Abstract

Breast cancer is one of the most prevalent and life-threatening diseases affecting women worldwide. With increasing incidence rates and high mortality, especially in developing and under-resourced regions, early and accurate diagnosis remains a critical aspect of improving patient survival and quality of life. Traditional diagnostic methods, including mammography, ultrasonography, and histopathological analysis, are effective but often involve time-consuming, subjective, and resource-intensive processes. These methods are also prone to human error and typically require invasive follow-up procedures. To address these limitations, this study investigates the integration of advanced deep learning techniques with image preprocessing methods to automate and enhance the breast cancer detection pipeline.

This report introduces a strong deep learning-based method on breast cancer classification with three well-acknowledged datasets: RSNA, mini-MIAS, and BCDR. All the datasets were preprocessed through the Wiener filter to accurately minimize image noise while maximizing the quality of images for improved model performance. For the RSNA dataset, further preprocessing was done by using a Deep Convolutional Generative Adversarial Network (DCGAN) to create synthetic mammogram images to increase the dataset and solve class imbalance. An ensemble learning approach was used for this dataset, which combined four strong convolutional neural network models: EfficientNet, DenseNet, ResNet, and AlexNet. This combination was focused on taking advantage of every model's strengths to enhance overall classification accuracy and generalization.

On the other hand, the mini-MIAS and BCDR datasets were trained and processed with a Custom CNN architecture especially for effective and accurate classification on comparatively smaller datasets. The obtained results proved the efficacy of the proposed models with achieved classification accuracies of 97.85%, 97.8%, and 98.20% on the RSNA, mini-MIAS, and BCDR datasets respectively. These findings show remarkable improvement in detection performance, with the models exhibiting an ability to eliminate false positives—a frequent issue in traditional digital mammography.

This study highlights the potential of deep learning techniques in non-invasive breast cancer detection, offering a promising direction for enhancing early diagnosis and supporting radiologists with reliable automated tools.

1 INTRODUCTION

Cancer is among the world's foremost causes of death, with breast cancer being the most common in women. Breast cancer arises in breast cells, most often in the ducts (milk-producing tubes) or lobules (milk-producing glands). It results when abnormal cells divide rapidly and may even spread to distant parts of the body via blood or lymph [**1**]. An estimated 670,000 women were killed by breast cancer in the world in 2024, according to figures from the Breast Cancer Research Foundation, citing reports from the World Cancer Research Fund International and World Health Organization [1]. Early detection improves survival rates. There is a wide range of technologies used by experts for the detection of breast abnormalities, such as Digital Mammography, MRI, CT, and PET [2].

Late stage diagnosis is one of the largest contributors for high mortality rates, as the cancer has oftentimes already progressed, if not metastasized, which makes the chances of effective treatment significantly lower. Symptoms often manifest at advanced stages of the disease which limits curative treatment options, particularly in more aggressive forms of breast cancer.Detection of the cancer in its early stages greatly enhances prognosis and breast cancer survival, hence using AI-driven technologies that expedite the analysis of mammograms helps classify lumps as benign or malignant swiftly. This aids in providing prompt treatment as delays in diagnosis are diminished, thus improving overall patient outcomes while simultaneously easing the strain on healthcare systems.

The modern approaches to diagnostics still have their restrictions, including over-reliance on expert analysis, lengthy procedures, and the potential for positive or negative misdiagnosis. Such problems stress the case of the lack of more advanced and efficient diagnostic tools. To fill these gaps, we developed an AI-based model for breast cancer detection that accurately and dependably classifies mammogram images as either benign or malignant. The model is user-friendly and simple to operate: the user uploads a breast mammogram image, and the model completes the processes of analyzing and supplying results automatically and instantly. The model reduces effort, time, and subjectivity in manually resolving diagnostic challenges, showcasing the ability of artificial intelligence to provide support for shallow-depth breast cancer detection enhancement and clinical decision-making aid.

Over the years, Deep learning (DL), a branch of artificial intelligence, has transformed breast cancer diagnosis through medical image analysis. In contrast to conventional approaches that involve feature extraction done manually, DL algorithms learn from raw histopathology images or mammograms directly to detect malignant features at high accuracy rates. By independently identifying subtle patterns usually not detectable by humans, DL minimizes errors in diagnosis and invasive procedures. Its capacity to analyze large data sets rapidly also simplifies clinical workflows, increasing diagnosis efficiency,

accessibility, and accuracy, a significant milestone for medical diagnostics.

Deep learning plays a vital role in breast cancer detection, primarily through image classification tasks that distinguish between benign and malignant cases. CNNs works excellently in this domain by automatically extracting hierarchical features from mammograms, learning complex patterns linked to tumors. In addition to classification, deep learning supports segmentation tasks, isolating specific regions like tumors for precise analysis. This dual capability enhances diagnostic accuracy and efficiency. As these models are trained on large datasets, they continuously improve, offering scalable, adaptable tools that assist clinicians in delivering more reliable and early breast cancer diagnoses.

1.1 Problem Statement

Today, detecting breast cancer early enough to make a difference and avoiding the psychological and financial burden of a misdiagnosis is more important than simply detecting the disease. Treatment success rates increase dramatically when caught early, but it's still very challenging to identify these minute cellular alterations. Mammography continues to be our front-line screening tool, but anyone who's worked in oncology knows its limitations all too well. I've seen countless cases where small tumors hide within dense breast tissue like needles in a haystack. The flip side is equally problematic: false positives that send people down a rabbit hole of anxiety, additional testing, and sometimes unnecessary biopsies. These aren't just clinical inconveniences; they represent real trauma and financial hardship for patients.

Over the past few years, this discipline has seen remarkable progress in computer-assisted detection (CAD) systems meant to assist radiologist.. These systems employ sophisticated image-processing algorithms to enhance contrast, remove visual noise, and identify potential lesions. Despite these advancements, many CAD tools suffer from frustratingly high false-positive rates. Digital mammography stands out as the most widely used screening tool, as it is accessible in most healthcare settings, relatively affordable (though still a financial burden for many uninsured patients), and quite effective at revealing subtle tissue abnormalities. Unfortunately, some methods such as CT and PET involve exposure to deadly radiation and therefore limit safety in the long run [2]. MRIs, for instance, have high cost per scan. This trade-off between cost, accessibility, and potential harm explains why mammography—especially when enhanced with computer assistance remains the cornerstone of screening programs worldwide.

Modern deep learning approaches show tremendous promise, but they're hampered by their notorious "black box" nature. Try explaining to a patient that an algorithm flagged something concerning in their scan, but you can't really say why or how it reached that conclusion. This lack of transparency isn't just an academic concern—it actively undermines clinician trust in moments when critical decisions hang in the balance. The data challenges don't help matters either. Most training datasets suffer from a fundamental imbalance—normal tissue samples dramatically outnumber cancerous ones. It's like trying to teach someone to identify rare birds by showing them mostly pigeons. Not surprisingly, these models end up performing admirably on common cases but stumbling when faced with rare presentations of malignancy. To make matters worse, datasets often reflect specific demographic groups and imaging protocols, limiting their applicability across different populations and clinical settings. A model trained on scans from predominantly Caucasian patients in well-funded academic medical centers might perform poorly when deployed in community clinics serving diverse populations.

Given these challenges, we desperately need a new approach to breast cancer de-

tection—one that delivers consistently accurate results across different tissue types while significantly reducing both false positives and false negatives. Such a system must provide clear visual explanations for its findings, allowing radiologists to understand and verify its reasoning rather than simply accepting or rejecting mysterious outputs. And perhaps most importantly, it needs to function reliably even in settings with limited resources, where advanced imaging options might not be available.

Our research tackles these issues head-on by integrating traditional image processing techniques (which offer transparency and established reliability) with cutting-edge deep learning approaches (which excel at pattern recognition). Though we've encountered numerous roadblocks along the way—including some frustrating early results with transfer learning that didn't generalize well—we believe this hybrid methodology holds the key to more reliable, explainable, and accessible breast cancer detection. If we succeed, the impact could extend far beyond academic publications, potentially saving thousands of lives through earlier intervention and more accurate diagnosis.

The performance of deep learning models like custom CNN can be considerably improved by employing sophisticated preprocessing and optimization methods. Active Wiener filter improves image acuteness, and data augmentation strategies such as flipping, rotation, and scaling enhance dataset diversity, which enhances generalizability. Normalization provides similar input scaling, which speeds up model convergence. In this study, two datasets were preprocessed with a custom CNN, and a third was preprocessed with DCGAN for generating synthetic images, followed by Ensemble learning, which enhanced overall robustness and accuracy. These aggregate methods make the system an effective, time-saving aid to clinical breast cancer classification.

1.2 Objective

Our research aims to develop a reliable, interpretable breast cancer detection pipeline that addresses these limitations. In this study we have started by collecting and preprocessing mammograms from public datasets, applying various preprocessing techniques such as wiener filter for noise reduction, DCGAN for generation of synthetic images to tackle the class imbalance problem. Next, we have used ensemble learning and custom CNN, for training. We have used standard metrics— accuracy (Acc), sensitivity (Sns), specificity, and area under the ROC curve (AUROC)—for performance evaluation.

The main highlights and outcomes of this study are:

- To build a DL model for classification of breast cancer Benign or Malignant.
- Three different datasets are used and preprocessed separately to understand patterns and characteristics.
- Image enhancement techniques like the Wiener filter are applied. In addition to this, DCGAN (a type of GAN) is used for generating synthetic images to solve the class imbalance issue.
- Model's performance is compared with other studies.

The rest of this report is structured as below. In Section 2, we discuss the existing work on deep learning methods to identify breast cancer, image preprocessing methods, and class imbalance solutions. Section 3 and Section 4 explains our Project Methodology, the data sets used in this project report and the preprocessing as well as DL architecture. Apart from this we will consider flowcharts and the mathematics involved in different techniques. In Section 5 and Section 6 we have demonstrated our simulation Environment and python implementation overview along with snippets of our code. Section 7 shows the study findings, performance matrix and comparison with other studies. Section 8 contains our conclusion and future work.

2 Literature Review

This section provides the background information about breast cancer detection, classification and diagnosis. In P1 S.Łukasiewicz et al. gives brief about the breast cancer classifications, risk factors, diagnosis and treatments [3]. P2 work by Nusrat Mohi ud din et al. tells us about the future challenges in detection and classification of breast cancer with DL techniques [4]. P3 (2020) by Al-Haija et al. and P4 (2021) by V. Jyoshna et al., covers about the comprehensive study on histopathological breast image [5] to classify cancer as malignant or benign using resnet50 CNN model along with transfer learning technique [6]. In P5 P. Patro has briefly discussed and implemented preprocessing algorithms like wiener filter and watershed. The training is done by using hybrid CNN model to achieve high accuracy [7]. P6 work by D. Shah et al. focus on DCGAN technique to generate synthetic images for better training. In addition to this ensemble learning models such as EfficientNet, ResNet, AlexNet and DenseNet is used for training purpose [8]. In P7 by E. Deniz et al. gives us brief about models like alexNet and Vgg16 along with transfer learning technique for breast cancer detection and classification [9]. In P8 work by A. Das et al., the classification accuracy of 98.08% has been achieved by using stack ensemble approach [10]. P9 written by H.N. Huynh et al. tells about deep learning models based on CNN such as visual geometry group (VGG), Googlenet, Residual Networks and EfficientNet. These models were used to classify mammograms into cancer or non-cancer categories [11]. P10 work by J.R. Pérez-Núñez et al. provides analysis of 62 different articles, exploring different types of CNN architectures [12]. In P11 by D.A. Zebari et al., the classifiers were trained and later used with ANN for features classification of mammograms on different datasets [13]. P12 work by K. Wisaeng proposes proposes a new K–means++ clustering algorithm, based on Cuckoo Search Optimization (KM++CSO) for breast cancer detection [14].

Table 1: Comprehensive Analysis on Breast Cancer Detection

SNo./Ref. /Year	**Objective**	**Dataset Used**	**Model /Technique**	**Pros**	**Cons**
P1 [3] 2021	Summarizes current breast cancer research in diverse domains	Not Applicable	Not Applicable	helps to identify cause of breast cancer	Biased due to non-systematic review.

SNo./Ref. /Year	Objective	Dataset Used	Model /Technique	Pros	Cons
P2 [4] 2022	Overview of DL techniques on diverse datasets	BreakHis, DDSM, INbreast, and MIAS	Deep Reinforcement Learning	Handles large datasets	Does not systematically present results.
P3 [5] 2020	Breast cancer detection using Transfer Learning	BreakHis dataset	ResNet-50	Achieved high accuracy	Limited dataset diversity.
P4 [6] 2021	Assess features in breast lesions	Wisconsin BCD	Decision Tree Algorithm	Early Detection for treatment	Showed lower precision despite high accuracy.
P5 [7] 2024	Image Classification using hybrid approach	DDSM and mini-MIAS	HCNN + ERNN	Achieved High accuracy	Not Generalizable Model
P6 [8] 2024	Optimising Breast Cancer Detection	CBIS-DDSM and RSNA	Ensemble Learning	Achived high accuracy and sensitivity	Trained on specific datasets only.
P7 [9] 2018	Breast Cancer Diagnosis	BreakHis	Transfer learning	Improved image classification process	Overfitting on small datasets
P8 [10] 2021	Implemented Transfer Learning and ensemble methods	Breast Histopathology Images	Ensemble Deep Learning	Stack ensemble improves classification accuracy	Requires large dataset for training

SNo./Ref. /Year	Objective	Dataset Used	Model /Technique	Pros	Cons
P9 [11] 2023	Implemented Attention mechanism with CNN	RSNA	CNN Models	Strong Detection Capability	Malignant Class Imbalance
P10 [12] 2024	Evaluates several CNN Models	BCDR	Transfer Learning	Effective Feature Extraction from images	Requires large labelled dataset.
P11 [13] 2021	Enhance breast cancer detection	mini-MIAS	M-FD with ANN	Effective Noise Reduction	Not optimal for real-time use.
P12 [14] 2022	Effective study across multiple datasets	BCDR	K-means++ CSO	Improved Image Segmentation with high accuracy	CSO may slow real-time applications

3 Methodology

Our primary objective within this work is to create a deep learning system that is more robust and dependable and that can correctly categorize mammograms as either benign or malignant. We use three complimentary datasets—RSNA, BCDR, and MIAS—each with unique advantages and disadvantages, to accomplish this goal. We start by giving each image a Wiener filter, which subtly reduces noise while maintaining important edge features. We subsequently apply data augmentation methods such as flips, rotations, zooms, and shifts to enhance the diversity of our training data and help the model learn different real-world imaging conditions.

We employ a straightforward convolutional neural network for the relatively small but densely labeled samples BCDR and MIAS datasets. The small visual cues that differentiate worrisome tumors from healthy tissue are picked up by this network. Since the RSNA dataset is much larger, we start by adding artificial mammography pictures produced by a Deep Convolutional GAN (DCGAN) to it. These extra examples help improve the model's ability to generalize and avoid overfitting.

Following the preparation of all mammograms, we use the RSNA set to train four cutting-edge architectures ResNet-50, DenseNet, AlexNet, and EfficientNet-B0 and then aggregate their predictions into an ensemble. ResNet's deep residual learning, DenseNet's feature reuse, AlexNet's shown simplicity, and EfficientNet's balanced scaling are each models' distinct representational strengths. We anticipate that combining them will result in greater accuracy and stability than using just one model. With the use of this meticulously planned pipeline of preprocessing, augmentation, generative modeling, and ensemble learning, our methodology aims to create a potent tool that helps radiologists identify breast cancer more quickly and confidently.

3.1 Proposed Design Flow

Below is an overview of how our system works, step by step, for each dataset. First, every mammogram is cleaned and augmented to make the data as rich and consistent as possible. Then, depending on the dataset, images either go into a custom neural network or through our GAN-powered ensemble of models. Finally, each branch produces its own benign vs. malignant score.

Figure 1 shows the overall flow of our system.

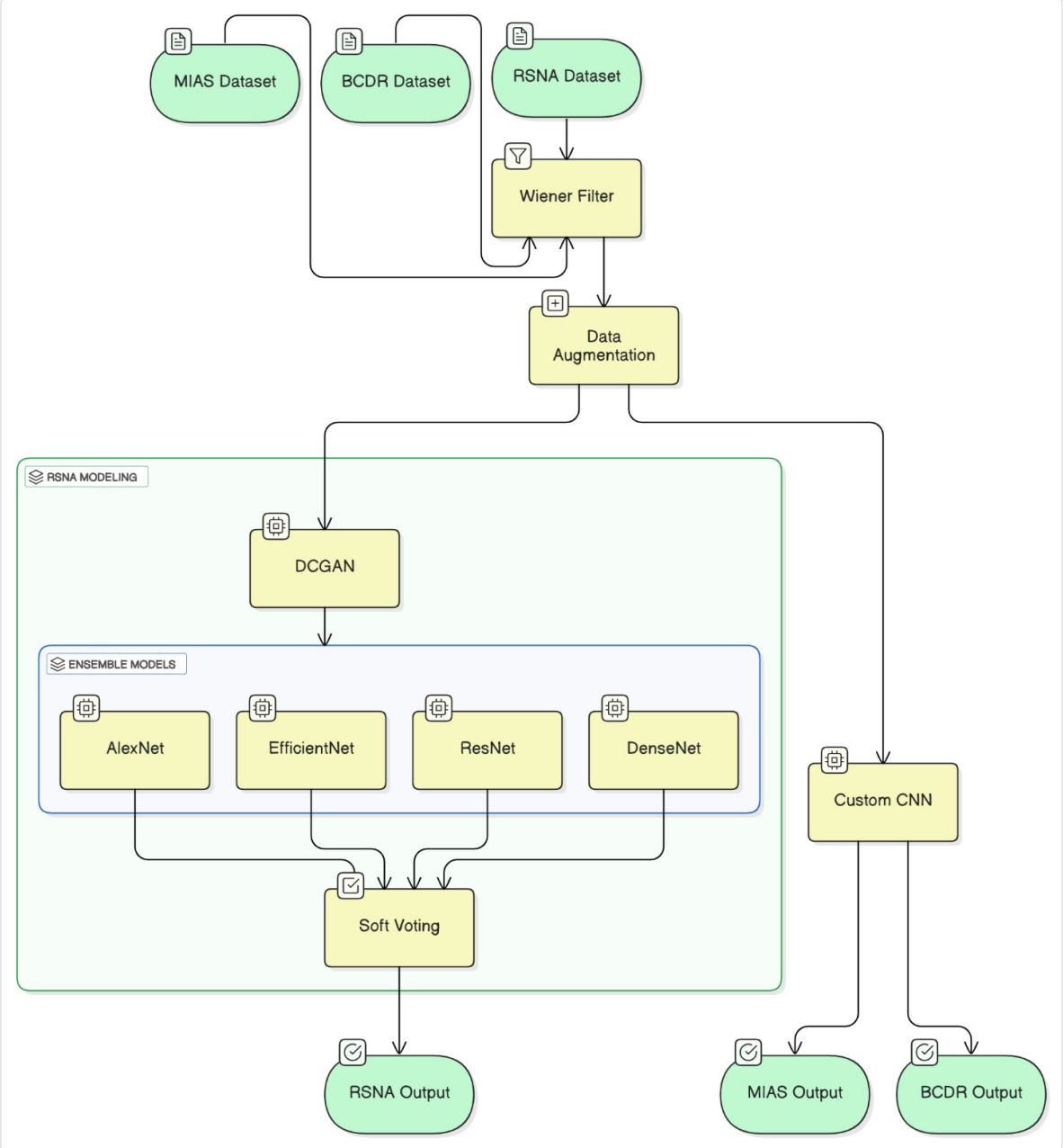


Figure 1: Workflow of the Project

3.2 Workflow Overview

3.2.1 RSNA Flow

1. **Clean & Grow Data**
    - Apply the Wiener filter to denoise RSNA mammograms.
    - Use data augmentation to simulate different imaging angles and scales.
2. **Generate Synthetic Images**
    - Train a DCGAN to make new, lifelike mammograms.
3. **Ensemble Learning**
    - Train four CNNs (ResNet-50, DenseNet, AlexNet, EfficientNet).
    - Combine their predictions using soft voting.
4. **Get Results**

3.2.2 BCDR Flow

1. **Clean & Grow Data**
    - Denoise BCDR images with the Wiener filter.
    - Augment with flips, rotations, and zooms.
2. **Learn Features**
    - Use the same custom CNN from MIAS on the BCDR images.
3. **Get Results**

3.2.3 MIAS Flow

1. **Clean & Grow Data**
    - Remove noise from MIAS images using a Wiener filter.
    - Create more examples by rotating, flipping, and zooming each image.
2. **Learn Features**
    - Feed these enhanced images into our custom CNN to spot benign vs. malignant tissue.
3. **Get Results**

4 Proposed Methodology

Following the overall Flow Chart described in the previous section, this section of the report presents a detailed explanation of the methodology adopted in the development of our breast cancer classification system. The primary objective is to design an efficient pipeline capable of accurately distinguishing between benign and malignant mammogram images.

4.1 Detailed Workflow

1. **Data Collection and Exploration**
2. **Preprocessing**
3. **Deep Convolution GAN**
4. **Ensemble Models**
5. **Custom CNN Model**

4.2 Data Collection and Exploration

Dataset 1 (DS-1): The RSNA Breast Cancer Dataset [15] is an open-source medical imaging data set created by the Radiological Society of North America (RSNA). It has screening mammograms images of patients and include annotations of malignancies and clinical data, which is crucial for training supervised machine learning models. These annotations include information on lesion type, size, location, density of lesion and BI-RADS categories.

Dataset 2 (DS-2): The Breast Cancer Digital Repository (BCDR) [16] is a publicly accessible dataset that consists two repositories, BCDR-FM and BCDR-DM. In this project, the BCDR-D01 was used, as it provides high resolution digital mammograms with ROI (region-of-interest) annotations. The data set has 452 clinically annotated lesions, along with 818 manual segmentations annotated by experienced radiologists as per the BI-RADS.

Dataset 3 (DS-3): The Mini-MIAS [17] dataset is a publicly available digitized film mammograms sourced from the UK National Breast Screening Programme. The dataset comprises 322 mammogram images from 161 patients, covering three primary categories: normal, benign, and malignant. Images are digitized at a 50-micron resolution, reduced to 1024×1024 pixels with an 8-bit grayscale depth, and stored in PGM format, making them suitable for deep neural network applications.

4.3 Preprocessing

Preprocessing is a critical step to improve mammogram quality and enhance model robustness.

DS-1: Missing values were removed to ensure data integrity, followed by an analysis of the distribution of the four BI-RADS density categories (A–D) to understand class skew and its correlation with malignancy. A stratified split was performed based on the density and cancer (0/1) columns, yielding training (70%), validation (15%), and test (15%) sets ensuring balanced training, validation, and test sets for robust model generalization.

DS-2: After eliminating missing values, patients with multi-class targets (e.g., lesions annotated as both benign and malignant) were excluded to maintain label consistency. The dataset was then partitioned using a stratified split on the classification column (benign/malignant), guaranteeing proportional representation of both classes across all subsets.

DS-3: The dataset was cleaned by removing incomplete entries, and a stratified split was applied based on the severity column (benign/malignant), guaranteeing proportional representation of both classes across all subsets.

This study employs a Wiener filter to suppress noise while preserving lesion features, followed by data augmentation is applied to expand and diversify the training set.

4.3.1 Wiener Filter

The Wiener filter [7] is a widely used and effective technique for image denoising, especially in medical image processing. Its main purpose is to reduce noise in images while preserving important details such as edges and textures. This is particularly important in mammograms, where even small features can indicate the presence of a tumor.

The filter works by analyzing a local neighborhood around each pixel using a sliding window, also known as a mask. Within this window, it estimates the local mean and variance of pixel intensities. In areas where the variance is low, typically smooth or uniform regions, the filter applies stronger smoothing to reduce random noise. On the other hand, in high-contrast areas, such as the boundaries of lesions or masses, it performs minimal smoothing to retain important structural details.

What makes the Wiener filter especially useful is its adaptive nature. Unlike traditional filters that apply the same level of smoothing across the entire image, whereas this technique adjusts itself based on the statistical characteristics of each region and this makes it highly effective for preparing mammogram images for further analysis, such as feature extraction and classification, by enhancing clarity while maintaining diagnostic features.

4.3.2 Data Augmentation

To improve the diversity of the dataset and reduce the risk of overfitting, data augmentation techniques were applied to all three datasets RSNA (DS-1), BCDR-D01 (DS-2), and Mini-MIAS (DS-3). Since medical imaging datasets are often limited in size, augmentation helps to artificially expand the dataset by creating modified versions of the original images.

Each mammogram was randomly transformed using a series of operations, including horizontal flipping, small rotations within ±45 degrees, slight zooming or scaling between 90% and 110%, and translations up to ±10% of the image dimensions. These variations simulate real-world conditions where mammograms may be captured from slightly different angles or positions.

Importantly, these transformations are designed to preserve the structure and shape of lesions, ensuring that the clinical features remain intact. By training the model on such diverse image representations, it becomes more robust and better equipped to generalize to new, unseen data during testing. This step is especially critical in medical image analysis, where subtle variations can significantly influence classification performance.

4.4 Architecture

This section discusses the overall architecture pipeline of our study, as illustrated in Figure 2. The pipeline comprises three main stages: (1) Deep Convolutional GAN (DCGAN), (2) An Ensemble framework combining multiple pre-trained networks, and (3) A Custom Convolutional Neural Network (CNN) tailored for breast cancer detection.

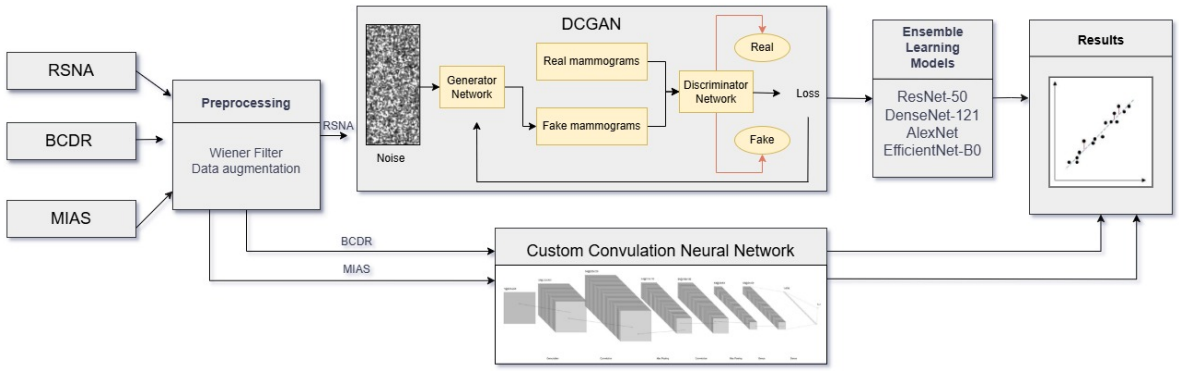


Figure 2: Architecture for Proposed Model

4.5 Deep Convolutional Generative Adversarial Network (DCGAN)

DCGANs were employed to create more synthetic mammogram images. This method helps to balance the dataset by augmenting the minority class with more samples, thus enhancing the model to learn from a more balanced dataset.

DCGANs have a generator and a discriminator. The generator is given a random 100-dimensional noise vector as input and converts it into a realistic 224×224 grayscale image through a chain of transposed convolutional layers.

The discriminator, conversely, is trained to distinguish original and fake images. It employs a chain of convolutional layers with LeakyReLU activations for feature extraction and prediction. While training, the two networks race against one another—the generator attempts to produce images that are more realistic, while the discriminator seeks to correctly classify whether an image is real or generated.

This adversarial training process results in high-quality synthetic mammograms that closely resemble real samples. By augmenting the RSNA dataset with these generated images, the model becomes more robust and better equipped to handle class imbalances during classification.

4.6 Convolution Neural Network

CNNs are a specific type of deep learning model extremely well suited for image classification problems. They are created to learn spatial hierarchies of features from input images automatically and adaptively using a sequence of convolutional, pooling, and fully connected layers. In the context of this research, CNNs play a central role in identifying key patterns within mammograms that help distinguish between benign and malignant cases.

Our system utilizes both pre-trained, state-of-the-art CNN models and a specially designed CNN model. The pre-trained models were selected for their proven performance on large-scale image datasets and their ability to generalize well to medical imaging tasks after fine-tuning. These include DenseNet-121, ResNet-50, EfficientNet-B0, and AlexNet. Each of these models brings unique architectural strengths, such as efficient feature reuse, residual learning, or optimized scaling. In addition, a lightweight custom CNN was developed to serve as a baseline for performance comparison, particularly useful for smaller datasets where computational efficiency and simplicity are essential.

The following subsections provide an overview of each CNN model used in this study, outlining their core architecture, functionality, and relevance to the breast cancer classification task.

4.7 Ensemble Models

Ensemble methods blend the forecasts of a variety of individual learners to create more precise and less vulnerable results than one individual model. Through the use of diverse algorithms—decision trees, support vector machines, and neural networks—ensembles minimize variance, bias, or both. Popular methods are bagging (e.g., Random Forests), where each base learner is trained on a randomly selected subset of data; boosting (e.g., AdaBoost, XGBoost), where consecutive misclassified instances are emphasized; and stacking, where a meta-learner is trained to combine base-model outputs. Since errors in diverse models tend to cancel out, ensembles usually generalize better to unseen data. Their robustness and flexibility explain their popularity in competitions and real-world applications. In our project we have done ensemble learning by taking ResNet-50, DenseNet-121, EfficientNet-B0 and AlexNet models.

4.7.1 ResNet-50 Architecture

ResNet-50 is one of the most popular deep learning models used for image classification problems. It was developed to resolve the vanishing gradient problem commonly faced by very deep networks, where the accuracy begins to decrease when the number of layers increases. ResNet works towards it by proposing **residual learning**, where shortcut (or skip) connections are employed so that gradients can pass more smoothly during backpropagation.

4.7.2 DenseNet-121 Architecture

DenseNet-121 is a deep CNN that improves feature propagation and reduces parameter redundancy by introducing **dense connections**. Within this architecture, every layer takes inputs from all previous layers and outputs to all the next layers in the same dense block. This design enhances gradient flow, alleviates the vanishing gradient problem, and encourages feature reuse, which is especially beneficial in medical imaging tasks.

The high connectivity pattern and transition layers make DenseNet-121 a small yet efficient model, ideal for identifying sophisticated patterns in breast cancer images with enhanced efficiency and accuracy.

4.7.3 EfficientNet-B0

EfficientNet-B0 is the base model of EfficientNet family that presents a *compound scaling* approach to uniformly scale network depth, width, and input resolution using a single coefficient. This balanced scaling achieves high accuracy with fewer FLOPs and parameters and is well-suited for resource-limited applications such as medical image classification.

This design leverages efficient depthwise separable convolutions and squeeze-and-excitation modules to maximize representational power while keeping computational cost low.

4.7.4 AlexNet Architecture

AlexNet was one of the first deep convolutional neural networks that proved deep learning worked for image classification. It uses ReLU activations for faster convergence, overlapping max-pooling to retain spatial information, local response normalization to encourage generalization, and dropout to reduce overfitting.

4.8 Custom Convolutional Neural Network Architecture

This custom CNN is tailored for binary classification of preprocessed breast images. It alternates convolutional and pooling layers to learn increasingly abstract spatial features, then uses dropout and dense layers to reduce overfitting and perform the final decision.

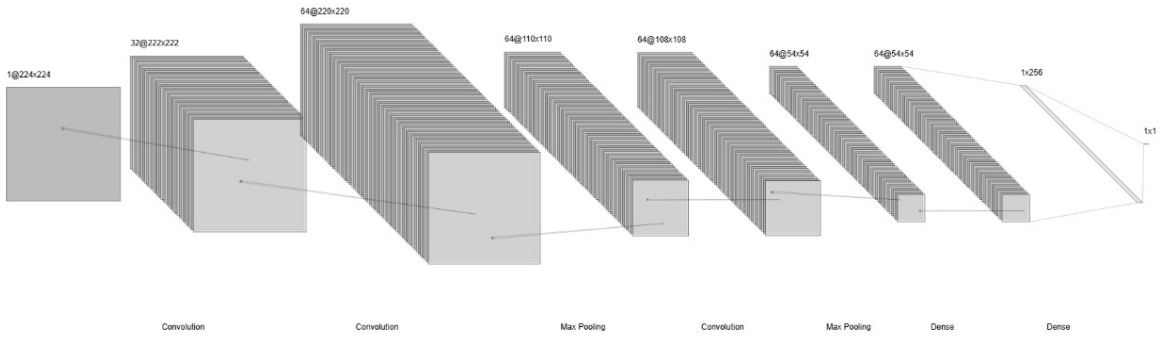


Figure 3: Custom CNN Architecture

- **Input:** Preprocessed breast image, resized to 224×224×1.
- **Conv2D Layer 1:** 32 filters, 3×3 kernel, stride 1, ReLU activation.
- **Conv2D Layer 2:** 64 filters, 3×3 kernel, stride 1, ReLU activation.
- **Max Pooling Layer 1:** 2×2 pool size, stride 2.
- **Conv2D Layer 3:** 64 filters, 3×3 kernel, stride 1, ReLU activation.
- **Max Pooling Layer 2:** 2×2 pool size, stride 2.
- **Dropout Layer 1:** Dropout rate of 25% to prevent overfitting.

- **Dense Layer:** 64 neurons, ReLU activation, serving as a compact feature integrator.
- **Dropout Layer 2:** Dropout rate of 25%.
- **Flatten Layer:** Transforms the 2D feature maps into a 1D feature vector.
- **Output Layer:** Dense layer with 1 neuron and sigmoid activation, yielding a probability score for benign vs. malignant classification.

This architecture effectively captures hierarchical patterns in the input image, while dropout layers help maintain generalization. The final sigmoid-activated unit produces a clear binary decision.

4.9 Mathematical Modelling

In this section, we outline the mathematical models and equations used to describe the underlying processes of the models implemented in this project. The core of the methodology involves Wiener Filter, Convolutional Neural Networks (CNNs) which are described mathematically.

4.9.1 Wiener Filter

To suppress noise in the mammogram images prior to segmentation and CNN processing, we apply a Wiener filter based on local statistics. Let the observed (noisy) image be

$$g(x, y) = f(x, y) * u(x, y) + n(x, y) \tag{1}$$

where where $f(x, y)$ is the original image, $u(x, y)$ is the degradation factor such as blur, $n(x, y)$ is the additive noise, and $g(x, y)$ is the observed noisy image. The Wiener-filtered output is denoted

$$h(x, y) = R\big[g(x, y)\big] \tag{2}$$

where $R[\cdot]$ is the Wiener restoration operator.

Over a local $N \times M$ neighbourhood η around pixel (x, y), we compute:

$$\mu = \frac{1}{NM} \sum_{(i,j)\in\eta} g(i, j) \tag{3}$$

$$\sigma^2 = \frac{1}{NM} \sum_{(i,j)\in\eta} g(i, j)^2 \; - \; \mu^2 \tag{4}$$

Here, μ is the local mean and σ^2 the local variance. Let v^2 be the (estimated) noise variance. Then the Wiener-filtered pixel value at (x, y) is

$$b_w(x, y) \; = \; \mu \; + \; \frac{\sigma^2 - v^2}{\sigma^2} \, \big(g(x, y) - \mu\big) \tag{5}$$

This formulation (cf. Eqns. (3)–(5) in the reference) adaptively attenuates noise while preserving edges and texture, and serves as the preprocessing step.

4.9.2 Convolutional Neural Network (CNN)

The convolutional operation in CNNs is mathematically represented as:

$$y_{i,j} = (x * w)_{i,j} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{i+m,j+n} \cdot w_{m,n} \tag{6}$$

Where x is the input image, w is the kernel (filter), y is the output feature map after convolution, M and N are the dimensions of the filter.

5 Simulation Environment

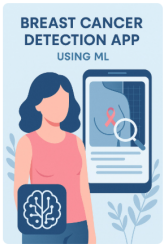
Deploy

Navigation

Go to

- Home
- Info & Resources
- Upload & Check
- Team
- Model Description
- Contact Us

Welcome to the Breast Cancer Detection Simulation App



This app is designed to simulate the use of Convolutional Neural Networks (CNNs) for detecting breast cancer from mammogram images. It is intended for educational and research purposes to demonstrate how machine learning can assist in medical imaging analysis.

Key Features

- **Upload & Check:** Upload mammogram images and receive predictions from our trained CNN models.
- **Model Description:** Learn about the architecture and performance of the CNN models used.
- **Info & Resources:** Access information about breast cancer and resources for further learning.
- **Team:** Meet the team of developers and researchers who created this app.
- **Contact Us:** Reach out to us with any questions or feedback.

Deploy

Navigation

Go to

- Home
- Info & Resources
- Upload & Check
- Team
- Model Description
- Contact Us

Info & Resources

About Breast Cancer

Breast cancer is a type of cancer that develops in breast cells, typically in the ducts (tubes that produce milk) or lobules (glands that generate milk). It is the most common cancer among women worldwide, but it can also affect men, though less frequently.

Statistics

According to the Breast Cancer Research Foundation, citing reports from the World Cancer Research Fund International and World Health Organization, an estimated 670,000 women died from breast cancer globally in 2024. Early detection significantly improves survival rates; for example, the 5-year survival rate for localized breast cancer is 99%, compared to 27% for metastatic cases.

Early Detection

Early detection of breast cancer is crucial as it allows for more effective treatment options and better prognosis. Various imaging technologies are used for screening and diagnosis, including digital mammography, MRI, CT scans, and PET scans.

Deploy

Navigation

Go to

- Home
- Info & Resources
- Upload & Check
- Team
- Model Description
- Contact Us

Early Detection

Early detection of breast cancer is crucial as it allows for more effective treatment options and better prognosis. Various imaging technologies are used for screening and diagnosis, including digital mammography, MRI, CT scans, and PET scans.

AI in Breast Cancer Detection

Artificial Intelligence, particularly deep learning models like Convolutional Neural Networks (CNNs), has shown great promise in medical image analysis. CNNs can automatically learn to identify patterns in mammogram images that may indicate the presence of cancer. By training on large datasets of labeled images, these models can achieve high accuracy in classifying images as benign or malignant.

Limitations of Traditional Methods

Traditional diagnostic methods rely heavily on the expertise of radiologists and can be time-consuming. There is also a risk of human error, leading to false positives or negatives. AI models can assist by providing a second opinion, potentially reducing misdiagnosis rates and speeding up the diagnostic process.

Purpose of This App

This app is designed to simulate the use of CNN models for breast cancer detection. It allows users to upload mammogram images and receive predictions from trained models. Please note that this is for

Navigation

Go to

- Home
- Info & Resources
- Upload & Check
- Team
- Model Description
- Contact Us

Purpose of This App

This app is designed to simulate the use of CNN models for breast cancer detection. It allows users to upload mammogram images and receive predictions from trained models. Please note that this is for educational and research purposes only and should not be used for actual medical diagnosis.

Disclaimer

This app is not a substitute for professional medical advice, diagnosis, or treatment. Always seek the advice of qualified healthcare providers for any medical concerns.

Additional Resources

- American Cancer Society - Breast Cancer
- World Health Organization - Cancer
- National Breast Cancer Foundation
- Deep Learning in Medical Imaging

Navigation

Go to

- Home
- Info & Resources
- Upload & Check
- Team
- Model Description
- Contact Us

Deploy

Upload & Check Mammogram

Upload a mammogram image (PNG, JPG, JPEG, DICOM, TIFF)

Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG, DCM, TIFF, TIF

Browse files

Navigation

Go to

- Home
- Info & Resources
- Upload & Check
- Team
- Model Description
- Contact Us

De

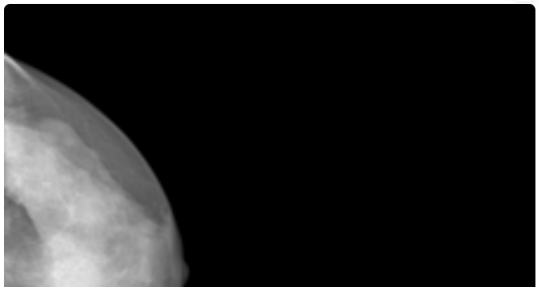
Drag and drop file here

Limit 200MB per file • PNG, JPG, JPEG, DCM, TIFF, TIF

Browse files

img_12_20_1_LCC.tif 1.8MB

The `use_column_width` parameter has been deprecated and will be removed in a future release. Please utilize the `use_container_width` parameter instead.

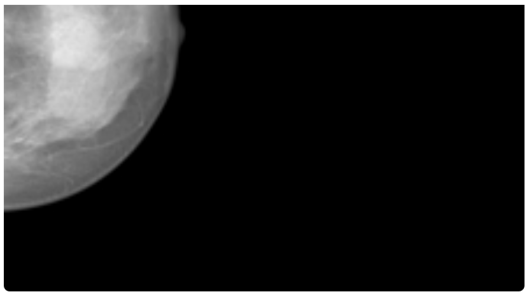


Deploy

Navigation

Go to

- Home
- Info & Resources
- Upload & Check
- Team
- Model Description
- Contact Us



Uploaded Image

Run Inference

Navigation

Go to

- Home
- Info & Resources
- Upload & Check
- Team
- Model Description
- Contact Us

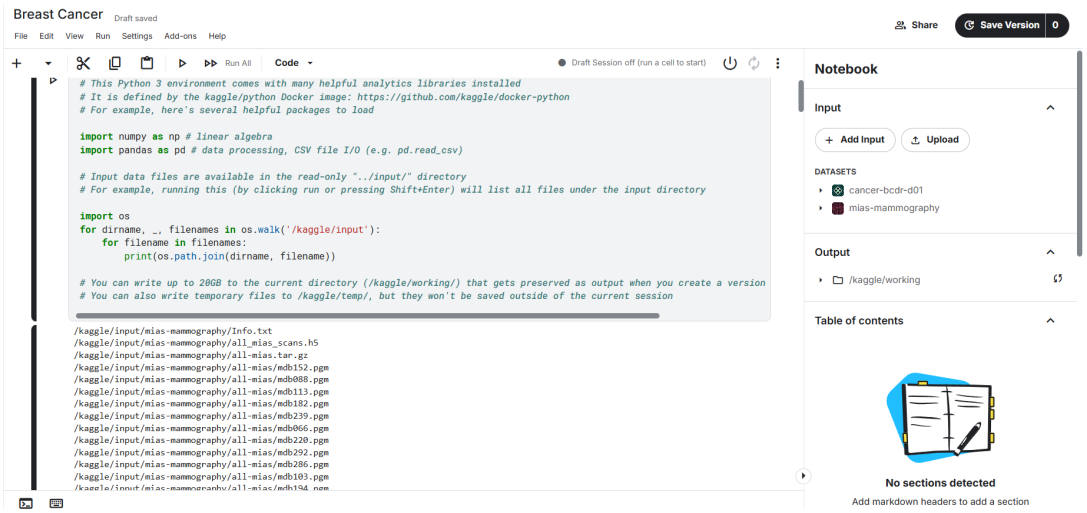
Uploaded Image

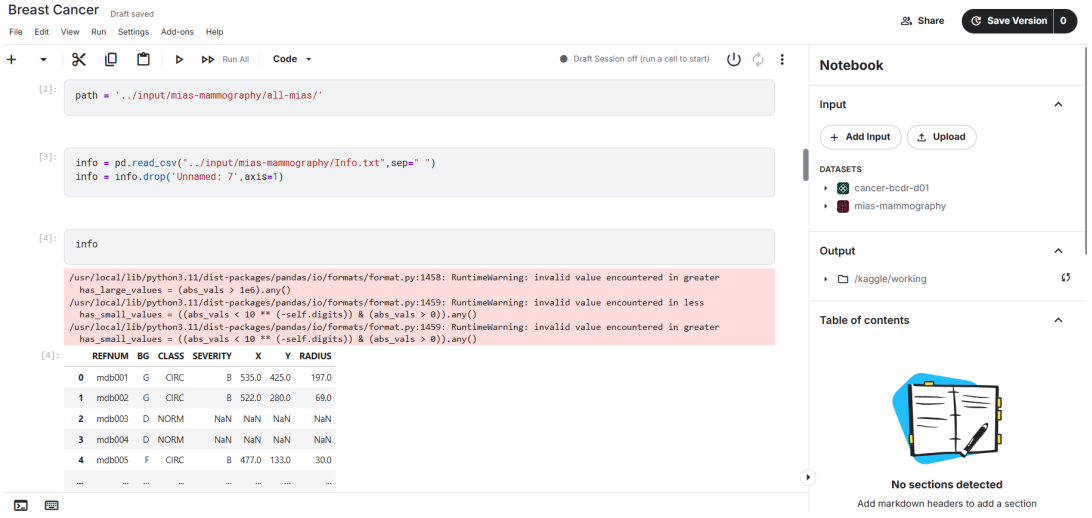
Run Inference

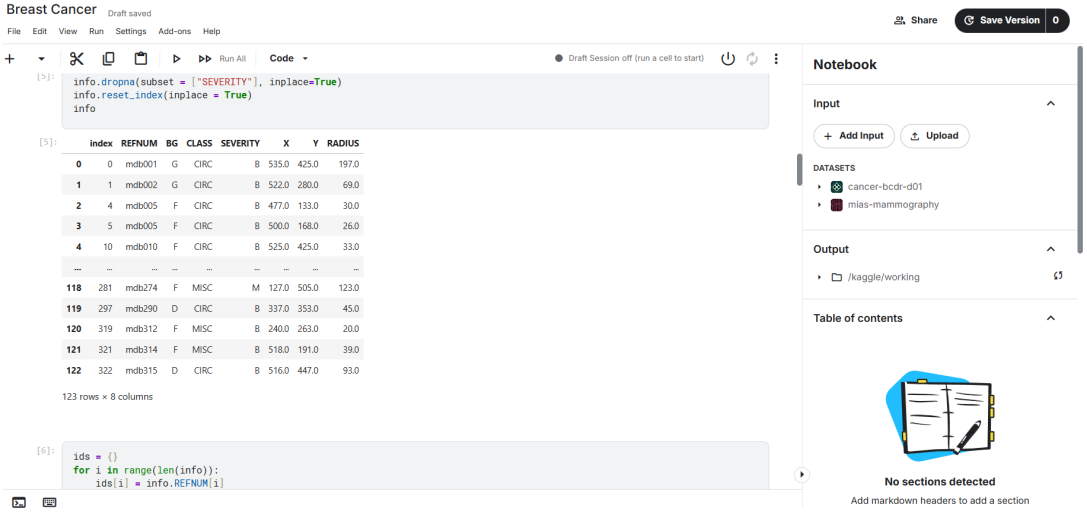
Prediction: Benign (Confidence: 86.07%)

6 Code Snippet

6.1 Python Implementation Overview







Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

[7]:
```
label = []
for i in range(len(info)):
    if info.SEVERITY[i] == 'B':
        label.append(1)
    else:
        label.append(0)
```

[8]:
```
label = np.array(label)
```

[9]:
```
label.shape
```

[9]: (123,)

[10]:
```
img_name = []

for i in range(len(label)):
        img_name.append(path + info.REFNUM[i]+ '.pgm')
```

[11]:
```
img_name = np.array(img_name)
```

Notebook

Input

Add Input Upload

DATASETS

cancer-bcdr-d01

mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

[29]:
```
aug_imgs = []
last_label = []
for i in range(len(img_name)):

    img = cv2.imread(img_name[i], 0)
    img = cv2.resize(img, (224,224))
    rows, cols= img.shape
    for angle in range(360):
            M = cv2.getRotationMatrix2D((cols / 2, rows / 2), angle, 1)
            img_rotated = cv2.warpAffine(img, M, (224, 224))
            aug_imgs.append(img_rotated)
            if label[i] == 1:
                last_label.append(1)
            else:
                last_label.append(0)

    img_flip = cv2.flip(img, 1)
    aug_imgs.append(img_flip)
    last_label.append(1 if label[i] == 1 else 0)
```

[30]:
```
len(aug_imgs)
```

[30]: 44403

Notebook

Input

Add Input Upload

DATASETS

cancer-bcdr-d01

mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

[33]:
```
x_train, x_test, y_train, y_test = train_test_split(aug_imgs, last_label, test_size = 0.2, random_state = 42)
```

[34]:
```
len(x_train),len(x_test),len(y_train),len(y_test)
```

[34]: (35522, 8881, 35522, 8881)

[35]:
```
(a,b,c)=x_train.shape
x_train = np.reshape(x_train, (a, b, c, 1))
(a, b, c)=x_test.shape
x_test = np.reshape(x_test, (a, b, c, 1))
```

[36]:
```
x_train.shape
```

[36]: (35522, 224, 224, 1)

[44]:
```
from keras.layers import Dense, Dropout, Activation, Conv2D, MaxPool2D, Flatten
def create_model2():
    model = Sequential()
    model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',input_shape=(224, 224, 1)))
```

Notebook

Input

Add Input Upload

DATASETS

cancer-bcdr-d01

mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

[44]:
```
from keras.layers import Dense, Dropout, Activation, Conv2D, MaxPool2D, Flatten
def create_model2():
    model = Sequential()
    model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',input_shape=(224, 224, 1)))
    model.add(Conv2D(64, kernel_size=(3,3),activation='relu'))
    model.add(MaxPool2D(pool_size=(2, 2)))

    model.add(Conv2D(64, kernel_size=(3,3),activation='relu'))
    model.add(MaxPool2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))

    model.add(Dense(64, activation='relu'))
    model.add(Dropout(0.25))
    model.add(Flatten())
    model.add(Dense(1, activation='sigmoid'))
    return model
```

[45]:
```
model2 = create_model2()
model2.summary()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
I0000 00:00:1747250821.631205      31 gpu_device.cc:2022] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 15513 MB memory:  -> device: 0, name: Tesla P100-PCIE-16GB, pci bus id: 0000:00:04.0, compute capability: 6.0
Model: "sequential"
```

Notebook

Input

+ Add Input Upload

DATASETS

- cancer-bcdr-d01
- mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	320
conv2d_1 (Conv2D)	(None, 220, 220, 64)	18,496
max_pooling2d (MaxPooling2D)	(None, 110, 110, 64)	0
conv2d_2 (Conv2D)	(None, 108, 108, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
dropout (Dropout)	(None, 54, 54, 64)	0
dense (Dense)	(None, 54, 54, 64)	4,160
dropout_1 (Dropout)	(None, 54, 54, 64)	0
flatten (Flatten)	(None, 186624)	0
dense_1 (Dense)	(None, 1)	186,625

Total params: 246,529 (963.00 KB)
Trainable params: 246,529 (963.00 KB)
Non-trainable params: 0 (0.00 B)

[39]:
```
early_stop = EarlyStopping(monitor='val_loss', mode='min', patience=5,restore_best_weights=True, verbose=1)

check_point_filepath = 'aug_model.keras'

model_check_point = ModelCheckpoint(filepath =check_point_filepath, monitor='val_loss', verbose=1, save_best_only=True,
                                    save_weights_only=False, mode='auto', save_freq='epoch')
```

Notebook

Input

+ Add Input Upload

DATASETS

- cancer-bcdr-d01
- mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

```
445/445 ━━━━━━━━━━━━━━━━━━━━ 44s 100ms/step - accuracy: 0.9894 - loss: 0.0334 - val_accuracy: 0.9804 - val_loss: 0.0781
Epoch 44/100
444/445 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.9870 - loss: 0.0384
Epoch 44: val_loss did not improve from 0.06725
445/445 ━━━━━━━━━━━━━━━━━━━━ 44s 100ms/step - accuracy: 0.9870 - loss: 0.0384 - val_accuracy: 0.9780 - val_loss: 0.0841
Epoch 44: early stopping
Restoring model weights from the end of the best epoch: 39.
```

[41]:
```
loss_value , accuracy = model2.evaluate(x_test, y_test)

print('Test_loss_value = ' +str(loss_value))
print('test_accuracy = ' + str(accuracy))

y_pred_prob = model2.predict(x_test)
y_pred = (y_pred_prob > 0.5).astype("int32")
```

```
278/278 ━━━━━━━━━━━━━━━━━━━━ 6s 15ms/step - accuracy: 0.9774 - loss: 0.0866
Test_loss_value = 0.0938863456249237
test_accuracy = 0.9782682061195374
278/278 ━━━━━━━━━━━━━━━━━━━━ 3s 11ms/step
```

[42]:
```
y_test_flat = y_test.flatten()
y_pred_flat = y_pred.flatten()

print("Accuracy:", accuracy_score(y_test_flat, y_pred_flat))
print("Precision:", precision_score(y_test_flat, y_pred_flat))
print("Recall:", recall_score(y_test_flat, y_pred_flat))
print("F1 Score:", f1_score(y_test_flat, y_pred_flat))
```

Notebook

Input

+ Add Input Upload

DATASETS

- cancer-bcdr-d01
- mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

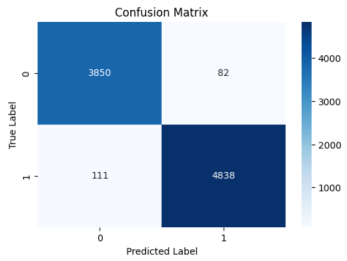
Share Save Version 0

Run All Code Draft Session off (run a cell to start)

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.98      0.98      3932
           1       0.98      0.98      0.98      4949

    accuracy                           0.98      8881
   macro avg       0.98      0.98      0.98      8881
weighted avg       0.98      0.98      0.98      8881
```



Notebook

Input

+ Add Input Upload

DATASETS

cancer-bcdr-d01

mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

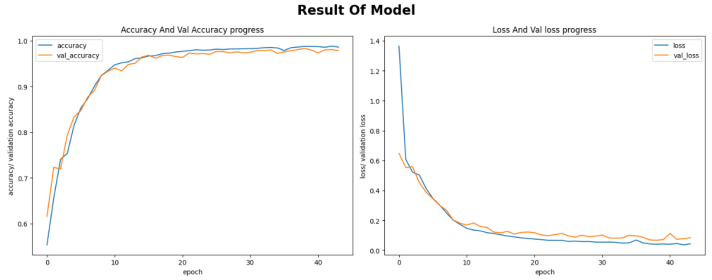
Share Save Version 0

Run All Code Draft Session off (run a cell to start)

```
    axz.legend()

    fig.suptitle('Result Of Model', fontsize = 20, fontweight = 'bold')
    fig.savefig('Accuracy_Loss_figure.png')
    plt.tight_layout()
    plt.show()

visualize_result = Visualize_Result(hist.history['accuracy'],hist.history['val_accuracy'], hist.history['loss'], hist.histor
```



Notebook

Input

+ Add Input Upload

DATASETS

cancer-bcdr-d01

mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

[2]:
```
feature_file_path1 = "/kaggle/input/cancer-bcdr-d01/BCDR-D01_dataset/bcdr_d01_features.csv"
```

[3]:
```
use_cols = ["patient_id", "study_id", "classification"]
file1 = pd.read_csv(feature_file_path1, usecols=use_cols)
```

[4]:
```
file1
```

[4]:

	patient_id	study_id	classification
0	3	4	Malign
1	3	4	Malign
2	4	8	Benign
3	4	8	Benign
4	5	10	Benign
...	...	...	...
138	254	337	Benign
139	255	338	Benign
140	255	338	Benign
141	255	338	Benign

Notebook

Input

+ Add Input Upload

DATASETS

cancer-bcdr-d01

mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

```
n_class_per_patient = file1.groupby('patient_id')['classification'].nunique()

inconsistent = n_class_per_patient[n_class_per_patient > 1]

if inconsistent.empty:
    print("All patients have a single, consistent classification.")
else:
    print("These patient_id(s) have conflicting classifications:")
    print(inconsistent)
```

```
These patient_id(s) have conflicting classifications:
patient_id
18     2
251    2
Name: classification, dtype: int64
```

```
to_drop = [18, 251]

file1 = file1[~file1['patient_id'].isin(to_drop)].reset_index(drop=True)
```

```
file1 = (
    file1[['patient_id', 'classification']]
    .drop_duplicates(subset='patient_id', keep='first')
    .reset_index(drop=True)
)
file1
```

Notebook

Input

Add Input Upload

DATASETS

- cancer-bcdr-d01
- mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

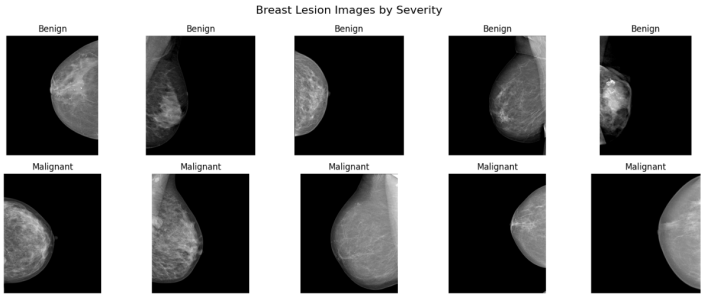
File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

```
    axes[1, i].set_title('Malignant')
    axes[1, i].axis('off')

plt.tight_layout()
plt.show()
```



Notebook

Input

Add Input Upload

DATASETS

- cancer-bcdr-d01
- mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

```
model5 = create_model2()
model5.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 222, 222, 32)	320
conv2d_4 (Conv2D)	(None, 220, 220, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 110, 110, 64)	0
conv2d_5 (Conv2D)	(None, 108, 108, 64)	36,928
max_pooling2d_3 (MaxPooling2D)	(None, 54, 54, 64)	0
dropout_2 (Dropout)	(None, 54, 54, 64)	0
dense_2 (Dense)	(None, 54, 54, 64)	4,160
dropout_3 (Dropout)	(None, 54, 54, 64)	0
flatten_1 (Flatten)	(None, 186624)	0
dense_3 (Dense)	(None, 1)	186,625

Total params: 246,529 (963.00 KB)
Trainable params: 246,529 (963.00 KB)
Non-trainable params: 0 (0.00 B)

Notebook

Input

Add Input Upload

DATASETS

- cancer-bcdr-d01
- mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

[49]:
```
(a,b,c)=x_train.shape
x_train = np.reshape(x_train, (a, b, c, 1))
(a, b, c)=x_test.shape
x_test = np.reshape(x_test, (a, b, c, 1))
```

[50]:
```
len(x_train),len(x_test),len(y_train),len(y_test)
```

[50]: (75088, 18772, 75088, 18772)

[51]:
```
x_train.shape
```

[51]: (75088, 224, 224, 1)

[52]:
```
early_stop = EarlyStopping(monitor='val_loss', mode='min', patience=5,restore_best_weights=True, verbose=1)

check_point_filepath = 'aug_model-bcdr-d01.keras'

model_check_point = ModelCheckpoint(filepath =check_point_filepath, monitor='val_loss', verbose=1, save_best_only=True,
                                    save_weights_only=False, mode='auto', save_freq='epoch')
```

Notebook

Input

Add Input Upload

DATASETS

- cancer-bcdr-d01
- mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

Breast Cancer Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session off (run a cell to start)

```
print('Test_loss_value = ' +str(loss_value))
print('test_accuracy = ' + str(accuracy))

y_pred_prob = model5.predict(x_test)
y_pred = (y_pred_prob > 0.5).astype("int32")
```

```
587/587 ━━━━━━━━━━━━━━━━━━━━ 10s 14ms/step - accuracy: 0.9894 - loss: 0.0336
Test_loss_value = 0.031531255692243576
test_accuracy = 0.9898252487182617
587/587 ━━━━━━━━━━━━━━━━━━━━ 7s 11ms/step
```

[55]:
```
y_test_flat = y_test.flatten()
y_pred_flat = y_pred.flatten()

print("Accuracy:", accuracy_score(y_test_flat, y_pred_flat))
print("Precision:", precision_score(y_test_flat, y_pred_flat))
print("Recall:", recall_score(y_test_flat, y_pred_flat))
print("F1 Score:", f1_score(y_test_flat, y_pred_flat))

print("\nClassification Report:\n", classification_report(y_test_flat, y_pred_flat))

cm = confusion_matrix(y_test_flat, y_pred_flat)

plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=[0, 1], yticklabels=[0, 1])
plt.title("Confusion Matrix")
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.show()
```

Notebook

Input

Add Input Upload

DATASETS

- cancer-bcdr-d01
- mias-mammography

Output

/kaggle/working

Table of contents

No sections detected

Add markdown headers to add a section

7 Results

This section presents the evaluation of the proposed model on the chosen datasets. Table 2 summarizes the performance metrics.

To further analyze model behavior, the corresponding confusion matrix along with training graphs are provided for each dataset to offer deeper insights.

Table 2: Classification results on different datasets

Dataset	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F_1-score (%)**
RSNA	97.85	96.77	97.13	96.95
BCDR	98.20	98.91	98.97	98.94
Mini-MIAS	97.82	98.33	97.75	98.04

7.1 RSNA (DS-1) Results

For the RSNA dataset, our ensemble of four networks yielded 97.85 % accuracy. Figure 4 shows the confusion matrix on the test set.

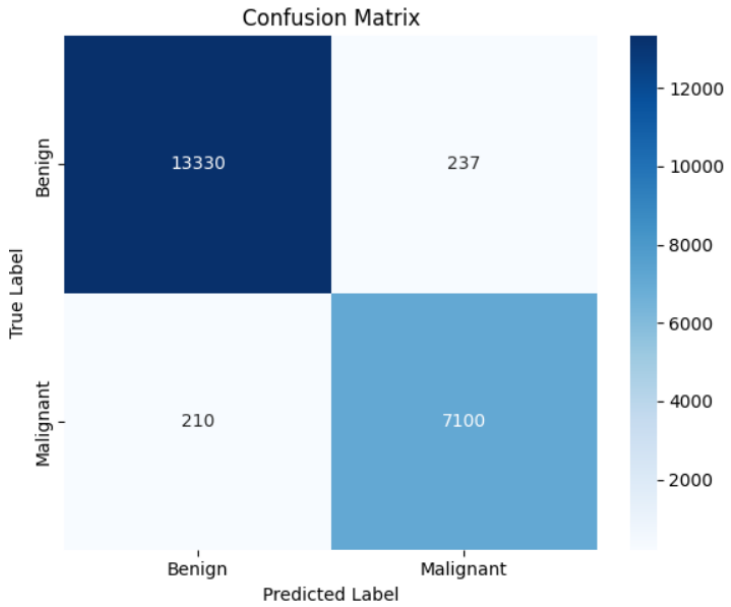


Figure 4: Confusion matrix for RSNA ensemble on the test set.

7.2 BCDR-D01 (DS-2) Results

Figure 5 shows the confusion matrix for our custom CNN on the BCDR-D01 test set. Figure 6 plots the training vs. validation accuracy and loss over epochs. Out of 18581

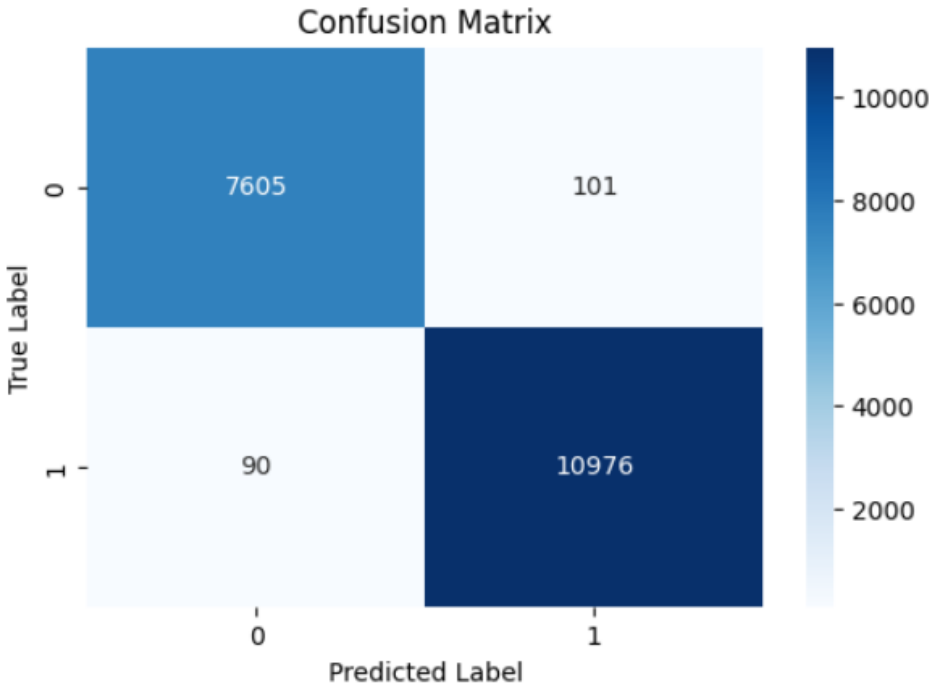


Figure 5: Confusion matrix for BCDR-D01 test set.

test cases, the model correctly identified 10976 malignant and 7605 benign samples, with only 101 false positives and 90 false negatives, underscoring its strong sensitivity and specificity.

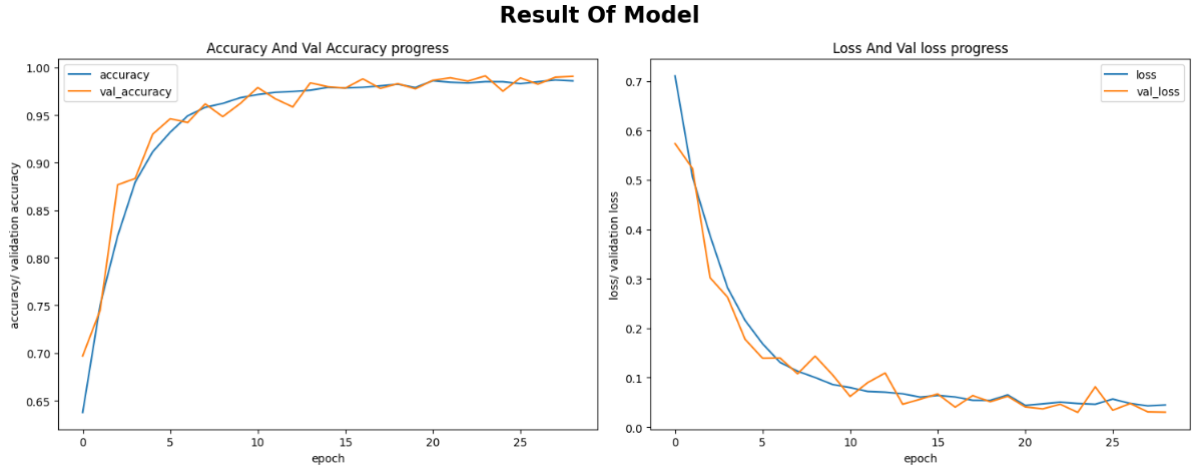


Figure 6: Progress of BCDR model

7.3 Mini-MIAS (DS-3) Results

Figure 7 presents the confusion matrix for Mini-MIAS, and Figure 8 shows its accuracy and loss curves. Our CNN achieved 97.8% accuracy, with just 82 false positives and 111 false negatives out of 8881 samples. The closely tracking training and validation curves

reaches above 95% accuracy and drops loss to near zero, confirming robust generalization.

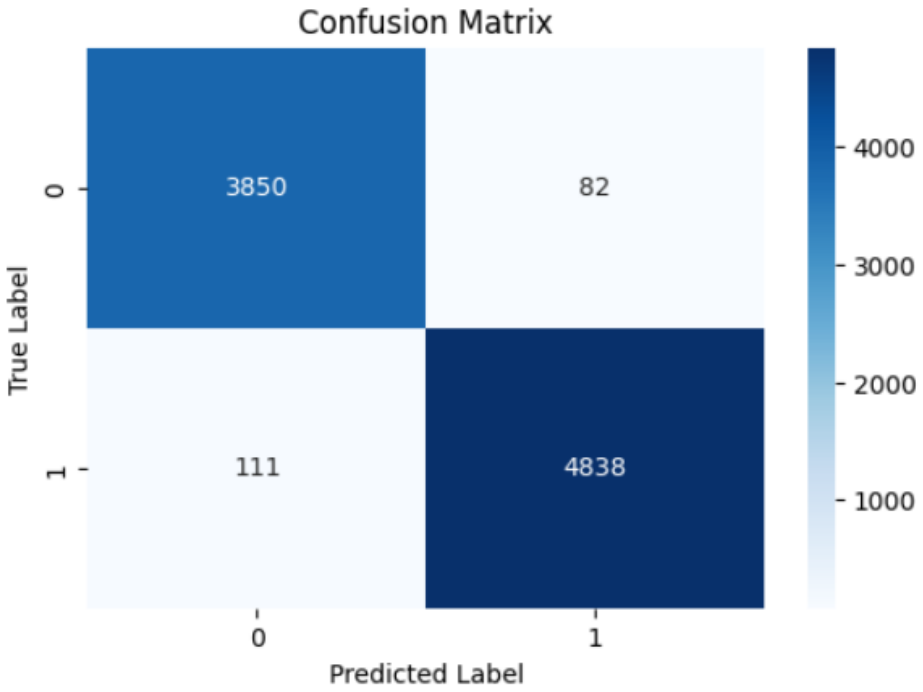


Figure 7: Confusion matrix for Mini-MIAS test set.

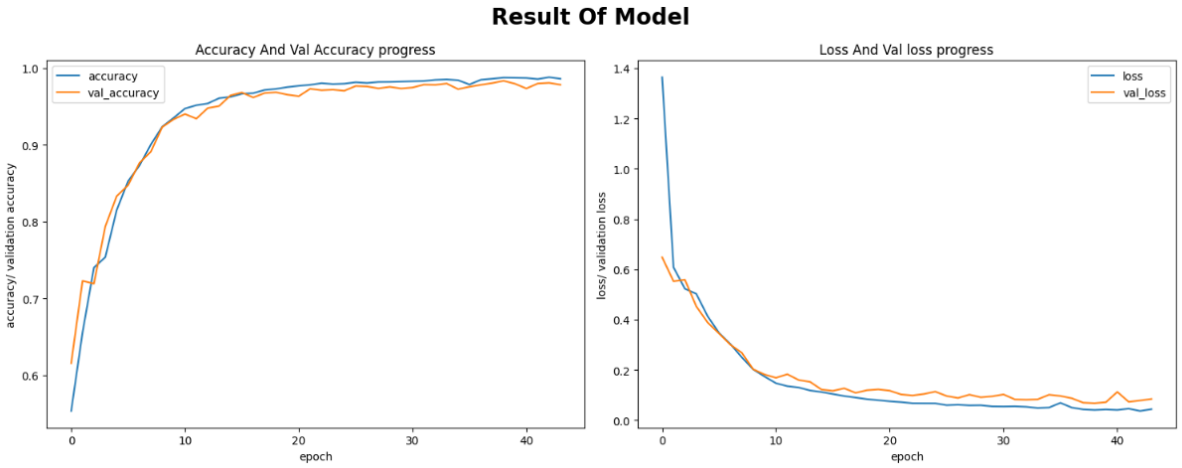


Figure 8: Progress for MIAS model

Table 3 summarizes the accuracy of our model along with those reported in previous studies on BCDR-D01, Mini-MIAS and RSNA data sets.

Table 3: Comparison of Research Papers Using Different Datasets

Research Paper	**Dataset Used**	**Algorithm Used**	**Accuracy**
[18]	BCDR	CNN + RNN	85%
[14]	BCDR	K-means ++ CSO	96.92%
This Study	BCDR	Wiener Filter + CNN	98.20%
[7]	Mini-MIAS	HCNN	96.12%
[13]	Mini-MIAS	GA + ANN	96.20%
This Study	Mini-MIAS	Wiener Filter + CNN	97.80%
[11]	RSNA	EfficientNet	95%
[8]	RSNA	DenseNet + ResNet	96%
This Study	RSNA	DCGAN + Ensemble Models	97.85%

On all three benchmark datasets—BCDR-D01, Mini-MIAS, and RSNA—our suggested architectures perform better than the previously published work, indicating the validity of combining traditional image-processing concepts with contemporary deep learning algorithms.

On the BCDR-D01 dataset, previous results utilizing a vanilla CNN+RNN pipeline were 85 percent accurate, whereas the hybrid clustering-based method (K-means++ with Cat Swarm Optimization) was 96.92 percent accurate. Our work, incorporating a Wiener filter for noise removal leading to a custom CNN, further improves performance to 98.20 percent.

For Mini-MIAS, two competitive baselines were a hierarchical CNN (HCNN) with accuracy of 96.12 percent and a genetic-algorithm-optimized artificial neural network with 96.20 percent. Our Wiener-filter-preprocessed CNN extends this frontier to 97.80 percent, showing that even a simple denoising process can improve classification accuracy by a substantial 1.6 points.

Lastly, on the big dataset of RSNA, a state-of-the-art backbone network—EfficientNet and a DenseNet+ResNet ensemble—reported 95 percent and 96 percent accuracy, respectively. Taking advantage of a DCGAN in realistic data augmentation and an ensemble of classifiers, our model predicts with 97.85 percent accuracy, improving over the current best by close to 1.9 points.

In brief, on small (Mini-MIAS), medium (BCDR-D01), and large (RSNA) mammography datasets, our pipeline, rooted in Wiener filtering or generative augmentation with deep learning, achieves stable and significant accuracy gains, highlighting the utility of hybrid classical-deep methods for breast cancer image classification.

8 Conclusion and future work

This study achieved the creation of a deep learning model for breast cancer classification with an accuracy of 97.85% for RSNA, 97.8% for mini-MIAS, and 98.20% for BCDR. The preprocessing using Wiener filter and DCGAN resulted in a highly accurate, non-invasive solution utilizing custom CNN and ensemble learning. It overcomes the constraints of conventional approaches in minimizing false positives while improving detection at earlier stages.

Following observation were made from the results:

- The models shows high performance metrics across all the three datasets, indicating model's robustness in classifying cancer as benign or malignant.
- Among the 3 models, BCDR has achieved the best results with accuracy 98.20%, precision 98.91%, recall 98.97% and f1-score 98.94%.
- The models are showing balanced precision and recall indicating robustness.

Given the high performance of models across the datasets, shows the models could be used for real time clinical applications. However, the clinical trials with large and real datasets will be required to confirm its generalizability. In addition to this we could integrate optimization algorithms like Swarm Intelligence in our model for hyperparameter tuning in case of large datasets to give satisfactory results.

References

[1] World Health Organization. *Breast cancer*. Accessed: 2025-05-15. 2024. URL: https://www.who.int/news-room/fact-sheets/detail/breast-cancer.

[2] Fatima Noreen, Li Liu, Hong Sha, and Haroon Ahmed. "Prediction of Breast Cancer, Comparative Review of Machine Learning Techniques, and Their Analysis". In: *IEEE Access* PP (Aug. 2020), pp. 1–1. DOI: 10.1109/ACCESS.2020.3016715.

[3] Sergiusz Łukasiewicz, Marcin Czeczelewski, Alicja Forma, Jacek Baj, Robert Sitarz, and Andrzej Stanisławek. "Breast cancer—epidemiology, risk factors, classification, prognostic markers, and current treatment strategies—an updated review". In: *Cancers* 13.17 (2021), p. 4287.

[4] Nusrat Mohi ud din, Rayees Ahmad Dar, Muzafar Rasool, and Assif Assad. "Breast cancer detection using deep learning: Datasets, methods, and challenges ahead". In: *Computers in Biology and Medicine* 149 (2022), p. 106073. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2022.106073. URL: https://www.sciencedirect.com/science/article/pii/S0010482522007818.

[5] Qasem Abu Al-Haija and Adeola Adebanjo. "Breast Cancer Diagnosis in Histopathological Images Using ResNet-50 Convolutional Neural Network". In: *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. 2020, pp. 1–7. DOI: 10.1109/IEMTRONICS51293.2020.9216455.

[6] V. Jyoshna and V. Shruthikamal. "Incidence of Benign and Malignant Lesions of Breast". In: *Journal of Pharmaceutical Research International* 33.63B (Dec. 2021), pp. 120–125. DOI: 10.9734/jpri/2021/v33i63B35260. URL: https://journaljpri.com/index.php/JPRI/article/view/5531.

[7] Pramoda Patro, Shaik Fathima, R. Harikishore, and Aditya Kumar Sahu. "Breast cancer image classification by using HCNN and LeNet5". In: *Discover Sustainability* 5 (Dec. 2024). DOI: 10.1007/s43621-024-00725-1.

[8] Dilawar Shah, Mohammad Khan, Muhammad Abrar, and Muhammad Tahir. "Optimizing Breast Cancer Detection With an Ensemble Deep Learning Approach". In: *International Journal of Intelligent Systems* 2024 (Oct. 2024). DOI: 10.1155/2024/5564649.

[9] Erkan Deniz, Abdulkadir Şengür, Zehra Kadiroğlu, Yanhui Guo, Varun Bajaj, and Ümit Budak. "Transfer learning based histopathologic image classification for breast cancer detection". In: *Health information science and systems* 6.1 (Dec. 2018), p. 18. ISSN: 2047-2501. DOI: 10.1007/s13755-018-0057-x. URL: https://europepmc.org/articles/PMC6162199.

[10] Abhishek Das, Mihir Narayan Mohanty, Pradeep Kumar Mallick, Prayag Tiwari, Khan Muhammad, and Hongyin Zhu. "Breast cancer detection using an ensemble deep learning method". In: *Biomedical Signal Processing and Control* 70 (2021), p. 103009. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2021.103009. URL: https://www.sciencedirect.com/science/article/pii/S1746809421006066.

[11] Hoang Nhut Huynh, Ngoc An Dang Nguyen, Anh Tu Tran, Van Chinh Nguyen, and Trung Nghia Tran. "Classification of Breast Cancer Using Radiological Society of North America Data by EfficientNet". In: *Engineering Proceedings* 55.1 (2023). ISSN: 2673-4591. DOI: 10.3390/engproc2023055006. URL: https://www.mdpi.com/2673-4591/55/1/6.

[12] Jhelly-Reynaluz Pérez-Núñez, Ciro Rodríguez, Luis-Javier Vásquez-Serpa, and Carlos Navarro. "The Challenge of Deep Learning for the Prevention and Automatic Diagnosis of Breast Cancer: A Systematic Review". In: *Diagnostics* 14.24 (2024). ISSN: 2075-4418. DOI: 10.3390/diagnostics14242896. URL: https://www.mdpi.com/2075-4418/14/24/2896.

[13] Dilovan Asaad Zebari, Dheyaa Ahmed Ibrahim, Diyar Qader Zeebaree, Mazin Abed Mohammed, Habibollah Haron, Nechirvan Asaad Zebari, Robertas Damaševičius, and Rytis Maskeliūnas. "Breast Cancer Detection Using Mammogram Images with Improved Multi-Fractal Dimension Approach and Feature Fusion". In: *Applied Sciences* 11.24 (2021). ISSN: 2076-3417. DOI: 10.3390/app112412122. URL: https://www.mdpi.com/2076-3417/11/24/12122.

[14] Kittipol Wisaeng. "Breast Cancer Detection in Mammogram Images Using K–Means++ Clustering Based on Cuckoo Search Optimization". In: *Diagnostics* 12.12 (2022). ISSN: 2075-4418. DOI: 10.3390/diagnostics12123088. URL: https://www.mdpi.com/2075-4418/12/12/3088.

[15] Theoviel. *RSNA Breast Cancer - 512 PNGs*. Accessed: 2025-05-15. 2023. URL: https://www.kaggle.com/datasets/theoviel/rsna-breast-cancer-512-pngs.

[16] Breast Cancer Digital Repository Consortium. *Breast Cancer Digital Repository (BCDR)*. Accessed: 2025-05-15. 2012. URL: https://bcdr.eu/information/about.

[17] kmader. *MIAS Mammography*. Accessed: 2025-05-15. 2017. URL: https://www.kaggle.com/datasets/kmader/mias-mammography.

[18] Hongyu Wang, Jun Feng, Zizhao Zhang, Hai Su, Lei Cui, Hua He, and Li Liu. "Breast mass classification via deeply integrating the contextual information from multi-view data". In: *Pattern Recognition* 80 (2018), pp. 42–52. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2018.02.026. URL: https://www.sciencedirect.com/science/article/pii/S0031320318300785.