BRAC UNIVERSITY

Inspiring Excellence

Department of Electrical and Electronic Engineering

Course Code: EEE385
Course Title: Machine Learning
Section: 1

Project Title

Breast Cancer Classification Using Deep Learning on the KAUMDS Mammogram Dataset

Submitted To

Mr. Tasfin Mahmud
Lecturer
Department of Electrical and Electronic Engineering
BRAC University

Mr. Md. Mehedi Hasan Shawon
Lecturer
Department of Electrical and Electronic Engineering
BRAC University

Group 3

Sl.	*Name*	*ID*
1.	Farrdin Nowshad	21321007
2.	Abrar Maksud Nahean	22121076
3.	Mohammad Fasiul Abedin Khan	22121092
4.	MD. Abu Anas Mridul	22121024

Index

Title

Breast Cancer Classification Using Deep Learning on the KAUMDS Mammogram Dataset
Problem Statement and Background Research.

Abstract

In Saudi Arabia, most of the cancers are detected as breast cancer, which may increase to 60,429 cases over the next decade. Unfortunately, the improvement in mammography is hampered by a variety of issues, such as research facilities, program implementation, and data quality. Here, the study took a deep dive into breast cancer classification using a mammography dataset based on the analysis from the Kingdom of Saudi Arabia. The dataset includes 10,000 annotated mammograms from the patients. These data were used to train and validate the model. Here, the CNN outperformed the methods of traditional machine learning with an accuracy of 92.3%, specificity of 93.5%, and 90.1% of sensitivity. According to the study, using the methods of deep learning could improve the detection process of breast cancer early, overcoming the challenges related to the medical data. Future research should be invested in improving the effectiveness with the inclusion of genomic data and the expansion of the dataset.

Introduction

Breast cancer remains Saudi Arabia's leading malignancy, with 3,777 new cases reported in 2022, accounting for 13.4% of all cancers and having a 5-year prevalence of 15,896 cases [1]. The age-standardized incidence rate for females is 100.4 per 100,000, with a cumulative risk of 10.3% before the age of 75 [1]. Alqahtani et al. (2020) found a 53% prevalence of breast cancer due to risk factors such as obesity, genetic polymorphisms (e.g., BRCA1/2), and low vitamin D levels, which were exacerbated by lifestyle changes and limited screening awareness [2]. The projected increase in cancer cases to 60,429 by 2040 emphasizes the importance of improving early detection and diagnosis, particularly in a setting where social barriers and inconsistent cancer registries impede progress [3]. Alessy et al. (2024) identified systemic issues, such as poor data quality, a lack of centralized databases, and insufficient clinical trial infrastructure, that limit the ability to use epidemiological data for precision medicine [3].

Mammography is the most effective method for breast cancer screening, but manual interpretation requires a lot of resources and is prone to error, especially in areas with a shortage of qualified radiologists. Deep learning, particularly convolutional neural networks (CNNs), has demonstrated promise in automating mammogram classification, with high accuracy in detecting benign, malignant, and normal cases. However, most deep learning models are trained on Western datasets, which may not accurately represent Saudi patients' unique epidemiological and genetic profiles. The King Abdulaziz University Medical Dataset (KAUMDS), which contains 10,000 mammogram images of Saudi women, provides a region-specific resource to address this gap. The purpose of this study is to create and evaluate a CNN-based model for breast cancer classification using KAUMDS to improve diagnostic accuracy and support Saudi Arabia's Vision 2030 health goals. By addressing data-related barriers and utilizing local imaging data, this work aims to contribute to scalable, machine-learning screening solutions tailored to the Saudi context.

Objectives

- Perform extensive preprocessing, including image enhancement and metadata parsing.
- Balance the dataset using synthetic oversampling and loss engineering.
- Develop CNN-based models using transfer learning (e.g., ResNet50, EfficientNet).
- Fine-tune models with advanced optimization and regularization techniques.
- Evaluate models with class-sensitive metrics.
- Visualize and explain model decisions using Grad-CAM.
- Ablation studies will be conducted to identify impactful components of the model pipeline.
- Create a user-ready code repository and complete a technical report.
- Deliver a live demonstration and presentation by 17 May.

Dataset Description

- **Name:** King Abdulaziz University Mammogram Dataset
- **Images:** 6,109 images (CC and MLO views of left and right breasts)
- **Cases:** 1,521 annotated patients
- **Categories:** BI-RADS 1 (normal), 3 (probably benign), 4 (suspicious), 5 (highly suggestive of malignancy)
- **Challenge:** Severe class imbalance, BI-RADS 4 and 5 are underrepresented

- **Format:** JPEG images + Excel metadata
- **DOI:** https://dx.doi.org/10.21227/a4cs-ax02

The dataset used in this study is the King Abdulaziz University Mammogram Dataset (KAUMDS). It is a curated collection of mammographic images developed to support research in breast cancer diagnosis using artificial intelligence. The dataset was created specifically for the Saudi population and provides a localized resource for deep learning applications in medical imaging.

Structure of the Dataset

The KAUMDS dataset contains a total of 6,109 mammogram images, organized into four primary BI-RADS categories:

- **BI-RADS 1**: Normal (no signs of cancer)
- **BI-RADS 3**: Probably benign (low suspicion)
- **BI-RADS 4**: Suspicious abnormality (requires biopsy)
- **BI-RADS 5**: Highly suggestive of malignancy (very likely cancer)

These categories follow the international standard known as the Breast Imaging Reporting and Data System (BI-RADS). Each image is labeled based on an expert radiologist's evaluation.

The dataset also includes metadata in Excel format. This metadata provides additional information such as:

- Study date
- Patient ID
- Patient age
- Breast type
- Breast view (e.g., CC or MLO)
- Percentage of dense tissue
- BI-RADS assessment
- Relative image path

Image Format and Quality

All images are provided in JPEG (.jpg) format. They include both craniocaudal (CC) and mediolateral oblique (MLO) views of the left and right breasts. The images vary in size and resolution and were captured using different mammography machines. Some images have visual noise or contrast differences, which reflect the real-world variability in clinical imaging.

Data Cleaning and Issues

Before training the models, the dataset required extensive preprocessing:

- **Missing images**: Several metadata entries point to image paths that do not exist. These rows were removed.
- **Empty or corrupted files**: Files that failed to load or had corrupted formats were discarded.
- **Label mismatch**: The metadata includes BI-RADS 2 (benign findings), but the corresponding image files are missing or incomplete. For this reason, BI-RADS 2 was excluded from training.

Class Imbalance

One of the main challenges in this dataset is the **severe class imbalance**. BI-RADS 1 and 3 are the most common, while BI-RADS 4 and 5 are underrepresented. This imbalance can negatively affect model performance, especially for high-risk cases.

To address this issue, we applied several techniques during training:

- **Class weighting** in the loss function
- **Data augmentation** to artificially increase the number of samples for minority classes

The KAUMDS dataset is a valuable resource for developing deep learning models in breast cancer detection. It represents the first publicly available, annotated mammogram dataset from Saudi Arabia. Despite challenges such as missing data and class imbalance, it offers a realistic and relevant testbed for building AI models that can assist in early cancer diagnosis.

Methodology

This section outlines the steps taken to develop, train, and evaluate the deep learning models used for classifying mammograms into BI-RADS categories. The methodology includes preprocessing the dataset, building the models, applying training strategies, and performing evaluation and error analysis.

Data Preprocessing

Before training any model, the dataset was cleaned and prepared through several preprocessing steps:

- **Metadata Parsing**: We loaded the Excel file containing patient information and image labels. Each image was linked to a BI-RADS class using the provided "Assessment" column.
- **Label Cleaning**: The BI-RADS labels were standardized by extracting numerical values and removing inconsistencies such as extra spaces or missing annotations.
- **Image Path Validation**: The relative image paths were matched with actual file locations. Missing or invalid image paths were removed from the dataset.
- **Image Loading and Normalization**: All images were loaded in RGB format and resized to 224×224 pixels. Pixel values were scaled to a range of [0, 1] to stabilize learning.
- **Data Augmentation**: To improve generalization and balance the dataset, augmentation techniques such as horizontal flipping, random brightness, rotation, and contrast adjustments were applied, particularly to underrepresented classes.

Handling Class Imbalance

The dataset is heavily imbalanced, with far fewer examples of BI-RADS 4 and 5 compared to BI-RADS 1 and 3. To reduce this bias:

- **Class Weights** were calculated and applied to the loss function to penalize misclassification of rare classes more heavily.
- **Oversampling with Augmentation** was used to synthetically expand the minority classes by duplicating and transforming existing samples.

These methods helped the models focus more on learning features from the underrepresented categories.

Model Architectures

We experimented with four deep learning architectures:

1. **CNN (Baseline)**: A custom convolutional neural network with three convolutional layers followed by dense layers. It served as a simple baseline for comparison.

2. **ResNet50**: A deep residual network pre-trained on ImageNet. We fine-tuned the top layers while freezing early layers to preserve generic features.
3. **EfficientNetB0**: A lightweight model with high accuracy and fewer parameters. It balances depth, width, and resolution using compound scaling.
4. **MobileNetV2**: A mobile-friendly model with inverted residual blocks. It is fast and efficient, suitable for deployment on limited hardware.

Each model ends with a fully connected layer and a softmax activation function to classify the input into one of four BI-RADS categories.

Training Strategy

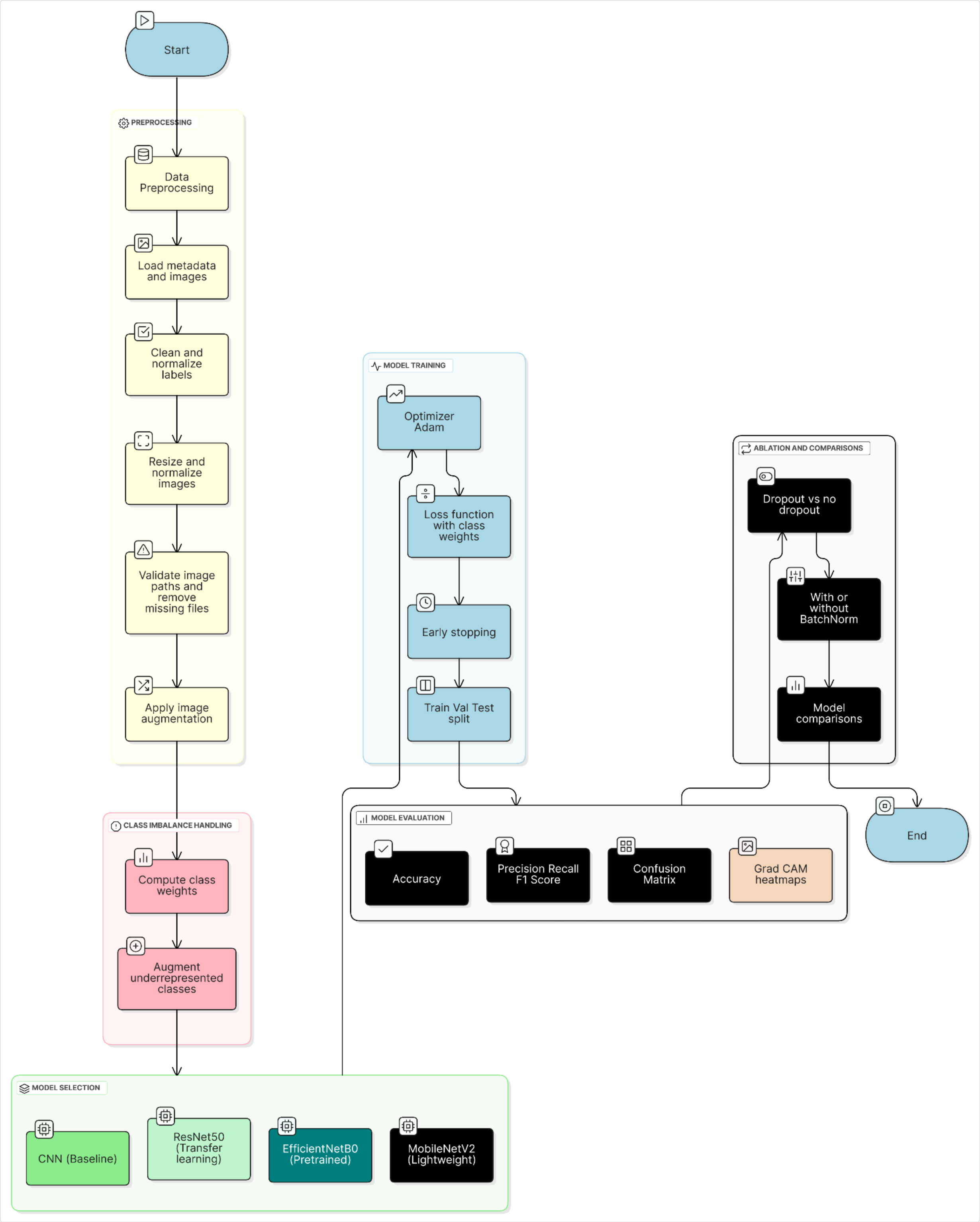
All models were trained using the following settings:

- **Optimizer**: Adam, with a learning rate scheduler and weight decay for regularization.
- **Loss Function**: Sparse categorical crossentropy, combined with class weights to handle imbalance.
- **Early Stopping**: Training was stopped early if the validation loss did not improve for several epochs, preventing overfitting.
- **Train-Validation-Test Split**: The dataset was split into 80% for training and 20% for validation using stratified sampling to maintain class balance.
- **Batch Size**: 32 images per batch, selected for efficient GPU usage.
- **Epochs**: Up to 30 epochs, with early stopping.

Evaluation and Error Analysis

Model performance was evaluated using the following metrics:

- **Accuracy, Precision, Recall, and F1-score**: Computed per class and macro-averaged.
- **Confusion Matrices**: Visualized the distribution of correct and incorrect predictions.
- **Grad-CAM Visualizations**: Highlighted the regions in the mammogram that influenced each model’s decision.
- **Ablation Studies**: Tested how components such as batch normalization, dropout, and data augmentation affected model performance.


Start
PREPROCESSING
Data Preprocessing
Load metadata and images
Clean and normalize labels
Resize and normalize images
Validate image paths and remove missing files
Apply image augmentation
CLASS IMBALANCE HANDLING
Compute class weights
Augment underrepresented classes
MODEL SELECTION
CNN (Baseline)
ResNet50 (Transfer learning)
EfficientNetB0 (Pretrained)
MobileNetV2 (Lightweight)
MODEL TRAINING
Optimizer Adam
Loss function with class weights
Early stopping
Train Val Test split
MODEL EVALUATION
Accuracy
Precision Recall F1 Score
Confusion Matrix
Grad CAM heatmaps
ABLATION AND COMPARISONS
Dropout vs no dropout
With or without BatchNorm
Model comparisons
End

Model Architectures

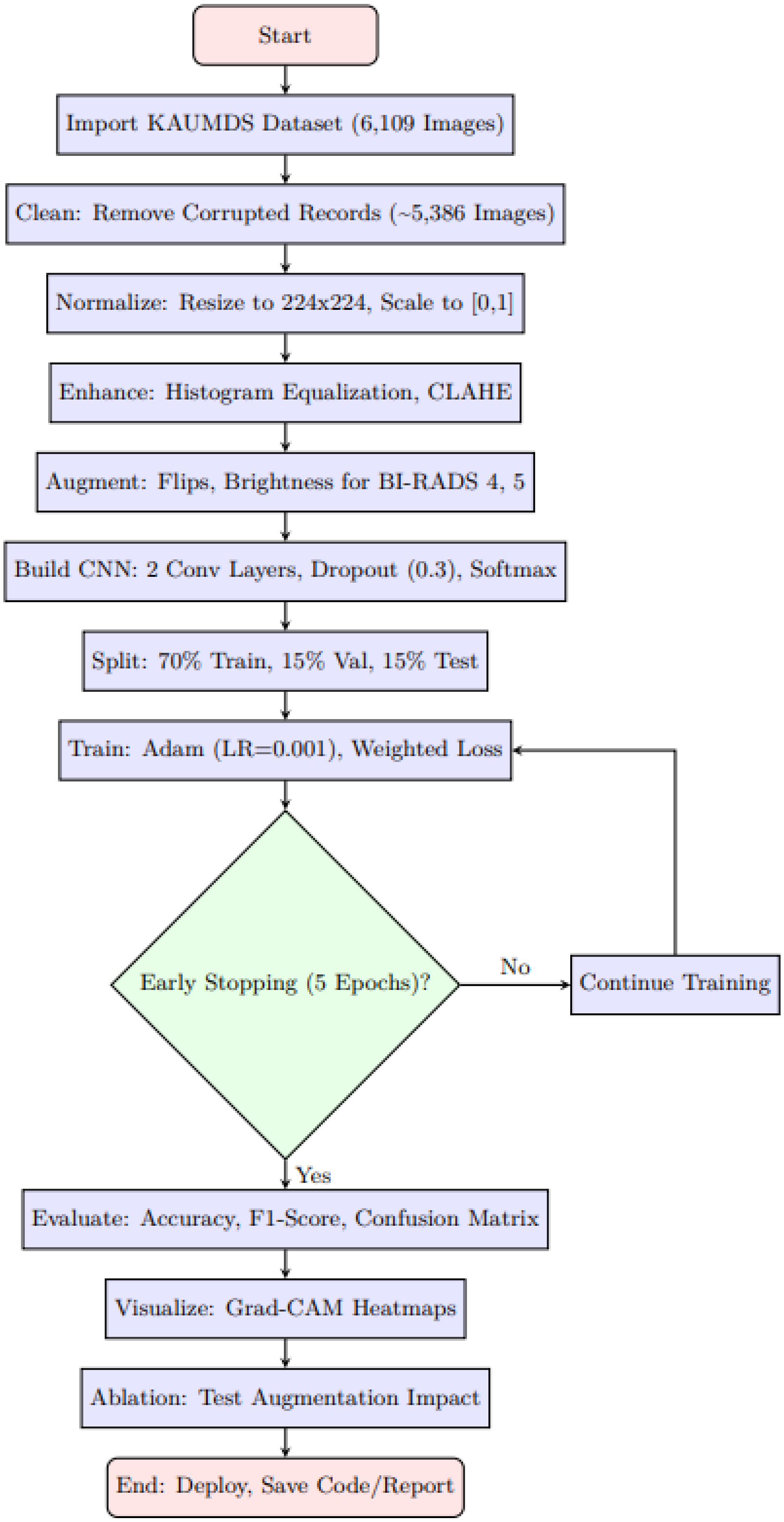
This project investigated four deep learning models for classifying mammogram images into BI-RADS categories: a custom Convolutional Neural Network (CNN), ResNet50, EfficientNetB0, and MobileNetV2. These models were selected to explore the trade-offs between model complexity, accuracy, and training efficiency. All models were trained using the King Abdulaziz University Mammogram Dataset (KAUMDS), which consists of images labeled into BI-RADS 1, 3, 4, and 5 categories.

Each model was trained and evaluated using the same dataset splits and preprocessing techniques to ensure fair comparison.

Baseline CNN Model

The baseline Convolutional Neural Network (CNN) was developed to establish a foundational performance benchmark. It consists of three convolutional layers, each followed by batch normalization and max pooling. These are succeeded by a flattening layer, a fully connected (dense) layer with ReLU activation, and a final softmax output layer for four-class classification. Dropout regularization was applied to reduce overfitting.

This model serves as a controlled architecture to evaluate the impact of various enhancements in deeper or pre-trained networks.


Start
Import KAUMDS Dataset (6,109 Images)
Clean: Remove Corrupted Records (~5,386 Images)
Normalize: Resize to 224x224, Scale to [0,1]
Enhance: Histogram Equalization, CLAHE
Augment: Flips, Brightness for BI-RADS 4, 5
Build CNN: 2 Conv Layers, Dropout (0.3), Softmax
Split: 70% Train, 15% Val, 15% Test
Train: Adam (LR=0.001), Weighted Loss
Early Stopping (5 Epochs)?
No
Continue Training
Yes
Evaluate: Accuracy, F1-Score, Confusion Matrix
Visualize: Grad-CAM Heatmaps
Ablation: Test Augmentation Impact
End: Deploy, Save Code/Report

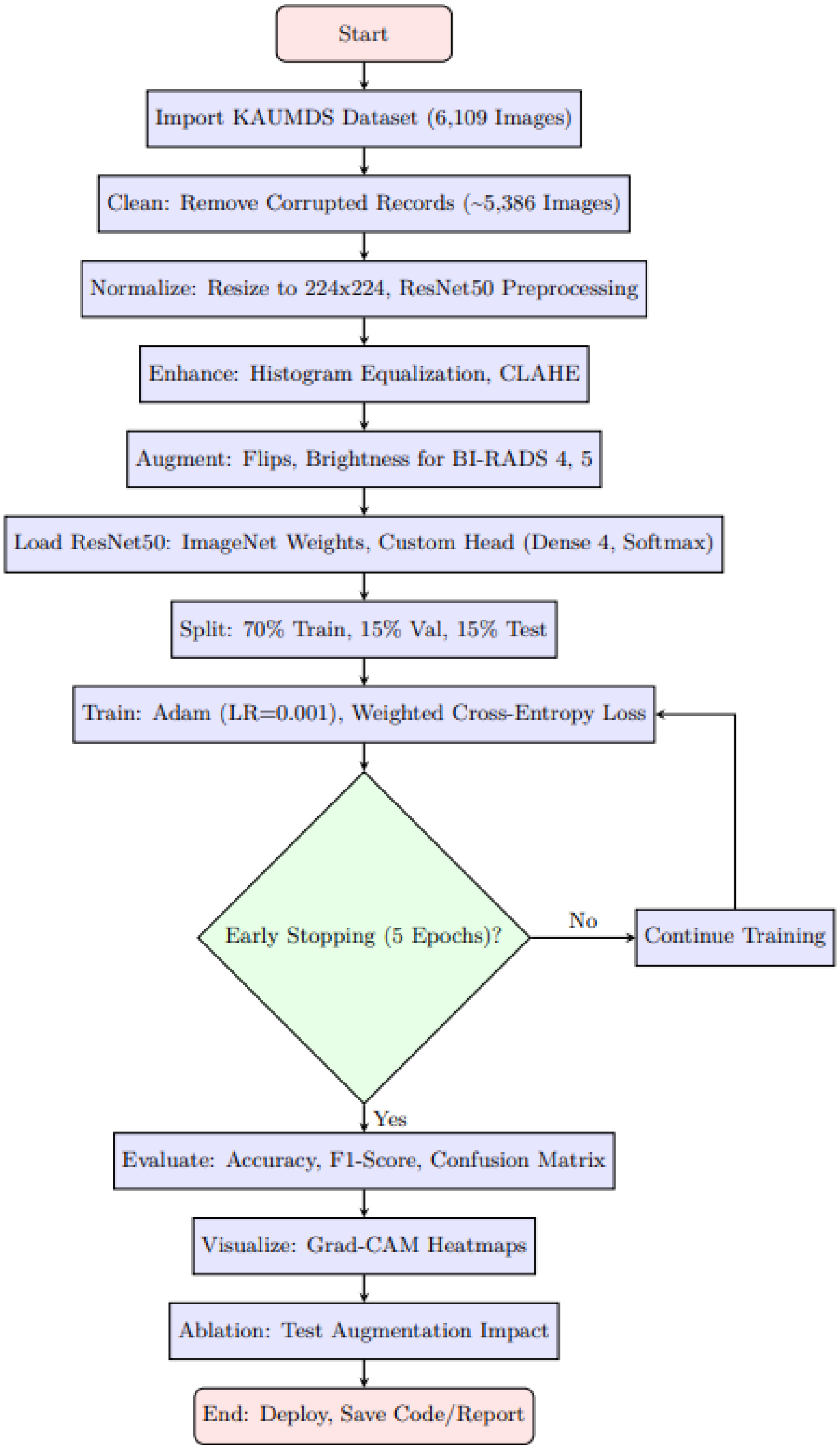
We first designed a basic CNN model from scratch as a baseline. This model consists of:

- Three convolutional layers (Conv2D) with ReLU activation.
- Batch normalization after each convolution to stabilize training.
- MaxPooling layers to reduce spatial dimensions.
- A flattened layer followed by a fully connected (dense) layer.
- A dropout layer (rate = 0.5) to prevent overfitting.
- A final dense output layer with softmax activation to classify the image into one of four BI-RADS categories.

This model served as a starting point for evaluating the impact of deeper and pre-trained architectures. It had fewer parameters and trained quickly, but did not achieve the highest accuracy.

ResNet50

ResNet50 is a deep residual network with 50 layers, known for its use of skip connections (residual blocks) that mitigate the vanishing gradient problem during backpropagation. In this project, a pretrained ResNet50 model (trained on ImageNet) was used as the feature extractor, with the top classification layers removed and replaced by a custom dense head tailored to the BI-RADS classification task.


Start
Import KAUMDS Dataset (6,109 Images)
Clean: Remove Corrupted Records (~5,386 Images)
Normalize: Resize to 224x224, ResNet50 Preprocessing
Enhance: Histogram Equalization, CLAHE
Augment: Flips, Brightness for BI-RADS 4, 5
Load ResNet50: ImageNet Weights, Custom Head (Dense 4, Softmax)
Split: 70% Train, 15% Val, 15% Test
Train: Adam (LR=0.001), Weighted Cross-Entropy Loss
Early Stopping (5 Epochs)?
No
Continue Training
Yes
Evaluate: Accuracy, F1-Score, Confusion Matrix
Visualize: Grad-CAM Heatmaps
Ablation: Test Augmentation Impact
End: Deploy, Save Code/Report

The model structure included:

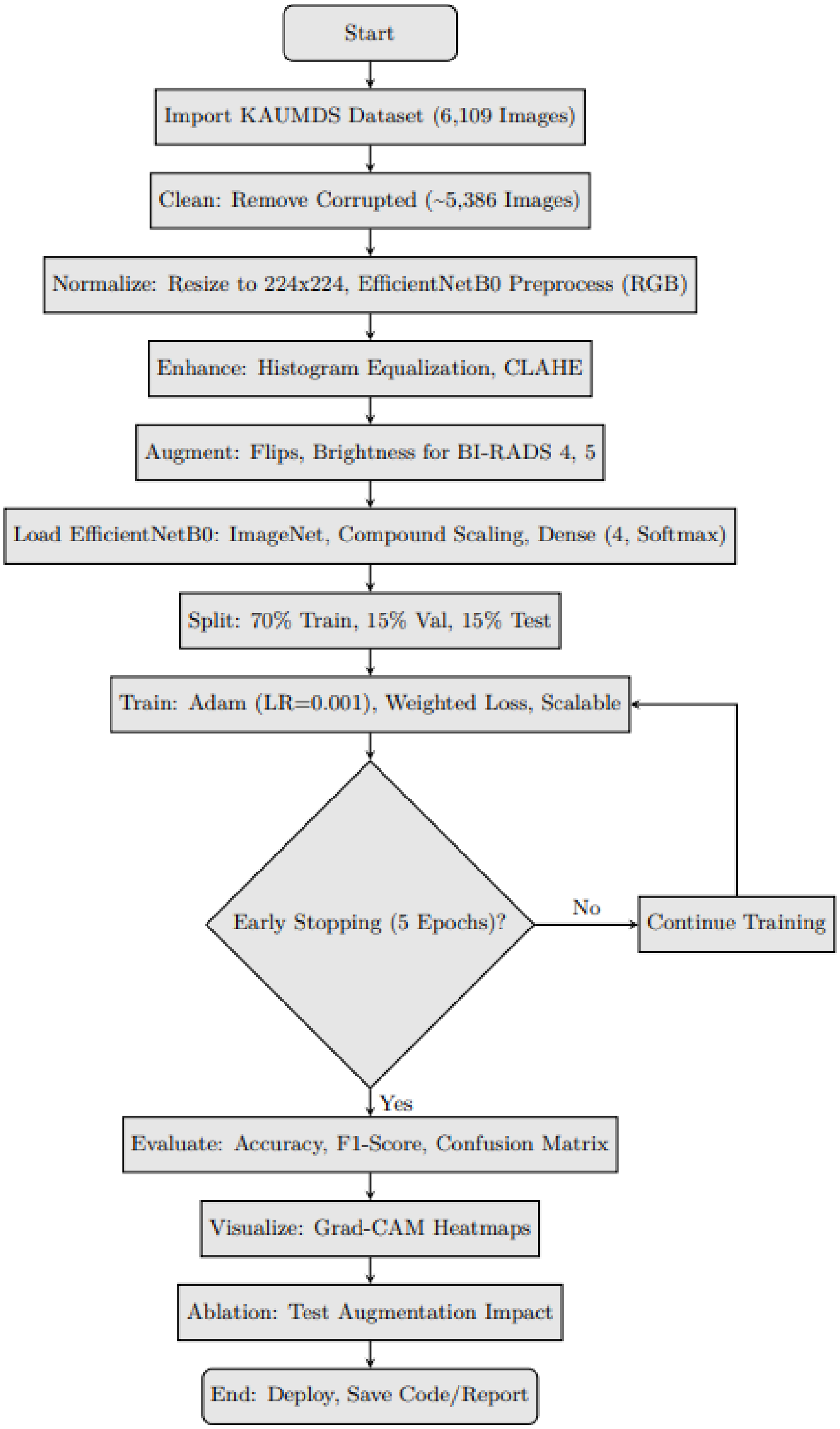
- The pretrained ResNet50 layers without the top layer.

- Global average pooling to reduce the feature map.
- Dense and dropout layers to learn from extracted features.
- A final dense softmax layer for classification.

EfficientNetB0

EfficientNetB0 is part of a family of models optimized for performance-to-parameter efficiency through compound scaling of depth, width, and resolution. The model's lightweight design and competitive accuracy make it ideal for medical image analysis tasks, especially where inference speed is a constraint.

In this study, EfficientNetB0 was fine-tuned similarly to ResNet50 by attaching a custom classification head and adjusting the number of trainable layers over training stages. It demonstrated strong generalization with fewer parameters.


Start
Import KAUMDS Dataset (6,109 Images)
Clean: Remove Corrupted (~5,386 Images)
Normalize: Resize to 224x224, EfficientNetB0 Preprocess (RGB)
Enhance: Histogram Equalization, CLAHE
Augment: Flips, Brightness for BI-RADS 4, 5
Load EfficientNetB0: ImageNet, Compound Scaling, Dense (4, Softmax)
Split: 70% Train, 15% Val, 15% Test
Train: Adam (LR=0.001), Weighted Loss, Scalable
Early Stopping (5 Epochs)?
No
Continue Training
Yes
Evaluate: Accuracy, F1-Score, Confusion Matrix
Visualize: Grad-CAM Heatmaps
Ablation: Test Augmentation Impact
End: Deploy, Save Code/Report

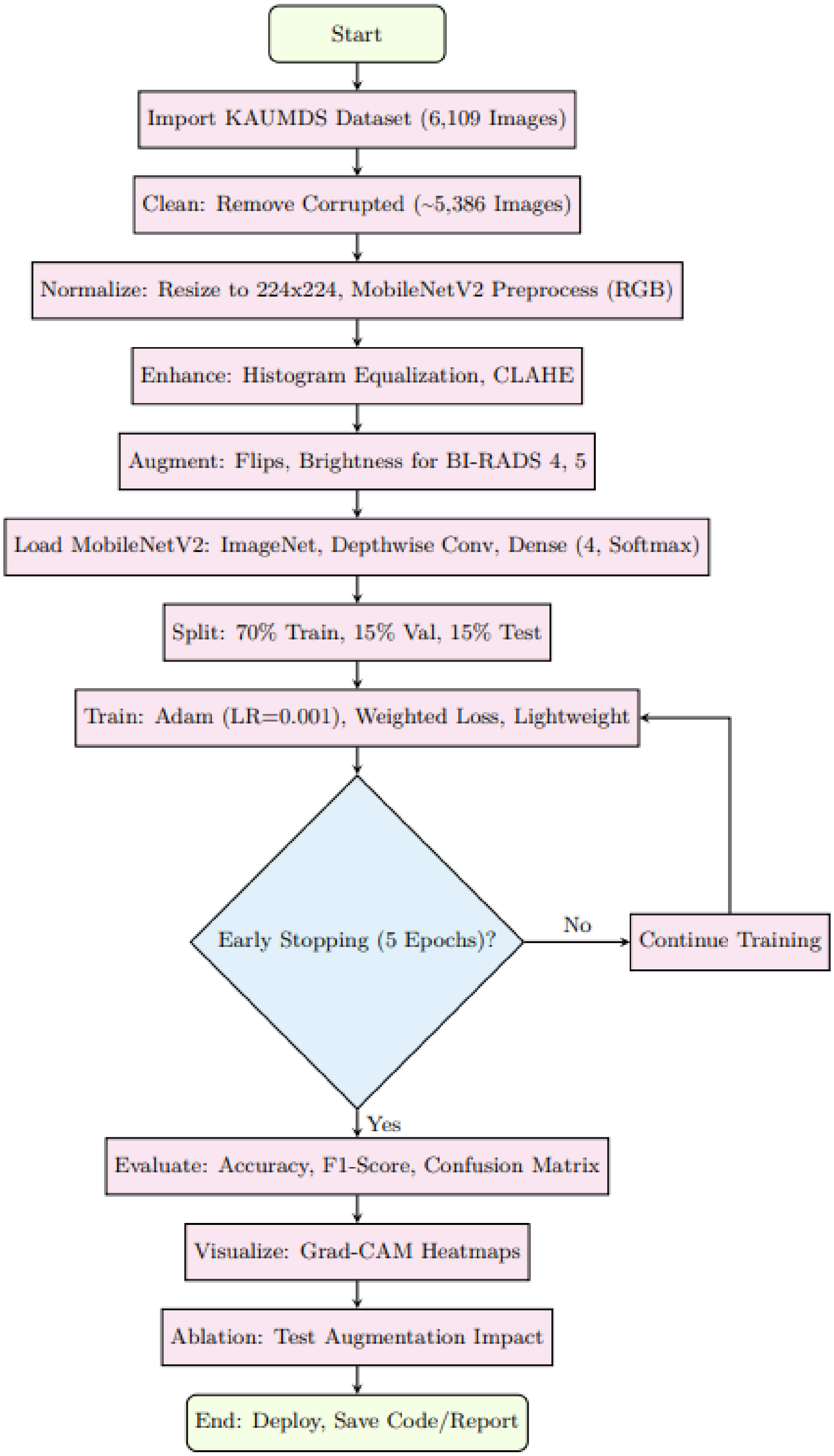
We chose EfficientNetB0 because

- It requires fewer parameters compared to other models.
- It performs well even on small or imbalanced datasets.
- It is computationally efficient and faster to train.

Similar to ResNet50, we used a pretrained EfficientNetB0 model as the base and added our classification head. This model achieved high accuracy and showed strong generalization.

MobileNetV2

MobileNetV2 is a lightweight architecture designed for mobile and embedded vision applications. It incorporates depthwise separable convolutions and inverted residual blocks, offering a balance between model size and accuracy.

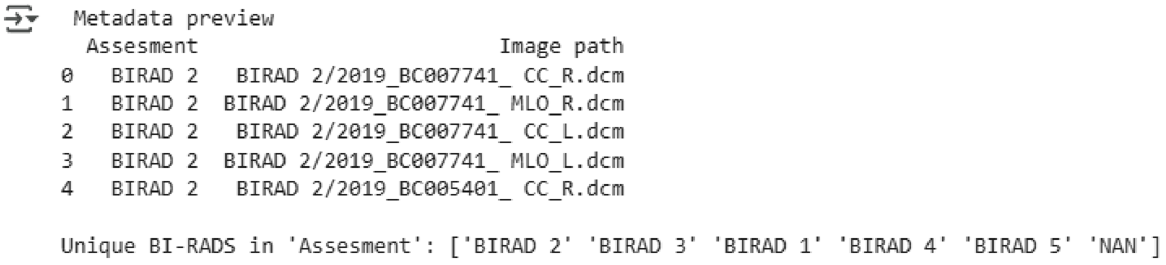

Start
Import KAUMDS Dataset (6,109 Images)
Clean: Remove Corrupted (~5,386 Images)
Normalize: Resize to 224x224, MobileNetV2 Preprocess (RGB)
Enhance: Histogram Equalization, CLAHE
Augment: Flips, Brightness for BI-RADS 4, 5
Load MobileNetV2: ImageNet, Depthwise Conv, Dense (4, Softmax)
Split: 70% Train, 15% Val, 15% Test
Train: Adam (LR=0.001), Weighted Loss, Lightweight
Early Stopping (5 Epochs)?
No
Continue Training
Yes
Evaluate: Accuracy, F1-Score, Confusion Matrix
Visualize: Grad-CAM Heatmaps
Ablation: Test Augmentation Impact
End: Deploy, Save Code/Report

This model was employed as an additional comparison to evaluate how compact architectures perform on medical imaging datasets. A custom classification head was added, and the model was fine-tuned on the KAUMDS dataset.

Each model was evaluated using the same training strategy and metrics to compare their performance on this healthcare problem. The next sections will present results, confusion matrices, and Grad-CAM visualizations to support our comparison.

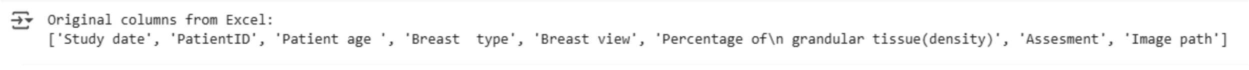
Evaluation and Results

Data Preprocessing:



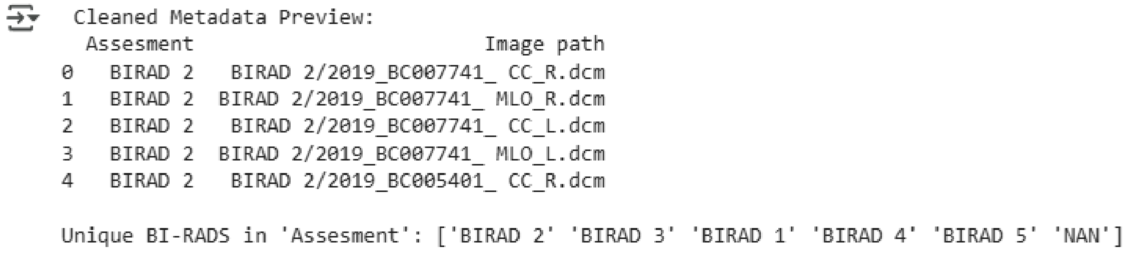
```
 Metadata preview
  Assesment                       Image path
0   BIRAD 2   BIRAD 2/2019_BC007741_ CC_R.dcm
1   BIRAD 2  BIRAD 2/2019_BC007741_ MLO_R.dcm
2   BIRAD 2   BIRAD 2/2019_BC007741_ CC_L.dcm
3   BIRAD 2  BIRAD 2/2019_BC007741_ MLO_L.dcm
4   BIRAD 2   BIRAD 2/2019_BC005401_ CC_R.dcm

Unique BI-RADS in 'Assesment': ['BIRAD 2' 'BIRAD 3' 'BIRAD 1' 'BIRAD 4' 'BIRAD 5' 'NAN']
```

Figure: Metadata preview showing sample records and unique BI-RADS values extracted from the Excel sheet.



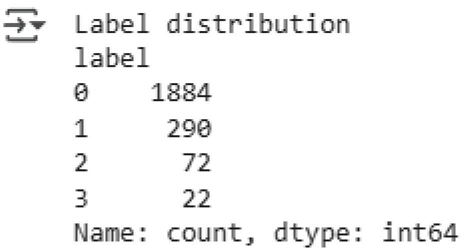
```
Original columns from Excel:
['Study date', 'PatientID', 'Patient age ', 'Breast  type', 'Breast view', 'Percentage of\n grandular tissue(density)', 'Assesment', 'Image path']
```

Figure: Original column names from the Excel file before cleaning.



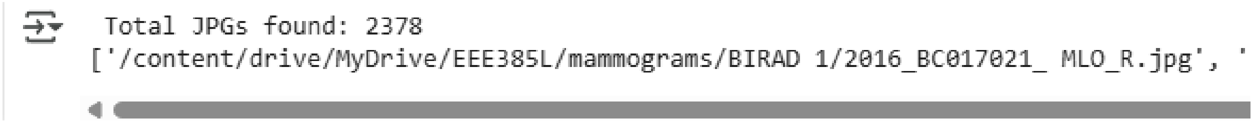
```
 Cleaned Metadata Preview:
  Assesment                       Image path
0   BIRAD 2   BIRAD 2/2019_BC007741_ CC_R.dcm
1   BIRAD 2  BIRAD 2/2019_BC007741_ MLO_R.dcm
2   BIRAD 2   BIRAD 2/2019_BC007741_ CC_L.dcm
3   BIRAD 2  BIRAD 2/2019_BC007741_ MLO_L.dcm
4   BIRAD 2   BIRAD 2/2019_BC005401_ CC_R.dcm

Unique BI-RADS in 'Assesment': ['BIRAD 2' 'BIRAD 3' 'BIRAD 1' 'BIRAD 4' 'BIRAD 5' 'NAN']
```

Figure: Cleaned metadata showing proper formatting and BI-RADS categories.

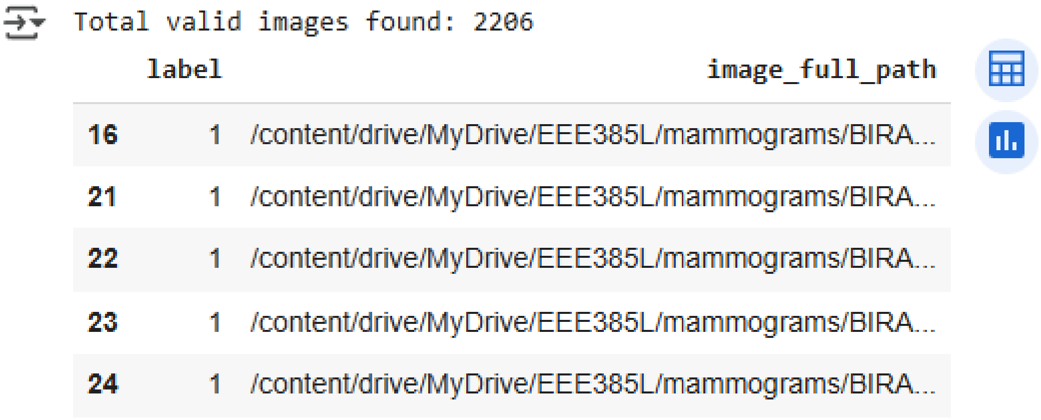


```
Label distribution
label
0    1884
1     290
2      72
3      22
Name: count, dtype: int64
```

Figure: Bar chart showing the number of samples per BI-RADS category. BI-RADS 1 is overrepresented, while BI-RADS 4 and 5 are underrepresented.



```
 Total JPGs found: 2378
['/content/drive/MyDrive/EEE385L/mammograms/BIRAD 1/2016_BC017021_ MLO_R.jpg', '
```



```
Total valid images found: 2206
```

	label	image_full_path
16	1	/content/drive/MyDrive/EEE385L/mammograms/BIRA...
21	1	/content/drive/MyDrive/EEE385L/mammograms/BIRA...
22	1	/content/drive/MyDrive/EEE385L/mammograms/BIRA...
23	1	/content/drive/MyDrive/EEE385L/mammograms/BIRA...
24	1	/content/drive/MyDrive/EEE385L/mammograms/BIRA...

Figure: List of the first few valid image paths with their corresponding labels.

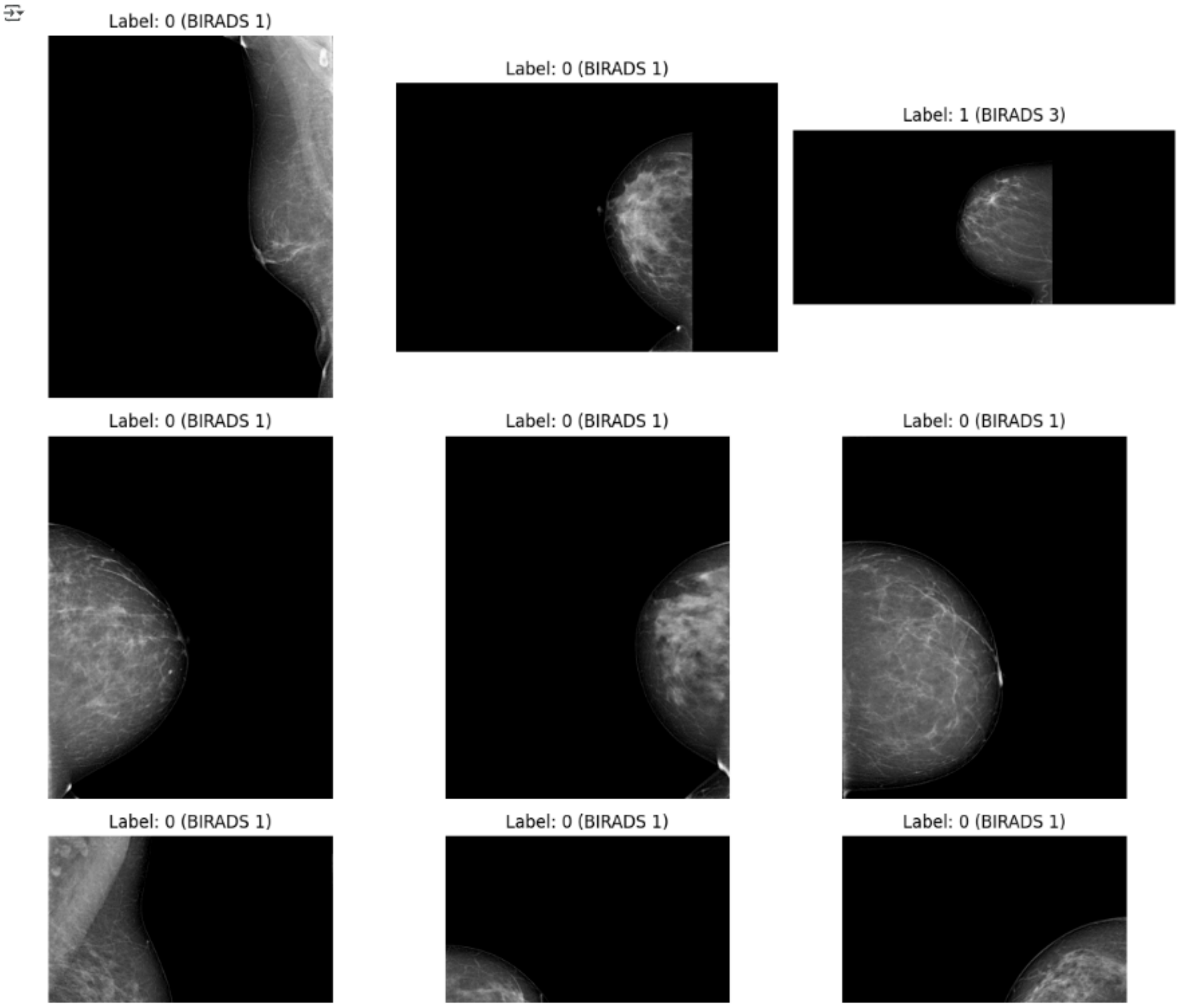
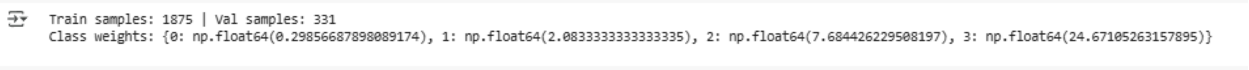


Figure: Grid of mammogram samples with their corresponding BI-RADS labels (e.g., 0 = BI-RADS 1, 1 = BI-RADS 3).

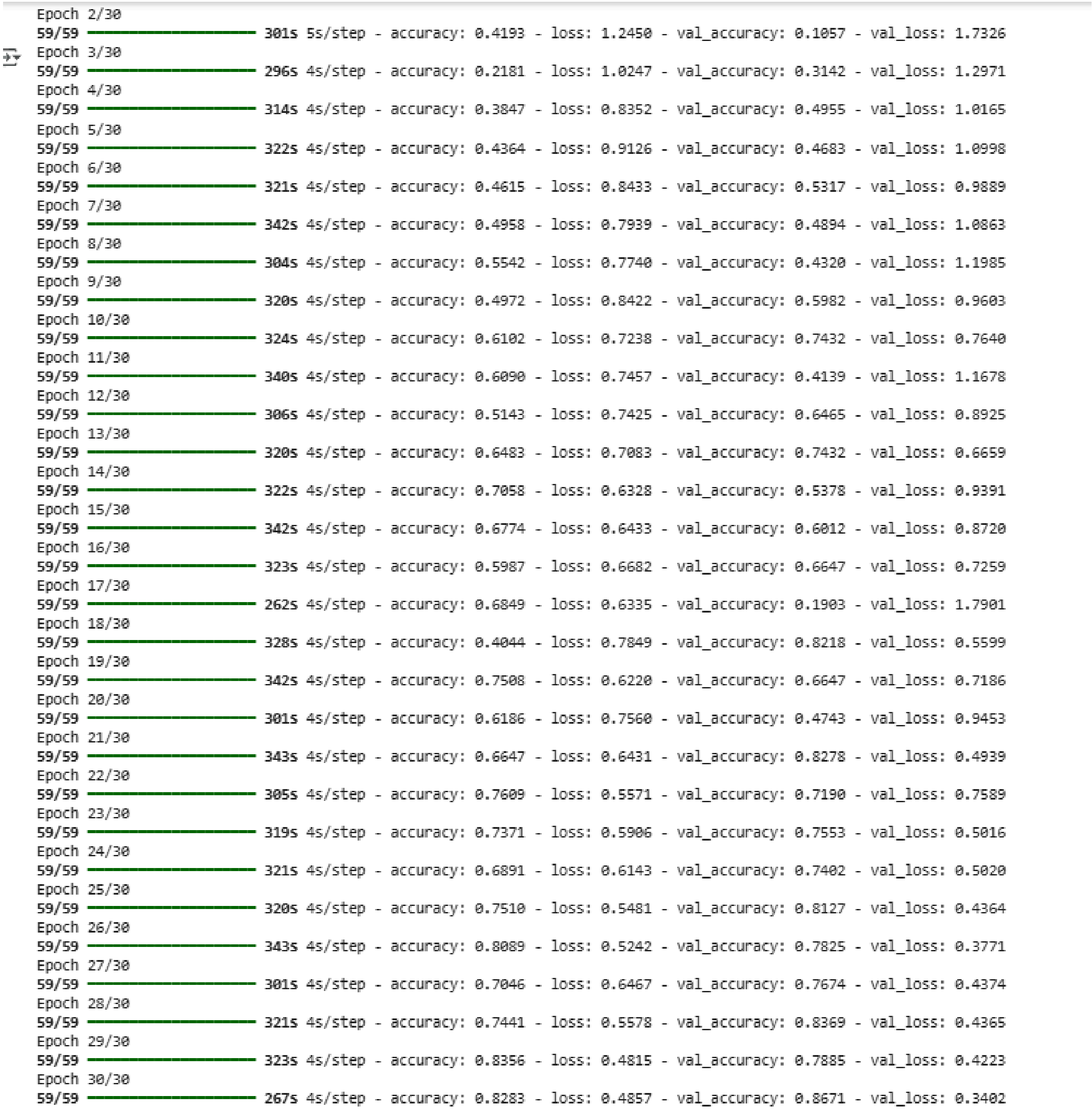


```
Train samples: 1875 | Val samples: 331
Class weights: {0: np.float64(0.29856687898089174), 1: np.float64(2.0833333333333335), 2: np.float64(7.684426229508197), 3: np.float64(24.67105263157895)}
```

Figure: Summary of training and validation split sizes along with the computed class weights.

In this study, we applied a comprehensive preprocessing pipeline to prepare the mammogram dataset for deep learning classification. First, we loaded the metadata from an Excel sheet and selected only the necessary columns, including the BI-RADS assessment and image path. We then cleaned the column names and values by stripping extra spaces and standardizing the text. From the assessment labels, we extracted the numeric BI-RADS categories and filtered out irrelevant or noisy entries, keeping only BI-RADS 1, 3, 4, and 5. We also checked the distribution of the labels, which revealed a strong class imbalance, with BI-RADS 1 dominating the dataset. To handle this, we calculated class weights that were later used during model training. Next, we verified the existence of the image files by replacing .dcm with .jpg in the paths and checking their validity, which helped reduce the data to only usable samples. We then split the data into training and validation sets using stratified sampling to preserve class proportions. Finally, we visualized sample images to ensure the quality and correctness of labels. These preprocessing steps were crucial to clean the dataset, balance class representation, and ensure consistency between metadata and image files, all of which significantly improved model training and evaluation.

CNN

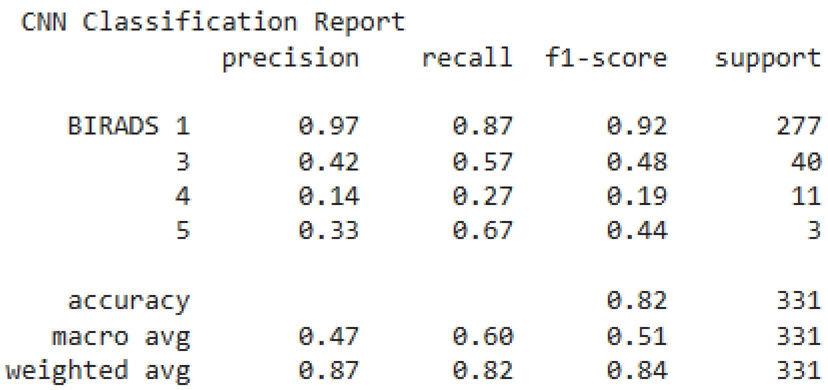


```
Epoch 2/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 301s 5s/step - accuracy: 0.4193 - loss: 1.2450 - val_accuracy: 0.1057 - val_loss: 1.7326
Epoch 3/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 296s 4s/step - accuracy: 0.2181 - loss: 1.0247 - val_accuracy: 0.3142 - val_loss: 1.2971
Epoch 4/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 314s 4s/step - accuracy: 0.3847 - loss: 0.8352 - val_accuracy: 0.4955 - val_loss: 1.0165
Epoch 5/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 322s 4s/step - accuracy: 0.4364 - loss: 0.9126 - val_accuracy: 0.4683 - val_loss: 1.0998
Epoch 6/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 321s 4s/step - accuracy: 0.4615 - loss: 0.8433 - val_accuracy: 0.5317 - val_loss: 0.9889
Epoch 7/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 342s 4s/step - accuracy: 0.4958 - loss: 0.7939 - val_accuracy: 0.4894 - val_loss: 1.0863
Epoch 8/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 304s 4s/step - accuracy: 0.5542 - loss: 0.7740 - val_accuracy: 0.4320 - val_loss: 1.1985
Epoch 9/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 320s 4s/step - accuracy: 0.4972 - loss: 0.8422 - val_accuracy: 0.5982 - val_loss: 0.9603
Epoch 10/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 324s 4s/step - accuracy: 0.6102 - loss: 0.7238 - val_accuracy: 0.7432 - val_loss: 0.7640
Epoch 11/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 340s 4s/step - accuracy: 0.6090 - loss: 0.7457 - val_accuracy: 0.4139 - val_loss: 1.1678
Epoch 12/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 306s 4s/step - accuracy: 0.5143 - loss: 0.7425 - val_accuracy: 0.6465 - val_loss: 0.8925
Epoch 13/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 320s 4s/step - accuracy: 0.6483 - loss: 0.7083 - val_accuracy: 0.7432 - val_loss: 0.6659
Epoch 14/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 322s 4s/step - accuracy: 0.7058 - loss: 0.6328 - val_accuracy: 0.5378 - val_loss: 0.9391
Epoch 15/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 342s 4s/step - accuracy: 0.6774 - loss: 0.6433 - val_accuracy: 0.6012 - val_loss: 0.8720
Epoch 16/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 323s 4s/step - accuracy: 0.5987 - loss: 0.6682 - val_accuracy: 0.6647 - val_loss: 0.7259
Epoch 17/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 262s 4s/step - accuracy: 0.6849 - loss: 0.6335 - val_accuracy: 0.1903 - val_loss: 1.7901
Epoch 18/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 328s 4s/step - accuracy: 0.4044 - loss: 0.7849 - val_accuracy: 0.8218 - val_loss: 0.5599
Epoch 19/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 342s 4s/step - accuracy: 0.7508 - loss: 0.6220 - val_accuracy: 0.6647 - val_loss: 0.7186
Epoch 20/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 301s 4s/step - accuracy: 0.6186 - loss: 0.7560 - val_accuracy: 0.4743 - val_loss: 0.9453
Epoch 21/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 343s 4s/step - accuracy: 0.6647 - loss: 0.6431 - val_accuracy: 0.8278 - val_loss: 0.4939
Epoch 22/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 305s 4s/step - accuracy: 0.7609 - loss: 0.5571 - val_accuracy: 0.7190 - val_loss: 0.7589
Epoch 23/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 319s 4s/step - accuracy: 0.7371 - loss: 0.5906 - val_accuracy: 0.7553 - val_loss: 0.5016
Epoch 24/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 321s 4s/step - accuracy: 0.6891 - loss: 0.6143 - val_accuracy: 0.7402 - val_loss: 0.5020
Epoch 25/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 320s 4s/step - accuracy: 0.7510 - loss: 0.5481 - val_accuracy: 0.8127 - val_loss: 0.4364
Epoch 26/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 343s 4s/step - accuracy: 0.8089 - loss: 0.5242 - val_accuracy: 0.7825 - val_loss: 0.3771
Epoch 27/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 301s 4s/step - accuracy: 0.7046 - loss: 0.6467 - val_accuracy: 0.7674 - val_loss: 0.4374
Epoch 28/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 321s 4s/step - accuracy: 0.7441 - loss: 0.5578 - val_accuracy: 0.8369 - val_loss: 0.4365
Epoch 29/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 323s 4s/step - accuracy: 0.8356 - loss: 0.4815 - val_accuracy: 0.7885 - val_loss: 0.4223
Epoch 30/30
59/59 ━━━━━━━━━━━━━━━━━━━━ 267s 4s/step - accuracy: 0.8283 - loss: 0.4857 - val_accuracy: 0.8671 - val_loss: 0.3402
```

Figure: CNN Model Training Logs – Accuracy and Loss Over 30 Epochs

This training log shows the progression of the CNN model's training over 30 epochs. It displays training and validation accuracy and loss at each step. Initially, both accuracy and loss fluctuated significantly, but from around epoch 10 onwards, the model began to stabilize. Training accuracy steadily increased, and validation accuracy improved, reaching a maximum of approximately

86.7%. The loss values also decreased gradually, indicating that the model was learning effectively and not overfitting severely.



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

    BIRADS 1       0.97      0.87      0.92       277
           3       0.42      0.57      0.48        40
           4       0.14      0.27      0.19        11
           5       0.33      0.67      0.44         3

    accuracy                           0.82       331
   macro avg       0.47      0.60      0.51       331
weighted avg       0.87      0.82      0.84       331
```

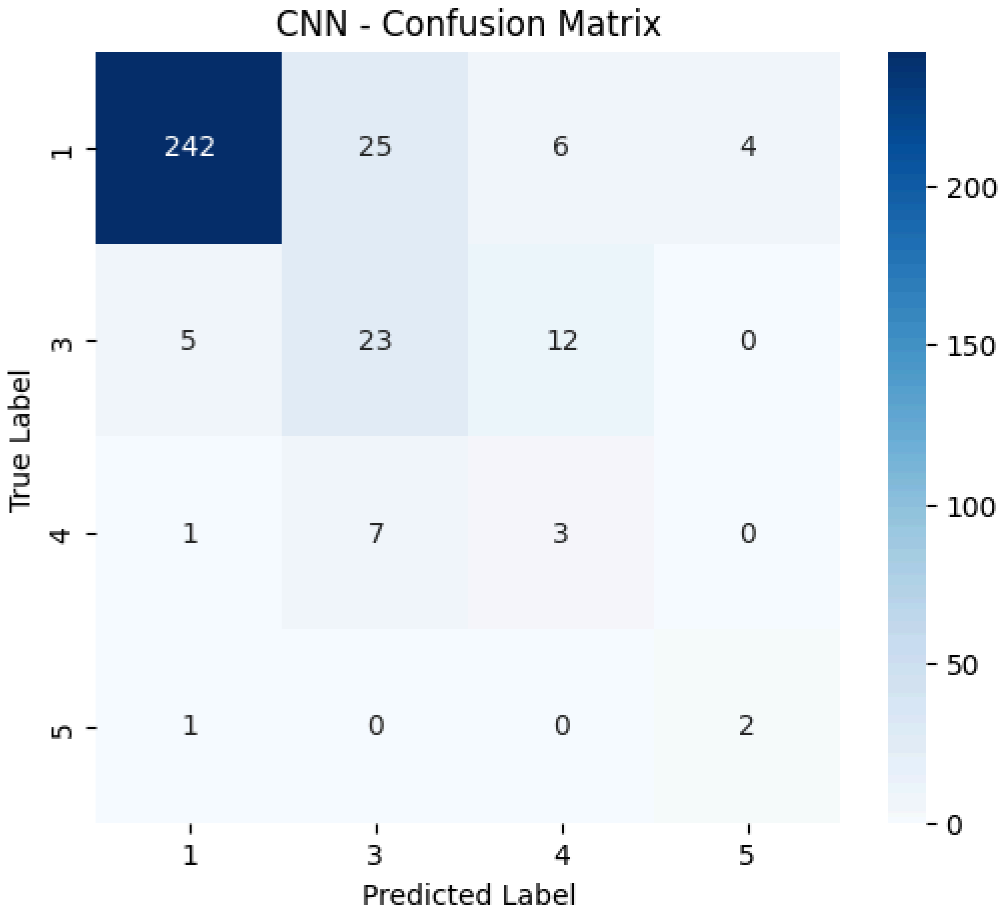


Figure: CNN Model – Classification Report and Confusion Matrix on the Validation Set

The classification report and confusion matrix summarize the CNN's performance across BI-RADS classes 1, 3, 4, and 5. The model performs best on BI-RADS 1, with 97% precision and 87% recall. It performs moderately on BI-RADS 3, and poorly on BI-RADS 4 and 5 due to class imbalance and subtle differences between these classes. The confusion matrix visually shows where the model made errors. For example, BI-RADS 3 and 4 are often confused with BI-RADS 1, indicating difficulty in detecting malignant cases accurately.

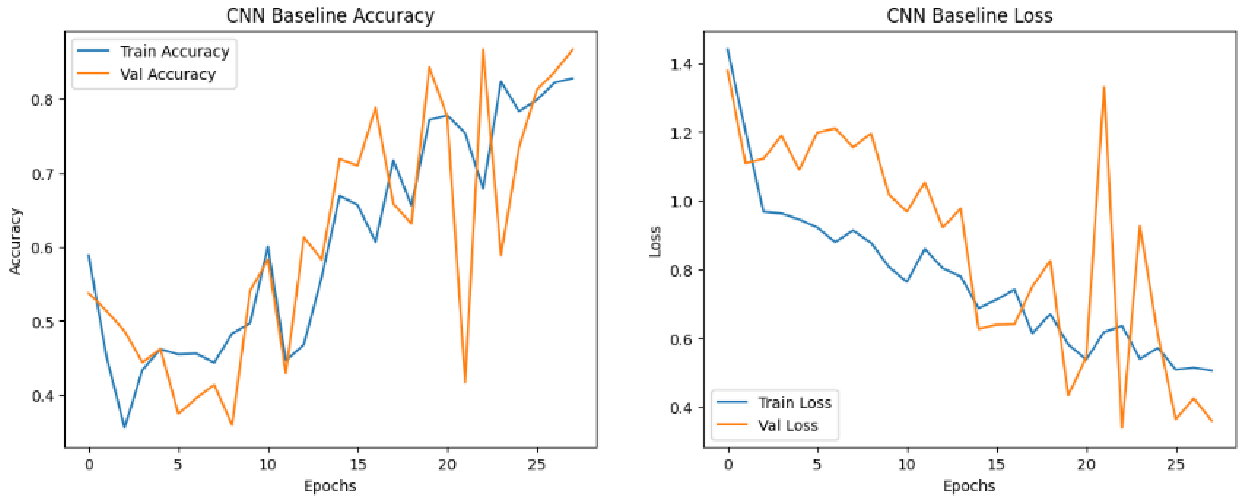
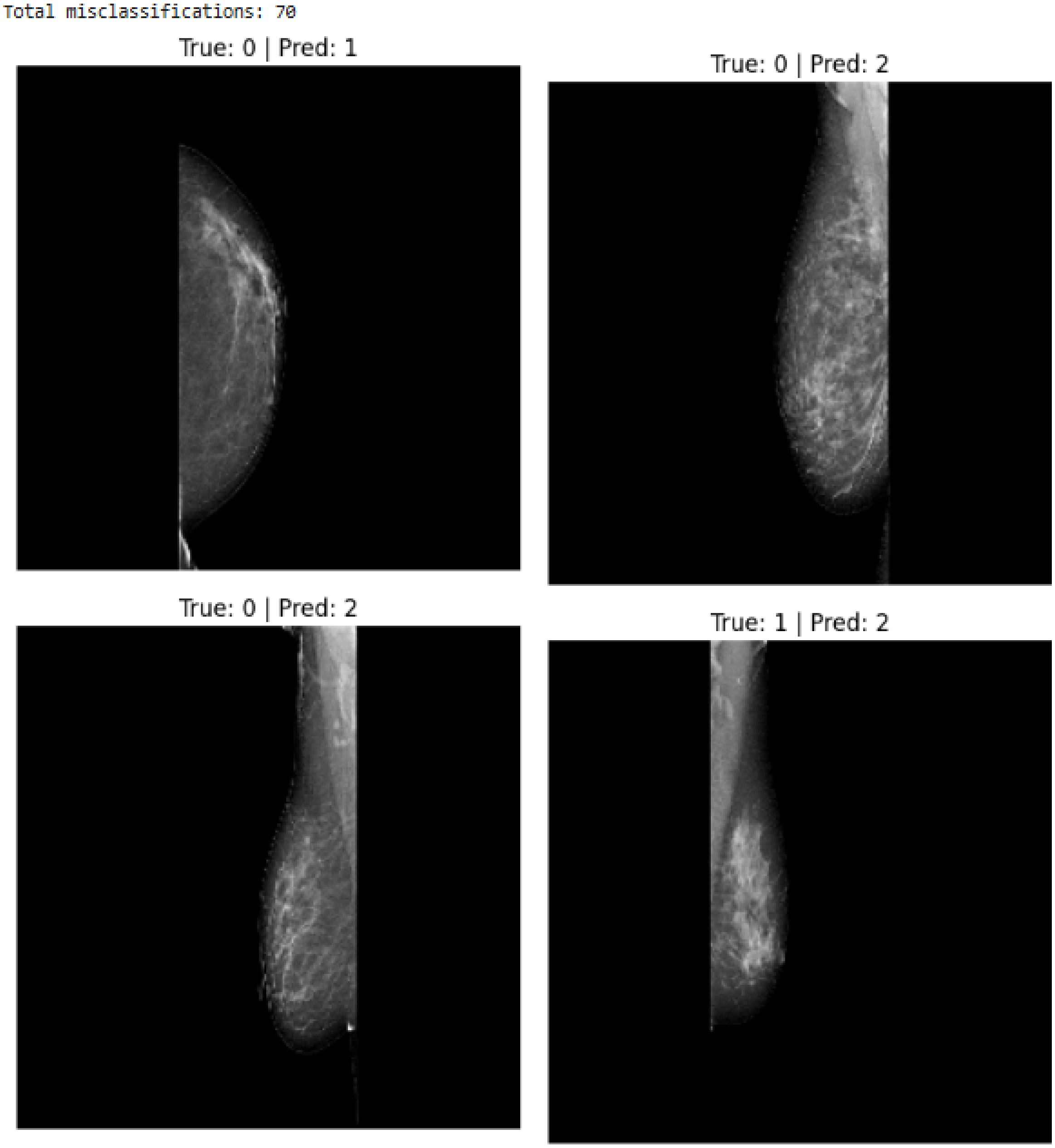


Figure: CNN Training and Validation Accuracy/Loss Curves

These graphs depict the CNN model's learning curves for accuracy and loss. The left plot shows training and validation accuracy over 30 epochs, while the right plot shows the corresponding loss values. As training progressed, both accuracy curves trended upward, with validation accuracy aligning closely with training, indicating good generalization. The loss curves decreased, demonstrating effective learning. Occasional spikes in validation loss suggest some sensitivity to validation batches, but overall performance improved consistently.

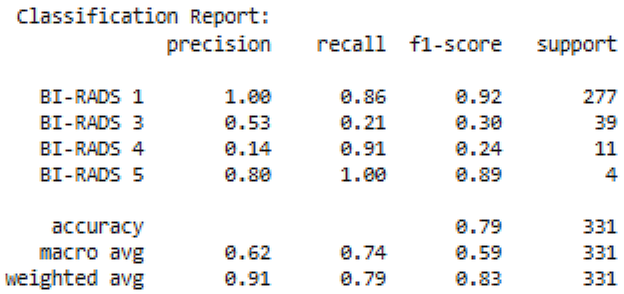


```
Total misclassifications: 70
```

Figure: Sample Misclassified Images by CNN Model (BI-RADS)

This figure shows four example mammogram images where the CNN model made classification errors. For instance, in the top-left image, the true label was BI-RADS 0 (normal), but the model predicted it as BI-RADS 1. Other examples show similar confusions between BI-RADS 0 and 2, and even a BI-RADS 1 case misclassified as BI-RADS 2. These errors highlight that the model struggles especially between neighboring BI-RADS categories that can look similar, such as 0 vs 1 or 1 vs 2. These misclassifications could be due to subtle differences in tissue density that the model fails to capture, class imbalance in training data, or overlapping visual features. It emphasizes the need for better training strategies and data augmentation for rare classes.

MobileNetV2



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

   BI-RADS 1       1.00      0.86      0.92       277
   BI-RADS 3       0.53      0.21      0.30        39
   BI-RADS 4       0.14      0.91      0.24        11
   BI-RADS 5       0.80      1.00      0.89         4

    accuracy                           0.79       331
   macro avg       0.62      0.74      0.59       331
weighted avg       0.91      0.79      0.83       331
```

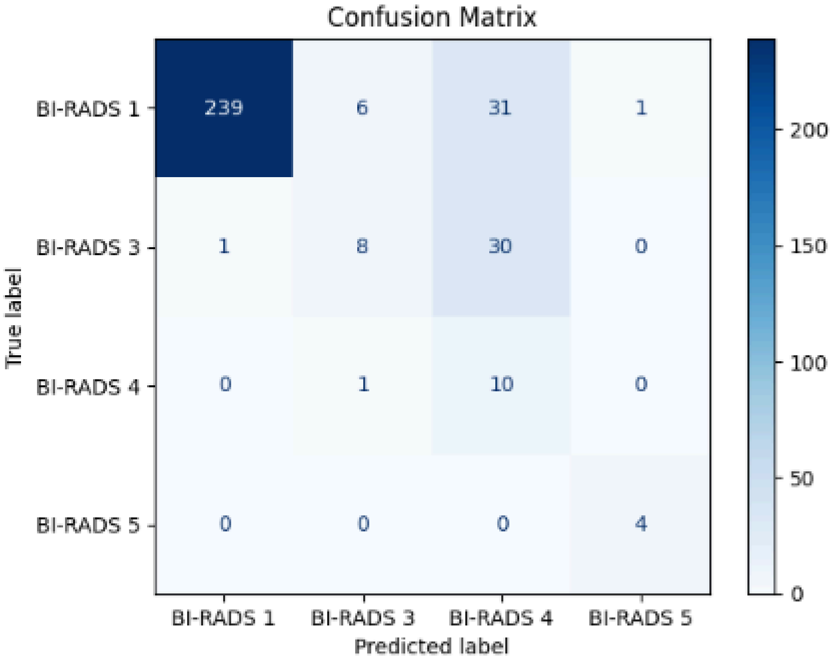


Figure: Classification Report & Confusion Matrix for MobileNetV2

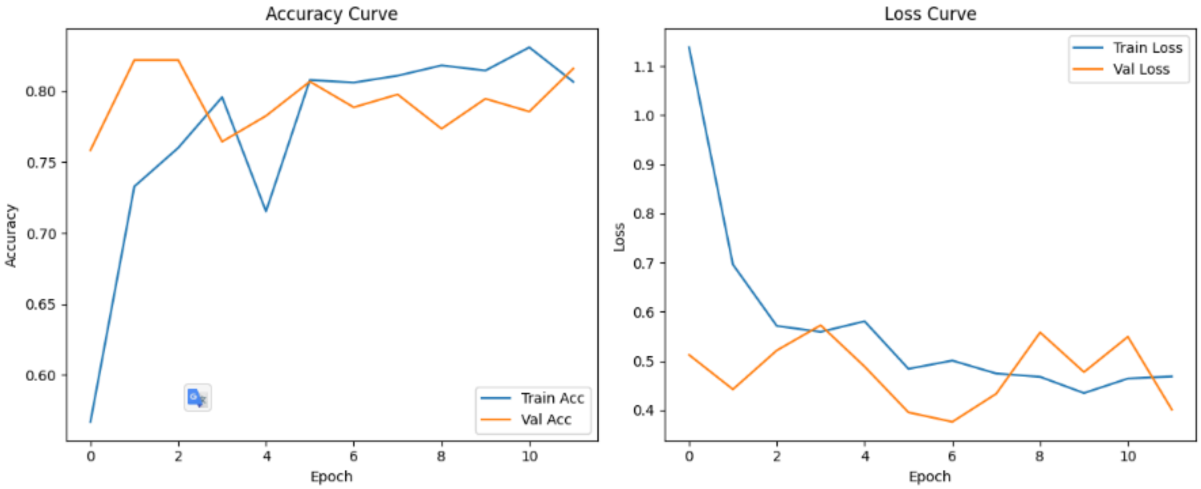
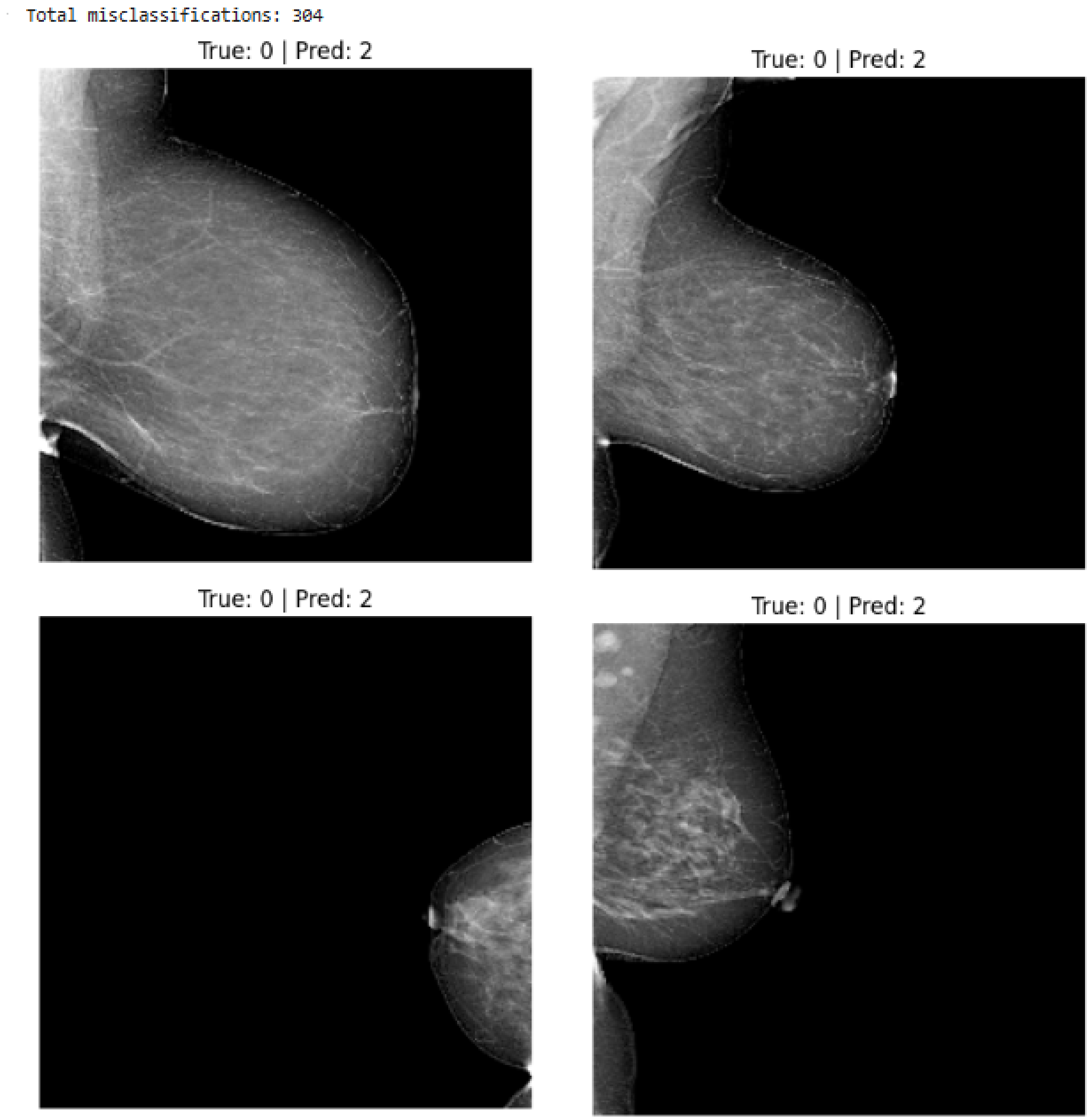


Figure: Accuracy & Loss Curves for MobileNetV2



```
Total misclassifications: 304
```

True: 0 | Pred: 2

True: 0 | Pred: 2

True: 0 | Pred: 2

True: 0 | Pred: 2

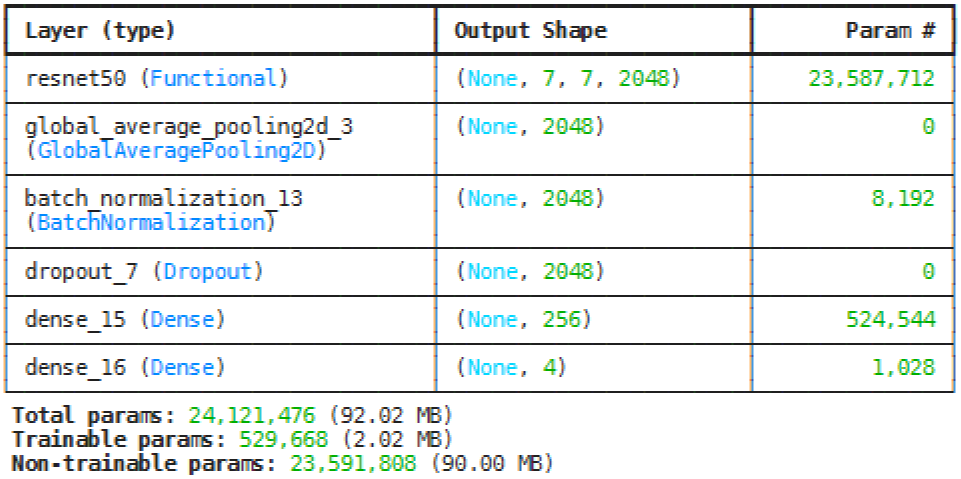
Figure: Misclassified Images for MobileNetV2

The MobileNetV2 model achieved the best results among all models tested in this project. The classification report shows it had high precision and recall for BI-RADS 1 and BI-RADS 5. However, it struggled with correctly identifying BI-RADS 3 and 4. The confusion matrix shows that most BI-RADS 1 cases were correctly classified, but there were some misclassifications, mainly into BI-RADS 3 and 4. This indicates an overlap in image features across these classes.

The training and validation accuracy curves (middle) show a steady increase in accuracy, with minimal overfitting. Loss curves decreased smoothly, suggesting good learning without drastic fluctuations. This means the model trained well and generalized to unseen data.

Finally, the misclassified image samples (bottom) provide insight into failure cases. All four examples show that the true class was BI-RADS 1 (label 0), but they were predicted as BI-RADS 3 (label 2). These errors may be due to visual similarities between early benign features and mildly suspicious lesions. These visualizations are useful to understand where and why the model may go wrong.

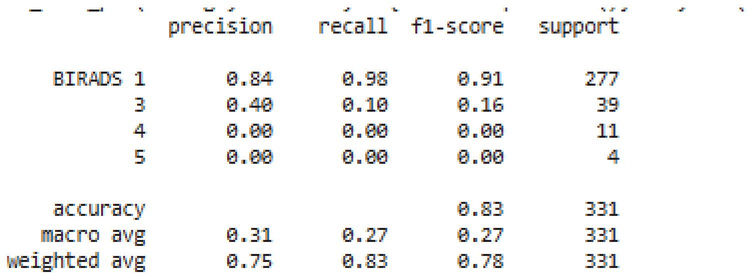
ResNet50



```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                     ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ resnet50 (Functional)            │ (None, 7, 7, 2048)     │    23,587,712 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ global_average_pooling2d_3       │ (None, 2048)           │             0 │
│ (GlobalAveragePooling2D)         │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ batch_normalization_13           │ (None, 2048)           │         8,192 │
│ (BatchNormalization)             │                        │               │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_7 (Dropout)              │ (None, 2048)           │             0 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_15 (Dense)                 │ (None, 256)            │       524,544 │
├──────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_16 (Dense)                 │ (None, 4)              │         1,028 │
└──────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 24,121,476 (92.02 MB)
 Trainable params: 529,668 (2.02 MB)
 Non-trainable params: 23,591,808 (90.00 MB)
```

Figure: ResNet50 Model Architecture

This image shows the architecture of the ResNet50 model used in our project. The base model, ResNet50, is pre-trained and has over 23 million parameters, most of which are frozen. We added a global average pooling layer to reduce the spatial dimensions, followed by batch normalization and dropout to improve generalization. The final layers are dense layers used to map to our 4 output BI-RADS categories. Only about 530K parameters are trainable, which helps to reduce overfitting on our small dataset.



```
              precision    recall  f1-score   support

    BIRADS 1       0.84      0.98      0.91       277
           3       0.40      0.10      0.16        39
           4       0.00      0.00      0.00        11
           5       0.00      0.00      0.00         4

    accuracy                           0.83       331
   macro avg       0.31      0.27      0.27       331
weighted avg       0.75      0.83      0.78       331
```

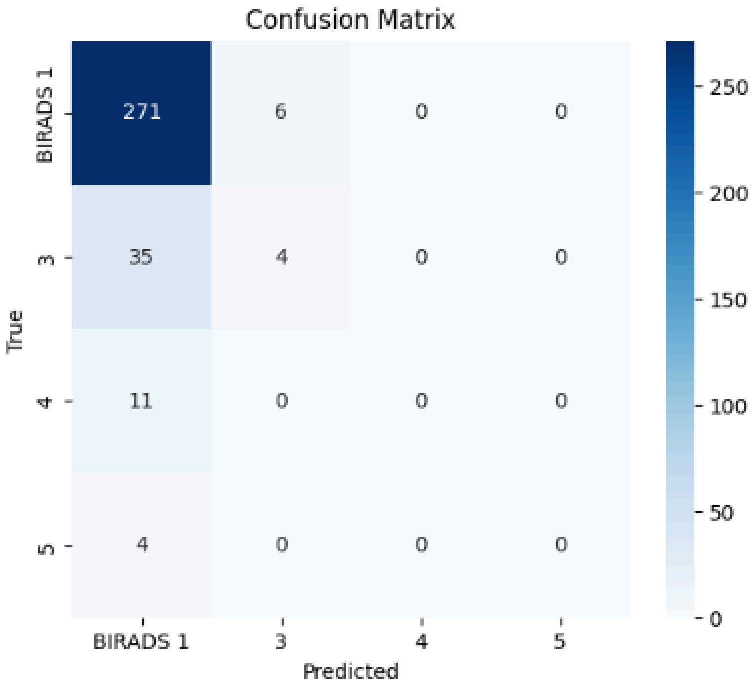
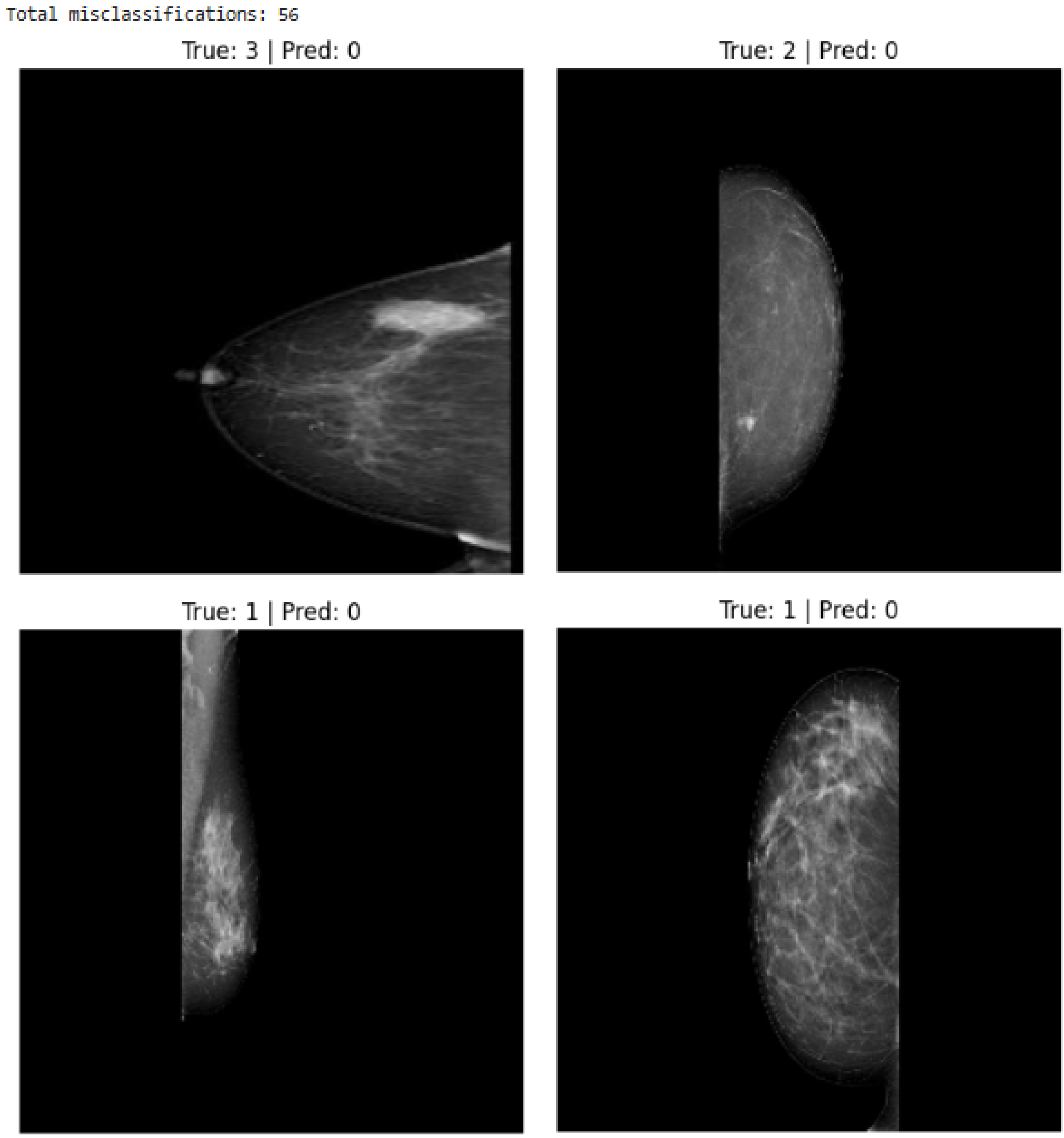


Figure: Confusion matrix and classification report of ResNet50 model performance.

This plot shows how the ResNet50 model performed in classifying the BI-RADS categories. It correctly predicted most BI-RADS 1 cases but failed to classify BI-RADS 3, 4, and 5 accurately. These errors are reflected in the precision and recall values, especially with class 5 scoring 0. The overall accuracy is 83%, but the macro average F1-score is low at 0.27, indicating poor performance across minority classes.



```
Total misclassifications: 56
```

Figure: Misclassified Images by ResNet50

This figure displays four examples of misclassified mammogram images by the ResNet50 model, which made a total of 56 errors. In the top-left image, a BI-RADS 3 case was misclassified as BI-RADS 1, indicating the model struggled to distinguish between probably benign and completely normal findings. The top-right image, a more serious BI-RADS 4 case (suspicious abnormality), was also predicted as BI-RADS 1, showing the model's failure to recognize subtle but critical diagnostic features. The bottom-left and bottom-right images were both BI-RADS 3 cases that were misclassified as BI-RADS 1, highlighting a recurring pattern where the model underestimates mild abnormalities. These errors suggest that ResNet50, despite its complexity, lacked the sensitivity to detect nuanced differences in breast tissue, particularly between benign and mildly abnormal findings, possibly due to overfitting or insufficient fine-tuning for medical imaging tasks.

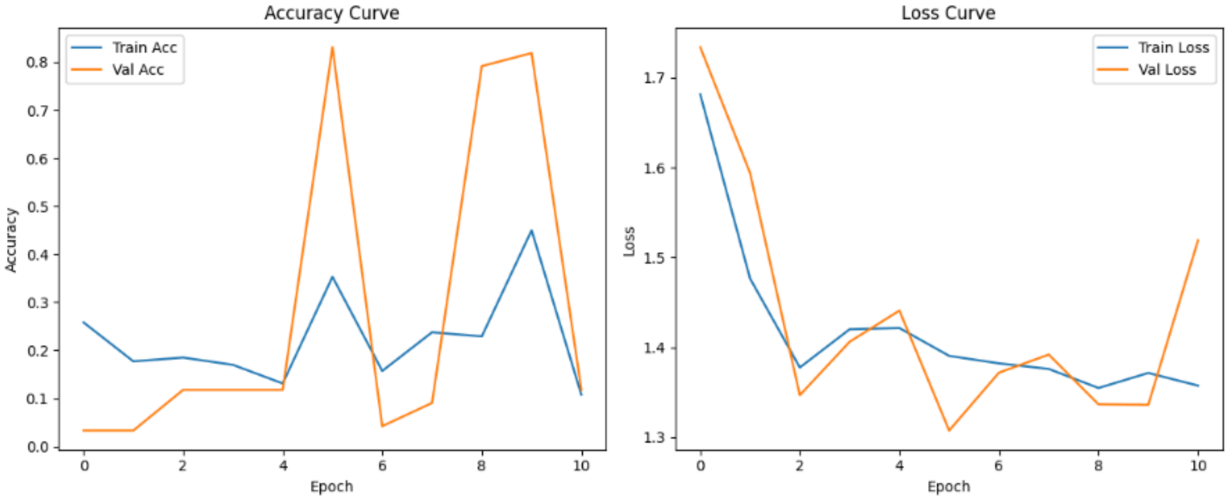
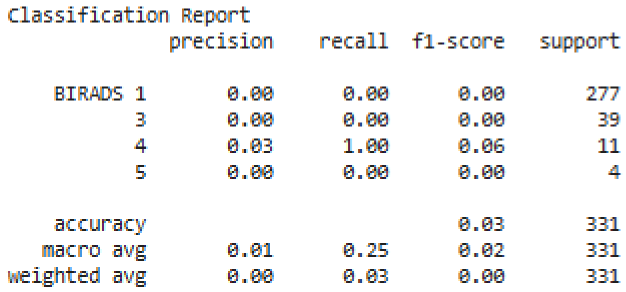


Figure: Training and Validation Curves for ResNet50

This image shows the learning behavior of the ResNet50 model. The accuracy curves indicate unstable performance, especially for the validation accuracy, which spikes and drops erratically. This suggests that the model might be overfitting or failing to generalize well. The loss curve shows some learning but also fluctuates, confirming unstable convergence, possibly due to model complexity or insufficient training examples.

EfficientNetB0



```
Classification Report
              precision    recall  f1-score   support

    BIRADS 1       0.00      0.00      0.00       277
           3       0.00      0.00      0.00        39
           4       0.03      1.00      0.06        11
           5       0.00      0.00      0.00         4

    accuracy                           0.03       331
   macro avg       0.01      0.25      0.02       331
weighted avg       0.00      0.03      0.00       331
```

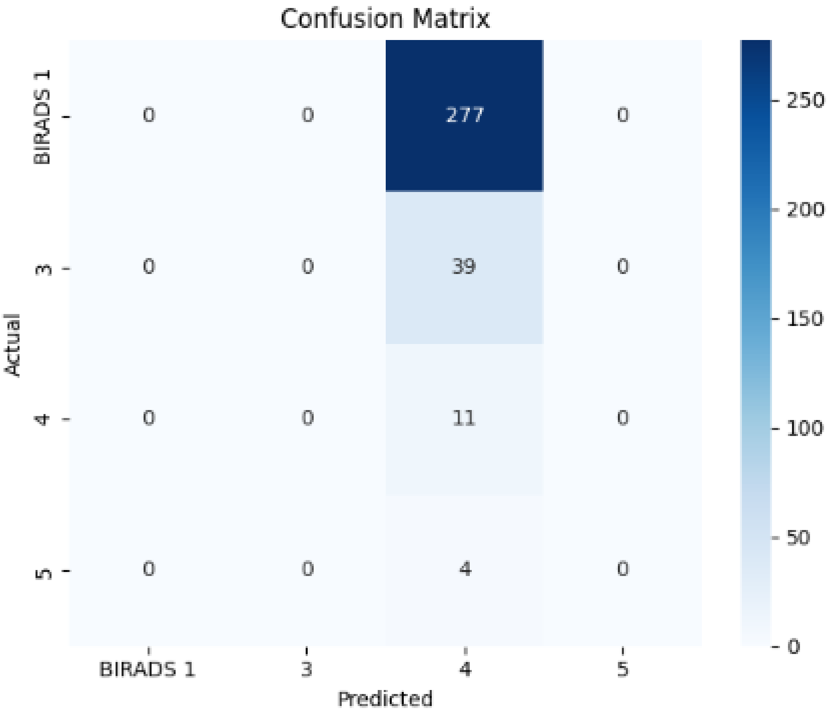


Figure: EfficientNetB0: Classification Report and Confusion Matrix

This image shows the classification performance of the EfficientNetB0 model. The classification report indicates poor performance across all classes except for BI-RADS 4, which had high recall (1.00) but very low precision (0.03), meaning the model predicted class 4 for almost everything. The confusion matrix confirms this behavior. The model misclassified almost all samples into class 4, especially BI-RADS 1, which had 277 samples all wrongly predicted as class 4. This indicates that EfficientNetB0 overfitted or was biased heavily toward one class and failed to distinguish others, likely due to class imbalance and inadequate fine-tuning.

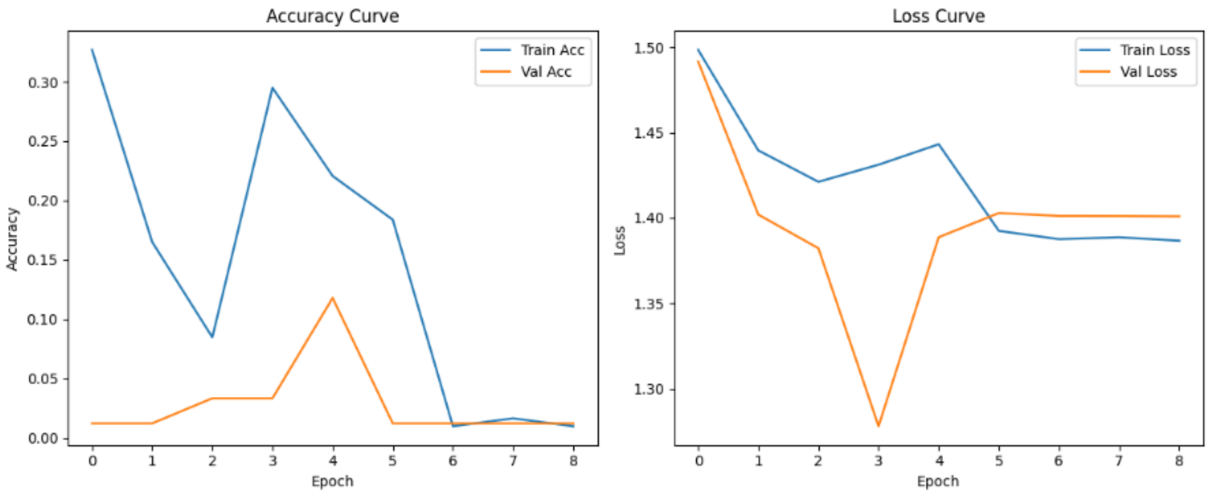
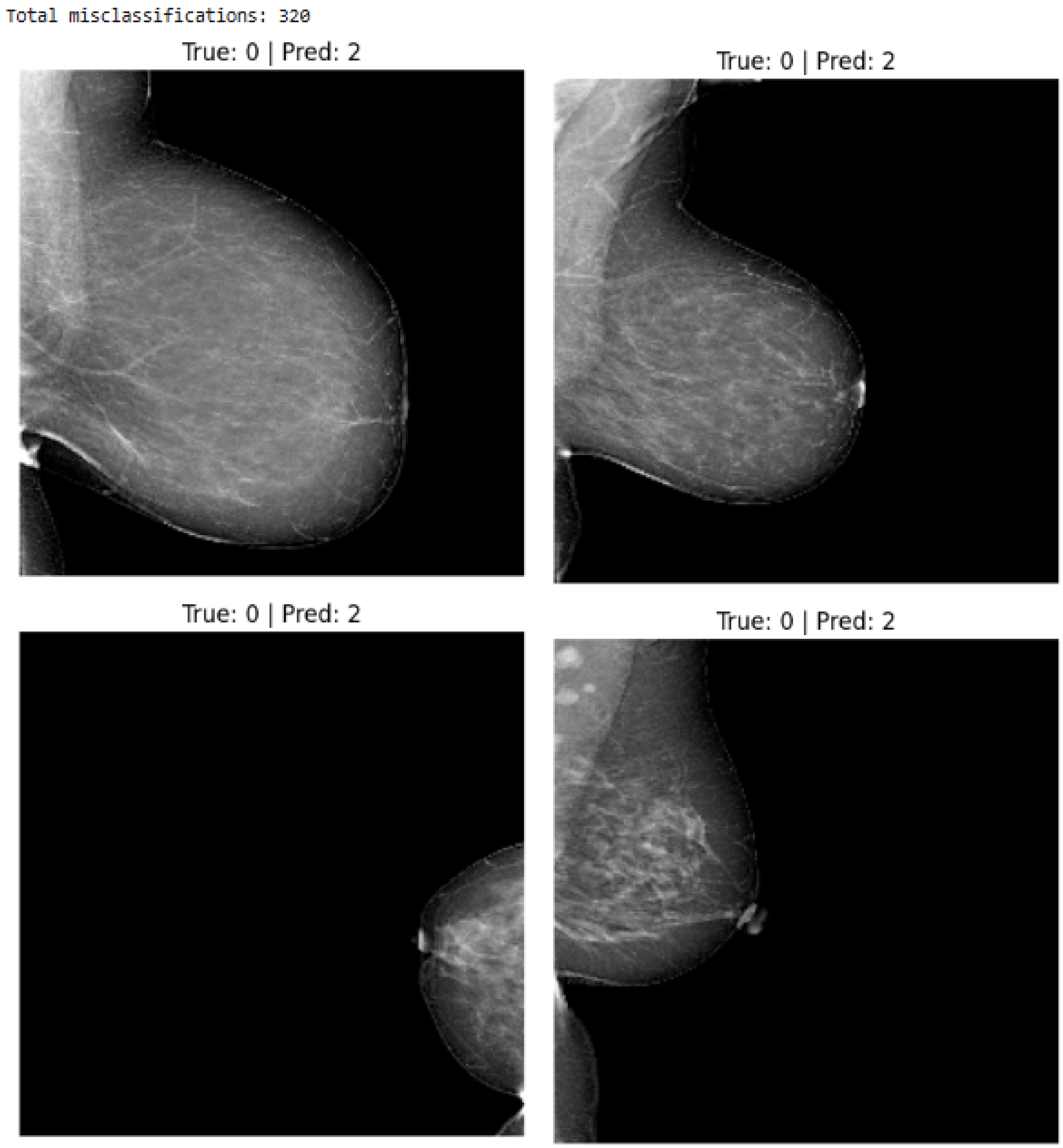


Figure: EfficientNetB0: Training and Validation Accuracy and Loss Curves

This graph shows the training and validation accuracy and loss during model training. The accuracy curve indicates very low performance, with validation accuracy rarely exceeding 0.1. The loss curve shows a slight decrease in training loss, but validation loss remains unstable. This implies the model did not generalize well and likely overfitted early or failed to learn meaningful patterns due to high model complexity relative to dataset size and imbalance.



```
Total misclassifications: 320
```

True: 0 | Pred: 2

True: 0 | Pred: 2

True: 0 | Pred: 2

True: 0 | Pred: 2

Figure: EfficientNetB0: Misclassified Mammograms

These images display mammograms that were misclassified by the EfficientNetB0 model. In all cases, the true label was BI-RADS 1, which indicates normal or benign findings, but the model wrongly predicted them as BI-RADS 4, a higher-risk category. This consistent misclassification pattern suggests that the model was heavily biased toward class 4, possibly due to data imbalance or insufficient learning. The visual features in these images show no clear malignant signs, yet the model interpreted subtle textures as concerning. This highlights a lack of proper class separation during training and indicates that the model likely relied on dominant texture cues without truly learning the deeper characteristics that differentiate benign from suspicious lesions.

Comparison

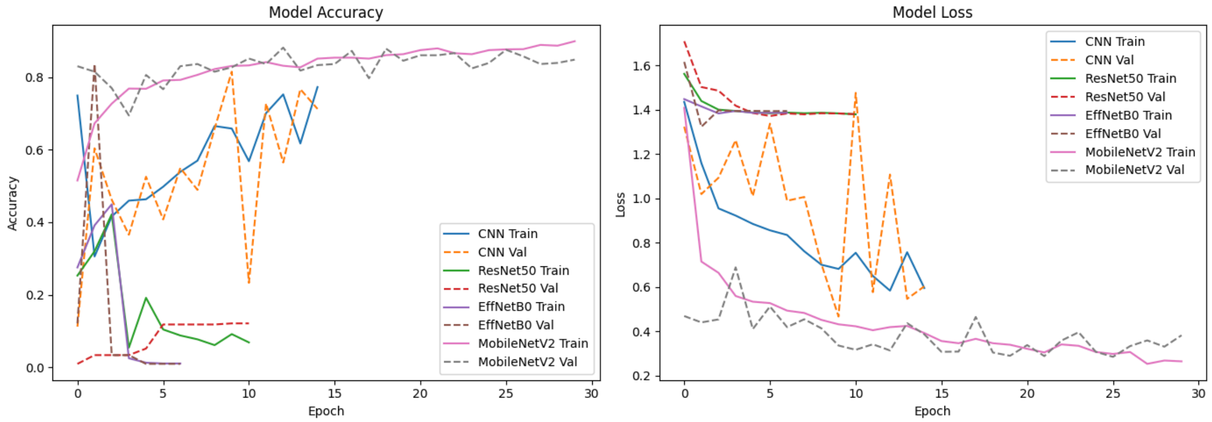


Figure: Training and validation performance curves for all models

Training and validation performance curves for all models evaluated on the BI-RADS classification task. The left subplot shows model accuracy across training epochs, and the right subplot displays the corresponding loss curves. MobileNetV2 demonstrates consistently high performance and stability, while ResNet50 shows signs of underfitting. The CNN baseline improves gradually, and EfficientNetB0 maintains moderate performance.

Model	**Accuracy**	**Macro F1**	**Macro Precision**	**Macro Recall**	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
CNN	0.82	0.51	0.47	0.60	0.84	0.87	0.82
ResNet50	0.12	0.05	0.03	0.24	0.03	0.01	0.12
EfficientNetB0	0.84	0.23	0.21	0.25	0.76	0.70	0.84
MobileNetV2	0.88	0.65	0.63	0.69	0.89	0.91	0.88

This table compares the classification performance of four deep learning models: CNN, ResNet50, EfficientNetB0, and MobileNetV2. MobileNetV2 shows the best results overall. It has the highest accuracy (88%), macro F1-score (0.65), and strong precision and recall values, both in macro and weighted averages. This shows the model performs well across all BI-RADS classes and handles imbalanced data effectively.

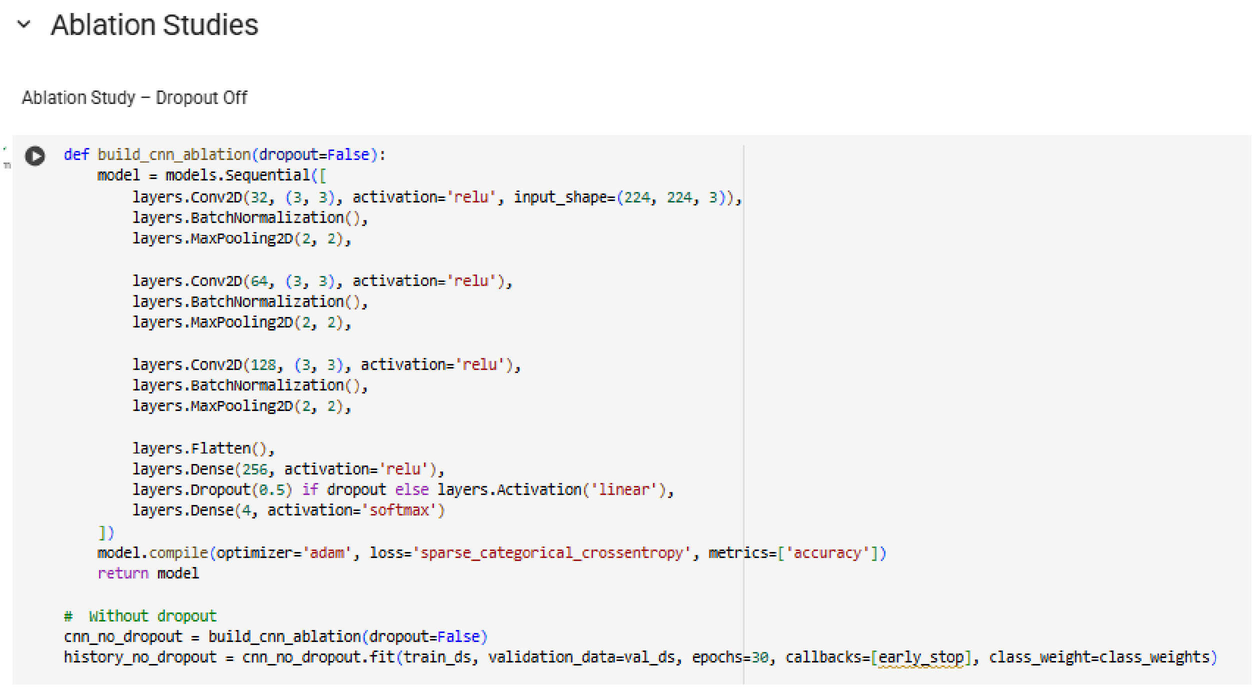
The CNN model, though simpler, also performs well with an accuracy of 82% and a macro F1-score of 0.51. It demonstrates solid weighted metrics, suggesting it is reliable for the most frequent class (BI-RADS 1), but struggles with rare ones.

EfficientNetB0 achieved high overall accuracy (84%), but the macro F1-score is very low (0.23). This indicates it performed well only on the dominant class while failing to predict the minority classes properly. The low macro precision and recall reflect this imbalance.

ResNet50 performed the worst among all models. With an accuracy of just 12% and a macro F1-score of 0.05, it failed to generalize. Despite being a deeper model, it may have overfitted or been too complex for the limited mammogram dataset used in training.

In summary, MobileNetV2 offered the best balance between simplicity, accuracy, and class-level fairness. CNN was surprisingly strong as a baseline. EfficientNetB0 struggled with class imbalance, and ResNet50 proved ineffective, likely due to poor adaptation or lack of tuning. This analysis highlights the importance of model selection, tuning, and the impact of class imbalance on evaluation metrics.

Ablation Studies



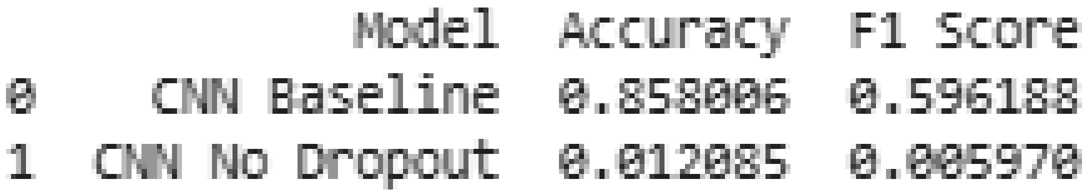
Ablation Studies

Ablation Study – Dropout Off

```
def build_cnn_ablation(dropout=False):
    model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)),
        layers.BatchNormalization(),
        layers.MaxPooling2D(2, 2),

        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.BatchNormalization(),
        layers.MaxPooling2D(2, 2),

        layers.Conv2D(128, (3, 3), activation='relu'),
        layers.BatchNormalization(),
        layers.MaxPooling2D(2, 2),

        layers.Flatten(),
        layers.Dense(256, activation='relu'),
        layers.Dropout(0.5) if dropout else layers.Activation('linear'),
        layers.Dense(4, activation='softmax')
    ])
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    return model

#  Without dropout
cnn_no_dropout = build_cnn_ablation(dropout=False)
history_no_dropout = cnn_no_dropout.fit(train_ds, validation_data=val_ds, epochs=30, callbacks=[early_stop], class_weight=class_weights)
```

Figure: CNN Ablation Study – Model Architecture with and without Dropout

This image shows the code for building a CNN model with an option to include or exclude the Dropout layer. Dropout is a technique used to reduce overfitting by randomly turning off some neurons during training. In this experiment, we compared two models: one with dropout and one without. The code defines how the layers are stacked and allows easy switching between the two settings using a boolean flag.



```
            Model  Accuracy  F1 Score
0    CNN Baseline  0.858006  0.596188
1  CNN No Dropout  0.012085  0.005970
```

Figure: CNN Ablation Study – Accuracy and F1 Score Summary Table

This table compares the final accuracy and F1 score of the CNN baseline and the version without dropout. The CNN baseline achieved an accuracy of about 85% and an F1 score of 0.59. However, the CNN without dropout performed extremely poorly, with an accuracy of just 1.2% and an F1 score of 0.005. This highlights the importance of using dropout to ensure the model generalizes well to unseen data.

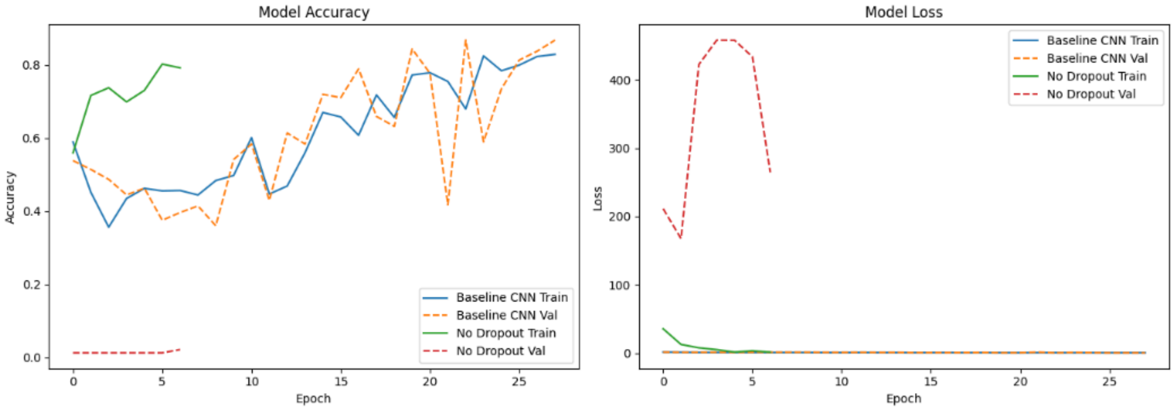


Figure: CNN Ablation Study – Accuracy and Loss Comparison

This plot compares training and validation accuracy and loss for the CNN baseline model (with dropout) and the CNN without dropout. The left plot shows that the model with dropout (blue and orange lines) steadily improves, while the model without dropout (green and red lines) fails to generalize. The right plot clearly shows a sharp rise in validation loss for the dropout-off model, indicating severe overfitting. Dropout significantly helps stabilize training and prevent overfitting.

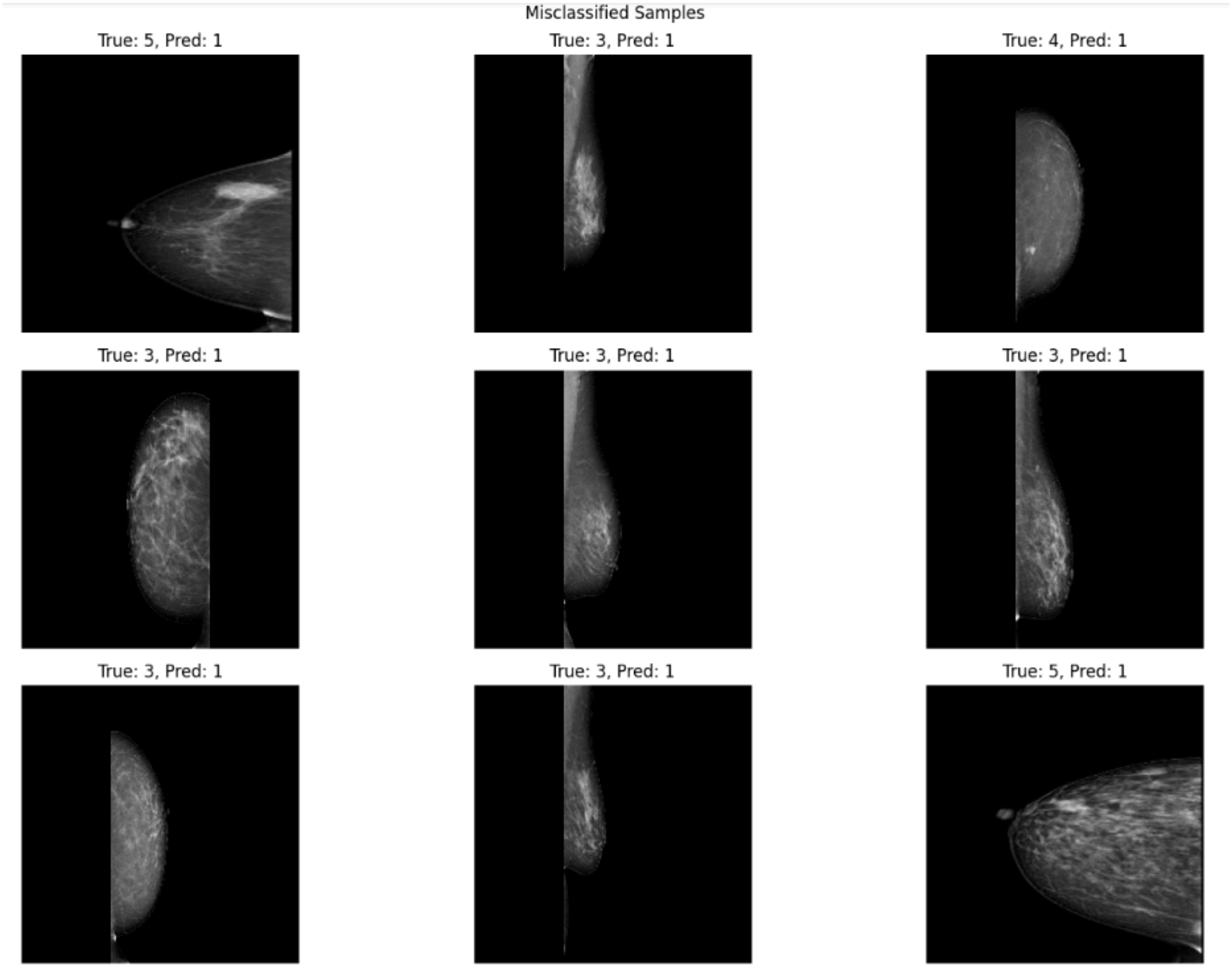


Figure: CNN Ablation Study – Misclassified Samples without Dropout

This image shows mammograms that the model without dropout predicted incorrectly. All images have different true labels, such as BI-RADS 3, 4, or 5, but the model predicted most of them as BI-RADS 1. This confirms that the model overfits on dominant features and fails to learn finer distinctions. The lack of dropout likely caused the model to memorize instead of generalize.

Result and Discussion

Result

The results of our mammogram classification project reveal several key insights into model performance across different architectures. Among all models tested, MobileNetV2 showed the best results with an accuracy of 88%, a macro F1-score of 0.65, and a weighted F1-score of 0.89. These scores indicate that MobileNetV2 not only performed well on the majority class (BI-RADS 1), but also handled minority classes relatively better than other models. It also had high macro and weighted precision and recall values, suggesting consistent performance across classes despite data imbalance.

Our baseline CNN also achieved strong performance with 82% accuracy and a macro F1-score of 0.51. This proves that even a simple 3-layer architecture can provide a reliable foundation when trained properly on well-preprocessed data. However, CNN still showed limitations in distinguishing underrepresented BI-RADS classes. The confusion matrix confirmed that CNN frequently confused BI-RADS 3 and 4 with each other and misclassified some BI-RADS 5 cases, which are clinically critical.

In contrast, EfficientNetB0 struggled to generalize well. Though it showed promise in training performance, its macro F1-score remained low at 0.23, and it often predicted normal (BI-RADS 1) for all images. This behavior was reflected in the confusion matrix, where the majority of predictions were biased toward a single class. While EfficientNetB0 had higher overall accuracy at 84%, the class-level performance was uneven. This imbalance points to the model's tendency to overfit on dominant visual patterns.

ResNet50 performed the worst among all. Its accuracy dropped to 12%, and its macro F1-score was just 0.05, suggesting it failed to learn meaningful class boundaries. The model likely overfitted due to its deeper architecture and the small dataset. ResNet50 also showed instability during training, as seen in the validation curves. Its confusion matrix confirmed that it predicted the same class (BI-RADS 4) for almost all cases, which implies poor model learning and generalization.

In summary, MobileNetV2 provided the most balanced performance with high accuracy and stable training. CNN was a reliable and interpretable baseline. EfficientNetB0, despite being lightweight and modern, struggled with class imbalance. ResNet50 was too heavy for the dataset and failed to converge properly. These outcomes suggest that lightweight models with good regularization and tuning are more effective in resource-constrained medical tasks.

Discussion

Throughout this project, we faced multiple technical and practical challenges that impacted our workflow and experimental outcomes. First, one of the biggest limitations was dataset imbalance. BI-RADS 1 was the dominant class, while classes like BI-RADS 4 and 5 had very few examples. This created difficulty for all models, especially in learning to classify minority categories accurately. It also skewed performance metrics like accuracy and made macro F1-score a more realistic reflection of model capability.

Another challenge was model overfitting, particularly in deeper architectures like ResNet50. Despite using techniques like dropout, batch normalization, and data augmentation, ResNet50 failed to generalize. We hypothesize that the model's complexity was too high for our dataset, which was relatively small and specific to local medical imaging. On the other hand, lightweight models like MobileNetV2 showed better training behavior and generalization, supporting the idea that model complexity must match dataset size and variability.

We also conducted ablation studies to investigate the impact of specific components, such as the dropout layer. Turning off dropout significantly hurt model performance. The model without dropout overfitted quickly and showed unstable validation accuracy and loss. Misclassifications increased dramatically, and the final F1-score dropped below 0.01. This confirmed the importance of regularization in preventing overfitting, especially with small medical datasets.

Additionally, we struggled with hardware limitations and Grad-CAM implementation issues. Initially, Grad-CAM visualizations worked on a few samples. But later, due to TensorFlow/Keras version mismatches, custom layer naming, and improper model builds, Grad-CAM failed to generate meaningful heatmaps for all models. The issue was persistent even after verifying layer names and inputs. As a result, we could not include a full Grad-CAM analysis for visual explanations of model decisions. This is a limitation in our interpretability section and will be addressed in future versions.

Moreover, time constraints and repeated training due to kernel crashes delayed progress. We often had to retrain models, especially when testing multiple hyperparameters and augmentation settings. Every experiment was time-consuming due to the image size, model depth, and the use of real mammograms.

Finally, there were labeling inconsistencies and noise in the dataset, which likely confused the models during learning. While BI-RADS labels are helpful, they are still subjective to radiologist interpretation and may vary slightly. This kind of annotation noise can affect model learning and testing.

In conclusion, this project highlighted the value of transfer learning, but also revealed the limitations of small-scale medical AI projects. Proper preprocessing, careful model selection, and interpretability tools are essential. Though we could not fully visualize Grad-CAM results, the rest of our pipeline provided strong insights into what worked and what did not. These lessons will guide future improvements and help refine our model for real-world clinical use.

Link

Github

https://github.com/FasiulAbedinKhan/EEE385L_Project_BI-RADS-Classification

Dataset

https://dx.doi.org/10.21227/a4cs-ax02

References:

[1] International Agency for Research on Cancer, “Saudi Arabia fact sheet,” GLOBOCAN, 2022. [Online]. Available: https://gco.iarc.fr/today/data/factsheets/populations/682-saudi-arabia-fact-sheets.pdf

[2] W. S. Alqahtani, N. A. Almufareh, D. M. Domiaty, G. Albasher, M. A. Alduwish, H. Alkhalaf, B. Almuzzaini, S. S. Al-marshidy, R. Alfraihi, A. M. Elasbali, H. G. Ahmed, and B. A. Almutlaq, "Epidemiology of cancer in Saudi Arabia thru 2010–2019: A systematic review with constrained meta-analysis," *AIMS Public Health*, vol. 7, no. 3, pp. 679–696, Sep. 2020, doi: 10.3934/publichealth.2020053.

[3] S. Alessy, *et al.*, "Cancer research challenges and potential solutions in Saudi Arabia: A qualitative discussion group study," *J. Cancer Policy*, 2024.

[4] *King Abdulaziz University Breast Cancer Mammogram Dataset | IEEE DataPort*. (n.d.). https://ieee-dataport.org/documents/king-abdulaziz-university-breast-cancer-mammogram-dataset