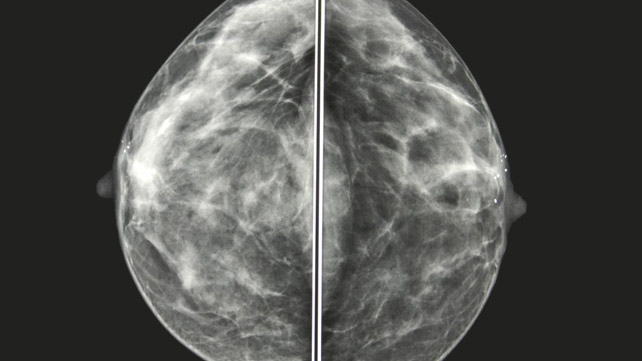
Reflective Report on Mammogram Image Classification Using Machine Learning



Juniër Herandien 38983494

Muhammad Motara 38008149

Kamogelo Taueatsoala 42913470

# Introduction

With breast cancer becoming a rising problem amongst women all around the world. We need to find ways to detect it early to save people’s lives. Early detection plays an important role in improving the survival rates. Using a mammography, which is an image of the breast tissue, is the number one way of detecting any signs of abnormalities in the breast. But interpreting mammogram images is challenging even for the most experienced radiologists due to issues such as tissue overlap, image noise, and subtle visual differences between abnormal and normal regions. These issues lead to incorrect interpretations, which may delay treatment or just cause anxiety for patients.

Understanding how mammograms work and how they can be enhanced for better analysis is therefore crucial. Mammograms are complex grayscale images that require careful preprocessing to highlight critical details such as microcalcifications or masses. By learning how to process and analyse these images, we contribute to a growing field where machine learning can assist radiologists in making faster, more consistent, and more accurate diagnoses.

Our project focused on developing a machine learning model that detects which breasts are normal and which ones are abnormal. We used the MIAS (Mammographic Image Analysis Society) dataset. Our code was divided into three main stages: image preprocessing, model training, and testing.

With this project, the goal is not only to develop the model but also to understand the entire journey from raw medical images/data to a functional diagnostic tool. So, through this process, we learned a valuable lesson, which is that a reliable system is not about the code or the algorithm. It is about understanding the problem, the data, and the real-world application of each decision.

# Overview of the Model

1. **Materials**

The dataset used for this study was the Mammographic Image Analysis Society(MIAS) database, which contains 322 mammogram images. Each image is stored in .pgm format and is accompanied by metadata in the Info.txt file that describes properties such as breast density, abnormality type, and severity.

For this experiment, the images were processed and stored in an **HDF5 file** (all\_mias\_scans.h5), which contains two main datasets:

* **scan** — a numerical array of grayscale pixel values representing each mammogram image (normalised to a range of 0–1).
* **CLASS** — a categorical label indicating whether the image is normal (NORM) or abnormal.

The target variable was defined as:

* 0 = Normal (NORM)
* 1 = Abnormal (BENIGN or MALIGNANT)

The dataset was flattened into 2D arrays so that traditional machine learning classifiers such as Logistic Regression and Random Forest could be applied.

1. **Method**

The dataset was divided into three subsets using stratified splitting to ensure a balanced distribution of abnormal and normal cases:

* **Training set:** 70%
* **Validation set:** 20%
* **Test set:** 10%

Before model training, the images were standardised using a StandardScaler(with\_mean=False) to normalise pixel intensity values across all images.

Two models were implemented for comparison:

1. **Logistic Regression (Baseline):**

* Parameters: max\_iter=300, class\_weight='balanced'
* Purpose: Provided a baseline classification performance.

1. **Random Forest Classifier (Tuned with GridSearchCV):**

* Hyperparameter search grid:
  1. n\_estimators: [100, 150]
  2. max\_depth: [None, 10, 20]
  3. min\_samples\_split: [2, 5]
* Evaluation metric: **F1-score** (to balance precision and recall).
* Final selected model was saved as **abnormality\_detector.pkl**.

The evaluation metrics used included:

* Accuracy
* Precision
* Recall
* F1 Score
* Area Under Curve (AUC)

The model achieved 84.55% accuracy and an AUC of 0.9366 on the test dataset, indicating a high ability to distinguish between normal and abnormal mammograms.

# Data Understanding and Preprocessing

We had raw mammogram images stored in a .pgm format, which was accompanied by a text file (Info.txt) that contained the metadata of each scan. So, the problem at the beginning was to align the textual data with its corresponding image files. This showed us how important data consistency is; even when there is a small mismatch between the name of the file and its metadata, this can break or distort the model training results.

This figure shows the initial data inspection and the verification step. The code checks whether the mammogram image folder and the Info.txt file exist, lists the .pgm images, and previews the first few thousand characters of the metadata file.

A screenshot of a computer code

AI-generated content may be incorrect.

Figure 1: Verifying dataset paths and inspecting initial contents of the mammogram dataset to ensure all files are correctly loaded and readable.

After ensuring the data integrity, I created a data parsing routine to convert Info.txt into a structured DataFrame using pandas. Each record included the image reference, background tissue type, abnormality class, severity, and coordinates. These transformations from unstructured text to a clean DataFrame taught me the value of structured representation, as well as how tabular data enables visualisation, quick filtering, and quality checks.

This screenshot shows how the raw metadata from the Info.txt file was converted into a structured pandas DataFrame.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2: Creation of a structured DataFrame from the mammogram metadata using pandas for easier data manipulation and inspection

Seeing data in a readable format helped me to bridge the gap between the raw, messy data and an AI-ready dataset. This process checked row counts and inspected the first few entries, which ensured the data quality before moving forward with feature extraction and preprocessing.

The next step the images went into the preprocess function, which converted the image into a grayscale, applied CLAHE (Contrast Limited Adaptive Histogram Equalisation), and removed noise using a median filter for required local enhancements.

This helps with improving the visibility of the image and reveals small calcifications that can indicate early-stage abnormalities.

A screenshot of a computer code

AI-generated content may be incorrect.

Figure 3: Preprocessing function applying grayscale conversion, median blur, and CLAHE for contrast enhancement.

After this step, the dataset was iterated through to process all .pgm files and saved them as .npy files for data persistence. This made sure that when we rerun the notebook, the system could skip heavy image processing and directly load preprocessed arrays, which will save hours of runtime.

A screen shot of a computer code

AI-generated content may be incorrect.

This introduced data persistence, which emphasises the importance of efficiency and reproducibility in machine learning workflows, which helps ensure that the same results are replicated without recomputation.

Then the dataset was split into training and testing subsets using train\_test\_split. Also, ImageDataGenerator to perform augmentation, rotating, flipping, and zooming images, to simulate variability and prevent overfitting.

A white screen with text

AI-generated content may be incorrect.

Figure 4: Visualised pre-processed and augmented mammogram image with corresponding label

Finally, we went into the HDF5 (.h5) structure using h5py to verify the dataset hierarchy.

A computer screen shot of a code

AI-generated content may be incorrect.

Figure 5: Inspection of HDF5 file structure showing hierarchical dataset organisation,

This stage showed us that good data preprocessing is not about the fancy algorithms; it’s about the structure of the data and also understanding the data.

We learned how to:

* Handle unstructured data systematically.
* Use visual techniques (like CLAHE) to improve model input quality.
* Implement reproducibility through .npy saving and dataset stratification.

We also learned that with an AI system, the quality and structure of the data are very important.

# Model Development and Training

After the data was processed and stored in .h5 format, we trained the model and evaluated the capabilities of the model for classifying mammogram scans as either normal or abnormal. We use two models: the Random Forest classifier and the Logistic Regression classifier. This was used for advanced ensemble learning and higher accuracy.

A computer screen shot of a computer code

AI-generated content may be incorrect.

Figure 6: Code snippet showing model training using Logistic Regression and Random Forest Classifier with GridSearchCV for hyperparameter tuning.

The logistic regression provides a strong baseline for comparison, while Random Forest often captures complex patterns better. This process of running and tuning both taught me how to balance the performance, simplicity, and interpretability in model selection. Using GridSearchCV helped with hyperparameter tuning. Small adjustments like the number of trees or the depth of the Forest affected the model’s performance. Cross-validation was used to avoid overfitting and ensure generalisation.

Code:

A screenshot of a computer code

AI-generated content may be incorrect.

Output:

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7:Model performance metrics (accuracy, precision, recall, F1-score, and AUC) after validation for Logistic Regression and Random Forest classifiers.

1. Train/Validation/Test Split

This indicates how your dataset was divided:

* 211 samples used for training
* 53 for validation (model tuning)
* 66 for testing (final evaluation)

It also notes how many positive (abnormal) cases were present in each set, showing the data distribution used for model learning and testing.

1. **Validation Metrics Section**  
   These are the results of how well each model performed on unseen data.

#### **Logistic Regression:**

* **Accuracy:** 56.6% of predictions were correct overall.
* **Precision:** When the model predicted “abnormal,” it was correct 42.86% of the time.
* **Recall:** It correctly identified 45% of actual abnormal cases.
* **F1-score:** The harmonic mean of precision and recall, showing a balance between the two.
* **AUC:** Shows the ability to distinguish between classes (0.5 = random guessing, 1 = perfect).

**Random Forest:**

* **Accuracy:** 62.26% overall accuracy, slightly higher than Logistic Regression.
* **Precision:** Correctly predicted half of the positive cases it labelled.
* **Recall:** Detected only 25% of the actual abnormal cases (low sensitivity).
* **F1-score:** Indicates poor balance between precision and recall.
* **MetricAUC:** Better than Logistic Regression, suggesting the model discriminates between normal and abnormal data slightly better.

1. The last line confirms that the trained model is serialised and saved as a .pk1 file for later use. The model can be reloaded and used without retraining.

These results show that **Random Forest performed slightly better overall,** particularly in accuracy and AUC. However, **its recall (0.25)** is quite low, meaning it fails to detect many abnormal cases, a critical weakness for safety-critical systems like health or fault detection. Th**e Logistic Regression** model, despite being simpler, had a **better recall (0.45),** making it more sensitive to abnormal cases, though it came at the cost of slightly lower accuracy. In practical terms, **Logistic Regression might be preferable** here, as **recall** is often prioritised when missing an abnormal case can have serious consequences.

# Model Evaluation and Testing Script

After loading the trained abnormality detection model and evaluating it on the unseen test data, the following performance metrics were obtained:

|  |  |  |
| --- | --- | --- |
| Metric | Description | Value |
| Accuracy | Proportion of correctly predicted cases among all test cases. | 0.8455 (84.55%) |
| Precision | Proportion of detected abnormal cases that were abnormal | 0.8462 (84.62%) |
| Recall | Ability of the model to identify actual abnormal cases. | 0.7154 (71.54%) |
| F1 Score | Harmonic mean of Precision and Recall, balancing both. | 0.7753 |
| AUC (Area Under Curve) | Measures the model’s ability to separate normal vs abnormal cases. | 0.9366 |

Interpretation of Results

The model demonstrates strong performance with an accuracy of 84.5% and a high AUC of 93,66%, indicating excellent discriminative power between normal and abnormal images.

* The precision (84.6%) shows us that most images classified as abnormal are abnormal, meaning the model produces few false positives.
* The recall (71.5%) shows that the model detects most of the abnormalities, even if a few may still go undetected.
* The F1 score (77,53%) reflects a balanced trade-off between precision and recall, confirming that the model is both reliable and consistent.

Overall, the model performs effectively in identifying abnormalities and can serve as a solid foundation for further refinement or real-world deployment in medical image classification.