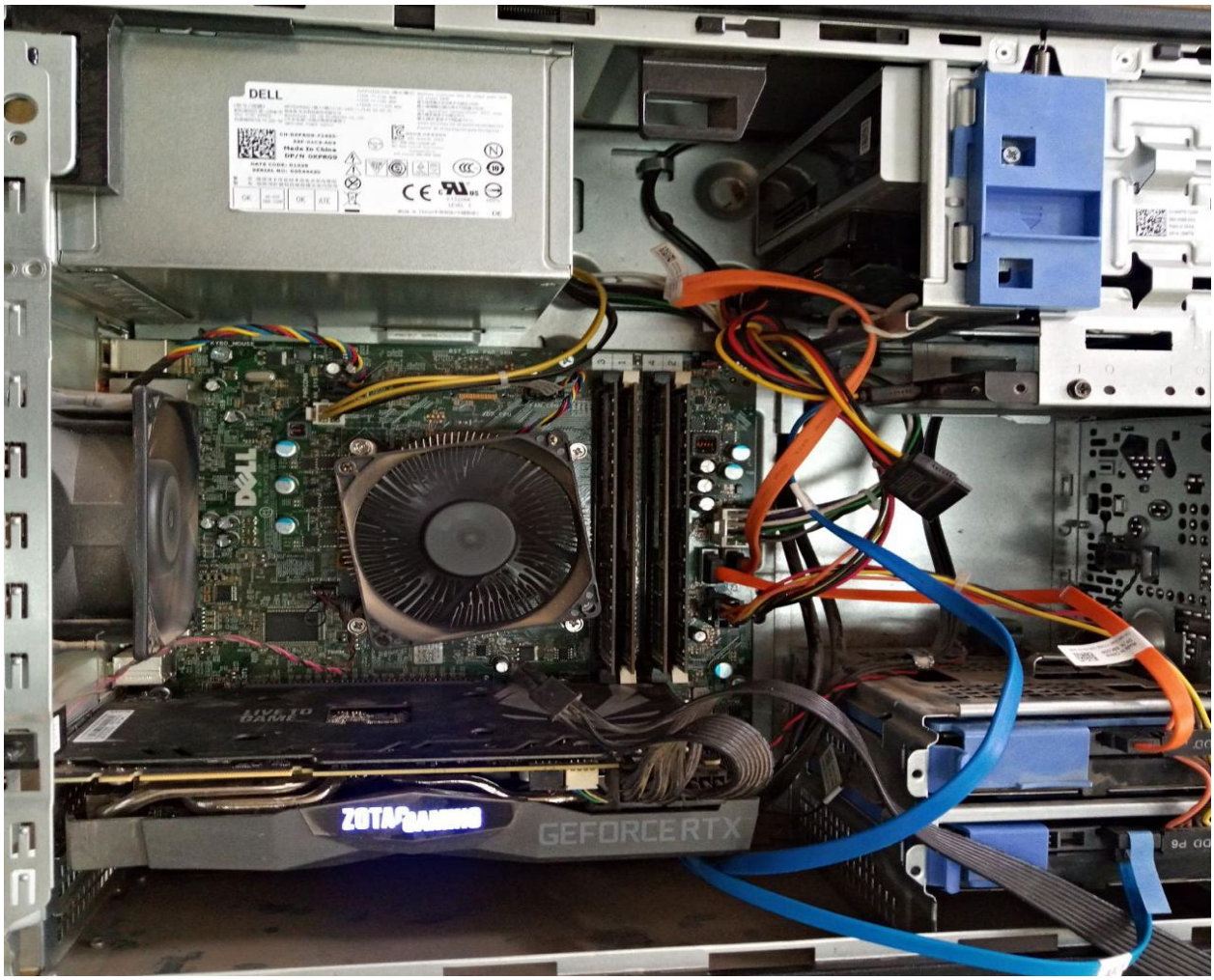


Figure 5.1.Hardware containing graphics card (GPU)

- The desktop used for documentation, diagram design, and testing has the following capabilities:
  - Processor: Intel ® Core™ i7-8700 CPU @ 3.20GHz, 3192Mhz, 6 Core(s), 12 Logical Processor(s)
  - Hard-Disk: configured with one terabyte of storage.
  - Installed Memory: Physical Memory (RAM) 8.00 GB
  - Graphics: Intel ® UHD Graphics 630
- The laptop used for documentation, diagram design, and testing has the following capabilities:
  - Processor: AMD A6-7310 APU with AMD Radeon R4 Graphics, 2000 Mhz, 4 Core(s), 4 Logical Processor(s)

- Hard-Disk: configured with 500 GB of storage.
- Installed Memory: Physical Memory (RAM) 4.00 GB
- Graphics: AMD Radeon™ R4 Graphics

5.2.2. Software Configuration

A. Software configuration for training model in GPU

- Software configuration for training the model on the desktop that contains a Graphics card or GPU is configured with:
  - OS: Microsoft Windows 10 Pro, x64-based PC, Version 10.0.17134 Build 17134
  - System Model: - Dell OptiPlex 3090.
  - GPU optimizer: the latest version of the NVIDIA GPU optimizer and scheduler.
  - CUDA toolkit
  - CuDNN

B. Software configuration for testing and design of a diagram

- Software configuration for testing, diagram design, and documenting the model results on desktop and laptop are configured with:
  - OS: Microsoft Windows 10 Pro, 64-bit (10.0 Build 19042)
  - System Model: - HP Notebook (laptop)
  - System Model: HP 290 G3MT Business PC (Desktop)

C. Application software

Tensorflow: - It is a machine learning platform to build a model and deploy it in client environments like other real-world applications. It supports machine learning, deep learning, and flexible numerical computation. The version of Tensorflow used in this work is 2.4.1.

Python: - Python programming with anaconda cooperation will use for the implementation of the study. The python version is 3.7.6.

Keras**:** is user-friendly, has easy extensibility supports modularity**,** and works with Python as a high-level framework or API. It is more user-friendly when compared to Tensorflow. A current working version of Keras is 2.4.3.

Jupyter Notebook: An interactive web-based application that helps configure, load python API, and write python code.

5.2.3. Proposed Solution Implementation

A. Image conversion snip code.

The proposed solution takes an image in jpg format and the dataset is in dcm format. To implement the proposed method/solution, first, the image is converted in a form accessed by the model. The snip code used to convert from dcm to jpg is:

```
#The path shows how folder embedded in each other
ds = pydicom.dcmread('E:/MassTrain/Mass-
Training_P_00001_LEFT_CC_1/07-20-2016-DDSM-20213/1.000000-
cropped images-58924/1-1.dcm')
#Convert the values into float
new_image = ds.pixel_array.astype(float)
scaled_image = (np.maximum(new_image, 0) / new_image.max()) *
255.0
scaled_image = np.uint8(scaled_image)
final_image = Image.fromarray(scaled_image)
final_image.show() # Display the Image
# Save the image as JPG
final_image.save('…Mammogram/image/Train/P_00001_1.jpg')
```

Snip Code 5.1. Format conversion

B. Global Contrast Normalization

To enhance the image one preprocessing method that is global contrast normalization is performed. The snip code of GCN is as follows:

```
def GCN(filename, s, lmda, epsilon):
    X = numpy.array(Image.open(filename))
    # replacement for the loop
    X_average = numpy.mean(X)
    print('Mean: ', X_average)
    X = X - X_average
    contrast = numpy.sqrt(lmda + numpy.mean(X**2))
    X = s * X / max(contrast, epsilon)
    # Save the file
    imageio.imwrite("E:/Mammogram/image/Train/P_00001_1.jpg", X)
GCN("E:/DDSM/image/Train/P_00001_1.jpg", 1, 10, 0.000000001)
```

Snip Code 5.2. GCN snip code

C. Dataset Loading and Integration with label description

Having enhanced the dataset in a proper format it is loaded to the model. The label detail of the dataset is in csv format and read separately. The image is integrated with its labels and then represented in the form of numpy array for next processing. The following snip code is used to process what is explained here.

```
#Loading Mammogram Images
TRAIN_PATH = 'E:/Mammogram/image/Train/'
TEST_PATH = 'E:/Mammogram/image/Test/'

#Reading the label's description of the image
TRAIN_CSV_PATH = 'E:/Mammogram/image/train_set.csv'
TEST_CSV_PATH = 'E:/Mammogram/image/test_set.csv'

#Construct data frames holding training and test data information
df_train = pd.read_csv(TRAIN_CSV_PATH)
df_test = pd.read_csv(TEST_CSV_PATH)

#Integretion of Image with their labels
df_train['patient_id'] = df_train['patient_id'].astype(str)+'.jpg'
df_test['patient_id'] = df_test['patient_id'].astype(str)+'.jpg'
```

Snip Code 5.3. Loading and integrating data with labels

5.2.4. F1_score custom objects

Accuracy is a commonly used parameter to monitor the progress of training in DL multi-class classification problems. However, using accuracy to monitor training progress in multi-label classification, specifically in this work, is impossible. Because it does not show increasing progress as the epoch of the training increase. The customized f1_score solve the problem and improve the model performance as general in multi-label classification problem. The snip code below shows the customize f1_score used in this works.

```
# Custom fbeta metrics for evaluation
def f2_score(y_true, y_pred):
    y_true = tf.cast(y_true, "int32")
    # implicit 0.5 threshold via tf.round
    y_pred = tf.cast(tf.round(y_pred), "int32")
    y_correct = y_true * y_pred
    sum_true = tf.reduce_sum(y_true, axis=1)
    sum_pred = tf.reduce_sum(y_pred, axis=1)
    sum_correct = tf.reduce_sum(y_correct, axis=1)
    precision = sum_correct / sum_pred
    recall = sum_correct / sum_true
    f_score = 5 * precision * recall / (4 * precision + recall)
    f_score = tf.where(tf.math.is_nan(f_score),
      tf.zeros_like(f_score), f_score)
    return tf.reduce_mean(f_score)
```

Snip Code 5.4. Customized f1_score

Hyperparameter is a configuration that is external to the model. However, it has a direct effect on the model and its value is also not forecasted from the data. During training, much amount of time is spent fine-tuning those hyperparameters to get a better model. To find the best performance giving hyperparameter, switching from one to another is exercised. For instance, to select an optimizer the model trained with Adam, SGD. After many experiments have been conducted the hyperparameter with the best result in this works are given in table 5.1.

Table 5.1. Hyperparameter configuration

Configuration	Value
Optimization function	Adam
Epoch	90
BatchSize	16
Learning rate	10^{-4}
Batch Normalization	True
Drop out	0.2
reduceLROnPlateau	monitor='loss', factor=0.1, patience=3, mode='min'
ModelCheckPoint	monitor='f2_score', save_best_only=True, save_weights_only=False, mode='max'

Cross-validation is a statistical method used to estimate the capability of the deep learning model while training going on. It is applied by dividing the training data into 5 folds and 80% is for training and 20% for validation and checking predictive results to fix the problem of overfitting. The training takes place 5 times to change each fold as validation data per training.

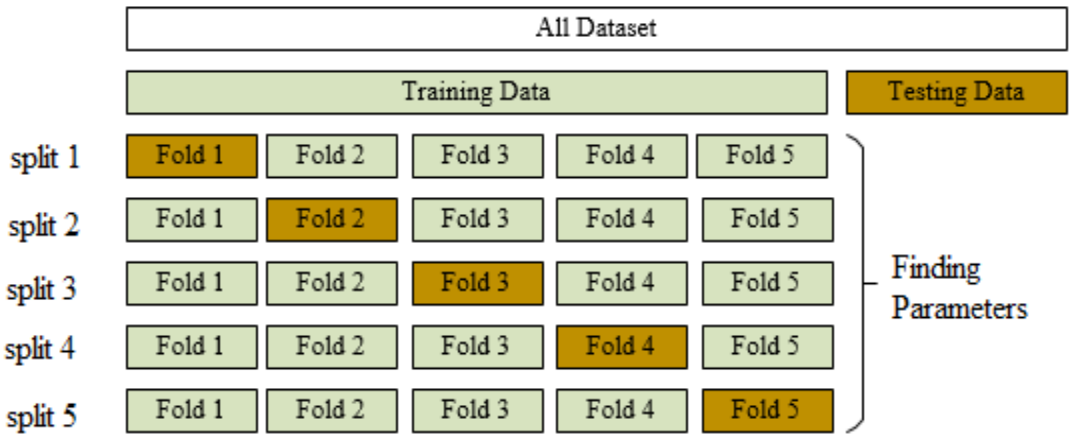


Figure 5.2. 5-fold cross-validation

The block diagram in figure 5.2. show the 5-fold cross-validation used in this works to get the advantage to afford by it. The following snip code illustrates its implementation in code.

```
# 5-fold is used for training
folds = KFold(n_splits=5, shuffle=True,
random_state=1).split(X_train_files, y_train)
for train_index, val_index in folds:
    X_train_files_fold = X_train_files[train_index]
    y_train_fold = y_train[train_index]
    X_val_files_fold = X_train_files[val_index]
    y_val_fold = np.array(y_train[val_index])
```

Snip Code 5.5. 5-fold snip code

Model training starts by running compile and fit-function. This is a snip code for compile and fits function.

```
# compile and fit the model
adam = Adam(learning_rate=10^-4)
model.compile(loss='binary_crossentropy', optimizer=adam,
 metrics=[f2_score])
model.fit(train_generator, epochs=EPOCHS,
validation_data=val_generator,  callbacks=callbacks)
```

Snip Code 5.6. Snip code for compile and fit

5.2.5. Model Implementation

In this work, experiments are conducted on Keras applications that are DL models which are supplied with pre-trained weights. After the empirical experiments, four models were implemented and the details are given below.

A. EfficientNetB4 Implementation

The efficientNetB4 model seems to have good f1_score accuracy. However, it needs a large amount of dataset unfortunately it's affected with high overfitting with a small dataset. Its snip code is given below:

```
# Transfer learning model -EfficientNetB4
def effnetb4_model():
    effnetb4 = EfficientNetB4(include_top=False,
    weights="imagenet", input_shape=INPUT_SHAPE)
    model = Sequential([
        base_model_efficientnet,
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.2),
        Dense(8, activation='sigmoid')
    ])
    return model
model = effnetb4_model()
```

Snip Code 5.7. Efficientnetb4 snip code

B. VGG16 Implementation

VGG16 is the baseline paper model and it follows the procedure explained in that paper. The feature extraction part and classification part are implemented as original papers(Simonyan & Zisserman, 2015) except for output layers. 15 epochs were trained using imagenet weight to initialize the kernel with Adam optimizer. Again, 90 epochs were trained by using weight trained on the first 15 epochs and SGD optimizer used. Its snip code is given below.

```
def vgg16_model():
    base_model_vgg = VGG16(include_top=False,
   weights="imagenet", input_tensor=None, input_shape=(224,
   224, 3))
    model = Sequential([
        base_model_vgg,
        Flatten(),
        Dense(4096, activation='relu')
        Dropout(0.2),
        Dense(4096, activation='relu')
        Dropout(0.2)
        Dense(8, activation='sigmoid')
        ])
    return model
model = vgg16_model()
```

Snip Code 5.8. VGG snip code

C. *DensenetNet121 Implementation*

Densenet121 is the second better result-producing model among deep learning models available on keras application for this specific work. The snip code for densenet121 is given as follows.

```
# Transfer learning model -DenseNet121
def dnet121_model():
    dnet121 = DenseNet121(weights='imagenet',
    include_top=False,  input_shape = INPUT_SHAPE)
    model = Sequential([
        dnet121,
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.2),
        Dense(8, activation='sigmoid')
    ])
    return model
model = dnet121_model()
```

Snip Code 5.9. Densenet121 snip code

D. *EfficientNetB3 + ONN implementation*

EfficientNetB3 is the model that gives the best result in this works. Its snip code showing transferable form is given below.

```
# Transfer learning model - EfficientNetB3
def effnetb3_model():
    effnetb3 = EfficientNetB3(include_top=False,
    weights="imagenet", input_shape=(224, 224, 3))
    model = Sequential([
        effnetb3,
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.2),
        Dense(8, activation='sigmoid')
    ])
    return model
model = effnetb3_model()
```

Snip Code 5.10. Efficientnetb3 snip code

5.2.6. Image Segmentation Implementation

The u-net model is implemented to assist physicians while using the model to identify and crop the ROI from a full mammogram. The pre-trained weight of densenet121 is added to the encoder block of the u-net. The convolution and decoder block give the latent to compare mask and images. *The sn*ip code of convolution and decoder:

```
def conv_block(inputs, num_filters):
    x = Conv2D(num_filters, 3, padding="same")(inputs)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    x = Conv2D(num_filters, 3, padding="same")(x)
    x = BatchNormalization()(x)
    x = Activation("relu")(x)

    return x

def decoder_block(inputs, skip_features, num_filters):
    x = Conv2DTranspose(num_filters, (2, 2), strides=2,
padding="same")(inputs)
    x = Concatenate()([x, skip_features])
    x = conv_block(x, num_filters)
    return x
```

Snip Code 5.11. Snip code for the decoder

After comparing the mask with an image to identify ROI the reverse operation takes place to show the region in full image. Decoder block is applied to reconstruct the encoded features and concatenation block helps to get information from the previous convolution with the same shape. Densenet121 pre-trained weight is added to increase the accuracy of the model. This is explained with the snipped code below.

```
def build_densenet121_unet(input_shape):
    """ Input """
    inputs = Input(input_shape)

    """ Pre-trained DenseNet121 Model """
    densenet = DenseNet121(include_top=False, weights="imagenet",
input_tensor=inputs)

    """ Encoder """
    s1 = densenet.get_layer("input_1").output       ## 512
    s2 = densenet.get_layer("conv1/relu").output    ## 256
    s3 = densenet.get_layer("pool2_relu").output    ## 128
    s4 = densenet.get_layer("pool3_relu").output    ## 64

    """ Bridge """
    b1 = densenet.get_layer("pool4_relu").output  ## 32

    """ Decoder """
    d1 = decoder_block(b1, s4, 512)             ## 64
    d2 = decoder_block(d1, s3, 256)             ## 128
    d3 = decoder_block(d2, s2, 128)             ## 256
    d4 = decoder_block(d3, s1, 64)              ## 512

    """ Outputs """
    outputs = Conv2D(1, 1, padding="same", activation="sigmoid")(d4)

    model = Model(inputs, outputs)
    return model
```

Snip Code 5.12. Tre-trained and encoder snip code

5.2.7. Using the Model to Screen an Image

To use the model adding the customized object is mandatory unless an error is generated. The f1_score custom object is integrated with a model while loading the model. The image to be screened must be resized to 224 x 224 x 3. The snip code for this step is:

```
# The model is loaded along with F1_score custom objects
model = load_model('E:/Model/ENETB3/efficientnetb3.h5',
custom_objects = {'f2_score':f2_score})
# The image is loaded for screening
img = image.load_img(E:/Mammogram/image/Test/p_00038_1.jpg',
target_size=(224, 224, 3))
```

Snip Code 5.13. Model loading

The image is converted to numpy are and normalized. This step is done by this piece of code.

```
# The image is converted in to numpy array and normalized
img = image.img_to_array(img)
img = img/255.
plt.imshow(img)
img = np.expand_dims(img, axis=0)
```

Snip Code 5.14. Normalizing snip code

The model gives the probabilities of all labels. Then, these probabilities are separated into subcategories depending on the position in the predicted list. The order is the same as shown in figure 4.9.

```
# The Label of image is predicted
y_pred = model.predict(img)
#The probabilities of all labels
y_preds = y_pred[0]
#The probabilities of subcatecory labels
density = [y_preds[1], y_preds[2], y_preds[3], y_preds[4]]
pathology = [y_preds[0], y_preds[6]]
finding = [y_preds[5], y_preds[7]]
```

Snip Code 5.15. Categorizing predictive value snip code

From the density subcategory, one label with the highest probability at a time is selected along with its probability. This is done with the snipped code given below.

```
# The maximum probabilities of out of four labels are selected
dens = max(density)
de = density.index(dens)
dens = round(dens, 2)
if de == 0:
    d = 'BI'
elif de == 1:
    d = 'BII'
elif de == 2:
    d = 'BIII'
else:
    d = 'BIV'
```

Snip Code 5.16. Selecting the highest value for density

Finding has two labels and one with the greater probability is selected while testing. This is returned from this piece of code.

```
# The maximum probabilities of out of two labels are selected
find = max(finding)
fi = finding.index(find)
find = round(find, 2)
if fi == 0:
    f = 'CALC'
else:
    f = 'MASS'
```

Snip Code 5.17. Selecting for finding

Lastly, benign or malign is selected with the bigger probability for the image screened. Finally, three labels one from each subcategory characterize one image at a time. The snip code for pathology is given below.

```
# The maximum probabilities of out of two labels are selected
patho = max(pathology)
pa = pathology.index(patho)
patho = round(patho, 2)
if pa == 0:
    p = 'BENIGN'
else:
    p = 'MALIGN'
```

Snip Code 5.18. Selecting for pathology

CHAPTER SIX

6. RESULTS AND DISCUSSIONS

6.1. Chapter Overview

This chapter gives the details of experimental results for this work. The image conversion and result of segmentation that helps for cropping ROI while using the model are included. The experimental results include the result of efficientnetb4, vgg16, densenet121, and efficientnetb4 are given in detail. Discussions of the results are explained at the last.

6.2. Experimental Results

This section gives details of each model's performance from the experimental analysis. Before explaining each model's experimental results, the result of image conversion and the benefit of using ROI given by dataset is briefed. Converting from dcm to jpg format is straightforward. But getting the right ROI from image segmentation is hard, especially when the breast is dense.

Figure 6.1. shows non-dense breast in full mammogram and its ROI as specified by special radiologist. It is easily understood where the ROI is located in a non-dense breast mammogram.

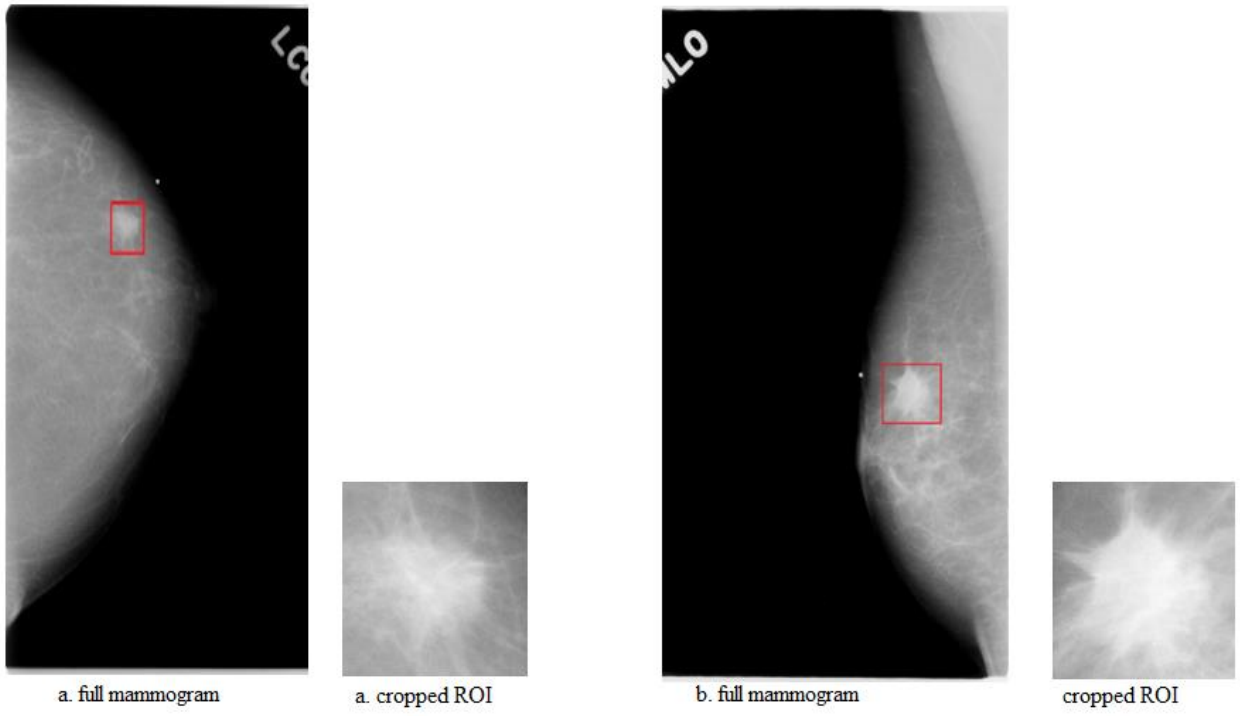


Figure 6.1. Full mammogram of non-dense and its ROIs

On the other hand, in dense breasts identifying small tumors in the dense tissue of the breast is very difficult. The dense breast mammogram looks white and finding its ROI is even tricky for the radiologist. Therefore, in this work ROI given in the dataset is used rather than getting them by using the segmentation model for training. Figure 6.2 visualize the full dense mammogram and its ROIs.

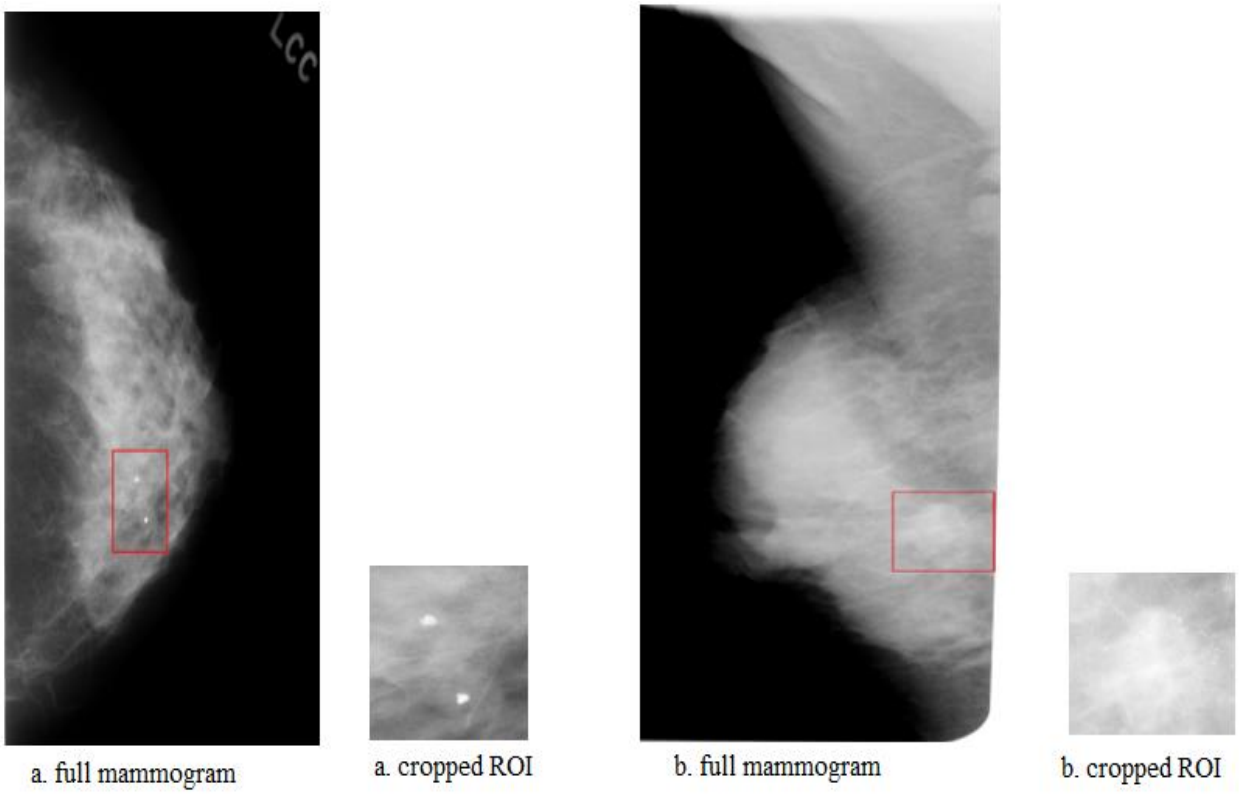


Figure 6.2. Full mammogram of dense and its ROIs

Furthermore, the ROI found within the dataset has a different resolution. However, the model processes all images in the same resolution which is 224 x 224 in this work. Bicubic interpolation is used to reshape the image. Figure 6.3 shows the results of two different resolutions after resizing.

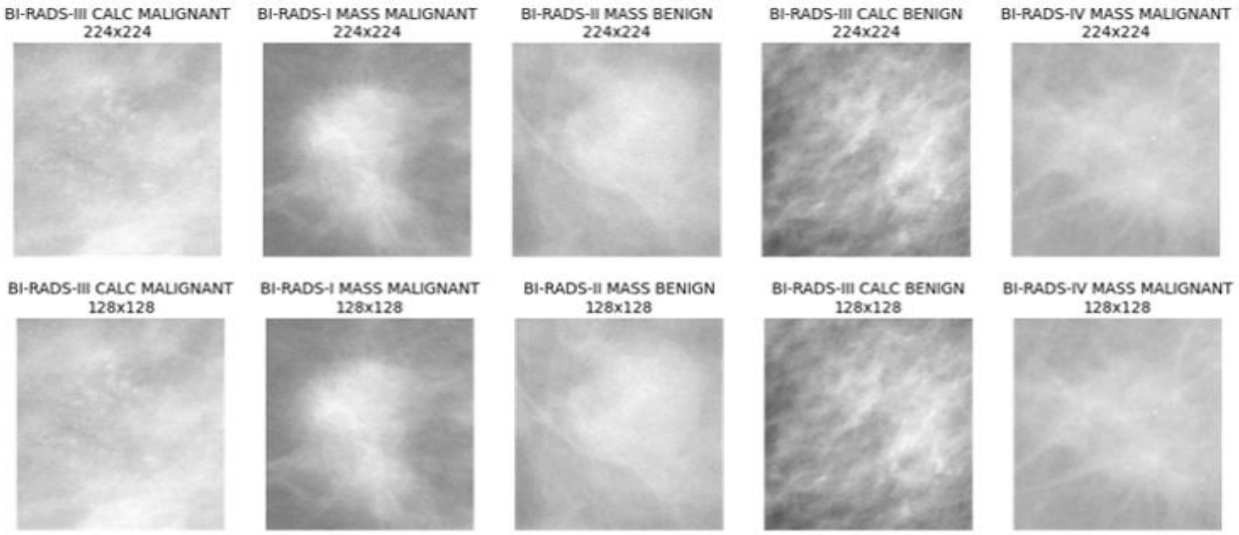


Figure 6.3. ROIs in dataset which resized in different resolutions

6.2.1. Segmentation Result

Medical image segmentation is a tool that helps to detect and crop the tumor to be classified as benign or malign. U-net with densenet121 pre-trained weight in encoder block was trained on the specified dataset. To help the user of this work, this architecture was incorporated to detect the ROI to be cropped. However, as shown in figure 6.1. and 6.2. the ROI manually cropped and annotated for multi-label classification dataset is perfect and used for model training. Figure6.4 shows the result of segmentation.

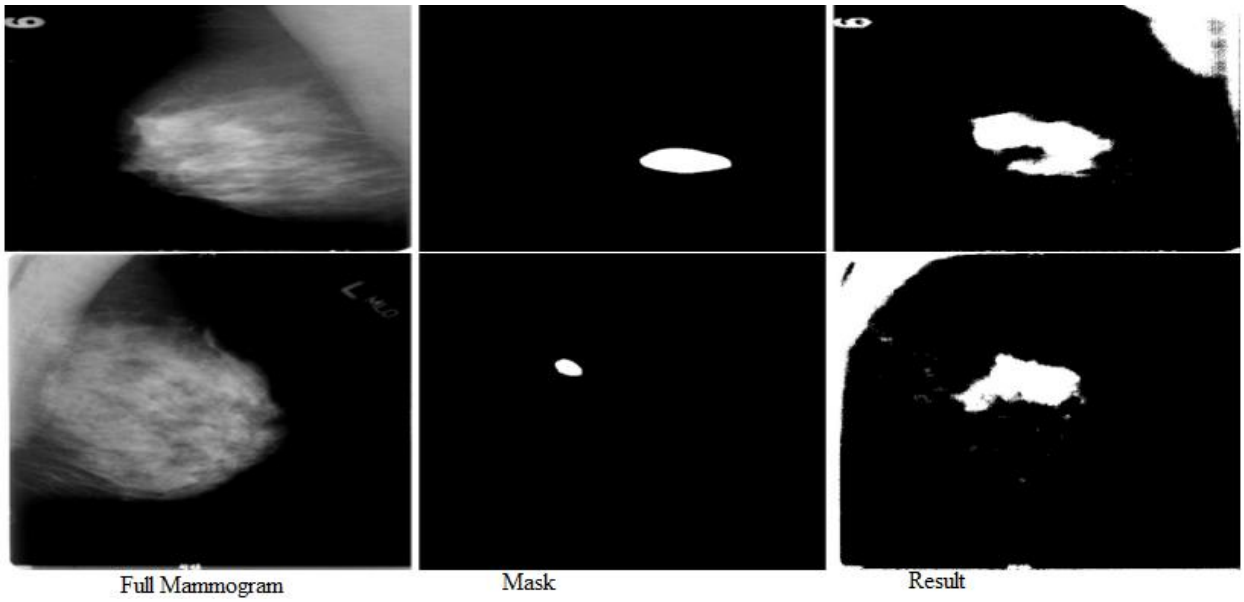


Figure 6.4. Sample of u-net segmentation results

6.2.2. EfficientNetB4 Result

The hyperparameter configuration has a direct effect on the performance of the model during the training time of the neural network. Efficientnetb4 is implemented with a 10^{-4} learning rate, 0.2 dropouts, and trained for 90 epochs. The experiment result of efficientnetb4 on CBIS-DDMS is 98% accuracy of customized f1_score as shown in figure 6.5 accuracy and loss graph. 0.66 f1-score, 0.27 hamming loss, 6.72 coverage error, o.575 is the result of the model on evaluation metrics.

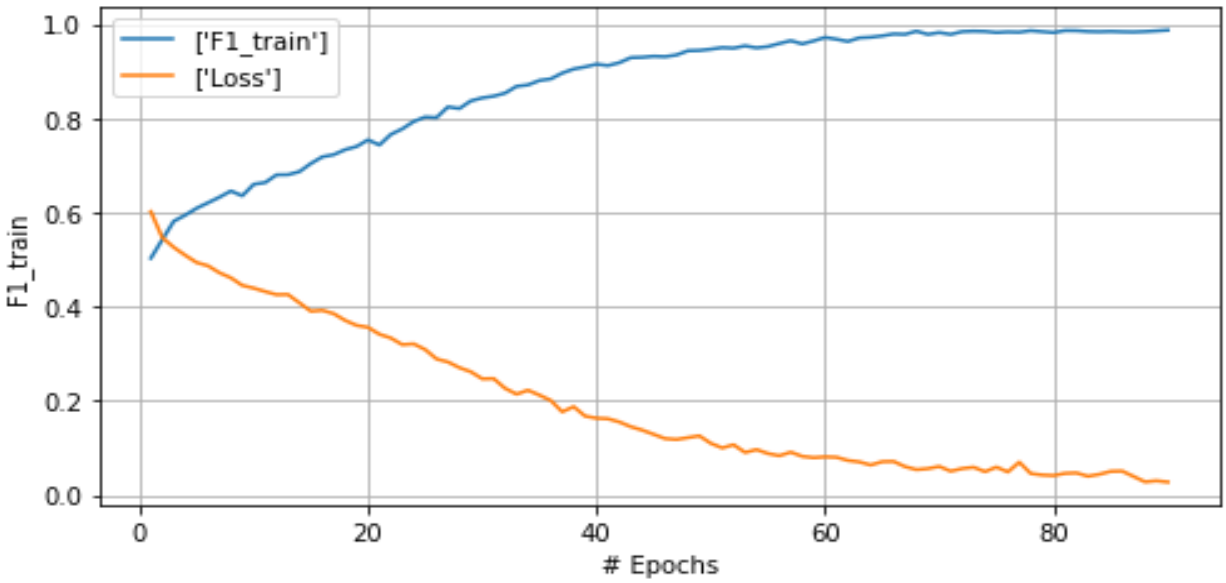
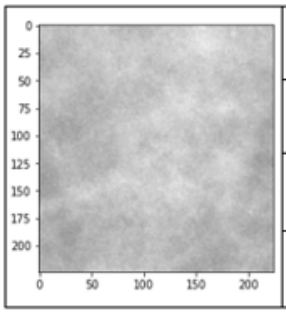
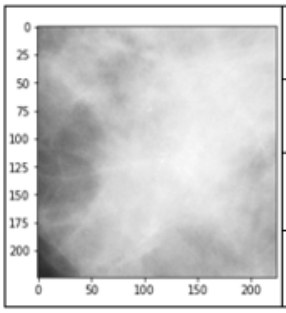


Figure 6.5. Training accuracy and loss of efficientNetb4

This result indicates the overfitting found in this model because of the high computing parameters of the model on a small dataset. Figure 6.6 shows ground truth, probabilities, and predictive (label) results of two samples benign and malign mammogram. On the upper image, the model predicts correctly as ground truth but with low probabilities. However, on the lower image, the model not predicts two subcategories density and finding as ground truth.



Category	Benign	BI	BII	BIII	BIV	Calc	Malign	Mass
Ground_Truth	1	0	1	0	0	1	0	0
Probability	0.606	0.152	0.382	0.334	0.178	0.633	0.389	0.362
Label	1	0	1	0	0	1	0	0



Category	Benign	BI	BII	BIII	BIV	Calc	Malign	Mass
Ground_Truth	0	0	0	1	0	1	1	0
Probability	0.264	0.143	0.385	0.321	0.143	0.147	0.739	0.853
Label	0	0	1	0	0	0	1	1

Figure 6.6. Benign and malign sample predicted by efficientnetb4 model.

6.2.3. VGG16 Result

VGG16 experiment is conducted by using all hyperparameter and implementation details as given in baseline works. The first 15 epochs were trained using imagenet weight to initialize the kernel with Adam optimizer. Another, 90 epochs were trained by using weight trained on the first 15 epochs with SGD optimizer. All layers of the architecture are fully fine-tuned. The learning rate used is 1e-2 and dropout of rate 0.2. Figure 6.7 shows 77.8% training accuracy of customized f1_score and loss of vgg16.

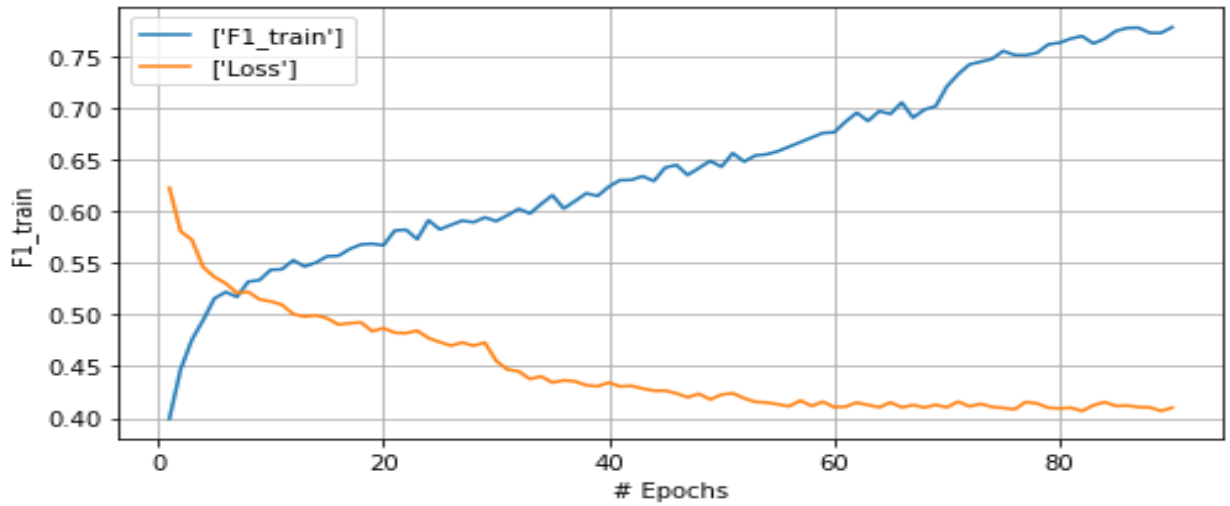
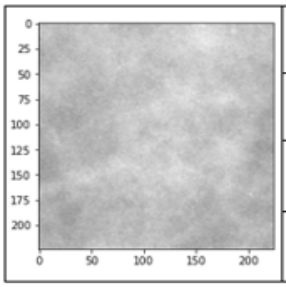


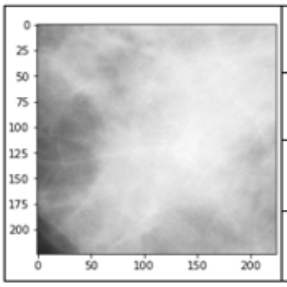
Figure 6.7. Training accuracy and loss of vgg16

The experiment result of vgg16 on the CBIS-DDMS dataset measured using evaluation metrics are 0.83 f1-score, 0.132 hamming loss, 5.88 coverage error, and 0.72 exact match results.

As seen from figure 6.8 on the upper image the model does not predict correctly as ground truth on finding subcategories. But on the lower image, the model predicts the image label correctly as ground truth with low probability except for malign.



Category	Benign	BI	BII	BIII	BIV	Calc	Malign	Mass
Ground_Truth	1	0	1	0	0	1	0	0
Probability	0.729	0.249	0.649	0.047	0.010	0.026	0.348	0.961
Label	1	0	1	0	0	0	0	1



Category	Benign	BI	BII	BIII	BIV	Calc	Malign	Mass
Ground_Truth	0	0	0	1	0	1	1	0
Probability	0.024	0.027	0.206	0.447	0.304	0.477	0.970	0.471
Label	0	0	0	1	0	1	1	0

Figure 6.8. Benign and malign sample predicted by vgg16 model.

6.2.4. DenseNet121 Result

The densenet121 has also the same hyperparameter configuration described in table 5.1. Feature map 7 x 7 x 1024 are extracted. It is a small feature extracted and decreases the computational performance of the model in the training time of the NN. The performance of the model is better on the experiment result next to efficientnetb3. In terms of computational cost, it is best among the performed experiment. 87% accuracy of customized f1_score and loss are shown in figure 6.9 for densenet121 training.

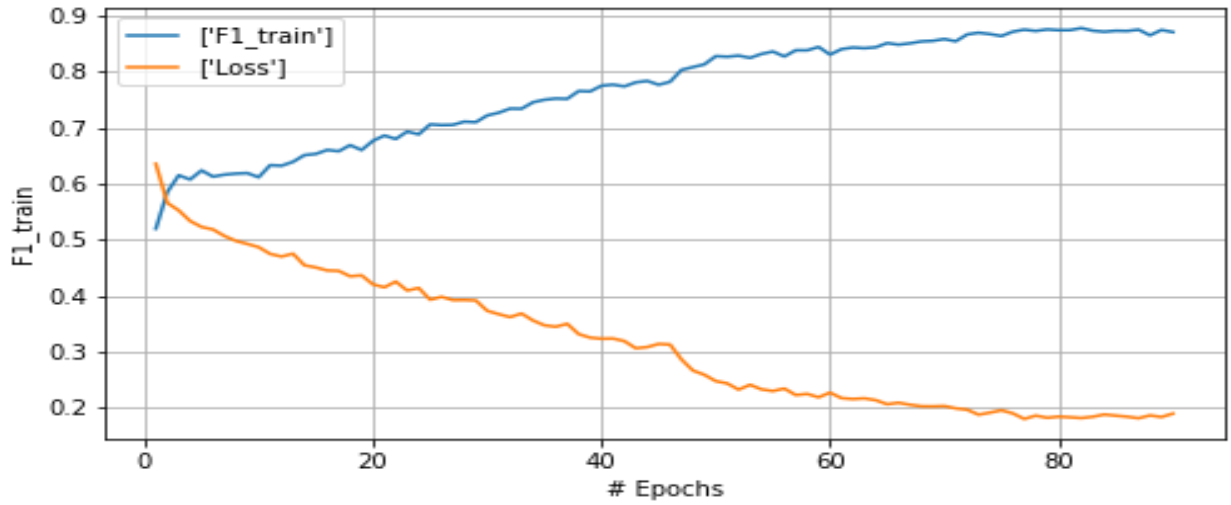
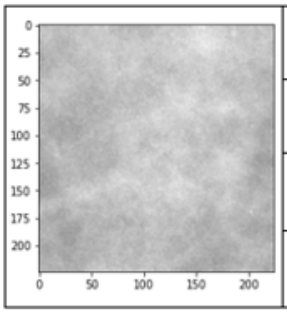


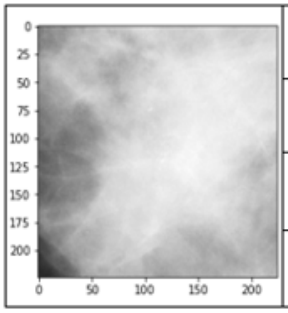
Figure 6.9. Training accuracy and loss of densenet121

The evaluation metrics results from densenet121 on CBIS-DDMS are 0.90 f1_score, 0.090 hamming loss, 5.01 coverage error, and 0.76 exact match results. This result indicates better performance next to efficientnetb3.

Figure 6.10 shows ground truth, probabilities, and predictive (label) results of two samples benign and malign mammogram of this model. On the upper image, the model predicts correctly as ground truth but with high probabilities. On the lower image also, the model predicts as ground truth. However, on finding subcategories probabilities are closer.



Category	Benign	BI	BII	BIII	BIV	Calc	Malign	Mass
Ground_Truth	1	0	1	0	0	1	0	0
Probability	0.978	0.274	0.858	0.122	0.179	0.999	0.183	0.240
Label	1	0	1	0	0	1	0	0



Category	Benign	BI	BII	BIII	BIV	Calc	Malign	Mass
Ground_Truth	0	0	0	1	0	1	1	0
Probability	0.004	0.041	0.386	0.542	0.065	0.559	0.994	0.515
Label	0	0	0	1	0	1	1	0

Figure 6.10. Benign and malign sample predicted by densenet121 model.

6.2.5. EfficientNetB3 + ONN Result

In the first phase of this works after an empirical survey, efficientnetb3 is identified as the best model for this works. Following that, different fine-tuning is conducted and the hyperparameter found in table 5.1 is identified. The classification part of the model as shown in figure 4.9 has two layers with 128, 8 nodes in the hidden and output layer respectively. 98.4% customized f1_score accuracy and loss of efficientnetb3 are seen in figure 6.11.

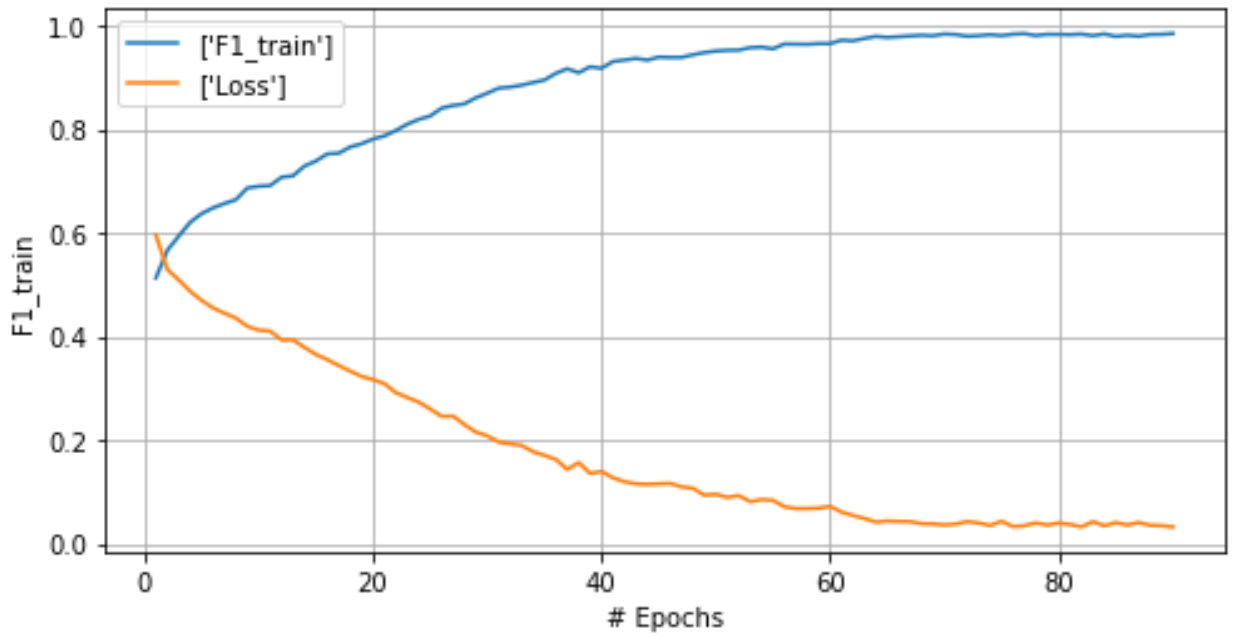
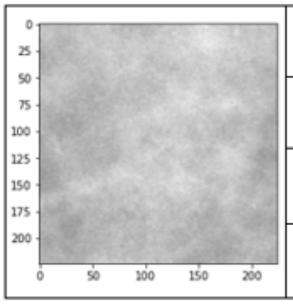
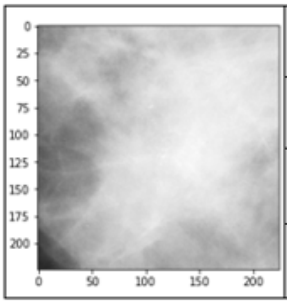


Figure 6.11. Training accuracy and loss of efficientnetb3

The experiment result of efficientnetb3 on CBIS-DDMS is the best result investigated by this works. Evaluation metrics results are 0.94 f1-score, 0.044 hamming loss, 3.72 coverage error and 0.81 exact match. This result indicates efficientnetb3 is the optimized model for the used dataset for multi-label classification problems. As visualized in figure 6.12 the model predicts the image of benign and malign mammograms as the ground with best probabilities than all of the experiments conducted for this work.



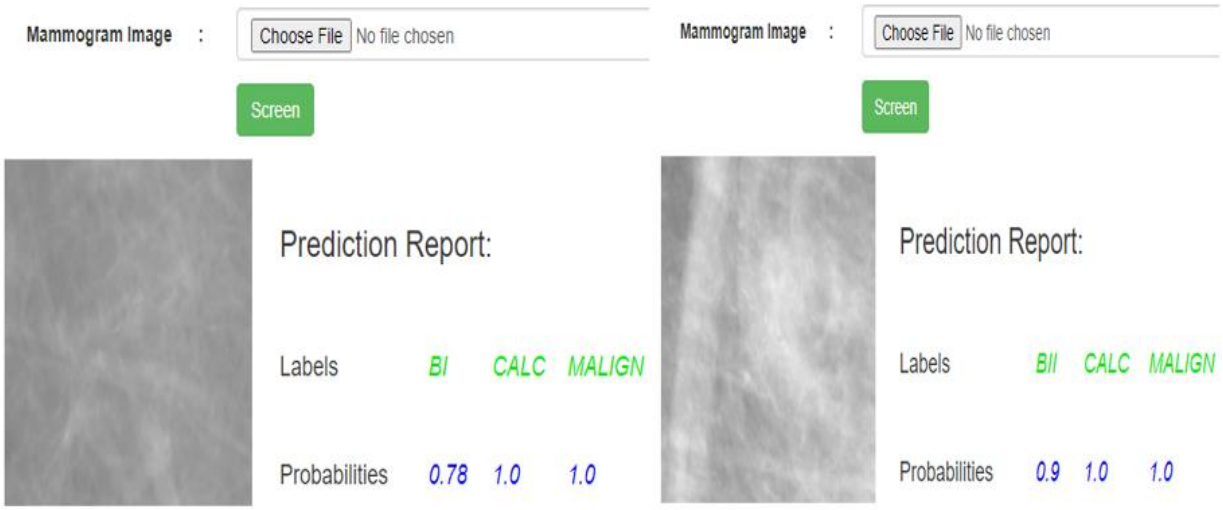
Category	Benign	BI	BII	BIII	BIV	Calc	Malign	Mass
Ground_Truth	1	0	1	0	0	1	0	0
Probability	0.994	0.312	0.992	0.796	0.127	0.974	0.198	0.513
Label	1	0	1	0	0	1	0	0



Category	Benign	BI	BII	BIII	BIV	Calc	Malign	Mass
Ground_Truth	0	0	0	1	0	1	1	0
Probability	0.017	0.000	0.002	0.996	0.000	0.999	0.993	0.001
Label	0	0	0	1	0	1	1	0

Figure 6.12. Result in benign and malign mammogram by efficientnetb3 model

The sample image screened by the efficientnetb3 model is shown in figure 6.13.



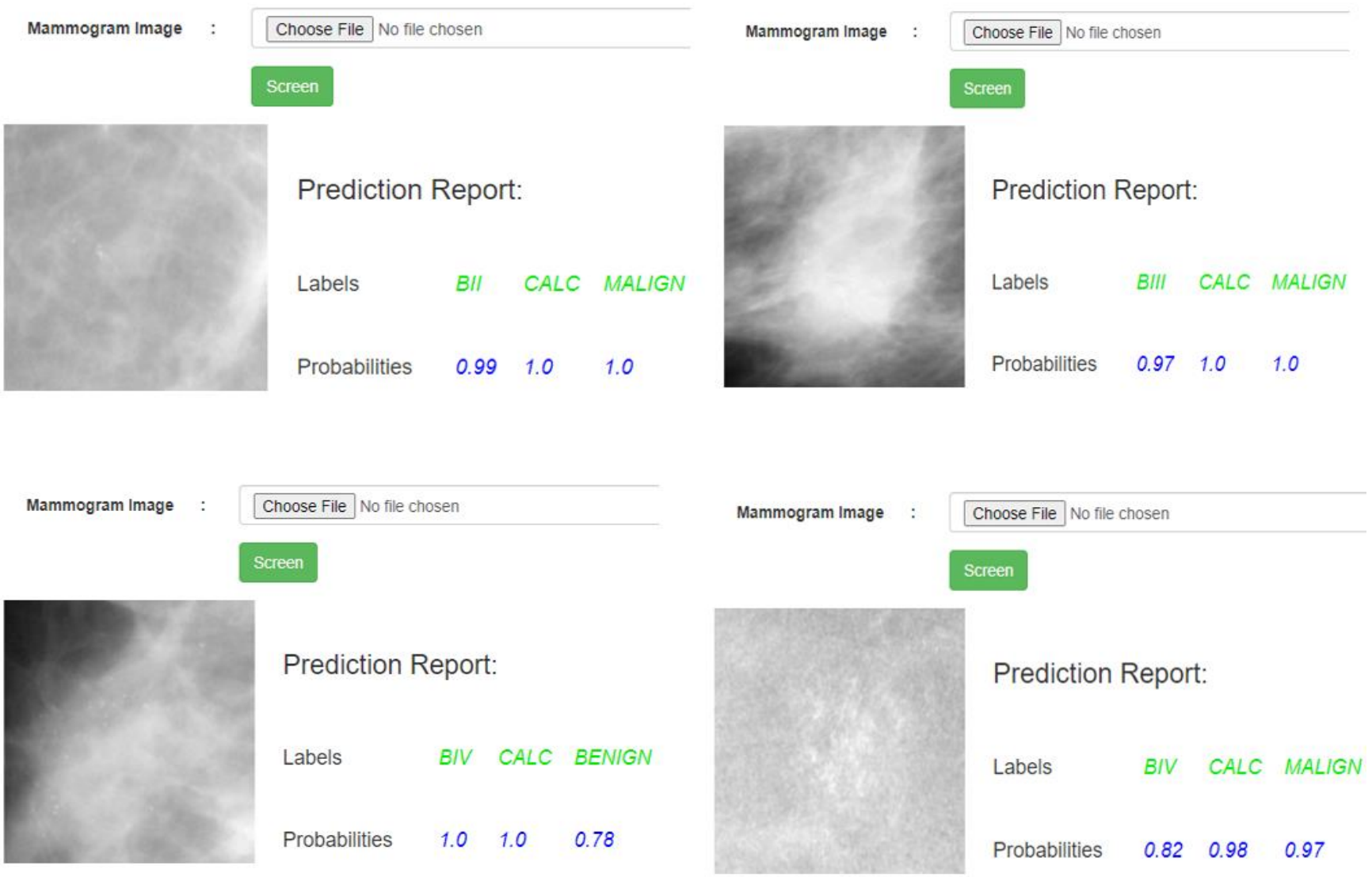


Figure 6.13. Sample Screened image with efficientnetb3

6.3. Discussions

EfficientNetB4 shows high accuracy on the training set. However, on testing set and evaluation metrics, it has the least value. The regularization method is used for reducing overfitting and it is experimentally identified as a problem of the small dataset used. This is due to the high parameters it calculates on a small number of data and overfitting is occurred. On the other hand, VGG16 has the lowest training accuracy. But on evaluation metrics, its values are greater than that of efficientnetb4. Densenet121 has the lowest computational cost and training time. But its accuracy is smaller than that of efficientnetb3.

Efficientnetb3 is investigated and identified as the best feature extractor by answering the first research question to be answered by this works. Among the conducted experiments efficientnetb3 model gave the best performance results. The model outperforms the previous work (vgg16 model) in all evaluation metrics. 13.25% in f1-score, 53% in hamming loss, 36.7% in coverage error, and 12.35% in an exact match are percentages improved over baseline paper by this works. This guarantees that efficientnetb3 is the best feature extractor for multi-label classification. The results indicate that the proposed model achieves a substantial improvement over existing methods in terms of all conducted evaluation metrics.

In addition, vgg16 has 7 x 7 x 512 shape extracted features followed by 3 layers with 4096, 4096, 8 nodes. The maximum number of nodes found in fully connected layer increases the computation cost of the model. 134M parameters are calculated which affect the model performance. To minimize this computational cost and increase accuracy fully connected NN with few layers and a small number of nodes are used. Accordingly, 2 layers with 128 and 8 nodes are optimized for this works. The classification part of the model is reduced and the number of parameters decreased to 20M. This optimization answers the second research question of this work by reducing the parameter in baseline work by 114M. Except for vgg16, all three models have the same classification part. However, the difference in fully connected NN parameters comes from the flatten function of each model. They have different feature maps which affect the number of parameters.

Table 6.1 compares the evaluation metrics of 4 models on the same datasets. Efficientnet3 shows the best experimental results on all the evaluation metrics.

Table 6.1. The result of evaluation metrics with different models

Models	Evaluation Metrics			
	F1-Score	Hamming_Loss	Coverage error	Exact Match
Efficientnetb4	0.66±0.025	0.27±0.026	6.72±0.015	0.575±0.015
VGG16	0.83±0.017	0.132±0.015	5.88±0.015	0.71±0.015
Densenet121	0.91±0.015	0.091±0.017	5.01±0.029	0.76±0.057
Efficientnetb3 + ONN (proposed)	**0.94±0.007**	**0.044±0.0025**	**3.72±0.015**	**0.81±0.0025**

Figure 6.14 shows the result comparison in a graph. Coverage error metric is excluded because its value differs from others and changes the graph visualization.

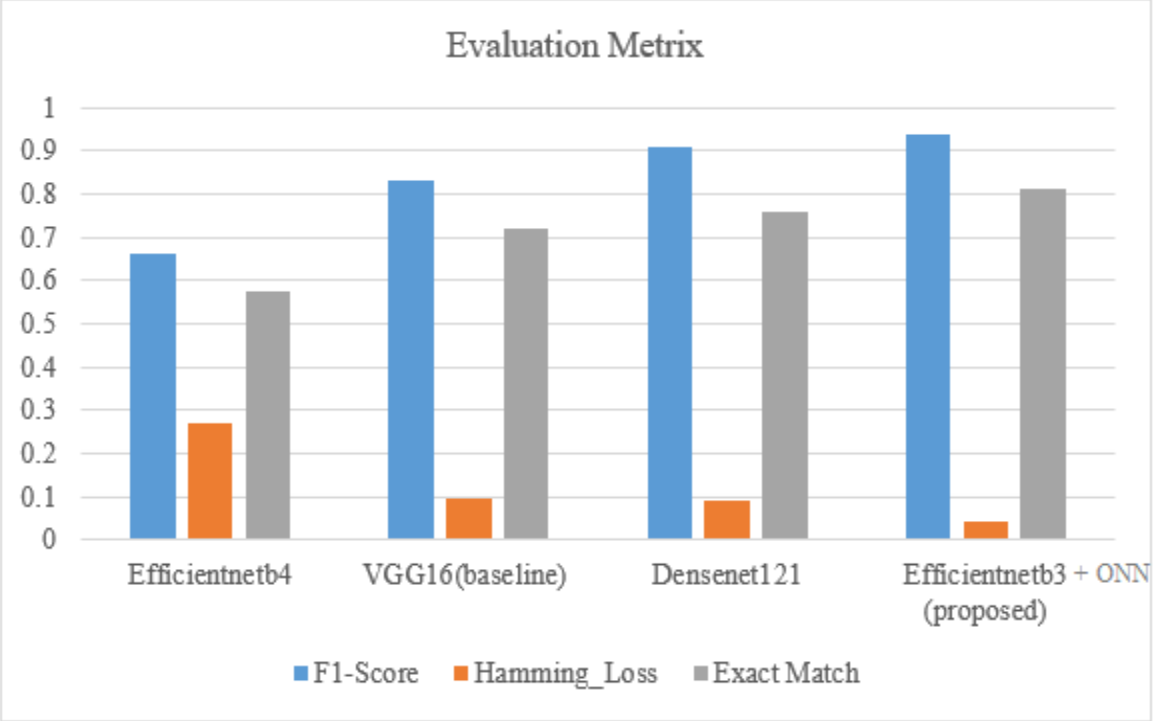


Figure 6.14. Evaluation metrics in graph

The result values that increment or decrement in small numbers come from 5-fold cross-validation applied during training. Figure 6.15 shows the accuracy and loss graph of 5-fold results for efficientnetb3.

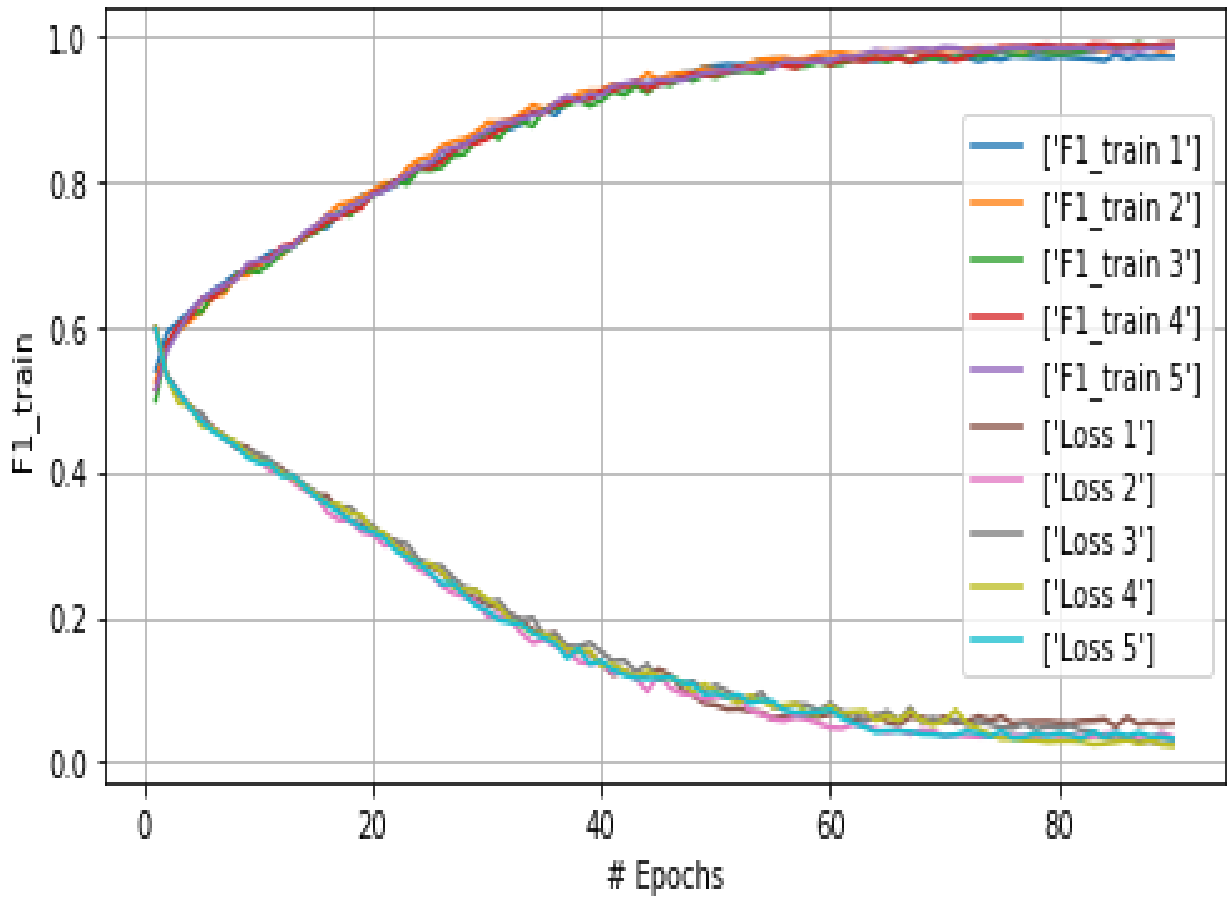


Figure 6.15. Accuracy and loss of 5-fold cross-validation

Figure 6.16 shows the comparison of four models' accuracy and loss in a graph. The difference between the models is a high range for two of them and small for others. Efficientnetb3 has the highest accuracy and the lowest loss in training.

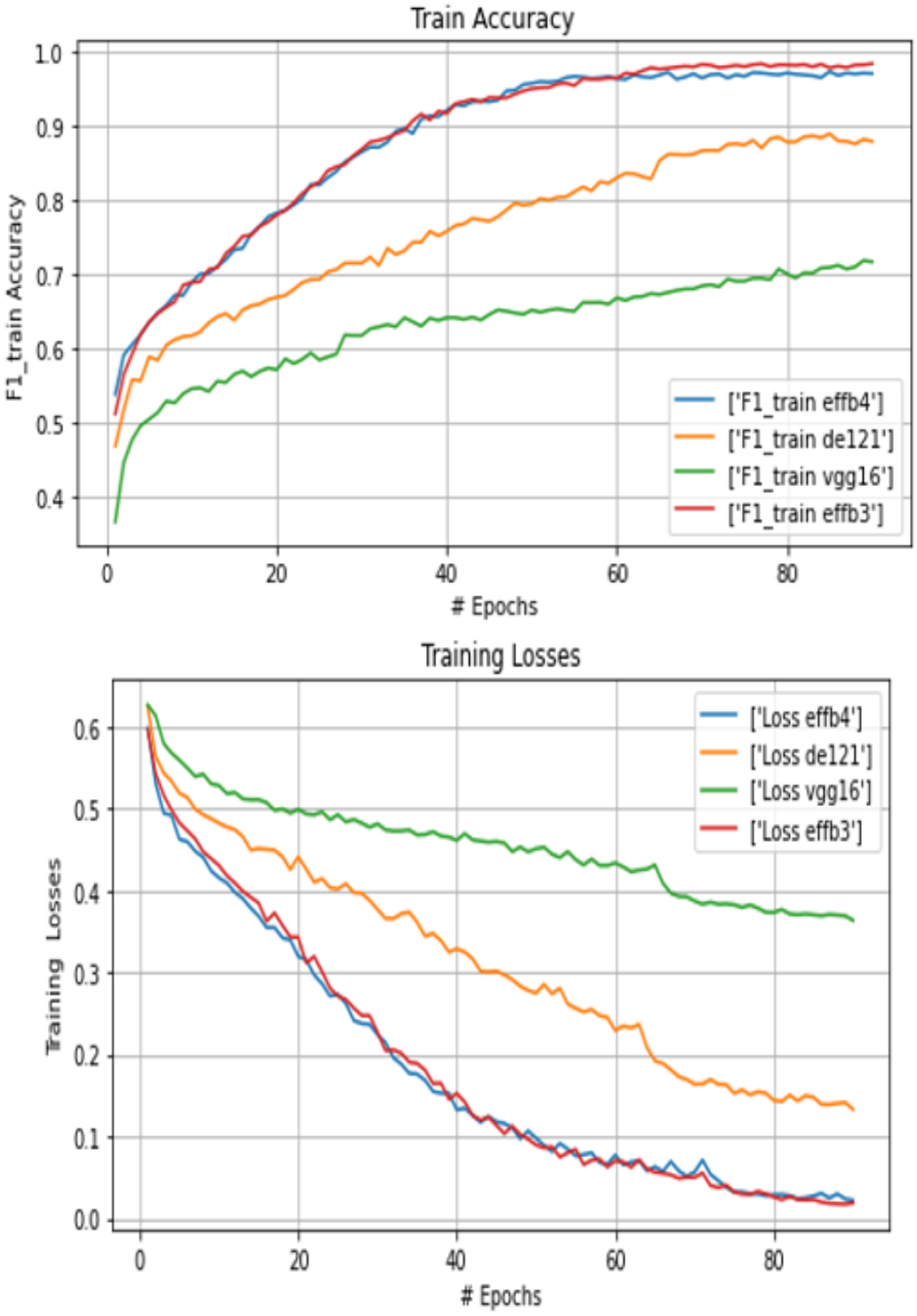


Figure 6.16. Accuracy and losses comparison of the models

Different experiments are conducted to optimize the neural network in the classification parts of the model. At first, the efficientnetb3 with 8 output layers is implemented. The result is good but needs to optimize more. Second, it is implemented with 64 in hidden layers and 8 in output layers are implemented. Next, 128 hidden layers with 8 output layers are implemented with efficientnetb3 resulting in an optimized solution compared to other results. It is practically optimal since the results of NN with greater values and smaller values are less than the result of 128 and 8 nodes. Finally, the value of 256 nodes in a first layer, 128 layers in hidden layers, and 8 nodes in the last layer result in nearly fewer values compared to the optimal values.

Table 6.2. results of efficientnetb3 within the models

Models	Evaluation Metrics			
	F1-Score	Hamming_Loss	Coverage error	Exact Match
Efficientnetb3 + 8	0.92±0.01	0.046±0.015	3.81±0.012	0.79±0.01
Efficientnetb3 + 64+8	0.92±0.009	0.045±0.006	3.73±0.034	0.80±0.02
Efficientnetb3 +256 + 128+8	0.94±0.002	0.044±0.015	**3.72±0.024**	0.80±0.01
Efficientnetb3 +128 + 8	**0.94±0.005**	**0.044±0.0017**	3.72±0.025	**0.81±0.0026**

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORKS

7.1. Conclusion

Cancer is a disease caused by uncontrolled cell replication and breast cancer is one type that originated in the breast. Breast cancer cells usually form tumors, which can usually be seen on X-rays. It occurs mostly in women, and sometimes men also develop breast cancer. The origin of breast cancer is the tube that carries milk to the nipple. Sometimes it begins in the glands that make breast milk.

Single label classification with one specific category was done previously. Some use mass finding while other use calcification. Density is also used by other studies to screen breast cancer. But mammogram gives more labels or information at the same time. Multi-label classification handles this problem by giving multiple predictive reports at the same time.

Efficientnetb4 shows good accuracy during training. Unfortunately, it has the least evaluation metrics compared to other conducted experiments. This comes from a high number of parameters it computes on a small dataset. Vgg16 has the least accuracy graph but in evaluation metrics, its values are greater than efficientnetb4 values. In addition, densenet121 is optimal in computational cost and training time.

The empirical study conducted specify that efficientnetb3 was investigated as the best feature extractor for this works from the pre-trained Keras application. The model results show that it has the highest values in all evaluation metrics. Again, it is efficient in terms of training time and computational cost. Furthermore, a fully connected neural network that has 2 layers excluding the input layer from the feature vector and 128 and 8 nodes optimize the numbers of parameters dramatically from 134M to 20M compared to vgg16. The model is multi-label classification and reports three subcategories for one image at the same time.

The results have shown that the efficientnetb3 model has outperformed all conducted experiments. It made more improvement on the baseline paper. 13.25% in f1-score, 53% in hamming loss, 36.7% in coverage error, and 12.35% in an exact match are percentages of improvement on evaluation metrics when compared with the previous works. Densenet121 has shown better

performance next to the efficientnetb3 in evaluation metrics and has the best computational cost and training time.

In conclusion, an efficientnetb3 feature extractor with ONN confirms the best improvement on all evaluation metrics and reduces the computational cost and training time effectively.

7.2. Future Works

Finding the best solution for a problem on hand and giving direction for the future is the nature of research. Optimizing feature extractor by cutting in different layers with compatible fully connected NN further is recommended since this increases the accuracy and decreases parameters to be calculated.

In addition, the dataset contains imbalanced numbers in the density subcategory. Depending on this imbalance data the validation accuracy and validation loss of the model are low and there is some incorrect classification. It is the recommendation of this works to use the customized loss function which is the dataset-specific loss function to solve this problem. Besides, using more mammogram datasets and other modalities of breast datasets is extend this work in the future.

REFERENCES

Allaouzi, I., & Ben Ahmed, M. (2019). A Novel Approach for Multi-Label Chest X-Ray Classification of Common Thorax Diseases. *IEEE Access*, *7*, 64279–64288. https://doi.org/10.1109/ACCESS.2019.2916849

Arevalo, J., González, F. A., Ramos-Pollán, R., Oliveira, J. L., & Guevara Lopez, M. A. (2016). Representation learning for mammography mass lesion classification with convolutional neural networks. *Computer Methods and Programs in Biomedicine*, *127*, 248–257. https://doi.org/10.1016/j.cmpb.2015.12.014

Ayache, N. (1995). Medical computer vision, virtual reality and robotics. *Image and Vision Computing*, *13*(4), 295–313. https://doi.org/10.1016/0262-8856(95)99717-F

Basha, S. H. S., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2020). Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, *378*, 112–119. https://doi.org/10.1016/j.neucom.2019.10.008

Berment, H., Becette, V., Mohallem, M., Ferreira, F., & Chérel, P. (2014). Masses in mammography: What are the underlying anatomopathological lesions? *Diagnostic and Interventional Imaging*, *95*(2), 124–133. https://doi.org/10.1016/j.diii.2013.12.010

Chougrad, H., Zouaki, H., & Alheyane, O. (2018). Deep Convolutional Neural Networks for breast cancer screening. *Computer Methods and Programs in Biomedicine*, *157*, 19–30. https://doi.org/10.1016/j.cmpb.2018.01.011

Chougrad, H., Zouaki, H., & Alheyane, O. (2020). Multi-label transfer learning for the early diagnosis of breast cancer. *Neurocomputing*, *392*(February), 168–180. https://doi.org/10.1016/j.neucom.2019.01.112

D'Orsi, C. J. (1996). The American College of Radiology mammography lexicon: an initial attempt to standardize terminology. *AJR. American Journal of Roentgenology*, *166*(4), 779–780. https://doi.org/10.2214/ajr.166.4.8610548

Diagnosis, S. V. (2005). *Benefits and Limitations of Mammography*. *Dc*, 1–18. http://www.ncbi.nlm.nih.gov/books/NBK22311/

Fei-Fei, L., Deng, J., & Li, K. (2010). ImageNet: Constructing a large-scale image database. *Journal of Vision*, *9*(8), 1037–1037. https://doi.org/10.1167/9.8.1037

Gastounioti, A., Conant, E. F., & Kontos, D. (2016). Beyond breast density: A review on the advancing role of parenchymal texture analysis in breast cancer risk assessment. *Breast Cancer Research*, *18*(1), 1–12. https://doi.org/10.1186/s13058-016-0755-8

Heywang-Köbrunner, S. H., Hacker, A., & Sedlacek, S. (2011). Advantages and disadvantages of mammography screening. *Breast Care*, *6*(3), 199–207. https://doi.org/10.1159/000329005

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Jiang, M., Zhang, S., Liu, J., Shen, T., & Metaxas, D. N. (2014). Computer-aided diagnosis of mammographic masses using vocabulary tree-based image retrieval. *2014 IEEE 11th International Symposium on Biomedical Imaging, ISBI 2014*, 1123–1126. https://doi.org/10.1109/isbi.2014.6868072

Jiao, Z., Gao, X., Wang, Y., & Li, J. (2016). A deep feature based framework for breast masses classification. *Neurocomputing*, *197*, 221–231. https://doi.org/10.1016/j.neucom.2016.02.060

Karahaliou, A. N., Boniatis, I. S., Skiadopoulos, S. G., Sakellaropoulos, F. N., Arikidis, N. S., Likaki, L. E. A., Panayiotakis, G. S., & Costaridou, L. I. (2008). Breast cancer diagnosis: Analyzing texture of tissue surrounding microcalcifications. *IEEE Transactions on Information Technology in Biomedicine*, *12*(6), 731–738. https://doi.org/10.1109/TITB.2008.920634

Kim, M., Yun, J., Cho, Y., Shin, K., Jang, R., Bae, H. J., & Kim, N. (2019). Deep learning in medical imaging. *Neurospine*, *16*(4), 657–668. https://doi.org/10.14245/ns.1938396.198

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, R. S., Gimenez, F., Hoogi, A., Miyake, K. K., Gorovoy, M., & Rubin, D. L. (2017). Data Descriptor: A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific Data*, *4*, 1–9. https://doi.org/10.1038/sdata.2017.177

Li, H., Meng, X., Wang, T., Tang, Y., & Yin, Y. (2017). Breast masses in mammography classification with local contour features. *BioMedical Engineering Online*, *16*(1), 1–12. https://doi.org/10.1186/s12938-017-0332-0

Li, M., & Zhou, Z. H. (2007). Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, *37*(6), 1088–1098. https://doi.org/10.1109/TSMCA.2007.904745

Madjarov, G., Kocev, D., Gjorgjevikj, D., & Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, *45*(9), 3084–3104. https://doi.org/10.1016/j.patcog.2012.03.004

Narod, S. A., Iqbal, J., & Miller, A. B. (2015). Why have breast cancer mortality rates declined? *Journal of Cancer Policy*, *5*, 8–17. https://doi.org/10.1016/j.jcpo.2015.03.002

Navab, N., Hornegger, J., Wells, W. M., & Frangi, A. F. (2015). Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015: 18th International Conference Munich, Germany, October 5-9, 2015 proceedings, part III. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9351*(Cvd), 12–20. https://doi.org/10.1007/978-3-319-24574-4

Nawaz, M., Sewissy, A. A., & Soliman, T. H. A. (2018). Multi-class breast cancer classification using deep learning convolutional neural network. *International Journal of Advanced Computer Science and Applications*, *9*(6), 316–322. https://doi.org/10.14569/IJACSA.2018.090645

Review, N., Diabetes, K. T., Umpierrez, G. E., Smiley, D., & Kitabchi, A. E. (2006). *Annals of Internal Medicine Review*. 350–358.

Ritter, F., Boskamp, T., Homeyer, A., Laue, H., Schwier, M., Link, F., & Peitgen, H.-O. (2011). Medical Image Analysis: A Visual Approach. *IEEE Pulse*, *2*(6), 60–70. https://doi.org/10.1109/MPUL.2011.942929

Sampat, M. P., Markey, M. K., & Bovik, A. C. (2005). Computer-Aided Detection and Diagnosis in Mammography. In *Handbook of Image and Video Processing* (Second Edi). Elsevier Inc. https://doi.org/10.1016/B978-012119792-6/50130-3

Sangeetha, V., & Prasad, K. J. R. (2006). Syntheses of novel derivatives of 2-acetylfuro[2,3-a]carbazoles, benzo[1,2-b]-1,4-thiazepino[2,3-a]carbazoles and 1-acetyloxycarbazole-2-carbaldehydes. *Indian Journal of Chemistry - Section B Organic and Medicinal Chemistry*, *45*(8), 1951–1954. https://doi.org/10.1002/chin.200650130

Shen, L. (2017). End-to-end training for whole image breast cancer diagnosis using an all convolutional design. *ArXiv*, *August*, 1–12. https://doi.org/10.1038/s41598-019-48995-4

Shiraishi, J., Li, Q., Appelbaum, D., & Doi, K. (2011). Computer-aided diagnosis and artificial intelligence in clinical imaging. *Seminars in Nuclear Medicine*, *41*(6), 449–462. https://doi.org/10.1053/j.semnuclmed.2011.06.004

Sickles, a. (n.d.). *Breast Periodic Benign Lesions : Results*. 463–468.

Siegel, R. L., Miller, K. D., & Jemal, A. (2017). Cancer statistics, 2017. *CA: A Cancer Journal for Clinicians*, *67*(1), 7–30. https://doi.org/10.3322/caac.21387

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Stoitsis, J., Valavanis, I., Mougiakakou, S. G., Golemati, S., Nikita, A., & Nikita, K. S. (2006). Computer aided diagnosis based on medical image processing and artificial intelligence methods. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *569*(2 SPEC. ISS.), 591–595. https://doi.org/10.1016/j.nima.2006.08.134

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, *0*(0), 1–41. https://doi.org/10.3322/caac.21660

Suzuki, K. (2013). Machine learning in computer-aided diagnosis of the thorax and colon in CT: A survey. *IEICE Transactions on Information and Systems*, *E96-D*(4), 772–783. https://doi.org/10.1587/transinf.E96.D.772

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *36th International Conference on Machine Learning, ICML 2019*, *2019-June*, 10691–10700.

Vachon, C. M., van Gils, C. H., Sellers, T. A., Ghosh, K., Pruthi, S., Brandt, K. R., & Pankratz, V. S. (2007). Mammographic density, breast cancer risk and risk prediction. *Breast Cancer Research*, *9*(6). https://doi.org/10.1186/bcr1829

Wang, J., Yang, X., Cai, H., Tan, W., Jin, C., & Li, L. (2016). Discrimination of Breast Cancer with Microcalcifications on Mammography by Deep Learning. *Scientific Reports*, *6*(June), 1–9. https://doi.org/10.1038/srep27327

Wenzhong, L., Huanlan, L., Caijian, H., & Liangjun, Z. (2020). *Classifications of Breast Cancer Images by Deep Learning*. https://doi.org/10.1101/2020.06.13.20130633

Whang, J. S., Baker, S. R., Patel, R., Luk, L., & Castro, A. (2013). The causes of medical malpractice suits against radiologists in the United States. *Radiology*, *266*(2), 548–554. https://doi.org/10.1148/radiol.12111119

Winkel, R. R., von Euler-Chelpin, M., Nielsen, M., Petersen, K., Lillholm, M., Nielsen, M. B., Lynge, E., Uldall, W. Y., & Vejborg, I. (2016). Mammographic density and structural features can individually and jointly contribute to breast cancer risk assessment in mammography screening: A case-control study. *BMC Cancer*, *16*(1), 1–12. https://doi.org/10.1186/s12885-016-2450-7

Zhang, M. L., & Zhou, Z. H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *26*(8), 1819–1837. https://doi.org/10.1109/TKDE.2013.39

APPENDIXES

Appendix A: Ablation Study

To understand the effect of hyperparameters and the classification part of fully connected networks different experiments are analyzed. Learning rate, dropout, optimizer, and loss function are investigated to analyze hyperparameter, whereas, NN with a different number of layers and nodes are conducted for optimizing the classification part of neural networks. Those studies are conducted using the efficientnetb3 model.

No	Parameter names	Values	Evaluation metrics			
			Fl_score	Hamming loss	Coverage error	Exact match
1	Dropout	0.1	0.51±0.2	0.476±0.38	7.78±1.5	0.47±54
		0.8	0.58±0.15	0.465±0.35	7.43±1.2	0.53±0.23
		0.2	**0.94±0.005**	**0.044±0.0017**	**3.72±0.025**	**0.81±0.0016**
2	Optimizer	SGD	0.63±0.03	0.42±0.015	7.08±0.057	0.60±0.5
		Adam	**0.94±0.005**	**0.044±0.0017**	**3.72±0.025**	**0.81±0.0016**
3	Loss	MAE	0.67±0.025	0.207±0.075	8±1.5	0.58±0.25
		MSE	0.93±0.045	0.046±0.07	3.75±0.035	0.80±0.005
		BCE	**0.94±0.005**	**0.044±0.0017**	**3.72±0.025**	**0.81±0.0016**
4	Learning rate	0.0002	0.33±1.5	0.59±1.4	8±2	0.30±1
		0.00009	0.47±1.3	0.499±1.2	7.69±1.7	0.40±0.85
		0.0001	**0.94±0.005**	**0.044±0.0017**	**3.72±0.025**	**0.81±0.0016**
5	No of NN layers and nodes	8	0.91±0.01	0.046±0.015	3.81±0.012	0.79±0.01
		64, 8	0.92±0.009	0.045±0.006	3.73±0.034	0.80±0.02
		256, 128, 8	0.94±0.007	0.044±0.0018	**3.72±0.024**	0.80±0.01
		128, 8	**0.94±0.005**	**0.044±0.0017**	3.72±0.025	**0.81±0.0026**

Appendix B: Block of Code

Sample codes for convolution block of densenet121

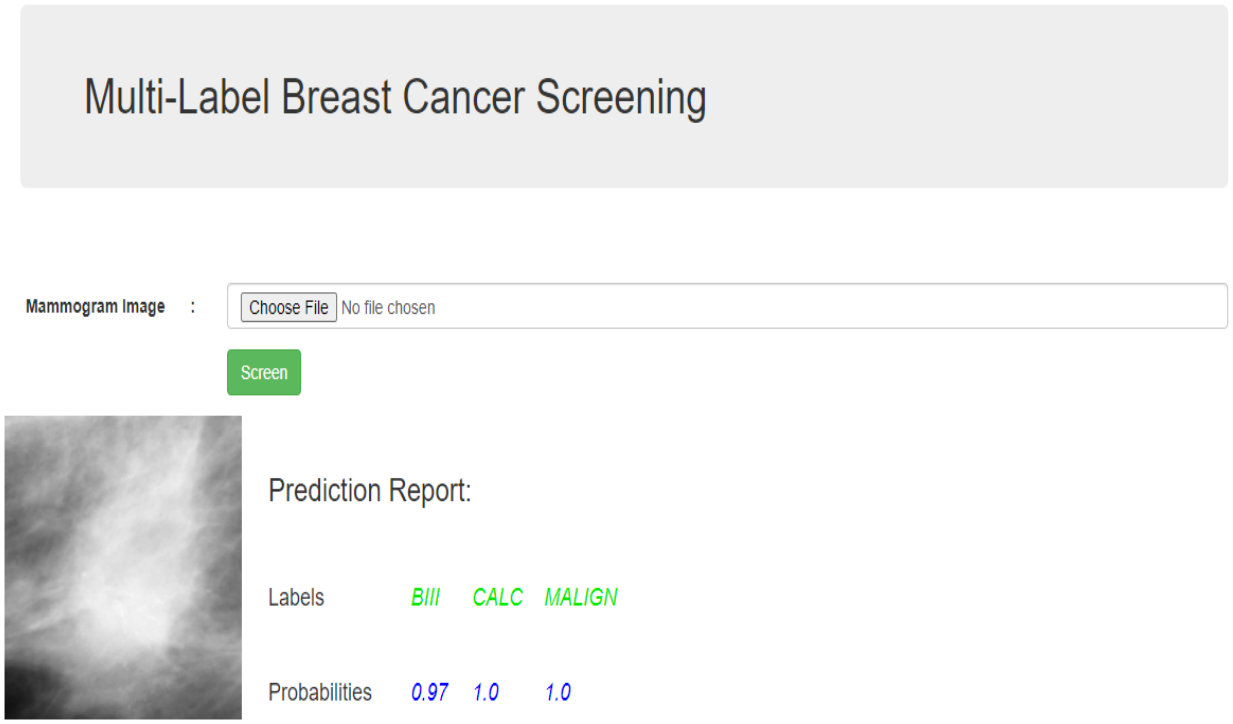
```
def conv_block(x, growth_rate, name):
    """A building block for a dense block.
    # Arguments
        x: input tensor.
        growth_rate: float, growth rate at dense layers.
        name: string, block label.
    # Returns
        Output tensor for the block.
    """
    bn_axis = 3 if backend.image_data_format() == 'channels_last' else 1
    x1 = layers.BatchNormalization(axis=bn_axis,
                                   epsilon=1.001e-5,
                                   name=name + '_0_bn')(x)
    x1 = layers.Activation('relu', name=name + '_0_relu')(x1)
    x1 = layers.Conv2D(4 * growth_rate, 1,
                       use_bias=False,
                       name=name + '_1_conv')(x1)
    x1 = layers.BatchNormalization(axis=bn_axis, epsilon=1.001e-5,
                                   name=name + '_1_bn')(x1)
    x1 = layers.Activation('relu', name=name + '_1_relu')(x1)
    x1 = layers.Conv2D(growth_rate, 3,
                       padding='same',
                       use_bias=False,
                       name=name + '_2_conv')(x1)
    x = layers.Concatenate(axis=bn_axis, name=name + '_concat')([x, x1])
    return x
```

Sample codes of efficientnetb3

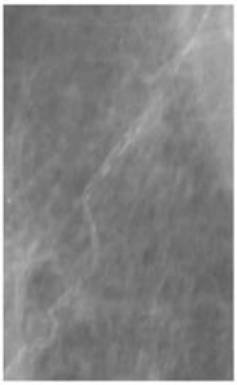
```
# Build top
    x = layers.Conv2D(round_filters(1280, width_coefficient, depth_divisor),
        1,
                      padding='same',
                      use_bias=False,
                      kernel_initializer=CONV_KERNEL_INITIALIZER,
                      name='top_conv')(x)
    x = layers.BatchNormalization(axis=bn_axis, name='top_bn')(x)
    x = layers.Activation(activation, name='top_activation')(x)
    if include_top:
        x = layers.GlobalAveragePooling2D(name='avg_pool')(x)
        if dropout_rate and dropout_rate > 0:
            x = layers.Dropout(dropout_rate, name='top_dropout')(x)
        x = layers.Dense(classes,
                         activation='softmax',
                         kernel_initializer=DENSE_KERNEL_INITIALIZER,
                         name='probs')(x)
    else:
        if pooling == 'avg':
            x = layers.GlobalAveragePooling2D(name='avg_pool')(x)
        elif pooling == 'max':
            x = layers.GlobalMaxPooling2D(name='max_pool')(x)

    # Ensure that the model takes into account
    # any potential predecessors of `input_tensor`.
    if input_tensor is not None:
        inputs = keras_utils.get_source_inputs(input_tensor)
    else:
        inputs = img_input
```

Appendix C: Flask Application for Screening



Appendix D: Sample Screened Mammograms

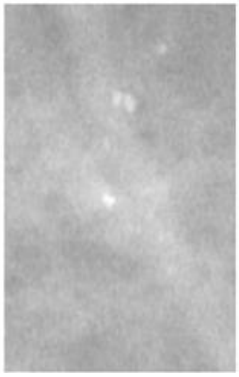


Prediction Report:

Labels	BI	CALC	BENIGN
Probabilities	1.0	1.0	1.0

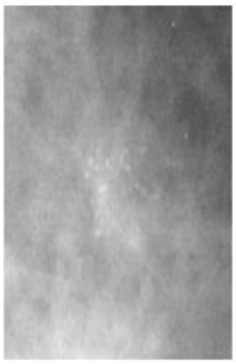
Prediction Report:

Labels	BIII	MASS	MALIGN
Probabilities	0.95	0.99	1.0



Prediction Report:

Labels	BII	CALC	BENIGN
Probabilities	1.0	1.0	1.0



Prediction Report:

Labels	BIII	CALC	MALIGN
Probabilities	1.0	1.0	0.91