Ministry of Education
Imam Abdulrahman Bin Faisal University
Collage of Computer Sciences
and Information Technology
Department of Computer Sciences

جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY

وزارة التعليم
جامعة الإمام عبد الرحمن
بن فيصل
كلية علوم الحاسب وتقنية المعلومات
قسم علوم الحاسب

Image processing - ARTI407

Project submission

Breast Cancer Detection Using Image Processing Techniques in Mammograms

No.	Name	ID
1	Ritaj Alhamli	2210002809
2	Nora Abdullah Aljomuh	2220004452
3	Mozoon Alkhalis	2220002873
4	Joud Alahmari	2220001872

Supervised by: *Dr. Noor Felemban*

Submitted: *3rd of May 2025*

PROJECT ABSTRACT

In this project, we aim to develop a system that can help detect breast cancer using image processing techniques. The goal is to enhance the quality of the mammogram images and to help doctors see if there are any signs of cancer earlier. We used image processing techniques that include intensity transformations, noise reduction, and segmentation. Additionally, we used a publicly available dataset named MIAS Mammography ROIs to test our system.

In the beginning, we used several methods to enhance the image quality and identify the cancer regions, some of them did not provide us with good results because we used very simple techniques. Later, in the final design, we were able to achieve a higher accuracy by combining some traditional machine learning algorithms such as Random Forest. Our findings show that combining image processing techniques with machine learning improves the results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

3 INTRODUCTION

3.1 Problem Formulation

3.1.1 Problem Statement:

One of the most common diseases in the world that effects the women life is breast cancer. Early detection is important because it can increase survival rates and enhance the effectiveness of treatment. Mammography is the most often used screening tool. However, it has some limitations such as low contrast, noise, and overlapping tissues, making it difficult to detect abnormalities [2]. To address these problems, image processing techniques can improve mammography pictures, allowing doctors to detect cancer more accurately. In this project, we will look at several image processing techniques that include intensity transformations, noise reduction, and segmentation. These techniques can enhance breast cancer diagnosis in mammograms, which will make it easier to detect abnormalities like tumours. Additionally, to improve the detection accuracy we will integrate machine learning approaches.

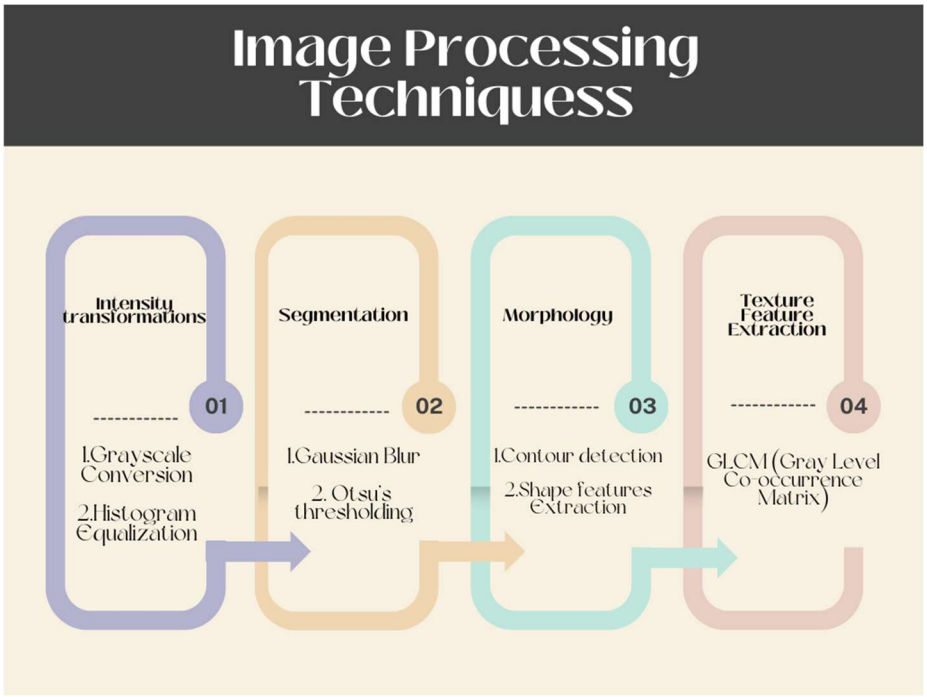

Figure 1: Image Processing Techniques Applied in the Proposed System

3.2 Project Specifications

To ensure that our project performs effectively and solves the problem, we developed a set of explicit and measurable design goals. These goals let us know if our approach is successful or if we need to improve it.

- ✓ **Image quality:** enhancing the quality of the mammography pictures is the main goal. After applying our approach, the images should be clearer for the machine.
- ✓ **Accuracy:** we must ensure that the model produces more accurate results whenever we improve the image, our goal is to achieve at least 80% accuracy.
- ✓ **Efficiency:** the system must be fast, our goal is to have the system process all the dataset in 5 minutes.

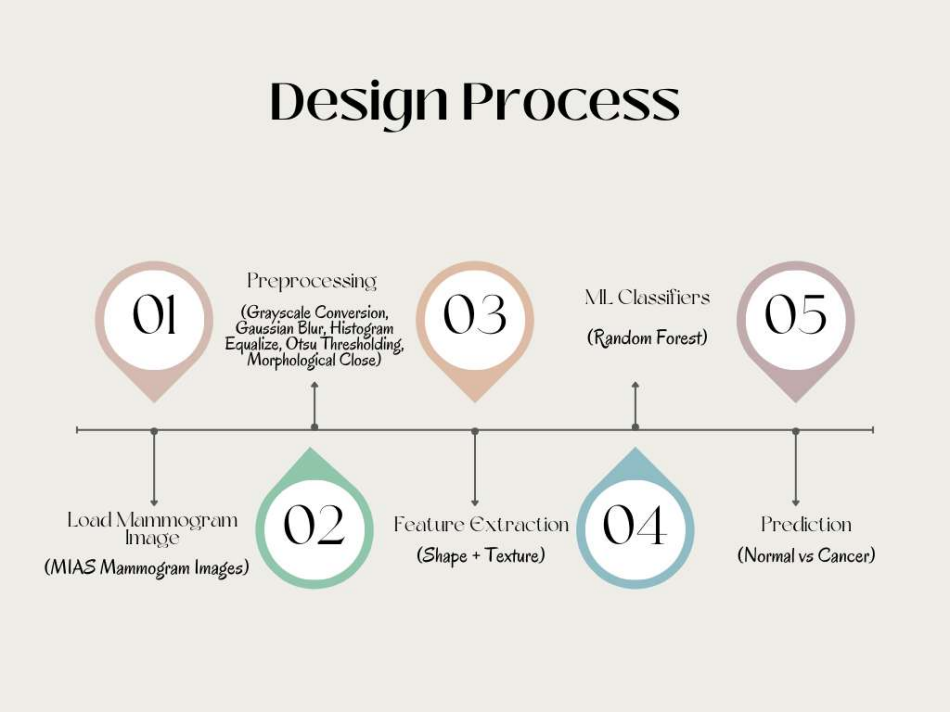

Figure 2: Design process

4 BACKGROUND

4.1 Literature Review:

Breast cancer is one of the leading causes of death among women worldwide, and early detection is important for improving survival rates and mammography is the standard imaging technique for identifying early signs of breast cancer. Microcalcifications (MCs) are tiny calcium deposits that looks like white spots in mammography images are often considered as early indicators of breast cancer. However, detecting them accurately remains a challenge due to their small size and the complexity of breast tissue structures whereby image processing techniques are mainly needed. This literature review discovers various methods used for microcalcification recognition and focuses on the latest advancements in image processing techniques.

Mammography has been the gold standard for breast cancer screening, but manual analysis by radiologists is probable to variability. Studies show that traditional visual analysis of mammograms has limitations, including observer bias and false-positive/negative rates. Tight clusters of microcalcifications usually are an indicator of malignant cells, yet their identification is subjective and relies roughly on expert evaluation and experience. To get past the limitations Computer-Aided Detection (CAD) techniques have been developed to enhance radiologists' accuracy in identifying microcalcifications. Early CAD systems used simple thresholding and edge detection methods, but their performance was limited too because of noise and poor contrast in mammography images. Through our search for Image processing papers in detecting breast cancer we found out that latest modern developments have combined both machine learning and deep learning for high accuracy automated detection systems.

Mahmood et al. (2021) illustrated a radiomics-based approach that combines classical image enhancement methods with machine learning classifiers for detecting and grouping MCs in the dataset. The method implements preprocessing mammogram images using wavelet transforms and the top-hat morphological operator to improve and enhance MC visibility and suppress background noise. Features were extracted using gray level co-occurrence matrices (GLCM), histogram-based statistics, and morphological descriptors and subsequently classified using support vector machines (SVM), k-nearest neighbors (K-NN), and random forests (RF). Their system achieved a high accuracy of 98% with matching sensitivity and significantly reduced false positives (1.2 FPi) showing the method's robustness.

The study proved how combining wavelet decomposition and morphological filtering improves segmentation and classification outcomes [1].

Avcı and Karakaya (2023) focused on the effect of image preprocessing techniques on the performance of breast lesion classification. Using the MIAS dataset they experimented with combinations of CLAHE (contrast-limited adaptive histogram equalization), median filtering, and unsharp masking. These preprocessing steps were followed by k-means clustering for segmentation and machine learning classifiers like SVM, RF, and neural networks (NN) for two-stage classification (normal vs. abnormal, and benign vs. malignant). The study concludes that using CLAHE alone reached to insignificant results, whereas combining it with median filtering and unsharp masking significantly improved classification accuracy and this is the goal. Their findings emphasized the crucial role of proper image enhancement in boosting machine learning performance for mammographic analysis [2].

Jafari and Karami (2023) proposed a deep learning-based pipeline that selects features from many pre-trained convolutional neural networks (CNNs), including AlexNet, ResNet, and MobileNet and then implements feature selection using mutual information. These features were classified using traditional machine learning algorithms such as SVM, RF, K-NN, and neural networks. The model is evaluated using three datasets achieving classification accuracies of up to 96% on DDSM and 94.5% on MIAS. Their study demonstrated that deep CNNs when combined with feature selection outshine classical models in terms of accuracy especially when large and varied datasets are available. The addition of non-image features such as patient age and breast density further improved the results, advising the value of multimodal data integration [3].

In contrast to the more data-hungry deep learning methods, a 2024 study proposed a classical image processing approach employing Wiener filtering, linear time-invariant (LTI) system modeling, and the morphological top-hat operator. This methodology's main objective is to improve the clarity of MCs in grayscale mammograms by reducing noise and increasing contrast making the images more appropriate and suitable for subsequent processing. While specific classification results were not fully detailed the emphasis was placed on improving the initial image quality, which is foundational for accurate segmentation and diagnosis in computer-aided detection (CAD) systems. Studies proved that comparing LFWT with other approaches show that LFWT achieves the highest accuracy (99.5%) this approach succeeds in detecting microcalcifications in mammography images [4].

Comparing the four studies reveals a clear direction toward hybrid techniques that combine classical enhancement and segmentation with machine or deep learning classification. Mahmood et al. and Avcı & Karakaya demonstrated the power and the strength of classical methods in enhancing diagnostic accuracy, mainly when using carefully tuned preprocessing filters. Jafari & Karami, on the other hand showed how CNN-based systems can succeed higher accuracy, although with increased computational complexity and dependence on large datasets. The 2024 study supported the continued relevance of classical filtering techniques specifically for systems focused on early-stage image enhancement.

In conclusion, all of the findings of these investigations demonstrate the advantages and strengths of both modern and conventional methods for MC identification in mammography. For preprocessing and segmentation traditional techniques are still crucial since they are interpretable and have a low-cost computing cost. In the meantime, when sufficiently trained on data, deep learning methods achieve higher classification accuracy. Building trustworthy and accurate CAD systems for breast cancer diagnosis may require combining the two paradigms, utilizing machine learning or CNNs for classification and traditional image processing for enhancement and segmentation.

Table 1: Comparison of Image Processing Approaches for Breast Cancer Detection

Paper Ref	Goal	Technique	Dataset(s)	Results	Strengths
[1]	Detect and classify microcalcifications using radiomics	Wavelet + Top-hat, GLCM, Morphology, SVM, K-NN, RF	MIAS	Accuracy: 98%, Sensitivity: 98%, AUC: 0.90	Combines wavelet + morphology for high accuracy; low false positives
[2]	Analyze effect of preprocessing on classification accuracy	CLAHE, Median Filter, Unsharp Masking + K-means + ML classifiers (SVM, RF, NN)	MIAS	Accuracy improved with filter combinations	Emphasizes importance of preprocessing combinations
[3]	CNN feature-based BC detection with feature selection	Feature extraction from CNNs (AlexNet, ResNet, MobileNet) + ML classification	RSNA, MIAS, DDSM	Accuracy: 94.5% (MIAS), 96% (DDSM), 92% (RSNA)	Hybrid deep learning + ML approach; uses multimodal features
[4]	Enhance image quality and detect MC using classical filters	Wiener filter + LTI system + Top-hat morphology	Likely MIAS	Accuracy: up to 99.5% (reported)	Classical method with strong enhancement and low complexity

4.2 Concept Synthesis

4.2.1 Concept Generation:

To address the challenge of detecting breast cancer in mammogram images, we explored multiple conceptual approaches through literature review, brainstorming, and practical experimentation. Our goal was to identify a pipeline that improves classification accuracy while staying within our computational and dataset limitations. We evaluated the following directions:

1. Baseline Models on Raw Data (No Preprocessing or Feature Engineering)

Initially, we applied Random Forest classifiers directly to the raw mammogram data, without any preprocessing. We also experimented with an ensemble of both classifiers to boost performance.

- Test Accuracy – Random Forest: 64.58%

  The model underperformed due to raw, unprocessed nature of the image data, which degraded the overall output.

- Test Accuracy – SVM: 35.42%

  The SVM model might struggle with image data, probably due to the fact that it usually relies on carefully chosen features.

This baseline phase provided a reference point for subsequent improvements and highlighted the limitations of using raw data without enhancement.

2. Deep Learning with CNNs (Not Implemented)

We considered using convolutional neural networks (CNNs) such as ResNet and MobileNet based on strong results from the literature (e.g., Jafari and Karami [3], reporting accuracy up to 96%). However, due to the limited size of the MIAS dataset and lack of GPU access, deep learning was ruled out for practical reasons.

3. Random Forest with Preprocessing and Feature Engineering (Chosen Model)

We then shifted to a classical image processing pipeline combined with traditional machine learning, aiming for a balanced approach that boosts accuracy without requiring high-end hardware.

Pipeline Details:

- **Preprocessing:** Histogram equalization, Gaussian blur, and Otsu's thresholding.
- **Feature Extraction:**
  - Shape Features: Area, circularity, solidity, extent from image contours.
  - Texture Features: Contrast, energy, and homogeneity using Gray Level Co-occurrence Matrix (GLCM).
- **Classification**:
  - Initially used an ensemble of Random Forest and SVM.
  - Conducted hyperparameter tuning for both models.

Results after Preprocessing and Tuning:

- Test Accuracy – Random Forest: 83.33%
- Test Accuracy – SVM: 35.42% (unchanged)
- Test Accuracy – Ensemble: 77.08% (No improvement still degraded by SVM)

Despite tuning, SVM's performance remained poor and did not complement Random Forest in the ensemble.

4.2.2 Concept Reduction:

After evaluating the performance of all models, we conducted a reduction process to eliminate less effective or redundant components:

- **SVM**: Removed due to consistently low accuracy (35.42%) and poor generalization.
- **Ensemble (RF + SVM)**: Dropped because it underperformed compared to Random Forest alone (77.08% vs. 83.33%).
- **Hyperparameter Tuning**: Retained only for exploration purposes; results did not outperform default parameters.

Final Decision: Use Random Forest with image processing techniques, and Default Parameters, since we believe that incorporating SVM model is actually reducing the performance rather than improving it.

- Achieved the best test accuracy (**83.33%**).
- Balanced performance, simplicity, and resource efficiency.

Performance Visualization: Final Random Forest Model To further analyze the classification performance of our final model, we include the confusion matrix below:

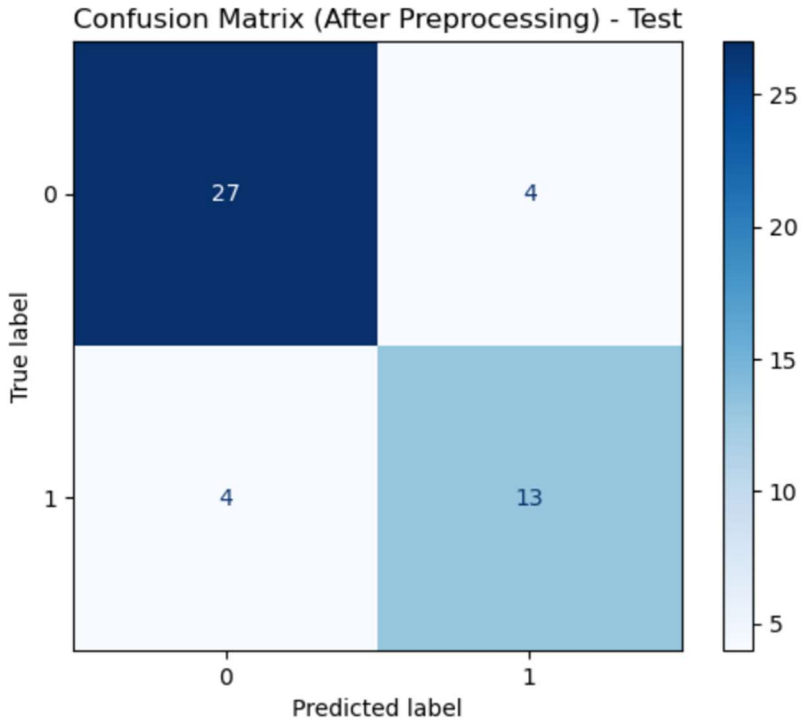

Figure 3: Confusion Matrix on the test set

Figure 3 demonstrates that our model achieves strong performance on the test set, which is notable given the challenge of identifying cancerous regions. The confusion matrix reveals relatively low counts of both False Positives (4) and False Negatives (4).

4.3 Detailed Engineering Analysis and Design Presentation

To solve the problem of early breast cancer detection in mammograms, we designed a hybrid approach that integrates classical image processing techniques with machine learning-based classification. The design process was carefully structured, starting from raw images and progressing through preprocessing, feature extraction, and classification.

4.3.1 Dataset

This project utilizes the **MIAS Mammography ROIs** dataset, a curated and preprocessed version of the original MIAS (Mammographic Image Analysis Society) database. The dataset comprises 1,679 region-of-interest (ROI) images, each extracted from full mammograms to focus on areas of clinical significance [5].

Key Characteristics:

- **Total Images**: 1,679 ROI images.
- **Image Format**: Grayscale PNG images.
- **Image Dimensions**: Each image is uniformly sized at 224×224 pixels.
- **Annotations**: Each ROI is labeled based on expert analysis, indicating the presence and type of abnormality such as benign, malignant, or normal tissue.

4.3.2 Preprocessing Steps: Enhancing Image Quality

4.3.2.1 Gray Scale:

converts RGB images to grayscale to simplify analysis and reduce computational complexity, as color information is not essential for mammograms [6].

Feature: Eliminates color information, focusing only on intensity—essential for mammogram analysis.

Function used: cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

Image Example:

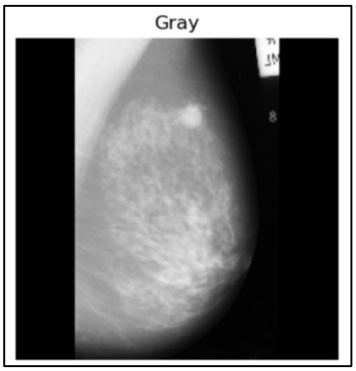

Figure 4: Gray Scale Image

4.3.2.2 Gaussian Blurring

Reduces high-frequency noise and smoothens the image, making it easier to detect the true contours of suspicious areas using kernel size (5, 5) defines the extent of the smoothing [7].

Feature: Enhances contour detection by suppressing small fluctuations.

Function used: cv2.GaussianBlur(gray, (5, 5), 0)

Image Example:

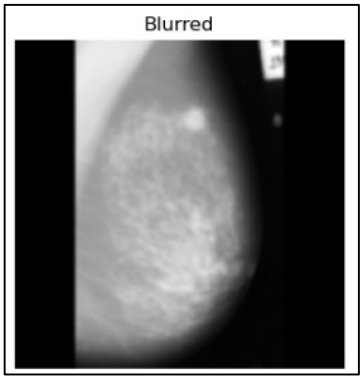

Figure 5: Gaussian Blurring

4.3.2.3 Histogram Equalization

improves contrast in grayscale images by spreading out the most frequent intensity values, helping to highlight tumor regions.

Feature: Redistributes pixel intensity for better visibility of tissue differences.

Function used: `cv2.equalizeHist(blurred)`

Image Example:

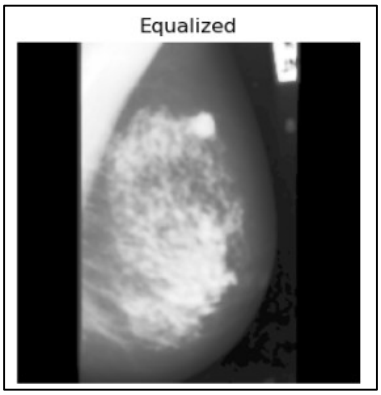

Figure 6: Histogram Equalization

4.3.2.4 Thresholding (Otsu's method)

Automatically separates the foreground (potential tumor regions) from the background using intensity values

Feature: Automatically finds optimal threshold for segmentation.

Function used: `cv2.threshold(equalized, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)`

Image Example:

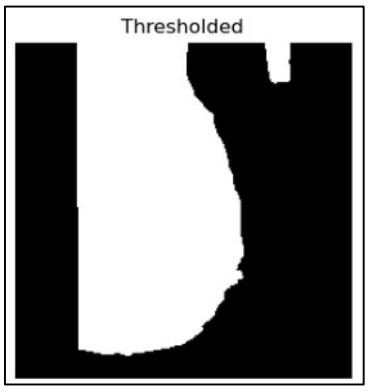

Figure 7: Thresholding

4.3.2.5 Morphological Closing

Closes small holes inside foreground regions and connects nearby objects, refining the segmentation output.

Feature: Refines the segmentation, preparing it for contour detection.

Function used: cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, kernel, iterations=2)

Image Example:

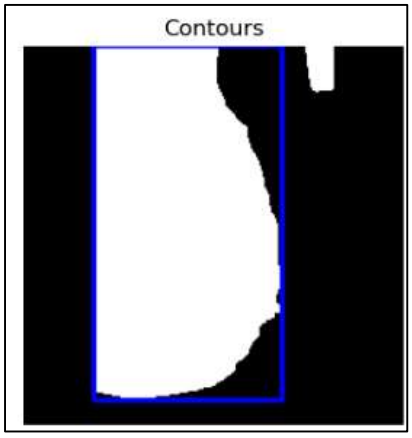

Figure 8: Morphological Closing

4.3.3 Feature Extraction: Understanding Suspicious Regions

After segmenting potential tumor regions through the preprocessing pipeline, we extracted descriptive features that would enable effective classification of mammogram images. These features were categorized into two main types: shape-based and texture-based, providing both geometric and statistical insights into the segmented areas.

4.3.3.1 Shape based features:

Shape-based features are crucial for identifying abnormalities that deviate from regular tissue shapes. For each significant contour (area > 1000 pixels), we extracted the following features:

- **Circularity:** Calculated as $4\pi \times {Area}/{\text{Perimeter}^2}$ indicating how round a shape is, tumours typically appear less circular.
- **Solidity:** Ratio of the contour area to its convex hull area. Lower solidity may indicate irregular growth.
- **Extent:** Ratio of the contour area to its bounding rectangle area.
- **Suspicious Region Count:** Number of detected contours with circularity less than 0.70.
- **Total Suspicious Area:** Sum of areas of all suspicious regions.

These values help differentiate between regular anatomical structures and tumour-like patterns.

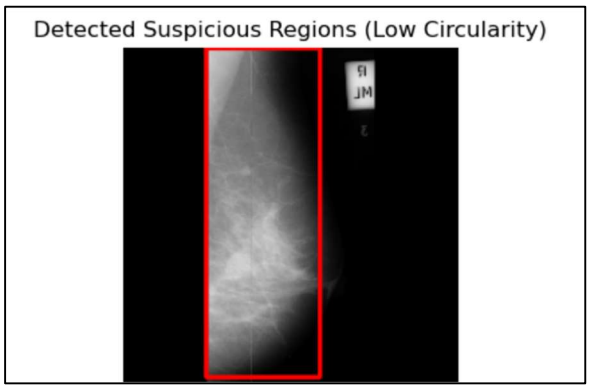

Figure 9: Detected contours representing suspicious regions.

Figure 8: Detected contours representing suspicious regions. Blue bounding boxes highlight regions with irregular shapes (circularity < 0.70), which are often correlated with malignant tissue.

4.3.3.2 Texture Based Features Shape based features:

Texture features help distinguish between normal and abnormal tissue patterns by analyzing the spatial relationship between pixel intensities. These are extracted using the **Gray-Level Co-occurrence Matrix (GLCM),** which is a statistical method that describes how often pairs of pixel intensities occur in an image at a given distance and orientation

Texture Based Features

Table 2: Gray-Level Co-occurrence Matrix (GLCM) properties

Contrast	Measures how similar or uniform the image is. High values mean pixels are similar to their neighbors.
	Interpretation: A low contrast might suggest abnormal textures typically associated with tumor tissue.
	Function used: `graycoprops(glcm, 'contrast')`
Homogeneity	Measures how similar or uniform the image is. High values mean pixels are similar to their neighbors.
	Interpretation: A low homogeneity might suggest abnormal textures typically associated with tumor tissue.
	Function used: `graycoprops(glcm, 'homogeneity')`
Energy	Also known as angular second moment; it measures image uniformity or smoothness.
	Interpretation: Lower energy can indicate coarse or grainy textures, which may correspond to pathological features.
	Function used: `graycoprops(glcm, 'energy')`

4.3.4 Classification

As suggested in our literature review to achieve better accuracy, we used **machine learning classifiers** to categorize the regions of interest as either cancerous or normal based on the features we extracted.

Supervised learning models were used:

- **Random Forest (RF)**: This model builds many decision trees and combines their results. We used 200 trees; each limited in depth to avoid overfitting and make the model more general.

4.3.5 Evaluation & Visualization

To evaluate the performance of the system, we used the **validation set**, which contains ROI images with known labels. For each image:

- The predicted class (normal or cancerous) is compared with the ground truth**.**
- Visual outputs are generated, showing:
  - The original ROI
  - Intermediate preprocessing steps (grayscale, equalized, thresholded)
  - The final detection result with bounding blue boxes around suspicious regions

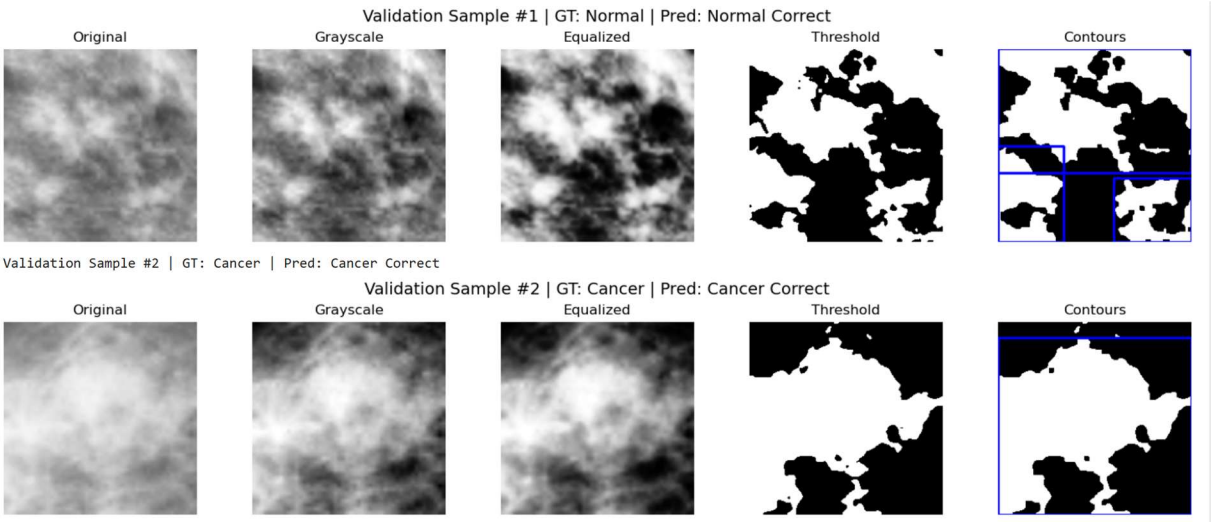

Figure 10: Visualization on performance and processing techniques (Validation set)

The Random Forest, after the image processing techniques mentioned in previous sections, achieved an **accuracy of 87.23%** on the validation set, indicating strong performance in detecting cancerous regions.

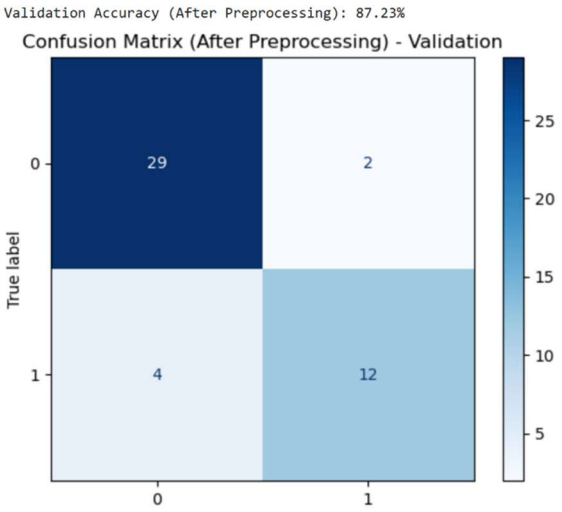

Figure 11: Confusion Matrix on the validation set

As shown in figure 11, our model's performance on the validation set in remarkably high considering a task such as detecting cancerous regions, in the confusion matrix we found that False Positives (2) and False Negatives (4) are relatively low. We believe the image processing technique had a big role in increasing our accuracy.

Additionally, as shown in Figure 3, the model achieved an accuracy of 83.33% on the test set, with both False Positives (4) and False Negatives (4) remaining relatively low. This indicates strong generalization to unseen data.

5 BREAST CANCER DETECTION USING IMAGE PROCESSING TECHNIQUES

5.1 System Architecture Overview

Our system follows a pipeline structure that combines classical image processing techniques with machine learning classifiers. The architecture consists of the following core stages:

1. **Preprocessing Module:** Handles image enhancement (grayscale conversion, Gaussian blur, histogram equalization, thresholding, and morphological operations).
2. **Feature Extraction Module:** Extracts shape-based and texture-based features.
3. **Classification Model:** Utilizes Random Forest classifier to determine whether the region is cancerous.
4. **Evaluation Module:** Validates model predictions and visualizes outputs.

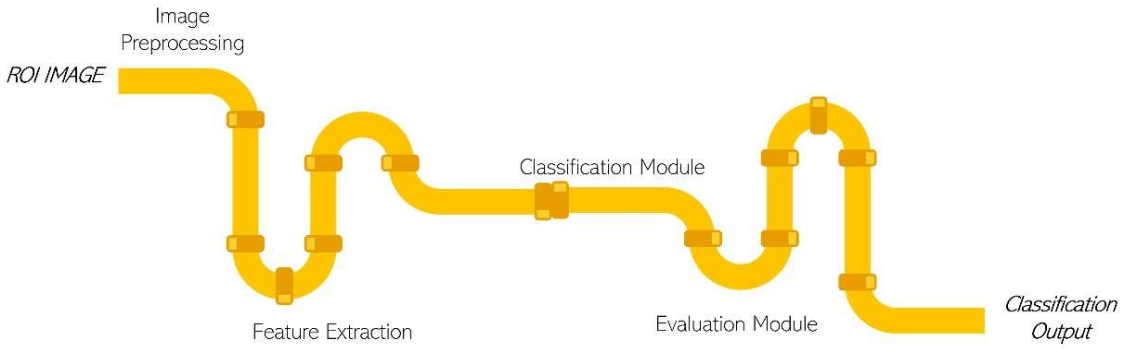

Figure 11: System Architecture Pipeline

5.2 Implementation Details

This section summarizes the tools, methods, and system setup used to implement our breast cancer detection pipeline, including preprocessing, feature extraction, and classification steps.

Table 3: Implementation Details Table

Category	Details
Programming Language	Python 3.10
Development Platform	Jupyter Notebook
Main Libraries Used	OpenCV, scikit-learn, NumPy, Matplotlib, skimage
System Specs	Intel Core i7, 16 GB RAM, Windows 10
Dataset Used	MIAS Mammography ROIs (PNG format, 224×224 pixels)
Preprocessing Steps	Grayscale → Gaussian Blur → Histogram Equalization → Otsu's Thresholding → Morphological Closing
Feature Extraction	- **Shape-based:** area, circularity, solidity, extent, suspicious count - **Texture-based:** contrast, homogeneity, energy using GLCM
Classification Models	- Random Forest
Output & Visualization	Prediction (Normal/Cancerous), with bounding boxes and processing steps visualization
Testing Accuracy	83.33%
Cost	All tools and datasets are publicly available and free to use

5.3 Challenges and improvements

One of the major challenges we encountered was improving the classification accuracy of our system. At the beginning, we experimented with using only classical image processing techniques such as thresholding and morphology to directly classify breast cancer regions. However, this approach proved to be insufficient, resulting in a low accuracy of approximately 50%, which was far below our target.

To overcome this limitation, we began integrating machine learning into our pipeline. By extracting meaningful shape and texture features from the processed images and applying models such as Support Vector Machine (SVM) and Random Forest (RF), we were able to significantly improve performance.

Through multiple iterations of tuning preprocessing steps and classifier settings, our final ensemble model reached a validation accuracy of 77.08%.

This transition from purely image-based methods to a hybrid ML approach was essential to achieving higher results, but after thorough evaluation and experimenting we found out that the Random Forest alone had higher accuracy (83.33%) than the hybrid approach (77.08%). SVM model is not the applicable choice in our case since it was actually degrading our model rather than improving it. The transition to using Random Forest alone led us to better results on our validation set and testing set.

Table 5 summarizes the key experiments conducted during the development of our breast cancer detection system, showing the impact of different techniques on classification accuracy.

Table 4: Experiment Table

Experiment	Preprocessing	Classifier	Accuracy
Exp. 1	Classical	None	47.92%
Exp. 2	None	SVM	35.42%
Exp. 3	None	Random Forest	64.58
Exp. 4	Classical	SVM + Random Forest	77.08%
Exp. 5 (Chosen Model)	Classical	Random Forest	83.33%

5.4 Model's result & discussion

To evaluate the performance of our breast cancer detection system, we calculated key classification metrics using the testing set using it to get the final results. The model achieved the following additional metrics:

- **Accuracy:** 83.33%
- **Precision (Weighted Average)**: 83%
- **Recall (Weighted Average)**: 83%
- **F1-Score (Weighted Average)**: 83%

These metrics indicate that our model is making good predictions overall, but there is still room for improvement, particularly in handling our Cancer class.

The confusion matrix provides deeper insight into the prediction results:

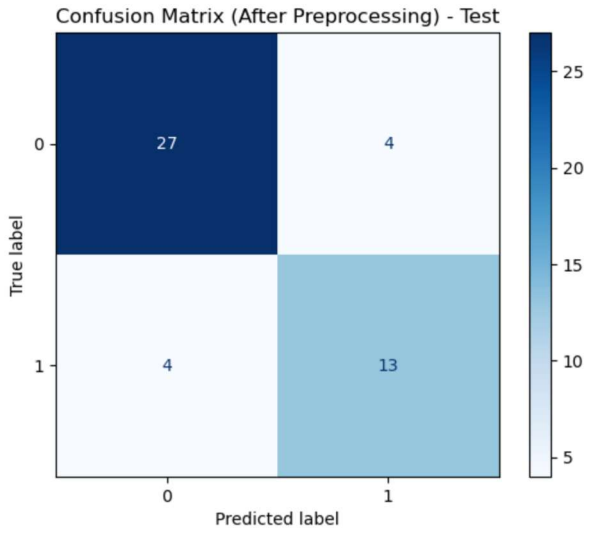

Figure 12:Model Confusion Matrix

The matrix shows that:

- **Normal Class (0)**: our model performs well with 27 correct predictions (TN) and only 4 false positives**.**
- **Cancer Class (1)**: our model performs well with 13 cancer predictions (TP) but misses 4 cancer cases (FN), which is a concern in detecting cancer precisely.

In detail:

- 27 normal cases were correctly classified.
- 13 cancer cases were correctly detected.
- 4 cancer cases were missed (false negatives).
- 4 normal cases were incorrectly classified as cancer (false positives)

This matrix highlights the model's strong sensitivity, which is essential in medical applications especially where missing a cancer diagnosis can have serious implications. However, the relatively lower precision indicates the presence of false alarms. We aim to improve this in our future work by refining feature selection and exploring more advanced classification techniques to enhance the model's overall reliability.

6 CONCLUSIONS AND FUTURE WORK

Working on this project our aim was to fulfil all requirements and produce an intelligent solution that meets our expectations to assist in detecting breast cancer using image processing techniques. After thorough research we found that combining image processing technique with the Random Forest classifier is the most suitable solution to our case, which is used on the MIAS Mammography ROIs dataset. Our solution provides considerably high accuracy (83.33%), For future work, the image processing pipeline can be improved using more advanced techniques. As an alternative of relying on classical image processing techniques such as thresholding, blurring, and equalization, using methods that is designed specifically to work better on X-ray images. Moreover, using noise reducing techniques such as the gaussian blur can remove details that are significant in dealing with detecting a tumor, refining how it works and limiting the blurring effect can significantly improve the solution. These suggested improvements may elevate the system to handle real-life images.

REFERENCES

[1] T. Mahmood, J. Li, Y. Pei, F. Akhtar, A. Imran, and M. Yaqub, "An Automatic Detection and Localization of Mammographic Microcalcifications ROI with Multi-Scale Features Using the Radiomics Analysis Approach," Cancers, vol. 13, no. 23, p. 5916, Nov. 2021.

[2] F. Avcı and A. Karakaya, "A Novel Medical Image Enhancement Algorithm for Breast Cancer Detection on Mammography Images Using Machine Learning," Diagnostics, vol. 13, no. 2, p. 348, Jan. 2023.

[3] Z. Jafari and E. Karami, "Breast Cancer Detection in Mammography Images: A CNN-Based Approach with Feature Selection," Information, vol. 14, no. 7, p. 410, Jul. 2023.

[4] High Accuracy Microcalcifications Detection of Breast Cancer Using Wiener LTI Tophat Model, IEEE Access, Aug. 2024.

[5] "MIAS Mammography ROIS," Kaggle, May 17, 2023. https://www.kaggle.com/datasets/annkristinbalve/mias-mammography-rois

[6] R. Kundu, "Image Processing: Techniques, Types, & Applications [2024]," V7. https://www.v7labs.com/blog/image-processing-guide

[7] "Gaussian Blur," ScienceDirect Topics, [Online]. Available: https://www.sciencedirect.com/topics/engineering/gaussian-blur.

APPENDICES

```
IMPORT NUMPY AS NP
IMPORT CV2
IMPORT MATPLOTLIB.PYPLOT AS PLT
FROM SKLEARN.ENSEMBLE IMPORT RANDOMFORESTCLASSIFIER
FROM SKLEARN.METRICS IMPORT ACCURACY_SCORE, CONFUSION_MATRIX, CLASSIFICATION_REPORT,
CONFUSIONMATRIXDISPLAY
FROM SKIMAGE.FEATURE IMPORT GRAYCOMATRIX, GRAYCOPROPS

# ========== LOAD DATASET ==========
TRY:
   X_TRAIN = NP.LOAD(R"MIAS_X_TRAIN_ROI_MULTI.NPY")
   Y_TRAIN = NP.LOAD(R"MIAS_Y_TRAIN_ROI_MULTI.NPY")
   X_TEST = NP.LOAD(R"MIAS_X_TEST_ROI_MULTI.NPY")
   Y_TEST = NP.LOAD(R"MIAS_Y_TEST_ROI_MULTI.NPY")
   X_VALID = NP.LOAD(R"MIAS_X_VALID_ROI_MULTI.NPY")
   Y_VALID = NP.LOAD(R"MIAS_Y_VALID_ROI_MULTI.NPY")
EXCEPT FILENOTFOUNDERROR:
   PRINT("DATASET NOT FOUND. CHECK THE FILE PATHS.")
   EXIT()

# ========== BINARIZE LABELS ==========
DEF BINARIZE(Y):
   RETURN NP.ARRAY([1 IF I IN [1, 2] ELSE 0 FOR I IN Y])

# BINARIZE THE LABELS FOR TRAIN, TEST, AND VALID DATASETS
Y_TRAIN_BIN = BINARIZE(Y_TRAIN)
Y_TEST_BIN = BINARIZE(Y_TEST)
Y_VALID_BIN = BINARIZE(Y_VALID)

# ========== FEATURE EXTRACTION ==========
DEF EXTRACT_FEATURES(X, Y_BIN):
   FEATURES, LABELS = [], []
   FOR I IN RANGE(LEN(X)):
      IMG = X[I].ASTYPE(NP.UINT8)
      GRAY = CV2.CVTCOLOR(IMG, CV2.COLOR_RGB2GRAY)
      BLURRED = CV2.GAUSSIANBLUR(GRAY, (5, 5), 0)
      EQUALIZED = CV2.EQUALIZEHIST(BLURRED)
      _, THRESH = CV2.THRESHOLD(EQUALIZED, 0, 255, CV2.THRESH_BINARY + CV2.THRESH_OTSU)
      MORPHED = CV2.MORPHOLOGYEX(THRESH, CV2.MORPH_CLOSE, NP.ONES((3, 3), NP.UINT8),
ITERATIONS=2)
```

```
    CONTOURS, _ = CV2.FINDCONTOURS(MORPHED, CV2.RETR_EXTERNAL,
CV2.CHAIN_APPROX_SIMPLE)

    SUSPICIOUS_COUNT, SUSPICIOUS_AREA, CIRCULARITY_LIST = 0, 0, []
    SOLIDITY_LIST, EXTENT_LIST = [], []

    FOR CNT IN CONTOURS:
      AREA = CV2.CONTOURAREA(CNT)
      IF AREA < 1000:
        CONTINUE
      PERIMETER = CV2.ARCLENGTH(CNT, TRUE)
      IF PERIMETER == 0:
        CONTINUE

      CIRCULARITY = 4 * NP.PI * AREA / (PERIMETER ** 2)
      X, Y, W, H = CV2.BOUNDINGRECT(CNT)
      RECT_AREA = W * H
      HULL = CV2.CONVEXHULL(CNT)
      HULL_AREA = CV2.CONTOURAREA(HULL)

      EXTENT = AREA / RECT_AREA IF RECT_AREA > 0 ELSE 0
      SOLIDITY = AREA / HULL_AREA IF HULL_AREA > 0 ELSE 0

      IF CIRCULARITY < 0.70:
        SUSPICIOUS_COUNT += 1
        SUSPICIOUS_AREA += AREA
        CIRCULARITY_LIST.APPEND(CIRCULARITY)
        SOLIDITY_LIST.APPEND(SOLIDITY)
        EXTENT_LIST.APPEND(EXTENT)

    GLCM = GRAYCOMATRIX(EQUALIZED, DISTANCES=[1], ANGLES=[0], LEVELS=256,
SYMMETRIC=TRUE, NORMED=TRUE)
    CONTRAST = GRAYCOPROPS(GLCM, 'CONTRAST')[0, 0]
    HOMOGENEITY = GRAYCOPROPS(GLCM, 'HOMOGENEITY')[0, 0]
    ENERGY = GRAYCOPROPS(GLCM, 'ENERGY')[0, 0]

    AVG_CIRCULARITY = NP.MEAN(CIRCULARITY_LIST) IF CIRCULARITY_LIST ELSE 1.0
    AVG_SOLIDITY = NP.MEAN(SOLIDITY_LIST) IF SOLIDITY_LIST ELSE 1.0
    AVG_EXTENT = NP.MEAN(EXTENT_LIST) IF EXTENT_LIST ELSE 1.0

    FEATURES.APPEND([
      SUSPICIOUS_COUNT,
      SUSPICIOUS_AREA,
      AVG_CIRCULARITY,
      AVG_SOLIDITY,
```

```
        AVG_EXTENT,
        CONTRAST,
        HOMOGENEITY,
        ENERGY
    ])
    LABELS.APPEND(Y_BIN[I])

  RETURN NP.ARRAY(FEATURES), NP.ARRAY(LABELS)

# ========== FEATURE EXTRACTION FOR TRAINING, VALIDATION, AND TEST ==========
FEATURES_TRAIN, LABELS_TRAIN = EXTRACT_FEATURES(X_TRAIN, Y_TRAIN_BIN)
FEATURES_VALID, LABELS_VALID = EXTRACT_FEATURES(X_VALID, Y_VALID_BIN)
FEATURES_TEST, LABELS_TEST = EXTRACT_FEATURES(X_TEST, Y_TEST_BIN)

# ========== TRAIN RANDOM FOREST (BEFORE PREPROCESSING) ==========
RF_RAW = RANDOMFORESTCLASSIFIER(N_ESTIMATORS=200, MAX_DEPTH=10, RANDOM_STATE=42)
RF_RAW.FIT(X_TRAIN.RESHAPE(LEN(X_TRAIN), -1), Y_TRAIN_BIN)

# ========== EVALUATE BEFORE PREPROCESSING (RAW DATA) ==========
RAW_PREDS = RF_RAW.PREDICT(X_TEST.RESHAPE(LEN(X_TEST), -1))
RAW_ACCURACY = ACCURACY_SCORE(Y_TEST_BIN, RAW_PREDS)

PRINT(F"TEST ACCURACY (BEFORE PREPROCESSING): {RAW_ACCURACY * 100:.2F}%")

# ========== CONFUSION MATRIX (BEFORE PREPROCESSING) ==========
CM_RAW = CONFUSION_MATRIX(Y_TEST_BIN, RAW_PREDS)
DISP_RAW = CONFUSIONMATRIXDISPLAY(CONFUSION_MATRIX=CM_RAW,
DISPLAY_LABELS=RF_RAW.CLASSES_)
DISP_RAW.PLOT(CMAP='BLUES')
PLT.TITLE("CONFUSION MATRIX (BEFORE PREPROCESSING) - TEST")
PLT.SHOW()
# ========== TRAIN RANDOM FOREST (AFTER PREPROCESSING) ==========
RF_MODEL = RANDOMFORESTCLASSIFIER(RANDOM_STATE=42)
RF_MODEL.FIT(FEATURES_TRAIN, LABELS_TRAIN)


# ========== EVALUATE AFTER PREPROCESSING (IMAGE PROCESSING) ==========
RF_PREDS = RF_MODEL.PREDICT(FEATURES_TEST)
ACCURACY = ACCURACY_SCORE(LABELS_TEST, RF_PREDS)
PRINT(F"\NTEST ACCURACY (AFTER PREPROCESSING): {ACCURACY * 100:.2F}%")


# ========== CONFUSION MATRIX (AFTER PREPROCESSING) ==========
CM_PROCESSED = CONFUSION_MATRIX(LABELS_TEST, RF_PREDS)  # USING THE PREPROCESSED
PREDICTIONS
```

```
DISP_PROCESSED = CONFUSIONMATRIXDISPLAY(CONFUSION_MATRIX=CM_PROCESSED,
DISPLAY_LABELS=RF_MODEL.CLASSES_)
DISP_PROCESSED.PLOT(CMAP='BLUES')
PLT.TITLE("CONFUSION MATRIX (AFTER PREPROCESSING) - TEST")
PLT.SHOW()

# ========== CLASSIFICATION REPORT ==========
PRINT(CLASSIFICATION_REPORT(LABELS_TEST, RF_PREDS, TARGET_NAMES=['NORMAL', 'CANCER']))
TRY:
   START = INT(INPUT("ENTER START INDEX: "))  # E.G., 0
   END = INT(INPUT("ENTER END INDEX (INCLUSIVE): ")) + 1  # E.G., 10 (INCLUSIVE)
EXCEPT VALUEERROR:
   PRINT("INVALID INPUT. PLEASE ENTER INTEGERS.")
   EXIT()

# INPUT VALIDATION TO ENSURE THE START AND END ARE WITHIN THE BOUNDS OF X_TEST
IF START < 0 OR END >= LEN(X_TEST):
   PRINT("INVALID RANGE! ENSURE THE START AND END VALUES ARE WITHIN THE BOUNDS OF THE
DATASET.")
   EXIT()

LABEL_MAP = {0: "NORMAL", 1: "CANCER"}

# PREDICT AND VISUALIZE THE SELECTED RANGE OF TESTING IMAGES
FOR I IN RANGE(START, MIN(END, LEN(X_TEST))):
   IMG = X_TEST[I].ASTYPE(NP.UINT8)
   GRAY = CV2.CVTCOLOR(IMG, CV2.COLOR_RGB2GRAY)
   BLURRED = CV2.GAUSSIANBLUR(GRAY, (5, 5), 0)
   EQUALIZED = CV2.EQUALIZEHIST(BLURRED)
   _, THRESH = CV2.THRESHOLD(EQUALIZED, 0, 255, CV2.THRESH_BINARY + CV2.THRESH_OTSU)
   MORPHED = CV2.MORPHOLOGYEX(THRESH, CV2.MORPH_CLOSE, NP.ONES((3, 3), NP.UINT8),
ITERATIONS=2)

   MORPHED_COLORED = CV2.CVTCOLOR(MORPHED, CV2.COLOR_GRAY2BGR)
   CONTOURS, _ = CV2.FINDCONTOURS(MORPHED, CV2.RETR_EXTERNAL,
CV2.CHAIN_APPROX_SIMPLE)
   FOR CNT IN CONTOURS:
      IF CV2.CONTOURAREA(CNT) > 1000:
         X, Y, W, H = CV2.BOUNDINGRECT(CNT)
         CV2.RECTANGLE(MORPHED_COLORED, (X, Y), (X + W, Y + H), (0, 0, 255), 2)

   GT = Y_TEST_BIN[I]  # CHANGED FROM LABELS_VALID TO Y_TEST_BIN
   PRED = RF_MODEL.PREDICT([FEATURES_TEST[I]])[0]  # CHANGED FROM FEATURES_VALID TO
FEATURES_TEST
   CORRECT = "CORRECT" IF GT == PRED ELSE "WRONG"
```

```
    PRINT(F"TEST SAMPLE #{I} | GT: {LABEL_MAP[GT]} | PRED: {LABEL_MAP[PRED]} {CORRECT}")

    PLT.FIGURE(FIGSIZE=(15, 3))
    PLT.SUPTITLE(F"TEST SAMPLE #{I} | GT: {LABEL_MAP[GT]} | PRED: {LABEL_MAP[PRED]}
{CORRECT}", FONTSIZE=14)
    PLT.SUBPLOT(1, 5, 1)
    PLT.IMSHOW(IMG)
    PLT.TITLE("ORIGINAL")
    PLT.AXIS('OFF')

    PLT.SUBPLOT(1, 5, 2)
    PLT.IMSHOW(GRAY, CMAP='GRAY')
    PLT.TITLE("GRAYSCALE")
    PLT.AXIS('OFF')

    PLT.SUBPLOT(1, 5, 3)
    PLT.IMSHOW(EQUALIZED, CMAP='GRAY')
    PLT.TITLE("EQUALIZED")
    PLT.AXIS('OFF')

    PLT.SUBPLOT(1, 5, 4)
    PLT.IMSHOW(THRESH, CMAP='GRAY')
    PLT.TITLE("THRESHOLD")
    PLT.AXIS('OFF')

    PLT.SUBPLOT(1, 5, 5)
    PLT.IMSHOW(MORPHED_COLORED)
    PLT.TITLE("CONTOURS")
    PLT.AXIS('OFF')

    PLT.TIGHT_LAYOUT()
    PLT.SHOW()
```