EARLY DETECTION OF BREAST CANCER USING MACHINE LEARNING

by

VEKANI REVIET BALOYI

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

In

Computer Science

in the

FACULTY OF SCIENCE & AGRICULTURE

(School of Mathematical and Computer Science)

at the

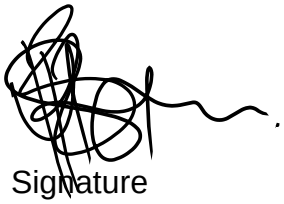
UNIVERSITY OF LIMPOPO

Supervisor: Professor SN MOKWENA

APRIL 2025

DECLARATION

I **Vekani Reviet Baloyi** declare that the title **"Early Detection of Breast Cancer Using Machine Learning,"** (thesis) hereby submitted to the University of Limpopo, for the degree of Doctor of Philosophy in Computer Science has not previously been submitted by me for a degree at this or any other university; that it is my work in design and in execution, and that all material contained herein has been duly acknowledged.



Signature Date: February 2025

ACKNOWLEDGMENTS

I sincerely thank Professor NS Mokwena, my supervisor, for his continuous support during my academic career. His patience was instrumental during challenging phases, and his guidance, fuelled by his passion for the field, helped me refine my approach and go deeper into [specific area of research]. Prof. Mokwena's extensive knowledge and mentorship have been invaluable, and his impact on the success of this study has exceeded my expectations.

My deepest gratitude goes to my wife, Mkateko Baloyi. Her enduring love, prayers, care, and sacrifices have been the bedrock of my success. Her steadfast support has helped me become the person I am today by providing me with a steady source of strength. The presence and persistent support of my siblings, Martha, Morris, Jane, Justice, and Ntsako, are deeply appreciated. Their camaraderie has been a constant source of strength throughout my academic journey. I would also like to acknowledge the immense influence of my late parents, Piet and Florah Baloyi (BB). Their unwavering love and dedication throughout my life have laid the foundation for this achievement.

To my extended family, friends, and colleagues, my deepest gratitude. You have all been a pillar of support, offering unwavering encouragement and understanding, especially during challenging times. Your presence in my life has been invaluable and I know that each of you played a crucial role in my academic success. Thank you for being part of this journey.

Finally, I thank God for leading me through all the difficulties and victories I have experienced throughout my academic career. His grace has been my ever-present companion, and I thank Him for making it possible for me to finish my degree. I put all my trust in Him for the trip ahead. I am grateful, Lord.

ABSTRACT

Recent developments in artificial intelligence have allowed significant gains in computer vision, cybersecurity, e-Commerce, and healthcare through the application of machine learning and deep learning models. A plethora of applications have offered effective ways to support physicians and radiologists in their medical imaging analysis, which continues to be the fundamental component of the visual representation used to formulate the final observation and diagnostic.

Expert knowledge in data science and computer engineering has recently been combined with medical studies in oncology. In this regard, automatic assistance, sometimes referred to as computer-aided diagnosis (CAD) systems, has gained popularity as a field for study and development in recent decades. As a result, experience and multidisciplinary knowledge were used in the creation of CAD systems, which are now used to analyse patient data and assist physicians and physicians in making decisions. Every day, radiologists and oncologists tackle the vital task of diagnosing, evaluating, and treating cancer, in addition to preventing it. To provide decision support for many applications in cancer patient care operations, such as lesion detection, cancer staging, characterisation, recurrence, tumour evaluation, and prognosis forecasting, a computer-aided design (CAD) system may be developed.

Throughout the world, breast cancer is believed to be one of the most prevalent cancers that affect women. It was also thought to be the main cause of death for women, and the number of deaths from it has been rising yearly. Early detection and diagnosis of breast cancer through regular screenings is the most effective way to significantly improve the chances of successful treatment by identifying breast cancer at its early stages.

Cancer arises from uncontrolled cell division that allows these cells to invade surrounding tissues. This uncontrolled growth is often triggered by mutations within genes that are the blueprints for cellular function. Breast cancer is a highly heterogeneous disease that exhibits variations in tumour biology, aggressiveness, and response to treatment. In South Africa, breast cancer is the most common cancer

among women, posing a significant health threat. Early detection is crucial, as it greatly improves treatment outcomes. While experienced physicians play a vital role in diagnosis, machine learning systems are emerging as promising tools with high accuracy in cancer identification. The primary goal of this research was to create a model that can assist radiologists in identifying and categorising breast cancer. The Mankweng Hospital repository provided the mammogram images that were used to create the deep learning algorithm. In transfer learning, a pre-trained model, VGG19, InceptionV3 and MobilenetV2, was utilised for fine-tuning the model. A convolutional neural network (CNN) model was developed and optimised using techniques to determine the best batch size, learning rate, epoch, and optimise parameters. During training, the InceptionV3 model achieved the best accuracy of 88%. The models generated are capable of classifying breast cancer cases. However, for some classes, there was not enough data available. This study applied augmentation to address the over-fitting of data. Thus, the next steps of this dissertation involve collecting a lot of data for every class and creating a more reliable categorisation model.

This dissertation therefore suggests a new model for breast cancer diagnostics that is based on the latest advances in computer vision and machine learning technologies. The best method to identify breast lesions early on and to reduce the risk of death is mammography screening. It helps to expose breast anomalies such as microcalcification, architectural distortion, and mass lesions. Having an additional reading tool or help system could improve the breast cancer diagnostic procedure, as the number of patients examined every day is always increasing. Several modalities, including an X-ray scanner and a full-field digital mammography (FFDM) system, can be used to obtain mammograms. The ultimate diagnosis may depend on the quality of the mammograms, the attributes (such as density and size) or the attributes (such as location, size, and form). Consequently, radiologists run the risk of overlooking the lesions, which could lead to incorrect diagnosis and detection. Therefore, the objective of this effort was to improve mammography reading to increase the accuracy of difficult assignments.

Incoming research may incorporate novel approaches to techniques by merging several mammography data sets and enhancing the extended training of deep learning models. Motives could also improve the model by including more breast

cancer lesions, such as calcification and architectural distortion, using annotated information. In this dissertation, a model was presented to help medical professionals and specialists determine the probability of the presence of breast cancer. All things considered, the suggested model approach combines the latest developments in deep learning, image processing, and picture-to-image translation for biomedical use.

Keywords: breast cancer, deep learning, computer-aided diagnosis, convolutional neural network, detection, classification, and multiple classification

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AHE	: Adaptive Histogram Equalisation
AI	: Artificial intelligence
ANN	: Artificial neural networks
APA	: Application Portability Profile
API	: Application Programming Interface
ARFF	: Attribute-Relation File Format
BACH	: Breast Cancer Histology
BAN	: Boosted Augmented Naive Bayes
BBN	: Bayes Belief Network
BCE	: Binary Cross-Entropy
BF	: Balance factor
BI-RADS	: Breast Imaging Reporting and Database System Score
CAD	: Computer-Aided Diagnosis
CART	: Classification and Regression Trees
CC	: Craniocaudal view
CCE	: Categorical Cross-Entropy
CDF	: Cumulative Distributive Function
CLAHE	: Contrast-Limited Adaptive Histogram Equalisation
CNN	: Convolutional neural network
CNTK	: Computational Network Toolkit
COD	: Cause of death
CPU	: Central Processing Unit
CRNN	: Convolutional Recurrent Neural Network
CSV	: Comma-Separated Values
CT	: Computed tomography
DCNN	: Deep Convolutional Neural Network

DNA	: Deoxyribonucleic Acid
DL	: Deep Learning
DT	: Decision Tree
ELM	: Extreme learning machine
ELU	: Exponential Linear Unit
FC	: Full Convolutional
FF	: Farthest First
FFDM	: Full-Field Digital Mammography
FLOPs	: Floating-Point Operations
FN	: False Negative
FP	: False Positive
FRDA	: Friedreich ataxia
GB	: Gradient Boosting
GIS	: Geographic Information Systems
GPU	: Graphical Processing Unit
HCM	: Hierarchical Cluster Method
IBK	: Instance-Based Learner
IBL	: Industry-Based Learning
ICIAR	: International Conference on Image Analysis & Recognition
IEEE	: Institute of Electrical and Electronics Engineers
ILSVR	: ImageNet Large-Scale Visual Recognition Challenge
IRCNN	: Inception Recurrent Convolutional Neural Network
IRRCNN	: Inception Recurrent Residual Convolutional Neural Network
K-NN	: K-Nearest Neighbour
KS	: Kappa Statistics
LMT	: Logistic Model Tree
LR	: Logistic regression

MAE	: Mean Absolute Error
ML	: Machine learning
MLO	: Mediolateral Oblique
MLP	: Multilayer Perceptron
MRI	: Magnetic resonance imaging
MSE	: Mean Squared Error
NAVE	: Navigating and Acting in Virtual Environment
NB	: Naive Bayes
NN	: Neural Network
ODA	: Outlier Detection Algorithm
PACS	: Picture Archiving and Communication System
PGM	: Portable Grey Map
RAM	: Random Access Memory
RBF	: Radial Basis Function
RBG	: Red, Blue, Green
RCL	: Recurrent Convolutional Layers
RNN	: Recurrent neural network
ROC	: Receiver Operating Characteristics
ROI	: Region of interest
RSNA	: Radiological Society of North America
SA	: South Africa
SEER	: Surveillance, Epidemiology, and End Results
SGD	: Stochastic Gradient Descent
SLFNs	: Single-Hidden-Layer Feedforward Networks
SMO	: Sequential Minimal Optimisation
SSA	: Sub-Saharan Africa
STR	: Survival Time Recode

SVM	: Support Vector Machine
TAN	: Tree-Augmented Naive Bayes
TN	: True Negative
TP	: True positive
UCI	: University of California, Irvine
UK	: United Kingdom
VSR	: Vital Status Recode
WBCD	: Wisconsin Breast Cancer Database
WDBC	: Wisconsin Diagnosis of Breast Cancer
WEKA	: Waikato Environment for Knowledge Analysis
WHO	: World Health Organisation

CHAPTER 1: INTRODUCTION

1.1. Background

The primary cause of cancer-related morbidity and death among women in sub-Saharan Africa (SSA) is breast cancer [1]. At 40 deaths per 100,000, breast cancer is still the most common cause of cancer-related deaths among women in sub-Saharan Africa [2]. The most common cancer in Sub-Saharan Africa (SSA) is breast cancer, with similar mortality rates observed in many countries in SSA despite lower incidence rates compared to high-income countries [2,3]. For example, age-standardised death rates for breast cancer are 14 and 16 per 100,000 women in the United Kingdom (UK) and South Africa (SA), respectively, but age-standardised incidence rates are 94 and 49 per 100,000 women in both countries [2,4]. South Africa has a wealth of hospitals, health centres, and health stations, but many lack doctors, according to the South African Ministry of Health [3,4]. According to [5], there is one doctor for many patients [6,7].

Breast cancer screening programmes are non-existent in most SSA nations, and many malignancies are discovered symptomatically and at an advanced stage [8]. Low survival rates for breast cancer are largely due to late-stage presentation [9,10]. One of the main objectives of comprehensive breast cancer strategies is early-stage presentation, which increases the chances of curative treatment and improves the prognosis of those who receive it [8]. Early intervention in the event of symptoms can result in a more accurate diagnosis and can be impacted by various factors, including awareness of symptoms and risk factors [11–15]. Laypersons' views about cancer risk and symptoms can have an impact on behaviours related to risk reduction and seeking help, both individually and in the community [16, 16]. Comprehending the characteristics and indicators of cancer symptoms and risk factors is essential to support the creation of locally appropriate therapies. Initiatives to increase breast cancer among lay health workers and the public have shown potential to encourage disease-staging and help-seeking behaviours [11, 17]. Research has indicated that cancer knowledge varies in African nations. Moreover, there may be variations between urban and rural environments within a nation [13, 18], highlighting the need for intervention programmes that are specifically tailored to certain geographic areas.

Most of the research on breast cancer awareness in Africa has been done in women who have already received a diagnosis of the disease in hospital settings [13, 14, 19]. Fewer studies document community awareness among women who remain undiagnosed, where perceptions and knowledge of cancer symptoms and risk factors can vary. Furthermore, locally validated metrics have not been included in most SSA investigations, hospital or community [20].

Many data sets on the Internet are available to develop deep learning models; however, they are not completely customised for South Africans or individuals with black skin. Black skin is thought to be associated with cancer, according to a machine learning model developed specifically for South Africa. Using medical images gathered from Mankweng hospitals, the data set was created to solve this issue using the latest deep learning-based algorithms to identify breast cancer. Mankweng Hospital, a public tertiary hospital in Limpopo Province, South Africa, serves the Capricorn District Municipality. Beyond providing specialised care, it is a vital training ground for future healthcare professionals. The hospital collaborates with the University of Limpopo's Faculty of Health Sciences, fostering a strong academic environment. Creating a model for identifying breast cancer is necessary to address this issue and assist professionals. A lot of interest has been generated in using deep learning, a class of machine learning algorithms, to medical imaging challenges because of its rapid progress. The purpose of this work is to investigate and create a breast cancer screening model using a local image database. Research indicates that while machine learning approaches can identify cancer with 91% precision, the majority experienced clinicians can diagnose the disease with 79% precision [21]. For this reason, the use of various deep learning and machine learning approaches might be quite helpful in identifying and categorising breast cancer.

Uncontrollable dividing cells that burst into surrounding tissues are the source of cancer [22]. Variations in deoxyribonucleic acid (DNA) are what cause cancer. DNA mutations that cause cancer mostly affect genes, which are portions of DNA. Furthermore, it is a general term for a wide range of diseases that appear as malignant versions of aberrant cell growth in one or more body organs. One of the features of cancer is the sudden emergence of aberrant cells that expand beyond their normal limits, allowing them to infect surrounding body components and spread to other

organs. Around 10.0 million cancer deaths (9.9 million excluding non-melanoma skin cancer) and 19.3 million new cancer cases (18.1 million excluding non-melanoma skin cancer) were globally predicted in 2020 (see Figure 1.1) [23]. As the most frequently diagnosed malignancy, female breast cancer has surpassed lung cancer with an expected 2.3 million new cases (11.7%), followed by lung (11.4%), colorectal (10%) and prostate (7%) [24].

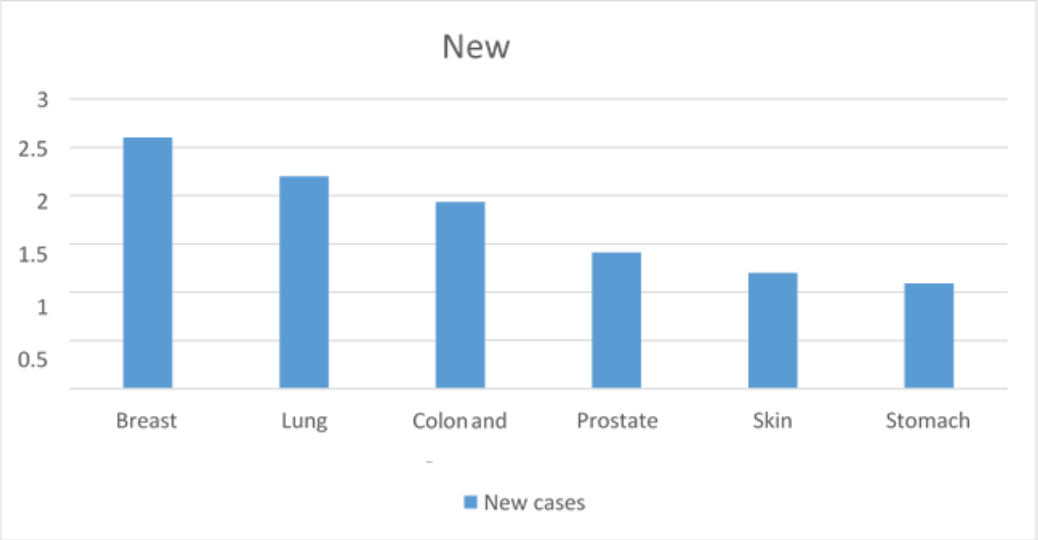


Figure 1: New cancer cases reported in 2020 [25]

One of the cancers that are generally recognised as a very diverse disease from a number of angles is breast cancer [26]. Variations in the histological and biological characteristics of the neoplasm, clinical results, and response to systemic therapy could be observed. Furthermore, it is possible to describe breast cancer as a collection of diseases caused by unchecked growth and alterations in breast tissue, which usually manifest as a lump or mass. It is the second most common cause of cancer in women worldwide and is becoming a greater public health concern in areas with limited resources. According to the World Health Organisation (WHO) 2020 cancer country profile report, South Africa has the highest age-standardised mortality rate (22.9 per 100,000 people) for breast cancer, making it the most common cancer [27].

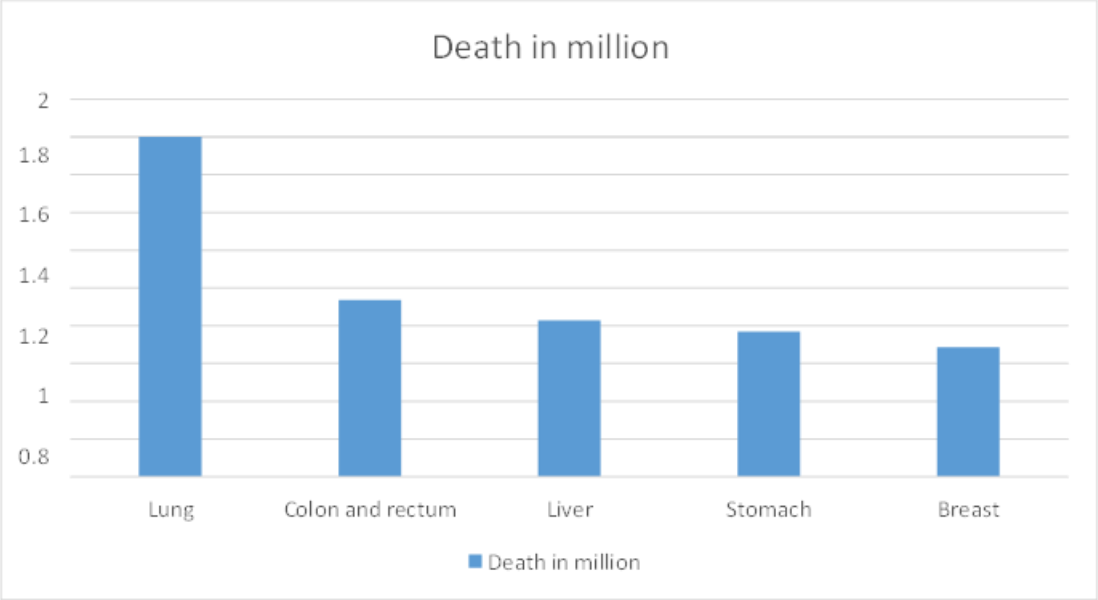


Figure 2: 2020's most prevalent cancer cause [25]

Deep learning is relatively young. It is a cross between machine learning and artificial intelligence methods [28]. It is one of the brain-inspired methods inspired by nature. This has led to the formation of additional neurones and has shown increased flexibility. Models have more accuracy than machine learning [29]. Artificial intelligence (AI) allows computers and other systems to perform tasks or make recommendations based on information obtained from digital images, videos, and other visual inputs [30]. Computer vision gives machines the ability to see, hear, and understand if artificial intelligence allows them to reason. Deep learning in computer vision has been used to address various challenges related to image classification, including those involving localisation, object detection, segmentation, style transfer, colourization, reconstruction, super-resolution, synthesis, and other issues [31]. This research focusses on customising the computer vision concept to identify and categorise breast cancer disease using colposcopy images.

In recent years, deep learning models have gained more significance in medical research [5,32]. Because it performs better than competitors in jobs that require large and complex training datasets, it is very helpful in the identification of cancer images. Good disease categorisation needs to be generally replicable, clinically helpful, easy to use, and supported by science. Regrettably, despite numerous attempts in the past and present, there is currently no ideal classification for breast cancer [4,33].

1.2. Motivation for the Study

Mammography is one of the most widely used forms of breast imaging. However, there are several restrictions to the imaging system. First, radiologists are in short supply in hospitals across the country who can interpret mammograms. Second, accurately interpreting the mammography image is a challenge for modern radiologists. To address the issue of reading and interpretation, the computer-aided diagnosis (CAD) system that uses deep learning (DL) is crucial to medical image analysis. When it comes to accuracy and performance, convolutional neural networks (CNNs) are the most popular and extensively utilised deep learning (DL) models for the task of image classification. It was designed especially for work with two-dimensional data, such as images taken in a medical environment [4]. Numerous single-label deep learning-based research on breast cancer screening has made use of the CAD system. However, there are no other multilabel classifications for breast cancer screening. Multi-label categorisation provides more detailed information about a single image, including its pathophysiology, density, and finding. This study was motivated by prior research on multi-label categorisation to capitalise on its advantages. To diagnose an illness, numerous medical equipment is used. Most devices require a significant level of physician interference to interpret the device's visual output. Therefore, to understand the image of these devices and enable early detection, a model is required.

1.3. Statement of the Problem

The incidence of breast cancer has declined in rich nations but has remained stable in most developing countries for other avoidable cancers including cervical, lung, colorectal, etc. Lower and middle-income countries account for more than 70% of breast cancer-related deaths. Early diagnosis and patient management could help reduce cancer burden. Traditionally, cancer risk and outcome predictions were based on a limited set of variables such as tumour size, demographics of patients, and environmental factors. This allowed for analysis using basic statistical methods or even a doctor's experience. However, advances in diagnostic and imaging technologies generate large amounts of data, including molecular and cellular information. This data overload overwhelms traditional methods, highlighting the need

for robust AI systems. These systems can analyse these complex data and provide fast and accurate cancer diagnoses.

Early cancer detection has consistently been shown to improve patient survival rates. This has led to the development of powerful machine learning models for accurate and rapid cancer diagnosis, often using data from developed nations. However, these models often lack generalisability, as they might not consider the specific characteristics of breast cancer in developing countries, such as the rising mortality rates seen in South Africa [11, 12, 15, 16]. Furthermore, limited pre-processed data from developing regions hinder further model development in these areas. This creates a situation where commercially available breast cancer models are trained in data sets that largely exclude data from developing countries. This research addresses this gap by providing much-needed preprocessed data specifically from developing regions.

Many internet datasets are available for use in the development of deep learning models; however, they are not entirely customised for South Africans or individuals with black skin. Black skin is thought to be associated with cancer, according to a machine-learning model developed specifically for South Africa. Using medical images gathered from Mankweng hospitals, the dataset was created to solve this issue by using the newest deep learning-based algorithms for the identification of breast cancer. Creating a model for identifying breast cancer is necessary to address this issue and assist professionals. A lot of interest has been generated in using deep learning, a class of machine learning algorithms, to medical imaging challenges due to its rapid progress. The purpose of this work is to investigate and create a deep learning algorithm-based breast cancer screening model using a local image database.

1.4. Research Questions

In this study, the following research questions and the suggested solution are attempted to be addressed.
Research Question 1. What is the current CNN approach to detecting breast cancer

disease and classifying more than one disease in real-time by low computational power devices?

Research Question 2. How can a CNN model for the detection of breast cancer be developed?

Research Question 3. What is the performance of the developed breast cancer detection in detecting disease?

1.5. The purpose of the study

The main aim of this study is to develop a breast cancer disease detection model using the CNN approach and mammogram images.

1.5.1. Objectives

To fulfil the overall goal, the following objectives are met.

i. To conduct a comprehensive literature review to identify this study's methods, algorithms and approaches.
ii. The data is labelled by their experts when collecting the mammogram images.
iii. To prepare training and test data sets after collecting a breast mammogram.
iv. To identify suitable deep learning algorithms for supervised learning.
v. To develop an optimal model to detect and classify breast cancer.
vi. To test and evaluate the performance of the proposed model.

1.6. Significance of the Study

One computer vision technology that can address the health problem is deep learning. Specifically, helping a doctor or specialist in South Africa. Early cancer prediction and detection are critical to accelerating the cancer diagnosis process and increasing the chance of survival. The medical facility uses the study findings to scan women's mammography images and determine their risk of breast cancer. Early detection and diagnosis of breast cancer save lives, as women are the foundation of the family. Therapy is straightforward and has a high chance of success when breast cancer is discovered early. Women are therefore encouraged to start screening when they

become thirty and to do so once a year thereafter. The individual listed below is one who specifically benefited from this work:

Radiologists: The study supports the ability to provide efficient, accurate, and timely care.

Woman: Cancer can be treated in a controllable way because the suggested model can confidently or effectively assess the cancer status.

Families: Since women are the backbone of families, they can live fearlessly or adhere to treatment plans once cancer has been identified during the screening process.

1.7. Scope and limitations of the Study.

The goal of the project was to create a deep learning-based model for the diagnosis of breast cancer. A methodology to identify and categorise breast cancers as malignant or not is currently being developed. A data collection that was used to create the suggested model was created by gathering images from Mankweng Hospital in Limpopo between 2023 (January) and 2023 (August). Images were obtained from data obtained from mammograms. The breast mammography image was used to classify the type of malignancy. To allow successful model training, the image was scaled to 448 by 448 pixels. The study is limited to images from colposcopy and X-rays. Furthermore, there are insufficient Category 2 and Category 6 data, which results in unbalanced data.

1.8. Structure of the dissertation.

This is the format for the remainder of the dissertation. The theory of the study is outlined in Chapter 2. A review of the literature and an overview of relevant studies are included in Section 3. An overview of a report on computer vision, machine learning, breast cancer, and deep learning concepts in general is provided by the literature review in Chapter 3. Chapter 3 featured works by many authors that are more pertinent to the deep learning approach to detecting breast and other cancer diseases. These works were included in the summary of related works. Next, the research approaches, experiment and the results are covered in Chapter 4. Furthermore, Chapter 5 explains the results and discussions. The study is finally summarised using a conclusion, contribution, and suggestions in Chapter 6.

CHAPTER 2: THEORY OF THE STUDY.

2.1. Introduction

The growing adoption of computer-aided diagnosis (CAD) in mammogram analysis at various screening locations and hospitals highlights its potential for a wider application in medical imaging [34, 35]. This suggests that CAD could become a valuable tool to identify and differentiate a wide range of anomalies in various imaging modalities used in medical examinations. In the fields of diagnostic radiology and medical imaging, CAD has become a prominent study topic [36, 37]. Early attempts at computer-aided medical image analysis emerged in the 1960s [38, 39]. However, a shift towards systematic research on computer-aided diagnosis (CAD) began in the 1980s. This marked a crucial change, moving from the concept of fully automated diagnosis by computers to a focus on computer-assisted tools for medical professionals. This chapter explores the philosophy and motivations behind this shift, along with the current state-of-the-art in CAD and its prospects within the context of Picture Archiving and Communication Systems (PACS).

2.2. An overview of CAD's progress over history

In the early 1980s, the Kurt Rossmann Laboratories for Radiologic Image Research at the University of Chicago pioneered extensive research on computer-aided diagnosis (CAD) schemes. This marked a significant shift toward the systematic development of CAD tools to assist radiologists. Meanwhile, research on Picture Archiving and Communication Systems (PACS) was also underway, with several researchers in the United States and Europe contributing to various aspects [40]. Although the benefits of PACS for improved image management and departmental efficiency were recognised, its impact on clinical diagnosis was still being explored [41, 42]. Thus, there was a belief that radiologists' daily work had to reap a significant benefit from digital images. Naturally, radiologists read images and make radiologic diagnoses daily. Consequently, the issue arose "How can the advantages of digital images aid radiologists in their diagnosis?" Immediately thereafter, the idea of computer-aided diagnostics was born. The concept of automated diagnosis using computers was not entirely novel in the 1980s. Studies conducted in the 1960s have

already demonstrated its potential feasibility [43, 44]. However, this has not been done until machine learning started to appear. In the next section, we will address the distinctions and similarities between computer-aided diagnostics and automated diagnostics. Thus, performing a computer analysis of the lesions seen in medical imaging seemed to be very challenging. Therefore, it was difficult to predict whether the creation of CAD systems would be successful. With a successful computer-aided diagnosis (CAD) system in place, the researchers set their sights on applying methodologies similar to those in other areas of medical research with the potential to significantly improve patient outcomes. Heart disease, lung cancer, and breast cancer were among the most significant medical topics at the time. Based on their success, the researchers targeted three specific diseases with a high clinical impact: lung cancer (focussing on nodule detection on chest radiographs [45, 46]), vascular diseases (exploring lesion analysis on vascular imaging [47, 48]), and breast cancer (investigating clustered microcalcification detection in mammograms [49]).

Since its inception, computer-aided diagnosis (CAD) research has been guided by three core principles. The first principle emphasises understanding the radiologists' workflow during image interpretation [50, 51]. This initial approach forms the foundation for developing methodologies and strategies to identify and measure lesions within medical images. This understanding of radiologists' workflow is crucial because it allows the development of algorithms that mimic human interpretation of medical images. Given the highly skilled nature of radiological diagnosis, which involves complex image interpretation [50, 51], the first principle was to understand the expertise of radiologists. This understanding would then be used to develop computer algorithms that mimic the radiologists' decision-making process to identify lesions, explain their reasoning behind potential oversights, and differentiate benign from malignant cases.

The second principle centred on the evaluation of the effectiveness of CAD after successful implementation. Widespread adoption in clinical practice in various hospitals around the world was considered the gold standard for success. This implied the need for the medical device industry to actively commercialise CAD products to cater to the global healthcare market. As a result, the choice was made to create and defend intellectual property about the fundamental technologies of computer-aided

design (CAD) schemes through the filing of patents, as well as to advocate for these properties through interactions with numerous stakeholders in the medical industry to facilitate future attempts to commercialise CAD products. Highlighting the groundbreaking nature of the patent, the 1987 filing for the first CAD system has become the most cited patent in the entire CAD technology domain [52].

The third principle aimed to promote widespread adoption of the CAD concept and facilitate collaboration between researchers. Recognising the multifaceted nature of CAD development, it emphasised the need for a broad research effort. This included conducting observer performance studies, clinical trials, and developing algorithms for various lesions across diverse imaging modalities. Crucially, this principle recognised that success would not hinge on a single research group. Instead, a collaboration between researchers from various institutions, each with experience in specific areas of CAD, was deemed essential for significant advancement. Therefore, rather than being seen as rivals, all CAD researchers should be viewed as collaborators and promoters. Consequently, massive scientific exhibits have been present at the Annual Meetings of the Radiological Society of North America (RSNA) for many years since the beginning of CAD research. The research team showcased significant progress in breast, chest and vascular imaging CAD in various exhibits. A key highlight was the development of a film digitiser and computer system in 1993. This system enabled real-time demonstrations of CAD to identify clustered microcalcifications in previously unseen mammograms. The researchers conducted an informal validation test in which 118 breast radiologists attending the RSNA congress presented their cases to the CAD system. The encouraging results suggested the potential of the technology [53].

Furthermore, to provide radiologists with hands-on experience and assess performance, observer studies were conducted with numerous participants at the RSNA meetings between 2004 and 2010. These studies focused on the detection of lesions on chest images, both with and without CAD assistance. The findings consistently demonstrated the benefits of CAD for radiologists [54, 55]. These demonstrations seem to have been effective in rapidly and broadly disseminating the CAD concept.

2.3. Automated Computer and Computer-Assisted Diagnosis

The 1960s saw the publication of the first studies on computer-assisted quantitative analysis of medical images [38, 59]. Since computers are more adept at certain jobs than humans are, it was widely believed at the time that computers could take the role of radiologists in detecting anomalies. As a result, the idea of automated or computerised diagnosis in radiography was formed at that point. These early attempts were unsuccessful despite the exciting results that were reported. The reasons for this were the lack of powerful computers, the lack of sophisticated image processing techniques, and the difficulty of obtaining digital images. However, having too high expectations for computers was a significant flaw. During the 1950s, a wide range of automated computer diagnosis techniques were tried as decision-support tools in several medical disciplines. Haematologist R.L. Engle, reflecting on his three decades of experience in the field, concluded his review article by stating [65] "Thus, we do not see much promise in the development of computer programmes to simulate the decision-making of a physician". However, the conclusion was that it was time to give up trying to train computers to function as diagnosticians after many years. These efforts failed because the researchers at that time compared computers with radiologists. They wanted to create rivalry instead of seeing this exercise as complementary.

However, in the 1980s, an alternative method surfaced, presuming that radiologists could make use instead of replacing them. These days, this idea is called computer-aided diagnosis, and it has become a big trend. Toward the beginning of the development of CAD schemes, some computer scientists expressed scepticism about its feasibility [56, 57]. However, subsequent research has demonstrably refuted these concerns. At that time, a failed attempt to develop automated computer diagnosis in earlier research attempts may have been the cause of this harsh critique. A failed research project in one area might lead to an unjustified bias towards a seemingly similar field, potentially hindering its progress.

Computer-aided diagnosis (CAD) continues to evolve, and ongoing research exploring its potential for various medical imaging tasks. While fully automated diagnosis remains an area of active investigation, CAD has become a valuable tool for assisting

radiologists in their work. For example, researchers have been devoting a significant amount of time to the creation of automated computer diagnostics. The potential for fully automated diagnosis using current CAD technology is a topic of ongoing discussion among researchers. A 2002 panel at the CARS meeting in Paris and the AAPM meeting in Montreal revealed mixed opinions, with approximately half of the participants believing that CAD could evolve into a fully automated diagnosis in 50 years [34, 35]. Therefore, understanding the similarities and differences between automated computer diagnostics and computerised design might be helpful. The analysis of digital medical images by computers is the common method used in both automated computer diagnosis and computer-aided diagnostics (CAD). This means that computer diagnostics and computer CAD both require the creation of computer algorithms.

The role of computer output is a key distinction between CAD and diagnostic computer systems. In CAD, radiologists maintain the ultimate decision-making authority, using the computer output as a second opinion to support their diagnosis. Consequently, radiologists may agree with the computer output in some clinical situations where they feel secure in their assessments, or they may disagree and subsequently ignore the computer. However, it is expected that the computer output will help improve the conclusion in circumstances where radiologists are less certain. Naturally, this enhancement is only feasible if the computer output is accurate. As computer processing power increases, CAD systems have the potential to become more accurate and reliable, potentially leading to improved diagnostic results. The computer's performance level does not need to match or surpass that of radiologists. The synergistic impact that results from combining computer capabilities and radiologist expertise accounts for the potential benefit of computer-aided diagnosis (CAD). These cumulative advantages have led to widespread use in real-world clinical settings.

However, very high-performance requirements for computer output are necessary when using automated computer diagnosis. Widespread adoption of automated diagnosis would be difficult to justify if the computer's sensitivity in lesion detection consistently falls below that of average physicians. In most developed countries, patients would probably not accept diagnostic results from computer systems that

perform below the average precision achieved by radiologists. Even if technological advancements improve the sensitivity of automated diagnosis, its widespread adoption faces hurdles. Insurance companies may be reluctant to reimburse diagnoses made solely by AI systems, especially in the absence of established guidelines. Additionally, a high rate of false positives, where the system identifies non-existent lesions, could overwhelm radiologists and erode trust in the technology. In these situations, the diagnosis made by the computer would not be regarded as automatic as each case it analysed might contain an irregularity. As a result, the doctor would need to confirm each case. Thus, to execute automated computer diagnosis, a computer must have high sensitivity and high specificity. Despite the success of CAD in mammography, achieving the high level of accuracy required for a fully automated diagnosis in various medical images remains a significant challenge for researchers developing lesion detection algorithms. This complexity limits the applicability of current computer-generated results to automated diagnosis in most developed countries.

The emphasis on quantitative evaluation is a key distinction between automated computer diagnosis and CAD. In CAD, the focus remains on helping radiologists, while automated diagnostic systems aim to replace them entirely. Evaluating a system's performance through quantifiable metrics becomes crucial for justifying its use in reaching a final diagnosis. The automatic computer diagnosis system operates at the same level of performance as the computer itself. Unlike automated diagnosis systems, which aim to replace physicians, CAD functions as a collaborative tool. Radiologists use the computer output as a second opinion to inform their diagnoses, potentially achieving a level of performance comparable to a team of two radiologists. Thus, it is imperative for CAD those radiologists make good use of computer data. Therefore, a quantitative assessment of the extent to which physicians can improve their overall performance using computer results is required. Even when computer results exhibit limitations, such as low sensitivity or specificity, physicians can still take advantage of this information with caution. While not a definitive answer, the results can serve as an additional data point to consider alongside other diagnostic tools and the patient’s history. It would be feasible to identify CAD, for example, if the computer was able to identify small lesions that radiologists might find difficult to identify and if radiologists could quickly reject "obvious" computer false positives. Automated

computer diagnosis, however, could not be used if the computer produced subpar results.

Consequently, CAD must assess both the performance of the machine and the physicians. Therefore, it is imperative to employ receiver operating characteristic (ROC) analysis [56, 57] to accurately and statistically assess whether the use of computer results could improve the performance of physicians. Even when the ROC curve generated by a CAD system for detecting clustered microcalcifications falls below that of radiologists, it can still be beneficial. This is because radiologists can integrate computer output into their workflow, potentially improving their diagnostic performance. A key study provided the first scientific evidence for this advantage, demonstrating that the improvement in radiologists' performance attributable to CAD was statistically significant [38, 39]. Subsequently, several researchers have documented congruent favourable results on the utility of CAD in the identification of diverse lesions, such as masses [58] and clustered microcalcifications [59] in mammography. As such, they have expedited the development of CAD schemes in numerous types of tests for the identification of various lesions.

In conclusion, automated computer diagnosis and computerised design share many similarities, but also differ greatly from each other. Computer-assisted diagnosis (CAD) and automated diagnosis represent distinct approaches. CAD emphasises collaboration, where radiologists leverage computer output as a second opinion. In contrast, automated diagnosis relies solely on computer algorithms, which require a significantly higher level of performance to achieve reliable diagnoses. For automated computer diagnosis to be effective, computer performance must match or exceed that of medical professionals. Radiologists and automated computer diagnosis systems are not replacements for each other, but rather powerful tools that can work together to achieve earlier and more accurate diagnoses of breast cancer. But when it comes to CAD, computer performance does not have to match or surpass that of radiologists; rather, it just needs to be an adjunct. To help radiologists identify breast tumours early on in mammograms, numerous CAD methods have been used. Although the overall precision of CAD systems may not yet match the average radiologist’s performance, they can still offer significant benefits in real-world clinical settings.

The quantity of CAD research-related papers that were presented at the RSNA meetings between 2004 and 2010. Most of these presentations focused on the chest, breast, and colon, although CAD research was also conducted on other organs, including the brain, liver, and skeletal and vascular systems. It should be mentioned that screening exams have been or are now being used to detect cancer in the breast [60, 61], lung [62, 63], and colon [64, 65]. A significant portion of these tests produce benign results and radiologists find that it is challenging and time consuming to identify the few problematic lesions they find. Thus, it is plausible that these screening exams have marked the start of the first stage of clinically applicable CAD. Clinical users can now take advantage of commercially available CAD systems to help identify these malignancies. This figure (Figure 3) illustrates the progress made in CAD system performance over time. It compares the sensitivity and false positive rates of two systems:

The most recent commercial CAD system achieves an estimated sensitivity of 98% with a false positive rate of 0.25 per image to detect clustered microcalcifications in mammograms. An earlier CAD system was implemented in 1993 when the technology was licenced, with a sensitivity of 87% and a false positive rate of 1.0 per image [35, 37, 42]. As the data show, the newer system offers a significant improvement in sensitivity (ability to detect true positives) while also reducing the false-positive rate (incorrectly flagged lesions). This advancement highlights the ongoing development and potential of CAD technology in medical imaging. This outcome indicates that industry efforts were necessary to achieve a significant improvement and, as a result, a viable clinical CAD system has developed.

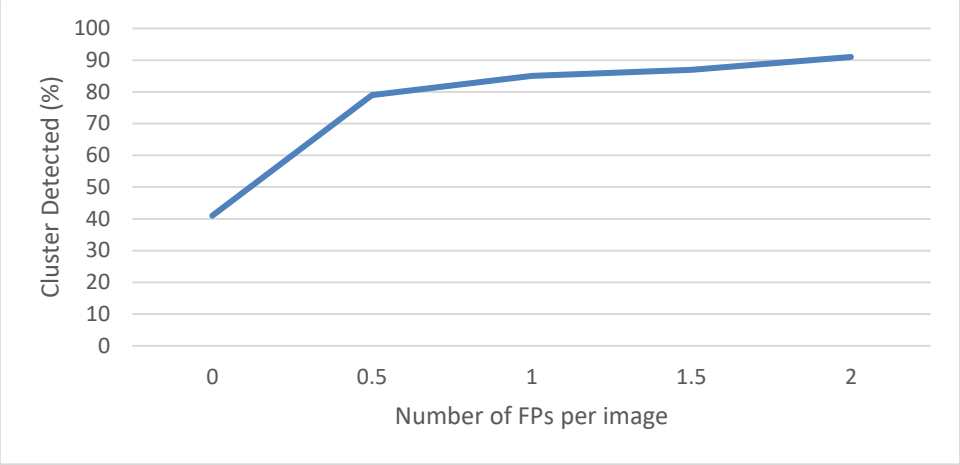


Figure 3: This graph visualises the significant progress made in computer-aided detection (CAD) of clustered microcalcifications over the past few decades.

2.4. How Computer Output Helps Radiologists

Unlike automated diagnosis systems, which solely rely on algorithms, radiologists in CAD workflows have the discretion to use or integrate the computer's output into their final decisions. As explained in Section 2.3, this human oversight is a crucial aspect of CAD. This is due to the idea of comparing radiologists with computers instead of seeing their complementarity. Therefore, it is vital to understand the purpose of the computer output and how radiologists could use it. CAD research has focused on two main areas, such as developing algorithms to identify potential abnormalities within medical images and assisting radiologists in differentiating between benign and malignant lesions, as well as distinguishing between various diseases, such as different types of interstitial lung disease.

Although CAD has demonstrably improved lesion detection in clinical settings, particularly breast lesions on mammograms, its application in differential diagnosis remains an area of active research. Currently, there are no commercially available CAD systems that address differential diagnosis. This distinction between lesion detection and classification tasks is crucial, as the literature will explore because the impact of computer output on radiologists' performance differs between these two areas.

Computer-aided diagnosis (CAD), which uses computerised feature extraction and classification, can be a valuable tool for radiologists to diagnose and identify abnormalities [66]. A CAD system's main function is to overcome the difficulty of reading DMs. The technology aims to correctly evaluate DM and identify cancer. The purpose of CAD structures was to reduce the cost of additional medical technology and alleviate the operator's dependence on diagnostics [67]. A study investigating the use of CAD for cancer cell detection found that radiologists could identify 80% of cancer cells without the help of CAD. However, the use of CAD increased the detection rate to 90%, suggesting that CAD can significantly improve the identification of cancerous cells [68]. The computerised diagnosis evaluates the information that a human or a machine collects and provides a result to determine the type of lesion that is present and whether it is malignant or not [71]. It is increasingly common to use medical imaging technologies in conjunction with CAD-based machine learning techniques (MLTs) to diagnose and detect cancer. Recent research highlights the importance of representation learning in overcoming limitations and improving the effectiveness of CAD algorithms [69, 70]. One prominent technique within representation learning is deep learning (DL).

Deep learning excels at extracting meaningful features from visual data using a hierarchical approach. Unlike traditional methods, deep learning can automatically encode complex features within a high-level representation (often visualised as a vector) without the need for manual post-processing [71]. This ability to automatically learn and encode features from raw data is a significant advantage of deep learning for CAD applications. The primary contribution is to demonstrate that radiologists and CAD can collaborate to identify and detect breast cancer in South Africa.

2.4.1 Impact of Computer Performance on Lesion Detection

Studies have shown that the area under the ROC curve (AUC) to identify clustered microcalcifications on mammograms improves significantly when radiologists can access computer output [72, 73]. This improvement comes from several factors, and radiologists may overlook some lesions during the initial review. The computer output can highlight these missed lesions, prompting the radiologist to re-evaluate the image and potentially correct the oversight. The false positives generated by the computer

system often differ from those identified by radiologists. As a result, radiologists can ignore these additional areas flagged without significant concern.

This combined approach, which takes advantage of the strengths of both radiologists and CAD, leads to a more accurate overall detection rate. However, radiologists are unlikely to take computer output seriously if there is an abnormally high proportion of false positives. Corresponding findings on improved efficacy in identifying interstitial opacities on chest radiographs were documented [38, 72]. This innovation has been made possible because radiologists sometimes make mistakes and recognise the proper computer output. For non-expert radiologists, such as residents, the degree of this kind of improvement in the CAD detection test was often higher than for expert radiologists, such as attendings [39, 73]. It may be argued that nonexpert radiologists have greater potential for improvement and that CAD can benefit nonexperts more than experts do because nonexperts without CAD typically perform at a lower level than experts do. Radiologists can correctly identify most computer output by determining whether a lesion is a true positive or a false positive caused by benign or normal patterns. This is evident from all the outcomes of the detection job in observer studies.

2.4.2 Differential diagnosis and the impact of computer output

Despite reports of extremely high-performance levels for CAD schemes for differential diagnosis [38, 73], there is currently no commercially accessible clinical CAD system for classification. This could be due in part to the presumption that radiologists' classification tasks may be more complex to deal with than detection tasks due to the impact of computer output. A study compared radiologists' performance in detecting malignant nodules on chest radiographs with and without the aid of computer-aided diagnosis (CAD) [39, 72]. The analysis used ROC curves, which measure the ability to distinguish between benign and malignant nodules. The results showed that radiologists achieved a statistically significant improvement in their likelihood of correctly identifying malignant nodules when using CAD. The AUC (area under the ROC curve) increased from 0.743 to 0.817. However, the study also found that radiologists could not exploit the full potential of the computer’s accuracy. The computer alone achieved a higher AUC of 0.889.

These findings suggest that, while CAD can be a valuable tool for radiologists, more research is needed to understand how radiologists can best integrate computer output into their workflow to maximise its benefit. This most likely indicates that the computer output was not thoroughly understood and that its dependability was not fully appreciated.

2.5. Conclusion

Computer-aided diagnosis (CAD) has become an established tool for radiologists in mammographic breast cancer screening. While its full potential for various types and modalities is still under development, CAD offers significant advantages over automated diagnosis systems. Unlike automation, which aims to replace physicians, CAD emphasises collaboration. Radiologists take advantage of computer output as a second opinion, potentially improving diagnostic accuracy.

This study proposes the integration of automated CAD schemes within the Picture Archiving and Communication System (PACS) as a comprehensive package for the detection and differential diagnosis of lesions. By seamlessly integrating these tools into their workflow, radiologists can use them as supportive aids during routine clinical practice.

CHAPTER 3: LITERATURE REVIEW AND RELATED WORKS

3.1. Introduction

This chapter delves into the following topics to establish the context for the proposed research on breast cancer detection using deep learning.

Review of the literature and related works (Section 3.1): This section reviews existing research to understand what previous scholars have accomplished and to identify potential knowledge gaps that this study aims to address.
Understanding breast cancer (Section 3.2): This section provides a basic understanding of breast cancer, including its definition and potential causes.
Mammography explained (Section 3.3): This section explains mammograms, a crucial imaging technique used for breast cancer detection.
Types of breast cancer (Section 3.4): This section differentiates between various types of breast cancer.
Digital Image Processing (Section 3.5): This section introduces the concept of digital image processing and its techniques.
Digital Image Processing for Breast Cancer Detection (Section 3.6): This section explores how digital image processing techniques are used for breast cancer detection.
Breast Image Analysis (Section 3.7): This section discusses the analysis of breast images, likely focussing on techniques relevant to cancer detection.
Deep Learning and Convolutional Neural Networks (Section 3.8): This section introduces the concept of deep learning and explains how convolutional neural networks (CNN) are specifically used for breast disease.
Transfer Learning (Section 3.9): This section explains the concept of transfer learning, a technique often used in deep learning for medical image analysis.
Related Works (Section 3.10): This section reviews deep learning and breast cancer detection research, highlighting relevant studies and their contributions.
Conclusions (Section 3.11): This section summarises the key points covered in the chapter and potentially outlines the research questions or objectives for the proposed work.

3.2. Breast Cancer Overview

Cancer is a condition of the body that occurs when cells cannot respond to normal stimuli. Abnormal cells with uncontrolled growth and spread can result in tumours [26]. Tumours that have migrated past their original location and are now encroaching on neighbouring healthy tissues are called invasive tumours. After lung, colon, liver, and stomach cancers, breast cancer affects more women than any other type of cancer and is one of the top five cancers causing mortality [59].

The human breast is an exocrine gland, similar to the sweat glands or salivary glands. Its primary function is milk production after childbirth. Here is a breakdown of the key structures relevant to mammography [74], the breast is composed of several lobes, which are tissue sections responsible for milk production. Each lobe contains numerous tiny cavities called alveoli. These cavities are lined with epithelial cells that produce milk. Milk produced in the alveoli travels through a network of small tubes called ducts toward the nipple. The nipple is a raised area in the centre of the breast, where the milk exits through the ducts.

Mammography is an X-ray imaging technique used for the detection of breast cancer. Denser breast tissue appears white on mammograms, whereas fatty tissue appears darker. During a mammogram, radiologists examine the image for abnormalities, such as areas of higher density that might indicate the presence of a mass or calcifications. These are tiny clusters of calcium deposits that can be a sign of early cancer development. Any unusual distortions or disturbances in the normal pattern of breast tissue can warrant further investigation. By understanding the underlying anatomy of the breast, radiologists can better interpret mammogram images and identify potential signs of breast cancer.

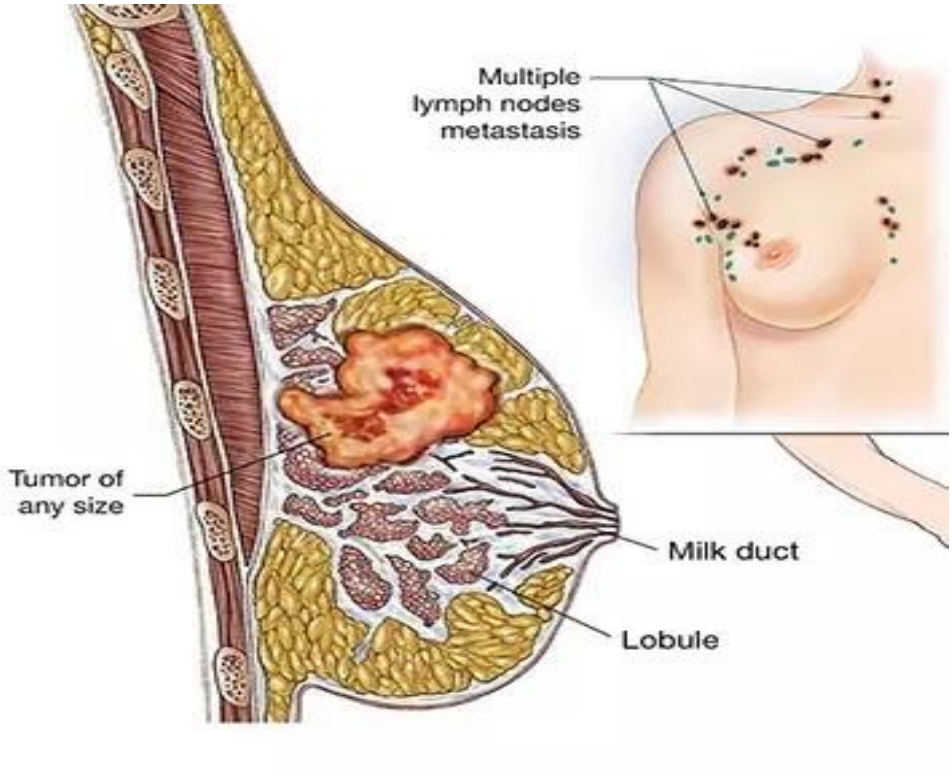


Figure 4: Individuals with metastatic breast cancer and their breast composition.

Source: https://www.cancersupportcommunity.org/metastatic-breast-cancer

3.3. Mammogram

Since the invention of X-rays, medical imaging has revolutionised cancerology. Today, a diverse range of imaging modalities plays a crucial role in various stages of cancer care, from early detection to treatment planning and monitoring [75]. Here are some of the key imaging techniques used in cancer diagnosis. Magnetic resonance imaging creates detailed cross-sectional images of organs and tissues, helping to detect and characterisation of tumours. Computed tomography (CT) scans provide 3D X-ray images, offering a comprehensive view of internal structures and helping to identify tumours and assess their spread. Ultrasound is the technology that uses sound waves to generate real-time images of soft tissues, which is useful for guiding biopsies and monitoring the response to treatment. Digital radiography is an advanced X-ray technique that captures high-resolution digital images, improving cancer detection and diagnosis.

As illustrated in Figure 5 (see the actual figure for placement), medical imaging

plays a critical role in the diagnostic process. By providing detailed visuals of internal structures, these imaging modalities empower radiologists to early detection and significantly improve treatment success rates. Imaging can help determine the type and aggressiveness of a tumour, guiding treatment decisions. Understanding the extent of cancer spread is crucial to selecting the most appropriate treatment approach. Beyond diagnosis, medical imaging also plays a vital role in helping define the target area for radiation therapy and guiding surgical procedures. Imaging helps assess the effectiveness of treatment and identify potential complications.

Figure 5: Using medical imaging to examine and detect breast lesions.

In conclusion, medical imaging has become an indispensable tool in cancer care. Its diverse modalities provide invaluable information that empowers clinicians to improve patient outcomes at every stage of the fight against cancer. Although several imaging technologies play an important role in the diagnosis of breast cancer, mammography remains the gold standard for detecting and evaluating suspicious breast lesions [76]. This is due to its effectiveness in identifying early-stage cancers, even those without noticeable symptoms. In some cases, additional imaging modalities may be used in conjunction with mammography to provide a more comprehensive picture.

Real-time 3D ultrasound imaging is the technology that offers a detailed view of breast tissue, which is useful for differentiating between solid masses and fluid-filled cysts.

2D and 3D cone beam CT (CBCT) is a specialised CT technique that provides high-resolution 3D images, particularly useful for evaluating complex breast structures or planning biopsies. Full-field digital mammography (FFDM) is an advanced form of mammography that captures high-quality digital images, improves cancer detection and reducing radiation dose compared to traditional film mammography. The choice of additional imaging modality depends on individual factors and the specific needs of each case. Radiologists will consider factors such as the age of a woman, the density of the breast, and the findings of a mammogram to determine the most appropriate approach.

Digital mammography and film-screen mammography, which produce scanned photographic films, are the two categories in which the procedures fall. In particular, the FFDM approach produces digital images by replacing the X-ray films with electronics. A low-dose X-ray device is used during the general screening procedure to view abnormal clusters and irregular nodules inside the breasts. The process takes a few seconds. To compress and scan digital images [77] for mammography, the patient places his breast between two transparent plates as shown in Figure 6 For a bilateral comparison, the device should take two images: a craniocaudal view (CC) and a mediolateral oblique view (MLO). For a complete view of the breast and the identification of abnormal areas, proper placement is essential.

Figure 6: *Procedure for mammography screening [78, 79]*

Mammography uses two key X-ray projections to capture detailed images of the median oblique (MLO) view and Craniocaudal (CC) view.

The mediolateral oblique (MLO) view positions the breast at an angle, allowing for visualisation of most breast tissue, ideally including the nipple, in a single image (see

Figure 7). Craniocaudal (CC) view takes the X-ray directly overhead, showcasing the medial tissue (inner portion) of the breast. The major pectoralis muscle, a chest muscle, should be centred in the image and not obscured by breast tissue, particularly the axillary tail (outer edge) (see Figure 8). Obtaining clear views of MLO and CC is crucial for radiologists to thoroughly examine all areas of breast tissue for potential abnormalities.

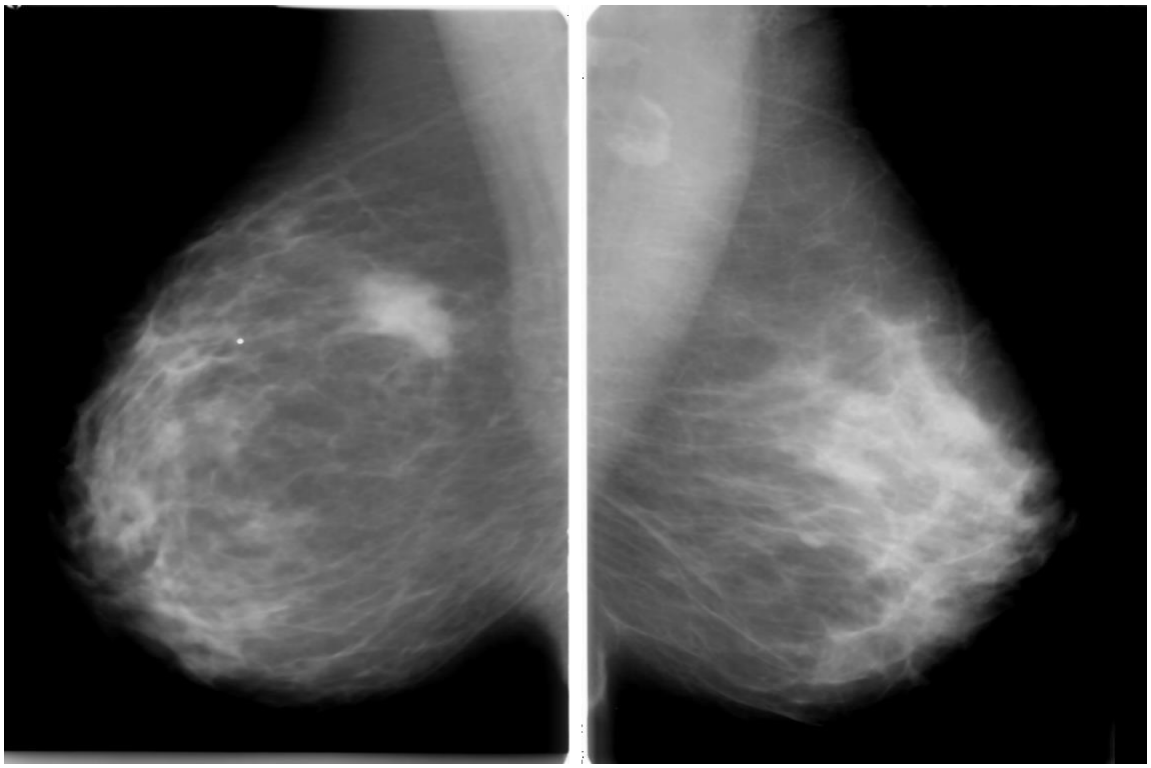


Figure 7: Mammography samples using the Mediolateral Oblique (MLO) view.

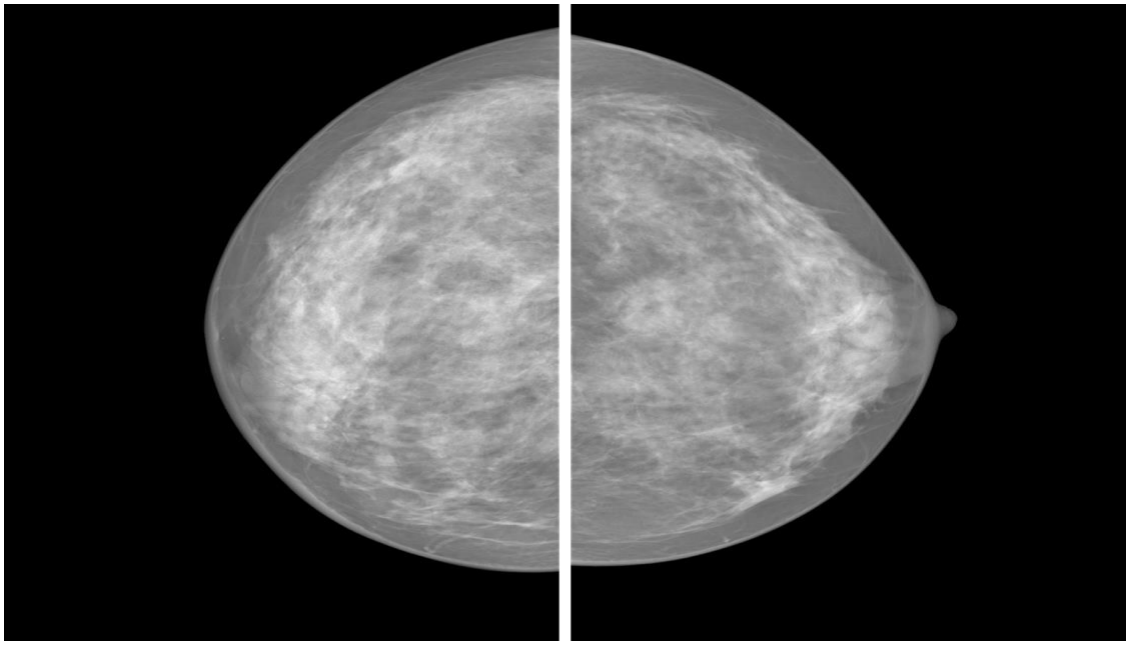


Figure 8: Mammography samples with craniocaudal (CC) views.

Radiologists play a critical role in the interpretation of mammograms, meticulously examining the images for potential signs of cancer risk. They focus primarily on identifying areas that can be caused by tumours or dense breast tissue. Radiologists can consider the location, size, and shape of these regions to determine if further investigation is needed. Distortions in the normal tissue pattern, such as asymmetry (unevenness between the breasts), can warrant further evaluation.

A well-defined and dense white mass on a mammogram is a concern, as it could be a malignant tumour. These tumours often grow and change shape over time. However, it is important to note that dense breast tissue can also appear white on mammograms. Radiologists can use their expertise to differentiate between the two. Although mammograms can reveal suspicious areas, it is important to understand the difference between benign and malignant findings. Benign are non-cancerous lumps or bumps that typically appear differently from malignant tumours on mammograms. They generally do not pose a threat to health and are not expected to spread.

Mammograms can sometimes reveal tiny white spots or patches. These are often benign (not cancer) and can be caused by uneven tissue density. However, it is important to monitor them over time to identify any changes that might require further investigation [78]. If the white spots or patches become larger, more defined, or develop irregular shapes, a radiologist might recommend additional tests. Although

some calcifications are benign, new clusters or changes in existing calcification patterns can be a concern. By regularly monitoring these white areas and any changes, radiologists can potentially detect early signs of cancer development, leading to faster diagnosis and treatment. The mammogram images in Figure 9 (assuming the figure shows mammogram examples) are included for illustrative purposes. These images were obtained from various sources, including public and private databases, and may have been captured using different techniques, such as full-field digital mammography (FFDM) or digital mammography screening.

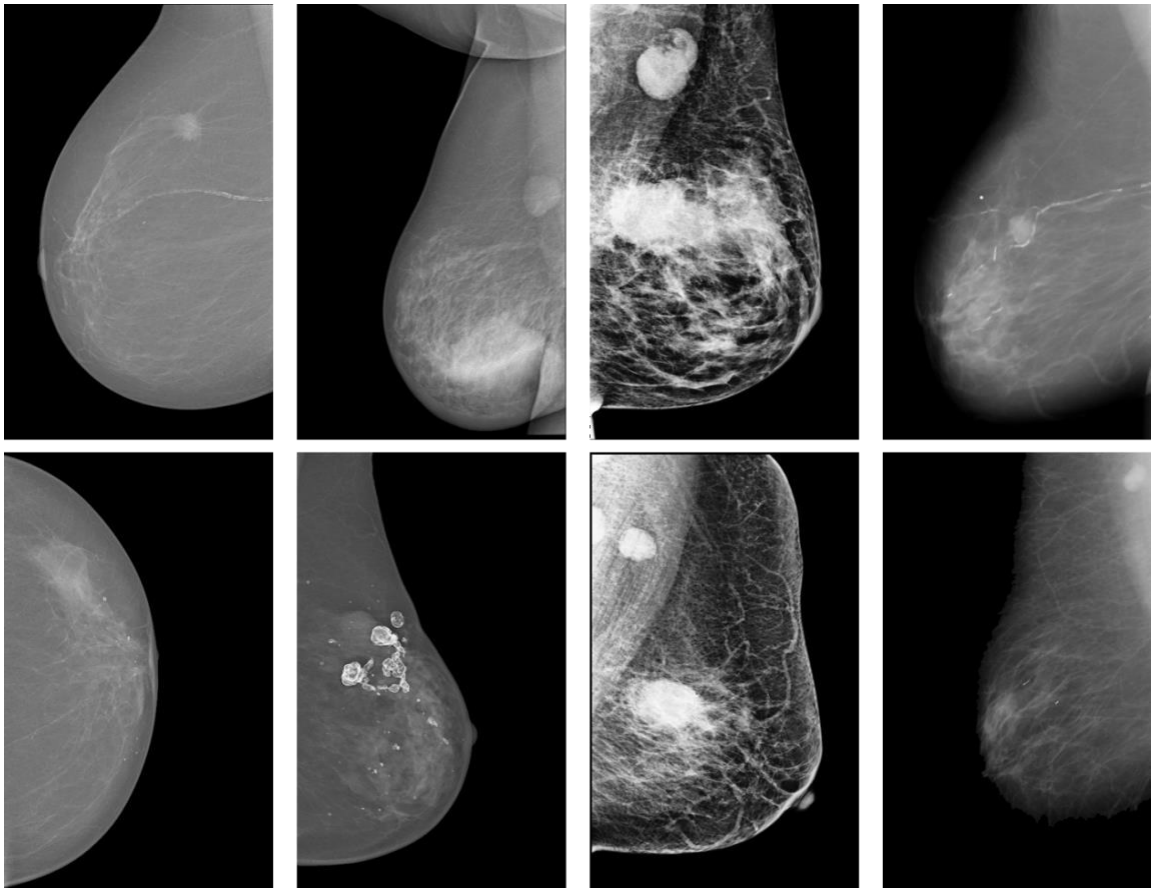


Figure 9: Mammography samples showing mass lesions in the first row and calcification lesions in the second row.

The mammogram images in Figure 9 (if the figure shows mammogram examples) illustrate the natural variations in the appearance of the mammogram. These variations can occur due to the following: resolution differences and image capture techniques. The quality can vary depending on the specific equipment used. The different techniques used to capture and store mammogram images can also influence their appearance.

It is important to note that mammograms can show one or more lesions (abnormalities) of various sizes and locations, regardless of the image quality. However, the expertise of a radiologist is crucial in interpreting these variations and identifying potential signs of concern. Mammography plays a vital role in detecting breast abnormalities. Radiologists meticulously analyse mammogram images, often comparing multiple images taken at different times, to identify any new or changing lesions; these can be masses, lumps, or suspicious areas that may require further investigation.

Figure 10 (if the figure shows mammogram examples) illustrates various types of breast mass lesions that mammograms can detect. Early detection of these abnormalities is crucial for timely diagnosis and treatment of potential breast cancer. A breast mass is a noticeable lump or growth in breast tissue. It can cause changes in breast texture, colour, size, or density and often feels firm or hard. These masses can be benign (non-cancerous) or malignant (cancerous). Radiologists use various methods to differentiate between benign and malignant breast masses. Here are some factors they consider; Mammogram appearance can include the shape (round, irregular), the margins (well-defined, blurred), and the density (uniform, variable) of the mass on a mammogram. The radiologist can feel the breast to assess texture (hard, soft), mobility (fixed, movable), and edges (distinct, unclear). Magnetic resonance imaging (MRI) can be used to see whether the mass absorbs contrast dye quickly (potentially cancerous) or slowly (more likely benign). A tissue sample is extracted and analysed under a microscope to determine the presence or absence of abnormal cell characteristics. Although breast masses can be concerning, early detection is crucial to successful treatment. If you notice a lump or any changes in your breast, it is important to consult with a doctor for prompt evaluation.

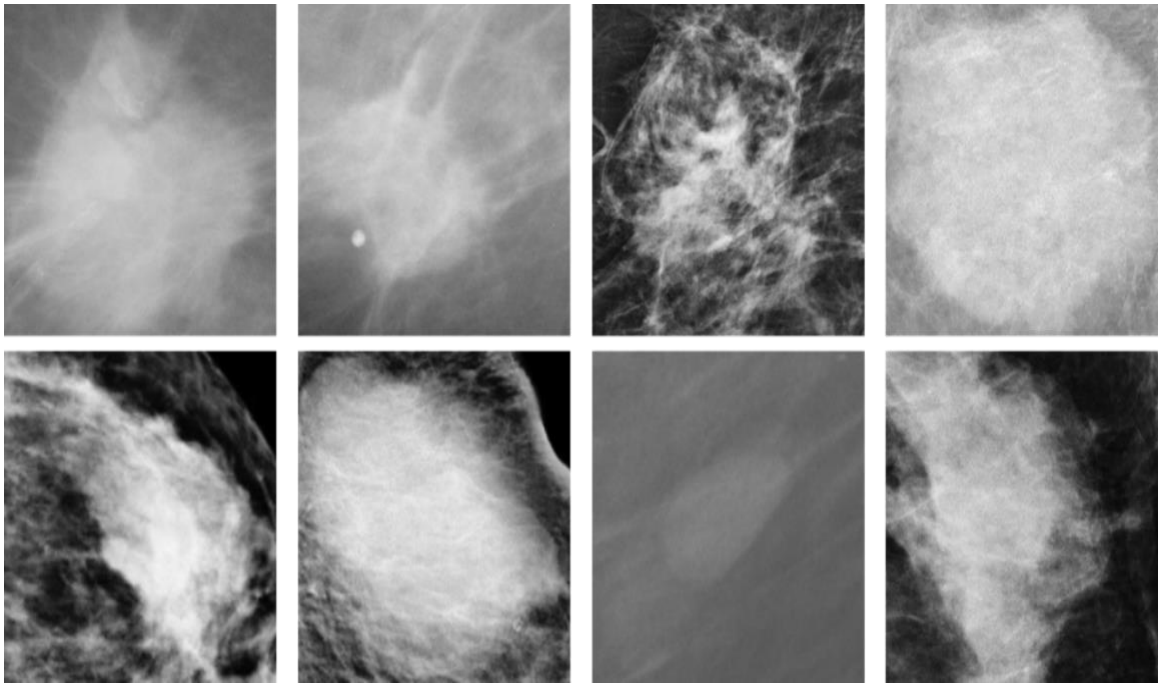


Figure 10: breast mass from both public and private data sets that represent the region of interest (ROI).

Figures 11 and 12 also show calcification and architectural distortion, two noncancerous breast abnormalities. Calcifications are often small calcium deposits that appear as white patches and dots. Calcifications are tiny clusters of calcium deposits that can sometimes form in breast tissue. During a mammogram, these calcium deposits appear bright white on the X-ray images (if Figure 11 shows mammograms with calcifications). This is because calcium absorbs X-rays more effectively than surrounding breast tissue. Calcifications are benign cells that eventually form and are not related to breast cancer [79]. However, they could be related to the existence of ductal carcinoma in situ. Large calcifications, or macro calcifications, are less speculative, according to studies. They should look like distinct flecks and spots. Microcalcifications look like tiny flecks and are smaller, but they need more attention.

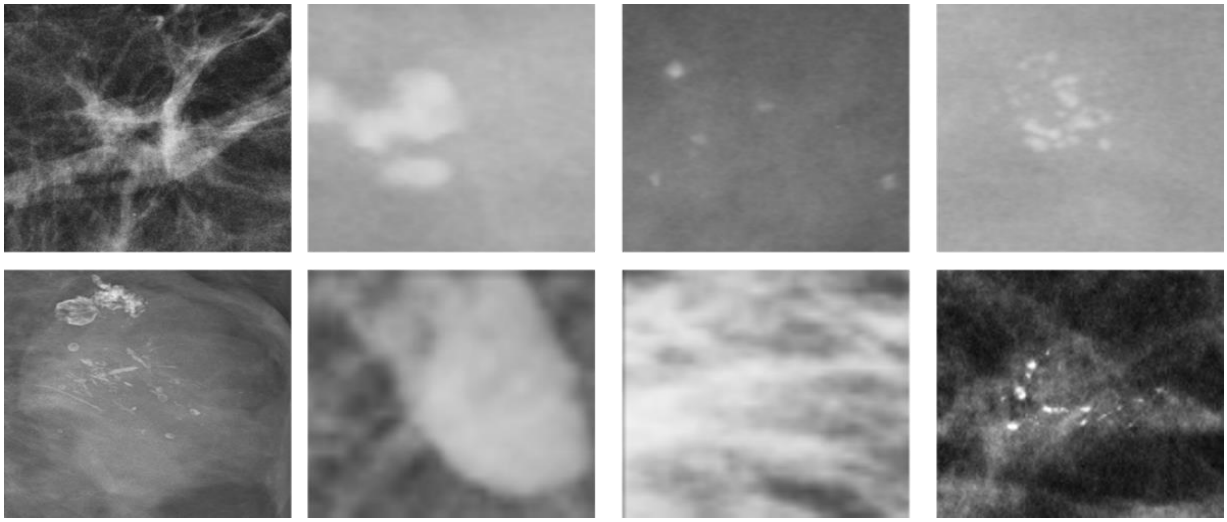


Figure 11: samples from public and private databases that represent the region of interest (ROI) of breast calcification lesions.

Architectural distortion, which is another typical benign breast lesion, refers to an evident distortion of the breast lobe without a definite mass [80]. These benign lesions, as depicted in Figure 11, frequently take the shape of straight, thin spiculum lines, distortion, and localised retraction [78]. The third most frequent breast lesion, architectural distortion, is often accompanied by calcifications or asymmetries and may be a sign of breast cancer in its early stages. Furthermore, due to its changing presentation, size, and location, it is thought to be the most difficult to diagnose, especially in 2D mammography.

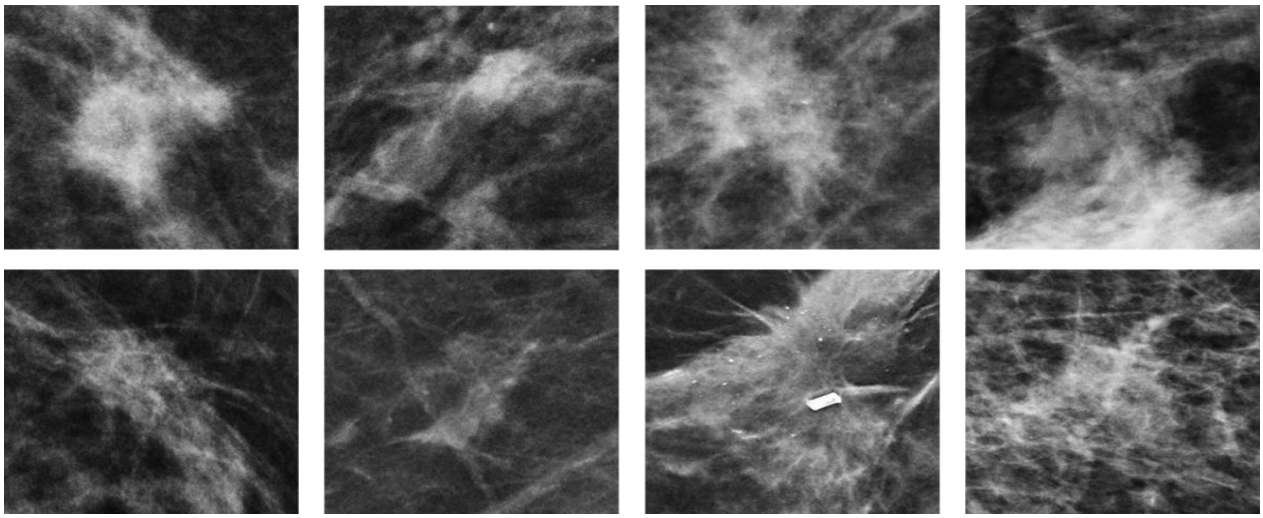


Figure 12: Samples of architectural deformation lesions in the breast from both public and private datasets, Regions of interest (ROI).

3.4. Category of breast cancer

After a mammogram, radiologists use the Breast Imaging Reporting and Database System (BI-RADS) score to categorise their findings. This score helps determine the likelihood of cancer and guides the next steps. Women over 40 are more likely to receive scores between 0 and 2, indicating normal findings or those likely benign. However, any score of 3 or higher warrants further evaluation by a radiologist to determine the most appropriate course of action, which may include additional imaging or a biopsy.

Category 0: A BI-RADS score of 0 indicates an incomplete mammogram. This can happen for a few reasons. Technical issues, sometimes the problem with the mammogram machine or image quality can make it difficult for radiologists to interpret the results. If important views of breast tissue are not captured during the mammogram, a score of 0 could be assigned. When a mammogram receives a BI-RADS score of 0, it is crucial to schedule additional imaging to obtain a complete assessment. This may involve a new mammogram with adjustments that may be needed to ensure clear images. Radiologists often compare new mammograms with previous ones to identify changes in breast tissue over time. Although an incomplete mammogram can be frustrating, it does not necessarily indicate a problem. By scheduling a follow-up exam as recommended by your radiologist, you can ensure a thorough evaluation and early detection of potential problems.

Category 1: A BI-RADS score of 1 on the mammogram is excellent news. It indicates a negative result, meaning that the radiologist did not find signs of cancer. This typically indicates that the mammogram shows an even distribution of breast tissue throughout both breasts. No lumps, masses, or calcifications raise concerns. Continued screening is important because even with a negative mammogram, it is crucial to maintain your regular screening schedule. Here is why mammography is more effective at detecting cancer in its early stages when it is often easier to treat successfully. Breasts can change with time, so regular screenings allow radiologists to monitor for new developments. A BI-RADS score of 1 provides peace of mind that your current mammogram is normal. However, remember that continued screening remains vital to maintaining your breast health and ensuring early detection of potential problems in the future.

Category 2: A BI-RADS score of 2 on the mammogram indicates that the findings are highly likely benign (not cancerous). However, your radiologist may recommend a follow-up mammogram in a specific time frame for monitoring purposes. The mammogram shows generally normal breast tissue with no signs of immediate concern. The radiologist may identify noncancerous lumps or cysts in your report. These typically do not require treatment, but will be documented for future reference. Although the score suggests a low risk, the radiologist might recommend a follow-up mammogram in a specific time frame. This is a precautionary measure to monitor any changes in the benign finding. By comparing your current mammogram with those of the future, doctors can ensure that any potential concerns are addressed promptly.

Category 3: A BI-RADS score of 3 on the mammogram indicates that the findings are highly likely benign (not cancerous). However, your radiologist may recommend a follow-up mammogram at 6 months for further evaluation. The mammogram generally shows normal breast tissue, but there may be an area that requires a closer look. The chance of cancer with a score of 3 is around 2%. A follow-up mammogram in 6 months allows radiologists to monitor any changes that are concerned. This can help prevent unnecessary biopsies and ensure early detection of any potential problems.

Category 4: A BI-RADS score of 4 on the mammogram indicates a suspicious finding that requires further investigation. Although the chance of cancer is between 20-35%, it is important to determine the cause for a more definitive diagnosis. The radiologist identified an area in your mammogram that requires further evaluation. A biopsy, which involves taking a small tissue sample, is often recommended to confirm or rule out cancer. This category is further divided into three subcategories according to the level of suspicion:
4A (low suspicion): This indicates that the finding is less likely to be cancerous.

4B (Moderate suspicion): The likelihood of cancer increases in this category.

4C (high suspicion): This suggests a high probability of cancer and a biopsy is strongly recommended. The radiologist can discuss the specific details of your case and the recommended course of action based on your BI-RADS score (4A, 4B, or 4C). This

may involve additional imaging tests or a biopsy to determine the best course of treatment.

Category 5: A BI-RADS score of 5 on the mammogram indicates a high suspicion of cancer. In this case, the probability of cancer is very high, around 95%. This score suggests a strong possibility of cancer and a biopsy is strongly recommended to confirm the diagnosis. A biopsy, which involves taking a small tissue sample, is crucial to determine whether cancer is present and guide the best course of treatment. The radiologist can discuss the specific situation and urgency of a biopsy. Radiologists also explain the different treatment options available if cancer is confirmed.

Category 6: A BI-RADS score of 6 is assigned only after a biopsy confirms the presence of breast cancer. This category compares previous mammogram images to assess how cancer responds to treatment. A biopsy has already identified breast cancer. The BI-RADS 6 score uses past mammogram images along with current ones to track changes in cancer and evaluate the effectiveness of treatment (chemotherapy, surgery, radiation). Regular follow-up appointments and imaging tests are crucial after a cancer diagnosis. The BI-RADS scores assigned during these follow-ups help doctors monitor your progress and adjust treatment plans if necessary.

3.5. Breast Image Processing

Image manipulation, in the medical field, refers to the process of modifying digital images to improve their clarity and reveal important details for accurate diagnosis. Radiologists use image processing techniques to sharpen specific features within an image, such as tumours or abnormalities, making them easier to identify. Image manipulation can help reduce grainy or blurry areas in an image, allowing a clearer view of underlying structures. Sometimes image-processing techniques are used to standardise the appearance of medical images, ensuring consistency for analysis and comparison over time. By manipulating medical images, radiologists can gain valuable insights that might not be visible in a raw image, which can lead to earlier and more accurate diagnoses, improved treatment planning and decision-making, and better monitoring of patient progress. Various image processing techniques are used in medicine, some of which include [81]. Optimising the image capture settings to obtain

the clearest possible image from the beginning. Enhancing specific features within the image for better visualisation. Separation of different structures or tissues within an image for individual analysis. In general, image manipulation plays a vital role in modern medicine, allowing radiologists to extract crucial information from medical images, leading to better patient care. Digital images, such as those used on medical scans or mammograms, go through various processing steps to make the information they contain clearer and more useful. These steps can be broadly classified as basic, advanced, and specialized [82].

Basic processing involves adjusting brightness, contrast, and sharpness to make it easier to see and analyse subtle details in the image. Smoothing out grainy or blurry areas in the image allows for a clearer view of what is beneath [81, 82]. Advanced Processing is a special technique that can highlight specific features within the image, such as tumours or abnormalities, making them stand out for easier identification by radiologists. Advanced processing can separate different structures or tissues within an image, allowing for individual analysis. Specialised Processing in this category involves more complex techniques used for specific medical applications, such as 3D reconstruction for in-depth analysis or image manipulation for standardised comparisons over time [80, 82]. In general, digital imaging processing helps radiologists extract the most valuable information, leading to better diagnoses, treatment plans, and patient care.

3.6. Procedures to follow to process images.

Developing an application that analyses or manipulates digital images involves several crucial steps. While Figure 13 illustrates the main stages, many sub steps and strategies come into play within each one.

Image input is the initial step that involves loading the digital image file into the application for processing. Basic Processing, Advanced Processing, and Specialised Processing are explained in Section 3.5. Output is the final step in which the processed image or the extracted information is presented in a user-friendly format, such as displaying the image on the screen or saving the results as a file. Following

these essential stages, developers can create powerful digital image processing applications that benefit various fields, including medicine, science, and engineering.

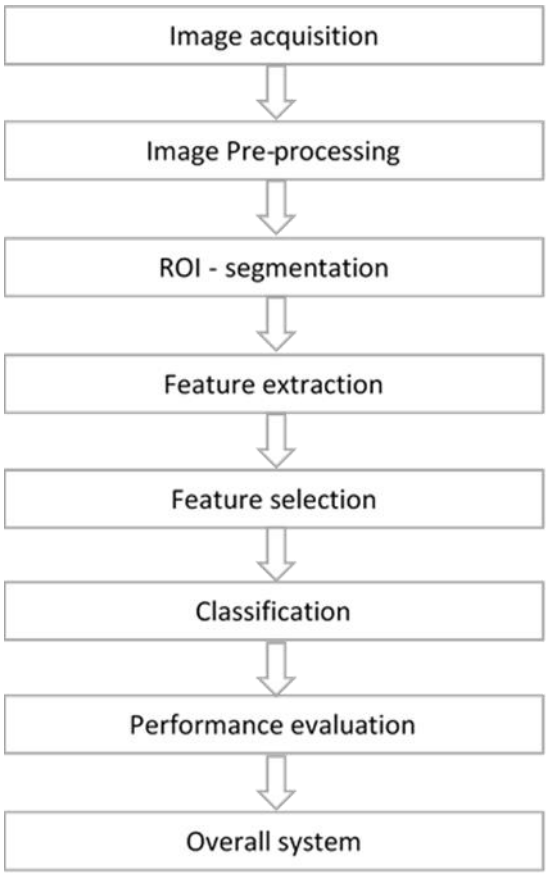


Figure 13: steps in image processing [83 – 85].

3.6.1 Image Acquisition

Every image processing project starts with the crucial step of image acquisition. This phase involves capturing the original image from its source, whether it is a camera, a scanner, or a digital file [83]. There are various methods for achieving image acquisition, each suited for different applications. While capturing images of our surroundings is a common image acquisition method, it is not the only one. We can also acquire images by retrieving existing image files from any electronic source. This retrieval process can involve various collection approaches, such as downloading from the Internet, transferring from a storage device, or accessing an image database. Regardless of the method, the acquired image remains unprocessed until it enters the image processing programme. The image's origin lies in the hardware used for its

creation. This hardware can range from a compact desktop scanner to a powerful astronomical telescope. In certain fields, such as scientific research, maintaining a consistent starting point is crucial for accurate comparisons and analysis.

Many image processing applications, such as face recognition, object identification, and plant disease detection, benefit from using raw images. These raw images can be captured from various sources depending on the application. Common capture devices include digital cameras, mobile phone cameras, and webcams for environmental situations. Image acquisition extends beyond capturing our surroundings. Medical imaging equipment, such as X-ray machines and MRI scanners, provides raw images crucial for medical diagnosis, such as tumour or cancer detection. Similarly, satellites transmit images used for geographic information systems (GIS) and other applications related to land management or environmental monitoring [84].

3.6.2 Image Pre-processing

Following image acquisition comes the crucial step of image pre-processing. This process refines the image data to prepare them for computer analysis. Pre-processing techniques include filtering noise, correcting lighting variations, and removing artefacts. Filtering can target specific issues such as background noise or uneven lighting, ultimately improving the clarity and consistency of image features. Formatting involves defining the characteristics of the image, such as size and data format, for efficient storage and processing [85]. Although digital image processing dominates the field, optical and analogue techniques still have their place. Now, let us dive into some common image pre-processing tasks.

3.6.2.1. Image resizing

One crucial technique is image resizing, often confused with compression. Resizing reduces the dimensions of the image (width and height) to a smaller size, improving computational efficiency. However, it is essential to only resize if the resulting image retains its original meaning and key features. Significant reductions can distort the image and hinder analysis [86]. Another key component of pre-processing is image

conversion. This involves transforming the image format or colour space to optimise processing speed or analysis. For instance, converting a colourful image to greyscale can simplify computations when analysing features that do not rely on colour information. However, conversion should be done strategically. Colour carries valuable information, and losing it can hinder tasks like disease detection in medical imaging, where colour variations are crucial. In such cases, the image might be left unconverted or converted back to colour later in the process [87]. Image pre-processing encompasses a variety of techniques, including image augmentation. Augmentation artificially expands the data set by creating variations of the original image. This can improve the performance of machine learning models by making them more robust to slight variations in real-world data. Contrast adjustment is a common example of augmentation.

3.6.2.2. Image Restoration vs. Enhancement

It is important to distinguish between image enhancement and restoration. Restoration aims to restore an image to its original state by correcting issues such as noise, blur, or artefacts. This process is objective and is often guided by mathematical models to reverse the degradation. In contrast, image enhancement modifies an image to improve its visual appeal, such as adjusting brightness or colour. Enhancement is subjective, as what constitutes "better" depends on human perception and the intended use of the image [88].

3.6.3 Image Segmentation

Image segmentation, also known as Region of Interest (ROI) identification, is a fundamental technique in image processing. It involves dividing an image into meaningful segments, such as objects, groups of pixels with similar characteristics, or even individual pixels. These segments often correspond to what we naturally perceive as separate objects in the image. The segmentation process relies on various image data points, such as colour information, boundaries between regions, or even specific textures within an image fragment. The accuracy of segmentation directly impacts how well features can be extracted from the image and how precisely objects are detected.

The choice of image segmentation algorithm depends on the specific image and the desired result. There are two main categories: region-based and edge-based approaches [88, 89]. Region-based methods group pixels with similar characteristics, such as colour or texture, into segments. Edge-based methods identify boundaries between objects by analysing sharp changes in pixel intensity. Within these categories, there are numerous algorithms, each with its strengths and weaknesses. We will explore some of the most popular techniques in more detail later. Structural segmentation techniques leverage prior knowledge about the image content to segment the image. This knowledge can come from the image itself or from external sources [82, 90]. Unlike structural segmentation, which relies on prior knowledge, stochastic techniques segment images based on the statistical properties of individual pixels within the image [90].

Hybrid techniques, as the name suggests, combine the strengths of structural and stochastic approaches. They use both prior knowledge about the image content and the statistical properties of individual pixels to achieve more robust segmentation [90]. There are several image segmentation techniques, each suited for different scenarios. Threshold is a simple technique that segments an image on pixel intensity. Pixels above or below a certain threshold are assigned to different segments. Edge-based segmentation is the approach that identifies object boundaries by analysing sharp changes in pixel intensity. Region-based segmentation This method groups pixels with similar characteristics, such as colour or texture, into segments. Clustering-based segmentation is the technique that groups pixels with similar features, creating distinct segments without predefined criteria. Watershed segmentation is like flooding a landscape; this method separates image regions based on their intensity gradients, treating darker areas as valleys and brighter areas like hills. Deep learning-based segmentation is a powerful approach that uses artificial neural networks to learn complex patterns within images, allowing highly accurate segmentation of intricate objects. This list is not exhaustive, but provides a foundation for understanding common segmentation methods. The choice of technique depends on the specific characteristics of the image and the desired segmentation result.

3.7. Breast Image Analysis

Medical image analysis plays a crucial role in healthcare care, enabling us to leverage the rich data within medical images for diagnosis, visualisation, and further study. This field has seen significant advancements, particularly with the rise of deep learning (DL) algorithms. Deep learning empowers computers to automatically identify patterns and features in medical images, leading to improved accuracy in tasks like anomaly detection. Radiologists are highly dependent on medical image analysis techniques to diagnose abnormalities on X-ray images, such as fractures or tumours. This analysis helps them determine whether the image is normal or abnormal. While computer-aided diagnosis (CAD) systems have been around since the 1980s to assist radiologists, they can sometimes generate false positive results. Deep learning offers a promising path to overcome these limitations and improve diagnostic accuracy [91]. Unlike regular image processing, which often focusses on enhancing aesthetics or artistic qualities, medical image processing has a singular goal: to maximise the diagnostic value of the image. This might involve improving the image itself to better visualise specific structures or automatically extract relevant information [92]. A crucial task in medical image processing is differential diagnosis, which involves the use of medical images to distinguish between different diseases. This can range from pinpointing the specific type of cancer to simply determining the presence or absence of a medical condition [93]. Analysing medical images can be time consuming for doctors. Since the 1980s, computer-aided diagnosis (CAD) systems have been introduced to streamline this process and free doctors' time for other tasks [92]. However, early CAD systems often generated many false positives. This not only increased overall evaluation times, but also led to unnecessary biopsies, adding a burden on patients and the healthcare system [93].

Advancements in computer vision have significantly transformed medical imaging, leading to improvements in image capture, disease detection, treatment planning, and even prediction. Computer vision techniques can analyse medical images to extract valuable data, such as texture, shape, contours, and even historical information from previous scans. This analysis can generate 3D and even 4D data, providing a more comprehensive picture for physicians compared to traditional 2D images.

Automated processing of 3D medical images plays a critical role in improving diagnostic accuracy and treatment outcomes. Furthermore, this automation opens new avenues for research in computer vision, medical graphics, and robotics. A key challenge in this field is access to a large pool of well-diagnosed medical images, which is essential for developing and validating robust algorithms.

3.8. Deep learning

Deep learning is a powerful subfield of machine learning that excels at extracting complex patterns from vast amounts of data. Unlike traditional machine learning algorithms, deep learning models can learn autonomously through a hierarchy of artificial neural networks. These networks, inspired by the human brain, process information at increasing levels of abstraction, allowing them to identify intricate relationships within the data. This capability has made deep learning a game changer in artificial intelligence, offering promising solutions to a wide range of challenges [94]. Deep learning's real breakthrough lies in its ability to handle complex tasks with vast amounts of data, while still being computationally efficient. This has led to significant advancements in image recognition, speech recognition, and other areas requiring intricate data analysis. Deep learning models can be trained in various ways depending on the problem and available data. Supervised learning In this approach the model is trained on labelled data, where each data point has a corresponding correct answer. The model learns to map the input data to the desired output. In unsupervised learning, the model is presented with unlabelled data and must identify patterns or relationships within the data itself, without any predefined categories. Semi-supervised learning is the approach that combines labelled and unlabelled data, using the labelled data to guide the learning process on the vast amount of unlabelled data.

Although deep learning has shown remarkable success, more research is crucial to address the specific challenges faced within our community. This ongoing exploration will ensure that deep learning reaches its full potential for solving real-world problems.

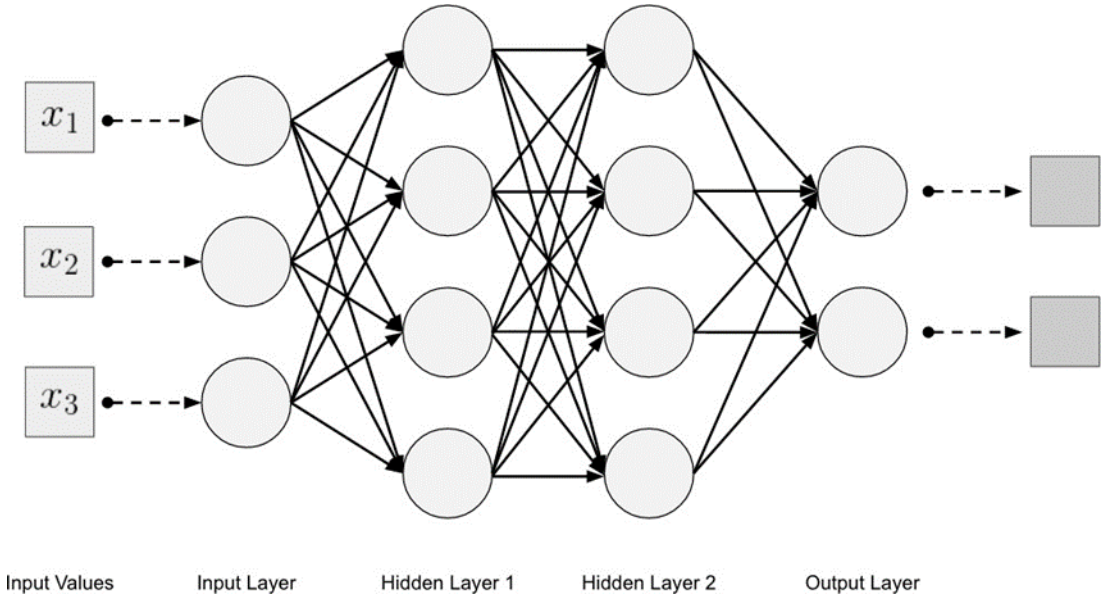


Figure 14: Architecture for Deep Learning [30, 98]

Convolutional neural networks (CNN) are a powerful architecture specifically designed for image analysis [95]. They excel at feature extraction and classification tasks. Feature extraction: At this stage, CNN scans the image using filters to identify and extract relevant features, such as edges, shapes, and textures. Techniques like combining and running are used to reduce data size while preserving key information. Classification: Once the features are extracted, CNN uses a fully connected neural network to analyse them and categorise the image. This classification assigns the image to a specific class, such as "cat" or "dog" in an image recognition task.

Deep Convolutional Neural Networks (CNNs) have become a cornerstone of computer vision and machine learning due to their exceptional performance [96]. A key factor contributing to their success is the use of multiple feature extraction stages. These stages automatically learn data representations from the input, progressively extracting increasingly complex features. Several factors have fuelled the surge in CNN research: Increased data availability: Large datasets are crucial for training complex deep learning models, and the recent growth in accessible data has empowered CNN development. Advancements in Hardware Technology: Powerful GPUs and specialised hardware have made it possible to efficiently train these computationally demanding models. Emerging CNN architectures: Researchers have continuously developed innovative CNN architectures, pushed the boundaries of performance, and tackled new challenges. Architectural innovation has been a driving

force in the development of convolutional neural networks (CNNs). While other factors such as activation functions, loss functions, and parameter optimisation play a role, novel CNN architectures have significantly pushed the limits of performance. Deep learning excels at learning representations of data, and CNNs are a prime example. These models do not require separate feature extraction and classification stages. Instead, they learn these features automatically during the training process through their unique architecture [94].

3.8.1. CNN and Conventional Classification Methods: A Comparison

Unlike traditional classification techniques that rely on hand-crafted features, CNNs excel at automatically learning features directly from the data. This makes them particularly powerful for complex image-recognition tasks. The extraction that is based on expertise is not necessary for CNN [29]. Unlike traditional methods that rely on hand-crafted features, CNNs can automatically learn these features directly from the data. This eliminates the need for human expertise in feature engineering, which can be time consuming and domain-specific. CNNs have achieved state-of-the-art performance in various computer vision tasks, including image classification, object detection, and image segmentation. Their ability to learn complex patterns from large amounts of data makes them highly effective. CNNs require a significant amount of labelled data for effective training. This can be a challenge for tasks where labelled data are scarce. Training CNNs involves many learnable parameters and complex calculations. This necessitates powerful computational resources, such as GPUs, which can be expensive.

Convolutional neural networks (CNNs) are a powerful type of deep learning model specifically designed for image analysis tasks. They excel in various applications, including image classification: Categorising images into different classes (e.g., classifying an image as containing a cat or a dog). Object detection: Identifying and locating objects within an image (e.g., detecting people or cars in a scene). Face recognition: Recognising and verifying the identity of individuals from images. Image segmentation: Dividing an image into meaningful segments (e.g., segmenting an image to separate the foreground from the background).

CNNs achieve high-level feature extraction through their unique architecture, which typically consists of four main layer types: convolutional layers, pooling layers, activation layers, and fully connected layers [97]. This layered architecture allows CNNs to learn features directly from the data, eliminating the need for manual feature engineering, a significant advantage compared to traditional machine learning methods.

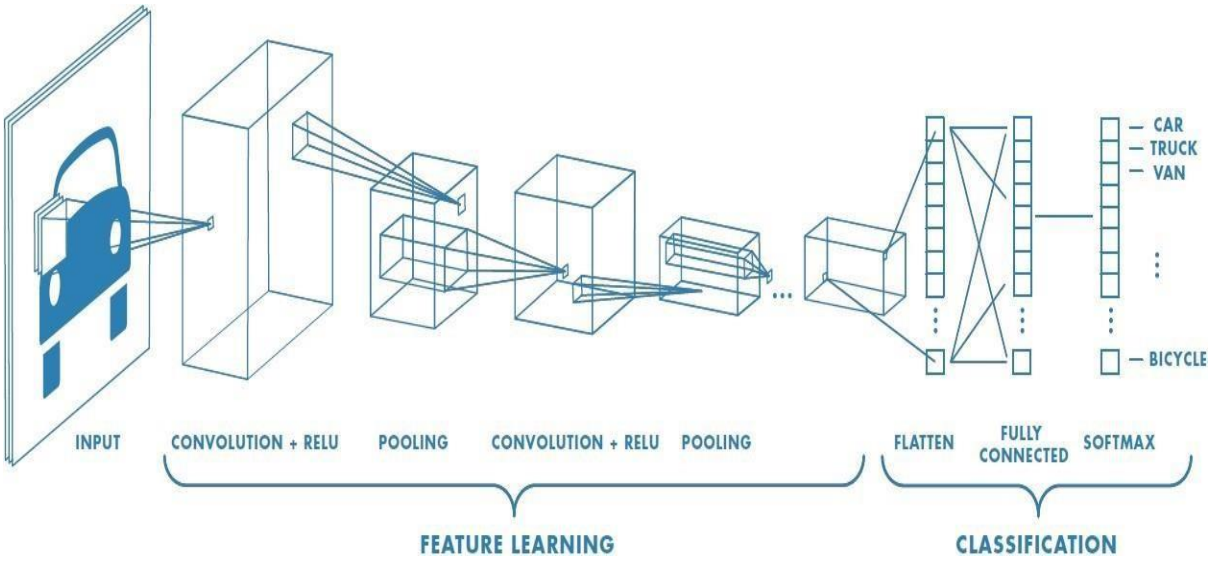


Figure 15: Example of CNN Layers [91, 113]

Convolutional layer: The core building block of CNNs is where the magic happens. In this layer, the network extracts feature from the input image through a process called convolution. Imagine a filter (also called a kernel) as a small window that slides across the image. As the filter moves, it multiplies the element-wise values of the image with the corresponding values. The sum of these products is then stored in a new image, called a feature map [98]. Figure 15 shows the mathematical operation. This process continues as the filter slides across the entire image, generating a feature map that highlights specific features the filter was designed to detect (e.g., edges, shapes, textures).

Pooling layer: Following the convolutional layer is the pooling layer. This layer serves two main purposes. It reduces the size of the feature maps generated by the convolutional layer. This helps to manage computational complexity and memory usage during training. Pooling introduces a degree of tolerance to minor variations in

the image, such as small shifts or distortions. This helps the network focus on the larger picture and reduces sensitivity to irrelevant details. There are different pooling techniques, but the most common is max pooling. In max-pooling, the pooling layer takes a small window (often 2x2 pixels) and outputs the maximum value within that window. This essentially captures the most dominant feature within that local area of the feature map. Other pooling methods include average pooling and sum pooling, which use the average or sum of values within the window, respectively [99].

Activation function: plays a crucial role in CNNs, especially for deep learning applications like image classification and object detection. These functions introduce non-linearity into the network, a critical property for learning complex patterns in the data [100]. Each neurone in a CNN layer receives input from multiple sources and assigns weights to these inputs. The activation function takes the weighted sum of these inputs. Based on the weighted sum, the activation function determines if the neurone should "fire" (output a value) or not. This essentially acts as a gatekeeper, controlling the flow of information within the network. The choice of activation function can significantly impact model performance. Common activation functions used in CNNs include ReLU (rectified linear unit) and Sigmoid. The selection of the right activation function depends on the specific task and the network architecture.

Dense layer (completely connected layer): The fully connected layer plays a critical role in the final stages of image classification and object detection tasks within CNNs. After the convolutional and pooling layers have extracted and reduced the features of the image, the fully connected layer takes over [101]. It transforms the output from the pooling layer, which is typically a 3D array of feature maps, into a single 1D vector. This allows the network to process all the extracted features together. The fully connected layer acts as a traditional artificial neural network. It uses this flattened feature vector and performs a series of computations to determine the most likely class for the image (e.g., "cat" or "dog" in an image classification task). Without the fully connected layer, CNNs would not be able to translate the extracted features into meaningful classifications, making it an essential component for image recognition tasks.

Overfitting: is a common challenge in machine learning, including deep learning. It occurs when a model memorises the training data too well, including noise and irrelevant details. This leads to the model performing well in training data but poorly in unseen data, which hinders its generalisability [102]. To avoid overfitting, it is crucial to assess a model's performance on a separate test set that the model hasn't been trained on. This helps to identify whether the model is truly learning the underlying patterns or simply memorising the training data.

Several techniques can help reduce overfitting in deep learning, as summarised in Table 1. These techniques aim to make the model less sensitive to noise and specific details in the training data, promoting better generalisation of new data.

Table 1: Overfitting mitigation technique.

Methods to reduce over-fitting	
1	sufficient training data
2	Data enhancement
3	Regularisation (a dropout or weight loss)
4	Bundling normalisation
5	Simplify the architecture

3.9. Transfer Learning

Transfer learning is a powerful technique in deep learning that addresses the challenge of training complex models with limited data. It involves leveraging knowledge gained from a pre-trained model on a related task and applying it to a new task. Training deep learning models from scratch can be extremely time-consuming. Transfer learning allows you to reuse the weights from a pre-trained model, significantly accelerating the training process for your specific task. Pre-trained models on large datasets like ImageNet have already learnt powerful feature representations that can be beneficial even for new tasks. By fine-tuning these pre-trained models, you can often achieve better performance compared to training a model from scratch on a smaller dataset. Several pretrained models are readily available for transfer learning, including VGG (VGG16, VGG19), InceptionV3 and ResNet [101]. These models were

trained on massive datasets with millions of images using powerful computing resources. This pretraining allows them to capture valuable generic features that can be adapted to various computer vision tasks, including those with limited data. Choose a pre-trained model that was trained on a task like your new task. The pre-trained model's weights are not directly used. Instead, the final layers of the model are re-trained on your specific dataset to adapt the learnt features to your new task. Using transfer learning, it can achieve good performance in the task even with limited data, making it a practical approach for many real-world deep learning applications.

3.9.1 Alex Net

AlexNet, developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, is a pioneering convolutional neural network (CNN) architecture that achieved remarkable success in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It significantly outperformed previous models by using several innovative techniques. AlexNet utilised a deep architecture with eight layers, including five convolutional layers and three fully connected layers. This depth allowed the model to learn complex feature hierarchies for image recognition. The model was trained on a massive dataset of 1.2 million images from ILSVRC, categorised into 1000 classes. This large amount of data provided the network with a rich source of information to learn effective image representations. To prevent overfitting, AlexNet used dropout regularisation, a technique that randomly drops out a certain percentage of neurones during training. This helps the model avoid memorising specific training examples and promotes better generalisation to unseen data. The model used the Rectified Linear Unit (RLU) activation function in its convolutional layers. ReLU offers advantages such as computational efficiency and the problem of ability to address the vanishing gradients that can hinder training in deep neural networks [103]. Overall, AlexNet's architecture, with its deep layers, large dataset, and innovative regularisation techniques, paved the way for the advancement of CNNs in computer vision tasks.

3.9.2 VGG19

VGG19, developed by K. Simonyan and A. Zisserman, is another popular pre-trained model based on convolutional neural networks (CNNs). It achieved impressive results

in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) competition. VGG19 employs a very deep architecture with 16 convolutional layers, followed by fully connected layers. This depth allows the model to learn complex feature representations for image recognition. The model was trained on a massive dataset of more than 15 million high-resolution images from ILSVRC. This large amount of data is crucial for effectively training such a deep model. VGG19 relies on a relatively simple approach, primarily using 3x3 convolutional filters and 2x2 max pooling layers throughout its architecture. This focus on uniformity simplifies the network and potentially reduces the risk of overfitting. With more than 144 million parameters, VGG19 is a computationally expensive model [104]. However, its pre-trained weights can be leveraged for transfer learning in various computer vision tasks. Overall, VGG19 demonstrates the effectiveness of deep architectures for image recognition tasks. Its reliance on simple building blocks offers an alternative approach to achieve good performance.

3.9.3 GoogleNet

The GoogleNet, developed by Christian Szegedy et al. at Google, is a groundbreaking pre-trained model known for its exceptional performance and focus on computational efficiency. It achieved first place in the 2014 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), surpassing previous models while requiring less computational power [105]. GoogleNet introduced a novel concept called Inception modules. These modules incorporate multiple paths with varying filter sizes within a single layer. This allows the network to learn a wider range of features efficiently, reducing the need to stack numerous traditional convolutional layers. The use of Inception modules helps GoogleNet achieve high accuracy with fewer parameters compared to other deep CNNs. This translates to lower computational resources required for training and inference. Despite reducing computational cost, GoogleNet maintains a level of width (number of filters) and depth (number of layers) sufficient for effective feature learning [105]. In general, GoogleNet demonstrates that achieving high performance in image recognition does not necessarily require brute force in terms of computational resources. Its innovative architecture paves the way for more efficient deep learning models.

3.10. Related Works

This review analyses 29 recent research articles exploring computational techniques for breast cancer prediction.

Reference [103] tackles the problem of unbalanced data in breast cancer diagnosis. This is a common issue where there are far fewer positive cases (cancerous) compared to negative cases (healthy). The study proposes a method using deep transfer learning to improve the accuracy of breast cancer detection despite this data imbalance. This approach involves pre-training a deep learning model on a large and balanced dataset and then fine-tuning it for the specific task of breast cancer classification using the imbalanced breast cancer data. The study's findings suggest that this deep transfer learning method can effectively handle imbalanced data, potentially leading to more accurate breast cancer detection in clinical settings. Most machine learning algorithms assume balanced class distributions (equal numbers of positive and negative examples) in the data. However, real-world data often deviates from this assumption, leading to performance problems. The study uses publicly available structured breast cancer data to pre-train a deep-learning model. Their core hypothesis is that deep transfer learning based on structured data can improve the precision of early breast cancer detection and cancer classification. The study's findings support their hypothesis. The proposed deep transfer learning method on structured data demonstrates effectiveness in handling imbalanced class issues, suggesting its potential as a valuable tool for clinical research [33].

Reference [104] proposes a novel approach based on deep learning for fully automated breast cancer detection, achieving high precision in early-stage cancer identification. Histological images stained of breast cancer tissue undergo stain normalisation to ensure consistency. A deep convolutional neural network (DCNN) model extracts high-level features from pre-processed images. The study explores different DCNN architectures including Inception v3, Inception ResNet v2, Xception, and two VGGNet models. The extracted features are fed into a conventional multilayer perceptron classifier for final cancer classification. To address the challenge of limited datasets, the study uses data augmentation techniques to artificially expand the training data. The researchers tested their system on 400 breast cancer histology

images from the ICIAR 2018 Grand Challenge dataset. The proposed system achieved a promising accuracy of 92.5%, outperforming other models [106]. In general, this study highlights the potential of deep learning to develop automated breast cancer detection systems with high accuracy.

Kim, Park, and Hong [65] propose a new approach for the early and accurate detection of breast cancer using medical ultrasound imaging. The study introduces a novel model called Elfa-CRNN, which combines edge extraction with a CRNN (Revolutional Recurrent Neural Network). The Elfa-CRNN model analyses line segments within the mass identified in the ultrasound image. The researchers achieved a promising accuracy of 99.75% using Elfa-CRNN, outperforming other models such as CRNN, AlexNet, and VGG. In addition, the study explores different edge extraction techniques. They found that a combination of their proposed Elfa method with Canny and Sobel edge detection achieved a maximum accuracy of 98% [107]. Overall, this research highlights the potential of Elfa-CRNN to improve breast cancer detection accuracy using ultrasound images.

The incidence of breast cancer is increasing, and traditional radiographs are the main screening tool [33]. A convolutional neural network (CNN), a type of artificial intelligence (AI), is used to analyse mammograms. Preprocessing mammogram images with techniques like median filters and contrast enhancement improves the accuracy of the AI model in the detection of cancer. This suggests that AI-assisted mammogram analysis, especially with preprocessing, could be a valuable tool for improving breast cancer detection. AI can potentially act as a second reader alongside radiologists, helping to identify cancers in human eyes [108]. By finding cancer earlier, treatment can be more effective and lead to better outcomes for patients. Although promising, this is a developing area, and more research is needed to validate the effectiveness of AI in breast cancer screening in diverse populations. Overall, this is a positive development in the fight against breast cancer.

Anupama, Sowmya, and Soman [106] investigated the use of deep learning to classify different types of breast cancer. Their study used a technique called a capsule network, which is a type of artificial neural network architecture designed to process spatial relationships within images. The researchers used breast cancer image data

from the BACH 2018 Grand Challenge dataset [110]. This freely available data set allows researchers to compare and improve their methodologies. The authors acknowledge the importance of data preprocessing, which involves the preparation of the images for analysis by the capsule network. Additionally, they reference research suggesting that transfer learning, where a pre-trained model is adapted to a new task, can improve the performance of convolutional neural networks, another form of deep learning commonly used in image analysis [109]. This approach highlights the potential of deep learning, particularly capsule networks, for classifying breast cancer subtypes using histological images. Using publicly available datasets and incorporating data preprocessing and transfer learning techniques, researchers can potentially develop more accurate and robust models for breast cancer diagnosis.

A study by [110] explored the use of data mining to predict benign and malignant breast cancer. They analysed the Wisconsin breast cancer dataset containing 699 cases with nine clinical characteristics (eg, cell size homogeneity) for each patient. After removing entries with missing values, they built prediction models using three popular data mining techniques: Naive Bayes, Radial Basis Function (RBF) Network, and J48 decision tree. The researchers evaluated the performance of the model using 10-fold cross-validation, a robust technique to assess generalisability. The Naive Bayes algorithm emerged as the strongest performer, achieving a classification accuracy of 97.36%. This indicates that the model correctly classified nearly 97.4% of breast cancer cases as benign or malignant. The RBF Network followed closely with 96.77% accuracy, while J48 lagged at 93.41%. Further analysis using sensitivity analysis explored the impact of individual features on prediction. Interestingly, the analysis revealed the "Class" (benign or malignant) as the most significant predictor, highlighting its crucial role in model accuracy. Overall, this study demonstrates the potential of data mining for the diagnosis of breast cancer. While promising, it is important to remember limitations such as the use of a single dataset. More research with larger and more diverse datasets is needed to confirm these findings in real-world applications.

A review by [111] examined the potential of machine learning (ML) for detection and prognosis. Researchers focused on four common ML algorithms: Artificial neural networks (ANNs), support vector machines (SVMs), decision trees (DTs), and nearest

neighbours (k-NNs). They analysed data from the Wisconsin Breast Cancer Database (WBCD). The review found that these machine learning approaches achieved impressive classification accuracy in differentiating between benign and malignant breast cancer cases. The authors commend the clear presentation of the findings, including a table that summarises algorithms, techniques, and accuracy rates. Despite the high accuracy of the WBCD, the review acknowledges the need for further development of even more accurate algorithms. The authors express their intention to further explore this area using a different data set for another disease.

Banu & Ponniah [112] investigated the use of classifiers for the prediction of breast cancer [96]. Their study compared three Bayes classifiers: Tree-augmented naive Bayes (TAN), boosted augmented naive Bayes (BAN), and Bayes belief network (BBN). The three classifiers were applied to the Wisconsin Diagnostic Breast Cancer (WDBC) dataset, containing 569 cases with 32 characteristics each. The study found that the three classifiers achieved an accuracy of 90.1% before employing a technique called Gradient Boosting (GB). However, the integration of the classifiers with GB further enhanced their performance. TAN showed the most significant improvement, suggesting its effectiveness in breast cancer classification with this approach. The researchers assessed performance using metrics such as accuracy, specificity, and sensitivity. Although specific values are not mentioned here, the results support TAN as a promising method for breast cancer diagnosis, especially when combined with Gradient Boosting.

A study by [113] explored the use of machine learning to predict breast cancer. Physicians could potentially take advantage of such a system to help diagnose breast cancer based on patient clinical information (benign or malignant tumours). The researchers used a data set that included over 1700 cases from a hospital in Lagos, Nigeria. Eleven clinical characteristics, such as cell size and shape. Three supervised learning algorithms were used for classification: C4.5 decision tree, multilayer perceptron (MLP) and Naive Bayes. WEKA, a popular machine learning toolkit, was used for analysis. C4.5 emerged as the most successful algorithm, achieving the highest precision (93.9%) and the fastest processing time (0.28 seconds). This suggests that C4.5 could be a viable option to build a breast cancer prediction system. The study highlights the need for an additional feature, a user interface, to allow

patients to interact with the system. This would broaden its potential application beyond healthcare professionals.

Mirajkar and Lakshmi [114] explored using the Naive Bayes classification method of data mining to assess cancer risk. Their proposed approach was designed to predict the likelihood of developing specific cancers based on the symptoms of the patient. The study specifically investigated applying the Naive Bayes algorithm to categorise symptoms associated with breast and ovarian cancer. This suggests the potential for this method to be a tool for the early detection of these cancers.

A comprehensive review by [115] examined 45 studies exploring data mining techniques for the diagnosis, prognosis, and treatment of breast cancer. The analysis identified key strengths and weaknesses in the current research landscape, highlighting opportunities for future studies. Most studies (21) investigated the accuracy of various classification algorithms for the diagnosis of breast cancer. Twelve studies focused on differentiating benign and malignant tumours. One study explored using regression techniques for early-stage breast cancer detection within existing databases. Eleven publications aimed to develop models to predict breast cancer or survival rates. Despite promising results in accuracy, the review highlights a crucial limitation: the absence of a clinically deployable software solution. Currently, no software can autonomously diagnose breast cancer or suggest optimal treatment plans based on data mining techniques. The authors propose future research efforts to develop such a programme, leveraging data mining's potential for real-world application in breast cancer diagnosis and treatment guidance.

Mandal [116] proposes a system for breast cancer classification that prioritises using a minimal set of characteristics for high precision. This approach offers a potentially faster and more efficient method compared to using a larger set of features. The system incorporates data pre-processing steps, such as cleaning, dimensionality reduction, and transformation before classification. The study uses the Wisconsin Breast Cancer (WDBC) dataset from the UCI Machine Learning repository, containing 569 cases with 32 features. For training and testing, a 70/30 split is used. Three classification methods, Naive Bayes, Logistic Regression (LR), and Decision Tree (DT) - were evaluated on the pre-processed data. The results revealed that logistic

regression emerged as the most accurate classifier. It achieved this high accuracy using only four features, demonstrating the effectiveness of feature selection in this approach. Furthermore, logistic regression exhibited lower time complexity compared to the other two methods, suggesting its potential for faster processing.

Sumalatha & Archana [117] investigated the use of data mining techniques for early detection and prediction of breast cancer. Their study specifically compared the performance of two algorithms: the J48 decision tree and ZeroR. The analysis focused on J48; a popular decision tree algorithm commonly used in classification tasks. Although the paper mentions 699 analyses using ZeroR, it is likely that this served as a baseline for comparison (explained below). WEKA, a machine learning software platform, was used to implement both algorithms. The study followed a standard data mining workflow that obtained a data set containing breast cancer information, cleaned and preparing the data for analysis, and applied the J48 and ZeroR algorithms to categorise data points (likely benign or malignant). The high number of ZeroR analyses (699) suggests that it could have been used as a baseline. ZeroR is a simple algorithm that predicts the most frequent class in the data for all instances. Although it provides a benchmark for comparison, the performance is likely what the authors aimed to evaluate for early breast cancer prediction.

A study by [118] proposes a three-step machine learning system for automated breast cancer detection. This system offers high precision and aims to assist healthcare professionals in identifying the disease. Data Clustering (Farthest-First Clustering): This technique groups similar data points together, reducing the overall size of the data and processing time. Outlier Detection (ODA): This step identifies data points that deviate significantly from the norm, potentially representing cancerous cells. Classification (J48 Decision Tree): The pre-processed data is then classified as benign or malignant using the J48 algorithm. The researchers evaluated their system on the Wisconsin Diagnosis Breast Cancer (WDBC) and Wisconsin Breast Cancer Dataset (WBCD), achieving impressive precision: 99.9% accuracy on the WBCD dataset and 99.6% accuracy on the WDBC dataset. These results suggest that the proposed system outperforms previous methods using the same datasets. By helping to detect early and accurate breast cancer, this approach has the potential to improve patient outcomes and recovery.

Padmapriya and Velmurugan [119] investigated the use of machine learning algorithms to analyse mammograms for the detection of breast cancer. Their study went beyond just image analysis, also considering patient information such as age, lifestyle, occupation, and dietary habits. The researchers employed WEKA, a machine learning toolkit, to evaluate three common classification algorithms. J48 decision tree, AD tree, and Classification and Regression Trees (CART). These algorithms were used to analyse mammogram data along with additional patient information. Model performance was assessed using metrics such as accuracy, specificity, sensitivity, and kappa statistics. All three algorithms achieved high accuracy, J48: 98.1% accuracy, AD-Tree: 97.7% accuracy, and CART: 98.5% accuracy (highest among the three). The study highlights CART's potential effectiveness in classifying mammograms for breast cancer detection. Interestingly, the authors suggest using a combination of different classification methods to analyse the same mammogram data. This is an area that needs further exploration to potentially improve detection accuracy.

A study by [120] proposes a hybrid classification system that combines a genetic algorithm with the k-nearest neighbour (kNN) algorithm for early detection and prediction of breast cancer. This approach uses data mining techniques to help radiologists identify breast cancer in the early stage. The core objective lies in optimising the kNN algorithm, a popular machine learning method for classification. The system achieves this by selecting the most relevant features of the data and identifying the optimal value for k (the number of nearest neighbours used for the prediction). The researchers evaluated their system using the Wisconsin Breast Cancer Dataset (WBCD) obtained from the UCI Machine Learning Repository. There seems to be a discrepancy in the reported number of instances and attributes between the passage and the standard WBCD dataset (which has 569 instances and 30 features). However, the proposed system achieved a high accuracy rate of 99% compared to other classification algorithms, suggesting its potential effectiveness.

Chidambaranathan [121] proposes a hybrid machine learning approach for breast cancer prediction. This method combines k-means clustering with Extreme Learning Machine (ELM) to improve accuracy and efficiency. The system first uses clustering

of k-means to group tumours according to extracted features. This helps to identify distinct patterns within the data that may correspond to different stages or types of cancer. The study extends ELM, a machine learning algorithm, by incorporating generalised single hidden layer feedforward networks (SLFN). This modification aims to achieve faster processing times and more accurate classifications compared to standard ELM. Although the passage does not mention the specific metrics used (Jaccard distance, accuracy, specificity, and sensitivity), the author suggests that the proposed method outperforms alternatives in predicting breast cancer. This approach has the potential to be a valuable tool for early detection and diagnosis.

A study by [122] proposes a hybrid system for the prediction of breast cancer that combines three techniques, classification and regression trees (CART): A powerful machine learning algorithm for classification, feature selection: This process identifies and removes irrelevant features from the data, improving model efficiency and potentially reducing overfitting, and Bagging: This technique involves creating multiple versions of the CART model using random subsets of data, then combining their predictions for improved accuracy. The authors claim that this hybrid approach leads to higher classification accuracy in the diagnosis of breast cancer. Feature selection helps eliminate irrelevant data, while bagging enhances prediction accuracy. The researchers evaluated their system on three publicly available breast cancer datasets from the UCI Machine Learning Repository.

- Wisconsin Breast Cancer (Original): 699 instances, 11 attributes
- Wisconsin Breast Cancer (Diagnostic): 569 instances, 32 attributes
- Wisconsin Breast Cancer (unknown details): 286 instances, 10 attributes (likely a subset of another dataset)

It is important to note that all datasets have two classes (likely benign and malignant).

Padmapriya and Velmurugan [119] investigated using machine learning to classify breast cancer based on the characteristics of the mammogram image. Their study involved a data set of 250 patients with benign or malignant tumours from a cancer institute in India, which features nine key characteristics extracted from mammograms, and machine learning tools. WEKA software and three classification algorithms, J48,

CART, and AD tree. The researchers transformed their data from an Excel spreadsheet (CSV format) to the attribute relation file format (ARFF) required by WEKA for analysis. The study evaluated the performance of each algorithm using metrics such as true positive rate (TPR), false positive rate (FPR), and overall precision. CART emerged as the most successful algorithm, achieving a classification accuracy of 98.5% for the identification of breast cancer in mammograms. J48 followed closely with a precision of 98.1%. The authors suggest repeating the experiment with other classification algorithms to explore their potential effectiveness in analysing mammograms for the detection of breast cancer.

Sivakami [123] proposes a hybrid model that combines Decision Tree (DT) and Support Vector Machines (SVM) algorithms for the classification of breast cancer. This approach aims to differentiate between benign and malignant tumours. The researcher used the Wisconsin Breast Cancer Dataset (WBCD) containing 699 instances with 11 attributes. The dataset includes 458 cases of benign breast cancer and 241 cases of malignant breast cancer, with 16 instances having missing values. WEKA software was used to evaluate performance against other classification methods such as IBL, SMO, and Nave Bayes (presumably meant instead of NAVE). The results suggest that DT+SVM achieved good classification accuracy (around 91%) compared to these alternatives. Although the model achieved 91% accuracy, it is important to note the limitations, there were still 240 incorrectly classified instances (around a 9% error rate), and missing values in the data (16 cases) could potentially affect the model's performance.

A study by [124] explored the use of machine learning for the prognosis of breast cancer, to predict patient survival rates. Researchers compared the performance of three classification algorithms, naive Bayes, neural networks, and the C4.5 decision tree. The study used the Surveillance, Epidemiology and End Results (SEER) dataset, which contains 151,886 instances. Data pre-processing involved handling missing values, and attributes with missing information were excluded from nearly half of the records (extensive disease and site-specific surgery) and sixteen relevant clinical features (age, tumour size, node size, etc.) were chosen for analysis. The researchers evaluated algorithms based on accuracy, precision, and recall (metrics for classification performance). The passage does not specify the exact results but

suggests that all three algorithms achieved good accuracy in predicting survival rates. This approach has the potential to help healthcare professionals make informed decisions about early diagnosis and potentially avoid unnecessary biopsies.

A study by [125] proposes a data mining system for diagnosis and prognosis. The system employs two techniques, diagnosis using FP-growth and Association Rule Mining: This approach probably uses the FP-growth algorithm to identify patterns in patient data that can differentiate between benign and malignant tumours and prognosis using Decision Trees: The researchers built a decision tree model based on three key characteristics: age, sex, and severity of symptoms. This model aims to predict the potential for breast cancer based on these factors. The Wisconsin Breast Cancer Dataset (699 records, 9 attributes) was used to evaluate the system. Although the passage mentions high diagnostic accuracy, it lacks details on specific metrics or performance compared to other methods. Additionally, using only three features for prognosis might limit the model's accuracy. Despite limitations, the ability to achieve high diagnostic accuracy suggests that it could be a valuable tool for healthcare professionals. Early diagnosis of breast cancer can significantly improve patient outcomes.

Chaurasia and Pal [126] investigated the performance of various classification algorithms to build accurate models for the prediction of breast cancer. The study used the Wisconsin Breast Cancer Dataset from the UCI Machine Learning repository. The data set contained 699 instances with 10 characteristics, including cell size uniformity and predicted class (malignant or benign). However, 16 cases with missing values were removed, resulting in a final dataset of 683 instances. WEKA, a machine learning platform, was used to evaluate three supervised learning algorithms for breast cancer classification, Instance-Based Learner (IBK): the algorithm classifies new data points based on their similarity to existing labelled data, Balance Factor (BF) Trees: which is likely a type of decision tree algorithm that incorporates class balancing techniques, and Sequential Minimal Optimisation (SMO): this algorithm is commonly used with Support Vector Machines (SVMs) for classification. Researchers assessed the performance of each algorithm based on accuracy, Kappa statistic (a measure of agreement beyond chance) and mean absolute error (MAE, a measure of prediction error). The study found that SMO achieved the highest prediction accuracy (96.2%)

compared to the IBK and BF Tree algorithms. Furthermore, SMO had a higher Kappa statistic (0.92) and lower mean absolute error, suggesting better overall performance. The passage mentions using Chi-square, information gain, and Gain Ratio tests to analyse feature importance, but these tests are not directly related to the classification algorithms mentioned earlier. It is not clear how these tests were used in the study.

A study by [127] investigated the use of clustering algorithms for the early detection of breast cancer. Researchers employed four algorithms from the UCI machine learning repository dataset that contained 286 instances with 10 attributes related to breast cancer (age, tumour size, etc.). Farthest First (FF), this method divided the data into two groups, identifying 78% of the patients as healthy and 22% as potentially unhealthy. Expectation Maximisation (EM), this approach resulted in three clusters: 42% healthy, 25% unhealthy, and 36% unclassified. The Hierarchical Cluster Method (HCM) achieved a high success rate, with 99.65% of the patients identified as healthy and only 0.35% classified as unhealthy. However, it is unclear whether this high precision reflects a true separation of healthy and unhealthy patients or limitations in the data set itself. By K-means, this algorithm formed two clusters, with 83% of patients classified as healthy and 17% as unhealthy. The passage emphasises the performance of Farthest First (77% accuracy) and K-means (83% accuracy) for identifying potentially unhealthy patients. However, it is important to consider the limitations, the dataset size (286) might be relatively small for robust clustering analysis, and the classification of "healthy" and "unhealthy" might be binary and lack gradations for early detection.

Table 2: Table 2 shows the results.

Clustering Method	Good health	Sick	Not in class
FF	78	22	-
EM	42	25	33
HCM	98,56	1,44	-
k-means	82	12	-

Reference [128] proposes the use of data mining techniques, specifically the J48 decision tree algorithm, to develop a predictive model for the early diagnosis of breast

cancer. The study offers a comprehensive background on breast cancer, including types, risk factors, symptoms, and treatments. This context helps us to understand the importance of early detection. The core approach involves building a J48 decision tree model. As explained earlier, decision trees classify the data by following a series of questions based on specific characteristics. In this case, the characteristics likely correspond to the patient's characteristics or medical history. The model arrives at a classification (healthy or potentially cancerous) by following the relevant branches based on the patient's data. The researchers used the Wisconsin Breast Cancer dataset, which contains 699 instances with 11 attributes related to breast cancer patients. The J48 model was applied, but the passage does not mention the specific modifications made to the attributes of the data set. The J48 model achieved a low error rate (around 5.4%) in classifying breast cancer cases. This suggests that the model, potentially combined with feature selection techniques, could be a valuable tool for early diagnosis. However, it is important to compare this performance to other classification algorithms to assess its overall effectiveness.

A study by [129] aimed to create a prototype system for diagnosing breast cancer patients. The system used machine learning algorithms to analyse patient data. The researchers used the data set from the UCI Machine Learning repository, which contains information on breast cancer patients. Ten key clinical characteristics, including age, tumour size, and node size, were chosen for analysis. The passage mentions using information from "web usage mining" to identify hidden patterns in the data. However, Web usage mining typically refers to analysing user behaviour on websites. It is unclear how this technique was applied in this specific context related to medical data. More information is needed to understand this aspect. WEKA, an open-source data mining platform, was used to evaluate 37 different classification algorithms to identify and predict breast cancer. The goal was to classify the patients as healthy or sick. The study found that 13 of the 37 classifiers achieved good diagnostic accuracy (around 76% healthy and 24% ill). These successful algorithms included the Bayes network, support vector machines (SMO), logistic regression, multilayer perceptron, J48 decision tree, stochastic gradient descent (SGD), and others. The authors recommend repeating the experiment using Tanagra and Orange, which are other data mining tools. This could help identify the most effective classifier for breast cancer diagnosis in their specific dataset.

A study by [130] investigated the use of data mining techniques, specifically neural networks, to predict breast cancer. The researchers aimed to develop reliable models and explore the potential of ensemble learning for further improvement. The study used the WBCD dataset containing 286 instances with 10 attributes related to breast cancer patients (e.g., age, tumour size). Four classification approaches were evaluated using WEKA software, Lazy IBK (probably a K-Nearest-Neighbours algorithm), Tree Random Forest (an ensemble method using multiple decision trees), Lazy K Star classifier (probably a variant of K-Nearest Neighbours) and Rules NNge (potentially a rule-based classifier using neural networks). The Tree Random Forest classifier achieved the highest accuracy of 98% in classifying breast cancer cases (benign or malignant). This suggests the effectiveness of ensemble learning for this task. The authors propose further exploration of ensemble classifier analysis to potentially achieve even higher accuracy (ideally 100%). However, it is important to consider the trade-offs between accuracy and factors like model complexity and generalisability.

This review by [131] summarises research on the use of data mining approaches for diagnosis and prognosis. The study analyses 18 articles classified according to their focus, diagnosis (10 studies): These studies explored data mining classification methods for diagnosing breast cancer and prognosis (8 studies): This group focused on the use of data mining classification algorithms to predict the prognosis of breast cancer. The review highlights the potential of data mining to identify hidden patterns in data: this can aid radiologists in making informed decisions about patient care. Achieving high diagnostic accuracy: Various data mining classification algorithms demonstrate good performance in diagnosing breast cancer. Improved prognostic prediction: Artificial neural networks (ANN) appear particularly promising for predicting the prognosis of breast cancer compared to other methods. The review emphasises the value of developing optimal models. This might involve creating different models or experimenting with various techniques and algorithms to achieve the best possible performance.

Bellaachia and Guven [132] explored the use of data mining techniques to predict breast cancer survival rates. They compared the performance of three classification

algorithms, naive Bayes, backpropagated neural networks, and the C4.5 decision tree. The study used the SEER breast cancer dataset, which contains information on over 151,000 patients. However, researchers encountered missing data on attributes related to the extent of disease and site-specific surgery (present in nearly half of the records). These attributes were excluded due to missing information. Sixteen relevant clinical characteristics (age, tumour size, node size, etc.) were chosen for analysis. The researchers divided the data into two sets for analysis. Each set was further labelled according to patient survival outcomes ("survived" or "not survived"). The proportions of these labels varied slightly between the two sets. All three algorithms were evaluated based on accuracy, precision, and recall that predicted survival in both data splits. The study found that the C4.5 decision tree algorithm consistently outperformed naive Bayes and backpropagated neural networks for this task. This study demonstrates the potential of data mining techniques, particularly C4.5 decision trees, to analyse large datasets and predict breast cancer survival rates. This information can be valuable for healthcare professionals in developing personalised treatment plans and improving patient outcomes.

3.11. Conclusions

Although significant research has been done on the detection of masses and calcifications in mammograms using algorithms, there are still areas for improvement. Existing algorithms set a high bar for accuracy, which requires new developments to achieve even better performance. Extracting the most relevant features from mammogram images is crucial for an accurate classification. However, current methods might include redundant or unimportant features. A key challenge lies in identifying the optimal subset of features that best differentiate between cancerous and benign lesions. Future research should focus on optimising model performance beyond just accuracy. Other metrics such as generalisability and robustness might also be important. Developing more robust methods for feature extraction and selection is crucial. This could involve advanced techniques to identify the most informative and discriminative characteristics for an accurate classification.

Although previous approaches might have relied on various classifiers, recent advances in mammogram analysis utilise convolutional neural networks (CNN) for

automated breast cancer diagnosis. One common classifier used in earlier systems, and still sometimes used today due to its simplicity, is the K-Nearest Neighbours (KNN) algorithm. Classifies new data points based on their similarity to existing labelled data. More advanced CNN models often use other powerful classifiers such as artificial neural networks (ANN) and support vector machines (SVMs). These algorithms can learn complex patterns in the data, which could lead to better accuracy in differentiating between benign and malignant lesions.

Although microcalcifications are often easier to identify in mammograms, detecting masses remains a crucial aspect of breast cancer screening. Here are why masses pose a challenge, masses can have diverse sizes and appearances, sometimes mimicking normal breast tissue. This makes it difficult to distinguish them using image analysis techniques. Current methods might have limitations in analysing masses across different scales. Cancerous lesions can manifest on various spatial scales within the mammogram image. Masses occupy specific regions within breast tissue, but their boundaries can be blurry and require detailed analysis. The level of detail needed for accurate detection can vary depending on the sharpness of the mass boundaries. Masses can exhibit variations in width, length, and density in different cancers. This complexity suggests the need to analyse masses across multiple scales to capture their full characteristics. One of the limitations suggests that existing methods might not fully address the challenges of scale dependence and the need for multiscale analysis for accurate mass characterisation.

Although mass detection is crucial, accurate breast cancer diagnosis using mammograms also relies on identifying other abnormalities. This refers to disturbances in normal tissue patterns within the breast, which can be a sign of underlying cancer and occurs when there are noticeable differences in the appearance of breast tissue between the left and right sides. These abnormalities pose challenges for automated detection systems, and research must focus more on addressing them. Compared to mass detection, architectural distortion, and bilateral asymmetry haven't received as much research attention. Developing effective methods to identify these abnormalities could significantly improve the overall performance of breast cancer detection systems.

CHAPTER 4: METHODOLOGY

4.1. Introduction

This chapter delves into the process of building an image classification model. We explored how the image data were collected and meticulously organised. We then examined how this data was used to train various deep learning models, including convolutional neural networks (CNNs) with specific architectures such as VGG19, InceptionV3, and MobilenetV2. Finally, we introduced evaluation metrics that were used to assess the effectiveness of these models in accurately classifying images. The chapter also delves into the technical details of the deep learning architectures and approaches that underpin our comprehensive breast cancer CAD system. Inspired by cutting-edge computer vision techniques, we propose novel models for each stage of the system. Trained in mammogram datasets, these models extract robust features to identify and pinpoint breast abnormalities within the mammogram, accurately delineating breast tumours and masking irrelevant background tissue, and classifying tumours to aid in the final diagnosis (malignant vs. benign). Using deep learning models, the integrated CAD system provides a more comprehensive and potentially more accurate approach to breast cancer diagnosis.

4.2. Research Design

To investigate the cause-and-effect relationship between [independent variable] and [dependent variable], this study uses an experimental research methodology. Age, family history, genetics, race, ethnicity, obesity, alcohol use, and lack of exercise are only a few examples of the many independent variables. The conclusion of an experimental research study supports or rejects the existence of a link between a particular attribute of an item and the variable under investigation. To determine whether breast cancer can be available or not on mammography images, the model will be developed to detect breast cancer on mammography images.

The flow of the process in Figure 17, which consists of three main processes, is used to achieve the objective of detecting breast cancer. The problem's domain is determined in the first stage so that it can be better understood through an examination

of various forms of literature. This thesis follows a three-phase approach to achieve its broad and specific objectives.

Phase 1: Data preparation lays the groundwork. Data were obtained from Mankweng Hospital in Limpopo, South Africa. The dataset was preprocessed, including cleaning, labelling, and dividing it into training, validation, and testing sets. These steps ensure that the data are suitable for the application of classification and regression methods in the next phase. Phase 2: Model development focusses on building the research prototype. The prepared data were used to develop and train the model.
Phase 3: Model Evaluation and Comparison involves evaluating the effectiveness of the developed model. The model was trained on the training data and its performance was continuously evaluated using the validation data. This iterative process helps to identify and optimise the model's performance. Once trained, the model performance was assessed on unseen data from the testing set. This provides a more realistic measure of its generalisability. The final evaluation involves comparing the model's performance to other existing models on the same task. This helps to determine the relative effectiveness of the proposed approach.

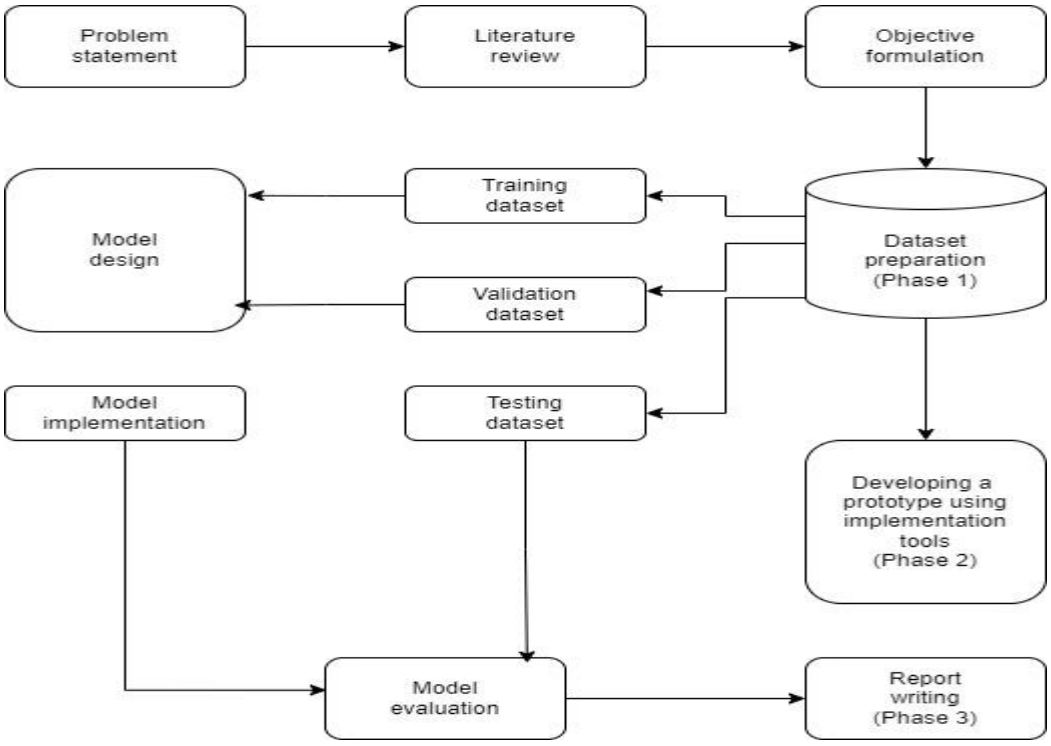


Figure 16: Research flow model.

4.3. Data Preparation

The foundation of any neural network application lies in the acquisition of suitable training data. In this dissertation, the focus is on breast cancer mammography imaging data to train the model. However, securing a publicly available database containing hundreds of mammograms from South African women presents a significant challenge. While publicly available datasets are ideal for research, obtaining one specifically suited to our needs presents challenges. A large database containing hundreds of breast cancer mammograms from South African women might be difficult to find due to the following; stringent regulations around patient privacy can limit the public availability of medical data. Therefore, with the help of radiologists and other medical professionals, breast cancer mammography images were obtained from Mankweng hospitals in Limpopo province. Images of sick and healthy people were collected from Mankweng Hospital.

Although collaborating with Mankweng Hospital to access their data offers a potential solution, using their database is likely to require meeting specific criteria. Research involving patient data typically requires ethics approval before using the data. The approval process ensures that the research adheres to ethical guidelines and protects the patient's privacy. The Appendix presumably includes the approval of the relevant ethics for this study. The images are organised into folders in the dataset and the image labels are contained in a text file. The text file content is shown in Figure 18 after it has been transformed into a pandas data frame in Python.

refnum	bg	ab_class	severity	x	y	radius
mdb001	G	CIRC	B	535.0	425.0	197.0
mdb002	G	CIRC	B	522.0	280.0	69.0
mdb003	D	NORM	A	NaN	NaN	NaN
mdb004	D	NORM	A	NaN	NaN	NaN
mdb005	F	CIRC	B	477.0	133.0	30.0

Figure 17: The first five rows of the MIAS dataset labelled as pandas data frame

The data includes several key columns relevant to breast cancer analysis:

- Column 1: It likely represents an identifier for each data point (e.g., image ID).
- Column 4: Anomaly severity: This column indicates the severity of the anomaly detected in the mammogram image. It uses letter codes:
  B: Benign
  M: Malignant
  A: Standard (no anomaly detected)
- Columns 5 & 6: Anomaly Centre: These columns specify the x and y coordinates of the centre of the abnormality within the image.
- Column 7: Anomaly Radius: This column probably represents the estimated radius of a circle that encompasses the detected abnormality.

Digital X-ray mammography, the scanning technique used to acquire these mammogram images, often involves breast compression. This compression can introduce artefacts like deformable breast regions and degrade image quality. To mitigate these issues and improve the analysis, some pre-processing steps are necessary. In this work, we focus on histogram equalisation, which will be explained in detail in Section 5.3.5 as a method to enhance compressed regions. By equalising pixel intensity distribution, histogram equalisation can aid in differentiating suspicious tissue from normal tissue.

Our CAD-based model requires two inputs: the mammogram images themselves and the corresponding ground-truth data. These ground-truth data specify the regions of interest (ROI) surrounding breast lesions. We used existing expert annotations to extract lesion coordinates (x, y, width, height) and their classification (mass or calcification). To ensure consistency for neural network analysis, all mammogram images underwent a resizing process. Bicubic interpolation was employed to resize each image to a standard dimension of 448 x 448 pixels. This choice ensures that the image size is divisible by 32, a requirement for the Darknet backbone architecture used in VGG19, Inception, and MobilenetV2. Additionally, it ensures that the images fit within the GPU memory limitations.

Deep-learning models thrive on vast amounts of labelled data to achieve robust generalisation. However, medical data sets often face limitations: they can be small and exhibit imbalanced class distributions, hindering model performance [133]. A common challenge in deep learning, especially in medical imaging applications, is the limited availability of training data. This dissertation addresses this by using two powerful techniques data Augmentation (discussed in Section 4.3.3) and Transfer Learning (discussed in section 3.9). The data augmentation technique involves artificially creating new variations of existing data to expand the training data set. This helps the model generalise better and avoid overfitting on the limited data. Transfer Learning, this approach utilises a pre-trained model on a vast dataset for a related task. The model's weights are then fine-tuned on the specific breast cancer mammogram data for improved performance.

Data augmentation artificially expands a dataset in a controlled manner. In this work, we specifically focus on the detection task and enhance our mammogram data set sixfold. This augmentation involves two steps. We rotated the original images by 0 °,, 90 °, 180 ° and 270 °. This dissertation employs Contrast Limited Adaptive Histogram Equalisation (CLAHE) [135, 138, 140] as a data augmentation technique to increase the size of the training dataset and improve model generalisability. CLAHE enhances image contrast by adjusting local regions, effectively introducing controlled variations that mimic real-world image variations.

We specifically explore two CLAHE parameter settings. Tile grid size (4x4) and contrast threshold of 40, this configuration applies CLAHE in smaller image blocks with moderate contrast adjustment. Tile grid size (8x8) and contrast threshold of 30. This setting uses larger image blocks for processing and a slightly lower contrast adjustment. These variations in tile size and contrast level create a wider range of training images that better reflect potential real-world mammogram variations. This, in turn, helps the model learn robust features and generalise more effectively to unseen data during testing.

The initial step in training a deep learning model involves setting the values of its trainable parameters, which include weights and biases [134]. These parameters essentially determine how the model transforms the input data into meaningful

outputs. There are two main approaches to initialising these parameters, random initialisation and transfer learning. Random Initialisation, this method assigns random values to the weights and biases. Although effective for smaller models, it can be inefficient for complex models such as the one used in this dissertation. Transfer learning, this powerful technique leverages a pre-trained model on a vast dataset related to the target task. The pre-trained model's weights are then fine-tuned on the specific breast cancer mammogram data for our task. The transfer learning approach offers some benefits, including reduced training time and improved performance. By starting with pre-learnt features, the model converges faster on a good solution. Transfer learning can improve the final accuracy of the model, especially when dealing with limited training data. Transfer learning's effectiveness hinges on utilising a pre-trained model with features relevant to the target task. In our case, we employed weights pre-trained on the Microsoft COCO dataset. This data set contains a vast collection of everyday objects, providing a solid foundation for image recognition tasks, although it was not specifically designed for medical images. The specific pre-trained model architecture utilised Darknet, which originated from the VGG-16 model. Darknet is known for its efficiency and strong image recognition capabilities, which makes it a suitable choice for our transfer learning approach.

4.3.1 Image Data Pre-Processing and Analysis

Although convolutional neural networks (CNNs, discussed in Section 2.8) can inherently extract features from raw images, some preprocessing steps can still be beneficial for improved model performance and efficiency. In this study, we applied some pre-processing techniques to mammogram data from the health sector. Techniques such as filtering were employed to remove unwanted noise from the mammogram images, potentially improving the model's clarity and accuracy of feature extraction. Image normalisation is a step that may involve techniques such as adjusting the pixel intensity values to a specific range. Normalisation helps the model to converge faster during training and reduces the computational load. Since the original images might have varying sizes, resizing them to a standard dimension (e.g., 448 x 448 pixels) ensures consistency for the CNN and reduces processing time during training, especially on a system with limited hardware resources.

4.3.2 Data Partitioning

Dividing the data set into training, validation and testing sets was crucial to developing effective deep learning models. From the 18494 images collected from Mankweng Hospital, we allocated a portion for training, validation, and testing. A common split ratio is 80% for training, 10% for validation, and 10% for testing, as supported by the literature [135]. This ensures that the model was trained on a substantial dataset while reserving a portion for unbiased evaluation. During training, data was fed iteratively through the model using forward propagation. During each iteration, the model predictions were compared to the actual labels in the training data. The errors are then propagated backward through the network using backpropagation. This helps the model adjust its internal weights and biases to minimise prediction errors. The validation set plays a vital role. The model's performance was assessed on the validation data throughout the training process. This helps identify and prevent overfitting, a situation where the model performs well on the training data, but poorly on unseen data.

By monitoring the validation performance, we can adjust hyperparameters (learning rate, optimiser, etc.) to optimise the model's generalisation ability. Once training is complete, the final evaluation is performed using unseen testing data. This provides a more realistic measure of the model's generalisability and performance on real-world data. The text mentions avoiding overfitting by using an equal number of images from each class for training and validation. This was a good strategy to address the class imbalance, where one class has significantly more data than the others. If this was the case in the dataset, we were going to consider additional techniques like oversampling the minority class or undersampling the majority class to achieve a more balanced representation.

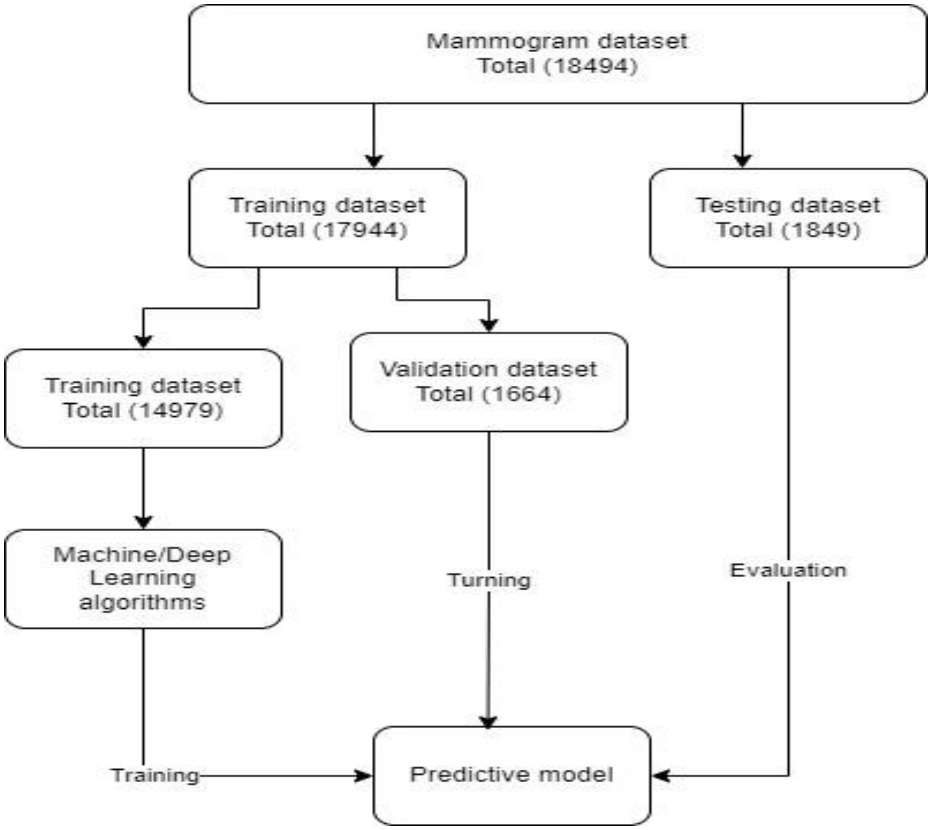


Figure 18: Partitioning of data.

4.3.3 Data Augmentation

Deep learning models often require vast amounts of training data for optimal performance. In image classification tasks, obtaining a large and diverse data set can be challenging. To address this limitation, we employed a technique called data augmentation. Data augmentation involves artificially creating new variations of existing training images. This effectively expands the training dataset without requiring additional data collection. Newly generated images introduce controlled variations that mimic real-world scenarios, improving the model's ability to generalise to unseen data [136]. Even with vast amounts of data collection, data augmentation techniques can further expand the effective size of the dataset. This reduces overfitting and helps the network extract more nuanced properties from the data [137]. To enrich our data set and avoid overfitting, we will employ a diverse set of data augmentation techniques on the original images during the training or pre-processing stage. To enhance the training data, we will use Keras libraries for real-time data augmentation during the training process. This means that each image fed into the network will be a version of the original image that has been randomly transformed using techniques such as

rotation, flipping, or scaling.

4.4. Implementation Techniques and Tools

The proposed model was meticulously designed and implemented using a robust development environment. Specific details about the tools employed (e.g., deep learning frameworks and programming languages) and their contributions to the development process will be provided in the following section.

4.4.1 Software Tools

To select the optimal software tool for breast cancer detection using CNN algorithms on mammographic images, we evaluated the available software packages and their libraries. This assessment revealed that some tools, such as Python, are general-purpose programming languages suitable for various machine learning and deep learning tasks. In contrast, Support Vector Machines (SVM) are machine learning algorithms, not software tools, and are typically implemented within broader software packages.

We identified suitable software tools and libraries for our CNN-based breast cancer detection CAD. The primary factor was to ensure that the tools chosen were seamless with CNN algorithms. We prioritise tools with abundant learning resources, including free video tutorials, to facilitate a smooth learning curve. Considering potential resource limitations, we favour tools that function effectively on CPU-only systems. Our evaluation led us to select Python as the programming language due to its versatility and extensive deep-learning libraries. Within the Anaconda environment, we used the TensorFlow and Keras libraries to build our CNN model. These tools not only excel within the Python environment but also address all our selection criteria.

TensorFlow, an open-source deep-learning library developed by Google, has become a leading choice due to its efficiency and comprehensive functionality [138]. Beyond traditional desktop environments (Windows, MacOS, Linux), TensorFlow seamlessly integrates with cloud services and even mobile platforms (iOS, Android). This versatility empowers developers to deploy their models across various computing

landscapes. TensorFlow streamlines the entire deep learning workflow, including data pre-processing, model architecture design, training, and evaluation. A core concept in TensorFlow is the tensor, a multidimensional array that represents data throughout computations. Additionally, TensorFlow leverages a visual graph structure to depict the training process, aiding in model understanding and optimisation. The library caters to diverse hardware environments by offering dedicated distributions for both CPUs and GPUs.

Keras, a user-friendly neural network API built on top of libraries such as TensorFlow or Theano, simplifies the development of deep learning for users [139]. This high-level interface allows rapid prototyping and experimentation thanks to its support for various architectures like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and even combinations of these. The inclusion of pre-trained models like VGG19 and Inception within Keras further streamlines the process, offering readily available building blocks for deep learning projects.

CUDA: a powerful parallel computing platform developed by NVIDIA plays a crucial role in accelerating deep learning workloads. Its programming model simplifies harnessing the processing power of GPUs, leading to significant performance gains compared to traditional CPU-based approaches.

CUDNN: Deep neural network primitives are available in the NVIDIA CUDA Deep Neural Network (cuDNN) package, which uses GPU acceleration. It offers highly specialised implementations of routines that typically appear in deep neural network applications.

For the implementation of this study, the computational resources were strategically divided. Data sets were stored on hard drives. To accelerate model training, we leverage the processing power of a dedicated Graphics Processing Unit (GPU). The trained models were then evaluated on the Central Processing Unit (CPU). Finally, sufficient random-access memory (RAM) was crucial to facilitate efficient training through collaboration between the CPU and the GPU. The detailed specifications of the laptop used for this experiment are presented in Figure 20.

Computer type: HP EliteBook X360 1030 G2
Operating system: Microsoft Windows 10 Pro
Processor: Intel® CoreTM i5-7300U CPU @ 2.60GHz, 2712Mhz
Installed RAM: 8GB.
Storage device: 1TB

Figure 19: specifications for laptop

4.5. Evaluation Methods

Assessing a model's effectiveness is paramount in determining its usefulness. In classification and detection tasks, we employ various evaluation metrics to gauge how well the model distinguishes between different classes. These metrics include precision, precision, recall, and the F1 score. To calculate these metrics, we rely on the classification metrics derived from the confusion matrix (presented in Table 3 below). The confusion matrix details the distribution of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for our classifications. These values serve as the basis for calculating all evaluation metrics.

Table 3 *Confusion matrix*

Prediction value	Real value		
		Negative	Positive
	Negative	TN	FN
	Positive	FP	TP

The confusion matrix is a valuable tool for visualising the performance of a classification model. It summarises the model's predictions by categorising them into four key groups.

True Positives (TP): These represent the instances where the model correctly identifies a positive image. In other words, the image truly belongs to the positive class and the model classified it correctly.

True Negatives (TN): These represent the instances where the model correctly identifies a negative image. The image belongs to the negative class, and the model's prediction reflects that.

False Positives (FP): These represent errors where the model incorrectly classifies a negative image as positive. The image does not belong to the positive class, but the model mistakenly identified it as such.

False Negatives (FN): These represent errors where the model incorrectly classifies a positive image as negative. The image truly belongs to the positive class, but the model missed it. By analysing these classifications within the confusion matrix, we gain valuable insight into the strengths and weaknesses of the model for our specific classification task.

Accuracy is a fundamental metric that is used to assess a model's overall classification performance. It essentially answers the question: "How often does the model correctly predict the class of an image?" In our context of breast cancer detection, this translates to correctly classifying mammograms as healthy or infected. Accuracy is calculated using the following formula [140]:

Precision provides information on the frequency of accuracy of a positive-value prediction. Using an image as an example, predict how often the prediction is accurate in identifying the disease. The formula shown below is used to calculate it.

$$Precision: P = \frac{tp}{tp + fp} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots . \ldots 4.1$$

Recall that this term, which also refers to sensitivity, indicates the classifier's sensitivity in identifying affirmative cases. This is how it is calculated [140].

$$Recall: R = \frac{tp}{tp + fn} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots 4.2$$

The F1 score is a valuable metric that strikes a balance between two key classification metrics: precision and recall. It takes the harmonic mean of these metrics, providing a more comprehensive picture of a model's performance beyond just accuracy. A perfect F1 score of 1 indicates that both precision and recall are ideal (1). In contrast, a score of 0 occurs when precision or recall is zero [140].

$$F - score: F1 = F = 2 * \frac{P * R}{P + R} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots 4.3$$

4.6. The proposed architecture

The system design process began with the crucial step of preparing high-quality training and validation data. Three specialists labelled the data throughout the mammography image collection process, after which the raw data were processed. The mammogram images were ready for use by applying cropping, data normalisation, and categorical disease labelling. The area of interest was visible in the preprocessed images. A feature extraction process was used to create a model, and the significant characteristics of each image were obtained from the CNN stacked layers. The classifier identifies benign and malignant breast mammography patterns in response to the characteristics of the breast model recovered during training. During training, performance was evaluated using the validation data set. After training, model performance was assessed using the validation dataset. Several pre-trained models will be tested before building the model from the ground up. The best model was chosen through the model evaluation. Ultimately, unseen and unlabelled data was used to assess the best performing model throughout the testing phase. Ultimately, the model predicts the class of an image (malignant or benign) based on the probabilities learnt during training.

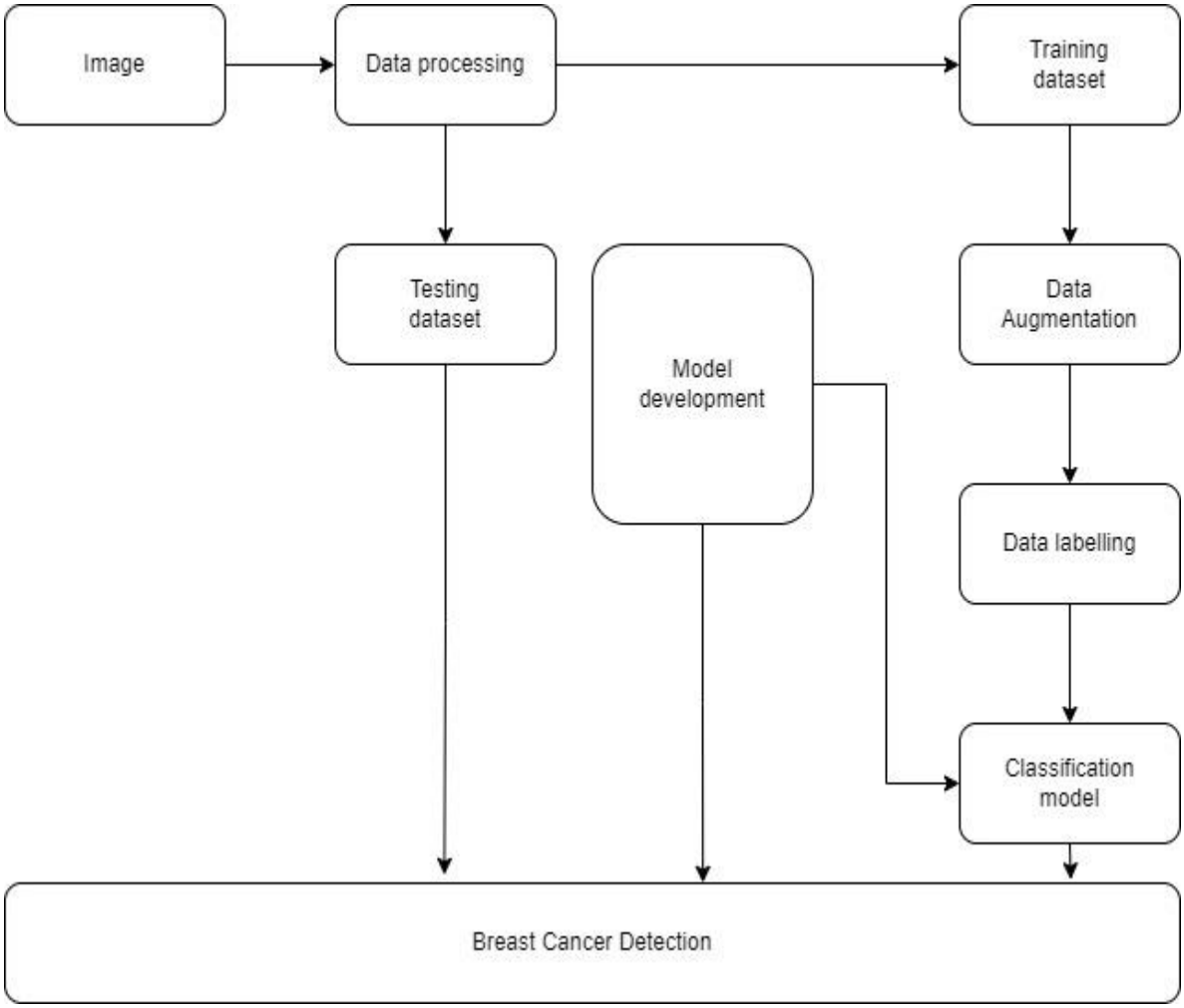


Figure 20: The suggested design for detecting breast cancer

4.7. Image data preparation

4.7.1 Data Preprocessing

The source code for a Python resize script will be used to resize the collected images to 448*448 pixels. Using label image tools, the scaled images will be labelled or classified into the corresponding categories of cancer stages. The file will be in the format of an a.xml file when the image is annotated. After using .xml_to_.csv to convert the .xml file to a .csv file, the object detector will read the ft. Note. For feeding the model, train.csv and test.csv will be transformed into train.tf record and test.tf record, respectively.

4.7.2 Data Augmentation

Submitting annotated data to Rob Flow Workspace automatically augments the data. This process involved rotating the original images 90 degrees, applying blurring effects, and flipping them horizontally and/or vertically. Convolutional neural networks (CNNs) are powerful image recognition models, but their complex nature (millions of parameters) requires a lot of training data to function well. If not trained with enough data, they can overfit, meaning that they learn the specifics of the training data too well and perform poorly on unseen data. Data augmentation (a technique explained in Section 4.3.3 [27]) helps address this by artificially creating more training data from existing images, improving the model's ability to generalise to new images.

4.7.3 Data Splitting

To address overfitting, the data were divided using a train-test split technique with a validation set. Training data (80%) were used to create the model, and the validation set (10%) was used to fine-tune the model's hyperparameters to improve generalisability. The remaining 10% of the data served as the test set for the final performance assessment.

4.7.4 The Proposed Model's Training Components

No deep learning model that can address all issues with machine learning is preinstalled. Several machine learning issues construct a model by the computational issue. The CNN model's proposal is impacted by its tiny dataset and low processing power. The two classes in a data set contain a healthy breast and a diseased breast. The scratch model employs a convolutional neural network architecture with three convolutional layers (Conv2d, Conv2d_1, Conv2d_2) using the ReLU activation function for non-linearity. These are followed by three pooling layers and two fully connected layers with ReLU activation. Finally, the output layer with three neurones uses a sigmoid activation function to predict class probabilities. Instead of relying on existing models, this work presents a novel image categorisation model built from scratch. The model was rigorously optimised through hyperparameter exploration to address various challenges.

Input layer: RGB (Red, Green, and Blue) The CNN model accepts input images with

a size of 448x448x3 belonging to five distinct categories: category1, category2, category3, category4, and category5. Without performing any computations, the input layer simply forwards the data to the first convolution layer. Because of this, the input layer will have zero parameters and no learnable characteristics.

Convolutional layers: The scratch model uses three convolutional layers: Conv2d, Conv2d_1, and Conv2d_2. The image was an input that was passed from the input layer with dimensions of (none, 417, 417, 3). Afterward, Con2d creates output (no, 417, 417, 16) using the input layer's output (no, 417, 417, 3).

```
model =tf.keras.models.sequential([
       tf.keras.layer.conv2D(16,(3,3),activation='relu', input_shape=(417,417,3))
       tf.keras.layers.maxpooling2D(2,2),
       tf.keras.layers.conv2D(32,(3,3),activation='relu'),
       tf.keras.layers.maxpooling2D(2,2),
       tf.keras.layers.conv2D(64,(3,3),activation='relu'),
       tf.keras.layers.maxpooling2D(2,2),
       tf.keras.layers.Flatten(),
       tf.keras.layers.Dense(512,activation='relu'),
       tf.keras.layers.Dense(1,activation='sigmoid')
])
model.compile(optimizer=RMSprop(lr=0.001, loss='binary-crossentropy', metrics=['acc']))
```

```
history= model.fit(training_set,
                   epochs=30,
                   verbose=1,
                   validation_data=
                   validation_set)
```

Figure 21: CNN prototype.

The Conv2d layer by default uses 16 filters with a kernel size of 3x3, a stride of 1, and a dilation rate of 1 (as shown in Figure 22). This configuration produced an output with a specific size, which can be calculated as follows:
(Image Height - KernelHeight+Stride,ImageWidth-KernelWidth+Stride, Filter).

Given a 448x448 input image, this Conv2d layer with a 3x3 kernel size, stride of 1, and dilation rate of 1 produces a new feature map with dimensions (415, 415, 16). We can calculate the number of parameters in this layer as follows:

*(KernelHeight*KernelWidth*InputChanal*OutputChanal+OutputChanal if bias is used)*

However, this formula only calculates the weights for the filters, not the bias terms. A

Conv2d layer also has a bias term for each filter, resulting in additional parameters. In this case, the layer will have a total of 16 (filters) * (3 x 3 x 3 - number of weights per filter) + 16 (biases) = 1,792 parameters. Considering the input image size and configuration, the correct output shape for this Conv2d layer is (415, 415, 16). This output will then be fed into the max-pooling layer.

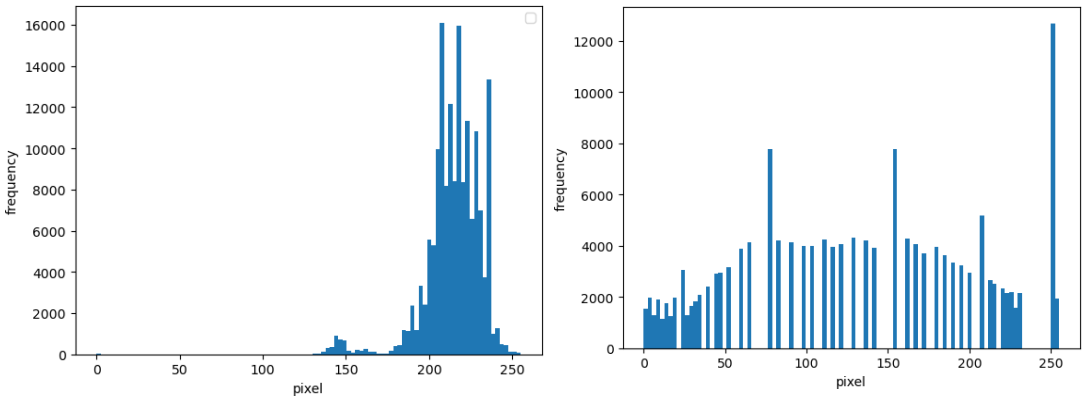
The proposed model incorporates three max-pooling layers strategically placed after specific convolutional layers. The first maximum-pooling layer follows the first convolutional layer. It uses a 3x3 filter with a stride of 1 to reduce the spatial dimensions of the feature maps generated by the first convolution while preserving the most prominent features. The second maximum pooling layer is positioned after the second convolutional layer. It employs a 2x2 filter with a stride of 1 to further down-sample the feature maps, capturing significant spatial features. The third maximum-pooling layer comes after the fifth convolutional layer. This layer uses a 2x2 filter with a stride of 2 for a more aggressive downsampling of the feature maps. Unlike the previous layers, it has no learnable parameters and focusses solely on reducing the data size along the spatial dimensions.

Following the convolutional layers, the model uses three fully connected (FC) layers, including the output layer. The first two FC layers contain 64 neurones, while the final output layer has only one neurone. The first FC layer takes the flattened output from the fifth convolutional layer. Performs dimensionality reduction by converting the 3D feature maps into a 1D vector. During training, the number of neurones (predefined as 64) and the weights are optimised to compute class scores. The second FC layer takes the output from the first FC layer and processes it further. The 64 neurones are fully connected to the neurones in the first FC layer, like standard neural networks. The output layer contains a single neuron responsible for the final prediction.

The final output layer uses a softmax activation function. This function is crucial for multiclass classification problems like this, as it converts the output from the last layer (typically real numbers) into a probability distribution between the six categories. This allows the model to predict the class with the highest probability.

4.7.5 Histogram Equalisation

The initial step to achieve contrast-limited adaptive histogram equalisation was to obtain the picture histogram equalisation. Histogram equalisation can be used alongside sharpening techniques to improve both contrast and edge definition in an image, potentially revealing hidden features. To do this, as shown in Figure 23, the image frequency histogram is spread over the range of possible grey scale levels, which are usually between 0 and 255. This causes the histogram to be less clustered and more evenly distributed throughout the range.



23a 23b

Figure 22: Histogram of an image before (a) and (b) after histogram equalization.

Mathematically, the equalisation of histograms is accomplished as follows:

Let m be an image that has been given and is shown as an integer pixel intensity matrix of size $x\ c$.

The images in this dataset use pixel intensities ranging from 0 to L-1, where L represents the maximum possible value (typically 256 for greyscale images).

Using image m as an example, let p be the normalised histogram.

$$P_n = \frac{number\ of\ pixel\ with\ intensity\ n}{total\ number\ of\ pixel} \quad n = 0,1,2,3,4, \ldots\ldots, L-1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.4)$$

Subsequently, the histogram equalised image g will be provided as

$$g_{ij} = floor\left((L-1)\sum_{n=0}^{ij} P_n\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.5)$$

The floor function reduces to the closest whole number.

Therefore, the function was used to change the pixels k of the image m to create the transformed image g.

$$T(k) = floor\left((L-1)\sum_{n=0}^{k} P_n\right) \quad (4.6)$$

The following transformation is motivated by considering the intensities of pixels as continuous random variables. We denote these intensities as X and Y, where both X and Y range from 0 to L-1 (inclusive). Additionally, Y is defined by

$$Y = T(X) = (L-1)\int_0^y P_x(X)d_x \quad (4.7)$$

$P_x(X)$ represents $m's$ probability density function m 's. $\int_0^y P_x(X)d_x$ represents a cumulative distributive function (CDF).

Under the assumption that T(X) is differentiable and invertible (for simplicity), we can demonstrate that Y, defined by Y = T(X), is uniformly distributed on the interval [0, L-1]. For example, $P_y(Y) = \frac{1}{L-1}$.

$$Y = T(X) \quad (4.8)$$

We were applying the formula for changes in variables then:

$$P_y(Y) = P_x(X)\left|\frac{dx}{dy}\right| \quad (4.9)$$

Given that $Y = T(X) = (L-1)\int_0^y P_x(X)d_x$ then the derivative of this equation was taken.

$$\frac{dy}{dx} = \frac{d(T(X))}{dy} = \frac{d(L-1)\int_0^y P_x(X)d_x}{dy} \quad (4.10)$$

$$\frac{dx}{dy} = (L-1)P_x(X) \quad (4.11)$$

When we substitute equation (9) from (7), we get this:

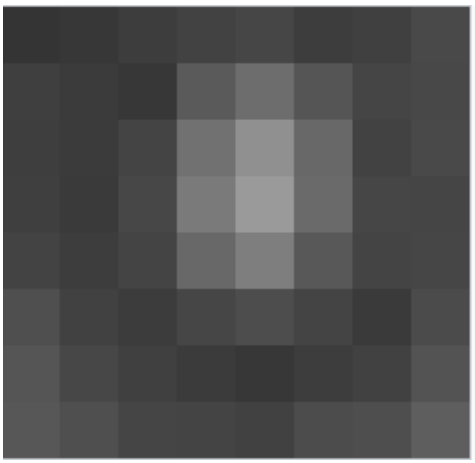
$$P_y(Y) = P_x(X)\left|\frac{1}{(L-1)P_x(X)}\right| \quad (4.12)$$

We can easily remove the absolute sign because L>1, and $P_x(X)$ cancels out to leave us with:

$$P_y(Y) = \frac{1}{(L-1)} \quad (4.13)$$

To demonstrate how the histogram was used in this project, look at the straightforward image in Figure 24. Pixel values are shown in Figure 24a, while the greyscale image is shown in Figure 24b.

52	55	61	59	79	61	76	61
62	59	55	104	94	85	59	71
63	65	66	113	144	104	63	72
64	70	70	126	154	109	71	69
67	73	68	106	122	88	68	68
68	79	60	70	77	66	58	75
69	85	64	58	55	61	65	83
70	87	69	68	65	73	78	90



24a 24b

Figure 23: grey scale image for histogram equalization illustration.

Getting the image histogram table is the first step. This is demonstrated in the following table 4.

Table 4 Histogram table of the grey image

value	count	Value	count	value	count	value	count	value	count
51	2	63	3	71	2	84	1	112	2
54	3	64	2	72	1	86	2	121	1
57	1	65	1	74	2	87	1	125	3
58	2	66	2	75	1	89	2	143	2
59	3	67	4	76	2	93	2	153	1
60	4	68	5	77	3	103	1		
61	2	69	3	78	1	105	2		

62	1	70	1	82	2	108	1	

The function below determines the image's CDF for each pixel, as seen below.

$$T(k) = \left(\sum_{n=0}^{k} P_n * m\right) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (4.14)$$

Where M is the total number of pixels, the total number of pixels is 64 from 8x8.

In this study, the computation of the cumulative distribution frequency for the first five pixels in ascending order is considered sufficient as a proof of concept, as shown in Table 5. To make the table smaller, only the first five pixels are visible.

Table 5 CDF for the first 5-pixel

Pixel intensity	CDF
52	1
55	4
58	6
59	9
60	10

After obtaining the CDF, the formula is used to generate the histogram equalisation $g(k)$, which scales the pixels between 0 and 255, inclusively.

$$g(k) = \ floor\left(\frac{CDF(k) - CDFmin}{(M - CDFmin)}\right) X(L-1) \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (4.15)$$

4.7.5.1. The OpenCV package used to implement histogram equalisation

The histogram equalisation was implemented using the straightforward flow chart shown in Figure 25.

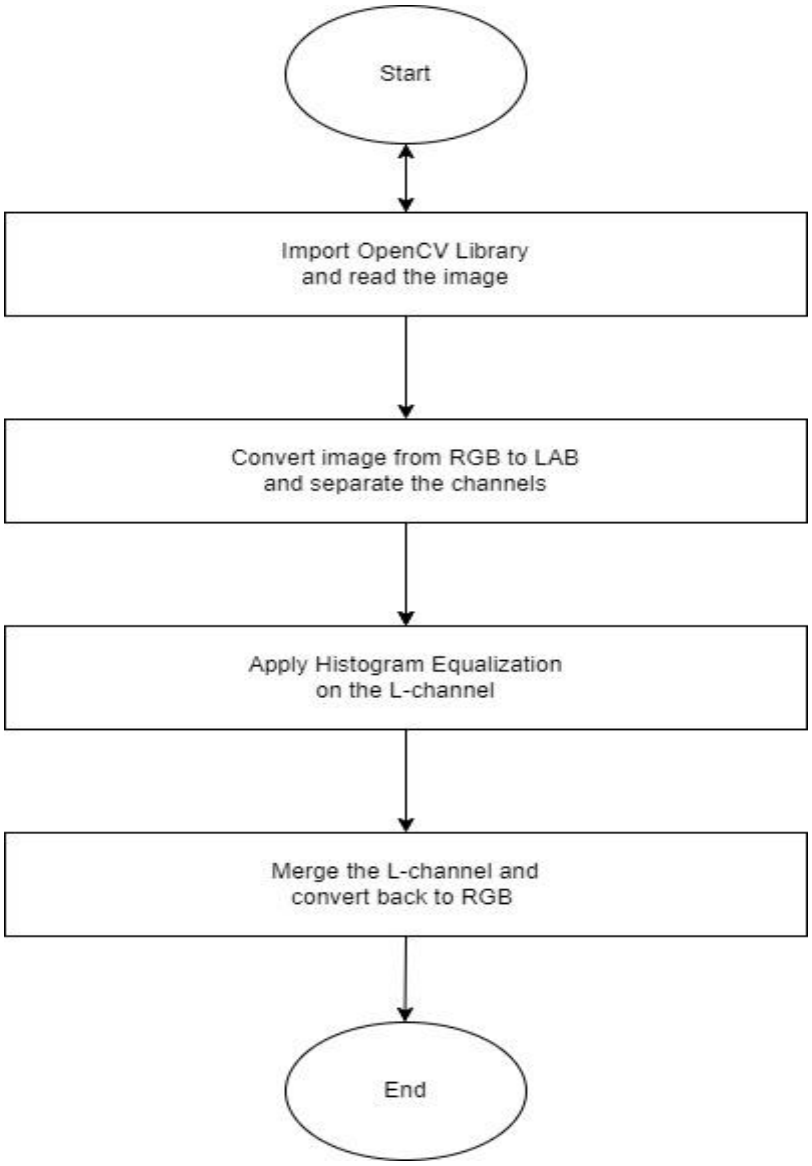
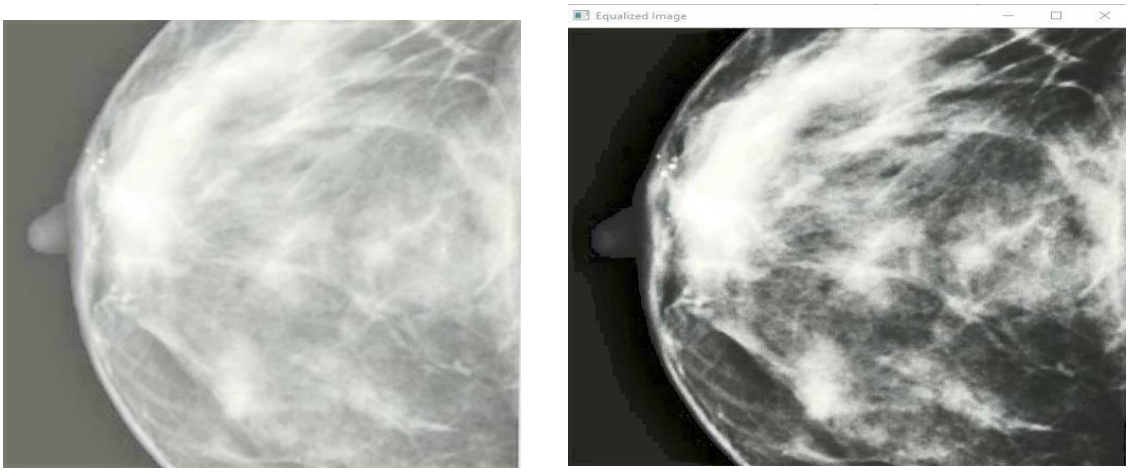


Figure 24: Flow chart of the Histogram Equalisation Algorithm using the OpenCV Library

Importing the openCV library was the initial step, as illustrated in Figure 25. Subsequently, the image was read from the location where it was stored. The red, blue, and green (RBG) format comprises most of the images; hence it is not the ideal format for histogram equalisation because it needs to be done on all three channels. Rather, the image was first transformed to Lightening, A channel, and B channel (LAB) format, where A and B store the image's colour information, while L serves as the image's Lightning channel. The L channel was then subjected to Histogram

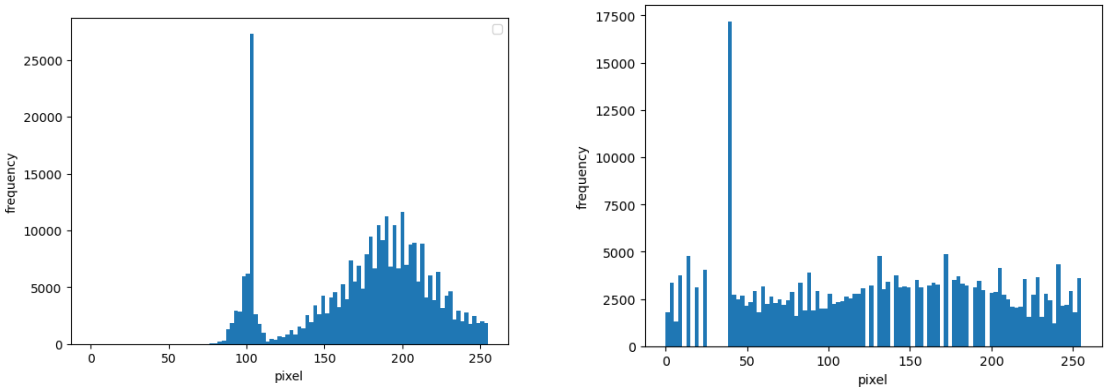
Equalisation; once equalised, the L channel was combined back into the A and B channels. After that, the image was converted back to RBG.

A sample of mammography images was compared with histograms and the result is displayed below.



26a 26b

Figure 25: Histogram Equalization Performed on a sample of breast image mammogram.



27a 27b

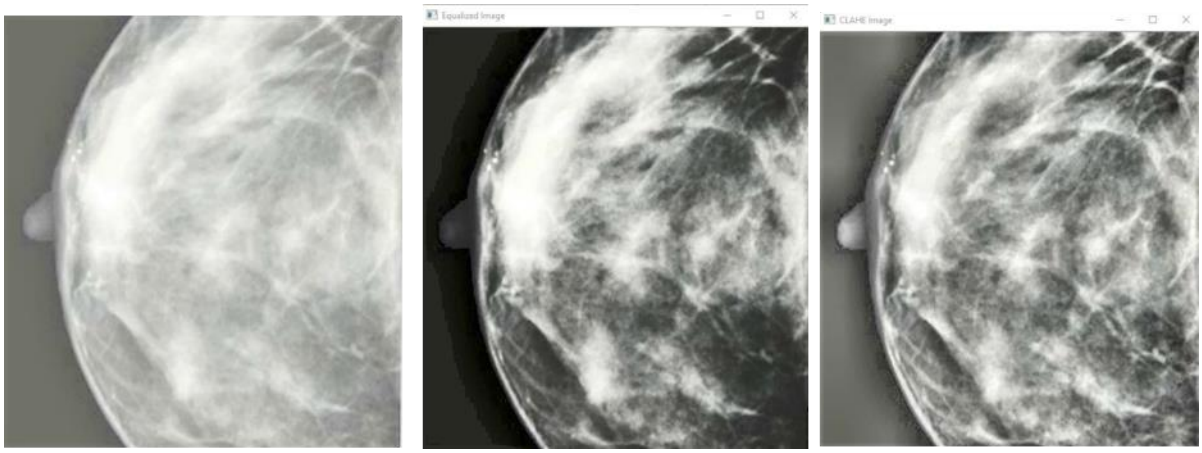
Figure 26: histogram of the images of Figure 23 respectively.

The histogram for Figure 26a is represented by Figure 27a, and the comparable histogram for Figure 26b is represented by Figure 27b. Since the image has been brightened and has lighter shades of grey, as can be seen in Figure 26a, the low contrast image's histogram is centred on the high value of 255. Figure 26b further illustrates how the histogram has now been uniformly dispersed throughout the greyscale.

4.7.5.2. Contrast-Limited Adaptive Histogram Equalisation (CLAHE)

When the supplied image contains a large area of low-intensity background, the histogram equalisation approach does not work as expected. An image's noise is amplified effectively due to the severe washing-out that occurs in such a scenario. Contrast-Limited Adaptive Histogram Equalisation, an advancement over Adaptive Histogram Equalisation (AHE), is the method used to get around this. The image is separated into discrete parts to apply the histogram equalisation procedure to each region separately in adaptive histogram equalisation. Although AHE is effective in many cases, contrast-limited adaptive histogram equalisation (CLAHE) is a variant that addresses the issue of noise amplification in homogeneous regions. Thus, by restricting amplification, contrast-limited adaptive histogram equalisation resolves each of these issues. CLAHE (Contrast-Limited Adaptive Histogram Equalisation) is an image-processing technique specifically designed to enhance the contrast of images with low contrast. It works by dividing the image into small regions (tiles). For each tile, CLAHE calculates a histogram of the intensities of the pixels. This histogram reveals the distribution of the pixel values within that region. To improve contrast locally, CLAHE manipulates the histogram.

The slope of the transformation function applied by CLAHE on each tile is directly proportional to the value of the histogram at that value of the pixels. In simpler terms, areas with more pixels of a certain intensity (higher histogram value) will experience greater enhancement, in contrast. CLAHE also considers the slope of the cumulative distribution function (CDF) in the neighbourhood surrounding each pixel. The CDF shows the probability that a pixel has an intensity value less than or equal to a specific value. By incorporating the neighbourhood's CDF slope, CLAHE ensures that contrast enhancement is limited locally, preventing the creation of excessive noise in the image. The image below (Figure 27) exemplifies the result of applying CLAHE to an image.



28a 28b 28c

Figure 27: Figure showing the original image (a), histogram equalised (b), and the contrast-limited adaptive histogram equalised image (c).

Several pre-processing steps were applied to the image to optimise it for deep learning algorithms. This improves the model's ability to learn relevant features and potentially leads to better performance. Zero-padding this technique adds additional pixels with a zero-intensity value around the borders of the image. Scikit-Learn provides functionalities to achieve this. Organ windowing is the process that involves extracting a specific region of interest (ROI) from the image, typically focussing on the organ of interest. OpenCV offers tools for image cropping and manipulation to perform windowing. Normalisation is the step that scales the intensity values of the image to a specific range (e.g., 0-1 or -1 to 1). Scikit-Learn provides various scaling methods for normalisation. Registration is the process that aligns the organ of interest across different images, ensuring consistency for the training data. OpenCV offers image-registration algorithms for this purpose. Interpolation is the technique that fills in missing values or resamples the image to a desired size. SciPy provides functionalities for interpolation tasks.

4.7.6 Hyper-Parameter Tuning

Hyperparameters are crucial settings in a deep learning model that define its learning process and performance. These parameters are defined before training and are not specific to a particular deep learning algorithm. Unfortunately, there is no single "best" set of hyperparameters for every problem. To find the optimal hyperparameters for this specific model, we perform a process called hyperparameter tuning. This involves

systematically trying out different combinations of hyperparameter values and evaluating the model's performance on a validation set. The following sections detail the hyperparameters chosen for this model and the reasoning behind their selection.

Optimisation algorithms play a critical role in the training of neural networks. They guide the model towards better performance by updating the weights based on the calculated error. Gradient descent is a fundamental optimisation technique widely used in deep learning. It iteratively adjusts the weights in the direction that minimises the loss function (error). Modern deep learning frameworks such as the one used in this dissertation often integrate Keras and various gradient descent variations. This project specifically explores the Adam optimiser, a popular choice that builds upon gradient descent. Adam addresses the issue of fixed learning rates by dynamically adjusting the learning rate (0.0001 and 0.00002) for each parameter based on the calculated gradients and their historical averages. This adaptation allows Adam to converge faster and achieve better performance compared to standard gradient descent.

The learning rate is a crucial hyperparameter in neural network training. Control the step size used to update the model's weights during the learning process. These weights are essentially the parameters that the model learns from the data. A lower learning rate results in smaller weight updates, leading to slower but potentially more stable convergence. In contrast, a higher learning rate leads to faster, but potentially more erratic, weight updates. Finding the optimal learning rate is an ongoing challenge that often requires experimentation. Through trial and error, we can determine the learning rate that allows the model to effectively learn from the data without overshooting the minimum loss or getting stuck in local minima (shallow improvements that hinder progress). The chosen learning rate could change based on the specific problem and the model architecture.

The selection of both the activation function and the loss function depends on the type of machine learning problem being addressed. The activation function is the binary classification problem; the final fully connected layer utilises a sigmoid activation function. This function outputs values between 0 and 1, making it suitable for predicting binary class probabilities (e.g., an image belonging to category A or category B). The

loss function is a binary classification task (only two possible outcomes), and the model uses the Binary Cross-Entropy (BCE) loss function. BCE is particularly well suited to measure the difference between predicted probabilities (between 0 and 1) and actual labels (0 or 1) in such scenarios.

Although there are alternative loss functions, such as categorical cross-entropy (CCE) and mean squared error (MSE), they are better suited for tasks. The mean squared error (MSE) is typically used for regression problems where the model predicts continuous values. Categorical Cross-Entropy (CCE) is a loss function typically used in multiclass classification problems where there are more than two possible classes. Measure the difference between the predicted class probabilities and the actual labels. Therefore, BCE is the most appropriate choice for this binary classification task.

The proposed model uses different activation functions throughout its layers. The hidden layers in this model are likely built with the ReLU activation function. ReLU's non-linearity allows the model to effectively learn intricate data patterns, potentially leading to improved performance on complex tasks. Since we are dealing with a multiclass classification problem, the final layer utilises the SoftMax activation function. SoftMax ensures that the output values sum up to 1 and represent probabilities for each class, making it ideal for predicting the most likely class for a given input. We trained the models for a variable number of epochs, monitoring their performance on the validation set to determine the optimal stopping point. The accuracy validation graphs were then compared to see how many epochs each model, including the proposed model and pretrained models, requires to achieve similar validation accuracy. Batch size is a crucial hyperparameter in neural network training. Determine the number of samples presented to the network and processed at once during each training iteration. The choice of batch size significantly affects the efficiency of the training. Although common options include 32, 64, and 128, the optimal value depends on several factors, the specific problem being addressed, the size of the dataset, and the available computational resources. Large Batch Sizes can lead to faster training per epoch (one pass through the entire dataset) as fewer parameter updates are needed. However, they can also require more memory and might result in slower convergence (reaching optimal performance) due to less frequent updates with potentially outdated gradients. Small batch sizes require less memory and may lead

to faster convergence due to more frequent updates. However, they can also result in slower training per epoch due to the increased number of updates needed.

Finding the best batch size often involves experimentation. To optimise training efficiency, we experimented with various batch sizes (for example, 32, 64, 128). The validation set was used to assess the trade-off between training speed (fewer updates with larger batches) and convergence (more frequent updates with smaller batches). This allowed us to select the batch size that achieved the best balance for this specific model and dataset.

4.8. Conclusions

This chapter gives the structure of how the proposed CAD system was developed. The chapter outlines how the data were prepared using the data set acquired from Mankweng Hospital in Limpopo province. The pre-processing of the image data must be done first, followed by the way the data were partitioned, data augmentation, and the tools were used during the implementation. The mathematical background on accuracy, precision, recall, and F1 score was also outlined. This section also describes the components and structure of the deep learning model used for breast cancer detection. It explained the reasoning behind the chosen architecture and its suitability for the task. This chapter delves into the critical processes involved in preparing image data for training a machine learning model. We will explore techniques for cleaning and normalising the data to optimize model performance. A key concept is data augmentation, which artificially expands the dataset by generating variations of existing images. This process enhances the robustness by exposing it to a wider range of visual scenarios, improving its ability to generalise to unseen data, and discussing the importance of dividing the data into distinct sets for training, validation, and testing. Each set plays a crucial role in the model development process. This seemingly simple step forms the foundation for reliable model evaluation and ultimately, a robust and generalisable model. The architecture of the proposed model described the components and structure of the proposed deep learning model for the detection of breast cancer. Hyperparameter tuning explored the optimisation process of finding the best settings for the model learning process. The learning rate controls the speed with which the model updates its weights during training. The epoch represents a complete

passage through the entire training data set. The batch size defines the number of images presented to the network at the same time during training. The chapter also mentioned the use of OpenCV, a Python library, for implementing image processing techniques like histogram equalisation and contrast-limited adaptive histogram equalisation (CLAHE) during data preprocessing.

CHAPTER 5: EXPERIMENTATION AND DISCUSSION OF RESULTS

5.1. Introduction

This chapter evaluates the accuracy, sensitivity, and specificity of the proposed methods and models used in the breast cancer CAD system. As the framework is designed to provide a comprehensive diagnosis, each step plays a crucial role and contributes to the overall outcome. Therefore, we will evaluate each step individually and collectively to assess their effectiveness and ensure optimal final performance. The chapter outlines various experimental scenarios to illustrate the application of the proposed methods and models. In addition, it details the specific hyperparameters chosen for optimal performance in this study.

5.2. Experimental Configuration

The study employs four diverse experimental scenarios, designed to comprehensively evaluate the system's ability to identify and categorise various breast cancer cases. The first scenario establishes a baseline by training a CNN model from scratch on the study's breast cancer dataset, utilising default hyperparameters. The other tests are carried out using current models for transfer learning. To explore the impact of model selection, the second experiment uses the VGG19 model, while the InceptionV3 and MobilenetV2 models are used in the third and fourth scenarios, respectively.

5.3. CNN features

An RBG image was fed into the model with filter size 3*3 and stride 2 (shown by the input 448*448*3). After feeding the image into the convolutional layer, the output was max aggregated at 415*415*16. After being added to the second layer (206*206*16), the 208 * 208 * 16 were maximally pooled using the second maximum pool layer. The 103 * 103 * 32 output was input into the last convolutional layer, which flattened the 50 * 50 * 64 output to display the 6-class output after max-pooling the 101 * 101 * 64 output. The entire connected layer is used by the Softmax activation algorithm to select and display one of the classes.

5.4. Augmentation Parameters

Figure 29 illustrates the data enhancement techniques used in this experiment. By artificially expanding the training data set to 18,494 images, data augmentation allows the model to learn from a wider variety of features, potentially improving its generalisability.

```
rescale=1./255,
rotation_range=40,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=true,
fill_mode='nearest'
```

Figure 28: Specific Augmentation Methods during training.

The shape was arbitrarily slanted using the shear transform. For this work, the image is stretched at an angle of 0.2 along one fixed axis. Zoom was used to enlarge the image details; the zoom value used was 0.2. The rotation taken into consideration here represents up to 40 degrees of random rotation of the dataset's images. Horizontal flipping is the process of flipping images in a diagonal direction. Ranges for both width and height changes were expressed as a percentage of the overall width or height of 0.2, determining how arbitrarily the image was shifted to the left or right. Newly formed pixels that emerged after a change in rotation or width/height were filled in using the nearest-fill-mode method.

Table 6 *hyper-parameter settings for pre-trained models.*

Measurement	Size of Batch	Time	Loss Function	Activation Mechanism	Algorithm for Optimisation	Rate of Learning
Value	32	90	BCE	ReLu	Adam	0.001
	64	30	CCE	Softmax		0.002
	128	65		Sigmoid		

5.5. Experimental Result

5.5.1 Classification of breast cancer using a custom-built model

The experiment began by building a convolutional neural network (CNN) model from scratch. This model's performance was then compared against pre-trained models to evaluate the effectiveness of leveraging pre-trained knowledge for the specific task. Subsequently, the most effective model was chosen for further use. Although it performs less well with unknown data than with pretrained models like VGG19, InceptionV3, and MobilenetV2, it has good training and validation accuracy. However, building a model from scratch with a limited dataset is often discouraged. Since the model has fewer examples to learn from, it may only capture a small set of image features. This can lead to poor performance on unseen data, where the model encounters variations on which it was not trained. Consequently, testing with unseen data yields low performance, but the scratch-built model performs well on the training and validation sets. To thoroughly evaluate the models' capabilities, the study employed a diverse set of experimental scenarios.

Scenario 1: Comparing the dimensions of the images, which are 224x224x3 and 300x300x3.
Scenario two: Verification using 0.00002 and 0.0001 learning rates.
Scenario three: For the final layer, replace the activation function with Softmax or ReLU depending on the task.
Scenario four: Modifying Adam and RMSprop's optimisation algorithm with the specified learning rate.

```
def __init__(self, params, lr=1e-2, alpha=0.9, eps=1e-10, weight_decay=0, momentum=0., centered=False,
             decoupled_decay=False, lr_in_momentum=True):
    if not 0.0 <= lr:
        raise ValueError("Invalid learning rate: {}".format(lr))
    if not 0.0 <= eps:
        raise ValueError("Invalid epsilon value: {}".format(eps))
    if not 0.0 <= momentum:
        raise ValueError("Invalid momentum value: {}".format(momentum))
    if not 0.0 <= weight_decay:
        raise ValueError("Invalid weight_decay value: {}".format(weight_decay))
    if not 0.0 <= alpha:
        raise ValueError("Invalid alpha value: {}".format(alpha))

    defaults = dict(lr=lr, momentum=momentum, alpha=alpha, eps=eps, centered=centered, weight_decay=weight_decay,
                    decoupled_decay=decoupled_decay, lr_in_momentum=lr_in_momentum)
    super(RMSpropTF, self).__init__(params, defaults)

def __setstate__(self, state):
    super(RMSpropTF, self).__setstate__(state)
    for group in self.param_groups:
        group.setdefault('momentum', 0)
        group.setdefault('centered', False)

def step(self, closure=None):
    """Performs a single optimization step.
```

Figure 29: Code snippet of RMSprop.

Among the four scenarios explored, the best performing model was a scratch-built CNN built from scratch with an input image size of 300x300x3 pixels. This model used a learning rate of 0.001, ReLU activation in the final layer, and the RMSprop optimiser. It achieved an accuracy of 88%.

Table 7 below demonstrates the four scenarios that were carried out. Scenario one demonstrates the dimensions of the images. The learning rates of 0.00002 and 0.001 were carried out using Scenario 2. The final layer of the CNN illustrates the activation functions utilised (Softmax and Relu) in Scenario three. The modified Adam and RMSprop optimisation algorithm with a specified learning rate was applied to scenario 4. Table 7 summarises the model's performance metrics, including accuracy, recall, and precision.

Table 7 *An overview of the experimental findings.*

	Accuracy	Recall	Precision
Scenario one (224*224)	86.1	81.3	82.1
Scenario two (0.001)	87.4	83.2	84.3
Scenario three (Softmax)	88.2	86.1	86.5
Scenario four (Adam)	88.6	86.8	87.2

Figure 31 below demonstrates the F1 score and loss of the model built from scratch. The figure shows that when the Epoch increases, the F1 score increases, and the loss decreases. When the Epoch reaches 80 the F1 score and the loss become stable.

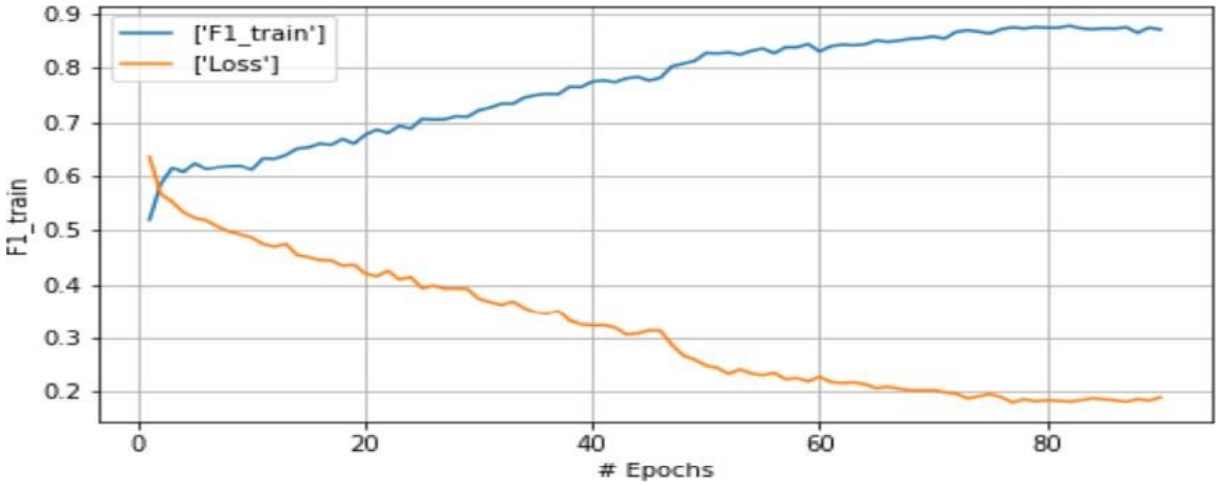


Figure 30: Diagram for the Scratch model.

Table 8 *train accuracy versus train scratch model loss*

Metrics	Mean Accuracy			Mean Loss		
	Training	Validation	Testing	Training	Validation	Testing
Value (%)	88	89	86	5.01	5.6	14

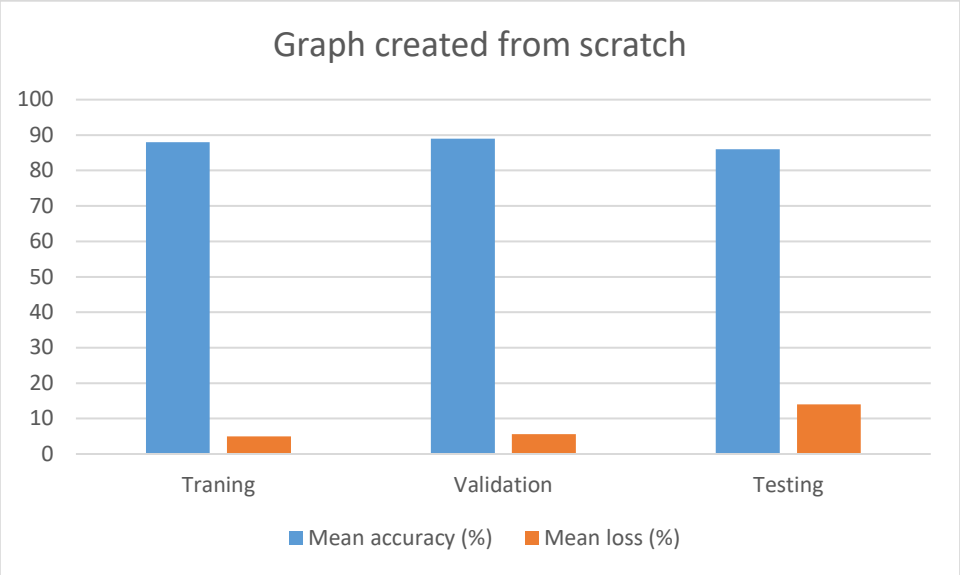


Figure 31: Diagram Model for the Scratch Process

5.5.2 Using a pre-trained algorithm to classify breast cancer diseases.

This study leverages three popular pre-trained convolutional neural networks (CNNs): VGG19, InceptionV3, and MobilenetV2. These models were originally trained on the massive ImageNet dataset, and their architectures are fine-tuned for the specific task of breast cancer classification in this work. The pre-trained models that were chosen were chosen based on their ability to identify breast cancer on devices with minimal processing power, like tablets and smartphones. The InceptionV3 model is used for its intricate features, the VGG19 model is selected for its ease of use, and MobilenetV2 is selected for its low processing overhead [141]. To evaluate the impact of model complexity on classification accuracy, we conducted experiments using a relatively simple and more intricate model. Importantly, all experiments used the same dataset and hyperparameter configuration to ensure a fair comparison.

5.5.2.1. Result of the experiment with VGG19

Three of the 19 levels of the VGG19 network design are completely connected, and the remaining 16 layers are convolution layers. Because more layers enable the network to analyse more precise features, deep neural networks become more accurate over time. Convolution operations are performed by the fully deep-trainable convolution layers, which are linked to max-pooling and dropout layers. Max-pooling

is employed as a downsampling technique to reduce the dimensionality of the output from the convolutional layer. The convolution layers are essentially 3x3 network layers. A single lesion is used to train the suggested network, and all lesions are checked to reduce the number of false positives. The convolutional layer acts as a feature extractor. It applies a filter, also called a kernel, that slides across the input image pixel-by-pixel. At each location, the element-wise product (multiplication) between the filter and the local image patch is calculated and summed. This operation captures spatial relationships between pixels, generating a new feature map that highlights specific characteristics of the image. The network can have multiple convolutional layers, each using different filters to extract progressively more complex features. There is a mathematical formulation for the convolution layer's output as follows:

$$M_i^{(x)} = B_i^{(x)} + \sum_{k=0}^{B_i^{(x-1)}} K_{i,j}^{(x-1)} * m_i^{(x)} \quad \ldots \ldots \ldots (5.1)$$

Where $M_i^{(x)}$ is the output function for the feature maps and $B_i^{(x)}$ denotes the matrix bias, and this $K_{i,j}^{(x-1)}$ denotes the filter size or convolution kernel.

The convolution procedure is followed by the application of an activation function to produce a nonlinear transfer function, which is given as:

$$N_i^{(x)} = N\left(M_i^{(x)}\right) \quad \ldots \ldots \ldots (5.2)$$

Here, the activation function is represented by $N_i^{(x)}$. While the ReLU (Rectified Linear Unit) activation function is a popular choice for CNNs due to its computational efficiency, it is important to consider other options. ReLU (rectified linear unit) allows values greater than zero to pass through unchanged while setting negative values to zero. This makes it computationally efficient and helps alleviate the problem of vanishing gradients, a phenomenon that can hinder training in deep neural networks. The sigmoid function outputs a value between 0 and 1, often interpreted as a probability. However, it can suffer from vanishing gradients in deep networks. The hyperbolic tangent function (tanh) outputs values between -1 and 1, offering a zero-centre alternative to the sigmoid function. However, like a sigmoid, it can also experience vanishing gradients. Mathematically, the ReLU function is expressed as follows.

$$N_i^{(x)} = Max\left(0, N_i^{(x)}\right) \quad \ldots \ldots \ldots (5.3)$$

The ReLU (Rectified Linear Unit) activation function is often used after a convolutional layer to introduce nonlinearity into the network. It achieves this by setting all negative values in the input to zero and focussing the model on learning from positive features. This also contributes to the computational efficiency of ReLU. Following the convolutional layer, a max-pooling layer is typically employed for dimensionality reduction. This downsampling technique reduces the size of the feature maps generated by the convolution, making the network more manageable and potentially reducing overfitting. The three types of sampling process that are commonly used are maximum pooling, average pooling, and sum pooling. In the suggested work, max pooling is employed, where the output image pixel is the maximum value of each block. Three fully linked networks are part of the VGG19 general design, and they are utilised to categorise the features that were acquired from the prior max-pooling and convolution layers.

Classification with softMax activation after the feature extraction stages (convolutional layers, pooling layers, etc.) in a feedforward neural network, the final step often involves classification. Here, the SoftMax activation function plays a crucial role. The SoftMax function takes a vector of real numbers as input and transforms them into a probability distribution over a set of discrete classes. Each element of the output vector represents the probability that the input belongs to a specific class. The sum of all elements in the output vector is always equal to one. This ensures that the function produces a valid probability distribution. All elements in the output vector are nonnegative values between 0 and 1. This makes it easier to interpret the results as probabilities.

$$N_{j}^{(x)} = f\left(x_{i}^{(x)}\right) \quad (5.4)$$

Here, f stands for the nonlinear transfer function, while $N_{j}^{(x)}$ represents the weight functions. In the suggested study, a straightforward and effective support vector machine was used to categorise the state of the tumour rather than a neural network. Compared to other classifiers in machine learning methods, support vector machines perform better in classification. The minimum features were processed by the support vector machine to achieve maximum accuracy, as the deep network identified the important characteristics. Finding the ideal hyperplane to divide the feature space was

done using the supervised learning algorithm. With the help of the high-dimensional hyperplane, the training dataset was split into several classes. To transfer data into a new vector space when it is not linearly separable, a kernel support vector machine was employed. Regarding the provided training dataset $D(x_1y_1), (x_2y_2), (x_3y_3), \ldots \ldots .., (x_ny_n)$.

The linear classification decision function is represented as

$$y_i = ((w\Delta x_i) + b) \ldots \ldots \ldots (5.5)$$

Data were transformed into a high-dimensional space using $x_i \in R^2 \rightarrow h$. The membership $y_i \in \pm 1$ stands for the label assigned to every activity specified within the data set.

The following criteria were used to determine the generic hyperplane.

$$w.x_i + b = 0 \ldots \ldots \ldots (5.6)$$

If the margins are defined, the hyperplanes are reorganised and provided as

$$y_i(w.x_i) + b) \geq 1 \ldots \ldots \ldots (5.7)$$

The element that must be minimised to create the ideal hyperplane was supplied by the functions as

$$1/2||w||^2 \ldots \ldots \ldots (5.8)$$

```
import torch
import torch.nn as nn
import torch.nn.functional as F

from timm.data import IMAGENET_DEFAULT_STD, IMAGENET_DEFAULT_MEAN, IMAGENET_INCEPTION_MEAN, IMAGENET_INCEPTION_STD
from .helpers import build_model_with_cfg
from .registry import register_model
from .layers import trunc_normal_, create_classifier, Linear


def _cfg(url='', **kwargs):
    return {
        'url': url,
        'num_classes': 1000, 'input_size': (3, 299, 299), 'pool_size': (8, 8),
        'crop_pct': 0.875, 'interpolation': 'bicubic',
        'mean': IMAGENET_INCEPTION_MEAN, 'std': IMAGENET_INCEPTION_STD,
        'first_conv': 'Conv2d_1a_3x3.conv', 'classifier': 'fc',
        **kwargs
    }
```

Figure 32: Codesnippet used to import VGG19.

The VGG19 experiment takes advantage of the baseline architecture and hyperparameters established in previous research. To fine-tune the model for breast cancer classification, the Adam optimiser is used during the initial training phase (first 15 epochs) while utilising the pre-trained weights from ImageNet on the convolutional kernels. Following the initial 15 epochs using the Adam optimiser with pre-trained ImageNet weights, the model underwent extensive fine-tuning for an additional 90 epochs. This fine-tuning involved training all layers with stochastic gradient descent (SGD) optimiser and experimenting with different dropout rates (0.001, 0.002, and 0.2) to prevent overfitting.

Table 9 *Configuration for the VGG19 experiment*

	Learning rate	Accuracy	Recall	Precision
Net weight of the image with Adam optimiser	0.001	78	76	75
	0.002	77	75.2	74.6
SGD optimiser for 15 epochs	0.001	74	72.4	72
	0.002	73	71	79.6

A training accuracy of 77.8% and a loss of 7% in VGG19 are shown for the customised F1 score in Figure 34. Table 10 presents the performance evaluation of the VGG19 model in the breast cancer diagnosis dataset, as measured by key assessment metrics.

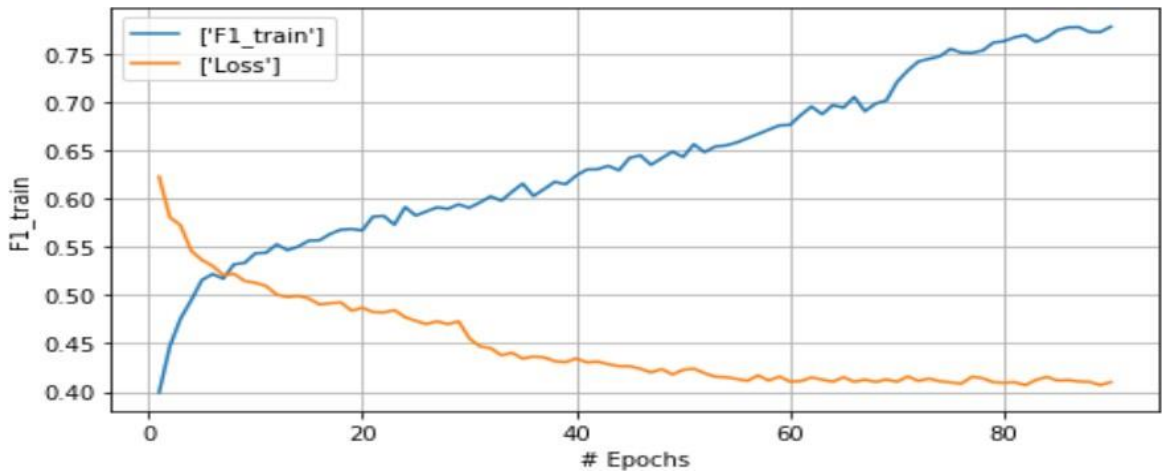


Figure 33: Loss of VGG19 and training accuracy.

According to the graph, the model's loss decreases and its learning capacity grows with each passing epoch. The model's performance was also altered by Adam optimisers.

Table 10 *Accuracy of the VGG19 model and Loss Outcome*

Metrics	Mean Accuracy			Mean Loss		
	Training	Validation	Testing	Training	Validation	Testing
Value (%)	77.8	78.4	76	7.0	7.70	22.5

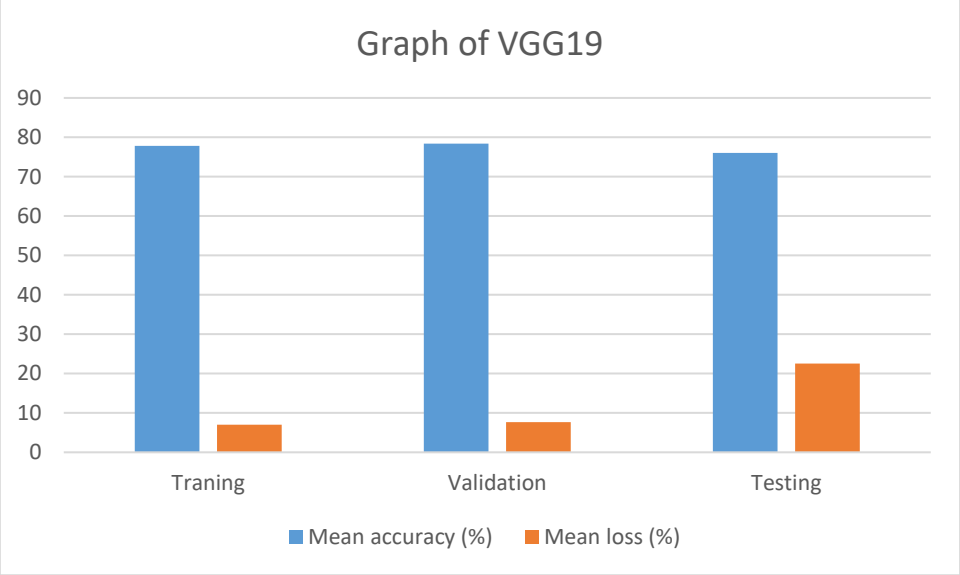


Figure 34: Graph VGG19: Mean Loss and Accuracy

5.5.2.2. Result of the experiment on InceptionV3

The Inception Recurrent Residual Convolutional Neural Network (IRRCNN) [142] employs recurrent convolutional layers (RCL) to process information in discrete time steps. Within an IRRCNN block, consider unit (i, j) of an input sample in the feature map K^{th} of the RCL. This unit receives input not only from its local neighbours in the same feature map but also from the corresponding unit (i, j) in the previous layer of the IRRCNN block. Furthermore, we can assume that the network $O_{ijk}^{l}(t)$ was at time step t. Based on this understanding, we can express the result as follows in Equation 5.9.

$$O_{ijk}^{l}(t) = (w_{k}^{f})^{T} * x_{l}^{f(i,j)}(w_{k}^{r})^{T} * x_{l}^{f(i,j)}(t-1)b_{k} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.9)$$

The input of the inputs of the typical convolution layers' inputs was $x_{l}^{f(i,j)}(t)$, whereas the l^{th} RCL's input was $x_{l}^{f(i,j)}(t-1)$. The conventional convolutional layer weight (w_{k}^{f}) and the K^{th} feature map's RCL weight (w_{k}^{r}) were respectively represented by these numbers and the bias was indicated by b_k.

$$y = f\left(O_{ijk}^{l}(t)\right) = \max\left(0, O_{ijk}^{l}(t)\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.10)$$

The Rectified Linear Unit (ReLU) is a widely used activation function denoted by max (0, x) in Equation (5.10). It essentially sets all negative values in the input to zero, allowing only positive values to pass through unchanged. ReLU's computational efficiency and ability to address the vanishing gradient problem make it a popular choice in neural networks. While Rectified Linear Unit) is a popular and effective default activation function, this experiment delved into the performance of ELU (Exponential Linear Unit) as well. ELU offers a key advantage over ReLU, it introduces a small, nonzero slope for negative input values. This slope allows information in the negative range to flow through the network more easily, potentially improving the network's ability to learn complex patterns in the data.

Within the average pooling layer, the inception units process the input using filters of varying sizes. The resulting outputs, denoted by $y_{1*1}(x)$, $y_{3*3}(x)$, and $y_{1*1}^{p}(x)$ respectively, correspond to these different filter sizes. We denote the final output of the IRCNN unit as $f(x_l, w_l)$ which was expressed in equation (6.11).

$$f(x_l, w_l) = y_{1*1}(x) \odot y(x) \odot y_{1*1}^{p}(x) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.11)$$

The feature map axis or the channel (denoted by ⊙) was used to concatenate the output of the IRCNN unit with the original input of the block. This essentially combines the processed information with the original data. The concept of adding the block's input to the IRCNN unit's output, as shown in equation (5.12), is the core of the residual connection in the IRRCNN block.

$$x_{l+1} = x_l + f(x_l, w_l) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.12)$$

In equation (5.12), w_l represents the kernel weights of the convolutional layers within the l^{th} IRRCNN block, x_l represents the input samples to the lth IRRCNN block,

typically arranged as a feature map, $f(x_l, w_l)$ represents the outputs from the l^{th} layer within the lth IRRCNN unit, and x_{l+1} represents the data that provides the initial activation for the immediate next transition block.

A normalised exponential function layer, often known as a softmax layer, was the final layer in the architecture. Equation (5.13) defines the mathematical relationship behind the softmax function, a crucial component in neural networks for multiclass classification tasks. The softmax function takes an input sample (x), a weight vector (W) and considers (K) unique linear functions to compute a probability distribution over i^{th} classes.

$$P(y = i|x) = \frac{e^{x^T w_i}}{\sum_{k=1}^{K} e^{x^T w_k}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.13)$$

The hyperparameter setting directly affected how well the model performed during neural network training. With InceptionV3, a training period of more than 90 epochs, a dropout rate of 0.2, and a learning rate of 10^{-4} were used. Table 11 presents the InceptionV3 experiment on the results of the breast cancer classification.

```
import torch
import torch.nn as nn
import torch.nn.functional as F

from timm.data import IMAGENET_DEFAULT_STD, IMAGENET_DEFAULT_MEAN, IMAGENET_INCEPTION_MEAN, IMAGENET_INCEPTION_STD
from .helpers import build_model_with_cfg
from .registry import register_model
from .layers import trunc_normal_, create_classifier, Linear


def _cfg(url='', **kwargs):
    return {
        'url': url,
        'num_classes': 1000, 'input_size': (3, 299, 299), 'pool_size': (8, 8),
        'crop_pct': 0.875, 'interpolation': 'bicubic',
        'mean': IMAGENET_INCEPTION_MEAN, 'std': IMAGENET_INCEPTION_STD,
        'first_conv': 'Conv2d_1a_3x3.conv', 'classifier': 'fc',
        **kwargs
    }
```

Figure 35: Code snippet used in the import of InceptionV3.

Table 11 *Experimental result for InceptionV3*

	Rate of Learning	Accuracy	Recall	Precision
Image's net weight using the Adam optimizer	0.001	88.5	87.3	86.3
	0.002	87.7	85.7	85.7
15 epochs of SGD optimisation	0.001	86.1	83.5	85.3
	0.002	85.2	84.1	86.4

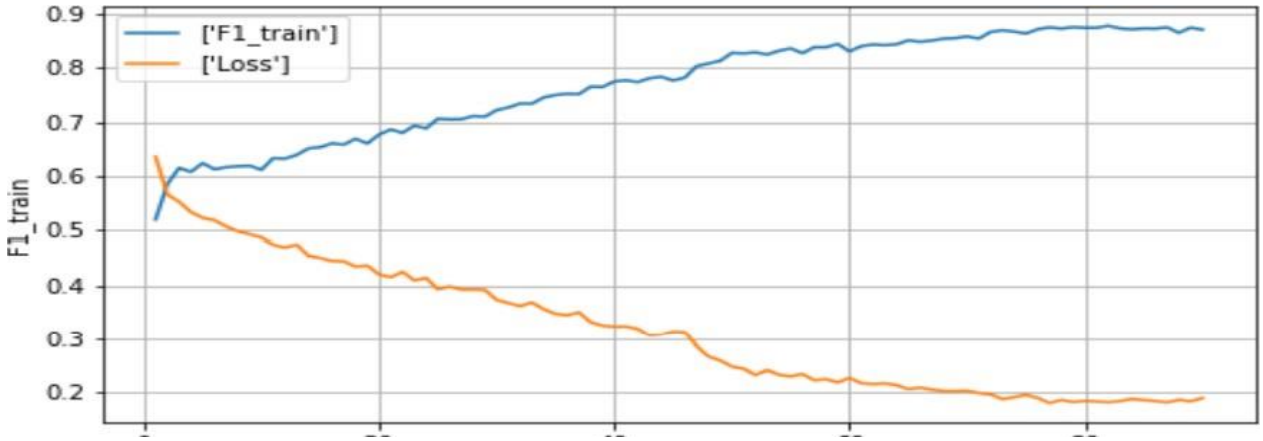


Figure 36: InceptionV3 Loss and Training Chart

Table 12 Loss Outcome *and Accuracy*

Metrics	Accuracy (Mean)			Loss (Mean)		
	Training	Validation	Testing	Training	Validation	Testing
Value as a percentage	88.3	88.8	85.2	6.73	5.6	9.1

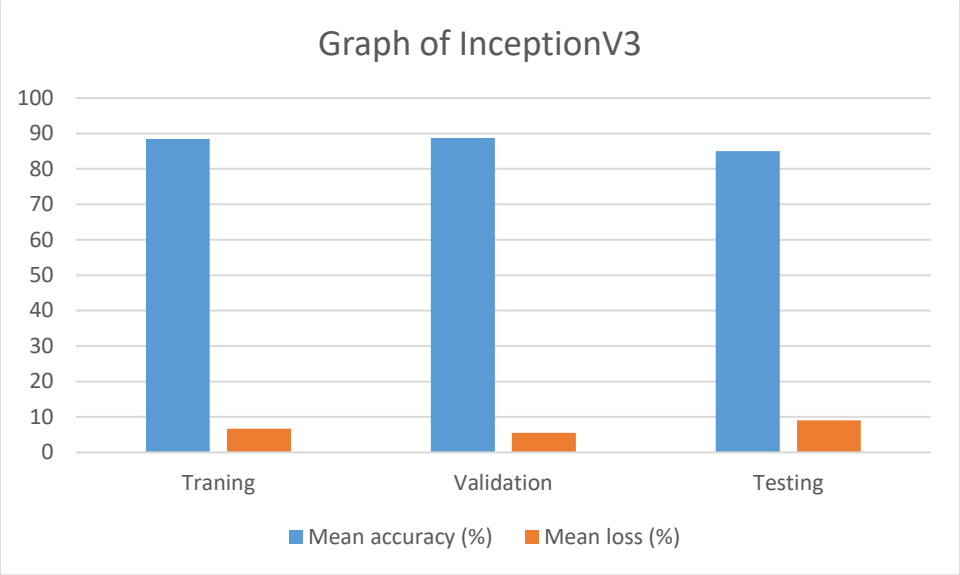


Figure 37: The mean loss and precision of the InceptionV3 graph.

Training and testing yield good results for InceptionV3. Figure 34 shows how the learning capacity of the model develops with epoch and how, when it comes to optimising evolved models, the Adam optimiser works better than the SGD optimiser.

5.5.2.3. Experimental result of MobilenetV2

The inverted residual structure of MobilenetV2 [143], which has a shortcut connection between two narrow bottleneck layers, makes it an efficient transfer learning technique. Another significant aspect of MobilenetV2 is the lightweight, separable convolution. Figure 38 shows the three convolution layers of the MobileNetv2 design, which consists of seventeen levels of bottleneck. Also, each layer's specific details are provided. Given that ReLU6 performs well with low-precision computations, nonlinearity was added by using it. The following is a description of some of MobilenetV2's key attributes.

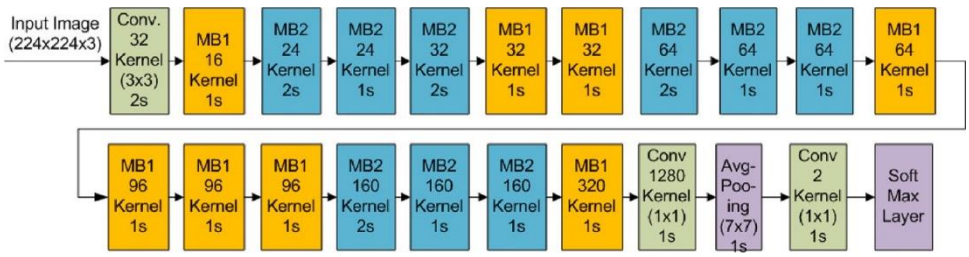


Figure 38: Breast cancer diagnosis using the MobileNetV2 model [85].

```python
import torch
import torch.nn as nn
import torch.nn.functional as F

from typing import List

from timm.data import IMAGENET_DEFAULT_MEAN, IMAGENET_DEFAULT_STD, IMAGENET_INCEPTION_MEAN, IMAGENET_INCEPTION_STD
from .efficientnet_blocks import round_channels, resolve_bn_args, resolve_act_layer, BN_EPS_TF_DEFAULT
from .efficientnet_builder import EfficientNetBuilder, decode_arch_def, efficientnet_init_weights
from .features import FeatureInfo, FeatureHooks
from .helpers import build_model_with_cfg, default_cfg_for_features
from .layers import SelectAdaptivePool2d, Linear, create_conv2d, get_act_fn, hard_sigmoid
from .registry import register_model
```

Figure 39: Code snippet used in the import of MobilenetV2.

i. Separable Convolutions

The procedure known as general convolution involves summarising all these values and filtering every input channel simultaneously. A filter was applied to every input channel simultaneously in the general convolution process, and the output was obtained by combining all the values. MobileNetV2 uses depth-wise separable convolutions to significantly reduce computational cost compared to traditional convolutions. This approach breaks down a regular convolution into two separate steps: a depth-wise convolution and a point-wise convolution. The depth-wise convolution applies a separate filter to each input channel, effectively capturing spatial information. The pointwise convolution then combines the filtered channels, reducing the number of computations needed compared to a single large convolution. depth-wise and point-wise. Convolution was applied simultaneously to one input channel when using depth-wise convolution. All output layers in depth-wise convolution are linearly combined to form a 1 × 1 convolution, which is a point-wise convolution.
When applying k kernels of size $X_f * X_f * 1$ to an input of size $X_t * X_t * p$, a classical convolution or normal produces an output of size $X_g * X_g * p$. Therefore, the processing expense of a standard convolution is high.

$$C_{gen} = X_f^2 * X_g^2 * k * p \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.14)$$

a. Depth-wise Convolutions

Here, the identical input of size $X_t * X_t * p$ was subjected to the filter of size $X_f * X_f *$

1. Since a single kernel has a multiplication number of X_f^2, The number of multiplications involved in applying a kernel over an input image was $X_f^2 * X_g^2$.

There are P channels in the input image. Thus, for a given number of channels, a certain number of kernels are needed. Consequently, to perform depth-wise convolutions, a total number of multiplications needed was:

$$Y_{dc} = X_f^2 * X_g^2 * p \quad \ldots\ldots (5.15)$$

b. Point-wise convolutions

Here, each depth-wise convolution output of type $X_g^2 * p$ is combined using a kernel of size $1 * 1 * p$. Therefore, $X_g^2 * p$ was the required multiplication. The required number of multiplications for f kernels is given by:

$$Y_{pc} = X_f^2 * p * f \quad \ldots\ldots (5.16)$$

When performing a separable depth-wise convolution, the total number of multiplications needed was.

$$Y_{ds} = Y_{dc} + Y_{pc} \quad \ldots\ldots (5.17)$$

Equation (5.18) can be rewritten as follows using the information from Equations (5.14) and (5.17):

$$Y_{ds} = X_f^2 * X_g^2 * p + X_g^2 * p * f \quad \ldots\ldots (5.18)$$

Comparing this equation $P_m(i,j) = I_m(i,j) + KL_{gf}(i,j)$ and (5.17) provides the depthwise separable convolution relative to depth separable computational complexity as a function of generic convolution.:

$$\lambda = \frac{Y_{ds}}{Y_{gen}} = \frac{X_g^2 * p\left(X_f^2 + f\right)}{X_g^2 * X_f^2 * p * f} = \frac{1}{X_f^2} + \frac{1}{f} \quad \ldots\ldots (5.19)$$

In this case, given a 3 × 3 kernel and a 224 × 224 × 3-pixel image, the factor.

$$\lambda = \frac{1}{3^2} + \frac{1}{75 * 75} \simeq \frac{1}{9} \quad \ldots\ldots (5.20)$$

A 224 × 224 × 3 image can be filtered with a minimum of 76 3 × 3 kernels, assuming that every pixel has been filtered simultaneously. More significantly, even with an

increase in the number of filters, factor 1 remains close to 9. This suggests that, compared to a normal convolution, the separable convolution has a computational complexity that is nearly nine times faster. Therefore, it was recommended for quick training and quick execution.

ii. Inverted residuals

Like residual blocks, inverted residual blocks enhance the gradient flow capability by offering a different gradient flow channel. However, unlike residual blocks, the short links were constructed directly between the bottlenecks. As illustrated in Figure 41, MobileNetV2 uses two different types of standard bottleneck structures to increase network efficiency: MB1 and MB2 are the names of MobileNetV2 blocks 1 and 2, respectively. Figure 41 (a) shows a residual bottleneck structure with stride = 1 (MB1), while Figure 41(b) shows the structure bottleneck with stride = 1 or 2 (MB2).

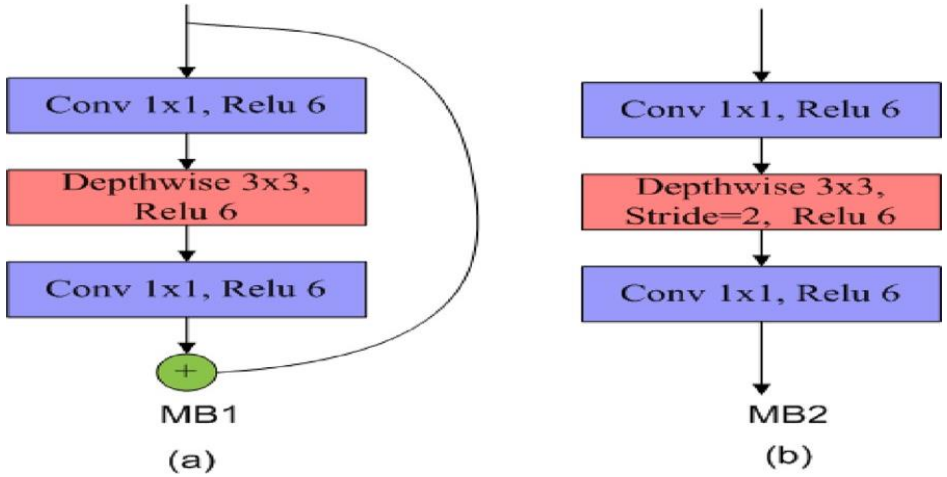


Figure 40: fundamental building pieces of the MobileNetV2 architecture: Blocks in (a) and (b) have one and two strides, respectively. [101].

iii. Linear Bottlenecks

High dimensionality can lead to challenges during training, such as vanishing gradients and increased computational cost. By focussing on the lower-dimensional manifold where the relevant information lies, the network can learn more efficiently. This focus on a low-dimensional manifold is precisely what a linear bottleneck structure helps to achieve. Bottleneck structures are characterised by the addition of

a 1 × 1 convolution before a more costly convolution (such as a 3 × 3 or 5 × 5 convolution). To achieve a network that is computationally more efficient, the depth was decreased while maintaining the same size. In MobileNetV2, the bottleneck layer was linear to reduce information loss.

Table 6's hyper-parameter setting is the same as the MobilenetV2's. A seven-by-seven-by-1024-pixel feature map was obtained. During the training phase of the neural network, the small feature extracted reduces the computing performance of the model. Table 13 displays the results of the experiment.

Table 13 *MobilenetV2 experimental outcome*

	Rate of Learning	Accuracy	Recall	Precision
Image's net weight using the Adam optimizer	0.001	88.7	87.2	87.3
	0.002	86.3	86.6	86.2
15 epochs of SGD optimisation	0.001	88.4	88.3	88.2
	0.002	87.2	87.6	87.8

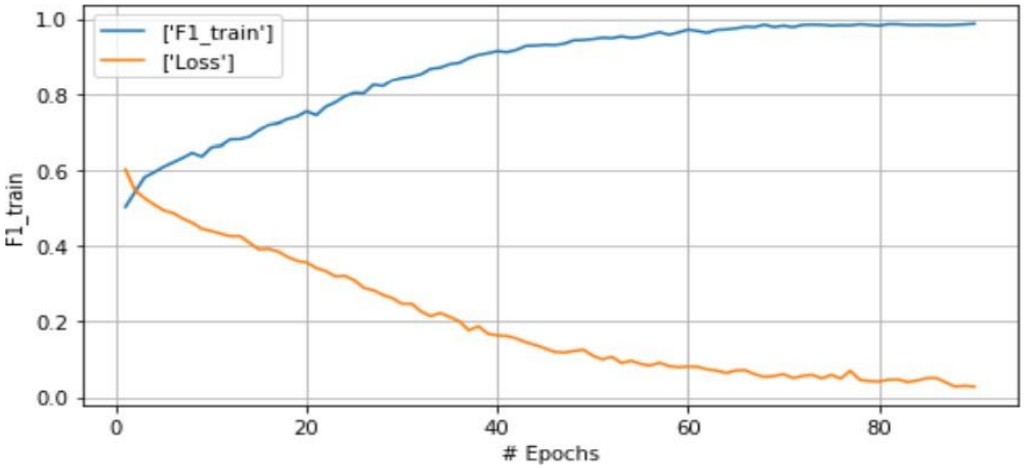


Figure 41: Training and loss of MobilenetV2.

Figure 42 demonstrates how the model performs better with an increasing epoch. Nevertheless, the graph becomes saturated after epoch 80.

Table 14 *Loss outcome of the MobilenetV2 model and accuracy*

Metrics	Accuracy (Mean)			Loss (Mean)		
	Training	Validation	Testing	Training	Validation	Testing
Value as a percentage	94	98	91	11.3	9.2	13.7

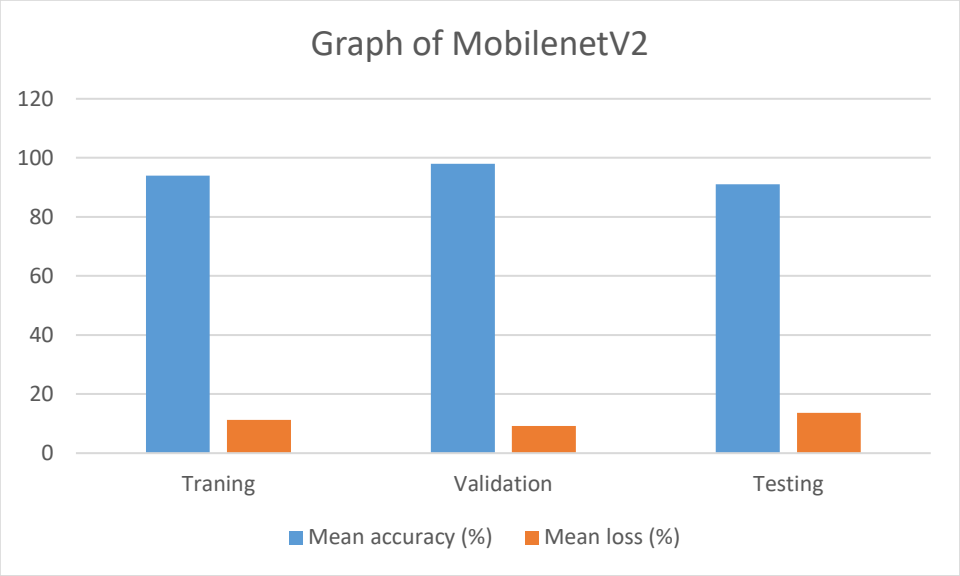


Figure 42: MobilenetV2 Mean Loss Compared to Mean Accuracy.

5.6. Discussion

5.6.1 Comparison of performance outcomes

The following table 15 and Figure 42 demonstrate the comparison results of the four models (CNN, VGG19, InceptionV3, and MobilenetV2). Comparisons were based on accuracy and loss.

Table 15 *Comparison outcomes between models.*

Name (Model)	Accuracy	Loss
CNN model	87.1	14.4
VGG19	79.3	22.0
InceptionV3	89.1	9.3
MobilenetV2	88.2	11.1

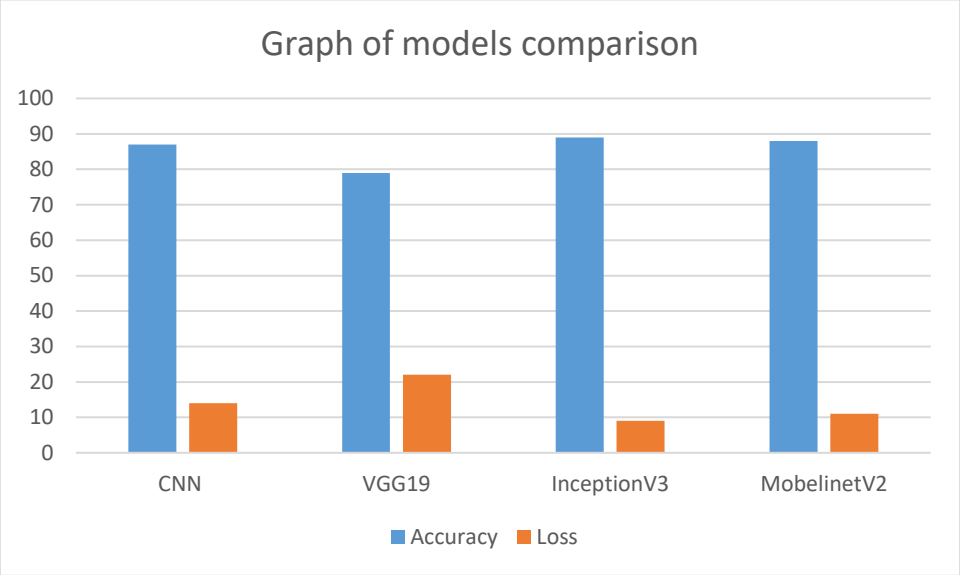


Figure 43: Comparison of models

This study aimed to develop a convolutional neural network (CNN) model to classify breast cancer. Building a CNN model for this task requires a substantial amount of training data. However, training a model from scratch can be limited by the availability of data. To address this, we explored two approaches: training a model from scratch and using transfer learning with pre-trained models like VGG19, InceptionV3, and MobileNetV2. Although the model trained from scratch achieved good accuracy in training (88%) and validation (89%), its performance dropped on unseen data during testing (86%). This highlights the limitations of training from scratch with limited data.

Transfer learning offers a solution by utilising pre-trained models on larger datasets. These models can then be fine-tuned for the specific task of breast cancer classification, even with limited data. The pre-trained models were further optimised by evaluating different hyperparameter configurations.

InceptionV3 achieved high accuracy on the training data, but its performance suffered on the test set. This suggests overfitting, where the model memorises the training data peculiarities and fails to generalise well to unseen data. The large number of parameters in InceptionV3, combined with a limited dataset, probably contributed to this overfitting. Although VGG19 exhibited lower training accuracy compared to InceptionV3, it performed better on the test set based on evaluation metrics. This

indicates a better generalisation of unseen data. MobileNetV2 showed a lower performance compared to both VGG19 and InceptionV3 in the evaluation metrics. The model trained from scratch offers the advantage of lower computational cost and training time. However, its accuracy falls short of InceptionV3, even though it avoids overfitting due to its simpler structure. These results suggest that transfer learning with VGG19 might be a good choice for this breast cancer classification task, balancing accuracy and potential computational limitations.

Based on Figure 45 below, the training and validation accuracies for all three models (VGG19, InceptionV3, and the CNN recommended model) are very similar. This suggests that overfitting might not be a major concern. However, it is important to consider other metrics for generalisability beyond this limited difference. Metrics such as testing accuracy on a completely unseen data set or performance on various data subsets would provide a more robust evaluation. The recommended model shows the highest training and validation accuracy among the three. To definitively determine its generalisability, testing on an unseen dataset is recommended.

Training loss is a metric that indicates how well a model's predictions align with the actual values during training. Lower training loss signifies better performance on the training data. The training loss values for VGG19 (7), InceptionV3 (14.3) and the recommended model (4.3) suggest that the recommended model achieved the best fit in the training data. This is because it has the lowest training loss, indicating a smaller difference between the predicted and actual values.

The validation loss values also follow the same trend. VGG19 (7.7) has the closest validation loss to its training loss, followed by the recommended model (5.7) and InceptionV3 (14.3). Although a small gap between training and validation loss is desirable, it is important to consider the limitations discussed earlier regarding generalisability. Testing on unseen data is crucial to confirm that low training and validation loss translates to good performance on new data.

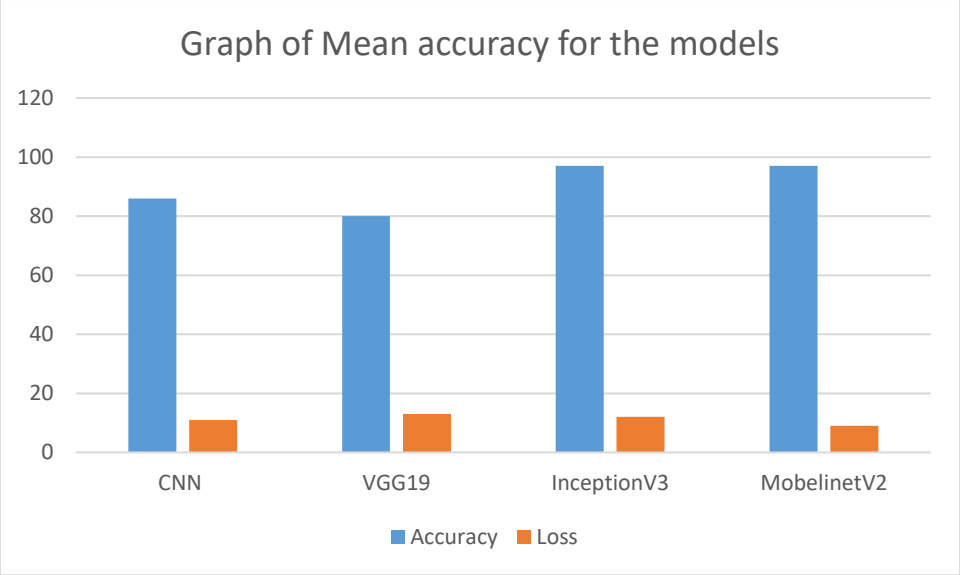


Figure 44: Average precision throughout the four tests.

Although testing the models with fictitious data produced positive results (recommended model: 77% on the VGG19 test, 88% on the InceptionV3 test, 88% on the MobilenetV2 test), it is important to acknowledge that this might not reflect real-world performance. Fictitious data may not capture the full complexity of real breast cancer images.

In particular, the recommended model achieved an accuracy of 88% in the InceptionV3 and MobilenetV2 test datasets, outperforming the pre-trained versions of these models (VGG19 at 77%, InceptionV3 at unknown accuracy, MobilenetV2 at unknown accuracy). This suggests that the recommended model may have learnt effective features for breast cancer. To definitively assess the generalisability of the model, testing on a real-world breast cancer image dataset is crucial. This helps to provide a more reliable indication of its effectiveness in a clinical setting.

A significant finding is that the test loss for the proposed model (6.23) is considerably lower than VGG19 (22.1) and InceptionV3 (27.2). Test loss measures how well the model performs on unseen data. This lower test loss suggests that the proposed model generalises well and performs effectively on new data, not just training data.

A possible explanation for the improved performance of the suggested model is that it effectively learnt the key characteristics that differentiate healthy and unhealthy tissues

in the training data set. These characteristics might be subtle details or patterns that are crucial to an accurate classification.

The use of smaller-sized filters in the convolution layers is another potential factor contributing to the model's success. Smaller filters can focus on capturing finer details within images, which may be crucial for distinguishing between healthy and unhealthy breast tissue. By using smaller filters, the model might be better at preserving these finer details during the convolution process, potentially leading to improved classification accuracy.

While some deep learning algorithms, particularly those related to computer vision for image categorisation, can be trained on massive datasets containing hundreds of millions to billions of images and tens of millions of parameters, recent advances have shown that efficient models can achieve good results with fewer parameters. This allows training on less powerful hardware. These algorithms often leverage high-performance computing hardware, particularly Graphics Processing Units (GPUs), which are well-suited for the parallel processing tasks involved in deep learning training. However, we may train and achieve better results with smaller networks, fewer hardware components, fewer parameters, and less data. More accurate results would have been obtained if the images in the dataset had been shot under steady lighting, with an item at a constant distance from the camera, and with excellent focus. Pre-processing the images to remove unwanted features and noise also increases the accuracy of the model.

5.6.2 Computational Complexity Analysis

In a neural network model, the number of floating-point operations (FLOPs) performed during a single forward pass through the network determines the inference time (how long it takes to make a prediction) of the model. This essentially means that a higher number of FLOPs translates into a longer time required for the model to predict new data. Here are the guidelines for calculating the FLOPs in a model.

Convolutions - FLOPs equal 2 x Kernel Count x Kernel Form x Output Shape

Layers that are fully connected: FLOPs = 2 x input size x output size layers for pooling: FLOPs = Height x Depth x Width of an image; if a layer has a stride other than 1, FLOPs = (Height / Stride) x Depth x (Width / Stride) of the image. In a convolutional operation, the output shape is equal to the sum of the input and kernel shapes plus one.

All 1000 neurons were fully connected to a layer of 512 neurons at the top of each network's development, and 2 neurons at the bottom connected to the output layer.

Therefore, the FLOPs for the final two layers are the
Penultimate layer: 512 x 1000 x 2 = 1,024,000 FLOPs
Output layer: 512 x 2 x 2 = 2,048 FLOPs

Total: 1,024,000 + 2,048 = 1,026,048 FLOPs

VGG19 has 19.6 billion FLOPs, InceptionV3 5.69 billion Flops, and MobilenetV2 has 336.34 million FLOPs. Table 16 illustrates how 1,026,048 FLOPs are added to the appropriate FLOPs of the parent network to get the total floating-point operations for each of the produced models.

Table 16 *Floating point operations for each model developed.*

Model	FLOPs
VGG19	19,7
InceptionV3	5,79
MobilenetV2	0,337

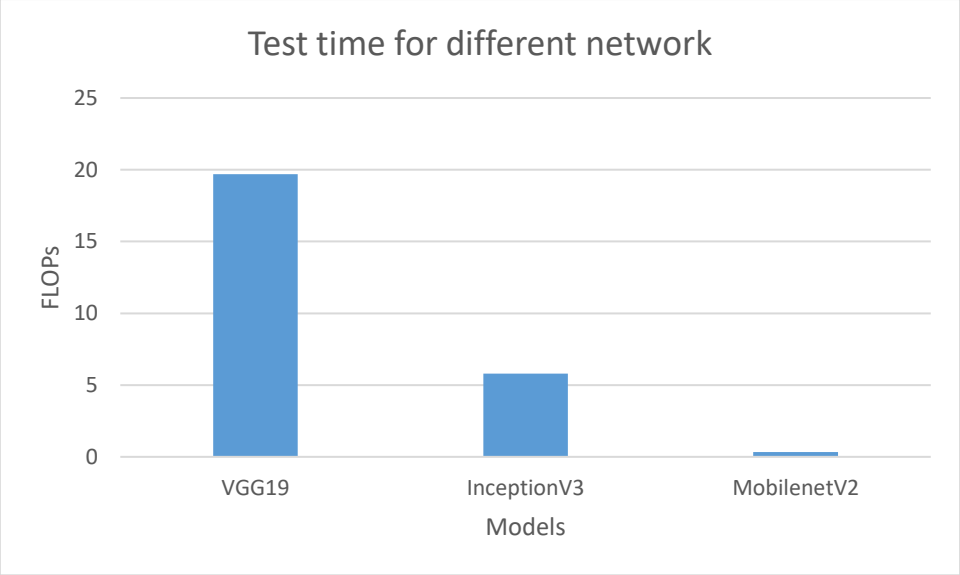


Figure 45: FLPOs of the developed models.

It is also important to pay attention to the neural network parameters, which dictate the size and space complexity of the created model. Table 17 presents the network parameters for each model created in this work, while Table 18 displays the network test times.

Table 17 N*umber of model parameters.*

Model	Trainable parameters	Non-trainable parameters	Total parameters
VGG19	16,754,778	20,384,024	36,138,803
InceptionV3	33,970,555	7,504,037	40,474,593
MobileNet-V2	41,578,944	2,984,257	44,562,202

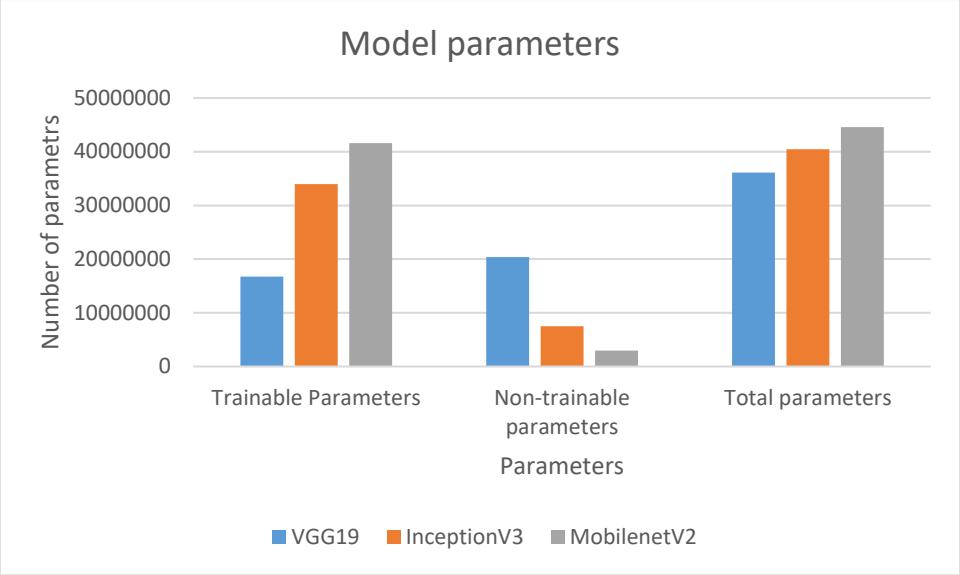


Figure 46: Parameters of the trained model.

Table 18 *Test time of the developed models.*

Model	**Test time(s)**
VGG19	10.86
InceptionV3	6.01
MobileNetV2	4.24

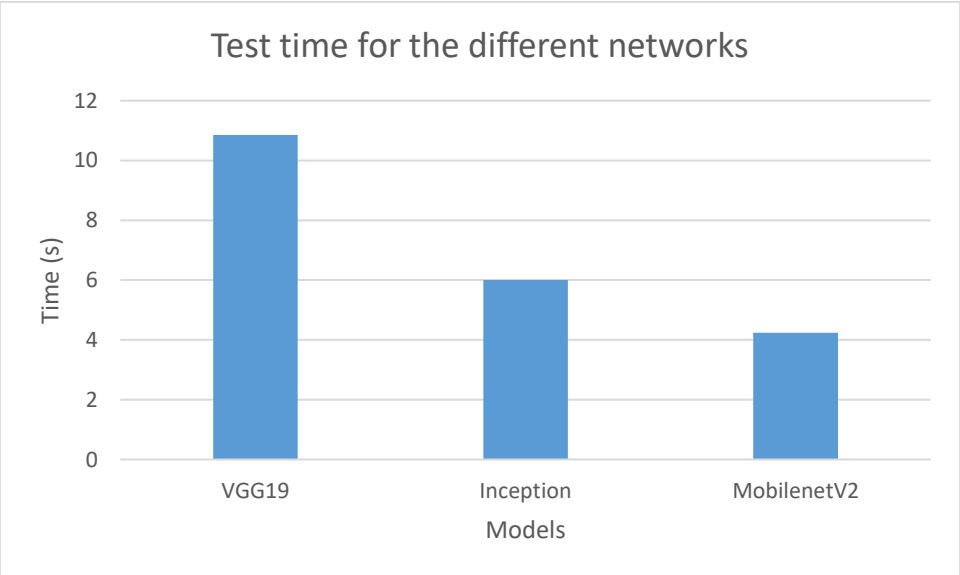


Figure 47: Test time for the different built networks.

As can be seen in the plot in Figure 48, the models with the shortest test times are MobilenetV2, InceptionV3, and VGG19. For every model in its forward pass, this is consistent with the floating-point operation. Although it is reasonable to expect the plots of Figures 48 and 46 to be the same, this is not always the case because the test times for each model are also influenced by other operations that the computer processor is running at the time of each test, which can cause significant variations from time to time.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1. Introduction

This dissertation proposes an integrated deep learning framework for a computer-aided diagnosis (CAD) system designed to assist radiologists. Functioning as a second-reading tool, the system aims to improve breast cancer risk assessment by providing advanced analysis of suspicious lesions. The presented work suggested three primary procedures for processing a mammogram to identify anomalies and determine whether they are of mass, calcification, architectural distortion, or another kind. This framework specifically targets mass lesions in mammograms, automatically generating accurate contours to aid radiologists in assessing tumour grade and malignancy. Furthermore, the dissertation uses an ensemble learning approach that combines the strengths of multiple deep learning models. This technique aims to achieve superior performance and reduce false negatives, leading to more accurate diagnoses.

Furthermore, our methodology offered better detection performance compared to previous models, as the detection of breast lesions is crucial for CAD systems and fully integrated breast cancer diagnostics. By automatically generating accurate contours of mass lesions, this framework not only helps radiologists assess malignancy, but also offers a unique architecture specifically designed for mammography breast mass segmentation, further reducing the risk of misdiagnosis. The suggested architecture included the latest changes, such as the Atrous Spatial Pyramid Pooling (ASPP) idea, residual blocks, the attention mechanism, and others that were presented to address the difficulties associated with pixel-by-pixel segmentation in medical images.

The final step uses a stacked ensemble of neural networks to achieve a comprehensive framework for the diagnosis of breast cancer. This model takes a multifaceted approach, predicting the BI-RADS score (2-6) and pathology (malignant or benign), while simultaneously classifying mass shape (round, oval, lobulated, irregular). Incorporating segmented regions of interest (ROIs) significantly improved

performance. This suggests that masking background tissue from tumour boundaries improves classification accuracy, reducing false positives and negatives.

This dissertation proposes a comprehensive deep learning-based framework for a breast cancer CAD system. The system operates in three stages: abnormality detection, precise mass segmentation, and assessment of malignancy and classification. This integrated deep learning framework aspires to become a valuable clinical decision support tool. By providing automated mass tumour analysis and a second opinion, it aims to help radiologists make more informed diagnoses.

This work classified mammography image scans for breast cancer analysis by developing three convolutional neural network models based on transfer learning from existing successful convolutional neural network models. The models that were pre-trained with roughly 18494 images to categorise 1000 objects in an image are VGG19, InceptionV3, and MobilenetV2. The data set was collected from Mankweng Hospital in Limpopo, South Africa. To improve the training data set, several preprocessing techniques were applied. Data balancing addresses the imbalances in the data to ensure that the model learns effectively from both categories. Contrast-limited adaptive histogram equalisation (CLAHE) is a technique that enhances image contrast, potentially improving the model's ability to detect subtle features. Data augmentation involves artificially increasing the size by rotating (0-40 degrees), scaling, flipping (horizontal/vertical), and translating the images. This helps the model to generalise better to unseen data. 10% went toward testing, 10 percent went toward validation, and 80 percent went toward training.

6.2. Conclusion about research questions.

Building upon the research questions established in Section 1.4 of Chapter 1, this chapter delves into a thorough investigation of each question. The findings of this investigation are presented in detail in the following sections.

6.2.1 Research question number one.

What is the current methodology for real-time multidisease classification using low-

power devices and breast cancer detection?

Convolutional neural networks (CNN) are powerful deep learning techniques that excel in image-related tasks such as classification, recognition, and detection [5, 29, 30, 31]. Their success is evident in numerous competitions. There are two main approaches to building CNN models, building from scratch and transfer learning with pretrained models. When building a CNN from scratch, the model architecture is meticulously designed with components such as convolutional layers, pooling layers, activation functions, and fully connected layers. In addition, hyperparameters such as the number of filters and the learning rate need to be configured. However, the main challenge with this approach is the requirement of data. A limited amount of training data can hinder the network's ability to learn effectively, potentially leading to poor performance on unseen data (testing set). Transfer learning with pre-trained models is an alternative; transfer learning leverages pre-trained models that have already been trained on vast datasets for generic image recognition tasks. These pretrained models can then be fine-tuned for the specific task at hand (e.g., breast cancer classification), even with limited data. This approach often leads to better performance compared to training a model entirely from scratch with limited data.

6.2.2 Research question number two.

How can a model be created to identify breast cancer?

Our investigation used a convolutional neural network (CNN) model, a leading deep learning technique for image analysis tasks such as breast cancer detection. The model produced encouraging results, suggesting its potential for real-world applications.

6.2.3 Research question number three.

How well does the developed breast cancer detection system identify illness?

To evaluate the models' performance, we employed common classification metrics such as precision, recall, accuracy, and F1-score. These metrics provide insight into how well the models can correctly identify different classes (e.g., healthy vs. cancerous tissue). Importantly, each data class contained an equal number of images. This balanced dataset ensures that the model does not get biased towards the majority

class and leads to a more reliable evaluation. Among the pre-trained models, InceptionV3 stood out with the best performance, achieving an impressive accuracy of 88.4% on unseen data. This suggests that the model can effectively generalise its knowledge to identify breast cancer in new images it has not encountered during training.

Our initial research question aimed to identify the best feature extractor for the classification of breast cancer. Through evaluation, InceptionV3 emerged as the best performer. InceptionV3 achieved the best performance among all pre-trained models evaluated, with MobilenetV2 coming in second. This suggests that InceptionV3 is a highly effective feature extractor for this task. Our findings reveal that InceptionV3 is a powerful feature extractor for breast cancer classification tasks. Additionally, the proposed model significantly outperforms current methods based on the evaluation criteria, suggesting its potential for real-world applications.

6.3. Future work

Future directions can focus on expanding the system's capabilities and enhancing performance. One key direction is to expand the system's capabilities by merging various mammogram datasets. This broader data set can expose the model to a wider range of variations, strengthening its generalisability and ability to perform well on unseen data. Currently, the system focusses on detecting breast cancer. Future iterations could integrate the detection and classification of additional abnormalities, such as calcifications and architectural distortions. This comprehensive approach would provide a more robust diagnostic tool for radiologists. Integrating radiology reports and medical text data along with image data holds immense potential. This approach could improve the interpretability of the model, allowing healthcare professionals to understand the reasoning of the model. In addition, it could pave the way for affordability in a clinical setting by potentially reducing the reliance on expensive imaging techniques. Long training times are a common challenge in deep learning. Exploring techniques to optimise training efficiency is crucial for practical implementation. By streamlining the training process, we can ensure that the model's effectiveness is readily available in clinical settings.

Although this work focused on mammogram analysis, the core principles and deep learning models developed here hold immense potential for broader applications in cancer diagnosis using medical imaging. Imagine the possibility of adapting these models to detect lung cancer on chest radiographs, analyse brain tumours on magnetic resonance imaging, or identify lung diseases on CT scans. By applying the learnings of this research to other imaging modalities, we could significantly expand the impact of deep learning in the fight against cancer. The proposed architecture, designed for 2D mammograms, demonstrates the power of deep learning in this domain. An exciting future direction lies in extending this architecture to handle 3D medical images such as CT scans and MRIs. This advancement would allow the model to capture the spatial relationships within these complex data structures, leading to even more robust and comprehensive diagnoses.

References

[1] Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA. Cancer J Clin 2018; 68:394–424. https://doi.org/10.3322/caac.21492 PMID: 30207593.

[2] GLOBOCAN. Cancer Today 2018. http://gco.iarc.fr/today/home (accessed 20 Jan 2020).

[4] I. Allaouzi and M. Ben Ahmed, "A Novel Approach for Multi-Label Chest X-Ray Classification of Common Thorax Diseases," IEEE Access, vol. 7, p. 64279–64288, 2019.

[5] T. G. Debelee, S. R. Kebede, F. Schwenker, and Z. M. Shewarega,, "Deep Learning in Selected Cancers' Image Analysis—A Survey,," Journal of Imaging, vol. 6, no. 11, p. 121, 2020.

[6] Biniyam Tefera Deressa, Nikola Cihoric, Daniel Rauch, "Breast cancer care in northern Ethiopia cross-sectional analysis," open accesss, pp. 1-6, 2019.

[7] T. Assefa, D. Haile Mariam, W. Mekonnen, M. Derbew, and W. Enbiale, "Physician distribution and attrition in the public health sector of Ethiopia," Risk Management and Healthcare Policy, vol. 9, no. 2, p. 285–295, 2016.

[8] WHO report on cancer: setting priorities, investing wisely, and providing care for all. Geneva: World Health Organization; 2020. Licence: CC BY-NC-SA 3.0 IGO.

[9] Joko-Fru WY, Miranda-Filho A, Soerjomataram I, Egue M, Akele-Akpo M-T, N'da G et al. Breast cancer survival in sub-Saharan Africa by age, stage at diagnosis and human development index: a population-based registry study. Int J Cancer 2020; 146:1208–18. https://doi.org/10.1002/ijc.32406 PMID: 31087650

[10] Sengayi-Muchengeti M, Joko-Fru WY, Miranda-Filho A, Egue M, Akele-Akpo M-T, N'da G et al. Cervical cancer survival in sub-Saharan Africa by age, stage at diagnosis and Human Development Index (HDI): A population-based registry study. Int. J. Cancer. https://doi.org/10.1002/ijc.33120 PMID: 32449157

[11] Ngoma T, Mandeli J, Holland J. Downstaging cancer in rural Africa. Int J Cancer 2015; 136:2875–79. https://doi.org/10.1002/ijc.29348 PMID: 25408458

[12] Macleod U, Mitchell ED, Burgess C, Macdonald S, Ramirez AJ. Risk factors for delayed presentation and referral of symptomatic cancer: evidence for common cancers. Br J Cancer 2009; 101: S92–101. https://doi.org/10.1038/sj.bjc.6605398 PMID: 19956172

[13] McKenzie F, Zietsman A, Galukande M, Anele A, Adisa C, Parham G et al. Drivers of advanced stage at breast cancer diagnosis in the multicountry African breast cancer—disparities in outcomes (ABCDO) study. Int J Cancer 2018; 142:1568–79. https://doi.org/10.1002/ijc.31187 PMID: 29197068

[14] Akuoko C, Armah E, Sarpong T, Quansah DY, Amankwaa I, Boateng D. Barriers to early presentation and diagnosis of breast cancer among African women living in sub-Saharan Africa. PLoS ONE 2017; 12:e0171024. https://doi.org/10.1371/journal.pone.0171024 PMID: 28192444

[15] McCutchan GM, Wood F, Edwards A, Richards R, Brain KE. Influences of cancer symptom knowledge, beliefs, and barriers on cancer symptom presentation in relation to socioeconomic deprivation: a systematic review. BMC Cancer 2015; 15:1000. https://doi.org/10.1186/s12885-015-1972-8 PMID: 26698112

[16] Shahab L, McGowan JA, Waller J, Smith SG. Prevalence of beliefs about actual and mythical causes of cancer and their association with socio-demographic and health-related characteristics: findings from a cross-sectional survey in England. Eur J Cancer 2018; 103:308–16. https://doi.org/10.1016/j.ejca.2018. 03.029 PMID: 29705530.

[17] Devi BCR, Tang TS, Corbex M. Reducing by half the percentage of late-stage presentation for breast and cervix cancer over 4 years: a pilot study of clinical downstaging in Sarawak, Malaysia. Ann Oncol 2007; 18:1172–6. https://doi.org/10.1093/annonc/mdm105 PMID: 17434897.

[18] Muthoni A, Miller AN. An Exploration of rural and urban Kenyan women's knowledge and attitudes regarding breast cancer and breast cancer early detection

measures. Health Care Women Int 2010; 31:801–16. https://doi.org/10.1080/07399331003628453 PMID: 20677038.

[19] Moodley J, Cairncross L, Naiker T, Constant D. From symptom discovery to treatment—women's pathways to breast cancer care: a cross-sectional study. BMC Cancer 2018; 18:312. https://doi.org/10. 1186/s12885-018-4219-7 PMID: 29562894.

[20] Moodley J, Scott SE, Mwaka AD, Constant D, Githaiga JN, Stewart TS et al. Development and validation of the African Women Awareness of CANcer (AWACAN) tool for breast and cervical cancer. PLoS ONE 2019; 14:e0220545. https://doi.org/10.1371/journal.pone.0220545 PMID: 31386684.

[21] S. Kaya and M. Yağanoğlu, "An Example of Performance Comparison of Supervised Machine Learning Algorithms Before and After PCA and LDA Application: Breast Cancer Detection," Istanbul, Turkey, 2020.

[22] Z. Zheng, H. Zhang, X. Li, S. Liu, and Y. Teng, "ResNet-Based Model for Cancer Detection," 2021 IEEE Int. Conf. Consum. Electron. Comput. Eng. ICCECE 2021, no. Iccece, p. pp. 325–328, 2021.

[23] M. Li, "Research on the detection method of breast cancer deep convolutional neural network based on computer aid," Proceedings of IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers,, pp. 536-540, 2021.

[24] H. Peng, Y. Jiang, X. Li, H. Luo, and S. Yin,, "A novel redundant information elimination aided classification approach for cervical cancer diagnosis,," Proc. 2019 11th CAA Symp. Fault Detect. Supervision, Saf. Tech. Process. SAFEPROCESS 2019, p. 631–636, 2019.

[25] Sung, Hyuna, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," CA: A Cancer Journal for Clinicians, Vols. 71, no. 3, p. 209–249, 2021.

[26] M. Slimi, B. Jmai, P. Mendes, and A. Gharsallah, "Breast cancer detection based on CPW antenna," IEEE Access, vol. 4, no. 8, pp. 3-6, 2019.

[27] B. T. Deressa, N. Cihoric, E. V. Badra, A. Tsikkinis, "Breast cancer care in northern Ethiopia-cross-sectional analysis,," BMC Cancer, vol. 19, no. 8, p. 1–7, 2019.

[28] U. Kamath, J. Liu, and J. Whitaker, Deep Learning for NLP and Speech Recognition, vol. 2, mumbai: IEEE, 2019.

[29] S. Khan, H. Rahmani, S. Afaq, A. Shah, and M. Bennamoun, "A Guide to Convolutional Neural Networks for Computer Vision," IEEE access, vol. 1, no. 8, pp. 29-224, 2020.

[30] Getachew, Sefonias, "Perceived barriers to early diagnosis of breast Cancer in south and southwestern Ethiopia: a qualitative study," BMC Women's Health, vol. 20, no.1, pp. 1-8, 2020.

[31] Patterson, Josh and Gibson, Adam, Deep learning: A practitioner's approach, O'Reilly Media, Inc., 2017.

[32] Deressa, Biniyam Tefera, "Breast cancer care in northern Ethiopia-cross-sectional analysis," BMC Cancer, vol. 19, no. 1, pp. 1-7, 2019.

[33] Akram Farhadi, David Chen, Rozalina McCoy, Christopher Scott, John A. Miller, Celine M. Vachon, Che Ngufor, "Breast Cancer Classification using Deep Transfer Learning on Structured Healthcare Data," in IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2019.

[34] Freer TW, and Ulissey MJ. “Screening mammography with computer-aided detection: a prospective study of 12,860 patients in a community breast centre.” Radiology 2019; 220:781–6.

[35] Gur D, Sumkin JH, Rockette HE, Ganott M, Hakim C, and Hardesty L. “Changes in breast cancer detection and mammography recall rate after the introduction of a computer-aided detection system”. J Natl Cancer Inst 2020; 96:185–90.

[36] Giger ML, Huo Z, Kupinski MA, and Vyborny CJ. “Computer-aided diagnosis in mammography”. In: Fitzpatrick JM, Sonka M, editors. The handbook of medical imaging. Medical imaging processing and analysis, vol. 2. SPIE; 2019. p. 915–1004.

[37] Giger ML. “Computerized analysis of images in the detection and diagnosis of breast cancer Seminars in Ultrasound”. CT MRI 2020;25: 411–8.

[38] Lodwick GS, Haun CL, and Smith WE. “Computer diagnosis of primary bone tumor”. Radiology 2019; 80:273–5.

[39] Myers PH, Nice CM, and Becker HC. “Automated computer analysis of radiographic images”. Radiology 2019; 83:1029–33.

[40] Huang HK. “PACS and imaging informatics: basic principles and applications”. John Wiley & Sons, Inc.; 2010.

[41] Ishida M, Kato H, Doi K, and Frank PH. “Development of a new digital radiographic image processing system”. Proc SPIE 2018; 347:42–8.

[42] Ishida M, Frank PH, Doi K, and Lehr JL. “High-quality digital radiographic images: improved detection of low-contrast objects and preliminary clinical studies”. Radio Graphics 2020; 3:325–38.

[43] Winsbarg F, Elkin M, and May J. “Detection of radiographic abnormalities in mammograms through optical scanning and computer analysis”. Radiology 2019; 89:211–5.

[44] Kruger RP, Towns JR, and Hall DL.” Automated radiographic diagnosis via feature extraction and classification of cardiac size and shape descriptors”. IEEE Trans Biomed Eng 2019; BME-19(3):174–86.

[45] Giger ML, Doi K, and MacMahon H. “Computerized detection of lung nodules in digital chest radiographs”. Proc SPIE 2020; 767:384–6.

[46] Giger ML, Doi K, and MacMahon H. “Image feature analysis and computer-aided diagnosis in digital radiography. 3. Automated detection of nodules in peripheral lung fields”. Med Phys 2019; 15:158–66.

[47] Fujita H, Doi K, Fencil LE, and Chua KG. “Image feature analysis and computer-aided diagnosis in digital radiography, and Computerized determination of vessel sizes in digital subtraction angiography”. Med Phys 2018; 14:549–56.

[48] Hoffmann KR, Doi K, Chan HP, Fencil L, Fujita H, and Muraki A. “Automated tracking of the vascular tree in DSA images using a double-square-box region-of-search algorithm”. Proc SPIE 2019; 626:326–33.

[49] Chan HP, Doi K, Galhotra S, Vyborny CJ, MacMahon H, and Jokich PM. “Image feature analysis and computer-aided diagnosis in digital radiography. 1. Automated detection of microcalcifications in mammography”. Med Phys 2019; 14:538–48.

[50] Doi K, MacMahon H, Giger ML, and Hoffmann KR, editors. “Computer-aided diagnosis in medical imaging”. Amsterdam: Elsevier; 2019. p. 3–560.

[51] Doi K. “Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology”. Phys Med Biol 2010;51: R5–27.

[52] Doi K, Chan H-P, and Giger ML. “Method and system for enhancement and detection of abnormal anatomic regions in a digital image”. United States Patent 4,907,156 (2019).

[53] Nishikawa RM, Doi K, Giger ML, Schmidt RA, Vyborny CJ, and Monnier Cholley L. “Computerized detection of clustered microcalcifications: evaluation of performance using mammograms from multiple centres”. Radio Graphics 2020; 15:443–52.

[54] Abe H, MacMahon H, Engelmann R, Li Q, Shiraishi J, and Katsuragawa S. “Computer-aided diagnosis in chest radiology: results of large-scale observer tests performed at the 2004–2010 RSNA Scientific Assemblies”. Radio Graphics 2019; 23:255–65.

[55] MacMahon H, Engelmann R, Behlen F, Hoffmann KR, Ishida T, and Roe C. “Computer-aided diagnosis of pulmonary nodules: results of a large-scale observer test”. Radiology 2020; 213:723–6.

[56] Metz CE. “ROC methodology in radiologic imaging”. Invest Radiol 2019; 21:720–33.

[57] Metz CE. “Some practical issues of experimental design and data analysis in radiological ROC studies”. Invest Radiol 2019; 24:234–45.

[58] Moberg K, Bjurstam N, Wilczek B, Rostgard L, Egge E, and Muren C. “Computer-assisted detection of interval breast cancers”. Eur J Radiol 2018; 39:104–10.

[59] Chan HP, Doi K, Vyborny CJ, Schmidt RA, Metz CE, and Lam KL. “Improvement in radiologists’ detection of clustered microcalcifications on mammograms: the potential of computer-aided diagnosis”. Invest Radiol 2019; 25:1102–10.

[60] Birdwell RL, Bandodkar P, Ikeda DM. “Computer-aided detection with screening mammography in a university hospital setting”. Radiology 2019; 236:451–7.

[61] Cupples TE, Cunningham JE, Reynolds JC. “Impact of computer-aided detection in a regional screening mammography program”. AJR 2018; 185:944–50.

[62] Warner EE, and Mulshine JL. “Lung cancer screening with spiral CT: toward a working strategy”. Oncology (Williston Park) 2019; 18:564–75 [discussion 578, 583–564, 587].

[63] Kaneko M, Eguchi K, Ohmatsu H, Kakinuma R, Naruke T, and Suemasu K. “Peripheral lung cancer: screening and detection with low-dose spiral CT versus radiography”. Radiology 2019; 201:798–802

[64] Yoshida H, Masutani Y, MacEneaney P, Rubin D, and Dachman AH. “Computerized detection of colonic polyps at CT colonography based on volumetric features: a pilot study”. Radiology 2020; 222:327–36.

[65] Yoshida H, Nappi J, MacEneaney P, Rubin D, and Dachman AH. “Computer-aided diagnosis scheme for the detection of polyps at CT colonography”. Radio Graphics 2020; 22:963–79.

[66]. Lahoura, V., H. Singh, A. Aggarwal, B. Sharma, M. A. Mohammed, R. Damaševičius, S. Kadry, and K. Cengiz, et al. 2021. Cloud Computing-Based Framework for Breast Cancer Diagnosis Using Extreme Learning Machine. Diagnostics 11 (2):241. doi:10.3390/diagnostics11020241.

[67]. Mohanty, A. K., M. R. Senapati, and S. K. Lenka. 2013. An improved data mining technique for classification and detection of breast cancer from mammograms. Neural computing & applications 22 (S1):303–10. doi:10.1007/s00521-012-0834-4.

[68]. Hagos, Y. B., A. G. Mérida, and J. Teuwen. 2018. Improving breast cancer detection using symmetry information with deep learning. In Danail Stoyanov et al. Image Analysis for Moving Organ, Breast, and Thoracic Images, 90–97. Springer.

[69]. Zeebaree, D. Q., H. Haron, A. M. Abdulazeez, and D. A. Zebari, et al. 2019a. Machine learning and region growing for breast cancer segmentation. In 2019 International Conference on Advanced Science and Engineering (ICOASE), Duhok, Iraq, 88–93.

[70]. Han, F. 2015. Texture feature analysis for computer-aided diagnosis on pulmonary nodules. Journal of digital imaging 28 (1):99–115. doi:10.1007/s10278-014-9718-8.

[71]. LeCun, Y., Y. Bengio, and G. Hinton. 2015. Deep learning. Nature 521 (7553):436–44. doi:10.1038/nature14539.

[72] Kruger RP, Thompson WB, Turner AF. “Computer diagnosis of pneumoconiosis”. IEEE Trans Syst Man Cybernetics 2020; SMC-4(1):44–7.

[73] Toriwaki J, Suenaga Y, and Negoro T. “Pattern recognition of chest X-ray images”. Computer Graphics Image Process 2019; 2:252–71.

[74] American Journal of Roentgenology. 2013;201: W662-W670. 10.2214/AJR.12.10153.

[75] Doi, K. (2006). Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology. Physics in Medicine & Biology, 51(13), R5.

[76] Dibden, A., Offman, J., Duffy, S. W., & Gabe, R. (2020). Worldwide review and meta-analysis of cohort studies measuring the effect of mammography screening programmes on incidence-based breast cancer mortality. Cancers, 12(4), 976.

[77] Fiorica, J. V. (2016). Breast cancer screening, mammography, and other modalities. Clinical obstetrics and gynecology, 59(4), 688-709.

[78] Johnson, B. (2021). Asymmetries in Mammography. Radiologic Technology, 92(3), 281M-298M.

[79] van Seijen, M., Lips, E. H., Thompson, A. M., Nik-Zainal, S., Futreal, A., Hwang, E. S., & Wesseling, J. (2019). Ductal carcinoma in situ: to treat or not to treat, that is the question. British journal of cancer, 121(4), 285-292.

[80] Lamb, L. R., Mohallem Fonseca, M., Verma, R., & Seely, J. M. (2020). Missed breast cancer: effects of subconscious bias and lesion characteristics. Radiographics, 40(4), 941-960.

[81] K. Varshini, R. K. Sethuramamoorthy, V. Kumar, S. A. Shree, and S. Deivarani, "Breast cancer prediction using machine learning techniques," Int. J. Adv. Sci. Technol, vol. 29 , no. 6 spatial issue, p. 2026–2032, 2020.

[82] Y. Cheng and B. Li, "Image segmentation technology and its application in digital image processing," Proc. IEEE Asia-Pacific Conf. Image Process. Electron. Comput., vol. 3, no. 5, p. 1174–1177, 2021.

[83] M. Lyra, S. Lyra, B. Kostakis, S. Drosos, C. Georgosopoulos, and K. Skouroliakou, "Digital Mammography texture analysis by computer assisted image processing," IST 2008 - IEEE Work. Imaging Syst. Tech. Proc, vol. 1, no. 3, p. 73–76, 2008.

[84] L. M. G. Fonseca, L. M. Namikawa, and E. F. Castejon, "Digital image processing in remote sensing," Tutorials of SIBGRAPI 2009 - 22nd Brazilian Symposium on Computer Graphics and Image Processing, vol. 4, no. C, p. 59–71, 2009.

[85] R. Mékle, A. F. Laine, and E. X. Wu, "Combined MR data acquisition of multicontrast images using variable acquisition parameters and K-space data sharing," IEEE Trans. Med. Imaging,, Vols. 22,, no. 7, p. 806–823, 2003.

[86] Z. Kuixing and J. Mei, "Study on digital acquisition method of infantile finger veinlet image," ICME Int. Conf. Complex Med. Eng., vol. 5, no. 6, pp. 421–424,, 2013.

[87] S. I. Workshop and M. Experience, "EFFECTIVE PREPROCESSING STAGE IN THE FOURIER TRANSFORM DOMAIN FOR IMAGE QUALITY ASSESSMENT," Workshop, Sixth International, vol. 5, no. 7, pp. 189-194, 2014.

[88] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," vol. 10, no. 1, pp. 1-22, 2021.

[89] Khatak, Anil, "A review on various filtering techniques in image processing," Natl. J. Multidiscip. Res. Dev, vol. 3, no. 1, pp. 1350-1355, 2018.

[90] ROSLIDAR , AULIA RAHMAN , BISWAJEET PRADHAN , KHAIRUL MUNADI, "A Review on Recent Progress in Thermal Imaging and Deep Learning Approaches for Breast Cancer Detection," IEEE Access, vol. 4, no. 6, pp. 1-20, 2020.

[91] Zhou, S. Kevin, "A Review of Deep Learning in Medical Imaging: Imaging Traits," Proceedings of the IEEE, vol. 109, no. 5, pp. 820-838, 2021.

[92] J. Latif, C. Xiao, A. Imran, and S. Tu, "Medical imaging using machine learning and deep learning algorithms: A review," 2019 2nd Int. Conf. Comput. Math. Eng. Technol. iCoMET 2019, vol. 17, no. 3, p. 1–5, 2019.

[93] N. Singh and S. Singh, "Object classification to analyze medical imaging data using deep learning," Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017, Vols. 2018-Janua, pp. 1-4, 2018.

[94] M. Pandiya, S. Dassani, and P. Mangalraj, "Analysis of Deep Learning Architectures for Object Detection - A Critical Review," Proceedings of 2020 IEEE-HYDCON International Conference on Engineering in the 4th Industrial Revolution, HYDCON 2020, vol. 2, no. 7, 2020.

[95] B. P. Amiruddin and R. E. A. Kadir, "CNN Architectures Performance Evaluation for Image Classification of Mosquito in Indonesia," Proceedings - 2020 International Seminar on Intelligent Technology and Its Application: Humanification of Reliable Intelligent Systems, ISITIA 2020, vol. 2, no. 3, pp. 223- 227, 2020.

[96] Ahmed, Eman, "A survey on Deep Learning Advances on Different 3D Data Representations," vol. 1, no. 1, pp. 1-35, 2018.

[97] Y. Seo and K. S. Shin, "Image classification of fine-grained fashion image based on style using pre-trained convolutional neural network," 2018 IEEE 3rd International Conference on Big Data Analysis, ICBDA 2018, vol. 4, no. 8, pp. 387-390, 2018.

[98] Shaily, Tumun, "Bacterial Image Classification Using Convolutional Neural Networks," 2020 IEEE 17th India Council International Conference, INDICON 2020, vol. 3, no. 9, 2020.

[99] Hakim, Luqman and Sari, Zamah and Handhajani, Handhajani, "Klasifikasi Citra Pigmen Kanker Kulit Menggunakan Convolutional Neural Network," Jurnal RESTI, vol. 5, no. 2, pp. 379--385, 2021.

[100] E. Chen, X. Wu, C. Wang, and Y. Du,, "Application of improved convolutional neural network in image classification," Proceedings - 2019 International Conference on Machine Learning, Big Data and Business Intelligence, MLBDBI 2019, pp. 109-113, 2019.

[101] K. Singh, A. Seth, H. S. Sandhu, and K. Samdani, "A comprehensive review of convolutional neural network based image enhancement techniques,," 2019 IEEE International Conference on System, Computation, Automation and Networking, ICSCAN 2019, vol. 1, no. 1, pp. 1-6, 2019.

[102] T. A. Korzhebin and A. D. Egorov, "Comparison of Combinations of Data Augmentation Methods and Transfer Learning Strategies in Image Classification Used in Convolution Deep Neural Networks," Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2021, vol. 3, no. 5, pp. 479-482, 2021.

[103] A. Titoriya and S. Sachdeva, "Breast Cancer Histopathology Image Classification using AlexNet," 2019 4th Int. Conf. Inf. Syst. Comput. Networks, vol. 3, no. 9, p. 708–712, 2019.

[104] R. Watanabe, N. Soga, and H. Noji, "Novel nano-device to measure voltage-driven membrane transporter activity," IEEE Trans. Nanotechnol, vol. 15, no. 1, p. 70–73, 2016.

[105] S. M. Badawy, A. E. N. A. Mohamed, A. A. Hefnawy, H. E. Zidan, M. T. Gadallah, and G. M. El-Banby, "Classification of Breast Ultrasound Images Based on Convolutional Neural Networks - A Comparative Study," 2021 Int. Telecommun. Conf. ITC-Egypt 2021 - Proc, vol. 4, no. 1, pp. 7-12, 2021.

[106] Kassani, Sara Hosseinzadeh and Kassani, Peyman Hosseinzadeh and Wesolowski, Michal J and Schneider, Kevin A and Deters, Ralph, "Breast Cancer Diagnosis with Transfer Learning and Global Pooling," IEEE, pp. 519--524, 2019.

[107] C. M. Kim, R. C. Park, and E. J. Hong, "Breast mass classification using eLFA algorithm based on CRNN deep learning model," IEEE Access, vol. 8, no. 1, pp. 197312–197323,, 2020.

[108] H. C. Lu, E. W. Loh, and S. C. Huang, "The Classification of Mammogram Using Convolutional Neural Network with Specific Image Preprocessing for Breast Cancer Detection," 2019 2nd International Conference on Artificial Intelligence and Big Data, ICAIBD 2019, vol. 1, no. 4, pp. 9-12, 2019.

[109] M. A. Anupama, V. Sowmya, and K. P. Soman, "Breast cancer classification using capsule network with preprocessed histology images," Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019, vol. 1, no. 2, p. 143–147, 2019.

[110] Chaurasia V, Pal S, Tiwari BB. “Prediction of benign and malignant breast cancer using data mining techniques”. Journal of Algorithms & Computational Technology. 2018; 12(2):119-26. DOI: 10.1177/1748301818756225

[111] Yue W, Wang Z, Chen H, Payne A, Liu X. “Machine learning with applications in breast cancer diagnosis and prognosis”. Designs. 2018; 2(2):13.

[112] Banu B, Thirumalaikolundusubramanian P.” Comparison of Bayes Classifiers for Breast Cancer Classification”. Asian Pacific journal of cancer prevention (APJCP). 2018; 19(10):2917-20.DOI: 10.22034/APJCP.2018.19.10.2917

[113] Akinsola AF, Sokunbi MA, Onadokun IO. “Data Mining For Breast Cancer Classification”. International Journal of Engineering And Computer Science. 2017; 6(8): 22250-22258. DOI: 10.18535/ijecs/v6i8.06

[114] Mirajkar P, Lakshmi P. “Prediction of Cancer Risk in Perspective of Symptoms using Naïve Bayes Classifier”. International Journal of Engineering Research in Computer Science and Engineering. 2017; 4(9):145-149.

[115] Oskouei RJ, Kor NM, Maleki SA. “Data mining and medical world: breast cancers" diagnosis, treatment, prognosis and challenges”. American journal of cancer research. 2017; 7(3):610-27.

[116] Mandal SK. “Performance analysis of data mining algorithms for breast cancer cell detection using Naïve Bayes, logistic regression and decision tree”. International Journal of Engineering and Computer Science. 2017; 6(2):20388-91. DOI: 10.18535/ijecs/v6i2.40

[117] Sumalatha G, Archana S. “A Study on Early Prevention and Detection of Breast Cancer using Data Mining Techniques”. International Journal of Innovative Research in Computer and Communication Engineering. 2017; 5(6):11045-11050. DOI: 10.15680/IJIRCCE.2017.

[118] Devi RD, Devi MI. “Outlier detection algorithm combined with decision tree classifier for early diagnosis of breast cancer”. International Journal of Advanced Engineering Technology.2016; VII (II): 93-98

[119] Padmapriya B, Velmurugan T.” Classification Algorithm Based Analysis of Breast Cancer Data”. International Journal of Data Mining Techniques and Applications. 2016; 5(1):43-49.

[120] Abed BM, et al. “A hybrid classification algorithm approach for breast cancer diagnosis”. In2016 IEEE Industrial Electronics and Applications Conference (IEACon). 2016: 269-274.

[121] Chidambaranathan S. “Breast Cancer Diagnosis Based on Feature Extraction by Hybrid of k-means and extreme learning machine algorithms”. ARPN Journal of Engineering and Applied Sciences. 2016; 11(7):4581-86

[122] Lavanya D, Rani KU. “Ensemble Decision Tree Classifier for Breast Cancer Data”. International Journal of Information Technology Convergence and Services (IJITCS). 2016; 2(1):17-24. DOI: 10.5121/ijitcs.2012.2103

[123] Sivakami K. “Mining big data: Breast cancer prediction using DT-SVM Hybrid model”. International Journal of Scientific Engineering and Applied Science (IJSEAS). 2015; 1(5):418-29.

[124] Zand HK. “A comparative survey on data mining techniques for breast cancer diagnosis and prediction”. Ind. J. Fundam. Appl. Life Sci. 2015; 5 (S1):4330-9.

[125] Majali J, Niranjan R, Phatak V, Tadakhe O. “Data Mining Techniques for Diagnosis and Prognosis of Cancer”. International Journal of Advanced Research in Computer and Communication Engineering. 2015; 4(3):613-16. DOI 10.17148/IJARCCE.2015.43147

[126] Chaurasia V, Pal S. “A Novel Approach for Breast Cancer Detection using Data Mining Techniques”. (IJIRCCE) International Journal of Innovative Research in Computer and Communication Engineering.2014; 2(1): 2456-65

[127] Joshi J, Doshi R, Patel J. “Diagnosis of Breast Cancer Using Clustering Data mining Approach”. International Journal of Computer Applications. 2014; 101(10):13-7.

[128] Sumbaly R, Vishnusri N, Jeyalatha S. “Diagnosis of Breast Cancer Using Decision Tree Data Mining Technique”. International Journal of Computer Applications. 2014; 98(10): 16-24.

[129] Joshi J, Doshi R, Patel J. “Diagnosis and prognosis breast cancer using classification rules”. International Journal of Engineering Research and General Science. 2014; 2(6):315-23.

[130] Chandrasekar RM, Palaniammal V, Phil M. “Performance and Evaluation of Data Mining Techniques in Cancer Diagnosis”. IOSR Journal of Computer Engineering (IOSR-JCE). 2013; 15(5):39-44.

[131] Gupta S, Kumar D, Sharma A. “Data mining classification techniques applied for breast cancer diagnosis and prognosis”. Indian Journal of Computer Science and Engineering (IJCSE). 2011; 2(2):188-95.

[132] Bellaachia A, Guven E. “Predicting breast cancer survivability using data mining techniques”. J Am Aging Assoc. 2006; 58(13): 10-14.

[133]. Wang, Y., Tao, D., Gao, X., Li, X., & Wang, B. (2011). “Mammographic mass segmentation: embedding multiple features in vector-valued level set in ambiguous regions.” Pattern Recognition, 44(9), 1903-1915.

[134]. Yu, L., Chen, H., Dou, Q., Qin, J., & Heng, P. A. (2016). “Automated melanoma recognition in dermoscopy images via very deep residual networks.” IEEE transactions on medical imaging, 36(4), 994- 1004.

[135] M. A. Anupama, V. Sowmya, and K. P. Soman, "Breast cancer classification using capsule network with preprocessed histology images," Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019, vol. 1, no. 2, p. 143–147, 2019.

[136] A. Mikołajczyk and M. Grochowski,, "2019 International Interdisciplinary PhD 60 Workshop," IIPhDW 2019,2019 Int. Interdiscip. PhD Work. IIPhDW 2019, vol. 4, no. 3, p. 117–122, 2019.

[137] F. Ertam, "Data classification with deep learning using tensorflow," 2nd Int. Conf. Comput. Sci. Eng. UBMK 2017, p. 755–758, 2017.

[138] K. Jakhar and N. Hooda, "Big data deep learning framework using keras: A case study of pneumonia prediction," 2018 4th Int. Conf. Comput. Commun. Autom. ICCCA 2018, vol. 6, no. 3, p. 1–5, 2018.

[139] H. Dalianis, "Evaluation Metrics and Evaluation," Clin. Text Min, vol. 1967, no. 2, pp. 45–53, 2018.

[140] Sara Hosseinzadeh Kassani, Peyman Hosseinzadeh Kassani, Michal J. Wesolowski, "Breast Cancer Diagnosis with Transfer Learning and global pooling," in ICTC 2019, 2019.

[141] Hao-Chun Lu, El-Wui Loh ,Shih-Chen Huang, "The Classification of Mammogram Using Convolutional Neural Network with Specific Image Preprocessing for Breast Cancer Detection," in IEEE, 2019.

[142] Alom MdZ, Hasan M, Yakopcic C, Taha TM, Asari VK: Improved inception-residual convolutional neural network for object recognition. arXiv preprint arXiv:1712.09888 2017.

[143] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.

Appendix

University of Limpopo
Department of Research Administration and Development
Private Bag X1106, Sovenga, 0727, South Africa
Tel: (015) 268 3935, Fax: (015) 268 2306, Email: tukiso.sewapa@ul.ac.za

TURFLOOP RESEARCH ETHICS COMMITTEE

ETHICS CLEARANCE CERTIFICATE

MEETING: **26 SEPTEMBER 2023**

PROJECT NUMBER: **TREC/869/2023: PG**

PROJECT:

Title:	Early detection of breast cancer using machine learning.
Researcher:	VR Baloyi
Supervisor:	Prof SN Mokwena
Co-Supervisor/s:	N/A
School:	Mathematical and Computer Sciences
Degree:	Doctor of Philosophy in Computer Science

PROF D MAPOSA
CHAIRPERSON: TURFLOOP RESEARCH ETHICS COMMITTEE

The Turfloop Research Ethics Committee (TREC) is registered with the National Health Research Ethics Council, Registration Number: **REC-0310111-031**

Note:

i) **This Ethics Clearance Certificate will be valid for one (1) year, as from the abovementioned date. Application for annual renewal (or annual review) need to be received by TREC one month before lapse of this period.**

ii) **Should any departure be contemplated from the research procedure as approved, the researcher(s) must re-submit the protocol to the committee, together with the Application for Amendment form.**

iii) **PLEASE QUOTE THE PROTOCOL NUMBER IN ALL ENQUIRIES.**

Finding solutions for Africa

"""Run tests for all models

Tests that run on CI should have a specific marker, e.g. @pytest.mark.base. This

marker is used to parallelize the CI runs, with one runner for each marker.

If new tests are added, ensure that they use one of the existing markers

(documented in pyproject.toml > pytest > markers) or that a new marker is added

for this set of tests. If using a new marker, adjust the test matrix in

.github/workflows/tests.yml to run tests with this new marker, otherwise the

tests will be skipped on CI.

"""

import pytest

import torch

import platform

import os

import fnmatch

_IS_MAC = platform.system() == 'Darwin'

try:

    from torchvision.models.feature_extraction import create_feature_extractor, get_graph_node_names, NodePathTracer

    has_fx_feature_extraction = True

except ImportError:

    has_fx_feature_extraction = False

import timm

from timm import list_models, create_model, set_scriptable, get_pretrained_cfg_value

```
from timm.layers import Format, get_spatial_dim, get_channel_dim

from timm.models import get_notrace_modules, get_notrace_functions

import importlib

import os

torch_backend = os.environ.get('TORCH_BACKEND')

if torch_backend is not None:

    importlib.import_module(torch_backend)

torch_device = os.environ.get('TORCH_DEVICE', 'cpu')

timeout = os.environ.get('TIMEOUT')

timeout120 = int(timeout) if timeout else 120

timeout300 = int(timeout) if timeout else 300

if hasattr(torch._C, '_jit_set_profiling_executor'):

    # legacy executor is too slow to compile large models for unit tests

    # no need for the fusion performance here

    torch._C._jit_set_profiling_executor(True)

    torch._C._jit_set_profiling_mode(False)

# models with forward_intermediates() and support for FeatureGetterNet features_only wrapper

FEAT_INTER_FILTERS = [

    'vision_transformer', 'vision_transformer_sam', 'vision_transformer_hybrid', 'vision_transformer_relpos',

    'beit', 'mvitv2', 'eva', 'cait', 'xcit', 'volo', 'twins', 'deit', 'swin_transformer', 'swin_transformer_v2',
```

```
    'swin_transformer_v2_cr', 'maxxvit', 'efficientnet', 'mobilenetv3', 'levit',
'efficientformer', 'resnet',
    'regnet', 'byobnet', 'byoanet', 'mlp_mixer', 'hiera', 'fastvit',
]
# transformer / hybrid models don't support full set of spatial / feature APIs and/or have
spatial output.
NON_STD_FILTERS = [
    'vit_*', 'tnt_*', 'pit_*', 'coat_*', 'cait_*', '*mixer_*', 'gmlp_*', 'resmlp_*', 'twins_*',
    'convit_*', 'levit*', 'visformer*', 'deit*', 'xcit_*', 'crossvit_*', 'beit*',
    'poolformer_*', 'volo_*', 'sequencer2d_*', 'mvitv2*', 'gcvit*', 'efficientformer*',
    'eva_*', 'flexivit*', 'eva02*', 'samvit_*', 'efficientvit_m*', 'tiny_vit_*', 'hiera_*', 'vitamin*'
]
NUM_NON_STD = len(NON_STD_FILTERS)
# exclude models that cause specific test failures
if 'GITHUB_ACTIONS' in os.environ:
    # GitHub Linux runner is slower and hits memory limits sooner than MacOS, exclude
bigger models
    EXCLUDE_FILTERS = [
        '*efficientnet_l2*', '*resnext101_32x48d', '*in21k', '*152x4_bitm', '*101x3_bitm',
'*50x3_bitm',
        '*nfnet_f3*', '*nfnet_f4*', '*nfnet_f5*', '*nfnet_f6*', '*nfnet_f7*', '*efficientnetv2_xl*',
        '*resnetrs350*', '*resnetrs420*', 'xcit_large_24_p8*', '*huge*', '*giant*', '*gigantic*',
        '*enormous*', 'maxvit_xlarge*', 'regnet*1280', 'regnet*2560']
    NON_STD_EXCLUDE_FILTERS = ['*huge*', '*giant*', '*gigantic*', '*enormous*']
```

```
else:
    EXCLUDE_FILTERS = ['*enormous*']
    NON_STD_EXCLUDE_FILTERS = ['*gigantic*', '*enormous*']
EXCLUDE_JIT_FILTERS = ['hiera_*']
TARGET_FWD_SIZE = MAX_FWD_SIZE = 384
TARGET_BWD_SIZE = 128
MAX_BWD_SIZE = 320
MAX_FWD_OUT_SIZE = 448
TARGET_JIT_SIZE = 128
MAX_JIT_SIZE = 320
TARGET_FFEAT_SIZE = 96
MAX_FFEAT_SIZE = 256
TARGET_FWD_FX_SIZE = 128
MAX_FWD_FX_SIZE = 256
TARGET_BWD_FX_SIZE = 128
MAX_BWD_FX_SIZE = 224
def _get_input_size(model=None, model_name='', target=None):
    if model is None:
        assert model_name, "One of model or model_name must be provided"
        input_size = get_pretrained_cfg_value(model_name, 'input_size')
        fixed_input_size = get_pretrained_cfg_value(model_name, 'fixed_input_size')
        min_input_size = get_pretrained_cfg_value(model_name, 'min_input_size')
    else:
```

```
        default_cfg = model.default_cfg
        input_size = default_cfg['input_size']
        fixed_input_size = default_cfg.get('fixed_input_size', None)
        min_input_size = default_cfg.get('min_input_size', None)
    assert input_size is not None

    if fixed_input_size:
        return input_size
    if min_input_size:
        if target and max(input_size) > target:
            input_size = min_input_size
    else:
        if target and max(input_size) > target:
            input_size = tuple([min(x, target) for x in input_size])
    return input_size
@pytest.mark.base
@pytest.mark.timeout(timeout120)
@pytest.mark.parametrize('model_name',
list_models(exclude_filters=EXCLUDE_FILTERS))
@pytest.mark.parametrize('batch_size', [1])
def test_model_forward(model_name, batch_size):
    """Run a single forward pass with each model"""
    model = create_model(model_name, pretrained=False)
```

```
    model.eval()
    input_size = _get_input_size(model=model, target=TARGET_FWD_SIZE)
    if max(input_size) > MAX_FWD_SIZE:
        pytest.skip("Fixed input size model > limit.")
    inputs = torch.randn((batch_size, *input_size))
    inputs = inputs.to(torch_device)
    model.to(torch_device)
    outputs = model(inputs)
    assert outputs.shape[0] == batch_size
    assert not torch.isnan(outputs).any(), 'Output included NaNs'
@pytest.mark.base
@pytest.mark.timeout(timeout120)
@pytest.mark.parametrize('model_name',
list_models(exclude_filters=EXCLUDE_FILTERS, name_matches_cfg=True))
@pytest.mark.parametrize('batch_size', [2])
def test_model_backward(model_name, batch_size):
    """Run a single forward pass with each model"""
    input_size = _get_input_size(model_name=model_name,
target=TARGET_BWD_SIZE)
    if max(input_size) > MAX_BWD_SIZE:
        pytest.skip("Fixed input size model > limit.")
    model = create_model(model_name, pretrained=False, num_classes=42)
    num_params = sum([x.numel() for x in model.parameters()])
```

```
    model.train()
    inputs = torch.randn((batch_size, *input_size))
    inputs = inputs.to(torch_device)
    model.to(torch_device)
    outputs = model(inputs)
    if isinstance(outputs, tuple):
        outputs = torch.cat(outputs)
    outputs.mean().backward()
    for n, x in model.named_parameters():
        assert x.grad is not None, f'No gradient for {n}'
    num_grad = sum([x.grad.numel() for x in model.parameters() if x.grad is not None])
    assert outputs.shape[-1] == 42
    assert num_params == num_grad, 'Some parameters are missing gradients'
    assert not torch.isnan(outputs).any(), 'Output included NaNs'
# models with extra conv/linear layers after pooling
EARLY_POOL_MODELS = (
    timm.models.EfficientVit,
    timm.models.EfficientVitLarge,
    timm.models.HighPerfGpuNet,
    timm.models.GhostNet,
    timm.models.MetaNeXt, # InceptionNeXt
    timm.models.MobileNetV3,
    timm.models.RepGhostNet,
```

```
    timm.models.VGG,
)
@pytest.mark.cfg
@pytest.mark.timeout(timeout300)
@pytest.mark.parametrize('model_name', list_models(
    exclude_filters=EXCLUDE_FILTERS + NON_STD_FILTERS, include_tags=True))
@pytest.mark.parametrize('batch_size', [1])
def test_model_default_cfgs(model_name, batch_size):
    """Run a single forward pass with each model"""
    model = create_model(model_name, pretrained=False)
    model.eval()
    model.to(torch_device)
    assert getattr(model, 'num_classes') >= 0
    assert getattr(model, 'num_features') > 0
    assert getattr(model, 'head_hidden_size') > 0
    state_dict = model.state_dict()
    cfg = model.default_cfg
    pool_size = cfg['pool_size']
    input_size = model.default_cfg['input_size']
    output_fmt = getattr(model, 'output_fmt', 'NCHW')
    spatial_axis = get_spatial_dim(output_fmt)
    assert len(spatial_axis) == 2  # TODO add 1D sequence support
    feat_axis = get_channel_dim(output_fmt)
```

```
    if all([x <= MAX_FWD_OUT_SIZE for x in input_size]) and \
            not any([fnmatch.fnmatch(model_name, x) for x in EXCLUDE_FILTERS]):
        # output sizes only checked if default res <= 448 * 448 to keep resource down
        input_size = tuple([min(x, MAX_FWD_OUT_SIZE) for x in input_size])
        input_tensor = torch.randn((batch_size, *input_size), device=torch_device)
        # test forward_features (always unpooled) & forward_head w/ pre_logits
        outputs = model.forward_features(input_tensor)
        outputs_pre = model.forward_head(outputs, pre_logits=True)
        assert outputs.shape[spatial_axis[0]] == pool_size[0], f'unpooled feature shape {outputs.shape} != config'
        assert outputs.shape[spatial_axis[1]] == pool_size[1], f'unpooled feature shape {outputs.shape} != config'
        assert outputs.shape[feat_axis] == model.num_features, f'unpooled feature dim {outputs.shape[feat_axis]} != model.num_features {model.num_features}'
        assert outputs_pre.shape[1] == model.head_hidden_size, f'pre_logits feature dim {outputs_pre.shape[1]} != model.head_hidden_size {model.head_hidden_size}'
        # test forward after deleting the classifier, output should be poooled, size(-1) == model.num_features
        model.reset_classifier(0)
        model.to(torch_device)
        outputs = model.forward(input_tensor)
        assert len(outputs.shape) == 2
        assert outputs.shape[1] == model.head_hidden_size, f'feature dim w/ removed classifier {outputs.shape[1]} != model.head_hidden_size {model.head_hidden_size}'
```

```
        assert outputs.shape == outputs_pre.shape, f'output shape of pre_logits {outputs_pre.shape} does not match reset_head(0) {outputs.shape}'

        # test model forward after removing pooling and classifier

        if not isinstance(model, EARLY_POOL_MODELS):

            model.reset_classifier(0, '')  # reset classifier and disable global pooling

            model.to(torch_device)

            outputs = model.forward(input_tensor)

            assert len(outputs.shape) == 4

            assert outputs.shape[spatial_axis[0]] == pool_size[0] and outputs.shape[spatial_axis[1]] == pool_size[1]

        # test classifier + global pool deletion via __init__

        if 'pruned' not in model_name and not isinstance(model, EARLY_POOL_MODELS):

            model = create_model(model_name, pretrained=False, num_classes=0, global_pool='').eval()

            model.to(torch_device)

            outputs = model.forward(input_tensor)

            assert len(outputs.shape) == 4

            assert outputs.shape[spatial_axis[0]] == pool_size[0] and outputs.shape[spatial_axis[1]] == pool_size[1]

    # check classifier name matches default_cfg

    if cfg.get('num_classes', None):

        classifier = cfg['classifier']

        if not isinstance(classifier, (tuple, list)):
```

```
            classifier = classifier,

        for c in classifier:

            assert c + ".weight" in state_dict.keys(), f'{c} not in model params'

    # check first conv(s) names match default_cfg

    first_conv = cfg['first_conv']

    if isinstance(first_conv, str):

        first_conv = (first_conv,)

    assert isinstance(first_conv, (tuple, list))

    for fc in first_conv:

        assert fc + ".weight" in state_dict.keys(), f'{fc} not in model params'

@pytest.mark.cfg

@pytest.mark.timeout(timeout300)

@pytest.mark.parametrize('model_name',      list_models(filter=NON_STD_FILTERS,
exclude_filters=NON_STD_EXCLUDE_FILTERS, include_tags=True))

@pytest.mark.parametrize('batch_size', [1])

def test_model_default_cfgs_non_std(model_name, batch_size):

    """Run a single forward pass with each model"""

    model = create_model(model_name, pretrained=False)

    model.eval()

    model.to(torch_device)

    assert getattr(model, 'num_classes') >= 0

    assert getattr(model, 'num_features') > 0

    assert getattr(model, 'head_hidden_size') > 0
```

```
    state_dict = model.state_dict()
    cfg = model.default_cfg

    input_size = _get_input_size(model=model)
    if max(input_size) > 320:  # FIXME const
        pytest.skip("Fixed input size model > limit.")
    input_tensor = torch.randn((batch_size, *input_size), device=torch_device)
    feat_dim = getattr(model, 'feature_dim', None)
    outputs = model.forward_features(input_tensor)
    outputs_pre = model.forward_head(outputs, pre_logits=True)
    if isinstance(outputs, (tuple, list)):
        # cannot currently verify multi-tensor output.
        pass
    else:
        if feat_dim is None:
            feat_dim = -1 if outputs.ndim == 3 else 1
        assert outputs.shape[feat_dim] == model.num_features
        assert outputs_pre.shape[1] == model.head_hidden_size
    # test forward after deleting the classifier, output should be poooled, size(-1) == model.num_features
    model.reset_classifier(0)
    model.to(torch_device)
    outputs = model.forward(input_tensor)
```

```
    if isinstance(outputs,  (tuple, list)):
        outputs = outputs[0]
    if feat_dim is None:
        feat_dim = -1 if outputs.ndim == 3 else 1
    assert outputs.shape[feat_dim] == model.head_hidden_size, 'pooled num_features
!= config'
    assert outputs.shape == outputs_pre.shape
    model = create_model(model_name, pretrained=False, num_classes=0).eval()
    model.to(torch_device)
    outputs = model.forward(input_tensor)
    if isinstance(outputs, (tuple, list)):
        outputs = outputs[0]
    if feat_dim is None:
        feat_dim = -1 if outputs.ndim == 3 else 1
    assert outputs.shape[feat_dim] == model.num_features
    # check classifier name matches default_cfg
    if cfg.get('num_classes', None):
        classifier = cfg['classifier']
        if not isinstance(classifier, (tuple, list)):
            classifier = classifier,
        for c in classifier:
            assert c + ".weight" in state_dict.keys(), f'{c} not in model params'
    # check first conv(s) names match default_cfg
```

```
    first_conv = cfg['first_conv']
    if isinstance(first_conv, str):
        first_conv = (first_conv,)
    assert isinstance(first_conv, (tuple, list))
    for fc in first_conv:
        assert fc + ".weight" in state_dict.keys(), f'{fc} not in model params'
if 'GITHUB_ACTIONS' not in os.environ:
    @pytest.mark.timeout(240)
    @pytest.mark.parametrize('model_name', list_models(pretrained=True))
    @pytest.mark.parametrize('batch_size', [1])
    def test_model_load_pretrained(model_name, batch_size):
        """Create that pretrained weights load, verify support for in_chans != 3 while doing so."""
        in_chans = 3 if 'pruned' in model_name else 1  # pruning not currently supported with in_chans change
        create_model(model_name, pretrained=True, in_chans=in_chans, num_classes=5)
        create_model(model_name, pretrained=True, in_chans=in_chans, num_classes=0)
    @pytest.mark.timeout(240)
    @pytest.mark.parametrize('model_name', list_models(pretrained=True, exclude_filters=NON_STD_FILTERS))
    @pytest.mark.parametrize('batch_size', [1])
    def test_model_features_pretrained(model_name, batch_size):
```

```
        """Create that pretrained weights load when features_only==True."""
        create_model(model_name, pretrained=True, features_only=True)
@pytest.mark.torchscript
@pytest.mark.timeout(timeout120)
@pytest.mark.parametrize(
    'model_name', list_models(exclude_filters=EXCLUDE_FILTERS + EXCLUDE_JIT_FILTERS, name_matches_cfg=True))
@pytest.mark.parametrize('batch_size', [1])
def test_model_forward_torchscript(model_name, batch_size):
    """Run a single forward pass with each model"""
    input_size = _get_input_size(model_name=model_name, target=TARGET_JIT_SIZE)
    if max(input_size) > MAX_JIT_SIZE:
        pytest.skip("Fixed input size model > limit.")
    with set_scriptable(True):
        model = create_model(model_name, pretrained=False)
    model.eval()
    model = torch.jit.script(model)
    model.to(torch_device)
    outputs = model(torch.randn((batch_size, *input_size)))
    assert outputs.shape[0] == batch_size
    assert not torch.isnan(outputs).any(), 'Output included NaNs'
EXCLUDE_FEAT_FILTERS = [
```

```
    '*pruned*',  # hopefully fix at some point
] + NON_STD_FILTERS

if 'GITHUB_ACTIONS' in os.environ:  # and 'Linux' in platform.system():
    # GitHub Linux runner is slower and hits memory limits sooner than MacOS, exclude bigger models
    EXCLUDE_FEAT_FILTERS += ['*resnext101_32x32d', '*resnext101_32x16d']

@pytest.mark.features
@pytest.mark.timeout(120)
@pytest.mark.parametrize('model_name', list_models(exclude_filters=EXCLUDE_FILTERS + EXCLUDE_FEAT_FILTERS))
@pytest.mark.parametrize('batch_size', [1])
def test_model_forward_features(model_name, batch_size):
    """Run a single forward pass with each model in feature extraction mode"""
    model = create_model(model_name, pretrained=False, features_only=True)
    model.eval()
    expected_channels = model.feature_info.channels()
    expected_reduction = model.feature_info.reduction()
    assert len(expected_channels) >= 3  # all models here should have at least 3 default feat levels
    input_size = _get_input_size(model=model, target=TARGET_FFEAT_SIZE)
    if max(input_size) > MAX_FFEAT_SIZE:
        pytest.skip("Fixed input size model > limit.")
    output_fmt = getattr(model, 'output_fmt', 'NCHW')
    feat_axis = get_channel_dim(output_fmt)
```

```
    spatial_axis = get_spatial_dim(output_fmt)
    import math
    outputs = model(torch.randn((batch_size, *input_size)))
    assert len(expected_channels) == len(outputs)
    spatial_size = input_size[-2:]
    for e, r, o in zip(expected_channels, expected_reduction, outputs):
        assert e == o.shape[feat_axis]
        assert o.shape[spatial_axis[0]] <= math.ceil(spatial_size[0] / r) + 1
        assert o.shape[spatial_axis[1]] <= math.ceil(spatial_size[1] / r) + 1
        assert o.shape[0] == batch_size
        assert not torch.isnan(o).any()
@pytest.mark.features
@pytest.mark.timeout(120)
@pytest.mark.parametrize('model_name',
list_models(module=FEAT_INTER_FILTERS,    exclude_filters=EXCLUDE_FILTERS
+ ['*pruned*']))
@pytest.mark.parametrize('batch_size', [1])
def test_model_forward_intermediates_features(model_name, batch_size):
    """Run a single forward pass with each model in feature extraction mode"""
    model = create_model(model_name, pretrained=False, features_only=True,
feature_cls='getter')
    model.eval()
    expected_channels = model.feature_info.channels()
    expected_reduction = model.feature_info.reduction()
```

```
    input_size = _get_input_size(model=model, target=TARGET_FFEAT_SIZE)
    if max(input_size) > MAX_FFEAT_SIZE:
        pytest.skip("Fixed input size model > limit.")
    output_fmt = getattr(model, 'output_fmt', 'NCHW')
    feat_axis = get_channel_dim(output_fmt)
    spatial_axis = get_spatial_dim(output_fmt)
    import math
    outputs = model(torch.randn((batch_size, *input_size)))
    assert len(expected_channels) == len(outputs)
    spatial_size = input_size[-2:]
    for e, r, o in zip(expected_channels, expected_reduction, outputs):
        print(o.shape)
        assert e == o.shape[feat_axis]
        assert o.shape[spatial_axis[0]] <= math.ceil(spatial_size[0] / r) + 1
        assert o.shape[spatial_axis[1]] <= math.ceil(spatial_size[1] / r) + 1
        assert o.shape[0] == batch_size
        assert not torch.isnan(o).any()
@pytest.mark.features
@pytest.mark.timeout(120)
@pytest.mark.parametrize('model_name',
list_models(module=FEAT_INTER_FILTERS,    exclude_filters=EXCLUDE_FILTERS
+ ['*pruned*']))
@pytest.mark.parametrize('batch_size', [1])
```

```
def test_model_forward_intermediates(model_name, batch_size):
    """Run a single forward pass with each model in feature extraction mode"""
    model = create_model(model_name, pretrained=False)
    model.eval()
    feature_info = timm.models.FeatureInfo(model.feature_info, len(model.feature_info))
    expected_channels = feature_info.channels()
    expected_reduction = feature_info.reduction()
    assert len(expected_channels) >= 3  # all models here should have at least 3 feature levels
    input_size = _get_input_size(model=model, target=TARGET_FFEAT_SIZE)
    if max(input_size) > MAX_FFEAT_SIZE:
        pytest.skip("Fixed input size model > limit.")
    output_fmt = 'NCHW'  # NOTE output_fmt determined by forward_intermediates() arg, not model attribute
    feat_axis = get_channel_dim(output_fmt)
    spatial_axis = get_spatial_dim(output_fmt)
    import math
    output, intermediates = model.forward_intermediates(
        torch.randn((batch_size, *input_size)),
        output_fmt=output_fmt,
    )
    assert len(expected_channels) == len(intermediates)
    spatial_size = input_size[-2:]
```

```
    for e, r, o in zip(expected_channels, expected_reduction, intermediates):
        assert e == o.shape[feat_axis]
        assert o.shape[spatial_axis[0]] <= math.ceil(spatial_size[0] / r) + 1
        assert o.shape[spatial_axis[1]] <= math.ceil(spatial_size[1] / r) + 1
        assert o.shape[0] == batch_size
        assert not torch.isnan(o).any()
def _create_fx_model(model, train=False):
    # This block of code does a bit of juggling to handle any case where there are multiple outputs in train mode
    # So we trace once and look at the graph, and get the indices of the nodes that lead into the original fx output
    # node. Then we use those indices to select from train_nodes returned by torchvision get_graph_node_names
    tracer_kwargs = dict(
        leaf_modules=get_notrace_modules(),
        autowrap_functions=get_notrace_functions(),
        #enable_cpatching=True,
        param_shapes_constant=True
    )
    train_nodes, eval_nodes = get_graph_node_names(model, tracer_kwargs=tracer_kwargs)
    eval_return_nodes = [eval_nodes[-1]]
    train_return_nodes = [train_nodes[-1]]
    if train:
```

```
        tracer = NodePathTracer(**tracer_kwargs)
        graph = tracer.trace(model)
        graph_nodes = list(reversed(graph.nodes))
        output_node_names = [n.name for n in graph_nodes[0]._input_nodes.keys()]
        graph_node_names = [n.name for n in graph_nodes]
        output_node_indices = [-graph_node_names.index(node_name) for node_name
in output_node_names]
        train_return_nodes = [train_nodes[ix] for ix in output_node_indices]
    fx_model = create_feature_extractor(
        model,
        train_return_nodes=train_return_nodes,
        eval_return_nodes=eval_return_nodes,
        tracer_kwargs=tracer_kwargs,
    )
    return fx_model
EXCLUDE_FX_FILTERS = ['vit_gi*', 'hiera*']
# not enough memory to run fx on more models than other tests
if 'GITHUB_ACTIONS' in os.environ:
    EXCLUDE_FX_FILTERS += [
        'beit_large*',
        'mixer_l*',
        '*nfnet_f2*',
        '*resnext101_32x32d',
```

```
        'resnetv2_152x2*',
        'resmlp_big*',
        'resnetrs270',
        'swin_large*',
        'vgg*',
        'vit_large*',
        'vit_base_patch8*',
        'xcit_large*',
    ]

@pytest.mark.fxforward
@pytest.mark.timeout(120)
@pytest.mark.parametrize('model_name',
list_models(exclude_filters=EXCLUDE_FILTERS + EXCLUDE_FX_FILTERS))
@pytest.mark.parametrize('batch_size', [1])
def test_model_forward_fx(model_name, batch_size):
    """
    Symbolically trace each model and run single forward pass through the resulting
GraphModule
    Also check that the output of a forward pass through the GraphModule is the same
as that from the original Module
    """
    if not has_fx_feature_extraction:
        pytest.skip("Can't test FX. Torch >= 1.10 and Torchvision >= 0.11 are required.")
    model = create_model(model_name, pretrained=False)
```

```
    model.eval()
    input_size = _get_input_size(model=model, target=TARGET_FWD_FX_SIZE)
    if max(input_size) > MAX_FWD_FX_SIZE:
        pytest.skip("Fixed input size model > limit.")
    with torch.no_grad():
        inputs = torch.randn((batch_size, *input_size))
        outputs = model(inputs)
        if isinstance(outputs, tuple):
            outputs = torch.cat(outputs)
        model = _create_fx_model(model)
        fx_outputs = tuple(model(inputs).values())
        if isinstance(fx_outputs, tuple):
            fx_outputs = torch.cat(fx_outputs)
    assert torch.all(fx_outputs == outputs)
    assert outputs.shape[0] == batch_size
    assert not torch.isnan(outputs).any(), 'Output included NaNs'
@pytest.mark.fxbackward
@pytest.mark.timeout(120)
@pytest.mark.parametrize('model_name', list_models(
    exclude_filters=EXCLUDE_FILTERS + EXCLUDE_FX_FILTERS,
name_matches_cfg=True))
@pytest.mark.parametrize('batch_size', [2])
def test_model_backward_fx(model_name, batch_size):
```

```
    """Symbolically trace each model and run single backward pass through the resulting GraphModule"""
    if not has_fx_feature_extraction:
        pytest.skip("Can't test FX. Torch >= 1.10 and Torchvision >= 0.11 are required.")
    input_size = _get_input_size(model_name=model_name, target=TARGET_BWD_FX_SIZE)
    if max(input_size) > MAX_BWD_FX_SIZE:
        pytest.skip("Fixed input size model > limit.")
    model = create_model(model_name, pretrained=False, num_classes=42)
    model.train()
    num_params = sum([x.numel() for x in model.parameters()])
    if 'GITHUB_ACTIONS' in os.environ and num_params > 100e6:
        pytest.skip("Skipping FX backward test on model with more than 100M params.")
    model = _create_fx_model(model, train=True)
    outputs = tuple(model(torch.randn((batch_size, *input_size))).values())
    if isinstance(outputs, tuple):
        outputs = torch.cat(outputs)
    outputs.mean().backward()
    for n, x in model.named_parameters():
        assert x.grad is not None, f'No gradient for {n}'
    num_grad = sum([x.grad.numel() for x in model.parameters() if x.grad is not None])
    assert outputs.shape[-1] == 42
    assert num_params == num_grad, 'Some parameters are missing gradients'
```

```
    assert not torch.isnan(outputs).any(), 'Output included NaNs'

if 'GITHUB_ACTIONS' not in os.environ:

    # FIXME this test is causing GitHub actions to run out of RAM and abruptly kill the test process

    # reason: model is scripted after fx tracing, but beit has torch.jit.is_scripting() control flow

    EXCLUDE_FX_JIT_FILTERS = [

        'deit_*_distilled_patch16_224',

        'levit*',

        'pit_*_distilled_224',

    ] + EXCLUDE_FX_FILTERS

    @pytest.mark.timeout(120)

    @pytest.mark.parametrize(

        'model_name', list_models(

            exclude_filters=EXCLUDE_FILTERS + EXCLUDE_JIT_FILTERS + EXCLUDE_FX_JIT_FILTERS, name_matches_cfg=True))

    @pytest.mark.parametrize('batch_size', [1])

    def test_model_forward_fx_torchscript(model_name, batch_size):

        """Symbolically trace each model, script it, and run single forward pass"""

        if not has_fx_feature_extraction:

            pytest.skip("Can't test FX. Torch >= 1.10 and Torchvision >= 0.11 are required.")

        input_size = _get_input_size(model_name=model_name, target=TARGET_JIT_SIZE)
```

```
if max(input_size) > MAX_JIT_SIZE:
    pytest.skip("Fixed input size model > limit.")
with set_scriptable(True):
    model = create_model(model_name, pretrained=False)
model.eval()
model = torch.jit.script(_create_fx_model(model))
with torch.no_grad():
    outputs = tuple(model(torch.randn((batch_size, *input_size))).values())
    if isinstance(outputs, tuple):
        outputs = torch.cat(outputs)
assert outputs.shape[0] == batch_size
assert not torch.isnan(outputs).any(), 'Output included NaNs'
```