Master Thesis

The Effects of Data Augmentation and Synthetic Data in Breast Cancer Detection

Author: Renske Dijkstra

First supervisor:	Karine da Silva Mias de Araujo
Second reader:	Sandjai Bhulai
Company supervisor:	Walid Alaoui Mdaghri

September 5, 2024

Abstract

This thesis studies the effects of augmented and synthetic data in detecting breast cancer using Convolutional Neural Networks (CNNs). Breast cancer is a significant health concern, yearly approximately 2.3 million new cases are reported worldwide and 670.000 women die of breast cancer every year. During routine scanning, approximately 25% of breast cancer cases are missed. Despite the many research and high expectations surrounding the use of convolutional neural networks as detection methods, their effectiveness is often limited by the availability of large, well-curated datasets. This study researches the use of data augmentation techniques and Generative Adversarial Networks (GANs) to generate synthetic data, to enhance the performance of CNNs in breast cancer detection. The study is structured around four subquestions to first study the performance of various CNN architectures, then the effects of data augmentation, the effects of synthetic data, and finally the robustness of the models. EfficientNet achieved the highest performance of several CNN architectures, with an F1-score and balanced accuracy of 0.95 and 0.93 respectively. The EfficientNet model was further used as a baseline model to evaluate the effects of augmented and synthetic data. The addition of augmentation and synthetic data yielded varying results. Adding moderate amounts of augmented and synthetic data slightly improved performance, but further increases did not lead to additional improvements. Additionally, to evaluate the generalization of the model, an adversarial attack was performed. This showed that the use of data augmentation and synthetic data can lead to a significant improvement in terms of model robustness. However, careful consideration of the quantity, quality, and diversity of the data is crucial for the development of effective and robust models. Future research should continue to explore and improve these techniques, with a focus on mitigating overfitting and enhancing model robustness. The study also highlights the challenges in training and evaluating CNN models for medical image analysis, including dataset quality, GAN stability, and variability in external datasets. These findings contribute to the ongoing efforts to improve early breast cancer detection, ultimately contributing to better patient outcomes.

Acknowledgements

This thesis has been written for the completion of the Master of Science degree in Business Analytics at the Vrije Universiteit (VU) Amsterdam. The research for this thesis was conducted during an internship at Accenture in the Data Management team.

I would like to express my gratitude to everyone who has supported and guided me throughout this research project. First, I would like to thank the company supervisors from Accenture, Walid Alaoui Mdaghri and Liridon Hajdini for their guidance, expertise and advice throughout my master's project. I would also like to thank Karine da Silva Miras de Araujo from Vrije Universiteit Amsterdam for her support as my supervisor. Additionally, I extend my appreciation to Sandjai Bhulai, the second reader from Vrije Universiteit. Finally, I would like to thank all other colleagues at Accenture for their assistance, expertise, and creating a positive work atmosphere during my internship.

Contents

List of Abbreviations

Adam	Adaptive Momentum Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Network
BCE	Binary Cross Entropy Loss
BI-RADS	Breast Imaging Reporting and Data System
BP	Backwarc Propagation
CAD	Computer Aided Diagnosis
CBIS-DDSM	Curated Breast Imaging Subset of Digital Database for Screening Mammography
CCE	Categorical Cross Entropy Loss
cGAN	Conditional Generative Adversarial Network
DCNN	Deep Convolutional Network
DL	Deep Learning
DDSM	Digital Database for Screening Mammography
FDA	Food and Drug Administration
FGSM	Fast Gradient Sign Method
GAN	Generative Adversarial Network
KNN	K-Nearest Neighbours
MIAS	Mammographic Image Analysis Society
MBConv	Mobile Inverted Bottleneck Convolution
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
ReLu	Rectified Linear Unit
ResNet	Residual Network
ROI	Region of Interest
SEER	Surveillance, Epidemiology, and End Results
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TL	Transfer Learning
VGG	Visual Geometry Group

List of Figures

List of Tables

Chapter 1

Introduction

Breast cancer remains a leading global health concern, affecting millions of women every year. With approximately one in seven women facing a breast cancer diagnosis at some point in their lives, it is the second most common cancer for women worldwide [12]. Early detection is crucial as the disease can progress rapidly, decreasing survival rates. Despite advancements in medical technology and treatment, many lives are still lost to breast cancer every year. To address this, many countries have established screening programs to detect the disease at an early stage. These programs have been effective in saving lives, however, during routine screening 25% of breast cancer cases are missed [13]. This issue highlights the difficulties in accurately interpreting mammographic images and the need for improved diagnostic methods.

Recent research has shown that artificial intelligence (AI), particularly convolutional neural networks (CNNs), holds significant promise for improving mammogram-based breast cancer detection. Computer Aided Diagnosis (CAD) systems are systems designed to assist radiologists in classifying mammograms. Studies have demonstrated that CNNs can achieve high accuracy in detecting cancerous lesions when implemented in CAD systems [14]. However, despite these promising results in research settings, these advancements have not yet translated into clinical practice. Challenges such as limited data availability and the large variability in real-world cases contribute to this gap.

Problem Statement

The primary challenge in developing effective CNN models for breast cancer detection is the scarcity of large, well-annotated datasets. In medical imaging, obtaining such extensive datasets is often impractical due to privacy regulations [15], the high cost of annotation by experts, and the imbalance in class distributions. Obtaining large well-curated datasets takes a lot of time and collaborative efforts by many stakeholders. These challenges result in the exploration of techniques to augment the available data and address class imbalance, thereby enhancing the performance of CNNs in detecting breast cancer.

Proposed Solution

This research aims to address these issues by exploring innovative approaches to improve the performance and applicability of CNNs in breast cancer detection. By investigating techniques such as data augmentation and synthetic data generation, this study seeks to enhance the accuracy and reliability of mammographic analysis, ultimately contributing to better diagnostic tools and patient outcomes.

Data augmentation is a technique that is used to artificially expand the size of a training data set by generating modified versions of existing data [16]. Common augmentation techniques include rotations, flips, translations, and the addition of noise. These transformations help improve the robustness and generalization ability of CNN models by exposing them to a variety of image variations. Although data augmentation can mitigate overfitting and improve performance, its effectiveness is limited by the diversity and quality of the original dataset [16]. An alternative approach to augmenting training data is the generation of synthetic data using Generative Adversarial Networks (GANs) [17]. GANs are a type of neural network that generates realistic images by learning the underlying distribution of the training data. Conditional GANs (cGANs), an extension of GANs, allow the generation of images conditioned on specific labels, making them suitable for creating synthetic mammographic images with accurate labels. The potential of cGANs to generate high-quality synthetic data could address the limitations of traditional augmentation techniques. cGANs can provide more diversity and a larger amount of training data.

The purpose of this study is to investigate the impact of data augmentation and synthetic data generated by GANs on the performance and generalisability of CNNs in detecting breast cancer from mammographic images. The primary research question addressed is:

> *How does the use of augmented and synthetic data impact the effectiveness of convolutional neural networks in detecting breast cancer from mammograms?*

To address this question, the following subquestions are formulated regarding breast cancer detection:

1. What are the performances of different CNN architectures in detecting breast cancer from mammographic images?

2. How do traditional data augmentation techniques affect the performance of CNN models?

3. How does the inclusion of synthetic data generated by GANs impact the performance of CNN models?

4. How does the inclusion of augmented and synthetic images impact the generalization ability of CNN models?

By exploring these subquestions, the study aims to evaluate the performance of CNN architectures, data augmentation techniques and the use of synthetic data, ultimately providing insights to answer the main research question.

Structure of the Thesis

The structure of the thesis is as follows: Chapter 2 provides background information, discussing breast cancer, detection methods, the current status of Computer-Aided Detection systems, and the challenges of acquiring data sets for breast cancer detection through a literature review. Chapter 3 describes the dataset used in this study, including its source, characteristics, and preprocessing steps. Chapter 4 discusses neural networks and Convolutional Neural Networks, covering data augmentation techniques and Generative Adversarial Networks. Chapter 5 details the methodology, including CNN architectures, data preprocessing steps, augmentation techniques, and the implementation of GAN used in this study. Chapter 6 presents the experimental results and analysis, comparing the performance of various CNN models with and without augmentation and synthetic data, and discusses the findings. Finally, chapter 7 provides the discussion and conclusion, where also the limitations, and potential future research directions are discussed.

Chapter 2

Background

This chapter provides an overview of breast cancer, including its stages, survival rates, and methods of detection. It also explores the history and current status of Computer-Aided Diagnosis (CAD) systems. The chapter reviews relevant literature and discusses the challenges facing these technologies, including privacy regulations, data annotation, and potential biases.

2.1 Breast cancer

Breast cancer is a large global health concern. In 2022, around 2.3 million new cases were reported, with 670.000 deaths globally [18]. It affects approximately one in seven women at some point in their lives, making it the second most common cancer worldwide and the number one cancer among women globally [19].

2.1.1 Stages of Breast Cancer

Breast cancer is typically classified into stages based on the severity of the illness. The staging system most often used is the TNM system, which considers the size of the tumour (T), the involvement of lymph nodes (N) and the occurrence of distant metastases (M). Figure 2.1 shows the different stages of breast cancer. Stage 0 is the earliest stage of breast cancer, in this stage, abnormal cells are found in the milk ducts. This is a non-invasive cancer with no indication of cancerous, or abnormal cells in the surrounding tissue [20]. This stage is highly treatable with surgical and radiation therapy. Invasive breast cancer, stages I to III, indicates that the cancer has spread beyond the ducts or lobules into surrounding breast tissue. Stages I and II mean there is a localised invasive disease, while stage III stands for a locally advanced cancer that may have spread to nearby lymph nodes [20]. The treatment of invasive breast cancer involves a combination of surgery, chemotherapy, and radiation therapy. Stage IV, also known as metastatic breast cancer, is the most advanced stage of cancer. In this stage, cancer cells have spread to other organs such as the lungs, liver, bones, or brain. This stage of cancer is often not curable, the focus is on controlling the disease and improving quality of life through treatments such as chemotherapy, hormonal therapy, targeted therapy, and supportive care [20]. While metastatic breast cancer is not curable, advances in treatment have led to improved survival rates and quality of life for many patients.

2.1.2 Survival rate

The survival rates for cancer patients vary according to the stage of the disease at the time of diagnosis. People who are diagnosed with non-invasive cancer have higher survival rates than those diagnosed with advanced cancers. The American Cancer Society uses the Surveillance, Epidemiology, and End Results (SEER) database, maintained by the National Cancer Institute (NCI), to provide survival statistics for different types of cancer. The SEER database tracks 5-year relative survival rates for breast cancer in the United States, based on how far the cancer has spread [21]. The relative survival rate compares women with a certain stage of breast cancer to women in the overall population. For example, if the 5-year relative survival rate for a specific stage of breast cancer is 90%, it means that women who have

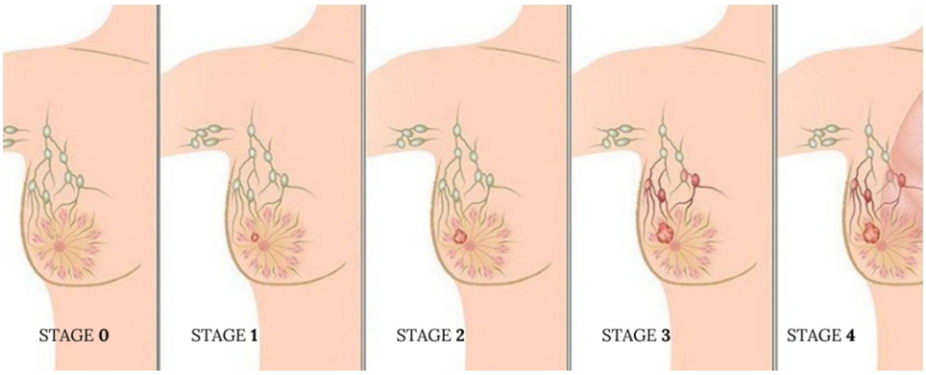


Figure 2.1: The stages of breast cancer [1]

that cancer are, on average, about 90% as likely as women who don't have that cancer to live for at least 5 years after being diagnosed [21]. The cancer is grouped in three different stages; localized (stage 0), regional (stage I, II, and III) and distant (stage IV). The survival rates can be found in table 2.1.

Table 2.1: 5-year relative survival rates for breast cancer

Stage	**5-year Relative Survival Rate**
Localized	99%
Regional	86%
Distant	31%

Early detection of breast cancer is crucial to increase the chances of successful treatment, especially when the disease is in its early stages and survival rates are highest.

2.1.3 Breast cancer detection

Given the importance of early detection, numerous countries, including the Netherlands, have implemented breast cancer screening programs. In the Netherlands, the Population-Based Breast Cancer Screening Program invites women aged 50 to 75 years to be screened every two years [13]. The goal of this screening is to detect breast cancer before the patient experiences any symptoms. This early detection can prevent the cancer from spreading, this increases the likelihood of survival. Every year, approximately 7000 women in the Netherlands are diagnosed with a tumour through the screening program, this results in 1300 fewer deaths from breast cancer each year [13].

Screening programs rely on the use of mammography to detect breast cancer. A mammography screening exam usually involves two or more X-ray images of each breast, these images are referred to as mammograms. Mammograms serve two purposes: screening and diagnosis. Screening mammograms are used to screen asymptomatic women for breast cancer, whereas diagnostic mammograms are used when symptoms or abnormalities are present. Diagnostic mammograms may also be conducted to evaluate changes identified during a screening mammogram or in special circumstances, such as when breast implants are present.

While mammography is a crucial tool for detecting breast cancer, it is not without limitations. Approximately 25% of breast cancers may be missed during routine screenings [13]. These missed cases are referred to as false negatives. False negatives may occur due to dense breast tissue or a small tumour size, this makes the tumour hard to detect. False negatives in breast cancer screening present significant risks, primarily due to missed opportunities for early detection and treatment. A result of delayed diagnosis from false negatives is that the disease is allowed to progress further. This can result in more tumour growth, an increased likelihood of the spreading of cancer, and a reduction in treatment options. This results in poorer treatment outcomes and decreased survival rates. Furthermore, false negatives can give a false sense of security, which may lead patients to delay seeking further evaluation or treatment until symptoms get worse. False negatives also lead to financial burdens, this is caused by the need for more extensive and costly treatments for advanced-stage cancer. To address these risks, ongoing efforts are required to improve the sensitivity and accuracy of screening methods, enhance physician awareness and education, and ensure timely access to appropriate healthcare services for all individuals at risk of breast cancer. Early detection remains essential in reducing the burden of breast cancer and improving patient outcomes.

For every 1000 women screened, around 100 may be called back for further tests because of suspicious findings, of these, only 2-7 cases are typically confirmed as cancer [13]. False positive results cause many problems, they can lead to unnecessary invasive procedures such as biopsies, causing discomfort and potential complications. False positives can also trigger emotional distress, affecting mental well-being long after the results are clarified. Another issue that arises is overdiagnosis. Overdiagnosis refers to situations where benign lesions are discovered, but this detection is not significant as these lesions would not have led to any symptoms or issues if they had not been diagnosed [22]. Overdiagnosis of benign lesions can result in unnecessary treatments, exposing patients to risks without benefits [23]. There can be a financial burden for both individuals and healthcare systems due to follow-up tests and treatment. High rates of false positives strain healthcare resources and can undermine confidence in screening programs. While false-positive results are unavoidable in screening, efforts to minimise their occurrence and mitigate the associated harms are essential.

To improve detection rates, there's ongoing work to enhance mammography technology and combine it with other screening methods. Early detection remains critical since it offers the best chance of successful treatment and improves survival rates.

2.2 Computer-Aided Diagnosis

Over the past few decades, computer-aided diagnosis (CAD) has emerged as a popular field of research and development. CAD systems are computerised tools that assist clinicians in making accurate diagnoses by detecting abnormalities within medical images. These systems employ machine learning algorithms, such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), and K-nearest neighbours (KNNs), to analyse patterns and features in medical images. This can help clinicians in making more accurate diagnoses. These systems are versatile and can be applied across various medical imaging modalities, including mammography, magnetic resonance imaging (MRI), computed tomography scans, ultrasound, and more. The primary function of CAD is to provide clinicians with a second opinion by processing imaging data and extracting relevant information indicative of certain disease outcomes. CAD can also help in lesion detection by highlighting suspicious areas that may require further evaluation, characterising abnormalities, assisting in cancer staging, guiding treatment planning, evaluating treatment responses, monitoring for disease recurrence, and predicting prognosis [24].

The development of computer-aided detection systems for various diseases started in the 1980s. Several researchers have attempted to automate the detection of breast abnormalities. These initial studies showed potential in CAD systems, however, there were limitations due to a lack of high-quality data and limited computational resources [25]. Systematic research and development of CAD systems started in the early 1980s. In 1998, the U.S. Food and Drug Administration (FDA) approved the first commercial computer-aided detection (CAD) system to assist in the detection of breast cancer in screening mammography. Since then, research in this field has been expanding, with advancements in technology and algorithms continually improving CAD performance.

2.2.1 Literature review CAD systems

This section provides an extensive literature review and a comprehensive analysis of the existing research on the application of CAD in breast cancer diagnosis. It is divided into two subsections: traditional Machine Learning-based CAD systems and Deep Learning-based CAD systems.

Machine Learning based CAD systems

The initial findings of computer-aided diagnosis systems were very promising. The original purpose of CAD was to identify cancers that might otherwise be overlooked by radiologists. A study on CAD and mammographic screening revealed that CAD marked 77% of cancers (88 of 115) that would have otherwise been overlooked at the screening [26]. Other studies demonstrated the potential of CAD systems to identify cancerous lesions that were not visible to the human eye. One study found that CAD was able to detect 40% of cases that developed into cancers [27]. This could potentially accelerate cancer detection by up to 2-12 months[28].

Many studies have been conducted to research machine learning methods for detecting breast cancer from mammograms. A comprehensive systematic review of 154 articles sourced from various scientific databases was conducted [14]. Among the articles, 98 studies used mammogram databases for their research. The most common method used was the support vector machine, which was used in 50 articles. The SVM demonstrated a wide range of accuracy, from 64.7% to 100%. Two studies obtained flawless accuracy, while 19 others achieved high accuracy within the range of 90% to 99.5%. ANNs were also commonly used, appearing in 34 papers. Notably, 20 of these studies reported accuracy between 90% and 98.14%, while the others fell within the range of 71% to 89.38%. Additionally, KNNs were used in 14 papers, with the highest accuracy recorded at 98.69%. Other methods used included decision trees, random forests, naive Bayes, etc.

These findings demonstrate the prominence of SVM, followed closely by ANN and KNN, in the realm of mammogram data analysis. Several studies have demonstrated exceptional accuracy, especially with SVM and ANN, suggesting their efficacy in this domain [14].

Deep Learning based CAD systems

In recent years, the field of deep learning has experienced exponential growth. This growth is largely due to increased computational power, the availability of large-scale datasets, and innovations in algorithmic techniques. This progress has led to significant breakthroughs in various fields, including the analysis of medical images. Researchers have increasingly studied deep learning models for tasks like detecting breast cancer from mammograms which shows promising results.

Where traditional ML-based diagnosis has shown great promise, it often faces challenges in identifying the most relevant features for effective classification. In contrast, Deep Learning (DL) does not use any feature selection which simplifies the process. However, compared to traditional ML methods, deep learning requires a large volume of labelled data to obtain an accurate prediction. Despite this requirement, deep learning systems have demonstrated good diagnostic performance, particularly in scenarios involving low-quality breast images or complex variations in the disease.

S. Bharathi et al provided a systematic review of the literature on artificial neural networks-based models for the diagnosis of breast cancer via mammography [29]. The research presents different deep neural network models applied to different breast cancer classification datasets. Most studies employ CNNs for classification tasks using datasets such as DDSM, INBreast, and MIAS, with evaluation metrics including accuracy, AUC, sensitivity, and specificity. The results obtained from various deep learning models applied to breast cancer diagnosis include high accuracies, ranging from 65% to 100%, across different datasets and architectures, with models like Faster R-CNN achieving a sensitivity of 95% on the INbreast dataset and ROI-based CNN achieving an accuracy of 97% on the DDSM dataset. Transfer learning techniques and novel approaches like Multitask Deep Neural Networks and Generative Adversarial Networks combined with CNNs have also shown promising results, further highlighting the effectiveness of deep learning in breast cancer diagnosis. [29]

2.2.2 CAD systems in clinical practice

After the success of CAD systems in the literature, CAD systems were implemented in clinical trials. Unfortunately, the results of these trials did not match the expectations. Several studies have been conducted to compare radiologists' reading with and without CAD systems or to compare single radiologist reading with CAD to double reading mammographies. Gromet conducted a clinical trial in a single centre to compare double reading with radiologists to single reading with a CAD system [30]. They reported that the sensitivity and recall rate from double reading were 88% and 11.9%, and single reading with CAD were 90.4% and 10.6%, respectively. This is not a major difference, Gromet also looked at the results of single reading without CAD, where the sensitivity and recall rate were 81.4% and 10.2% respectively. Single reading with CAD, therefore achieved 11% higher sensitivity than the first reading and a 2.4% higher sensitivity than the double reading. Single reading with CAD also found a 3.9% higher recall rate than single reading, but an 11% lower recall rate than double reading. Gilbert also conducted a clinical trial, comparing single reading using CAD to double reading, conducted in three separate centres [31]. Each of the three centres enrolled around 9000 patients. The results showed that the sensitivity of double reading was on average 87.7%, and for single reading 87.2%. The recall rates in the two centres were very similar, but the third centre had significantly higher recall rates in single reading with CAD than double reading, 5.2% vs 3.8%, resulting in an overall recall rate over all centres at 3.9% vs 3.4%. Taylor [32] conducted a meta-analysis of studies comparing single reading with CAD or double reading to single reading. The pooled results did not show significant improvement, but the study revealed the performance of radiologists varied over a wide range.

A 2015 study compared the performance of 271 radiologists across 66 facilities in the Breast Cancer Surveillance Consortium, evaluating a total of over 600.000 mammograms with and without CAD [33]. The results showed that the sensitivity was 85. 3% with CAD and 87.3% without CAD. The specificity was 91.6% with CAD and 91.4% without. These results suggest that the use of CAD did not improve the performance of digital screening mammography in terms of sensitivity and specificity [33]. Another study in 2007 [34] found no change in cancer detection rate with and without CAD (4.2 vs 4.15 per 1000), a non-significant increase in sensitivity (84% vs 80.4%), but a significant decrease in specificity (87.2% vs 90.2%), resulting in a nearly 20% increase in biopsy rate and lower overall accuracy (AUC, 0.871 vs 0.919).

The clinical environment turned out to be far more complex than the research setting. These results from clinical setting resulted from several technical limitations, the development of CAD relied on limited computational resources, small datasets and poor image quality [35].

2.2.3 Current status CAD systems

Despite its potential, CAD-based diagnosis has limitations. Ensuring the generalisability and robustness of CAD systems remains a challenge, especially when dealing with diverse image datasets and patient variations. Overcoming challenges such as a lack of data, the need for high-quality data, and appropriate annotation is important for realizing its potential in real-world clinical scenarios.

The current performance of CAD systems is promising, but not sufficient to make CAD systems standalone detection and diagnosis clinical systems [36]. Despite significant interest and investment, traditional computer-aided detection has led to minimal or no significant improvement in performance and results [37]. Currently, there are more than 20 FDA-approved AI applications for breast imaging, but adoption and utilisation are widely variable and generally low [37]. As of now, Denmark is the only country to have implemented a deep learning-based CAD system in their population-based screening program. The current state of CAD systems for breast cancer detection is a topic of ongoing research, with a focus on improving the accuracy and efficiency of these systems [36][37].

2.3 Challenges for CAD systems

The application of deep learning models in medical imaging has shown promising results in disease detection in the literature. However, clinical practice turned out to be far more complex. Having a high-quality data set plays an important role in the effectiveness of deep learning models [38]. Deep learning algorithms, such as convolutional neural networks, are designed to automatically learn patterns and features from raw data. With a large amount of diverse data, deep learning models can capture a wide range of variations, complexities, and nuances present in image data [39]. By training a model on a large, well-curated dataset, it is possible to develop robust and generalizable models that are capable of accurately detecting various diseases and abnormalities [39].

Large datasets are essential for training deep learning models with sufficient complexity and capacity to learn hierarchical representations of medical image features. Moreover, large datasets enable the regularisation of deep learning models, helping to prevent overfitting and improve their ability to generalise to unseen data [40]. Therefore, the availability of large, well-curated datasets is of importance for the realisation of the potential of deep learning in medical imaging and the advancement of CAD systems for improved patient care.

2.3.1 Privacy Regulations

The sharing of sensitive data remains a significant challenge in medical research [15]. Mammographic images contain sensitive medical information, including details about a person's health status and potentially identifiable characteristics. Therefore, it is essential to implement measures to protect patient privacy and comply with ethical standards. Obtaining consent from patients for the use of their medical images in research is crucial. Sensitive medical data must be managed in a privacy-preserving manner. To achieve this, data must follow a legal framework such as HIPAA (Health Insurance Portability and Accountability Act) [41] in the United States or GDPR (General Data Protection Regulation) [42] in the European Union. These frameworks specify the responsibilities of organisations in protecting the privacy of personal health information [15]. However, adhering to these frameworks can be a big challenge and financial burden for healthcare organisations [15].

Furthermore, the anonymisation of data through techniques such as de-identification is essential to mitigate the risk of unintended disclosure of personal information [43]. But even with these precautions in place, the potential for re-identification or unauthorised access remains a concern This necessitates robust security measures throughout the data collection, storage, and usage processes. Therefore, addressing privacy concerns effectively is essential for the ethical and responsible use of mammographic datasets in scientific research.

2.3.2 Data Annotation

To create a well-annotated dataset, each image must be reviewed and annotated by trained professionals to identify and classify abnormalities such as masses and calcifications. Annotating mammograms requires expertise and adherence to standardized terminology and classification systems such as BI-RADS (Breast Imaging Reporting and Data System) [44]. Additionally, ensuring consistency and accuracy among annotators is very important to maintain the integrity of the dataset. Furthermore, metadata such as patient demographics, imaging parameters, and clinical outcomes need to be collected and integrated into the dataset. Overall, creating a labelled mammogram dataset demands collaboration among healthcare providers, researchers, and data annotators, along with rigorous quality control measures to produce a reliable resource for developing and validating deep learning models in breast cancer detection and diagnosis.

2.3.3 Biased data

Fairness in healthcare is based on the fundamental ethical principles of justice, beneficence, and non-maleficence. Healthcare systems must provide access to high-quality care for all individuals without discrimination. Unbalanced data, in the context of machine learning and data analysis, refers to a situation where the distribution of classes or labels within a dataset is heavily skewed. This means that one or more classes are significantly overrepresented or underrepresented compared to others. In other words, there is an imbalance in the number of instances that belong to each class. Unbalanced data is a common challenge in many real-world applications, including medical diagnosis, where positive cases are less common compared to negative cases. Managing unbalanced data is important for developing accurate and reliable models. When class imbalances are not addressed, it can lead to biased model predictions and decreased performance in minority classes.

Chapter 3

Literature

Artificial Neural Networks and Convolutional Neural Networks have revolutionised the field of artificial intelligence, providing powerful tools for complex tasks such as image recognition, natural language processing, and more. This chapter provides an in-depth exploration of these models, their architecture, and their applications. Furthermore, the chapter discusses data augmentation and enhancement methods, evaluating various data augmentation techniques, transfer learning, and the utilization of generative adversarial networks.

3.1 Neural Networks

In the last years, Artificial Neural Networks have emerged as a fundamental component of artificial intelligence. Artificial Neural Networks are models that are structurally and conceptually inspired by the human biological nervous system [45]. ANNs consist of neurons, organized in layers. Each neuron receives input signals, processes them, and produces an output signal [45].

An artificial neuron is made up of a single layer of input nodes that are directly connected to an output node. Each node in the input layer represents a specific feature or attribute of the input data. The nodes in the input layer are connected to the output layer with a certain weight, which signifies the strength of the connection between the node and the output. The weighted inputs are added together to form a linear combination, which is then passed through an activation function to produce a final output [45]. Figure 3.1 visualizes an artificial neuron, having X_i with $i \in \{1, 2, .., k\}$ input signals, where each input has a certain weight w_i with which it contributes to the output [2].

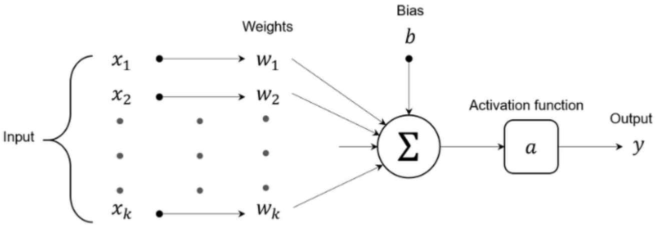


Figure 3.1: Artificial Neuron [2]

The learning process in the perceptron model involves adjusting the weights to make accurate predictions. This process starts with assigning random weights to the input, after this, a summation and activation function is applied. This will yield a predicted output which is compared to the actual output. If the inputs are misclassified the weights are adjusted. This is an iterative process that is repeated until the neuron accurately classifies the inputs. An artificial neural network can learn complex features and patterns. A deep neural network is a neural network with a certain level of complexity, usually having more than two layers [46]. Each layer adds complexity to the model while allowing the model to process the inputs concisely to output the ideal solution [46].

3.1.1 Activation functions

An important feature of the neuron is the activation function. An activation function is a function that is added to an ANN to help the network learn complex patterns in the data [47]. The activation function takes the output signal from the previous cell and converts it into some form that can be taken as input to the next cell. Activation functions are crucial as they introduce non-linearity into the network, this allows it to learn and compute non-linear mappings from inputs to outputs. Without activation functions, a neural network's output would be a simple linear function. The non-linearity enables neural networks to handle complicated, high-dimensional, and non-linear datasets effectively[47]. Activation functions transform the summed weighted input from the neuron into an output that can be fed to the next layer. The most commonly used activation functions are sigmoid, tanh, ReLU, and Leaky ReLU [48]. Each of these functions contributes to the network's ability to learn complex data representations.

Choosing the appropriate activation function for a neural network depends on the specific problem, network architecture, and the characteristics of the activation functions. For binary classification problems, the sigmoid function is suitable since it outputs values between 0 and 1, representing probabilities [48]. The tanh function, which outputs values between -1 and 1, is beneficial when zero-centered data is needed [47]. Both sigmoid and tanh can suffer from vanishing gradients. The ReLU function is commonly used in deep networks, especially for image, speech, or audio tasks, due to its ability to mitigate the vanishing gradient problem and speed up training [48]. The ReLu function can sometimes result in neurons becoming inactive, a solution is the Leaky ReLu activation, which addresses this by allowing a small gradient when inputs are negative [48]. For multi-class classification problems, the softmax function is suitable as it converts logits into probabilities that sum to 1.

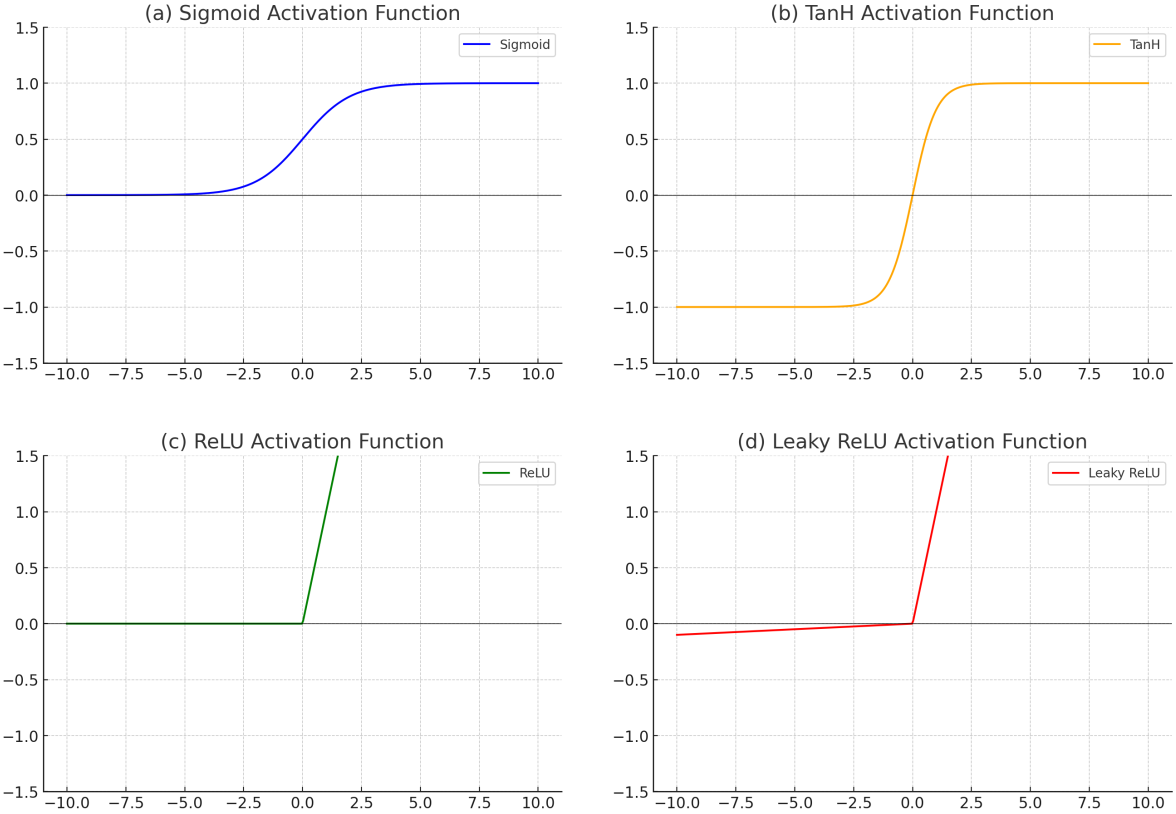


Figure 3.2: Comparison of various activation functions. Graphs were created using Python's Matplotlib library.

3.1.2 Forward propagation

Forward propagation is the initial phase of training a neural network, where input data is passed through the network to generate an output. During forward propagation, input data is fed to the neural network and then propagated through each layer of the network. It involves the following steps:

- **Input Layer:** The input data is fed into the input layer of the neural network.
- **Hidden Layers**: The input data is processed through one or more hidden layers. Each neuron in a hidden layer receives inputs from the previous layer, applies an activation function to the weighted sum of these inputs, and passes the result to the next layer.
- **Output Layer:** The processed data moves through the output layer, where the final output of the network is generated. The output layer typically applies an activation function suitable for the task, such as softmax for classification or linear activation for regression.
- **Prediction:** The final output of the network is the prediction or classification result for the input data.

Let's consider a neural network with L layers. Let X be the input data. For each layer l (where $l = 1, 2, \ldots, L$):

- $W^{[l]}$ is the weight matrix for layer l.
- $b^{[l]}$ is the bias vector for layer l.
- $Z^{[l]}$ is the linear combination of inputs for layer l.
- $A^{[l]}$ is the activation output for layer l.

To perform forward propagation, the following steps must be followed for each layer l (where $l = 1, 2, \ldots, L$):

1. Compute the linear combination of the inputs. For the input layer, set $A^{[0]} = X$. The linear combination can be calculated using the following equation:
$$Z^{[l]} = W^{[l]}A^{[l-1]} + b^{[l]} \tag{3.1}$$
2. Apply the activation function σ_l to the linear combination to obtain the activation output:
$$A^{[l]} = \sigma_l(Z^{[l]}) \tag{3.2}$$

The goal of forward propagation is to calculate the predicted outputs of the neural network based on the input data. By making a prediction, the model performance can be evaluated by a chosen loss function. The loss function indicates how far the model's predictions are from the actual true values. This will help the optimization process by adjusting the model's weights to make better predictions.

3.1.3 Loss function

Loss functions, also known as cost functions or objective functions, are very important in neural networks as they quantify the difference between the predicted outputs and the actual target values. This measurement of error helps the optimization process, by helping the model to improve its predictions over time. Different types of loss functions are used depending on the nature of the problem. For regression tasks, the Mean Squared Error (MSE) is commonly used. MSE calculates the average of the squares of the errors between predicted and actual values [49]. The MSE function is defined by equation 3.3 where y_i is the ith observed value, $\hat{y}_i$ is the corresponding predicted value, and N is the number of observations.

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.3}$$

For classification tasks, the Cross-Entropy Loss is often employed. Cross-entropy loss measures the difference between the predicted probabilities and the actual labels in classification tasks [50]. This can be used to evaluate how well the model's predictions are. There are two types of cross-entropy loss depending on the classification task, binary and multiclass.

In binary classification, there are only two possible classes. Binary Cross-Entropy loss (BCE) calculated the difference between the actual binary labels and the predicted probabilities. The formula for binary cross-entropy loss is:

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \tag{3.4}$$

In this formula again y_i is the actual label, $\hat{y_i}$ is the predicted probability of the positive class, and N is the number of instances.

In multiclass classification, where there are more than two classes, Categorical Cross-Entropy loss (CCE) is used. CCE compares the predicted probability distribution over the classes with the actual distribution, this is usually one-hot encoded. Categorical cross-entropy uses the following equation:

$$\mathcal{L}_{CCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log(\hat{y}_{ic}) \tag{3.5}$$

Here, y_{ic} is 1 if the actual label if instance i is c and 0 otherwise, $\hat{y_i}$ is the predicted probability of instance i being class c. And again, N is the number of instances and C is the number of classes.

3.1.4 Backward propagation

Backward Propagation (BP), was first introduced in 1986 by Rumelhart [51]. Backward propagation, also known as backpropagation, is the process of adjusting the weights to minimize the loss function. After forward propagation is performed and the loss is calculated, backward propagation calculates the gradient of the loss function with respect to each weight [51].

This involves computing the partial derivatives of the loss function with respect to the network's weights, starting from the output layer and moving backwards through the hidden layers to the input layer. These gradients indicate how much the loss would change with a small change in each weight. Using these gradients, the weights are updated in the opposite direction of the gradient by a certain amount determined by the learning rate. This iterative process of forward propagation followed by backpropagation continues until the network's performance converges to an optimal or satisfactory level, effectively reducing the error and improving the network's predictions.

Backpropagation follows the following steps:

1. **Compute the Loss Gradient at the Output Layer**: First, compute the gradient of the loss function with respect to the output $A^{[L]}$. Let Y be the true labels and $\mathcal{L}$ be the loss function.

$$dA^{[L]} = \frac{\partial \mathcal{L}}{\partial A^{[L]}} \tag{3.6}$$

2. **Propagate the Gradient Back Through the Network**: For each layer l (where $l = L, L-1, \ldots, 1$):

   - **Compute the Gradient of the Linear Combination**: Calculate the loss gradient with respect to $Z^{[l]}$, which involves the derivative of the activation function σ':

$$dZ^{[l]} = dA^{[l]} * \sigma'(Z^{[l]}) \tag{3.7}$$

   where $*$ stands for element-wise multiplication.

- **Compute the Gradients of the Weights and Biases**: Calculate the loss gradients with respect to the weights $W^{[l]}$ and biases $b^{[l]}$:

$$dW^{[l]} = \frac{1}{N} dZ^{[l]} (A^{[l-1]})^T \tag{3.8}$$

$$db^{[l]} = \frac{1}{N} \sum_{i=1}^{m} dZ_i^{[l]} \tag{3.9}$$

where m is the number of samples.

- **Compute the Gradient of the Activation from the Previous Layer**: Calculate the gradient with respect to the previous layer's activation $A^{[l-1]}$ to propagate the error backwards:

$$dA^{[l-1]} = (W^{[l]})^T dZ^{[l]} \tag{3.10}$$

3. **Update the Parameters**:

- Using the gradients computed, update the weights and biases for each layer l using an optimization algorithm like gradient descent. If η is the learning rate:

$$W^{[l]} = W^{[l]} - \eta dW^{[l]} \tag{3.11}$$

$$b^{[l]} = b^{[l]} - \eta db^{[l]} \tag{3.12}$$

This iterative process updates the parameters of the weights and bias to minimize the difference between the predicted value and the target value. This process helps the neural network to learn the patterns in the training data, improving its accuracy and generalisation to new, unseen data.

3.2 Convolutional Neural Networks

Convolutional neural networks are a type of deep learning algorithm designed to process arrayed data, particularly images. Unlike traditional artificial neural networks, CNNs are specialized in pattern recognition within images and are widely used in various computer vision tasks [52].

One key difference between CNNs and traditional ANNs lies in their architecture and operation. While ANNs consist of neurons that receive inputs and perform operations to self-optimize through learning, CNNs are structured with specific layers tailored for image processing tasks. These layers include convolutional layers, pooling layers, and fully connected layers [52]. The architecture of a CNN can be found in figure 3.3.

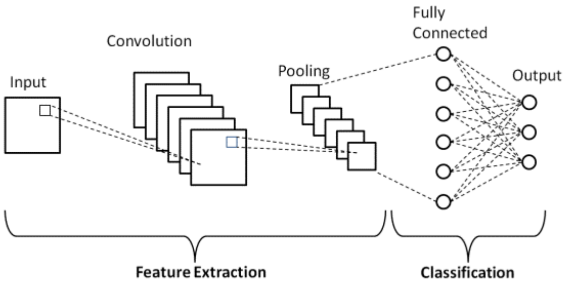


Figure 3.3: The Architecture of a Convolutional Neural Network [3]

3.2.1 Input layer

The input layer of a CNN contains image data, where images are represented as arrays of pixels. In greyscale images, each pixel spans a range from 0 to 255, where 0 signifies black and 255 signifies white. In colour images, pixels contain three channels, red, green, and blue, with each channel also ranging from 0 to 255. Greyscale images can be portrayed as 2D arrays, height and width, while colour images are represented in 3D arrays, height, width, and colour channel.

3.2.2 Convolutional layer

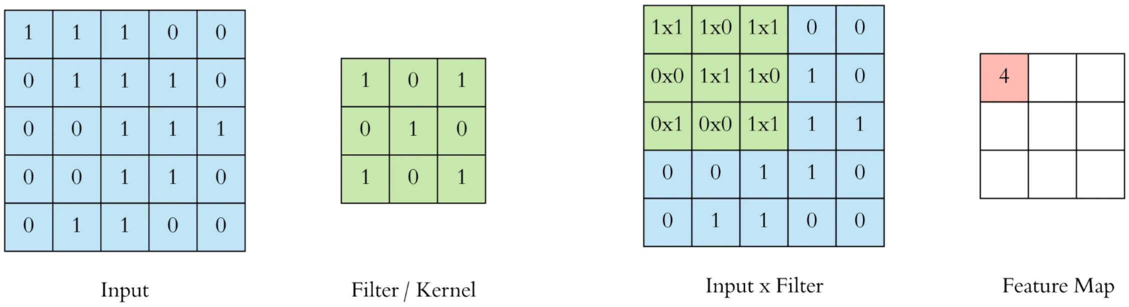
The convolutional layer is the core building block of a convolutional neural network, containing the main computational load of the network [53]. Its primary function is to extract features from the input data. This is achieved through a mathematical operation called convolution, which involves the use of kernels, also known as filters. Kernels are small weight matrices that slide over the input image, performing an element-wise multiplication between their weights and the corresponding pixel values in the image region they cover [54]. The results of these multiplications are then summed to produce a single value. The mathematical representation of the convolution operation can be found in equation 3.13.

$$(I * K)(i, j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i + m, j + n) \cdot K(m, n) \tag{3.13}$$

where:

- I is the input image (or input feature map).
- K is the kernel (filter).
- $I(i, j)$ is the pixel value at position (i, j) in the input image.
- $K(m, n)$ is the weight at position (m, n) in the kernel.
- M and N are the height and width of the kernel, respectively.
- $(I * K)(i, j)$ is the output value at position (i, j) in the output feature map.

The output of a convolution is called a feature, the features are then added to a feature map. This process is repeated as the kernel moves across the image, creating a feature map that highlights certain features within the input data. Figure 3.4 shows a simple convolutional operation. Figure 3.4a shows an input of 5x5 and a kernel of 3x3, a convolution operation is applied in figure 3.4b that results in a feature on the feature map.



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

Input x Filter

4		

Feature Map

(a) Input and filter matrices in convolution.

(b) Feature map generation via convolution.

Figure 3.4: Convolution operation: From input to feature map. [4]

Several important hyperparameters influence the behaviour and performance of the convolutional layer [54]:

- **Kernel size**: Kernels are typically small, such as 3x3 or 5x5, and are initially set with random values. Smaller kernels capture fine details, while larger kernels can capture broader features.
- **Stride**: The stride is the number of pixels by which the kernel moves each step. A stride of 1 means the kernel moves one pixel at a time, resulting in a detailed feature map. Larger strides reduce the size of the feature map and lead to a coarser representation.
- **Padding**: Extra pixels are added around the border of the input image to control the spatial size of the output feature map. The most common form of padding is zero-padding, where zeros are added around the border

- **Number of Kernels**: The number of different kernels applied in a convolutional layer. Each kernel produces a separate feature map, allowing the network to detect multiple features.

Equation 3.14 determines the spatial dimension and depth of the output feature map after applying the convolutional operation. This takes the kernel size, stride, padding, and number of kernels into account.

$$(W_{l+1}, H_{l+1}, D_{l+1}) = \left(\frac{W_l - K_l + 2P}{S} + 1, \frac{H_l - K_l + 2P}{S} + 1, N_K \right) \tag{3.14}$$

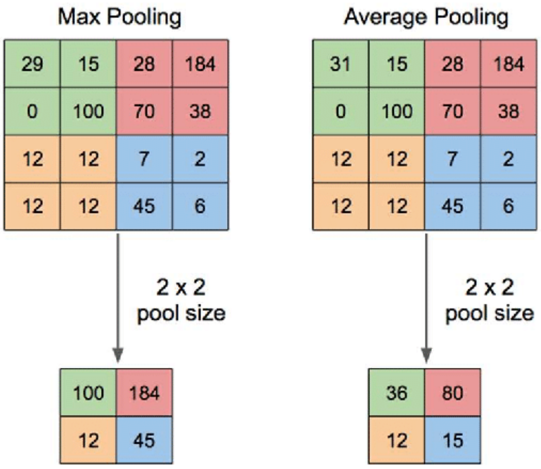
After the convolution operation, the values are passed through an activation function. One of the most common activation functions used in CNNs is the ReLu function. The activation function introduces non-linearity into the model, this allows the CNN to learn complex features.

During the training of the CNN, the weights of the kernels are adjusted by backpropagation and gradient descent. This adjustment process minimises the difference between the predicted outputs and the ground-truth labels, allowing the kernels to learn and extract meaningful features from the input data effectively. Convolutional layers are essential for detecting patterns and features, such as edges, textures, and shapes, which are crucial for tasks such as image classification and object detection.

3.2.3 Pooling layer

Pooling is a fundamental operation in convolutional neural networks that plays a crucial role in down-sampling feature maps while retaining important information [52]. It helps in controlling the model's complexity, reducing overfitting, and improving computational efficiency by reducing the number of parameters and computation required in subsequent layers. The pooling operation involves sliding an n-dimensional filter over each channel of the feature map and summarising the features lying within the region covered by the filter.

The two most common types of pooling are Max Pooling and Average Pooling [54]. Max-pooling returns the maximum value from the portion of the image covered by the kernel. This operation retains the most prominent features, making the network more robust to variations and distortions in input data [54]. Average-pooling returns the average of all the values from the portion covered by the kernel, providing a smoothed version of the feature map [54]. A visual representation of 2x2 pooling is shown in Figure 3.5.



Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

100	184
12	45

Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

36	80
12	15

Figure 3.5: Comparison of Max Pooling and Average Pooling [5].

3.2.4 Fully connected layer

The fully connected layer is important for the final stages of the CNN. Before the input is in the fully connected layer, the input will get flattened into a one-dimensional vector. This allows each neuron in a layer to connect to all the neurons in the next layer. This connectivity allows the network to learn complex features and make global decisions. During training the weights and biases are adjusted through backpropagation and an optimiser to minimise the loss function. Like the convolutional layer, the fully connected layer applies an activation function to introduce non-linearity into the model. In classification tasks the fully connected layer will give a probability distribution over the classes, allowing the network to make a prediction.

3.2.5 Hyperparameters

Hyperparameters are configuration values that are set before the learning process begins. Hyperparameters are tunable and can directly affect how well a model trains and can make accurate predictions.

Batch Size

The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. Choosing a batch size can be important for the performance of a model. Generally, a smaller batch size performs better [55]. Research of [56] claims that large batch sizes tend to result in models that get stuck in local minima. A smaller batch size is more likely to push out local minima and find the global minima. However, a smaller batch size results in many updates and higher computational costs.

Learning Rate

The learning rate is a hyperparameter that determines how much to change the model in response to the estimated error each time the model weights are updated. The learning rate determines the strength with which the newly acquired information will override the old information. A factor of 0 will make the agent not learn anything, while a factor of 1 will make the agent consider only the most recent information [57]. When the learning rate is too small, it may result in a slow learning process, while a big learning rate could mean that the loss value does not converge, leading to a failure in the learning process [57]. There is a high correlation between the learning rate and the batch size; when the learning rates are high, larger batch sizes tend to perform better than with small learning rates [55].

Optimiser

An optimiser is an algorithm or method used to change the parameters of a neural network, such as weights and learning rates. By adjusting the weights of the network to help reduce loss and improve accuracy. Two of the most common optimisers are Stochastic Gradient Descent and Adam.

- **Stochastic Gradient Descent (SGD)** is one of the most basic, but effective optimization algorithms. It randomly selects one data point from the whole data set at each iteration to reduce the computations enormously and still performs robustly. The weights are updated based on the gradient of the error with respect to the current weight. Despite its simplicity, SGD often takes longer to converge and may get stuck in local minima, unlike other more sophisticated optimisers [58].
- **Adam (Adaptive Moment Estimation)** is an algorithm for first-order gradient-based optimization of stochastic objective functions [59]. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. The algorithm calculates an exponential moving average of the gradient and the squared gradient, and the parameters beta1 and beta2 control the decay rates of these moving averages. The initial learning rate is 0.001 by default and it's suitable for most of the cases [59]. Adam is often a good choice for many deep learning problems, as it usually never underperforms methods such as gradient descent and momentum [60].

Each optimiser has its strengths and weaknesses, and the choice of optimiser can significantly affect the speed of convergence and the final performance of the network.

3.2.6 Classification models

In recent years, several CNN architectures have been presented. The model architecture is a crucial factor in improving the performance of different applications. A reliable benchmark for evaluating the efficacy of these models is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). This is an annual competition that was held from 2010 to 2017. It was an annual software contest where models competed to correctly classify and detect objects and scenes. The challenge played a significant role in the advancement of deep learning and computer vision. The following section explores the ImageNet Challenge and the evolution of classification models that it has fostered over the years.

ImageNet challenge

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC), more commonly referred to as the ImageNet challenge, is an annual competition that has been a significant driving force in developing and benchmarking computer vision models [61]. The primary objective of the ImageNet challenge is to advance the state-of-the-art in image classification and object recognition. The competition uses the ImageNet dataset, a large-scale, labelled dataset of images designed for visual object recognition research(source). The dataset contains millions of images categorised into thousands of classes. For example, it includes images of animals, objects, scenes, and various everyday items. The challenge consists of several tasks, including image classification, object detection, and image segmentation [61][62]. The performance of the classification models is evaluated using the top-5 error rate, which refers to the proportion of test images for which the model's five most probable labels match the true label. The top-5 error rate is useful in the context of large-scale classification problems with many classes, where even highly accurate models may sometimes miss the exact label but still provide a highly relevant top-5 list.

In 2012, AlexNet presented a model that reduced the top-5 error rate from 25% to 15.3%, marking a significant improvement in performance and demonstrating the superiority of deep learning over traditional methods in image classification tasks. Following AlexNet, VGGNet was introduced and achieved a top-5 rate error of 7.3% in the 2014 ImageNet challenge [63]. Around the same time, Google introduced GoogLeNet, also known as Inception, which used a unique module that combined multiple convolutional filters of different sizes to make the network deeper and more efficient [64]. GoogLeNet won the 2014 competition with a top-5 error rate of 6.7% [65]. In 2015, ResNet, developed by Kaiming He and his team, solved the problem of training very deep networks by introducing residual connections [66]. ResNet-152 achieved a top-5 error rate of 3.6%, winning the ImageNet challenge and pushing the boundaries of deep learning [67]. DenseNet, introduced in 2017 by Gao Huang and colleagues, used dense connections to improve information flow and efficiency [68]. DenseNet-121 achieved a top-5 error rate of 5.92% on ImageNet, while DenseNet-264 achieved a top-5 error rate of 3.46% [69]. More recent advances include EfficientNet. EfficientNet-B7, developed by Google in 2019, optimised the scaling of network depth, width, and resolution, achieving a top-five state-of-the-art error rate of 2.7% [10].

3.2.7 Classification for mammography

The application of convolutional neural networks in mammography has been a topic of significant interest in recent years. CNNs have been used for various mammography tasks, including lesion localisation and detection, risk assessment, image retrieval, and classification tasks. Several models that emerged from the ImageNet challenge have been applied in breast cancer detection.

AlexNet has been used to categorise mammography images into benign and malignant tumours, achieving an overall system accuracy of 95.70% [70]. Another study by Shayma Hassan et al. compared the performance of Adam and SGD optimisers for AlexNet and GoogleNet architectures. The results showed that in both architectures, the Adam optimiser outperformed SGD, with AlexNet achieving an accuracy of 97.18% compared to 94.17% using GoogleNet. Thus, AlexNet outperformed GoogleNet in this comparison.[71]

The Visual Geometry Group (VGG) network has also been utilized in mammography classification. A study by G. Jayandhi et al. employed a Deep Convolution Neural Network (DCNN) with Transfer Learning (TL) that utilizes mammogram image samples for breast cancer diagnosis. Results showed that the VGG-16 architecture provided promising results on Mammographic Image Analysis Society

(MIAS) database images with 82.5% accuracy [72].

ResNet, a model known for its residual connections that help train deeper networks, has also been benchmarked for breast cancer diagnosis using mammography images. The highest classification accuracy was achieved by using ResNet18, which achieved an accuracy of 93.83% [73].

EfficientNet, which optimises the scaling of network depth, width, and resolution, has shown superiority in cancer classification in a study comparing various CNN architectures, achieving an accuracy of 95% [74].

These findings highlight the potential of CNNs in improving the accuracy of breast cancer detection in mammography. Advancements in deep learning techniques, such as residual connections in ResNet and the scaling efficiency in EfficientNet, have significantly contributed to their applicability in medical imaging. The successful application of these architectures to mammography classification demonstrates the continuous progress in CNN research and its critical role in enhancing diagnostic tools for breast cancer detection.

3.3 Transfer learning

Transfer learning is a machine learning technique where a pretrained model, developed for one task, is reused as the starting point for a model on a second task. This approach leverages the knowledge and features learned by the model from a large, well-labelled dataset, often used in tasks like image classification or natural language processing, and applies them to a new task with limited data. By transferring the learned features and weights, transfer learning significantly reduces the time and computational resources required to train a new model from scratch, improves the model's performance, and helps in achieving better generalization, especially in scenarios where the available data is sparse or expensive to collect. Transfer learning is beneficial in fields such as image classification, where training models from scratch requires significant computational resources. Research indicates that transfer learning can significantly improve model performance compared to training alone. Studies have demonstrated enhanced accuracy in neural networks and convolutional neural networks when transfer learning is applied either before or after the learning process.

3.4 Data Augmentation

Data Augmentation is a technique used in machine learning to increase the diversity and size of a training dataset. This is achieved by applying various transformations to the existing data and creating modified versions of the existing data. Data augmentation is widely used and has been shown to be highly effective for image datasets. The goal of data augmentation is to help models generalize better by exposing them to a variety of transformations of the training images, making the model more robust. By making the model more robust, it can reduce the risk of overfitting. Numerous studies have shown that data augmentation can lead to substantial improvements in the performance of image classification and other computer vision tasks. Two common types of data augmentation techniques are geometric transformations and photometric transformations. Geometric data augmentation techniques involve spatial transformations that alter the layout of the original images. On the other hand, photometric data augmentation techniques are based on manipulating the colour properties or pixel values of the images.

3.4.1 Geometric transformations

Geometric transformations are among the most common data augmentation techniques. These are transformations that alter the geometry of the image by mapping the individual pixel values to new destinations [75]. These transformations are widely used because they are easy to implement and effective. Some common techniques include:

- **Rotating**: This involves changing the orientation of an image by rotating it across an axis by a certain degree. This is useful in scenarios where the orientation of the object in the image can vary. Source

- **Flipping**: Generate a mirror image of an image, this can be done either vertically or horizontally. This technique is typically used in classification tasks, where orientation may vary.
- **Translation**: Shift images horizontally or vertically. This simulates objects appearing in different positions within the frame, improving the model's ability to detect objects regardless of their position, thus preventing a positional bias.
- **Scaling**: Scale the images along different axes with a scaling factor. Objects may vary in size, this can bring realistic augmented images into the training set.

These techniques are commonly used in data augmentation, a strategy to increase the diversity of the training set without actually collecting new data. By applying these transformations, we can teach our model to focus on the essential features of the objects, making it more robust and improving its ability to generalize from the training data to new, unseen data.

3.4.2 Photometric Transformations

Photometric transformations are also popular data augmentation techniques, photometric data augmentation involves manipulating the colour properties of images to create augmented data. These transformations are particularly useful for grayscale images, which often suffer from variability in contrast and brightness. Techniques include:

- **Contrast adjustment:** This technique modifies the contrast of an image by scaling the distance of the pixel values from the image mean. High-contrast images have a wider range of pixel values, while low-contrast images have a narrower range. Adjusting the contrast can help the model learn to recognize features under different lighting conditions.
- **Brightness adjustment:** This technique changes the brightness of an image by adding or subtracting a constant value from every pixel in the image. This can help the model learn to recognize features under varying levels of illumination
- **Gaussian noise:** Gaussian noise involves adding a random variation of brightness to an image at each pixel. The variation is sampled from the Gaussian normal distribution. This technique can help improve the robustness of a model by allowing it to learn from images with various levels of noise, simulating real-world conditions where images may be affected by different types of noise.

3.4.3 Data Augmentation for Medical Imaging

P. Pratheep Kumar conducted a comparative study of various augmentation techniques for classifying breast cancer into benign and malignant categories using the INBreast dataset [76]. The techniques employed included flipping, cropping, rotation, noise injection, random brightness, and a combination of all these methods. Without any data augmentation, the model achieved a classification accuracy of 94.56%. The implementation of image data augmentation techniques significantly improved the model's performance. The flipping technique resulted in an accuracy of 96.46%, while the cropping technique yielded an accuracy of 95.21%. The rotation method further improved the accuracy to 97.36%. Interestingly, the noise injection resulted in a slightly lower accuracy of 94.56%. Most notably, the combination of all augmentation techniques led to the highest classification accuracy of 98.91%, outperforming all individual methods. This underscores the effectiveness of using a comprehensive set of data augmentation techniques in improving model performance.

Several other studies have utilized augmentation techniques such as flipping, rotation, translation, and scaling to improve the performance of their models [77]. For instance, a study applied R-CNN for mass detection using horizontal and vertical flipping, achieving a sensitivity of 0.83 [78]. Another study used a modified AlexNet for tumour detection and achieved 95.70% using scaling, horizontal flip, and rotation (90, 180, 270 degrees) [70]. Another study also trained an AlexNet for mammogram classification, obtaining an accuracy of 93.2% using several geometric transformations [79].

3.5 Generative Adversarial Networks

A Generative Adversarial Network is a deep learning architecture first developed in 2014 by Ian Goodfellow and his colleagues [80]. GANs have gained significant attention due to their ability to generate realistic data samples that closely resemble real-world data. This section delves into the architecture, training process, and objective functions of GANs.

3.5.1 Basic Architecture

GANs are composed of two main neural networks: the generator and the discriminator. The two networks are pitted against each other, with one generating new data, such as images, that the second network then tries to identify as real or generated. The basic architecture of a GAN can be found in figure 3.6.

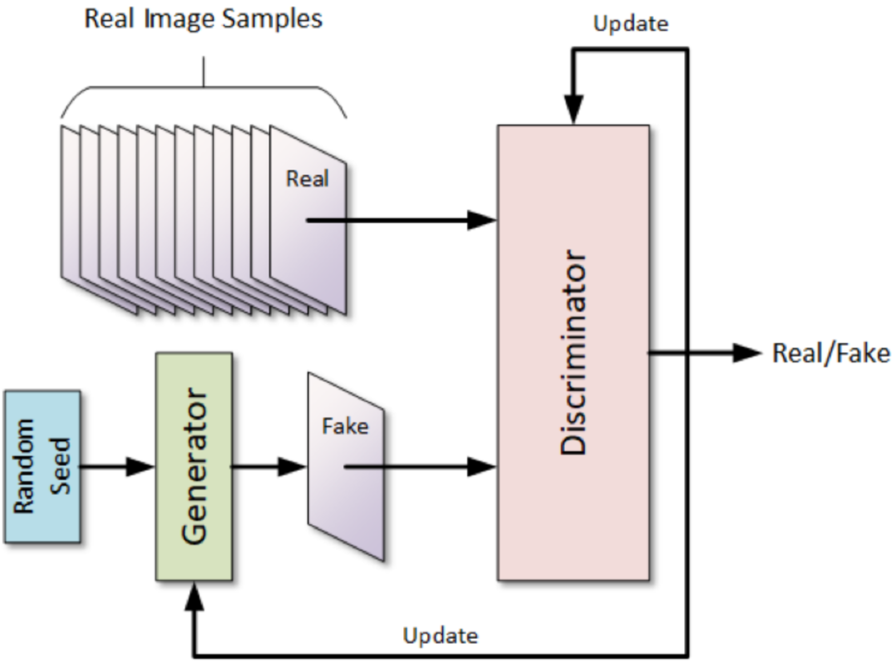


Figure 3.6: Architecture of a Generative Adversarial Network [6]

3.5.2 Generator

The generator's purpose is to create new data instances that resemble the properties of the training data. It takes random noise as input and transforms this noise into data samples, such as images, sounds, or text. The primary objective of the generator is to fool the discriminator by creating data samples that are realistic enough to be indistinguishable from real data.

The architecture of a generator may vary, but in many popular GANs the generator typically comprises a sequence of the following components:

- **Input Layer**: The generator initiates with an input vector of random noise, which is typically drawn from a simple distribution such as Gaussian or uniform. This vector serves as a starting point for the data generation process.
- **Fully Connected Layers**: These initial layers serve to expand the dimensionality of the input noise, preparing it for further transformations.
- **Batch Normalization Layers**: These layers These layers stabilize and accelerate the training process by normalizing the outputs of the previous layers, this results in consistent learning gradients and faster convergence.

- **Activation Functions**: These functions introduce non-linearity to the model, enabling it to generate more complex data. The ReLU and Leaky ReLU activation functions are commonly employed in the generator.
- **Transposed Convolutional Layers**: These layers upsample the input from the previous layer to a higher spatial dimension, which is needed to increase the resolution of the generated image. This effectively acts as the inverse of regular convolutional layers. These layers are also known as deconvolutional layers, these layers are fundamental in the generator.
- **Reshaping layers**: These layers are used to reshape the data into the desired output format.
- **Output layer**: The final layer usually employs a tanh or sigmoid activation function, depending on the nature of the data being generated. For image generation, a tanh function is often used to output pixel values in a normalized range.

The objective of the generator is to produce outputs that can be classified as real by the discriminator. During the training phase, the generator receives feedback from the discriminator regarding the realism of the generated images. The objective of the generator is to minimise the probability that the discriminator detects fake images.

3.5.3 Discriminator

The discriminator's purpose is to distinguish between real data and fake data. It outputs a probability value that indicates whether a given sample is real or fake. The discriminator is essentially a binary classifier that learns to correctly identify real versus generated data through continuous training.

The discriminator receives both real data samples from the training dataset and synthetic samples from the generator as input. It then processes these inputs through multiple layers to output a probability score indicating the likelihood that each sample is real. The architecture of the discriminator typically consists of convolutional layers, batch normalization layers, and fully connected layers.

- **Convolutional Layers**: These layers apply filters to the input data to extract hierarchical features. In image processing, these features can include edges, textures, and more complex structures as the layers deepen.
- **Batch Normalization Layers**: These layers stabilize and accelerate the training process by normalizing the outputs of the previous layers, this results in consistent learning gradients.
- **Fully Connected Layers**: These layers aggregate the features extracted by the convolutional layers into a final probability score, using a sigmoid activation function to produce a value between 0 and 1.

3.5.4 Adversarial training

Generative Adversarial Networks are trained through an adversarial process that aims to improve both the generator and the discriminator. This competing dynamic improves the generator's ability to create realistic data that can fool the discriminator, while the discriminator improves its ability to distinguish between real and fake data. Through iterative improvement, both networks continuously improve as they learn from each other's feedback.

The training process begins with the initialization of both the generator and the discriminator. Once initialized, the discriminator is trained first. The discriminator is provided with a batch of real data from the training set and a batch of fake data generated by the generator. The discriminator processes these batches separately, giving a probability for each instance. For real data, it aims to output a probability close to 1, indicating that the data is real. For fake data, it aims to output a probability close to 0, indicating that the data is synthetic. The loss for the discriminator is computed based on its ability to correctly classify real and fake data. This loss is a combination of two terms: one for real data classification and one for fake data classification [81]. Mathematically, the discriminator's loss is expressed as:

$$\mathcal{L}_D = -(E[\log(1 - D(x))] + E[\log(1 - (D(G(z))))]) \tag{3.15}$$

Where $D(x)$ is the discriminator's prediction for real data x, and $D(G(z))$ is its prediction for fake data generated by generator $G(z)$ from noise z.

After updating the discriminator's weights to minimize this loss, the generator is trained. The generator's objective is to produce data that the discriminator cannot distinguish from real data. To achieve this, the generator's loss is calculated based on the discriminator's response to fake data. Specifically, the generator tries to maximize the discriminator's error in identifying fake data as fake, which is equivalent to minimizing the negative log probability of the discriminator classifying fake data as real. The generator's loss is given by:

$$\mathcal{L}_G = -E[\log(D(G(z)))] \tag{3.16}$$

Where $D(G(z))$ is the discriminator's prediction for fake data generated by the generator. By back-propagating this loss, the generator's weights are adjusted to produce more realistic data

The training process of GANs is framed as a minimax game, where the generator and the discriminator have opposing objectives. The generator seeks to minimize its loss, which translates to generating data that the discriminator misclassifies as real. Conversely, the discriminator seeks to maximize its ability to distinguish real from fake data. This adversarial relationship can be formalized into a single objective function that both networks optimize:

$$\min_G \max_D V(G, D) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.17}$$

This function encapsulates the adversarial nature of the GAN training, where the generator aims to minimize this value while the discriminator aims to maximize it. Through this process, both networks improve, leading to the generation of realistic data and the accurate classification of real versus fake samples. This adversarial relationship is what gives GANs their name and their power.

Stability Issues in GAN Traning

Despite their potential, the training of GANs is notoriously challenging due to several stability issues. These problems often result in poor convergence and suboptimal performance:

- **Non-Convergence:** GANs frequently fail to converge, this issue arises due to the adversarial nature of the training process. The generator and discriminator both try to minimize their loss while maximizing the other's. This dynamic can lead to oscillations, where the generator and discriminator keep outsmarting each other without reaching a stable solution. In some cases, the discriminator becomes too good too quickly causing the generator to fail. In other cases, if the generator becomes too good, the discriminator might fail to provide useful feedback to the generator. Moreover, the loss landscape of GANs is highly non-convex, making it difficult to find a global optimum. The training process can get stuck in local optima or saddle points, leading to suboptimal solutions. [82]

- **Mode Collapse:** Mode collapse is a significant challenge in training Generative Adversarial Networks. It occurs when the generator of a GAN starts producing similar or identical samples, leading to a collapse in the modes of the data distribution1. This means the generator fails to capture the full diversity of the real data distribution. In other words, the generator becomes stuck in a particular mode or pattern, failing to generate diverse outputs that cover the entire range of the data. This can result in the generated output appearing repetitive, lacking in variety and detail, and sometimes even being completely unrelated to the training data. [83]

- **Vanishing Gradients:** The vanishing gradients problem is a significant issue in the training of Generative Adversarial Networks. This problem arises when the gradients used to update the weights of the neural network become very small, almost zero1. As a result, the weights are not updated anymore, and learning stalls. In the context of GANs, if the discriminator becomes too good, then generator training can fail due to vanishing gradients. In effect, an optimal discriminator doesn't provide enough information for the generator to make progress. This is because the gradients that flow back from the discriminator to the generator during backpropagation become very small. This results in very slow learning or a complete halt in learning for the generator. [83]

In conclusion, GANs have the potential to be a powerful tool in several fields, but their training process presents significant challenges. These include non-convergence, mode collapse, and vanishing gradients. Despite these issues, the promise of GANs in generating complex, high-dimensional distributions is significant, and ongoing research continues to explore solutions to these stability issues.

3.5.5 Hyperparameters

Hyperparameters play an important role in the training and performance of Generative Adversarial Networks. Proper tuning of these hyperparameters can significantly affect the stability, convergence speed, and quality of the generated images. This section discusses the key hyperparameters involved in training GANs and their impact on the model's performance.

Batch size

The batch size determines the number of training examples used in one forward or backward pass. A large batch size provides stable updates but requires more memory and computational resources. While a small batch size is less memory-intensive but can lead to less stable updates. Common batch sizes range from 32 to 256, with 64 being a typical choice for many GAN applications.

Learning Rate

The learning rate controls the step size to update the model parameters during training. Both the generator and the discriminator have their learning rate, and tuning these rates is essential for balanced training. If the generator learning rate is too high, it may cause instability, if it is too low, the convergence may be slow. If the discriminator learning rate is too high, it can overpower the generator, while a low learning rate will slow down the discriminator's ability to detect.

The learning rate values typically range from 0.001 to 0.00001, a good starting point is 0.0002 which is often used initially and adjusted based on training dynamics.

Optimiser

Similar to a CNN, the Adam Optimizer is widely used in GAN training for its adaptive learning. The default beta values ($\beta_1 = 0.5, \beta_2 = 0.999$) are often used.

Regularization techniques

Regularisation techniques are crucial for training Generative Adversarial Networks as they help in stabilizing the training process and improving the model's performance.

- **Batch Normalisation:** This is a technique used to standardise the inputs to a layer for each mini-batch. This results in stabilising the learning process and reducing the number of training epochs required to train deep networks.

- **Dropout:** Dropout is a regularization technique that is used in neural networks to prevent overfitting. During training, dropout randomly deactivates a proportion of neurons, effectively dropping out these neurons. This process forces the network to learn redundant representations and become more robust. By reducing reliance on specific neurons, dropout helps improve the generalization ability of both the generator and discriminator, leading to more stable and effective GAN training.

- **Label Smoothing:** This is a technique to prevent the discriminator from becoming too confident in its predictions. Instead of using binary labels, 1 for real and 0 for fake, label smoothing uses decimal values such as 0.9 for real and 0.1 for fake. This approach can improve the performance of GANs by making the discriminator more robust to variations in the training data and better able to distinguish between real and fake samples from different distributions. It also helps reduce the chance of mode collapse1.

- **Gradient Penalty:** Gradient Penalty is a regularization technique used to enforce the Lipschitz continuity condition in Wasserstein GANs (WGANs), enhancing training stability. Instead of clipping weights, the Gradient Penalty adds a term to the loss function that penalises the critic's gradient norm deviation from 1. This penalty term ensures smoother and more reliable gradients, reducing issues like mode collapse and improving convergence. It is calculated as:

  $$\mathrm{GP} = \lambda E_{\hat{x}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right] \tag{3.18}$$

  where $\hat{x}$ are interpolated points between real and generated data, and λ is a regularization coefficient.

- **Feature Matching:** This technique aims to improve the stability of GAN training by modifying the generator's objective to match the statistics of features extracted by the discriminator from real and generated data. Instead of directly minimizing the discriminator's output, the generator minimizes the difference between the feature representations of real and generated samples. This encourages the generator to produce data that is not only indistinguishable from the discriminator but also similar in structure and content to the real data, leading to more realistic outputs and stable training.

These techniques are commonly used in GANs to improve the stability of the training process and the quality of the generated samples.

3.5.6 Variants of GANs

Over the years, several variants of GANs have been developed to address specific challenges and improve their performance. These variants improve the basic GAN architecture to better handle issues like mode collapse, and training instability, and to cater to specific applications. This section will explain some of the most common variants of GANs.

Deep Convolutional GAN

A well-known and effective network GAN design is the Deep Convolutional GAN (DCGAN) [84]. DCGAN was introduced by Radford et al in 2015 and has since become one of the most popular variants of GANs. The DCGAN have a couple of architectural guidelines:

- Replacing any pooling layers with stride convolutions, allows the network to learn its spatial downsampling
- Use batch normalization in both the generator and the discriminator
- Remove fully connected hidden layers for deeper architectures
- Use ReLu activation for the generator for all layers, except the output which uses Tanh
- Use LeakyReLu activation in the discriminator for all layers

Conditional GAN

Generative Adversarial Networks can be extended to Conditional Generative Adversarial Networks (cGANs) [85]. A cGAN is a type of GAN that uses labels or conditions to generate text or images that have characteristics similar to its training data set. The difference between a typical generative adversarial network (GAN) and a conditional GAN is that labelled data is used to provide context to a conditional GAN, allowing you to get better, more targeted results from the generator.

In cGANs, both the generator G and the discriminator D are conditioned on some extra information y, this conditioning can be achieved by feeding y as an additional input to both networks. The training process for cGANs involves a two-player minimax game similar to the original GANs but with the inclusion of conditional information y. The objective function is given by:

$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[\log D(x|y)] + E_{z \sim p_z(z)}[\log(1 - D(G(z|y)))] \tag{3.19}$$

In this formulation:

- The discriminator D tries to maximize the probability of correctly distinguishing real samples x from generated samples $G(z|y)$, conditioned on y.
- The generator G aims to minimize the probability that the discriminator correctly identifies the generated samples as fake.

By using conditional information, cGANs improve the ability of traditional GANs to produce more specific and contextually relevant outputs, making them valuable tools when precise and controlled data is required. However, training cGANs can be more complex due to additional conditional information. They also still face common GAN challenges such as training instability and mode collapse.

Chapter 4

Dataset

In the field of breast cancer research, there are several image datasets that researchers frequently employ. The selection of an appropriate dataset is an important step in the research process. This chapter aims to provide an overview of these datasets, evaluate their strengths and weaknesses, and then select the dataset for this research. The chapter will also delve into the details of data preparation, an essential step to ensure the quality and usability of the dataset.

4.1 Comparison of Datasets

This section describes some mammography datasets that are often employed in studies. By comparing these datasets a choice will be made for which dataset is going to be utilized in this study.

4.1.1 Digital Database for Screening Mammography

One of the most popular datasets is the Digital Database for Screening Mammography (DDSM). It was created in 1998 as a collaborative effort involving co-PI's at the Massachusetts General Hospital (D. Kopans, R. Moore), the University of South Florida (K. Bowyer), and Sandia National Laboratories (P. Kegelmeyer) [86]. The dataset contains 2,620 scanned film mammography studies, containing normal, benign and malignant cases [86]. Each study includes left and right cranio-caudal and mediolateral-oblique views. The dataset has been referenced in over 80 papers in the field of mammographic imaging.

Despite its popularity, the DDSM dataset has several constraints. Firstly, the dataset, being quite dated, is of low resolution. Secondly, the scanned images are stored in an outdated file format (LJPG), necessitating the use of antiquated decompression code to retrieve the data. Lastly, the segmentation labels, intended to outline the boundaries of lesions within the mammograms, are known to contain inaccuracies. These limitations hinder the DDSM dataset's reliability when creating tools that demand precise localization or high accuracy.

4.1.2 Curated Breast Imaging Subset of DDSM

In 2017, an updated subset of the DDSM database was introduced, known as the Curated Breast Imaging Subset of DDSM (CBIS-DDSM) [87]. This version removed low-quality images, re-annotated regions of interest which were deemed inaccurate, and re-categorized data into two groups. The CBIS-DDSM dataset contains 753 calcification cases, and 891 mass cases, all of which are in industry-standard DICOM files, with a corresponding file that contains accurate annotations and ground-truth labelling for each region of interest. The dataset contains a total of 2288 images. Based upon these improvements, the CBIS-DDSM dataset became more usable, as the accurate scanned film captures added much-needed analogue images into an otherwise digital training dataset.

One of the main criticisms of CBIS-DDSM is that the commitment to quality has resulted in a significant reduction in the overall size of the data set. This generally requires it to be combined with other datasets. Additionally, it only contains benign and malignant, no normal or non-cancerous cases.

4.1.3 INbreast

The INbreast dataset is a full-field digital mammographic repository made public by the Hospital de São João in Portugal in 2011 [88]. The dataset contains a total of 410 full-resolution digital mammographic images with polygonal segmentation within a separate XML file in OSIRIX format. The dataset contains four classes: mass, calcifications, asymmetries, and distortions, with no background tissue descriptors. Furthermore, all pathologies contained within INbreast are confirmed histologically, via a core biopsy or post-surgical specimen assessment.

INbreast has often been credited as the most reliable and precise open-source mammographic dataset. Despite its relatively small size, with only 410 full-resolution digital mammographic images, it has been highly valued in the research community. The authors of INbreast have not provided any specific licensing terms, and have made the dataset freely available to researchers and commercial vendors.

4.1.4 MIAS

The Mammographic Image Analysis Society (MIAS) dataset was one of the first public datasets [89]. It was published in 1994 and was generated by a consortium of UK research groups. The database includes 209 normal mammograms and 113 abnormalities with radiologist annotations, indicating the type of abnormality [90].

4.1.5 Evaluation of the Datasets

A summary of the characteristics of the datasets can be found in table 4.1. While DDSM contains the most images, the resolution of the images is very low. CBIS-DDSM contains a relatively high number of images, however, it only contains benign and malignant images. This limits its utility for this study, which aims to classify between cancerous and non-cancerous cases. Both the INbreast and MIAS datasets also contain good quality images, however, the limited number of images is a significant challenge in training a convolutional neural network. This highlights the difficulty in creating large, high-quality datasets.

Table 4.1: Summary of Key Characteristics of Datasets

Dataset	**Year**	**No. of Studies**	**No. of Images**	**Resolution**	**Cases Included**
DDSM	1998	6.775	10.239	Low	Normal, Benign, Malignant
CBIS-DDSM	2017	1.644	3.288	High	Benign, Malignant
INbreast	2011	115	410	High	Mass, Calcifications, Asymmetries, Distortions
MIAS	1994	161	322	Medium	Normal, Abnormal

Due to the limitations of these datasets, further exploration of alternative datasets is necessary for this research.

4.2 Dataset

The CBIS-DDSM dataset, despite being the best dataset upon evaluation, lacks normal, non-cancerous cases. While the MIAS and INBreast datasets do include non-cancerous cases, their limited size poses a challenge to training a deep convolutional neural network effectively.

To overcome these limitations, a new dataset was presented that combines all positive studies of CBIS-DDSM and all negative studies of DDSM [7]. The combined data set takes advantage of the strengths of both sources, providing a more comprehensive collection of mammographic images for the training of deep convolutional neural networks. In this dataset, the Region of Interests (ROIs) of the images have been extracted. The images have been pre-processed to convert them into a uniform size of 299x299 pixels. This standardization ensures consistency across the dataset, making it easier to train and evaluate the model [7].

For the negative images from the DDSM dataset, the preprocessing involved dividing the original larger images into 598x598 tiles. These tiles were then resized to 299x299 pixels. This tiling and resizing process helps to manage the image size while retaining important details that are necessary for accurate analysis [7]. By using this technique and extracting ROIs, multiple ROIs can be extracted from each image, effectively increasing the size of the dataset. For the positive images from the CBIS-DDSM dataset, the ROIs were extracted using predefined masks. These masks were applied to the images to isolate the regions containing abnormalities, with a small amount of padding added to provide additional context around the anomalies. This padding ensures that the extracted ROIs include enough surrounding tissue to help the model distinguish between normal and abnormal areas effectively. By combining the positive cases from CBIS-DDSM and the negative cases from DDSM, the resulting dataset provides a diverse set of images that enhances the training process. The use of ROIs and consistent image resizing ensures that the data set is well-prepared for deep learning applications [7]. Figure 4.1 shows images from the dataset.

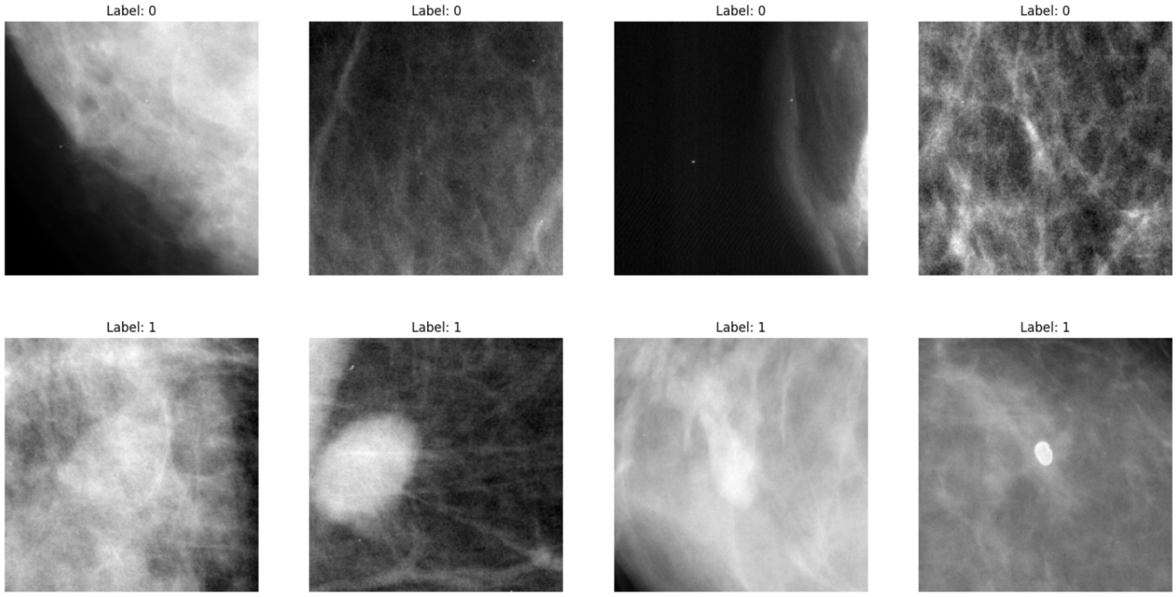


Figure 4.1: Mammographic images from the dataset. The top row is non-cancerous mammographs and the bottom row is cancerous mammographs [7]

Given the larger size of the original images, processing and training on these high-resolution images require substantial computational resources. To mitigate these demands and ensure efficient training, the images are down-sampled to a resolution of 128x128 pixels. Figure 4.2 shows the same images as 4.1 but downsized to 128x128 pixels. This reduction strikes a balance between maintaining enough detail for accurate analysis and maintaining manageable computational requirements. Despite the reduction in resolution, the important features necessary for distinguishing between cancerous and non-cancerous cases remain visible.

The images are labelled as negative or positive, the dataset contains 55.890 images however the distribution of these images is very skewed. Only 14% of the images are positive and the remaining 86% are negative. During the training of the convolutional neural network, the images will be divided into three distinct sets: training, validation, and testing, to ensure an evaluation of the model's performance.

To evaluate the ability of the model to generalise, a separate validation set has been chosen from an alternate database, MIAS. The MAIS dataset, like the primary one, also includes Region of Interest (ROI) images. The images from the MIAS dataset have been pre-processed to match the format used in our primary dataset. Specifically, the images have been resized to 128x128 pixels, ensuring consistency in the image dimensions and allowing a direct comparison of the model performance across different datasets. Some images can be found in figure 4.4. It is visible that the images look different in terms

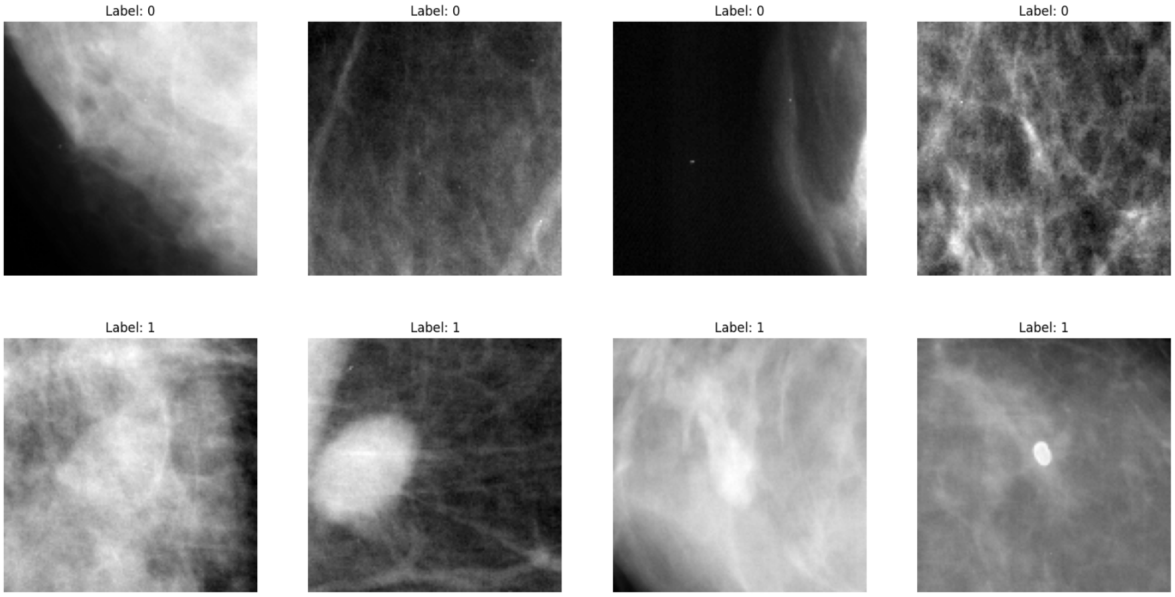


Figure 4.2: Mammographic images from the dataset after down-sampling to 128x128

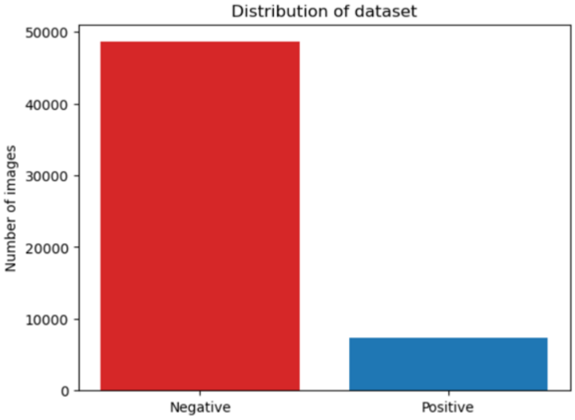


Figure 4.3: Distribution of dataset

of texture, contrast, and overall appearance compared to those from the primary dataset. Despite these differences, the goal is for the model to detect relevant features and abnormalities effectively, demonstrating its robustness and ability to generalize across various types of mammographic images.

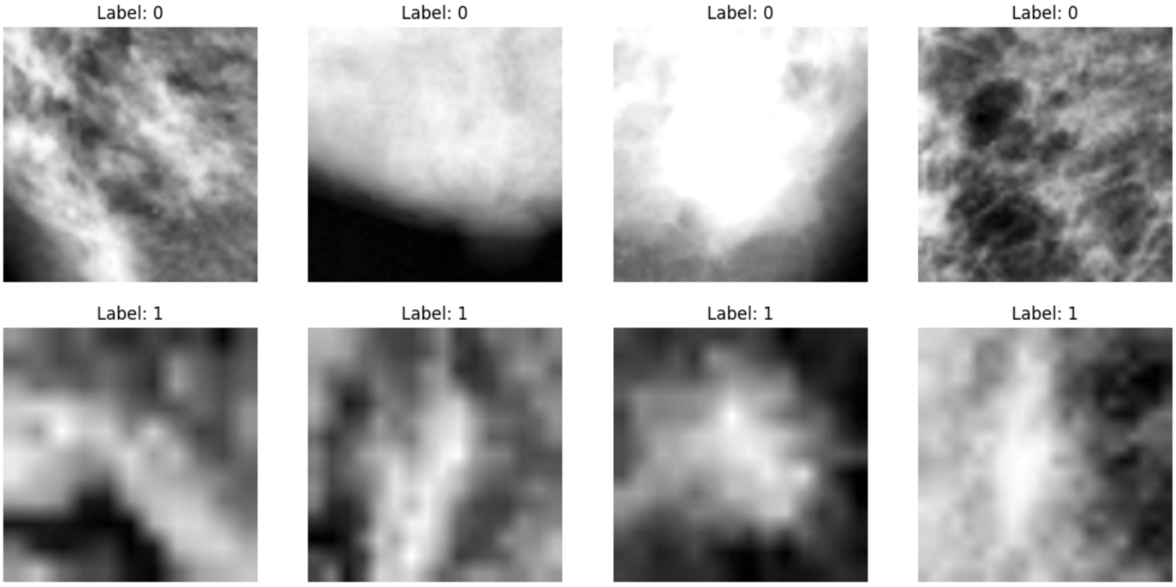


Figure 4.4: Mammographic images of alternative dataset

Chapter 5

Methodology

This chapter outlines the methodology employed for training and evaluating the performance of three convolutional neural network architectures: AlexNet, ResNet, and EfficientNet. It also details the data augmentation techniques implemented and describes the GAN used to generate synthetic images. Additionally, the chapter covers the model configurations, training procedures, and evaluation metrics.

5.1 Models

In this study, several Convolutional Neural Network architectures were utilized to classify mammography images as cancerous or non-cancerous. The chosen models are AlexNet, ResNet, and EfficientNet, these models have demonstrated their effectiveness in image classification tasks in the ImageNet Challenge. Furthermore, these models have shown promising results in breast cancer classification, as detailed in Section 4.2.6. This section will further go into the specifics of each model and its application in this study.

5.1.1 AlexNet

AlexNet, developed by Alex Krizhevsky and his team in 2012 [91], was one of the first CNNs to make a significant impact in the field of deep learning. The model has been used for various tasks, including image classification. In a study by Omonigho et al, AlexNet has been used to classify mammography images into two classes, benign and malignant, achieving an accuracy of 95.70% [70]. This shows that AlexNet can be a suitable model for classifying mammograms.

AlexNet contains a total of eight layers: five convolutional layers and three fully connected layers. The architecture can be found in figure 5.1. For the first two convolutional layers, each convolutional layer is followed by an Overlapping Max Pooling layer. The third, fourth and fifth convolution layers are directly connected. The fifth convolutional layer is again followed by Overlapping Max Pooling Layer, which is then connected to fully connected layers. The fully connected layers have 4096 neurons each and the second fully connected layer is fed into a softmax classifier having 1000 classes. [91]

Overlapping max pooling is a variation of the traditional max pooling operation used in CNNs. In overlapping max pooling, the pooling regions overlap, meaning that some pixels are considered in more than one pooling operation. This is achieved by setting the stride to be less than the size of the pooling window. The use of overlapping max pooling can help to reduce overfitting in CNNs. By considering some pixels in more than one pooling operation, the model can potentially learn more robust and diverse features from the input data. This can make the model more generalizable and less likely to overfit the training data [91]. Another key contribution of AlexNet is the use of the Rectified Linear Unit (ReLU) as the activation function, this helps to mitigate the vanishing gradient problem. AlexNet also implements dropout layers, which randomly ignore a subset of neurons during training [91]. This prevents overfitting by ensuring that the model does not rely too heavily on any single neuron.

For this study, AlexNet was adapted to accept one-channel grayscale images by modifying the first convolutional layer. The classifier was modified to output a single value for binary classification with a

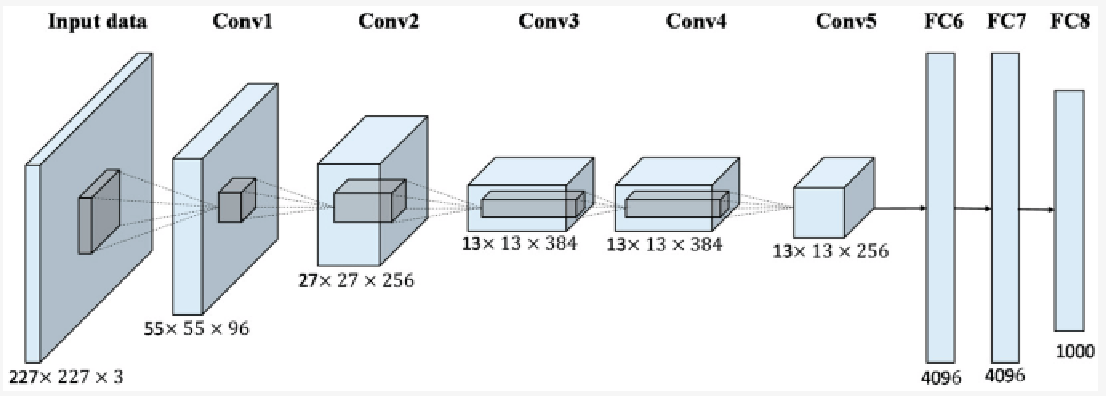


Figure 5.1: AlexNet Architecture [8]

Sigmoid activation function.

5.1.2 ResNet

ResNet, short for residual networks, was developed in 2016 by K. He and his team [66]. ResNet revolutionized deep learning by enabling the training of much deeper networks than previously possible. Like AlexNet, ResNet aims to solve the vanishing gradient problem. But instead of changing the activation function, ResNet introduced residual connections, also known as skip connections. A residual connection takes the activation from the $(n-1)$th convolution layer and adds it to the convolution output of the $(n+1)$th layer, it then applies ReLu un this sum and hereby skips the nth layer. Now, if the nth layer is not learning anything we won't lose any information.

The main component of a ResNet is the residual block. A residual block consists of multiple convolutional layers with a residual connection that directly connects the input of the block to the output. Each residual block has a convolutional layer, followed by a batch normalization and a ReLu activation. The batch normalization normalizes the output of the previous layer to speed up the training and improve the performance of the model. After the ReLu activation, there is again a convolutional layer, a batch normalization and another ReLu activation. The residual connection goes directly from the input of the block to the output. Residual blocks can be stacked together to create a deep neural network with hundreds of layers [66]. The architecture of a residual block can be found in figure 5.2.

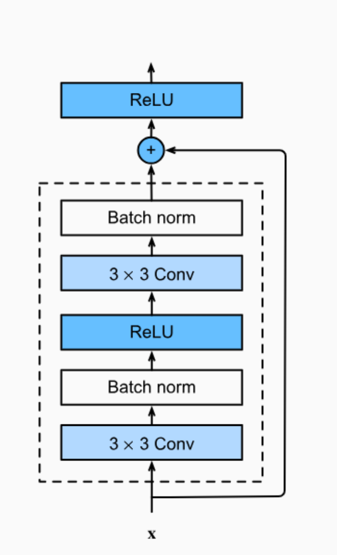


Figure 5.2: Residual Block [9]

There are several variants of the ResNet architecture; e.g. ResNet-18, ResNet-34, ResNet-50, and ResNet-101. Each model follows the same basic ResNet architecture but varies in depth. The choice of model depends on the complexity of the task and the available computational resources. For this study, ResNet50 was used. The first convolutional layer was adjusted to accept one-channel grayscale images, and the final fully connected layer was modified to output a single value for binary classification, followed by a Sigmoid activation.

5.1.3 EfficientNet

EfficientNet was introduced by Mingxing Tan and Quoc V. Le from Google Research in their 2019 [10]. EfficientNet presents a new approach to model scaling for convolutional neural networks. The process of scaling up CNNs is widely used to achieve better results. For instance, ResNets can be easily scaled in depth by adding more layers. A study has shown that it is to balance all dimensions of network width, depth, and resolution. The intuition behind this is that if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image [10]. EffiecientNet proposed a compound scaling method that uniformly scales width, depth, and resolution, with a set of fixed scaling coefficients [10].

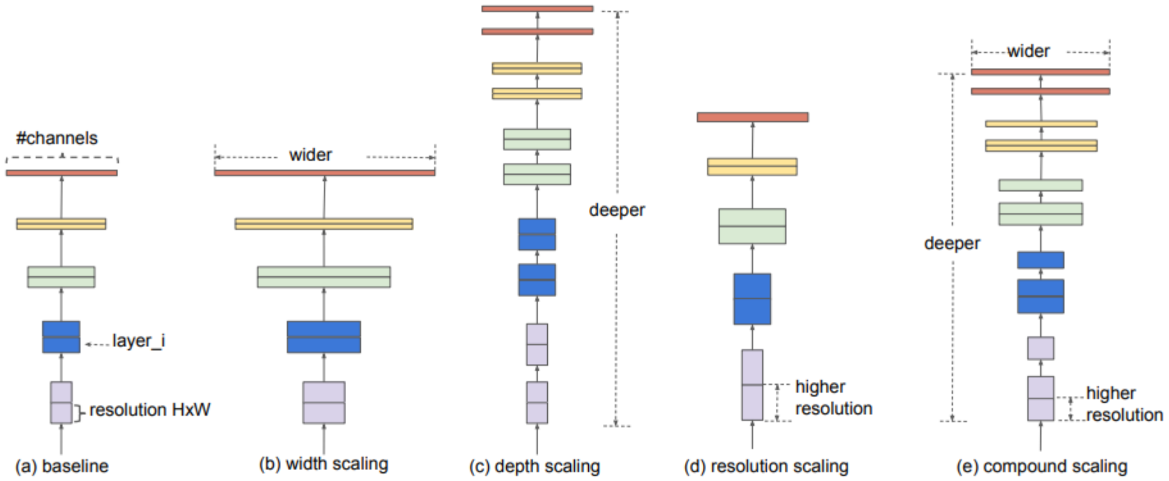


Figure 5.3: Model scaling: (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of the network width, depth, or resolution. (e) is the proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio. [10]

Since model scaling does not change the layer operators in the baseline network, having a good baseline network is also critical [10]. This study proposes a new baseline by performing a neural architecture search using the AutoML MNAS framework [92]. The framework optimizes both accuracy and efficiency to find a balance between the two. The resulting architecture uses Mobile Inverted Bottleneck Convolution (MBConv), this convolutional block was first introduced in MobileNetV2 [93]. In this block, the input is first expanded to a higher dimension, then a depth-wise convolution is applied, and then it is projected back to a lower dimension. In this structure, the input and the output are high-dimensional, and the computation is done in a lower-dimensional space. This makes the network computationally efficient [93].

There are different variants of EfficientNet, EfficientNet-B0 is the baseline model with moderate depth, width, and resolution. EfficientNet-B1 to B7 are all scaled versions of the model, they have increased depth, width, and resolution, which makes them more computationally intensive but also more accurate [10]. In this research, EfficientNet-B0 is utilized. The first convolutional layer was modified to accept grayscale images, and the classifier was adjusted to output a single value with a Sigmoid activation for binary classification.

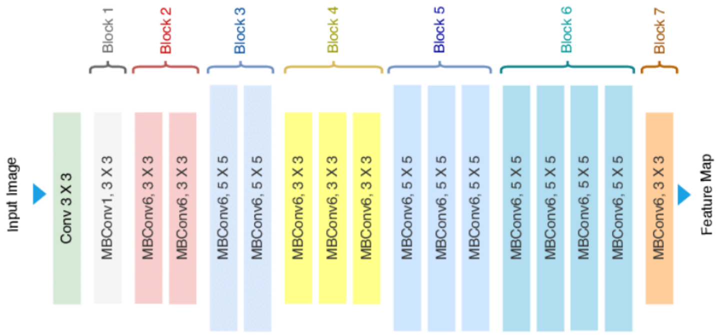


Figure 5.4: Architecture of EfficientNet-B0 with MBConv as Basic building blocks [11].

5.2 Data Augmentation and GAN Techniques

This study employs data augmentation and generative adversarial networks to enhance model performance and generalisation. Data augmentation introduces variability into the training data through geometric and photometric transformations. Generative Adversarial Networks are used to generate synthetic images to also introduce variability. This section details the specific data augmentation techniques employed, describes the GAN used to create synthetic images and outlines the strategy for incorporating both augmented and synthetic images into the dataset. Additionally, an overview of the experimental setup is provided to explain the approach and methodology.

5.2.1 Data Augmentation

Data augmentation involves applying various transformations to the training images to create new, varied examples, which helps improve the model's generalization. The augmentation techniques used in this study can be found in table 5.1.

Table 5.1: Data Augmentation Techniques and Parameters

Technique	**Description**	**Parameters**
Random Rotation	Rotate the image in a random direction.	Angle range: (-30°, 30°)
Random Flipping	Flips the image either horizontally or vertically at random.	Probability: 0.5
Translation	Shifts the image in both the horizontal and vertical directions.	Max shift: 20%
Gaussian Noise	Adds Gaussian noise to the image.	Standard deviation: 0.1
Brightness Adjustment	Changes the brightness of the image.	Factor: 1.2
Contrast Adjustment	Adjusts the contrast of the image.	Factor: 1.2

As explained in section 3.4 there are two types of data augmentations: geometric transformations and photometric transformations. The geometric transformations used in this study can be found in figure 5.5. The figure shows the application of geometric transformations on an image that shows a mass. In these augmented images, the mass remains visible but appears in different locations or orientations. This variation is beneficial because it helps the model learn to identify the mass regardless of its position, thereby improving the model's robustness and generalisation capabilities. By exposing the model to masses in various locations, geometric transformations ensure that the model can detect abnormalities more reliably across different contexts and orientations.

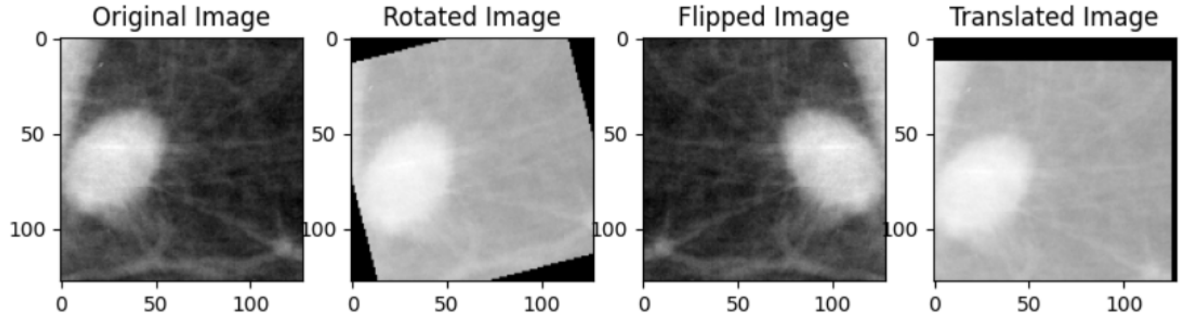


Figure 5.5: Geometric Transformations

Photometric transformations, involve altering the pixel values of the image without changing its geometry. These transformations include adjustments to brightness, contrast, and. Figure 5.6 illustrates the application of photometric transformations on the same original image. In these augmented images, the mass remains in the same location, but the colour and intensity values of the image are modified. This is advantageous because it helps the model become invariant to changes in lighting conditions and colour variations, ensuring that the detection of the mass is based on its intrinsic features rather than external factors. Photometric transformations enhance the model's ability to generalize across different imaging conditions, making it more robust in real-world scenarios where lighting and exposure can vary.

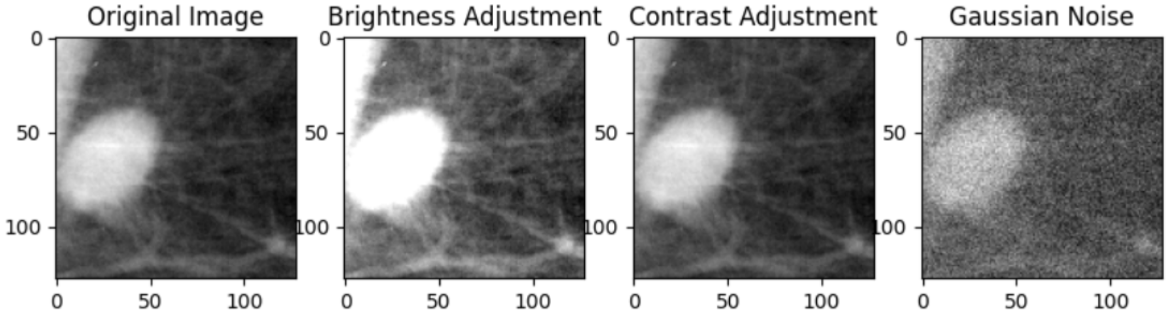


Figure 5.6: Photometric Transformations

Application of Augmented Images

The goal of adding augmented images is to enrich the training dataset with different transformations of existing images, thereby improving the model's ability to generalise across different scenarios, reduce overfitting, enhance its robustness in handling real-world variations, and mitigate data imbalance by providing additional instances of underrepresented classes.

To address the data imbalance, a strategy of data augmentation is employed, specifically targeting the positive class. This augmentation process involves generating additional augmented images based on the original ones, thereby increasing the size of the positive class in the dataset. Experiments will be conducted by adding 1, 2, and 3 augmented images per positive image, and then evaluating the performance of the model under each scenario to determine the most effective approach.

A potential risk associated with this strategy is that, since only the positive class is augmented, the model may learn to recognise the features of the augmented images rather than the cancerous features that are critical for accurate classification. To mitigate this risk, the model is also trained by adding augmented images to both classes. This process will be conducted in three stages, each with a different probability of an image generating an augmented image: 0.3, 0.6, and 1. This approach ensures that the model is exposed to a variety of training scenarios, which could improve its ability to generalise and accurately classify new, unseen images.

5.2.2 Generative Adversarial Networks

This research utilizes a Conditional Generative Adversarial Network to generate images conditioned on class labels. The cGAN framework consists of two neural networks: the Generator (G) and the Discriminator (D), which are trained adversarially. Unlike traditional GANs, cGANs incorporate class labels as an additional input to both networks, enabling the generation of images specific to the given labels. The generator and the discriminators both follow the guidelines of a deep convolutional GAN as described in section 3.5.6.

The generator uses a series of transposed convolutional layers to upsample the noise vector to the desired image dimensions. The architecture is designed to ensure that the generated images are realistic and are conditioned on the specified labels. The generator starts with an embedding layer that transforms the class labels into dense vectors of the same dimension as the noise vector. These label embeddings are then concatenated with the noise vector, this doubles the channel dimensions for the first layer of the network. This concatenated input is processed through several transposed convolutional layers, each followed by batch normalization and ReLU activation functions, except for the final layer, which uses a Tanh activation to output images in the range of [-1, 1].

Similar to the generator, the discriminator begins with an embedding layer for the class labels. These embeddings are concatenated with the input images along the channel dimension, this also increases the number of input channels for the first convolutional layer. The network consists of several convolutional layers, each followed by batch normalization and Leaky ReLU activation functions, except for the final layer, which uses a Sigmoid activation to produce a probability score.

- **Initialization:** The weights of the networks are initialized using a normal distribution with a mean of 0 and a standard deviation of 0.02. This initialization helps in stabilizing the training process and improving convergence.
- **Loss function:** Binary Cross-Entropy Loss is utilized for both the generator and the discriminator. This loss function is chosen because it is well-suited for binary classification problems, such as distinguishing between real and fake images.
- **Optimizer:** The Adam optimizer is employed with a learning rate of 0.0002 and betas (0.5, 0.999). The choice of the Adam optimizer is due to its adaptive learning rate properties, which help accelerate the convergence of the networks.
- **Learning Rate Scheduler:** A Learning Rate Scheduler is implemented for both the generator and discriminator optimisers, reducing the learning rate by a factor of 0.5 every 10 epochs. This dynamic adjustment helps maintain training stability and effective convergence.

Generative Adversarial Network Training

The dataset comprises images at a resolution of 128x128 pixels. While higher-resolution images can potentially yield more detailed and higher-quality generated outputs, training GANs on such images requires significant computational resources and can pose challenges in terms of training stability. To address these concerns and facilitate a more efficient and stable training process, the images are converted to a lower resolution of 64x64 pixels.

The Deep Conditional GAN was trained over 100 epochs with a batch size of 64. To monitor the progression of the generator's performance, fixed noise vectors and labels are used to generate images at the end of each epoch. This approach provides a consistent basis for visualizing improvements in the quality and realism of the generated images over time. Comparing these generated images across epochs allows for a clear observation of the model's learning dynamics.

The model is initialized with random weights. Figure 5.7 shows the images at the first epoch, after initialization.

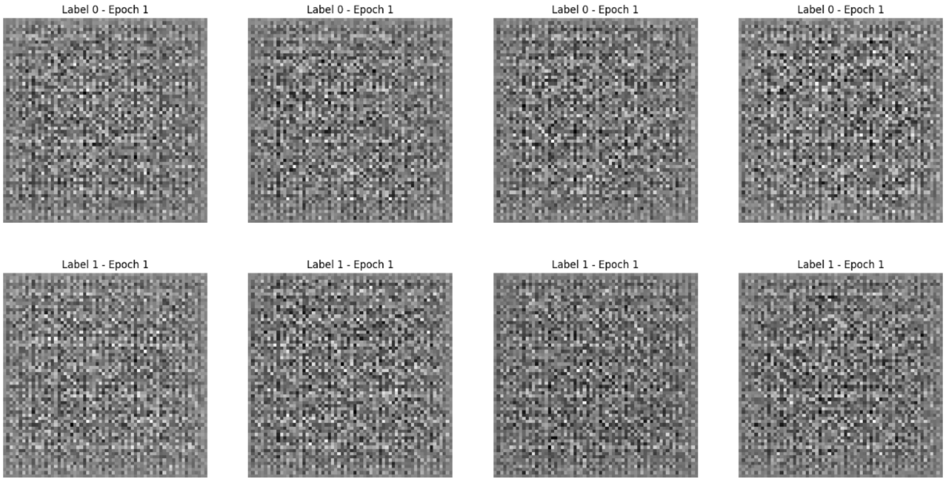


Figure 5.7: Random initialization of cGAN

The generated images after 100 epochs can be found in figure 5.8.

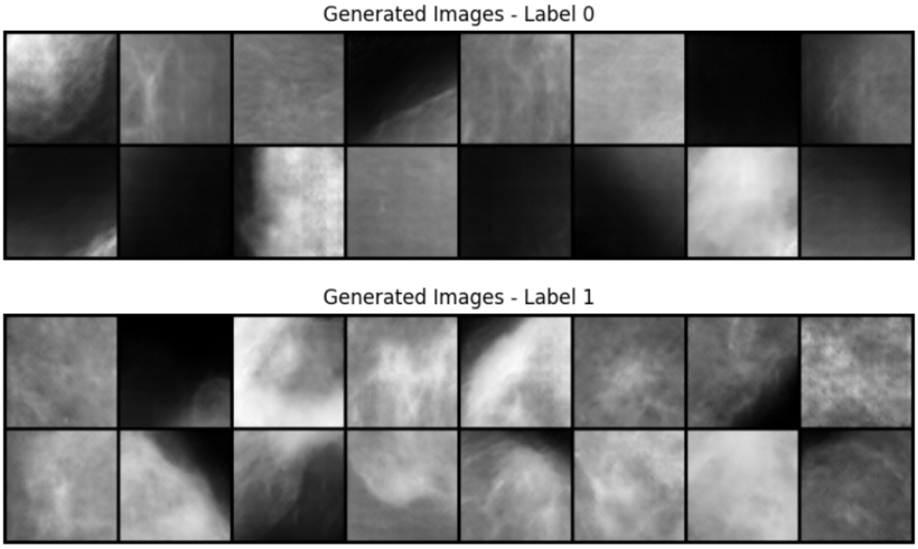


Figure 5.8: Generated images after 100 epochs

The training loss for both the generator and discriminator is tracked and plotted to analyze the training dynamics. A graph of the training loss helps in understanding how well the model is learning and whether it is converging towards an optimal solution. Figure 5.9 shows the generator and discriminator loss during training.

The loss graph for the GAN training shows the typical behaviour of both the generator and discriminator losses over time. Initially, both losses are high, with the generator loss above 20, indicating that the networks start with random weights and the discriminator easily distinguishes fake from real data. Shortly after, both losses decreased significantly, demonstrating that the networks are learning and adapting. The discriminator loss stabilizes around a low value, this shows its ability to differentiate between real and fake samples. The generator loss, while more variable, remains within a manageable range, indicating ongoing improvements in generating realistic data.

Throughout the training process, the generator loss fluctuates, which is a common trait in GAN training due to the adversarial nature of the setup. These fluctuations suggest that the generator continues to challenge the discriminator, while the discriminator also challenges the generator, maintaining dynamic learning. Although there are fluctuations in the generator loss, the relatively low and stable discrimina-

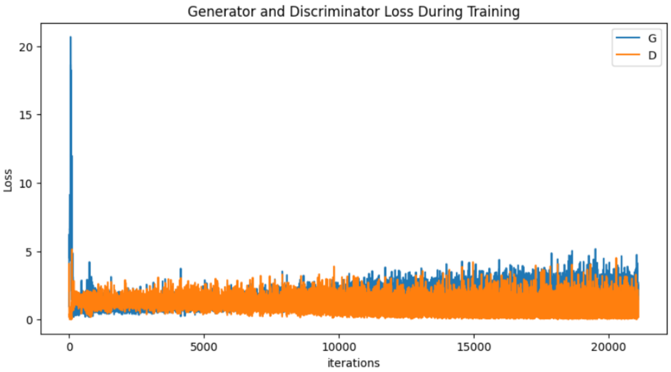


Figure 5.9: Generator and Discriminator loss during training of the GAN

tor loss suggests that the GAN is generally learning and improving its ability to create realistic images. When evaluating the images in figure 5.8, there are clear features of the mammographic training data, however, it is still possible to differentiate between the real and the generated images. This indicates room for further improvements in the training process of the GAN.

Application of Synthetic images

Similar to data augmentation, synthetic data is added to improve the model's ability to generalise, reduce overfitting, and mitigate data imbalance. The strategy for incorporating synthetic data mirrors the data augmentation approach. To address data imbalance, synthetic images are initially added to the positive class. Experiments are conducted by adding 1000, 5000, and 10.000 synthetic images of the positive class to evaluate the impact on the model's performance.

However, to mitigate the risk that the model will learn to detect synthetic images and classify them correctly based on their synthetic nature rather than the features of the positive class, synthetic images will also be added to both classes. This strategy ensures that the model does not become biased towards identifying synthetic images instead of learning the critical features necessary for accurate classification. Experiments are conducted by adding 1000, 5000, and 10.000 synthetic images to both classes to evaluate the impact on the model's performance.

5.3 Experimental Set Up

Extensive experiments were conducted using three state-of-the-art convolutional neural network architectures: AlexNet, ResNet, and EfficientNet. A description of the models and their key details can be found in table 5.2. Each model was trained with both random initialization and pretrained weights. The model that performs the best out of these models will be selected as a baseline model. During the rest of the study, this model will be trained with different augmented and synthetic data strategies to study the effects of these techniques.

Model	**Description**	**Key Details**
AlexNet	Early CNN model with deep architecture and ReLU activation. Modified for grayscale images. Binary classification output.	Five convolutional layers Three fully connected layers Softmax classifier adapted to Sigmoid
ResNet50	Deep network with residual connections to combat vanishing gradients. Residual blocks enhance learning. Adjusted for grayscale images; binary classification output.	Residual blocks with convolutional layers and batch normalization Sigmoid output for binary classification
EfficientNet-B0	Efficient CNN with compound scaling for balanced width, depth, and resolution. Uses MBConv blocks for efficiency. Adjusted for grayscale images; binary classification output.	Compound scaling for balance MBConv blocks Sigmoid output for binary classification

Table 5.2: Summary of Convolutional Neural Network Architectures

After the baseline model was set, a variety of data augmentation techniques were employed, including rotation, flipping, translation, brightness adjustment, contrast adjustment, and Gaussian noise. The strategy of how these augmentation techniques are added to the data are explained in section 5.2.1 and further summarized in table 5.3. This strategy will help to understand the influence of these techniques on model performance and robustness.

Strategy	**Description**
Add Augmented Images to Positive Class	Add 1, 2, and 3 augmented images per positive image
Add Augmented Images to Both Classes	Add an augmented image to both classes with probabilities of 0.3, 0.6, and 1.
Experiment Evaluation	Compare model performance under different augmentation techniques, amounts, and probabilities

Table 5.3: Augmented Data Strategy and Experimental Setup

To further enhance the diversity of the training data, Generative Adversarial Networks were incorporated to generate synthetic images. The selected baseline model was trained with different levels of synthetic data added to the dataset. This strategy is explained in section 5.2.2 and summarized in table 5.4.

Strategy	**Description**
Add Synthetic Images to Positive Class	1000, 5000, 10.000 synthetic images
Add Synthetic Images to Both Classes	1000, 5000, 10.000 synthetic images for each class
Experiment Evaluation	Compare performance with different quantities of synthetic images

Table 5.4: Synthetic Data Strategy and Experimental Setup

This comprehensive testing strategy enables a systematic evaluation of the impact of pretraining, data augmentation, and synthetic data generation on the performance of a neural network model in handling the given classification task.

5.3.1 Hyperparameters

The training of CNNs involves several hyperparameters as described in section 3.2.5.

- **Optimizer:** We used the Adam optimizer. Adam combines the benefits of two other extensions of stochastic gradient descent, namely AdaGrad and RMSProp. It computes adaptive learning rates for each parameter.
- **Learning Rate:** The initial learning rate was set to 0.001. While Adam uses this as a starting point, it adaptively adjusts the learning rates for each parameter during training.
- **Loss Function:** For the loss function, we employed a binary cross entropy loss with a positive weight to handle the class imbalance (BCEWithLogitsLoss). This loss function combines a Sigmoid layer and the binary cross-entropy loss in one single class.
- **Batch size:** A batch size of 32 was used. This size balances the need for efficient computation with the requirement to have diverse samples in each batch for stable training.
- **Number of Epochs:** The model will be trained for a total of 32 epochs. Training logs will be monitored to observe the model's performance and determine if it has converged. If the model has not converged after 32 epochs, the training duration will be extended. To enhance training efficiency, early stopping will be implemented. Specifically, if the validation loss does not improve for 10 consecutive epochs, the training process will be halted to prevent overfitting and save computational resources.

5.3.2 Dataset Split

The dataset will be split into three subsets: training, validation, and test sets, with a 60/20/20 ratio respectively. The sizes of each set can be found in table 5.5. To test how robust the model is, the model is also tested on a different dataset containing similar images.

Set	Number of images
Train	33531
Validation	11177
Test	11177

Table 5.5: Dataset Split

5.3.3 Performance Evaluation

The models will first be evaluated using performance metrics to measure how accurately they classify mammography images. Following this, their robustness will be evaluated to test how well they handle different conditions and variations in the data. This two-step evaluation approach provides a clear understanding of both the models' effectiveness and their ability to generalize to new, real-world situations. This section will detail the metrics used to evaluate the models' performance.

Model performance

Classifying images into cancerous and non-cancerous categories is a binary classification problem. In binary classification, the goal is to correctly identify each image as belonging to the correct class. A classification can have one of four outcomes:

- True Positives (TP): Instances correctly classified as positive.
- True Negatives (TN): Instances correctly classified as negative.
- False Positives (FP): Instances incorrectly classified as positive.
- False Negatives (FN): Instances incorrectly classified as negative.

These four items form the basis of a confusion matrix, which is a crucial tool in evaluating the performance of classification models and can be found in figure 5.10 [94].

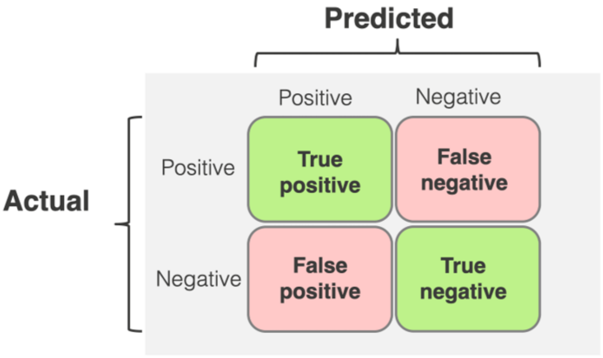


Figure 5.10: Confusion Matrix

The confusion matrix can be used to calculate several evaluation metrics. One of the most commonly used metrics is accuracy, accuracy measures the overall correctness of the model and is calculated with equation 5.1.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

While accuracy is intuitive, it can be misleading for imbalanced datasets where the number of instances in one class is much higher than the other.

Other common metrics are precision and recall. Precision measured the instances of true positives, amongst all the instances that are predicted as positive. It assesses, from all instances that are predicted positive, how many were correct. Recall measures the proportion of positive instances that were correctly identified by the model. It evaluates, from all positive instances, how many were detected by the model [94].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.3}$$

These measures have an inverse relationship, meaning improving one can lead to a decrease in the other. Finding the right balance between these metrics is crucial for developing effective classifiers. The F1-score is the mean of the precision and recall, this provides a single metric that balances both. The F1-score balances the trade-off between the two metrics, offering a useful metric when both false positives and false negatives need to be minimized [94]. This is particularly important in scenarios where the positive class is of high importance, such as in medical diagnoses, where identifying all cases of disease and ensuring those identified are correct is critical.

$$\text{F1-Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.4}$$

Balanced Accuracy, is the average of recall obtained on each class. This treats both classes equally regardless of their imbalance. This metric is useful when both classes are equally important. It ensures that the model's performance is evaluated equally across all classes. In medical diagnosis, this is important as both false negatives, missing a positive case, and false positives, misidentifying a negative case, have serious implications. Balanced accuracy prevents the model from being biased towards the majority class, which is a common issue in imbalanced datasets.

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{NP}{TN + FP}\right) \tag{5.5}$$

The choice between F1-score and balanced accuracy depends on the specific priorities of the classification task. If the main concern is to accurately identify positive cases and balance the trade-off between precision and recall, the F1-score is more informative. If the objective is to ensure equal performance across both positive and negative classes, balanced accuracy is a better metric. This metric is particularly useful in scenarios where the cost of misclassification is high for both classes, ensuring that the classifier is not biased towards the majority class and provides a fair assessment of its performance across all categories.

In summary, both F1-score and balanced accuracy are important metrics for evaluating classification models, especially when the dataset is imbalanced. The F1-score is more useful when the focus is on the positive class and balancing precision with recall, while balanced accuracy gives a fair assessment across all classes, making it valuable in applications where both classes are equally important. For breast cancer classification, where accurately identifying both cancerous and non-cancerous cases is crucial, using both metrics can provide a comprehensive evaluation of the model's performance.

Robustness performance

After evaluating the performance of the image classification models, it is important to also evaluate the model's robustness. A model's robustness refers to the model's ability to classify new, unseen data. It is important to evaluate the model's robustness to test how it would handle real-world applications. This section describes two methods to assess the robustness of the model, this includes adversarial attacks and testing on an additional dataset.

Adversarial Attack Adversarial attacks are techniques used to test the robustness of the model by adding small perturbations to the data [95]. The goal is to test whether the model is still able to classify the images after adding a small change to the data. If the model can still correctly classify these manipulated images, it demonstrates that the model is robust and can handle variations in the data.

A common adversarial attack method is the Fast Gradient Sign Method (FGSM) [95]. Fast Gradient Sign Method uses the gradients of the neural network's loss function with respect to the input image to create a new image that maximizes the loss. Using the gradients of the loss function with respect to the input image shows how each pixel in the image contributes to the increase in loss. This helps identify the weak spots of the model. By adding a small perturbation in the direction of these gradients, FGSM creates an adversarial image that maximizes the loss. This is essentially a deliberate attack on the model's vulnerabilities, causing the model to misclassify the adversarial image.

The perturbation that is added to the original image is calculated by:

1. **Forward Pass:** Pass the input image through the model to compute the loss based on the predicted class
2. **Compute the gradient:** Compute the gradient of the loss function, $J(\theta, x, y)$, with respect to the input image x, where θ is the model parameters, and y is the true label. The gradient shows how much the loss function changes with respect to small changes in the input.
3. **Generate Perturbation:** The perturbation η is computed by taking the sign of the gradient and scaling it by a small constant ϵ, this controls how big the perturbation is.
$$\eta = \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)) \tag{5.6}$$
4. **Generate Adversarial Image:** To create the adversarial image, the perturbation η is added to the original image x.
$$x' = x + \eta \tag{5.7}$$

FGSM is easy to implement as it only involves calculating the gradients and applying a linear transformation. This also makes it computationally efficient. In this study, an adversarial attack using FGSM will be performed after the models are trained to evaluate their robustness.

Figure **??** shows an example of an image with the perturbation and the adversarial image.

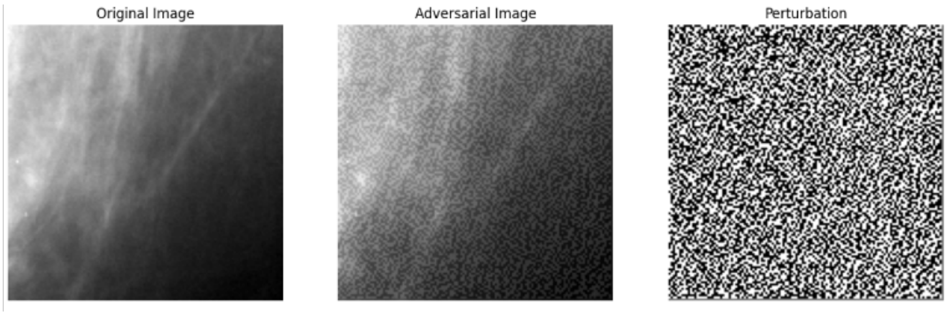


Figure 5.11: Adversarial Attack

Additional Dataset To further evaluate the robustness of the models, they are evaluated on an additional dataset. This dataset is extracted from the MIAS dataset as explained in section 4.2. Evaluating the models on this additional dataset provides insight into their ability to generalise to new, unseen data. This is crucial for understanding the model's performance in real-world scenarios. The models will also be evaluated on balanced accuracy and F1-score, this will highlight the difference between classifying the test set and the additional set.

If the model is able to correctly classify the images from the additional dataset, it can handle differences from real-world data.

Chapter 6

Results

This section presents the results of the experiments to evaluate the performance of various Convolutional Neural Network architectures in breast cancer classification. It explores the impact of data augmentation and synthetic data, generated by the Conditional Generative Adversarial Network, on the model's performance. Additionally, the robustness of the models is evaluated by adversarial attack and testing on an additional dataset.

6.1 Performance of Different CNN Architectures

Three different CNN architectures were evaluated to identify the most effective model for breast cancer classification. The evaluated models are AlexNet, ResNet and EfficientNet. The models are trained with random weight initialization and pretrained weight initialization.

Baseline Performance with Random Weight Initialization

The baseline performance of AlexNet, ResNet, and EfficientNet was evaluated using random weight initialization. The results are summarized in Table 6.1. The results indicate that ResNet and EfficientNet struggled to train effectively with random weight initialization, as reflected in their relatively low balanced accuracy scores. AlexNet outperformed both ResNet and EfficientNet, achieving an F1-score and balanced accuracy of 0.86 and 0.87, respectively.

Table 6.1: Balanced accuracy and F1-Score for different CNN Architectures with random weight initialization.

Model	F1-score	Balanced Accuracy
Alexnet	0.86	0.87
ResNet	0.81	0.50
EfficientNet	0.81	0.50

Baseline Performance with Pretrained Weight Initialisation

The performance of the models was further evaluated using pretrained weights through transfer learning. The results are summarised in Table 6.2. EfficientNet achieved the highest F1-score of 0.95 and a balanced accuracy of 0.93, demonstrating better performance in both metrics. AlexNet and ResNet also significantly improved the F1-score and balanced accuracy compared to the randomly initialized models. This highlights the effectiveness of transfer learning from pretrained models. The results indicate that EfficientNet outperformed both AlexNet and ResNet in terms of balanced accuracy and F1-score, making it the preferred choice for further experiments.

The comparison underscores the importance of pretrained models in medical image classification tasks. Pretraining enhances model performance by transferring knowledge from large, diverse datasets, en-

Table 6.2: Balanced accuracy and F1-Score for different CNN Architectures with pretrained weights.

Model	F1-score	Balanced Accuracy
AlexNet	0.91	0.83
ResNet	0.84	0.81
EfficientNet	0.95	0.93

abling better feature extraction and classification capabilities. In contrast, models with random weight initialisation may struggle initially with learning effective representations, leading to lower performance.

Confusion Matrix and Training Performance

To further analyse the performance of the best model, Figure 6.1 presents the confusion matrix. The confusion matrix provides a detailed view of the performance of the pretrained EfficientNet-B0 model, illustrating its ability to correctly classify the majority of breast cancer images.

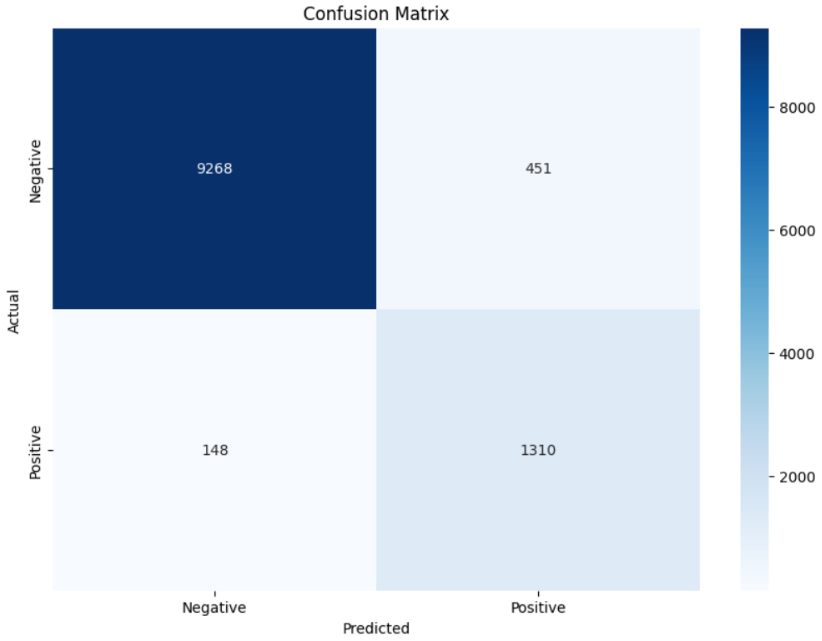


Figure 6.1: Confusion Matrix pretrained EfficientNet-B0

The confusion matrix provides a detailed overview of true positive, true negative, false positive, and false negative classifications, offering insights into the model's precision and recall across different classes.

Figure 6.2 illustrates the training performance of EfficientNet-B0 over 32 epochs, highlighting the convergence behaviour of the model and the effectiveness of the training process. The training and validation loss curves demonstrate a consistent decrease, indicating effective learning by the model. The balanced accuracy for both training and validation data sets increased steadily, suggesting that the model was improving its prediction capability without significant overfitting. Despite minor fluctuations in the validation metrics, the close convergence of training and validation accuracy demonstrates good generalization performance.

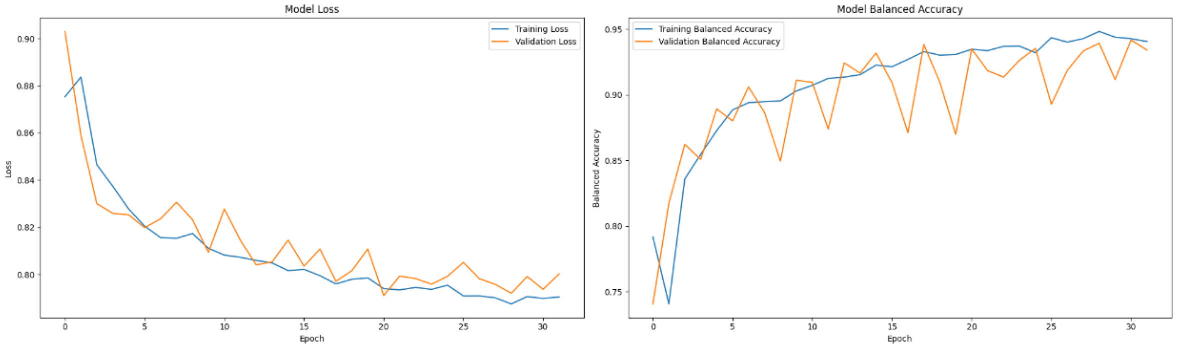


Figure 6.2: Training and Validation Performance of EfficientNet-B0 over 32 Epochs

Summary of Key Findings

The evaluation of CNN architectures for breast cancer classification showed that AlexNet outperformed ResNet and EfficientNet with random weight initialization, achieving an F1-score of 0.86 and a balanced accuracy of 0.87. However, when training the model with pretrained weights, EfficientNet scored the highest performance, with an F1-score of 0.95 and balanced accuracy of 0.93. This highlights the effectiveness of transfer learning. These findings show that EfficientNet resulted in good performance which will provide a strong starting point as a baseline model.

6.2 Impact of Data Augmentation on Model Performance

This section presents the impact of applying various data augmentation techniques on the performance of the EfficientNet model for breast cancer classification. The data augmentation techniques include rotation, flipping, translation, contrast adjustment, brightness adjustment, and Gaussian noise.

Addressing Class Imbalance: Addition of Augmented Images to Class 1

Augmented images were added to the positive class in the training set. Augmented images were added with varying numbers per original image: 1, 2, and 3. Table 6.3 presents the balanced accuracy of the pretrained EfficientNet with different data augmentation techniques applied.

Table 6.3: Balanced Accuracy with different data augmentation techniques applied on the positive class

Augmentations per image	Rotation	Flip	Translation	Brightness Adjustment	Contrast Adjustment	Gaussian Noise
1	0.90	0.94	0.93	0.92	0.91	0.90
2	0.88	0.93	0.88	0.90	0.91	0.88
3	0.89	0.91	0.84	0.91	0.90	0.86

For a single augmentation per image, the flip technique yields the highest balanced accuracy of 0.94, closely followed by Translation at 0.93. The other techniques also perform well, with Brightness Adjustment at 0.92, Contrast Adjustment at 0.91, and both Rotation and Gaussian Noise yield a balanced accuracy of 0.90.

When the number of augmentations per image is increased, there is a slight decrease in balanced accuracy across all techniques. The flip technique still performs the best with an accuracy of 0.93, contrast adjustment stays at 0.91, and brightness adjustment decreases to 0.92. while rotation and Translation both yield a balanced accuracy of 0.88. With three augmentations per image, the balanced accuracy further decreases for all techniques, except for brightness adjustment, which increases to 0.91. This is, together with the flip technique the highest accuracy at 0.91, while rotation yields an accuracy of 0.89. The translation, brightness adjustment, and contrast adjustment techniques all have similar accuracies ranging from 0.84 to 0.91.

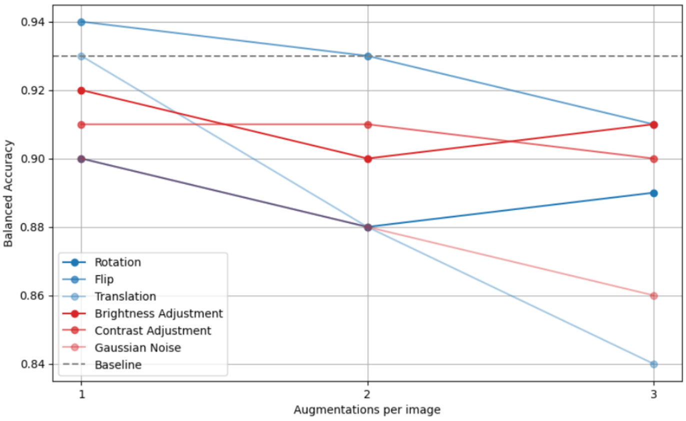


Figure 6.3: Balanced Accuracy of EfficientNet with different data augmentation techniques on the positive class

The results are visualised in figure 6.3. This figure clearly shows the trend of decreasing balanced accuracy with the addition of more augmentations. The results suggest that the Flip augmentation technique yields the best results. Furthermore, the results indicate that adding one augmented image per original image generally results in the best-balanced accuracy, with a noticeable performance drop when adding more than 1 augmented image per original image. This suggests a potential risk of overfitting when too many augmented images are used, possibly causing the model to learn overly specific features that do not generalise well to unseen data.

To further analyse the results, the F1-scores of each model are presented in table 6.4.

Table 6.4: F1-score with different data augmentation techniques applied on the positive class

Augmentations per image	Rotation	Flip	Translation	Brightness Adjustment	Contrast Adjustment	Gaussian Noise
1	0.96	0.96	0.97	0.96	0.93	0.96
2	0.96	0.97	0.96	0.97	0.96	0.96
3	0.95	0.97	0.95	0.96	0.95	0.95

When a single augmented image per positive image is added to the dataset, translation achieves the highest F1-score of 0.97. This is followed by rotation, flip, brightness adjustment and Gaussian noise, all at 0.96. The contrast adjustment achieved an F1-score of 0.93. All augmentation techniques were an improvement compared to the baseline model, except for contrast adjustment. When the number of augmented images per image is increased to two, flip, brightness adjustment, and Gaussian noise achieve an F1-score of 0.97. Rotation and translation maintain their score of 0.96, and contrast adjustment also improves to 0.96. All techniques improve the baseline model. With three augmented images per image, flip remains the highest at 0.97. Brightness adjustment achieves a high score of 0.96, and all the other techniques yield an F1-score of 0.95, which is the same as the baseline model.

The scores are visualised in figure 6.4. The figure clearly shows that the F1-scores have improved for all augmentation techniques.

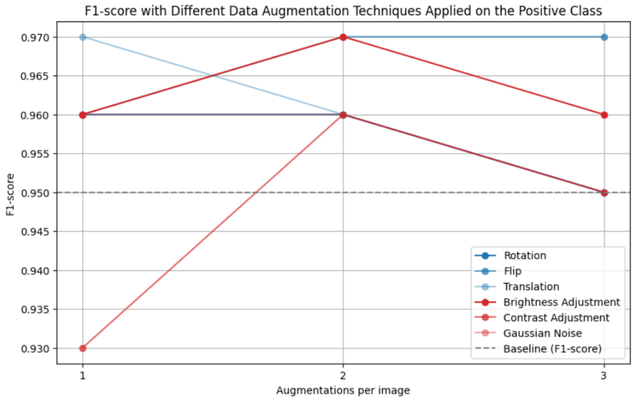
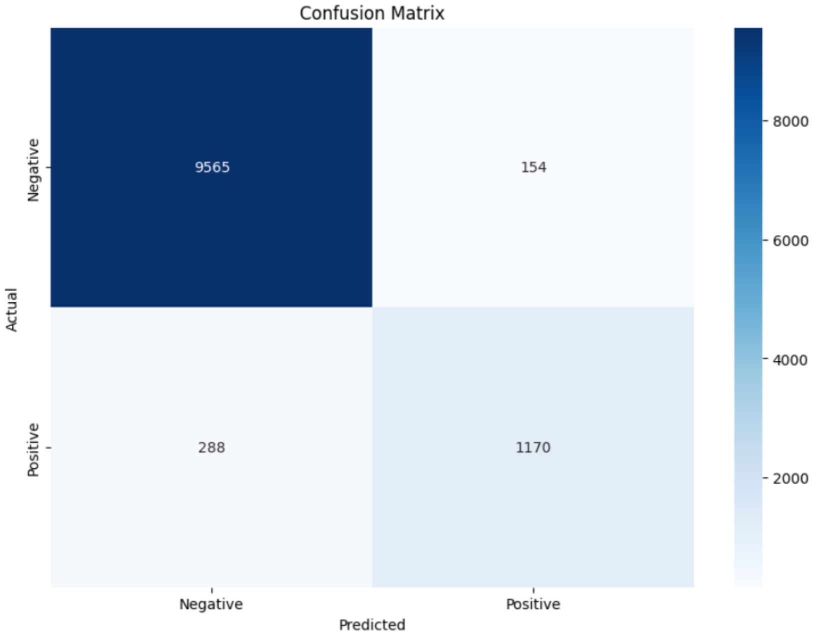


Figure 6.4: F1-score of EfficientNet with different data augmentation techniques on the positive class

Comparison F1-Score and Balanced Accuracy Figure 6.5 illustrated the confusion matrix of the model when one augmented image per image was generated using Gaussian noise. With this augmentation, the balanced accuracy decreased, while the F1-score improved compared to the baseline model. When comparing the confusion matrix of this model with the confusion matrix of the baseline, which can be found in figure 6.1, the augmented model showed a change in the distribution of prediction errors. The baseline model had fewer false negatives but more false positives. This indicates that while the augmented model reduced the number of incorrect positive predictions, it also failed to identify more positive images. While the F1-score improved due to better precision, the decrease in balanced accuracy shows the trade-off between correctly identifying all breast cancer cases, and not misidentifying images as positive which are not.

Figure 6.5: Confusion Matrix of EfficientNet trained with Gaussian Noise Data Augmentation



Preventing Overfitting: Balanced Addition of Augmented Images to Both Classes

To mitigate the risk of overfitting and create a more balanced model, the next phase of training involved adding augmented images to both positive and negative classes. This approach aimed to ensure that the model learns features of both cancerous and non-cancerous tissues, thereby enhancing its ability to generalise and make accurate predictions across all classes. By incorporating augmented data from both classes, the study aims to optimize the model's performance while minimizing the effects of overfitting and bias towards any specific class characteristics. In this phase, augmented images are added to both classes. This is done with a probability of 0.3, 0.6 or 1. The balanced accuracies of all models are presented in table 6.6.

Table 6.5: Balanced Accuracy with different data augmentation techniques applied on both classes

Augmentations per image	Rotation	Flip	Translation	Brightness Adjustment	Contrast Adjustment	Gaussian Noise
0.3	0.92	0.92	0.94	0.93	0.91	0.90
0.6	0.91	0.95	0.93	0.89	0.90	0.91
1	0.92	0.94	0.94	0.92	0.90	0.90

When augmented images are added to the dataset with a probability of 0.3, translation achieves the highest balanced accuracy of 0.94. This is followed closely by all other techniques, brightness adjustment at 0.93, rotation and flip both at 0.92, contrast adjustment at 0.91 and Gaussian noise at 0.90. Only translate improved the baseline and balanced accuracy achieved the same score. The other techniques resulted in a slight decrease in balanced accuracy compared to the baseline model.

When adding augmentations with a probability of 0.6, the flip technique improved to 0.95, while translation decreased slightly to 0.93. The other techniques resulted in balanced accuracies ranging from 0.89 to 0.91.

Adding an augmentation to every image, with a probability of 1, the flip and translation technique again yielded the best results with a balanced accuracy of 0.94. This is followed by a balanced accuracy of 0.92 for rotation and brightness adjustment and 0.90 for contrast adjustment and Gaussian noise.

These results are visualised in figure 6.6. In comparison to when data was only added to the positive class, these results are more stable and there is no sign of overfitting when more data is added to the dataset.

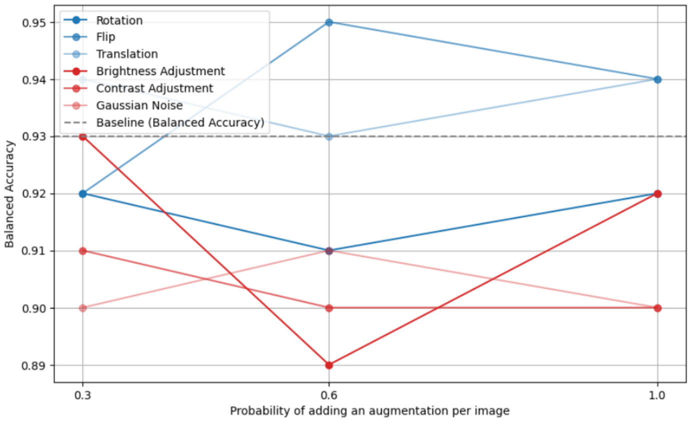


Figure 6.6: Graph visualization of the balanced accuracy of EfficientNet trained with different data augmentation techniques applied on both classes

To further analyse the effects of data augmentation, the F1-scores were compared to the baseline model. These results can be found in table 6.6.

When augmentations were added to the dataset with a probability of 0.3, the model showed strong performance for almost all techniques. The flip technique, again, showed the best performance followed by rotation, translation, Gaussian noise, and brightness adjustment. Contrast adjustment had a small degradation and achieved an F1-score of 0.97. When the probability of adding augmentations was

Table 6.6: F1-score with different data augmentation techniques applied on both classes

Augmentations per image	Rotation	Flip	Translation	Brightness Adjustment	Contrast Adjustment	Gaussian Noise
0.3	0.96	0.97	0.96	0.95	0.93	0.96
0.6	0.96	0.97	0.97	0.94	0.96	0.96
1.0	0.96	0.95	0.97	0.94	0.96	0.95

increased to 0.6, all the results remained very similar. Only brightness adjustment decreased to 0.95, translation increased to 0.97, and also contrast adjustment made a significant improvement. This is also similar to when the probability was increased to 1.0. These results are visualized in a graph in figure 6.7, this highlights that the results are fairly stable.

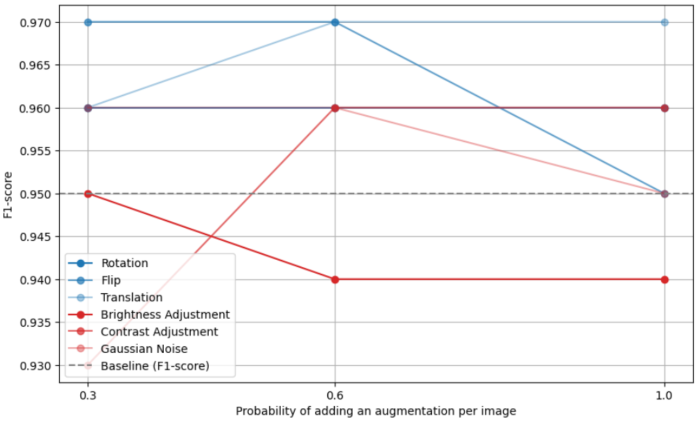


Figure 6.7: Graph visualization of the F1-score of EfficientNet trained with different data augmentation techniques applied on both classes

Summary of the Key Findings

The impact of data augmentation on EfficientNet's performance in breast cancer classification was analyzed using various techniques including rotation, flipping, translation, contrast adjustment, brightness adjustment, and Gaussian noise. When adding augmented images to the positive class, the flip technique provided the highest balanced accuracy and F1-score, particularly with one augmentation per image. Adding multiple augmentations per image led to a decrease in performance, indicating a risk of overfitting. When adding augmented images to both classes the balanced accuracy stayed more stable as more augmentations were added, this indicates less overfitting. The flip and translation techniques remained effective, achieving the highest balanced accuracies and F1-score. Overall, while data augmentation generally improved F1-scores, balanced accuracy slightly declined. highlighting the need for careful management of augmentation volume and type to optimize model performance.

6.3 Impact of Synthetic Data on Model Performance

In this section, the results are presented of the impact of adding synthetic data generated by a Generative Adversarial Network on the performance of the EfficientNet model for breast cancer classification. First, to solve class imbalance, images are added to the positive class. Then, to prevent overfitting, images are added to both classes.

Addressing Class Imbalance: Addition of Synthetic Images to Class 1

Synthetic images generated by the GAN developed in section 5.2.2 were added to the dataset. Different amounts of synthetic images were added. First, to solve the data imbalance, only images with label 1 are included. Table 6.7 presents the performance of the EfficientNet model trained with different amounts of synthetic data added to the dataset.

Table 6.7: Balanced accuracy of EfficientNet with synthetic images added to the positive class

Number of Synthetic Images Added	F1-score	Balanced Accuracy
1000	0.97	0.94
5000	0.96	0.93
10.000	0.96	0.92

When 1000 synthetic images were added to the dataset, the model's performance improved, achieving an F1-score of 0.97 and a balanced accuracy of 0.94. This indicates a slight improvement in both metrics compared to when no augmented or synthetic data is added. However, as the number of synthetic images increased to 5000, the F1 score decreased slightly to 0.96, and the balanced accuracy returned to the baseline level of 0.93. When 10.000 synthetic images were added, the F1-score remained at 0.96, while the balanced accuracy decreased further to 0.92.

The results indicate that adding synthetic images slightly improved the performance of the EfficientNet model. The addition of 1000 synthetic images improved both the F1-score and balanced accuracy. However, adding more synthetic images did not yield further improvements and slightly decreased performance. This decline indicates that, while synthetic data can be beneficial, there is a point at which adding too many synthetic images may lead to worse returns. This could be due to the model overfitting to the synthetic data or the synthetic data not being diverse enough to provide additional value.

Preventing Overfitting: Balanced Addition of Synthetic Images to Both Classes

To prevent overfitting on 1 class, synthetic images of both classes were added next. The results can be found in table 6.8.

Table 6.8: F1-score and Balanced Accuracy of EfficientNet with Synthetic Images Added to Both Classes

Number of Synthetic Images Added	F1-score	Balanced Accuracy
1000	0.95	0.93
5000	0.96	0.90
10.000	0.95	0.91

When 1000 synthetic images were added to both classes, the model's F1-score matched the baseline at 0.95, and the balanced accuracy also matched the baseline at 0.93. This indicates that the introduction of a moderate amount of synthetic data to both classes maintained the performance levels achieved by the baseline model.

With an increase to 5000 synthetic images, the F1-score improved slightly to 0.96, but the balanced accuracy decreased to 0.90. This suggests that while the model's precision and recall improved, the overall balance between classes was negatively impacted. When 10.000 synthetic images were added, the F1-score returned to the baseline level of 0.95, while the balanced accuracy was 0.91. This showed a slight improvement over the 5000 synthetic image scenario but still did not improve the baseline performance.

The performance metrics of adding synthetic data to both the positive class, and both classes, can be found in 6.8. From this figure, it is clear that while the F1-score improves by adding synthetic data, the balanced accuracy decreases slightly most of the time.

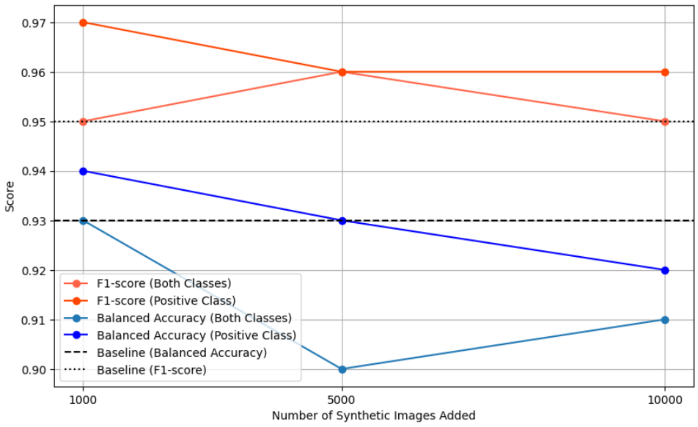


Figure 6.8: Graph visualization of the balanced accuracy and F1-score of EffiecientNet with different amounts of synthetic images added to the dataset

Summary of Key Findings

The impact of synthetic data on the performance of the EfficientNet model for breast cancer classification was evaluated by adding synthetic images generated by a GAN. When synthetic images were added exclusively to the positive class, the model showed improved performance with 1000 synthetic images, achieving an F1-score of 0.97 and a balanced accuracy of 0.94. However, adding more synthetic images led to a decrease in balanced accuracy, indicating that excessive synthetic data could potentially cause overfitting.

When synthetic images were added to both classes, the model's performance varied. With 1000 synthetic images for each class, performance was stable and matched baseline levels. Adding 5000 synthetic images improved the F1-score to 0.96 but decreased balanced accuracy to 0.90. Adding 10.000 images resulted in an F1-score of 0.95 and balanced accuracy of 0.91. Overall, while synthetic data generally improved F1-scores, it slightly reduced balanced accuracy.

6.4 Robustness Evaluation

In this section, the models are evaluated on their robustness to test how they would perform with the variability of real-world cases. First, an adversarial attack is performed, and then the model is evaluated on a separate dataset.

Adversarial Attack

The models are evaluated based on the Adversarial Attack as described in section 5.3.3. First, the models trained with augmented data are evaluated, and then the models trained with synthetic data are evaluated.

Augmented Data After the adversarial attack was performed, the models were re-evaluated based on balanced accuracy and F1-score. This section will highlight the results of the models that added one augmented image to the positive class, as these models performed the best out of the augmented models. The balanced accuracy before and after the attack can be found in table 6.9. The table also shows the percentual difference after the attack was performed.

The balanced accuracy of the baseline model dropped from 0.93 to 0.53, which is an 43.01% decrease. The decrease indicates a significant impact of the adversarial attack on the baseline model. The balanced accuracy of all other models decreased after the adversarial attack, however, the reductions were significantly less compared to the baseline model. For instance, the adversarial attack reduced the performance of the rotate augmentation by 25.00% and the flip augmentation reduced by 16.84%. The contrast adjustment technique had the smallest drop of 6.59%, and the Gaussian Noise had the largest

Table 6.9: Balanced Accuracy and Adversarial Balanced Accuracy Percentual Differences for Augmented Data

Model	Balanced Accuracy	Adversarial Balanced Accuracy	BA Diff (%)
Baseline	*0.93*	*0.53*	*-43.01*
Rotate	0.92	0.69	-25.00
Flip	0.95	0.79	-16.84
Translate	0.94	0.65	-30.85
Brightness	0.93	0.63	-32.26
Contrast	0.91	0.85	-6.59
Gaussian	0.91	0.53	-41.76

drop of 41.76%. This is a significant decrease, but still less than the baseline.

Table 6.10: F1-Score and Adversarial F1-Score Percentual Differences for Augmented Data

Model	F1-Score	Adversarial F1-Score	F1 Diff (%)
Baseline	*0.95*	*0.56*	*-41.05*
Rotate	0.97	0.90	-7.22
Flip	0.97	0.94	-3.09
Translate	0.96	0.89	-7.29
Brightness	0.95	0.82	-13.68
Contrast	0.93	0.94	1.08
Gaussian	0.96	0.70	-27.08

Table 6.10 describes the F1-score before and after the adversarial attack, along with the percentage differences. The baseline model's F1-score dropped significantly from 0.95 to 0.56, resulting in a 41.05% decrease. This large decrease again highlights the substantial impact of the adversarial attack on the baseline. In terms of the F1-score, the model using the contrast adjustment technique experienced a slight increase of 1.08%. All other models saw a decrease, but these were significantly less than the baseline's decline.

Overall, while the performance metrics for all models have decreased due to the adversarial attack, the augmented models show improvement over the baseline. This indicates that the applied augmentations contribute positively to the models' robustness against adversarial attacks.

Synthetic Data Next, the models with synthetic data were re-evaluated after performing the adversarial attack. Table 6.11 shows the balanced accuracy before and after the adversarial attack, along with the percentage difference.

Table 6.11: Balanced Accuracy and Adversarial Balanced Accuracy Percentual Differences for Synthetic Data

Model	Balanced Accuracy	Adversarial Balanced Accuracy	BA Diff (%)
Baseline	*0.93*	*0.53*	*-43.01*
Positive - 1000	0.94	0.84	-10.64
Positive - 5000	0.91	0.79	-13.19
Positive - 10.000	0.92	0.84	-8.70
Both - 1000	0.93	0.51	-45.16
Both - 5000	0.89	0.62	-30.34
Both - 10.000	0.91	0.56	-38.46

Again, there is a significant drop in balanced accuracy after the adversarial attack. Interestingly, the decrease for models where data were added to the positive class ranged from 8.70% to 13.19%, while when adding to both classes the decrease ranged from 30.34% to 45.16%.

The F1-scores of the models with synthetic data were also evaluated, these results can be found in table 6.12.

Table 6.12: F1-Score and Adversarial F1-Score Percentual Differences for Synthetic Data

Model	F1-Score	Adversarial F1-Score	F1 Diff (%)
Baseline	*0.95*	*0.56*	*-41.05*
Positive - 1000	0.97	0.95	-2.06
Positive - 5000	0.95	0.94	-1.05
Positive - 10.000	0.96	0.95	-1.04
Both - 1000	0.95	0.81	-14.74
Both - 5000	0.96	0.85	-11.46
Both - 10.000	0.96	0.57	-40.62

The models with synthetic data added to the positive class significantly improved compared to the baseline. While there is still a slight decrease after the adversarial attack, the F1-score ranges from 0.94 to 0.95 which is similar to the baseline evaluation without adversarial attack. When synthetic images are added to both classes, the robustness performance decreases a bit more. Adding 1000 and 5000 images results in a decrease of 14.74% and 11.46% respectively, while adding 10.000 images results in a 40.62% decrease.

Comparison to Additional Dataset

To further test the generalizability of the model, the model was evaluated on a different dataset. This dataset also contains ROIs of mammograms. The top-performing models from each strategy were assessed: the baseline EfficientNet with pre-trained weights, the model with flip data augmentation to the positive class, the model with data augmentation to both classes, the model with synthetic data added to the positive class, and finally the model with synthetic data added to both classes. The results are presented in table 6.13.

Model	**F1-score**	**Balanced Accuracy**
Pretrained EfficientNet	0.43	0.45
Augmented images positive	0.56	0.51
Augmented images both	0.35	0.50
Synthetic images positive	0.16	0.50
Synthetic images both	0.28	0.50

Table 6.13: Performance metrics of trained models evaluated on an additional dataset

The baseline EfficientNet model with pretrained weights achieved an F1-score of 0.43 and a balanced accuracy of 0.45. These results serve as a reference point for evaluating the impact of data augmentation and synthetic data.

The EfficientNet model trained with augmented images is also tested on the additional dataset. The best model achieved an F1-score of 0.56 and a balanced accuracy of 0.51. Although this is an improvement compared to the baseline model, the model is unable to correctly classify the images. Figure 6.9 presents the confusion matrix for the model.

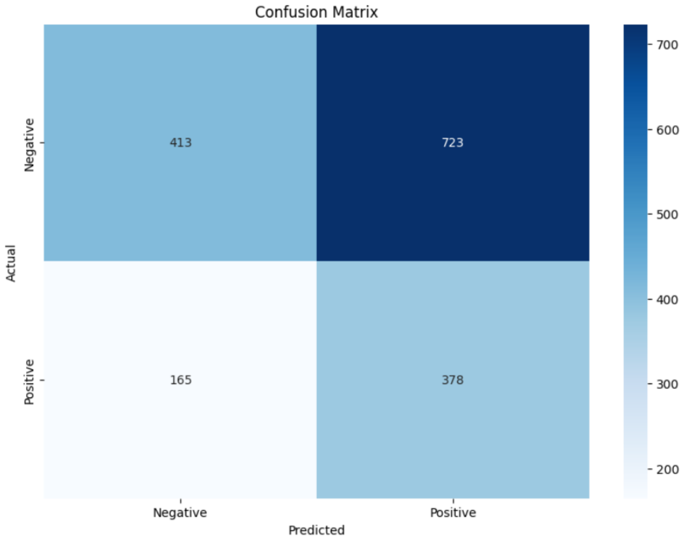


Figure 6.9: Confusion matrix of EfficientNet with augmented data evaluated on the additional MIAS dataset

When synthetic images generated by GAN were added, the performance decreased. The model has an F1-score of 0.28 and a balanced accuracy of 0.50. Upon further inspection of the results, it is shown that the model classifies almost all images as positive. This results in a decrease in performance metrics. The confusion matrix can be found in 6.10.

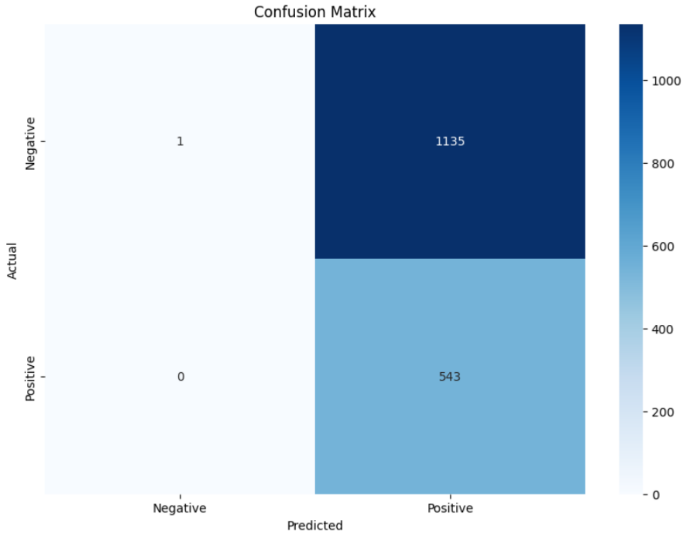


Figure 6.10: Confusion matrix of EfficientNet with synthetic data evaluated on the additional MIAS dataset

Summary of Key Findings

The robustness evaluation provides insights into how the models perform under variable circumstances. The adversarial attack led to a significant decrease in the performance of the baseline model, with the balanced accuracy and F1-score dropping 43.01% and 41.05% respectively. While the performance of the models with augmented and synthetic images also decreased, it was a lot less compared to the baseline model. The models that performed the best, are contrast adjustment and adding 10.000 synthetic images to the dataset. Contrast adjustment resulted in a decrease of 6.59% in balanced accuracy, but an increase of 1.08% in terms of F1-score. Adding 10.000 synthetic images resulted in a decrease in balanced accuracy of 8.70% and a decrease in F1-score of 1.04%. While the adversarial attack showed a decrease in performance, this decrease was significantly less when data augmentation and synthetic data were added to the dataset.

Evaluation of an additional dataset also provides valuable insights into the generalisability of the model. The results suggest that, while the model trained with augmented and synthetic data performed well on the training data, it did not generalise as effectively to the additional dataset. The baseline model yielded an F1-score and balanced accuracy of 0.43 and 0.45 respectively. This is a large drop compared to the original dataset. While adding augmented images to the positive class led to an improvement in the F-score and balanced accuracy of 0.56 and 0.51, this is not a sufficient performance. This decrease in performance could be due to the additional dataset having different characteristics compared to the training data. While both datasets contain images of the Region of Interest, differences in imaging techniques or equipment can cause the images to appear visually different. These variations make it challenging for the model to accurately detect breast cancer in the new dataset, highlighting the importance of the variability of the dataset when training and evaluating models for medical image analysis.

Chapter 7

Discussion and Conclusion

This section provides an in-depth analysis of the results, examines the study's limitations, suggests directions for future research, and presents a conclusion.

7.1 Discussion

In recent years, the use of Convolutional Neural Networks has shown significant promise in medical image analysis. However, the results thus far are limited due to a lack of data caused by privacy regulations, the lack of accurate annotations and the lack of variability in the data. To address these challenges, this research explores the impact of data augmentation and synthetic data on the performance of CNN models in breast cancer detection. The study aims to answer the following question:

How does the use of augmented and synthetic data impact the effectiveness of convolutional neural networks in detecting breast cancer from mammograms?

To address this question, the research is structured around four subquestions, each focussing on different aspects of data augmentation and synthetic data generation. This discussion will examine the findings for each subquestion, highlighting the performance of various CNN architectures, the effects of data augmentation techniques, the impact of synthetic data generated by Generative Adversarial Networks, and the generalisability of these models when evaluated on a separate dataset.

Subquestion 1: Performance of Different CNN Architectures

What are the performances of different CNN architectures in detecting breast cancer from mammographic images?

To address this question, the study evaluated various CNN architectures, including AlexNet, ResNet, and EfficientNet. EfficientNet demonstrated the highest performance with pretrained weights, achieving a balanced accuracy of 0.93 and an F1-score of 0.95. These results highlight its potential for detecting breast cancer from mammographic images.

Subquestion 2: Impact of Data Augmentation

How do traditional data augmentation techniques affect the performance of CNN models?

To answer this question, different types of data augmentation techniques were applied and added to the training set. The techniques include rotation, flipping, translation, brightness adjustment, contrast adjustment and the adding a Gaussian noise. The results showed that adding the flipping technique yielded the best results with a balanced accuracy and F1-score of 0.94 and 0.97 respectively. The study found that adding too many augmented images per original image led to a decrease in performance metrics. This is likely due to the risk of overfitting on the augmented data.

In conclusion, data augmentation can positively impact the performance of a CNN if applied correctly and in appropriate quantities. However, excessive augmentation can lead to a decrease in performance.

Therefore, it is important to balance the quantity and quality of augmented data to optimise the model's performance.

Subquesion 3: Impact of Synthetic Data

textitHow does the inclusion of synthetic data generated by GANs impact the performance of CNN models in breast cancer detection?

The study explored the effects of adding synthetic images created by Generative Adversarial Networks into the training process. Adding a moderate amount of synthetic data (1000 images) to the positive class resulted in an F1-score of 0.97 and balanced accuracy of 0.94, this a slight improvement in model performance compared to the baseline. When more images were added, it led to a small decrease in performance, with the F1-score stabilising around 0.96 and the balanced accuracy decreasing to 0.92. This suggests that while synthetic data can be beneficial, excessive amounts may cause overfitting or insufficient diversity. These results are very similar to when adding augmented data, although synthetic data did not significantly enhance the performance of the EfficientNet model, it showed a small improvement with the right amount of synthetic images. Therefore, the study concludes that the strategic use of synthetic data could potentially enhance the performance of CNN models in breast cancer detection.

Subquestion 4: Generalizability of CNN Models

How does the inclusion of augmented and synthetic images impact the generalisability of CNN models for breast cancer detection?

This was evaluated by an adversarial attack and by evaluating the model with an additional dataset. Adversarial attacks led to a significant decrease in both balanced accuracy and F1-scores for all models. The baseline showed a large drop in performance. In contrast, the addition of augmented and synthetic data resulted in a significantly smaller drop. This suggests that incorporating augmented and synthetic data can enhance model robustness.

Evaluation of an additional dataset also provides valuable insights into the generalisability of the model. The results suggest that, while the model trained with augmented and synthetic data performed well on the training data, it did not generalise as effectively to the additional dataset. This is indicated by the decrease in the F1 score and the balanced accuracy when the model was evaluated in a separate data set. Differences in imaging techniques or equipment can cause the images to appear visually different, making it challenging for the model to accurately detect breast cancer in the new dataset. This highlights the importance of the variability of the dataset when training and evaluating models for medical image analysis.

Answer to the Main Research Question

This research studied the effects of augmented and synthetic data on the effectiveness of convolutional neural networks in breast cancer detection. Although EfficientNet showed high baseline performance, both traditional data augmentation and synthetic data techniques contributed to further improvements. The study found that from the traditional augmentation techniques, the flip technique was most effective in improving model performance. It also showed the addition of synthetic images in the positive class can enhance model performance. To optimize the results, is important to balance the amount of augmented and synthetic data to avoid overfitting and maintain diversity, ensuring optimal model performance. In addition, the results of the adversarial attack showed that the use of augmented and synthetic data has significantly enhanced the robustness and generalisation ability of convolutional neural networks. Overall, the research demonstrated that data augmentation and synthetic data can be effective in enhancing the performance of CNN models for breast cancer detection.

7.2 Conclusion

This study explored the impact of data augmentation and synthetic data on the effectiveness of convolutional neural networks in detecting breast cancer from mammographic images. Early detection of breast cancer is crucial, as survival rates decline significantly as the disease progresses. While computer aided diagnosis systems show promise in assisting with breast cancer detection, currently the performance is limited and CAD systems are not widely used in clinical practice. A challenge in developing effective CAD systems is the need for large, high-quality datasets. Obtaining such datasets is challenging due to several factors, including privacy regulations, data annotation, and the variability in mammographic images. These challenges make it difficult to obtain a large, well-annotated dataset. This research aimed to address these challenges by investigating whether augmenting data with traditional techniques and generating synthetic data using Generative Adversarial Networks could enhance the performance and robustness of CNN models.

The study demonstrated that both traditional data augmentation techniques and synthetic data generated by GANs can significantly enhance the performance and robustness of CNN models, particularly the EfficientNet model. The addition of augmented and synthetic data led to an improvement of 1% in balanced accuracy and 2% in F1-score. Additionally, it showed significant improvements in the robustness of the model. While the performance decreased after the adversarial attack was performed, it was significantly less than the baseline model. This indicates an improvement in the robustness of the model.

Every year, approximately 900.000 women participate in the population-based breast cancer screening program, and 7000 women are diagnosed with cancer. With an improvement of 1% in balanced accuracy and 2% in F1-score, the use of data augmentation and synthetic data could result in more accurate detections and fewer missed cases. This can potentially lead to earlier interventions for hundreds of women. While this improvement seems small, it can have substantial real-world implications, potentially saving lives and reducing delayed diagnoses. The results also show that data augmentation and synthetic data have promise for improving the generalisation ability of CNNs in breast cancer detection. While current performance levels are still not sufficient for full clinical adoption, these techniques show potential for further research and development, which could lead to more effective and reliable CAD systems in the future.

In conclusion, this thesis has demonstrated that augmented and synthetic images can positively impact CNN performance in breast cancer detection. While these techniques offer promising improvements, further research is needed to further improve the models and address current limitations in clinical applications.

7.3 Limitations

While this study provides valuable insights into the use of augmented and synthetic data in detecting breast cancer from images, several limitations should be noted.

1. **Quality of Dataset:** Although the dataset contained many high-quality images, there were several instances of images with anomalies or poor quality. These irregularities could have affected the model's training and evaluation processes, potentially skewing the results. Ensuring a consistently high-quality dataset is crucial for reliable model performance.

2. **Stability of the GAN:** The Generative Adversarial Network used in this study struggled to stabilize during training. GANs are known for their instability, which can result in synthetic data that does not adequately capture the variability and realism of the original data. A more stable and advanced GAN architecture could potentially yield better and more consistent results.

3. **Additional dataset:** The additional dataset, while containing the same type of images (ROIs of mammograms), had visually different characteristics compared to the training dataset. This variability in imaging techniques could explain the model's poor generalization to the new data. The lack of a reliable metric to account for these differences suggests that the model's performance on the additional dataset may not accurately reflect its true generalisability.

These limitations highlight the challenges in training and evaluating CNN models for medical image analysis. Addressing these issues in future research could lead to more robust and generalizable models. Ensuring dataset quality, improving GAN stability, and accounting for variability in external datasets are critical steps toward developing effective tools for breast cancer detection.

7.4 Future research

While this research provides valuable insights into the use of augmented and synthetic data for improving convolutional neural network performance in breast cancer detection, future research is suggested to build upon these findings and to address the limitations.

1. **Improved Synthetic Data Generation:** The stability and quality of synthetic image generation can be improved by employing more advanced GAN architectures, such as StyleGAN, CycleGAN, or WassersteinGANs.

2. **Integration with Clinical Data:** Combining image data with clinical data, such as genetic information and medical history, could lead to more comprehensive models that provide better results.

3. **Validation dataset:** As mentioned in the limitations, the validation dataset had visually different characteristics. It is recommended to find an additional dataset with similar features.

4. **Additional testing** To further evaluate how well the augmented and synthetic data worked, it could be interesting to intentionally train the model on a worse dataset by removing images. This will also highlight how diverse the used dataset is.

5. **Train on more, diverse data:** Although the goal is not to rely on large datasets due to their limited availability, it is advised to train on a more diverse dataset. When a wider range of images is introduced the model will be able to learn a wider variety of features, enhancing its robustness and performance in real-world scenarios.

By addressing these issues, future research can contribute to developing more accurate and generalizable CNN models for breast cancer detection, this will result in improved patient outcomes and advances in the field of medical imaging.

Bibliography

[1] Dr. Pragnya. Understanding the staging of breast cancer, 2021. https://drpragnya.com/blog/understanding-the-staging-of-breast-cancer/].

[2] Mohammadmehdi Ataei, Markus Bussmann, Vahid Shaayegan, Franco Costa, Sejin Han, and Chul B. Park. Nplic: A machine learning approach to piecewise linear interface construction. *Computers and Fluids*, 223, 2021.

[3] Unknown. An introduction to convolutional neural networks, 2022.

[4] A. Dertart. Applied deep learning - part 4: Convolutional neural networks. *Towards Data Science*, 2017.

[5] Si. M.T. Budhi Irawan Muhamad Yani, S. Application of transfer learning using convolutional neural network method for early detection of terry's nail. *IOP Conference Series: Materials Science and Engineering*, 879:012052, 2019.

[6] Generative adversarial network (gan) - semiconductor engineering. https://semiengineering.com/knowledge$_{c}enters/artificial - intelligence/neural - networks/generative - adversarial - network - gan/$.

[7] skooch. Ddsm mammography, 2019. [Data set].

[8] Xiaobing Han, Yanfei Zhong, Liqin Cao, and Liangpei Zhang. Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing*, 9, 2017.

[9] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning Release 0.17.5*, volume 17. 2021.

[10] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. volume 2019-June, 2019.

[11] Tashin Ahmed and Noor Hossain Nuri Sabab. Classification and understanding of cloud structures via satellite images with efficientunet. *SN Computer Science*, 3, 2022.

[12] World Cancer Research Fund. Breast cancer statistics. https://www.wcrf.org/cancer-trends/breast-cancer-statistics/, 2024.

[13] RIVM. Bevolkingsonderzoek borstkanker, 2024.

[14] Nisreen I.R. Yassin, Shaimaa Omran, Enas M.F. El Houby, and Hemat Allam. Machine learning techniques for breast cancer computer aided diagnosis using different image modalities: A systematic review, 2018.

[15] Aris Gkoulalas-Divanis and Grigorios Loukides. *Introduction to medical data privacy.* 2015.

[16] Atharva Tapkir. A comprehensive overview of gradient descent and its optimization algorithms. *IARJSET*, 10, 2023.

[17] Aghiles Kebaili, Jérôme Lapuyade-Lahorgue, and Su Ruan. Deep learning approaches for data augmentation in medical imaging: A review, 2023.

[18] World Health Organization. Breast cancer, 2024. https://www.who.int/news-room/fact-sheets/detail/breast-cancer].

[19] World Cancer Research Fund International. Breast cancer statistics, 2022. https://www.wcrf.org/cancer-trends/breast-cancer-statistics].

[20] Jamie DePolo. Breast cancer stages, 2023. https://www.breastcancer.org/pathology-report/breast-cancer-stages].

[21] American Cancer Society. Survival rates for breast cancer, 2024. https://www.breastcancer.org/pathology-report/breast-cancer-stages].

[22] Harminder Singh, James A. Dickinson, Guylène Thériault, Roland Grad, Stéphane Groulx, Brenda J. Wilson, Olga Szafran, and Neil R. Bell. Overdiagnosis: Causes and consequences in primary health care. *Canadian Family Physician*, 64, 2018.

[23] Lars J. Grimm, Carolyn S. Avery, Edward Hendrick, and Jay A. Baker. Benefits and risks of mammography screening in women ages 40 to 49 years. *Journal of Primary Care and Community Health*, 13, 2022.

[24] Heang Ping Chan, Ravi K. Samala, and Lubomir M. Hadjiiski. Cad and ai for breast cancer - recent development and challenges, 2020.

[25] Heang Ping Chan, Lubomir M. Hadjiiski, and Ravi K. Samala. Computer-aided diagnosis in the era of deep learning. volume 47, 2020.

[26] R. L. Birdwell, D. M. Ikeda, K. F. O'Shaughnessy, and E. A. Sickles. Mammographic characteristics of 115 missed cancers later detected with screening mammography and the potential utility of computer-aided detection. *Radiology*, 219, 2001.

[27] Debra M. Ikeda, Robyn L. Birdwell, Kathryn F. O'Shaughnessy, Edward A. Sickles, and R. James Brenner. Computer-aided detection output on 172 subtle findings on normal mammograms previously obtained in women with breast cancer detected at follow-up screening mammograhy. *Radiology*, 230, 2004.

[28] David Gur and Jules H. Sumkin. Cad in screening mammography, 2006.

[29] Subrato Bharati, Prajoy Podder, and M. Rubaiyat Hossain Mondal. Artificial neural network based breast cancer screening: A comprehensive review. *International Journal of Computer Information Systems and Industrial Management Applications*, 12, 2020.

[30] Matthew Gromet. Comparison of computer-aided detection to double reading of screening mammograms: Review of 231,221 mammograms, 2008.

[31] Fiona J. Gilbert, Susan M. Astley, Magnus A. McGee, Maureen G.C. Gillan, Caroline R.M. Boggis, Pamela M. Griffiths, and Stephen W. Duffy. Single reading with computer-aided detection and double reading of screening mammograms in the united kingdom national breast screening program. *Radiology*, 241, 2006.

[32] Paul Taylor and Henry W.W. Potts. Computer aids and human second reading as interventions in screening mammography: Two systematic reviews to compare effects on cancer detection and recall rate. *European Journal of Cancer*, 44, 2008.

[33] Constance D. Lehman, Robert D. Wellman, Diana S.M. Buist, Karla Kerlikowske, Anna N.A. Tosteson, and Diana L. Miglioretti. Diagnostic accuracy of digital screening mammography with and without computer-aided detection. *JAMA Internal Medicine*, 175, 2015.

[34] Joshua J. Fenton, Stephen H. Taplin, Patricia A. Carney, Linn Abraham, Edward A. Sickles, Carl D'Orsi, Eric A. Berns, Gary Cutter, R. Edward Hendrick, William E. Barlow, and Joann G. Elmore. Influence of computer-aided detection on performance of screening mammography. *New England Journal of Medicine*, 356, 2007.

[35] Yiming Gao, Krzysztof J. Geras, Alana A. Lewin, and Linda Moy. New frontiers: An update on computer-aided diagnosis for breast imaging in the age of artificial intelligence, 2019.

[36] Zicheng Guo, Jiping Xie, Yi Wan, Min Zhang, Liang Qiao, Jiaxuan Yu, Sijing Chen, Bingxin Li, and Yongqiang Yao. A review of the current state of the computer-aided diagnosis (cad) systems for breast cancer diagnosis, 2022.

[37] Clayton R. Taylor, Natasha Monga, Candise Johnson, Jeffrey R. Hawley, and Mitva Patel. Artificial intelligence applications in breast imaging: Current status and future directions, 2023.

[38] Data sharing in the age of deep learning, 2023.

[39] Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, A. S. Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A. Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H. Al-Timemy, Ye Duan, Amjed Abdullah, Laith Farhan, Yi Lu, Ashish Gupta, Felix Albu, Amin Abbosh, and Yuantong Gu. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, 10, 2023.

[40] Li Shen, Yan Sun, Zhiyuan Yu, Liang Ding, Xinmei Tian, and Dacheng Tao. On efficient training of large-scale deep learning models: A literature review. 4 2023.

[41] Jeffrey N. Weiss. *The Health Insurance Portability and Accountability Act (HIPAA)*. 2023.

[42] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council.

[43] Stéphane Monteiro, Diogo Oliveira, João António, Filipe Sá, Cristina Wanzeller, Pedro Martins, and Maryam Abbasi. Data anonymization: Techniques and models. volume 344, 2024.

[44] Laura Liberman and Jennifer H. Menell. Breast imaging reporting and data system (bi-rads), 2002.

[45] Andrej Krenker, Janez Bester, and Andrej Kos. *Introduction to the Artificial Neural Networks*. 2011.

[46] Jonathan Johnson. What's a deep neural network? deep nets explained, 2020. https://www.bmc.com/blogs/deep-neural-network].

[47] Siddharth Sharma, Simone Sharma, and Anidhya Athaiya. activation functions in neural networks, 2020.

[48] Johannes Lederer. Activation functions in artificial neural networks: A systematic overview. 1 2021.

[49] Timothy O. Hodson, Thomas M. Over, and Sydney S. Foks. Mean squared error, deconstructed. *Journal of Advances in Modeling Earth Systems*, 13, 2021.

[50] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *Schedae Informaticae*, 25, 2016.

[51] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323, 1986.

[52] Aarush Saxena. An introduction to convolutional neural networks. *International Journal for Research in Applied Science and Engineering Technology*, 10, 2022.

[53] Anirudha Ghosh, Abu Sufian, Farhana Sultana, Amlan Chakrabarti, and Debashis De. *Fundamental concepts of convolutional neural network*, volume 172. 2019.

[54] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology, 2018.

[55] Ibrahem Kandel and Mauro Castelli. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, 6, 2020.

[56] Nitish Shirish Keskar, Jorge Nocedal, Ping Tak Peter Tang, Dheevatsa Mudigere, and Mikhail Smelyanskiy. On large-batch training for deep learning: Generalization gap and sharp minima. 2017.

[57] Tad Gonsalves and Jaychand Upadhyay. *Integrated deep learning for self-driving robotic cars.* 2021.

[58] Jun Lu. Gradient descent, stochastic optimization, and other tales.

[59] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. 2015.

[60] Dami Choi, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. On empirical comparisons of optimizers for deep learning. 10 2019.

[61] Stanford Vision Lab. Imagenet large scale visual recognition challenge (ilsvrc), 2022. https://www.image-net.org/challenges/LSVRC/].

[62] Heang Ping Chan, Lubomir M. Hadjiiski, and Ravi K. Samala. Computer-aided diagnosis in the era of deep learning. volume 47, 2020.

[63] Vaibhav Verdhan. *VGGNet and AlexNet Networks.* 2021.

[64] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. volume 07-12-June-2015, 2015.

[65] Stanford Vision Lab. Large scale visual recognition challenge 2014 (ilsvrc2014), 2014. https://image-net.org/challenges/LSVRC/2014/results].

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. volume 2016-December, 2016.

[67] Stanford Vision Lab. Large scale visual recognition challenge 2015 (ilsvrc2015), 2015. https://image-net.org/challenges/LSVRC/2015/results].

[68] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. volume 2017-January, 2017.

[69] Stanford Vision Lab. Large scale visual recognition challenge 2017 (ilsvrc2017), 2017. https://image-net.org/challenges/LSVRC/2017/].

[70] Emmanuel Lawrence Omonigho, Micheal David, Achonu Adejo, and Saliyu Aliyu. Breast cancer:tumor detection in mammogram images using modified alexnet deep convolution neural network. 2020.

[71] Shayma'a A. Hassan, Mohammed S. Sayed, Mahmoud I. Abdalla, and Mohsen A. Rashwan. Breast cancer masses classification using deep convolutional neural networks and transfer learning. *Multimedia Tools and Applications*, 79, 2020.

[72] G. Jayandhi, J. S.Leena Jasmine, and S. Mary Joans. Mammogram image classification system using deep learning for breast cancer diagnosis. volume 2519, 2022.

[73] Hasan Serdar Macit and Kadir Sabanci. Benchmarking of resnet models for breast cancer diagnosis using mammographic images. *International Journal of Applied Methods in Electronics and Computers*, 2023.

[74] Romario Sameh Samir Anwar. Efficientnet algorithm for classification of different types of cancer. *Artificial Intelligence and Applications*, 2023.

[75] Luke Taylor and Geoff Nitschke. Improving deep learning with generic data augmentation. 2018.

[76] P. Pratheep Kumar, V. Mary Amala Bai, and Geetha G. Nair. Augmentation techniques on mammogram images for cnn based breast cancer classification. volume 841, 2022.

[77] Parita Oza, Paawan Sharma, Samir Patel, Festus Adedoyin, and Alessandro Bruno. Image augmentation techniques for mammogram analysis. *Journal of Imaging*, 8, 2022.

[78] Kenny H. Cha, Nicholas Petrick, Aria Pezeshk, Christian G. Graff, Diksha Sharma, Andreu Badal, and Berkman Sahiner. Evaluation of data augmentation via synthetic images for improved breast mass detection on mammograms using deep learning. *Journal of Medical Imaging*, 7, 2019.

[79] Hasan Nasir Khan, Ahmad Raza Shahid, Basit Raza, Amir Hanif Dar, and Hani Alquhayz. Multi-view feature fusion based four views model for mammogram classification using convolutional neural network. *IEEE Access*, 7, 2019.

[80] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 6 2014.

[81] Afia Sajeeda and B. M.Mainul Hossain. Exploring generative adversarial networks and adversarial training, 2022.

[82] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *35th International Conference on Machine Learning, ICML 2018*, volume 8, 2018.

[83] Sanjeet Singh Khanuja and Harmeet Kaur Khanuja. Gan challenges and optimal solutions. *International Research Journal of Engineering and Technology*, 2021.

[84] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 2016.

[85] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets mehdi. *arXiv:1411.1784v1 [cs.LG] 6 Nov 2014 Conditional*, 2018.

[86] University of South Florida. Digital database of screening mammography, 1998. http://www.eng.usf.edu/cvprg/Mammography/Database.html].

[87] Rebecca Sawyer Lee, Francisco Gimenez, Assaf Hoogi, Kanae Kawai Miyake, Mia Gorovoy, and Daniel L. Rubin. Data descriptor: A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific Data*, 4, 2017.

[88] Inês C. Moreira, Igor Amaral, Inês Domingues, António Cardoso, Maria João Cardoso, and Jaime S. Cardoso. Inbreast: Toward a full-field digital mammographic database. *Academic Radiology*, 19, 2012.

[89] John Suckling, J. Parker, D. Dance, S. Astley, I. Hutt, C. Boggis, I. Ricketts, E. Stamatakis, N. Cerneaz, S. Kok, P. Taylor, D. Betal, and J. Savage. Mammographic image analysis society (mias) database v1.21. 2015.

[90] Richa Agarwal, Moi Hoon Yap, Md Kamrul Hasan, Reyer Zwiggelaar, and Robert Martí. *Deep Learning in Mammography Breast Cancer Detection*. 2022.

[91] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60, 2017.

[92] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. volume 2019-June, 2019.

[93] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018.

[94] Steven A. Hicks, Inga Strümke, Vajira Thambawita, Malek Hammou, Michael A. Riegler, Pål Halvorsen, and Sravanthi Parasa. On evaluation metrics for medical applications of artificial intelligence. *Scientific Reports*, 12, 2022.

[95] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6, 2020.