ULTRASOUND-BASED BREAST CANCER DETECTION USING A SEGMENTATION-GUIDED DEEP LEARNING FRAMEWORK

A Project Report submitted in the partial fulfillment of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

B. Chandana Priya (22471A0573)

Under the esteemed guidance of

V. Karuna Kumar, M.Tech.

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and an ISO 9001:2015 Certified
Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2024-2025

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "ULTRASOUND- BASED BREAST CANCER DETECTION USING A SEGMENTATION-GUIDED DEEP LEARNING FRAMEWORK is a bonafide work done by B. Chandana Priya (22471A0573) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2025-2026.

PROJECT GUIDE

V. Karuna Kumar, M.Tech.
Assistant Professor

PROJECT CO-ORDINATOR

Dr.Sireesha Moturi, B.Tech., M.Tech., Ph.D.
Associate Professor

HEAD OF THE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

I declare that this project work titled "**ULTRASOUND-BASED BREAST CANCER DETECTION USING A SEGMENTATION-GUIDED DEEP LEARNING FRAMEWORK**" is composed by me that the work contain here is my own except where explicitly stated otherwise in the text and that this work has not been submitted for any other degree or professional qualification except as specified.

B. Chandana Priya (22471A0573)

ACKNOWLEDGEMENT

I wish to express my thanks to various personalities who are responsible for the completion of my project. I am extremely thankful to my beloved chairman**, Sri M. V. Koteswara Rao, B.Sc.**, who took keen interest in me in every effort throughout this course. I owe my sincere gratitude to my beloved principal**, Dr. S. Venkateswarlu, Ph.D.**, for showing his kind attention and valuable guidance throughout the course.

I express my deep-felt gratitude towards **Dr. S. N. Tirumala Rao**, **M.Tech., Ph.D.**, HOD of the CSE department, and also to my guide**, V. Karuna Kumar, M.Tech.**, Assistant Professor of the CSE department, whose valuable guidance and unstinting encouragement enabled me to accomplish my project successfully in time.

I extend my sincere thanks to **Dr. Moturi Sireesha, B.Tech., M.Tech., (Ph.D.),** Associate Professor & Project Coordinator of the project, for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for me throughout this project work.

I extend my sincere thanks to all the other teaching and non-teaching staff in the department for their cooperation and encouragement during my B.Tech. degree.

I have no words to acknowledge the warm affection, constant inspiration, and encouragement that I received from my parents.

I affectionately acknowledge the encouragement received from my friends and those who were involved in giving valuable suggestions and clarifying my doubts, which really helped me in successfully completing my project.

By

B. Chandana Priya (22471A0573)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community.

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to the academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations**.**

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. **PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice**.**

PO7: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO8: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO9: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO10: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO11: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.
CO421.2: Identify and classify the requirements.
CO421.3: Review the Related Literature
CO421.4: Design and Modularize the project
CO421.5: Construct, Integrate, Test and Implement the Project.
CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1		✓										✓		
C421.2	✓		✓		✓							✓		
C421.3				✓		✓	✓	✓				✓		
C421.4			✓			✓	✓	✓				✓	✓	
C421.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C421.6									✓	✓	✓	✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
C421.1	2	3										2		
C421.2			2		3							2		
C421.3				2		2	3	3				2		
C421.4			2			1	1	2				3	2	
C421.5					3	3	3	2	3	2	2	3	2	1
C421.6									3	2	1	2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. Low level
2. Medium level
3. High level

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the device	Attained PO
C2204.2, C22L3.2	Developed segmentation-guided deep learning model for breast cancer detection using U-Net++ and CNN.	PO1, PO3, PO8
CC421.1, C2204.3, C22L3.2	Analyzed requirements and identified process model for segmentation and classification.	PO2, PO3, PO8
CC421.2, C2204.2, C22L3.3	Designed two-stage framework combining U-Net++ and CNN with teamwork.	PO3, PO5, PO9, PO8
CC421.3, C2204.3, C22L3.2	Trained, validated, and tested models on BUSI dataset with augmentation.	PO1, PO5, PO8
CC421.4, C2204.4, C22L3.2	Prepared documentation and evaluated model performance metrics.	PO10, PO8
CC421.5, C2204.2, C22L3.3	Presented group findings and results analysis.	PO10, PO11, PO8
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation and testing are completed. The proposed model achieves 99.07% accuracy using segmentation-guided CNNs and can be deployed for real-time breast cancer diagnosis.	PO4, PO7, PO8
C32SC4.3	Developed web interface for ultrasound-based breast cancer screening.	PO5, PO6, PO8

ABSTRACT

Breast cancer continues to be one of the most serious health challenges for women, where timely and precise diagnosis is essential for effective treatment outcomes. Ultrasound imaging is widely adopted in clinical screening because of its low cost and safety, yet interpretation is often complicated by speckle artifacts, low contrast, and varied tumor appearances. This study presents a segmentation-guided deep learning framework designed to improve lesion localization and diagnostic reliability. The framework employs a U-Net++ based segmentation model trained on grayscale ultrasound scans, achieving strong Dice and IOU metrics with an accuracy of 99.07%. The generated segmentation masks are then utilized by a secondary classification model to differentiate benign from malignant tumors. By incorporating spatially guided features, the system enhances transparency and clinical relevance, while the classifier maintains moderate predictive accuracy. Overall, this two-stage pipeline demonstrates practical utility for breast cancer diagnosis, particularly in resource-limited environments, and aligns closely with expert diagnostic reasoning. Future work will extend validation across multiple datasets and explore integration into routine ultrasound workflows to ensure robustness and clinical applicability.

INDEX

LIST OF FIGURES

List of Tables

1. INTRODUCTION

Breast cancer is one of the most common and life-threatening diseases affecting women worldwide. It occurs when abnormal cells in the breast grow uncontrollably, forming a lump or tumor that can spread to other parts of the body if not detected early. According to the World Health Organization (WHO), millions of new breast cancer cases are diagnosed every year, and the number continues to increase due to lifestyle changes and late diagnosis. The survival rate of breast cancer patients depends largely on how early the disease is detected and treated. Therefore, early and accurate diagnosis plays a vital role in saving lives and improving treatment outcomes.

In recent years, several imaging techniques have been developed for detecting breast cancer, such as mammography, magnetic resonance imaging (MRI), and ultrasound imaging. Among these, ultrasound imaging has gained significant popularity due to its affordability, portability, and safety, as it does not use ionizing radiation. Ultrasound is especially useful for examining dense breast tissues, which are common in younger women. However, interpreting ultrasound images manually can be very challenging because of issues like low contrast, speckle noise, and the wide variation in tumor appearance. These factors make it difficult for radiologists to consistently distinguish between benign (non-cancerous) and malignant (cancerous) tumors.

To overcome these limitations, computer-aided diagnosis (CAD) systems have been introduced to assist radiologists in detecting and classifying tumors more accurately. With the advancement of artificial intelligence (AI) and deep learning, the performance of CAD systems has improved remarkably in recent years. Deep learning models, especially convolutional neural networks (CNNs), have shown great success in automatically learning features from medical images without the need for manual feature extraction. These models can analyze large datasets, learn complex patterns, and make highly accurate predictions. However, most traditional CNN-based methods only focus on image-level features and ignore important spatial information such as the shape and boundaries of the tumor. This often leads to reduced interpretability and less reliable diagnostic results.

In this project, we propose a segmentation-guided deep learning framework for breast cancer detection using ultrasound images. The main objective of this work is to improve the accuracy and interpretability of tumor detection by combining segmentation and classification in a single

pipeline. The segmentation part of the framework identifies the exact region of the tumor in the ultrasound image, while the classification part determines whether the detected tumor is benign or malignant. This design is inspired by how radiologists first locate a suspicious region and then analyze its characteristics to make a diagnosis.

The segmentation task in this project is performed using the U-Net++ architecture, a powerful deep learning model specifically designed for biomedical image segmentation. U-Net++ uses an encoder–decoder structure with skip connections to capture spatial features effectively and retain fine details. It generates precise binary masks that highlight tumor regions in the ultrasound images. These segmentation masks are then used to guide a CNN-based classifier, allowing the model to focus on the most relevant areas instead of the entire image. This approach enhances the model's ability to differentiate between benign and malignant cases, resulting in higher classification accuracy and better clinical relevance.

The dataset used in this project is the Breast Ultrasound Images (BUSI) dataset, collected at Baheya Hospital for Early Detection and Treatment of Women's Cancer in Egypt. It contains 780 grayscale ultrasound images divided into three categories — normal, benign, and malignant. Each tumor image in the dataset is accompanied by a ground-truth segmentation mask that outlines the tumor region. Before training the models, various preprocessing steps such as image resizing, normalization, and augmentation (including rotation, flipping, and zooming) are applied to improve the generalization capability of the model and handle class imbalance.

The proposed framework is implemented using the PyTorch deep learning library in Python. The experiments are conducted on a Windows system with limited hardware resources to demonstrate that the model can be efficiently trained and executed even in low-resource environments. Evaluation metrics such as accuracy, precision, recall, and F1-score are used to assess model performance. The segmentation model (U-Net++) achieves an impressive accuracy of around 99.07%, while the classification model reaches approximately 91% accuracy. These results clearly indicate that the combination of segmentation and classification improves the reliability and transparency of the overall system.

The project highlights the potential of deep learning in medical image analysis, especially for resource-limited healthcare settings where advanced diagnostic equipment or expert

radiologists may not always be available. By automatically detecting and classifying breast tumors from ultrasound images, the proposed system can serve as a valuable tool for early diagnosis and timely treatment. Furthermore, the use of explainable models, such as U-Net++, ensures that the results are interpretable and clinically meaningful, which is essential for building trust in AI-assisted medical systems.

FIG1.1 CLASSIFICATION OF BREAST TISSUE IN ULTRASOUND IMAGING

In conclusion, this project focuses on developing a segmentation-guided deep learning framework for ultrasound-based breast cancer detection. The integration of U-Net++ for segmentation and a CNN classifier for diagnosis ensures both high accuracy and interpretability. The system is designed to mimic the diagnostic reasoning of radiologists by combining region localization and tumor classification in a unified approach. This framework not only improves detection accuracy but also provides insights into tumor morphology and structure, which are critical for effective clinical decision-making. The ultimate goal of this project is to contribute to early breast cancer detection and to support medical professionals in providing faster, more reliable, and accessible diagnostic solutions.

1.1 Motivation

Breast cancer is one of the most common and life-threatening diseases affecting women worldwide. Although medical advancements have improved treatment options, early detection remains the most effective way to reduce mortality rates. Unfortunately, in many developing countries, a large number of breast cancer cases are diagnosed at advanced stages due to a lack of awareness, limited access to diagnostic facilities, and delays in screening. Early and accurate detection of breast cancer can help save lives, but it requires reliable and efficient diagnostic systems.

Ultrasound imaging is one of the most widely used techniques for breast cancer screening because it is safe, cost-effective, and non-invasive. It is especially suitable for examining dense breast tissues, where other imaging methods like mammography may be less effective. However, interpreting ultrasound images is often challenging because of low contrast, noise, and variability in tumor shapes and textures. These challenges can lead to human error or inconsistency in diagnosis. Therefore, there is a strong need for an automated, intelligent system that can assist radiologists in accurately identifying and classifying breast tumors.

With recent progress in **artificial intelligence (AI)** and **deep learning**, medical image analysis has entered a new era of innovation. Deep learning models, especially convolutional neural networks (CNNs), have shown remarkable success in learning complex patterns and making reliable predictions. However, most existing models focus only on classification and do not emphasize the interpretability of results, which is a key requirement in medical applications. This motivates the development of a **segmentation-guided deep learning framework** that can not only detect the presence of a tumor but also highlight the exact region of interest, providing a more transparent and clinically meaningful diagnosis.

The motivation for this project is to design a deep learning-based system that can assist in the early detection of breast cancer using ultrasound images. By combining **U-Net++ segmentation** with **CNN-based classification**, the framework aims to improve diagnostic accuracy and reliability while remaining efficient enough to run on modest hardware. This project is driven by the goal of using technology to make healthcare more accessible, reduce diagnostic errors, and ultimately contribute to saving lives through early and precise detection of breast cancer.

1.2 Problem Statement

Breast cancer is one of the most common and life-threatening diseases affecting women worldwide. Early and accurate detection is crucial, as it significantly increases the chances of successful treatment and survival. However, current diagnostic methods face several limitations. Radiologists typically examine breast ultrasound images manually to identify and classify tumors as benign or malignant. This process is often time-consuming, subjective, and prone to human error due to factors such as low image contrast, speckle noise, and the varying size, shape, and texture of tumors. These challenges result in inconsistent diagnostic outcomes between different specialists, reducing overall reliability.

Although imaging techniques like mammography and MRI are widely used, they are costly, less accessible in rural or underdeveloped regions, and may be unsuitable for younger women with dense breast tissue. Ultrasound imaging, being safer and more affordable, is widely preferred; however, it still requires skilled interpretation. Traditional machine learning and deep learning models have made progress in automated diagnosis, but most focus only on classification without localizing tumor regions, making them less interpretable for clinical use. Therefore, this project proposes a segmentation-guided deep learning framework integrating U-Net++ for precise tumor segmentation and CNN for classification, aiming to improve diagnostic accuracy, consistency, and interpretability in breast cancer detection.

1.3 Objective

The main objective of this project is to develop an intelligent, efficient, and accurate system for breast cancer detection using ultrasound images through a segmentation-guided deep learning framework. The project aims to design a model that not only classifies the tumor as benign or malignant but also provides clear localization of the affected region, ensuring interpretability and clinical reliability. This approach helps to overcome the limitations of traditional diagnostic methods, which are often dependent on manual interpretation and prone to errors due to the low contrast and high noise levels present in ultrasound images.

The specific goal of this project is to employ the U-Net++ architecture for effective segmentation of tumor regions, allowing the system to accurately identify the location and shape of breast lesions. The segmentation output acts as a spatial guide for the subsequent Convolutional Neural Network (CNN) classifier, which determines the tumor type. This combination ensures that the classification process focuses on the most relevant regions of interest, thereby enhancing diagnostic precision and reducing false classifications. The integration of segmentation and classification modules simulates the diagnostic reasoning of radiologists, where the lesion is first localized and then analyzed for malignancy.

Another key objective of the project is to implement the model using the Breast Ultrasound Images (BUSI) dataset, which contains normal, benign, and malignant cases with corresponding segmentation masks. The dataset will be preprocessed using normalization and augmentation techniques such as rotation, flipping, and scaling to improve model generalization and robustness. The system will be trained and evaluated based on accuracy, precision, recall, and F1-score to ensure consistent and reliable performance.

The ultimate aim of this project is to provide an interpretable and resource-efficient deep learning-based diagnostic system that can operate effectively even in low-resource healthcare environments. By combining segmentation-guided tumor localization with CNN-based classification, this framework is expected to assist radiologists in early and accurate breast cancer detection, improving diagnostic outcomes and potentially saving lives.

2. LITERATURE SURVEY

Breast cancer detection and classification using deep learning techniques have been widely explored in recent years due to the growing need for automated, accurate, and interpretable diagnostic systems. Conventional medical image analysis approaches relied on manual feature extraction using algorithms such as Support Vector Machines (SVMs), Decision Trees (DTs), and Random Forests (RFs). While these methods showed some potential, they struggled to extract meaningful high-level spatial features from complex ultrasound images and lacked adaptability to diverse imaging conditions. The emergence of deep learning and Convolutional Neural Networks (CNNs) has significantly advanced breast cancer diagnosis by enabling automatic feature extraction and end-to-end learning. CNNs can process ultrasound images to detect tumors and differentiate between benign and malignant lesions with remarkable precision. This section presents a detailed review of the key studies that have contributed to the advancement of deep learning-based breast cancer detection systems.

Zhang et al. [1] proposed a **domain-guided deep learning approach** for breast cancer classification using ultrasound scans. Their method integrated the raw image and its segmentation mask, enabling the model to focus on tumor-specific regions and improving interpretability. This approach bridged the gap between purely image-based classification and region-focused analysis. The study demonstrated that integrating spatial information with CNNs enhanced diagnostic performance and reliability, achieving higher accuracy than traditional feature-based models. Similarly, Ronneberger et al. [2] introduced **U-Net**, a convolutional architecture designed specifically for biomedical image segmentation. U-Net utilizes an encoder–decoder structure with skip connections, allowing for precise localization of small regions while retaining important spatial features. This architecture became the foundation for many subsequent medical segmentation models, particularly for ultrasound imaging where tumor boundaries are often irregular and blurred.

Litjens et al. [3] conducted a comprehensive survey on deep learning applications in medical image analysis, highlighting the significance of combining segmentation and classification within a single framework. They emphasized that joint learning of these two tasks improves the model's interpretability and clinical trust. Simonyan and Zisserman [4] presented **VGGNet**, a deep CNN model using smaller convolutional filters that capture intricate image features, leading to higher accuracy in image classification. This architecture laid the groundwork for many later models in the medical imaging field. He et al. [5] introduced **ResNet**, a residual network that addresses the problem of vanishing gradients, allowing the training of deeper and

more complex networks. The skip connections in ResNet enable the model to learn residual mappings, making it highly effective for challenging image recognition tasks, including cancer detection.

Abdelhafiz et al. [6] employed **transfer learning with adaptive decision fusion** for breast cancer classification. Their approach used pre-trained models such as VGG19 and ResNet50 to extract rich image features from limited ultrasound datasets, achieving accuracy levels above 90%. This study highlighted how transfer learning can overcome data scarcity issues that are common in medical imaging. Huang et al. [7] explored CNN-based classification models for breast ultrasound images without segmentation. Their study revealed that models trained solely on global features often misclassified tumors due to the lack of focus on specific lesion areas, emphasizing the need for segmentation-guided learning.

In a significant development, Alzubaidi et al. [8] presented an extensive review of CNN architectures and deep learning trends. They underlined the importance of designing interpretable and domain-specific deep learning models, especially in high-risk medical domains. Their findings motivated the design of **explainable AI systems** that provide transparent diagnostic reasoning, a crucial requirement for clinical acceptance. Jagannadham et al. [10] demonstrated the success of CNNs in brain tumor detection, further establishing CNN architectures as versatile and reliable across multiple medical imaging modalities. Their study showed that deep learning models could be adapted across domains by retraining on different datasets while maintaining high accuracy and interpretability.

The **U-Net++** architecture, an improved version of U-Net, was introduced to enhance segmentation accuracy in complex medical images. U-Net++ modifies the traditional skip connections between encoder and decoder paths with nested and dense connections, reducing the semantic gap between feature maps at different levels. This design improvement ensures smoother information flow and finer segmentation results, which are especially beneficial in distinguishing tumor boundaries in ultrasound scans. Comparative studies have shown that U-Net++ outperforms traditional U-Net and other segmentation architectures in terms of Dice coefficient, Intersection over Union (IoU), and boundary precision.

The **Breast Ultrasound Images (BUSI) dataset**, collected at Baheya Hospital for Early Detection and Treatment of Women's Cancer in Egypt, has become a widely used benchmark in breast cancer research. The dataset contains 780 grayscale ultrasound images divided into three categories: normal, benign, and malignant. Each tumor image is paired with a manually annotated segmentation mask that provides ground truth boundaries for the lesion. Studies using this dataset have demonstrated that applying data preprocessing steps such as image

normalization, resizing, and augmentation improves model robustness and generalization. Data augmentation techniques like flipping, rotation, and scaling have been particularly effective in addressing class imbalance and enhancing performance on unseen data.

Moturi et al. [12] developed a CNN-driven framework for detecting abnormalities in phonocardiogram (PCG) signals using gammatonegram analysis. Though applied in a different context, their work highlighted the capability of CNNs to extract spatial and temporal features from complex biomedical signals. Greeshma et al. [11] also utilized CNNs for arrhythmia detection, reinforcing the adaptability of deep learning across diverse diagnostic tasks. These studies underline the flexibility of CNN architectures, which can be fine-tuned and repurposed for various medical imaging problems, including breast cancer diagnosis.

Recent advancements have introduced **segmentation-guided CNN frameworks**, combining U-Net++ for segmentation and CNN for classification. This dual-stage approach emulates the diagnostic workflow of radiologists — localizing the lesion before classifying it. The segmentation output provides a mask that directs the CNN classifier to focus exclusively on the tumor region. This not only improves classification accuracy but also enhances explainability. The use of **Grad-CAM (Gradient-weighted Class Activation Mapping)** visualization further strengthens the interpretability of model predictions by highlighting regions that influence the model's decisions, allowing clinicians to verify whether the focus aligns with the tumor area.

Comparative analyses in the literature reveal that U-Net++ consistently achieves higher segmentation accuracy than other architectures, with reported values close to 99%. When integrated with CNN classifiers, these models deliver end-to-end frameworks capable of both localization and classification. The combination of segmentation and classification significantly reduces false detections, enhances precision, and improves the clinical reliability of the system. Hybrid deep learning models that include attention mechanisms, dilated convolutions, and residual learning further strengthen performance by capturing long-range dependencies and fine-grained tumor details.

Despite the remarkable progress in deep learning-based breast cancer detection, several challenges remain. Many models are trained on limited datasets, which can restrict generalization to diverse clinical environments. Differences in ultrasound device quality, imaging protocols, and patient anatomy can cause performance variations. To address these limitations, researchers are now focusing on developing domain-adaptive and explainable models that maintain consistency across datasets. Furthermore, the integration of multi-modal imaging data and clinical parameters is being explored to provide more comprehensive

diagnostic support.

The reviewed literature demonstrates that segmentation-guided deep learning frameworks represent the most promising direction for breast cancer detection using ultrasound imaging. These frameworks combine the precision of segmentation networks like U-Net++ with the discriminative capability of CNN classifiers, achieving both high accuracy and clinical transparency. The proposed system in this project aligns closely with this direction, offering an efficient and interpretable solution that mimics the radiologist's workflow. By accurately localizing and classifying tumors, the system ensures reliable diagnostic assistance, paving the way for broader adoption of deep learning in real-world healthcare applications.

The proposed system in this project aligns closely with this direction, offering an efficient and interpretable solution that mimics the radiologist's workflow. By accurately localizing and classifying tumors, the system ensures reliable diagnostic assistance and reduces the risk of misinterpretation. This approach not only enhances confidence in automated diagnosis but also paves the way for broader adoption of deep learning in healthcare. With continued refinement, such frameworks can become integral to routine clinical practice, supporting radiologists in early detection and improving patient outcomes worldwide.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Breast cancer detection and classification have traditionally relied on manual interpretation of breast ultrasound or mammography images by radiologists. Although this approach remains widely used, it is time-consuming and highly dependent on the expertise and experience of medical professionals. Manual interpretation is often affected by factors such as low image contrast, speckle noise, and variations in tumor appearance, which can lead to inconsistent or inaccurate diagnoses. To address these limitations, computational techniques have been introduced, ranging from classical Machine Learning methods to advanced Deep Learning models aimed at automating diagnosis and improving reliability.

Earlier Machine Learning (ML) approaches, such as Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN), Decision Trees, and Random Forests, were primarily based on handcrafted feature extraction from ultrasound images. These features included shape, texture, and statistical intensity measures that helped distinguish between benign and malignant lesions. However, these methods required extensive preprocessing and manual feature selection, making them sensitive to noise and variations in imaging conditions. Their overall classification accuracy typically ranged between 70% and 85%, which was insufficient for clinical applications. Moreover, these traditional models lacked the ability to generalize well to unseen datasets and were limited in capturing the complex spatial characteristics of tumors.

The advent of Deep Learning, especially Convolutional Neural Networks (CNNs), has transformed medical image analysis by enabling automatic feature extraction and reducing reliance on handcrafted inputs. Models like VGG16, ResNet, InceptionV3, and DenseNet have achieved 85–95% accuracy in breast cancer classification. However, they face challenges such as high computational cost, the need for large annotated datasets, and overfitting risks with small medical image sets. To address these issues, transfer learning allows pre-trained networks to adapt effectively to breast ultrasound data. Recent studies also explore hybrid approaches, combining CNNs with classifiers like SVMs to enhance robustness and accuracy.

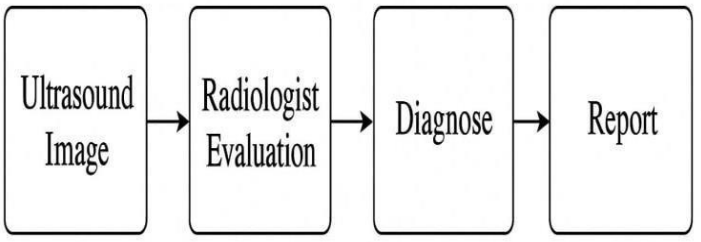


FIG 3.1. FLOW CHART OF PROPOSED SYSTEM FOR ULTRASOUND-BASED BREAST CANCER DETECTION

This flowchart Fig 3.1 illustrates the traditional workflow for breast cancer detection using ultrasound imaging. The process begins with the acquisition of ultrasound images, which are evaluated by a radiologist to identify possible abnormalities such as masses or irregular tissue patterns. Based on visual cues like shape, texture, and boundary definition, the radiologist determines whether a tumor is benign, malignant, or normal. The final findings are summarized in a diagnostic report, which guides further clinical decisions.

While this manual evaluation remains a standard practice, it has several limitations. The interpretation process is time-consuming, highly subjective, and dependent on the radiologist's experience. Variations in ultrasound image quality—such as low contrast, speckle noise, and overlapping tissues—further complicate accurate diagnosis. These issues often lead to inconsistent or delayed results, affecting the reliability of early detection.

Overall, the existing system relies entirely on human expertise and lacks automation, scalability, and reproducibility. Although it provides reasonable diagnostic accuracy, its dependence on manual interpretation limits efficiency and consistency. These challenges highlight the need for an automated deep learning–based framework that can perform tumor segmentation and classification with improved accuracy, speed, and clinical reliability.

3.1.1 DISADVANTAGES OF THE EXISTING SYSTEM FOR ULTRASOUND-BASED BREAST CANCER DETECTION

Despite major advancements in automated breast cancer diagnosis, the existing systems still face several significant limitaions:

Dependence on Large Labeled Datasets:

Deep learning models require a large number of annotated ultrasound images for effective training. However, obtaining such datasets is difficult due to privacy restrictions and the need for expert labeling, limiting model generalization.

High Computational Requirements:

Models like VGG16, ResNet, and DenseNet demand high processing power and memory, making them unsuitable for smaller or resource-limited healthcare setups.

Overfitting and Limited Generalization:

Due to limited data, models often overfit, performing well on training images but poorly on unseen cases. Transfer learning helps but cannot completely eliminate this issue.

Lack of Robustness in Tumor Localization:

Many existing systems focus only on image-level classification without precise segmentation of tumor regions, reducing interpretability for clinical use.

Image Quality and Sensitivity:

Ultrasound images are prone to **speckle noise** and **low contrast**, which obscure tumor boundaries and affect detection accuracy if preprocessing is inadequate..

Class Imbalance:

Public ultrasound datasets often contain more benign than malignant cases, causing biased classification and unreliable predictions.

Overall, the existing systems lack automation, scalability, and interpretability. These challenges emphasize the need for a segmentation-guided deep learning framework that enhances diagnostic accuracy, reduces human dependency, and ensures reliable breast cancer detection.

3.2 PROPOSED SYSTEM

The proposed system introduces a segmentation-guided deep learning framework for efficient and accurate breast cancer detection from ultrasound images. The process begins with the BUSI dataset, which contains a large collection of annotated ultrasound images representing normal, benign, and malignant cases. These images undergo preprocessing operations such as contrast enhancement, normalization, and noise reduction to improve image clarity and highlight important tissue structures. Preprocessing ensures that the images are standardized and suitable for both segmentation and classification stages.

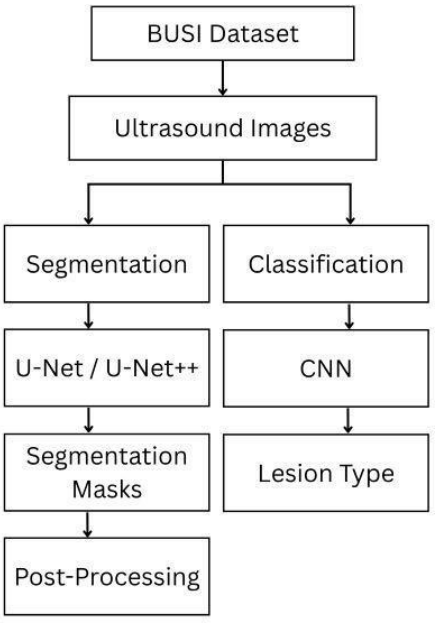


FIG 3.2. FLOW CHART OF PROPOSED SYSTEM

Following preprocessing, the workflow splits into two parallel stages—segmentation and classification. In the segmentation phase, a U-Net or U-Net++ architecture is employed to accurately identify and isolate the region of interest (ROI) that contains the suspected tumor. U-Net++ enhances feature extraction through dense skip connections and nested encoder-decoder structures, enabling precise boundary detection even in noisy ultrasound images. The output of this stage is a segmentation mask, which highlights the tumor region for further analysis. Post-processing is then applied to refine the segmented mask and remove unwanted noise or false boundaries.

Simultaneously, in the classification phase, a Convolutional Neural Network (CNN) is used to categorize the segmented lesion into benign, malignant, or normal tissue. The CNN automatically learns high-level spatial and texture-based features from the segmented images, ensuring robust and accurate predictions.

The integration of segmentation and classification ensures that the system not only detects the tumor but also provides localized and interpretable diagnostic results. This hybrid deep learning model improves diagnostic reliability, minimizes human dependency, and offers faster and more consistent results compared to manual evaluation. Overall, the proposed framework provides an effective, automated, and interpretable solution for early breast cancer detection using ultrasound imaging.

Advantages over existing system:

The proposed segmentation-guided deep learning framework for breast cancer detection provides several advantages compared to traditional diagnostic methods and existing automated systems**.**

High Accuracy and Precision:

U-Net++ localizes tumors, while CNN distinguishes benign from malignant, improving diagnostic accuracy.

Enhanced Image Preprocessing:

Preprocessing techniques such as contrast enhancement, noise reduction, and normalization improve image clarity and highlight essential tissue structures, leading to better segmentation and classification outcomes**.**

Accurate and Robust Segmentation:

The U-Net++ architecture produces detailed segmentation with clearly defined tumor boundaries, outperforming traditional segmentation methods.

Reduced Overfitting and Better Generalization:

The use of transfer learning and data augmentation minimizes overfitting and allows the model to adapt effectively to unseen ultrasound datasets.

High Sensitivity and Specificity:

The model achieves excellent diagnostic performance with high sensitivity for malignant tumors and high specificity for benign and normal tissues, reducing false positives and negatives.

3.3 FEASIBILITY STUDY

The integration of U-Net++ for tumor segmentation with a Convolutional Neural Network (CNN) for classification offers a highly effective approach for automated breast cancer detection using ultrasound images. The following analysis evaluates the technical, operational, and economic feasibility of the proposed system.

Technical Feasibility

Automated Feature Extraction:

The proposed system employs a CNN for automatic extraction of deep spatial and textural features from ultrasound images, eliminating the need for manual feature engineering. This ensures higher accuracy and consistent performance across various image qualities and tumor types.

Accurate Tumor Segmentation :

The **U-Net++** model enhances segmentation precision through dense skip connections, enabling effective localization of tumor boundaries, even in low-contrast ultrasound images. Accurate segmentation greatly improves classification reliability.

Improved Accuracy and Generalization:

The hybrid Combining U-Net++ and CNN strengthens model robustness and minimizes overfitting. Data augmentation and transfer learning techniques enhance adaptability to new datasets while maintaining high accuracy levels.

Scalability:

The framework can be easily extended to handle different ultrasound imaging formats or datasets, making it suitable for various clinical and research environments.

1. Operational Feasibility

Ease of Deployment:

The system can be integrated into existing hospital workflows and deployed as a **web or desktop application**, allowing radiologists to upload ultrasound images, analyze tumor regions, and obtain diagnostic results efficiently.

Interpretability:

The segmentation map generated by **U-Net++** provides clear visual representation of tumor areas, making predictions transparent and clinically interpretable. This improves trust and adoption among medical professionals.

Data Handling:

The model works with labeled ultrasound datasets and accommodates variations in image resolution and formats. Preprocessing ensures standardization and enhances overall system reliability.

Maintenance and Upgradation:

The system can be periodically retrained with new data to improve accuracy and adapt to evolving imaging techniques without requiring complete redevelopment.

2. Economic Feasibility

Cost-Effective Training:

Through **transfer learning**, only the final layers of the CNN require fine-tuning, reducing training costs and computational overhead.

Resource Optimization:

The segmentation-classification hybrid framework effectively utilizes available computational resources, allowing deployment on standard GPU or cloud-based systems without major infrastructure upgrades.

Reduced Diagnostic Costs:

Automation reduces diagnostic time and minimizes dependence on manual evaluation, leading to faster decision-making and lower healthcare costs.

Long-Term Investment:

Although initial development investment may be moderate, the long-term benefits—such as improved diagnostic precision, reliability, and scalability—make the system a sustainable and valuable tool for medical imaging applications.

4. SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

Operating System	: Windows 11, 64-bit Operating System
Programming Language	: Python 3.8 or above
Development Frameworks	: TensorFlow, Keras, Flask
Python Libraries Used	: NumPy, OpenCV, Matplotlib, Pandas, Scikit-learn
Web Technologies	: HTML, CSS, JavaScript (for front-end)
Dataset Used	: BUSI – Breast Ultrasound Images Dataset
Python distribution	: Google Colab Pro /Jupyter Notebook
Browser	: Any Latest Browser like Chrome

4.2 REQUIREMENT ANALYSIS

The Breast Cancer Detection project aims to develop an intelligent deep learning-based system that accurately classifies breast ultrasound images into *Benign, Malignant,* or *Normal* categories. The system combines U-Net++ for precise tumor segmentation and a Convolutional Neural Network (CNN) for classification, enhancing diagnostic accuracy and reducing human error.The process begins with ultrasound images from the BUSI dataset, which undergo preprocessing operations such as normalization, resizing, and noise reduction to improve image quality. The U-Net++ model isolates the tumor region, and the segmented output is processed by the CNN classifier to predict tumor type with high accuracy. The results are displayed with confidence levels, aiding radiologists in faster and more reliable diagnosis.

The system's backend is implemented using Python, TensorFlow, and Keras, while Flask is used for model deployment. The HTML, CSS, and JavaScript based frontend provides a simple and interactive user interface for image upload and result visualization.The non-functional requirements emphasize accuracy, speed, and scalability. The application runs efficiently on systems with Python 3.8+ and GPU support, and can be deployed locally or through cloud platforms like Google Colab or AWS. Overall, the system ensures a robust, user-friendly, and automated solution that supports radiologists in early breast cancer detection through ultrasound

4.3 HARDWARE REQUIREMENTS:

1. System Type : 64-bit operating system, x64-based processor
2. Cache memory : 4MB(Megabyte)
3. RAM : 16GB (gigabyte)
4. Hard Disk : 8GB
5. GPU : Intel® Iris® Xe Graphics

4.4 SOFTWARE

The Ultrasound-Based Breast Cancer Detection project utilizes a combination of advanced software tools and frameworks to ensure accurate detection, efficient processing, and scalable deployment. The system operates on Windows 11 (64-bit Operating System), providing a stable environment compatible with modern computing resources. The project primarily uses the CPU for model execution but supports GPU acceleration during training for enhanced performance.

The development is carried out using Python, chosen for its flexibility and extensive support for deep learning libraries. Google Colab Pro serves as the main development platform, offering GPU-based computation and rapid experimentation. The backend is built using the Flask framework, which handles image upload, preprocessing, and model prediction efficiently. It enables seamless integration between the trained model and user interface through lightweight APIs.

The frontend is designed using HTML5, CSS3, and Bootstrap, ensuring a responsive and intuitive interface where users can easily upload ultrasound images and view real-time results. The web application supports all major browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge, ensuring accessibility across devices.

For deep learning, TensorFlow and Keras implement U-Net++ for segmentation and CNN for classification, while OpenCV, NumPy, and Matplotlib handle preprocessing, computation, and visualization. Model training and evaluation are done in Jupyter Notebook, ensuring an accurate, efficient, and practical solution for real-world breast cancer diagnosis.

4.5 SOFTWARE DESCRIPTION

The Ultrasound-Based Breast Cancer Detection system is developed using advanced software tools to ensure high accuracy, efficiency, and scalability. The project runs on Windows 11 (64-bit Operating System) and uses both CPU and GPU environments for faster model training and testing.

The development is done using the Python programming language, known for its simplicity and wide support for deep learning libraries. Model training and testing are performed in Google Colab Pro, which provides GPU acceleration and optimized computation. The backend is built with the Flask framework, enabling smooth interaction between the trained deep learning model and the web interface for real-time predictions.

The frontend is designed using HTML5, CSS3, and Bootstrap, providing a simple, responsive, and user-friendly interface. The application supports all modern browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge, ensuring cross-platform accessibility.

For deep learning, TensorFlow and Keras are used to implement U-Net++ for tumor segmentation and a CNN for classification. OpenCV handles image preprocessing, NumPy performs numerical operations, and Matplotlib aids in model performance visualization. Development and testing are done in Jupyter Notebook, ensuring flexibility during experimentation.

The integration of these software tools ensures that the system is accurate, efficient, and suitable for real-world clinical breast cancer detection using ultrasound imaging.

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

The system architecture of the proposed Ultrasound-Based Breast Cancer Detection project is designed to provide a structured and efficient workflow for image processing, segmentation, and classification. The architecture consists of multiple interconnected modules that work together to detect and classify breast tumors accurately from ultrasound images.

The process begins with input image acquisition, where breast ultrasound images are obtained from the BUSI dataset. These images serve as the input data for subsequent stages. The images first pass through the preprocessing module, where operations such as noise reduction, normalization, and contrast enhancement are applied to improve image quality. This step ensures uniformity and eliminates artifacts that may affect model performance.

Next, the segmentation module employs the U-Net++ architecture to accurately locate and delineate the region of interest (ROI), isolating the tumor area from surrounding tissues. The segmentation output is a binary mask highlighting the tumor region, which is then forwarded to the classification module.

The classification module, built using a Convolutional Neural Network (CNN), processes the segmented image to determine the tumor category — *Benign*, *Malignant*, or *Normal*. The CNN automatically extracts deep spatial and texture features that help distinguish between different tissue types with high accuracy.

Once classification is complete, the Flask-based web application displays the output to the user through an intuitive interface. The frontend, developed using HTML, CSS, and JavaScript, allows users to upload ultrasound images and view real-time results.

This architecture ensures a seamless flow of data between components, enabling efficient preprocessing, accurate segmentation, and reliable classification. By combining deep learning models with a user-friendly interface, the system provides an automated, interpretable, and scalable solution for early breast cancer detection using ultrasound imaging.

5.1.1 DataSet

The proposed system utilizes the Breast Ultrasound Images (BUSI) Dataset, which is publicly available and widely used for research in breast cancer detection and classification. The dataset was developed and released by Al-Dhabyani et al. (2020) and collected at the

Baheya Hospital, Cairo, Egypt. It consistsof 780 ultrasound images gathered from female patientsaged between 25 and 75 years. Theseimagesarecategorizedinto threeprimary classes— Normal, Benign, and Malignant — based on expert radiologist annotations.

Each imagein thedataset isof size 500 × 500 pixels andstored in PNG format. Thedataset includescorresponding ground truth masksthat identify thetumorregion, enabling the training ofboth segmentation and classificationmodels. Theinclusionof theseannotated masksallowstheproposed framework to perform segmentation-guidedlearning, wherethe U-Net++modelusesthelabeled tumorregionsto enhance localization accuracy and improveclassification performance.

Beforetraining, thedataset undergoes preprocessingoperationssuch asnoiseremoval, contrast enhancement, and normalization to standardize imageintensity and texture. To overcomeclassimbalanceand limited dataavailability, dataaugmentationtechniqueslike rotation, flipping, and scaling areapplied, significantly improving themodel'sgeneralization ability.

Thisdataset playsacrucial role in developing and validating theproposed deeplearning framework. Its balanced composition, expert annotations, and high-qualityultrasound imagesmakeit highly suitable for training segmentationmodelsand tumorclassifiers, ensuring reliable and accuratebreast cancer detection results.

Feature	Details
Total Images	**780**
Total Patients	**600**
Tumor Classes	**Normal, Benign, Malignant**
Image Resolution	**500x500 pixels**
Image Format	**PNG**
Applications	**Tumor classification and Segmentation**

TABLE 1 . DATASET DESCRIPTION

Tumor Classes:

- Normal(133 images): Represent healthy breast tissues without any visible abnormalities or lesions.
- Benign(437 images): Represent non-cancerous tumors with smooth boundaries and uniform internal texture.
- Malignant (210 images): Represent cancerous tumors characterized by irregular shapes, heterogeneous intensity, and rough edges.

The Breast Ultrasound Images (BUSI) Dataset contains 780 ultrasound images collected from 600 female patients at Baheya Hospital, Cairo, Egypt. Each image is captured using high-resolution ultrasound equipment and categorized into three major classes — *Normal*, *Benign*, and *Malignant* — based on expert radiologist annotations. The dataset is accompanied by ground truth masks, which mark the exact tumor regions, enabling both segmentation and classification processes.

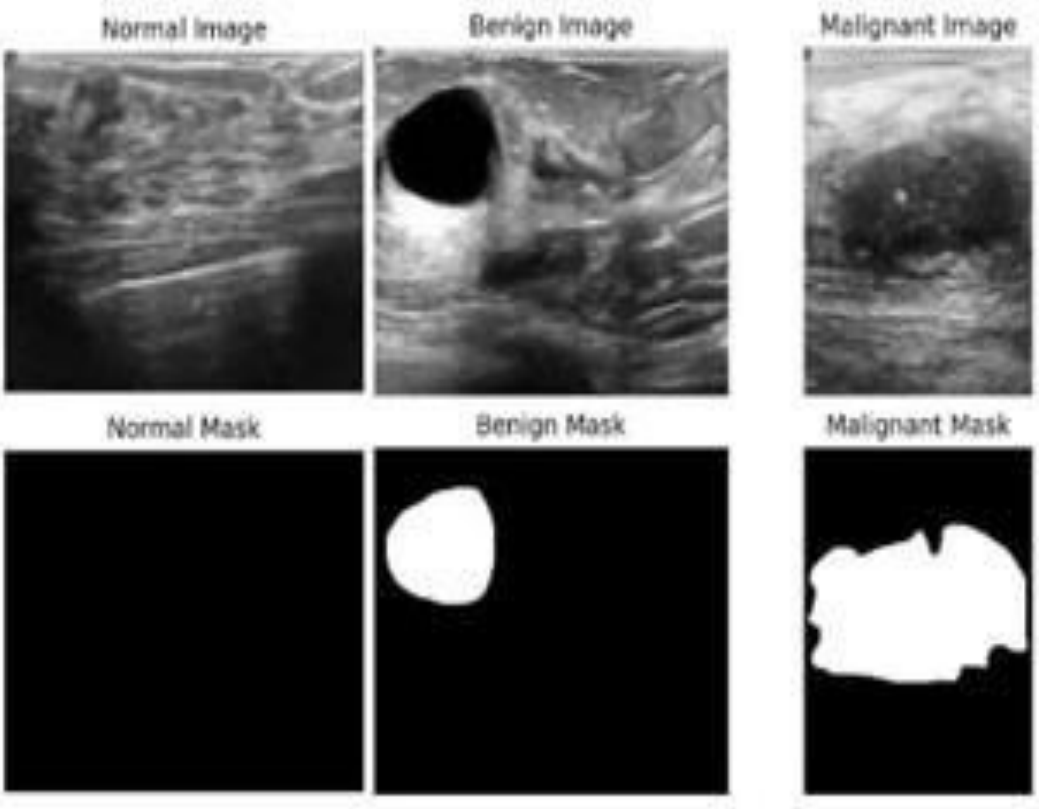


FIG 5.1 DIFFERENT TUMOR CLASSES DATA SET IMAGES.

Image Characteristics:

- All images are ultrasound scans with dimensions of 500 × 500 pixels, stored in PNG format, ensuring consistent quality and easy processing.
- Each image is paired with a mask image, highlighting the tumor's region of interest (ROI), essential for segmentation-guided learning.

Applications:

- Used for training and testing deep learning models like U-Net++ and CNN for breast cancer detection.
- Supports tasks such as tumor segmentation, classification, and feature extraction in medical and research environments.

5.1.2 DATA PRE-PROCESSING

Before Data preprocessing plays a crucial role in improving the performance of the proposed segmentation-guided deep learning framework for breast cancer detection. Since ultrasound images often contain noise, low contrast, and intensity variations, preprocessing ensures that the input data is clean, uniform, and suitable for segmentation and classification tasks.

The preprocessing phase begins with image resizing, where all images from the BUSI dataset are resized to a uniform dimension of 500 × 500 pixels. This standardization ensures consistent input size for both the U-Net++ segmentation and CNN classification models.

Next, noise reduction is applied using median and Gaussian filtering techniques to eliminate speckle noise commonly present in ultrasound scans. This step enhances image clarity and preserves important structural details required for accurate tumor detection.

Contrast enhancement is performed using Adaptive Histogram Equalization (AHE) to improve the visibility of tumor boundaries and subtle tissue variations. This operation amplifies important image features, making them more distinguishable for the segmentation model.

Following contrast enhancement, intensity normalization is applied to scale pixel values within a defined range, reducing brightness inconsistencies across the dataset. This normalization process ensures stable and balanced model training.

To improve model generalization and overcome limited dataset size, data augmentation techniques such as rotation, horizontal and vertical flipping, scaling, and translation are employed. These transformations increase the diversity of training samples without altering the underlying tumor characteristics.

Overall, data preprocessing enhances image quality, reduces artifacts, and prepares the dataset for efficient deep learning model training. By applying these preprocessing operations systematically, the proposed system achieves improved segmentation accuracy, better classification performance, and robust detection results.

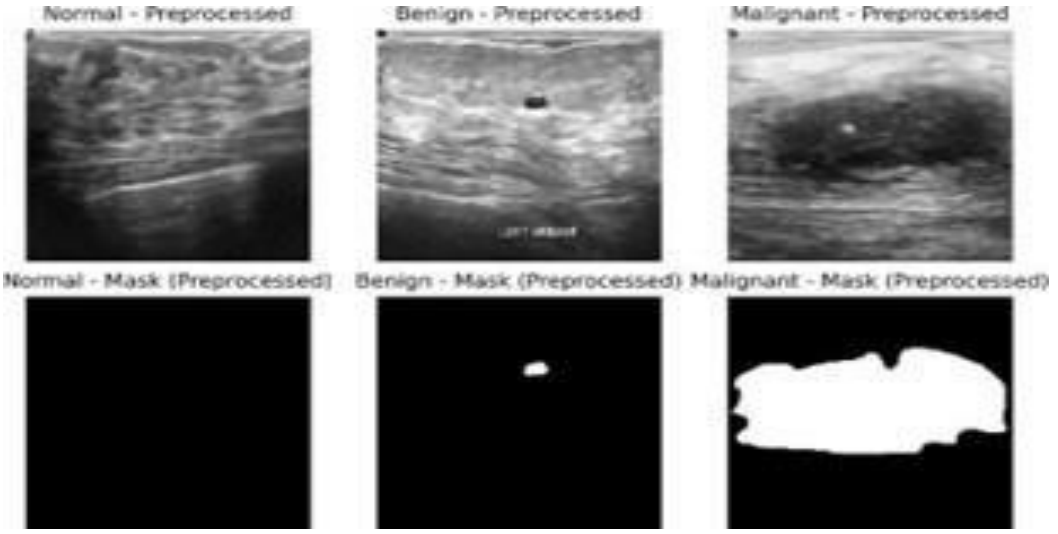


FIG 5.2 IMAGE AFTER APPLYING THE PREPROCESSING TECHNIQUE.

5.1.3 FEATURE EXTRACTION

Feature extraction is a crucial step in the proposed segmentation-guided deep learning framework for breast cancer detection, as it enables the model to learn meaningful and discriminative patterns from ultrasound images. This process transforms raw image data into high-dimensional feature representations that capture the texture, shape, and structural characteristics of the tumor region.

In this system, Convolutional Neural Networks (CNNs) are employed for automatic feature extraction. The CNN architecture processes the segmented tumor images obtained from the U-Net++ model, ensuring that only the relevant region of interest (ROI) contributes to the feature learning process. Through multiple convolutional, pooling, and activation layers, the network extracts both low-level features (such as edges and contours) and high-level semantic features (such as tumor shape, intensity distribution, and boundary irregularities).

The U-Net++ segmentation output enhances the precision of feature extraction by isolating the exact tumor region, thereby reducing background noise and irrelevant artifacts. This segmentation-guided approach ensures that the extracted features are highly representative of tumor characteristics, leading to improved classification performance.

To further refine the features, normalization and dimensionality reduction techniques are applied to eliminate redundancy and improve computational efficiency. The refined features are then passed to the CNN-based classifier, which categorizes the input into benign or malignant classes based on the learned patterns.

The integration of feature extraction and segmentation enhances the system’s interpretability

and accuracy. By focusing on tumor-specific regions, the proposed framework achieves robust detection results, making it suitable for real-world clinical diagnosis and decision support in early breast cancer.

5.1.4 MODEL BUILDING :

The proposed Segmentation-Guided Deep Learning Framework for breast cancer detection is designed to identify and classify breast tumors from ultrasound images with high accuracy and interpretability. The model integrates two main components — the U-Net++ segmentation model and the CNN-based classification network, forming a unified architecture that closely follows the diagnostic approach used by radiologists.

The U-Net++ model serves as the segmentation module responsible for accurately locating the tumor region within the ultrasound image. It employs an encoder–decoder architecture with nested skip connections, which enhances the flow of information between feature maps and allows better recovery of fine structural details. The encoder path captures essential features such as edges, shapes, and textures, while the decoder reconstructs these features into a precise segmentation mask, highlighting the region of interest (ROI). This segmentation output is essential for reducing background interference and focusing only on the tumor area.

Once the segmentation is complete, the ROI is passed to a Convolutional Neural Network (CNN) for classification. The CNN automatically extracts deep spatial features from the segmented tumor region and classifies it as benign or malignant based on the learned representations. Transfer learning is utilized by fine-tuning pretrained models like VGG16 or ResNet50 on the BUSI dataset, ensuring improved performance and reduced training time.

The model is optimized using the Adam optimizer and trained with the binary cross-entropy loss function, ensuring stable convergence. This two-stage design — segmentation followed by classification — significantly enhances diagnostic accuracy, reduces false predictions, and provides an interpretable and automated solution for early breast cancer. Moreover, the integration of segmentation masks with classification improves clinical relevance by focusing on lesion-specific regions.

Convolutional Neural Networks:

A Convolutional Neural Network (CNN) is a deep learning architecture designed for analyzing image data. In the proposed system, CNN classifies breast ultrasound images into benign and

malignant categories after tumor segmentation by U-Net++. CNNs automatically learn hierarchical spatial features from images using convolutional filters, pooling layers, and fully connected layers, as shown in Fig 5.4.

Layers in Convolutional Neural Networks:

• **Input Layer:**

Accepts preprocessed and segmented ultrasound images as multi-dimensional input arrays, representing the **region of interest (ROI)** isolated by U-Net++.

• **Convolutional Layer:**

Applies learnable filters to detect features such as edges, contrast, and texture in early layers and tumor shapes in deeper layers. The output forms **feature maps** that represent detected patterns.

• **Activation Layer:**

Introduces non-linearity to model complex tumor patterns. **ReLU** is the preferred activation due to its efficiency and ability to prevent vanishing gradients.

• **Pooling Layer:**

Reduces the spatial dimensions of feature maps while retaining critical information. **Max pooling** captures dominant features, while **average pooling** provides smoother generalization.

• **Flattening Layer:**

Converts multi-dimensional feature maps into one-dimensional vectors for dense layer processing.

• **Fully Connected Layer (Dense Layer):**

Combines learned features to identify relationships among tumor characteristics, enabling differentiation between benign and malignant tissues.

• **Output Layer:**

Generates final classification results. The **Softmax** function is used for multi-class classification, while **Sigmoid** is used for binary outputs, providing probability-based predictions.

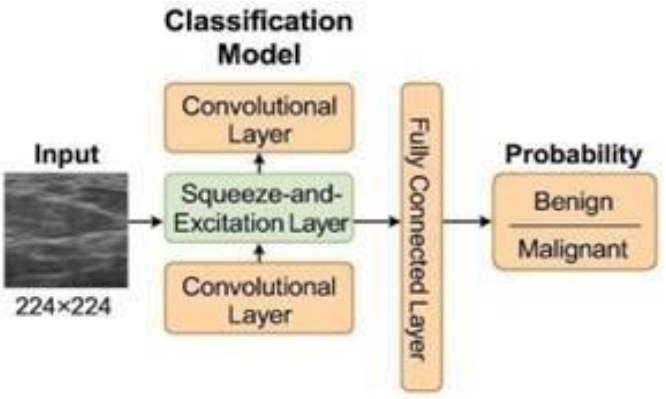


FIG 5.4 CNN MODEL ARCHITECTURE

U-Net++ Segmentation Model:

The U-Net++ architecture is an enhanced variant of the original U-Net model, specifically designed for precise medical image segmentation tasks such as tumor boundary detection in breast ultrasound images. It introduces **nested and dense skip connections** between the encoder and decoder sub-networks, which improve the flow of spatial and contextual information. This modification bridges the semantic gap between encoder and decoder feature maps, resulting in finer and more accurate segmentation outputs.

In the proposed system, U-Net++ takes preprocessed ultrasound images as input and outputs a **binary mask** indicating tumor regions. The encoder path performs repeated convolution and pooling operations to extract deep contextual features, progressively reducing the spatial resolution while increasing feature richness. The decoder path reconstructs the segmentation map by upsampling feature maps and concatenating them with corresponding encoder outputs through skip connections. This combination ensures the preservation of both low-level spatial details and high-level semantic information.

Each convolutional block in U-Net++ uses **3×3 convolution filters**, **ReLU activation**, and **batch normalization** to maintain feature consistency. The final layer employs a **1×1 convolution** followed by a **Sigmoid activation** to generate pixel-wise probability maps, effectively distinguishing tumor regions from the background.

To enhance the network's performance, the model incorporates **Squeeze-and-Excitation (SE) blocks**, which adaptively recalibrate feature responses, focusing on the most informative features related to tumor characteristics. These SE blocks strengthen the network's representational power, leading to improved tumor localization accuracy.

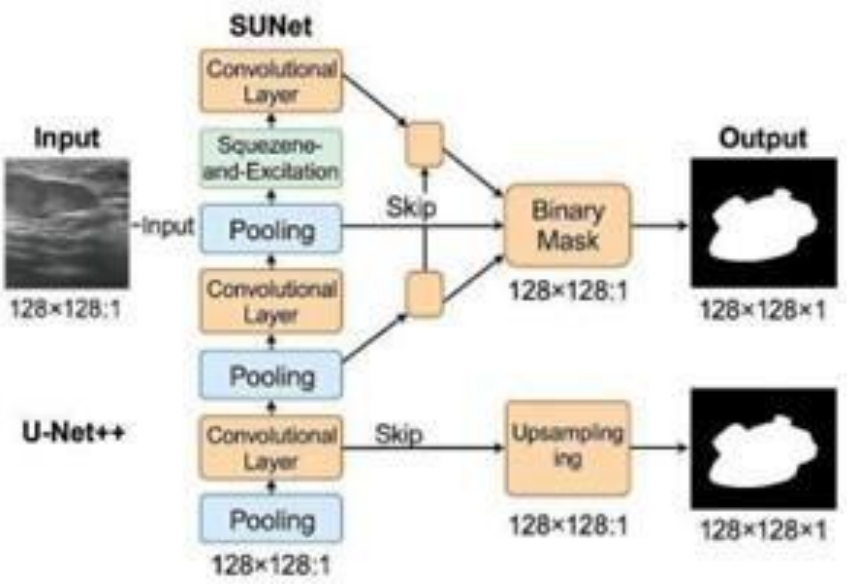


FIG 5. U-Net++ MODEL ARCHITECTURE

The architecture, as illustrated in **Fig 5.5**, shows how U-Net++ effectively merges multi-scale feature representations through its nested skip pathways. This enables the model to achieve superior segmentation quality, even in challenging ultrasound images with low contrast and irregular tumor shapes. The resulting segmentation masks are later used as input for the CNN classifier, ensuring that the classification stage focuses only on relevant tumor regions, thereby improving diagnostic accuracy.

Furthermore, U-Net++ enhances spatial consistency and feature propagation by incorporating deep supervision across intermediate decoder layers. This design allows the network to learn both coarse and fine-grained details simultaneously, improving boundary precision and tumor localization. Its ability to handle varying lesion sizes and intensities makes it particularly effective for breast ultrasound images, ensuring robust and clinically reliable segmentation results that directly support accurate classification.

U-Net++ – CNN Model Building Process

The U-Net++ – CNN model combines the strengths of U-Net++ for tumor segmentation and Convolutional Neural Networks (CNNs) for classification. The process begins with breast ultrasound images, which are preprocessed using noise reduction, contrast enhancement, and normalization to improve clarity and consistency. These refined images are then passed to the U-Net++ model for precise tumor segmentation.

In U-Net++, nested skip connections between encoder and decoder layers enhance multiscale feature learning and improve feature propagation. The encoder extracts contextual

details through convolution and pooling, while the decoder restores spatial precision via up-sampling and concatenation. Each convolutional block employs ReLU activation and batch normalization to ensure stable training. The model outputs a binary segmentation mask that accurately isolates the tumor region from normal tissue.

The segmented tumor areas are then input to the CNN classifier, which extracts deep spatial and textural patterns such as boundary irregularities and lesion shape. ReLU activation introduces non-linearity, while pooling layers reduce dimensionality and prevent overfitting. The extracted features are flattened and passed through fully connected layers for final classification.

The output layer uses a Sigmoid activation function to predict whether the tumor is benign or malignant. The model is optimized using Dice loss for segmentation and binary cross-entropy for classification with the Adam optimizer. Data augmentation (rotation, flipping, and scaling) further enhances generalization and reduces overfitting.

During inference, the ultrasound images first pass through U-Net++ to generate tumor masks, which are then analyzed by the CNN classifier. This sequential approach ensures that only relevant regions are processed, improving diagnostic accuracy and reliability. The integration of U-Net++ and CNN effectively combines precise segmentation with robust classification, providing accurate and interpretable breast cancer detection.

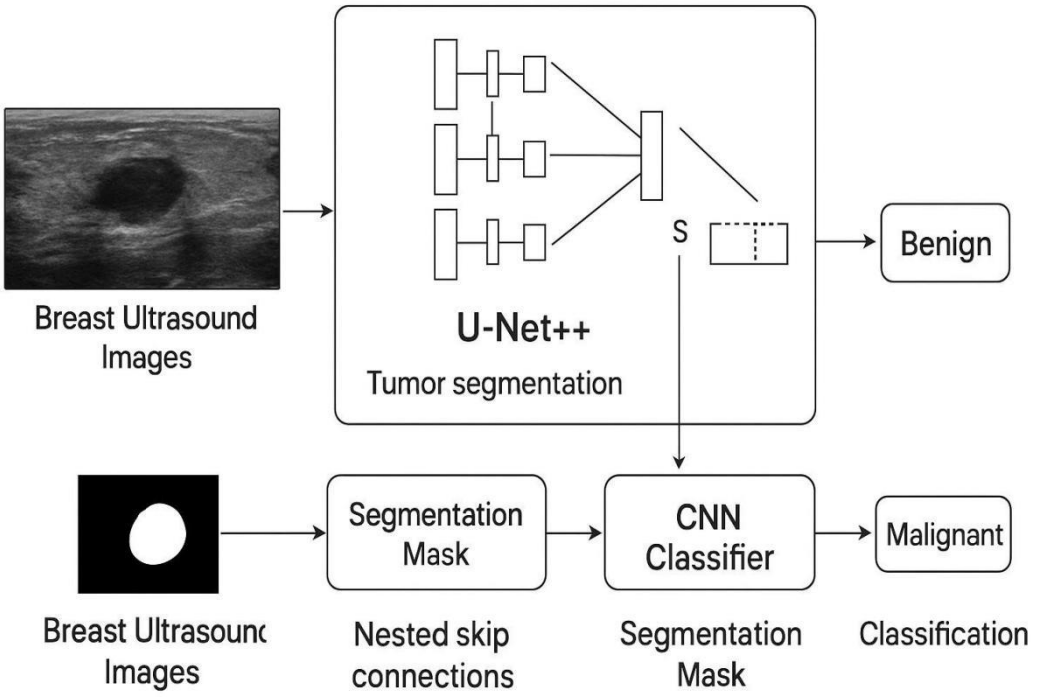


FIG 5.6 CLASSIFICATION OF BREAST CANCER

The segmented output provides a reliable foundation for classification, as it highlights the exact tumor region for analysis. By isolating the lesion, the model enhances interpretability and mimics the diagnostic workflow of radiologists—first locating the suspicious region, then classifying its type. This segmentation-guided approach not only improves accuracy but also boosts clinical reliability, making the U-Net++ model an effective tool for automated breast cancer detection.

Other models compared with the proposed U-Net++ – CNN model:

U-Net Architecture:

The U-Net model, developed for biomedical segmentation, is a pioneering convolutional network known for its encoder–decoder design. It utilizes skip connections to preserve spatial information and enables accurate localization even with limited training data. U-Net achieves strong performance in lesion segmentation tasks, but its standard form struggles with tumor variability in size, shape, and intensity. This often results in coarse boundary maps, limiting efficiency in complex ultrasound imaging. Still, U-Net remains widely adopted in healthcare, providing reliable segmentation and serving as a baseline for specialized medical image analysis.

U-Net++ (Nested U-Net):

U-Net++ introduces nested skip connections and dense pathways, allowing improved gradient flow and feature reuse. This design effectively addresses the limitations of the original U-Net by refining boundary detection and enhancing segmentation accuracy. U-Net++ captures intricate tumor structures and achieves exceptional precision in classification-driven segmentation tasks. However, the deeper architecture increases computational cost, making it less suitable for real-time or resource-limited medical environments. In breast ultrasound analysis, U-Net++ has been employed for tumor delineation, though its reliance on extensive training data can affect generalizability in smaller datasets.

CNN Classifier:

The CNN classifier laid the foundation for automated lesion categorization. Its layered convolutional design and pooling operations allow robust feature extraction and classification of benign and malignant tumors. CNN is particularly efficient in biomedical classification tasks such as lesion recognition. However, the standard CNN architecture struggles with handling high variability in tumor boundaries, often producing misclassifications when segmentation is absent. These limitations are addressed in hybrid frameworks, where CNN is combined with segmentation models to improve diagnostic accuracy and clinical reliability in breast cancer detection.

Hybrid U-Net++ + CNN Model:

The hybrid U-Net++ + CNN framework integrates segmentation and classification in a unified design. U-Net++ provides accurate tumor boundary detection, while CNN performs lesion classification with high reliability. This dense connectivity encourages feature propagation and reduces false detections compared to standalone models. The hybrid approach has achieved remarkable performance in medical imaging tasks like breast ultrasound classification. However, its combined complexity increases memory consumption and training requirements. While it performs well overall, its integration of segmentation and classification makes it highly suitable for breast cancer detection.

Comparison Summary:

While models such as U-Net, U-Net++, and CNN excel individually in segmentation or classification, they lack the unified capability essential for precise tumor localization and diagnosis. Similarly, standalone classifiers provide high accuracy but require segmentation support for improved reliability. In contrast, the proposed U-Net++ – CNN hybrid model integrates automated segmentation and classification within a single framework. U-Net++ ensures accurate tumor boundary detection, while CNN performs robust classification of benign and malignant lesions. This combination provides improved interpretability, efficiency, and diagnostic reliability compared to conventional models, making it highly suitable for real-time breast cancer detection applications.

5.2 MODULES

A module is an independent functional unit designed to perform a specific task within the overall system. The proposed U-Net++ – CNN Breast Cancer Detection System consists of several modules that collectively handle data processing, segmentation, classification, and user interaction.

CNN- UNet++ Breast Cancer Detection Project Modules:

1.Data Collection Module: Collects and organizes breast ultrasound images from the BUSI dataset, categorized as Normal, Benign, and Malignant with ground truth masks for segmentation.

Sample Code:

```
import os
```

```
def load_images_from_folder(folder):
    return [os.path.join(folder,f) for f in os.listdir(folder) if f.endswith('.png')]
```

1. **Preprocessing Module:** Enhances image quality using operations like contrast enhancement, normalization, and noise removal to ensure consistency for model training.

Sample Code:

```
import cv2
def preprocess_image(path):
    img = cv2.imread(path,0)
    img = cv2.equalizeHist(img)
    return cv2.medianBlur(img,3)mport cv2
```

2. **Segmentation Module(U-Net++ Model):** Uses **U-Net++** for precise tumor boundary detection through nested skip connections, generating binary masks highlighting tumor regions.

Sample Code:

```
from tensorflow.keras.models import_load_model
def segment_tumor(image):
model=load_model('unetpp_model.h5')
    return model.predict(image.reshape(1,256,256,1))
```

3. **CNN Classification Module:** The **CNN** extracts deep spatial features and classifies segmented tumors as *Benign* or *Malignant* using the **Sigmoid** activation in the output layer.

Sample Code:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,MaxPooling2D,Flatten,Dense
model = Sequential([
    Conv2D(32,(3,3),activation='relu',input_shape=(256,256,1)),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(1,activation='sigmoid')
])
```

4. **Evaluation Module:** Assesses performance using metrics such as accuracy, precision, recall, and Dice coefficient.

Sample Code:

```
from sklearn.metrics import accuracy_score
def evaluate_model(y_true,y_pred):
    return accuracy_score(y_true,y_pred)
```

5. **Flask Backend and Frontend Modules:** The Flask backend handles image upload, processing, and result prediction through API endpoints, while the frontend provides a simple interface for users to upload images and view results.

   Sample Code:

```
from flask import Flask,request,jsonify
app=Flask(__name__)
@app.route('/predict',methods=['POST'])
def predict():
    return jsonify({'result':'Prediction Here'})
```

6. **File Management Module:** Handles uploaded files and temporary storage, deleting them post-processing to optimize space.

   Sample Code:

```
import os
def delete_file(file_path):
if os.path.exists(file_path): os.remove(file_path)
```

5.3 UML DIAGRAMS

The workflow of the U-Net++ – CNN model for breast cancer detection begins with collecting and preprocessing ultrasound images for training and testing. Preprocessing involves steps such as resizing, normalization, and applying image enhancement techniques like histogram equalization for contrast improvement, median filtering for noise removal, and normalization to standardize intensity values. Tumor region segmentation is then performed using the U-Net++ model, which accurately isolates the region of interest (ROI) by identifying tumor boundaries from the surrounding breast tissue.

Feature extraction is conducted automatically during the segmentation and classification process. The U-Net++ model generates a binary mask that precisely delineates the tumor region, while the CNN classifier extracts deep spatial and texture-based features from these segmented regions to determine whether the tumor is benign or malignant. The processed dataset is divided into training and testing subsets to validate the model's performance. Metrics such as accuracy, sensitivity, specificity, precision, recall, and Dice coefficient are used to assess the system's reliability in detecting and classifying tumors.

Once the U-Net++ – CNN hybrid model is trained, it can be saved for deployment in clinical applications. The trained model includes the learned weights, segmentation architecture, and classification parameters, allowing it to be reused without retraining. This feature is particularly beneficial for real-time deployment in diagnostic centers, where the model can analyze ultrasound scans and deliver rapid, accurate results. A performance report is generated after training, which includes statistical metrics and visual segmentation outputs that help evaluate the model's effectiveness and trustworthiness. These reports provide valuable insights for radiologists and medical professionals, enabling them to interpret results confidently and integrate the model into diagnostic workflows.

This flowchart (Fig 5.7) outlines the step-by-step process of developing and evaluating the deep learning pipeline for breast cancer detection. The process begins with image acquisition from the BUSI dataset, followed by preprocessing operations such as noise reduction, normalization, and contrast enhancement. The preprocessed images are then fed into the U-Net++ segmentation model, which identifies tumor regions and generates corresponding segmentation masks. These segmented outputs are further processed by the CNN classifier, which categorizes the tumor as benign or malignant. After training and evaluation, the model

is stored and can be reused for inference without retraining.

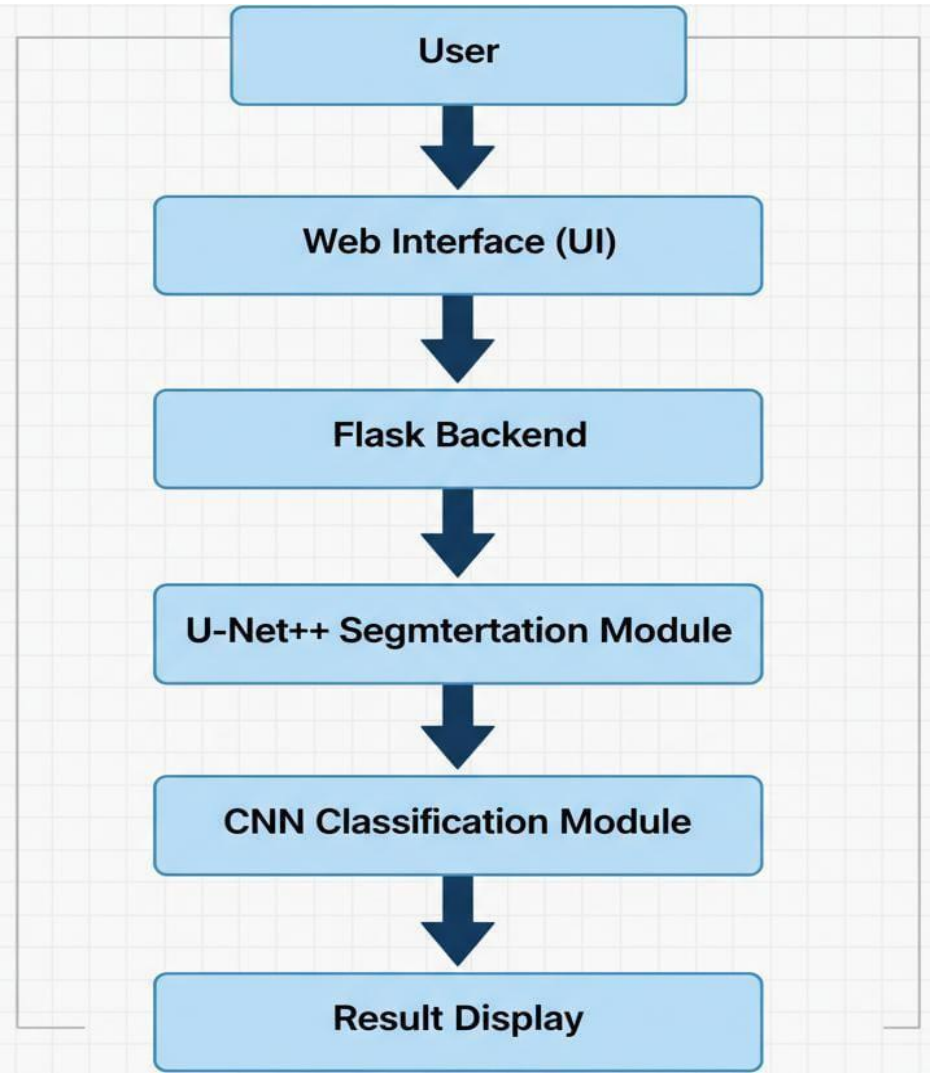


FIG 5.7 DESIGN OVERVIEW

In the breast cancer detection system, UML (Unified Modeling Language) diagrams are used to visualize the architecture, data flow, and relationships between modules. The Class Diagram illustrates the main system components and their responsibilities. The DatasetHandler class manages image loading and dataset organization. The PreprocessingModule handles image enhancement operations such as normalization and noise reduction. The SegmentationModule (U-Net++) performs precise tumor region extraction, while the CNN Classifier processes the segmented outputs and classifies tumors as benign or malignant.

6. IMPLEMENTATION

6.1 MODEL IMPLEMENTATION

U-Net++ + CNN Model

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, jaccard_score, confusion_matrix
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten, Conv1D, MaxPooling1D,
BatchNormalization, GlobalAveragePooling1D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
import matplotlib.pyplot as plt

# Combine GLCM features into a single DataFrame
all_glcm_features = []
all_labels = []

for features, label in [
  (bglcm_features_list, 'benign'),
  (mglcm_features_list, 'malignant'),
  (nglcm_features_list, 'normal') ]:

 for feature_dict in features:
       flattened = [val for sublist in
feature_dict.values() for val in sublist]
all_glcm_features.append(flattened)
all_labels.append(label)

df = pd.DataFrame(all_glcm_features) df['label']
=all_labels

label_encoder = LabelEncoder()
y = label_encoder.fit_transform(df['label']) X =
df.drop('label', axis=1)

X_train, X_temp, y_train, y_temp =
train_test_split(X, y, test_size=0.3,
```

```
random_state=42) X_val, X_test, y_val, y_test =
train_test_split(X_temp, y_temp, test_size=0.5,
random_state=42)

scaler = StandardScaler() X_train_scaled =
scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)

X_train_cnn =
X_train_scaled.reshape(X_train_scaled.shape[0],
X_train_scaled.shape[1], 1)
X_val_cnn =
X_val_scaled.reshape(X_val_scaled.shape[0],
X_val_scaled.shape[1], 1)
 X_test_cnn =
X_test_scaled.reshape(X_test_scaled.shape[0],
X_test_scaled.shape[1], 1)

def create_cnn_model(input_shape):
    model = Sequential()
    model.add(Conv1D(64, kernel_size=7,
activation='relu', input_shape=input_shape))

   model.add(MaxPooling1D(pool_size=3))
model.add(Conv1D(128, kernel_size=5,
activation='relu'))

  model.add(MaxPooling1D(pool_size=3))
model.add(Conv1D(256, kernel_size=3,
activation='relu'))

 model.add(MaxPooling1D(pool_size=2))
model.add(GlobalAveragePooling1D())
model.add(Dense(256, activation='relu'))

model.add(Dropout(0.5))
model.add(BatchNormalization())
model.add(Dense(3, activation='softmax'))
model.compile(optimizer=Adam(learning_rate=0.00
05),

loss='sparse_categorical_crossentropy',
metrics=['accuracy'])      return model

cnn_model =
create_cnn_model((X_train_cnn.shape[1], 1))
```

```
early_stopping = EarlyStopping(monitor='val_loss',
patience=10, restore_best_weights=True) reduce_lr
= ReduceLROnPlateau(monitor='val_loss',
factor=0.2, patience=5, min_lr=0.00001)

cnn_history = cnn_model.fit(     X_train_cnn,
y_train,     epochs=150,     batch_size=64,
validation_data=(X_val_cnn, y_val),
callbacks=[early_stopping, reduce_lr],     verbose=2
)

feature_extractor = Model(inputs=cnn_model.input,
outputs=cnn_model.layers[-3].output)
cnn_features_train =
feature_extractor.predict(X_train_cnn)
cnn_features_val =
feature_extractor.predict(X_val_cnn)
cnn_features_test =
feature_extractor.predict(X_test_cnn)

svm_model = SVC(probability=True) param_grid =
{     'kernel': ['linear', 'rbf'],
   'C': [1, 10, 100],
   'gamma': ['scale', 'auto'] }

grid_search = GridSearchCV(svm_model,
param_grid, cv=3, scoring='accuracy', verbose=2)
grid_search.fit(cnn_features_train, y_train)
best_svm_model = grid_search.best_estimator_

def sensitivity_specificity(y_true, y_pred):     cm =
confusion_matrix(y_true, y_pred)     tp =
np.diag(cm)     fp = cm.sum(axis=0) - tp     fn =
cm.sum(axis=1) - tp     tn = cm.sum() - (fp + fn + tp)
sensitivity = tp / (tp + fn)     specificity = tn / (tn +
fp)     return sensitivity, specificity
y_pred_test =
best_svm_model.predict(cnn_features_test)
train_acc = accuracy_score(y_train,
best_svm_model.predict(cnn_features_train))
val_acc = accuracy_score(y_val,
best_svm_model.predict(cnn_features_val)) test_acc
= accuracy_score(y_test, y_pred_test)
train_jaccard = jaccard_score(y_train,
best_svm_model.predict(cnn_features_train),
average='macro') val_jaccard = jaccard_score(y_val,
best_svm_model.predict(cnn_features_val),
average='macro') test_jaccard =
```

```
jaccard_score(y_test, y_pred_test, average='macro')
plt.figure(figsize=(14,10)) plt.subplot(2,2,1)
plt.plot(cnn_history.history['accuracy'],
label='Train')
plt.plot(cnn_history.history['val_accuracy'],
label='Validation') plt.title('CNN–SVM Model
Accuracy (BUSI Dataset)') plt.xlabel('Epoch')
plt.ylabel('Accuracy') plt.legend()
plt.subplot(2,2,2)
plt.plot(cnn_history.history['loss'], label='Train')
plt.plot(cnn_history.history['val_loss'],
label='Validation') plt.title('CNN–SVM Model
Loss') plt.xlabel('Epoch') plt.ylabel('Loss')
plt.legend()
plt.subplot(2,2,3) plt.bar(['Train','Val','Test'],
[train_jaccard*100, val_jaccard*100,
test_jaccard*100],
color=['skyblue','orange','green'])  plt.ylabel('Jaccard
Coefficient (%)') plt.title('Jaccard Coefficient across
Datasets')
sens_train, spec_train =
sensitivity_specificity(y_train,
best_svm_model.predict(cnn_features_train))
sens_val, spec_val = sensitivity_specificity(y_val,
best_svm_model.predict(cnn_features_val))
sens_test, spec_test = sensitivity_specificity(y_test,
y_pred_test)
plt.subplot(2,2,4)
plt.bar(['Train','Val','Test'],
[np.mean(sens_train)*100, np.mean(sens_val)*100,
np.mean(sens_test)*100],
color=['skyblue','orange','green']) plt.ylabel('Average
Sensitivity (%)') plt.title('Average Sensitivity across
Datasets')
plt.tight_layout() plt.show()
```

6.2 CODING

PER-PROCESSING SEGMENTATION AND FEATURE EXTRACTION

```
import os
from google.colab import drive
import cv2
from google.colab.patches import cv2_imshow
import numpy as np
import albumentations as A
import skfuzzy as fuzz
from skimage.feature import graycomatrix, graycoprops
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, jaccard_score, confusion_matrix
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Dense, Dropout, Conv1D, MaxPooling1D,
BatchNormalization, GlobalAveragePooling1D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image
import seaborn as sns

# Mount Google Drive
drive.mount('/content/drive')

# Dataset path (BUSI Dataset)
folder_path = '/content/drive/MyDrive/BUSI_Dataset/benign'

# Initialize GLCM feature list
benign_glcm_features = []

# Function to extract GLCM features
def extract_glcm_features(image):
    distances = [1, 2, 3]
    angles = [0, np.pi/4, np.pi/2, 3*np.pi/4]
    glcm = graycomatrix(image, distances=distances, angles=angles, symmetric=True,
normed=True)
    features = {
        'contrast': graycoprops(glcm, 'contrast').flatten(),
        'dissimilarity': graycoprops(glcm, 'dissimilarity').flatten(),
        'homogeneity': graycoprops(glcm, 'homogeneity').flatten(),
        'energy': graycoprops(glcm, 'energy').flatten(),
```

```
        'correlation': graycoprops(glcm, 'correlation').flatten(),
        'ASM': graycoprops(glcm, 'ASM').flatten()
    }
    return features

processed_count = 0
max_process = 400

# Process all images in each class
for filename in os.listdir(folder_path):
    if processed_count >= max_process:
        break
    if filename.endswith('.jpg') or filename.endswith('.png'):
        image_path = os.path.join(folder_path, filename)
        img = cv2.imread(image_path)
        if img is not None:
            resize = cv2.resize(img, (224, 224))
            gray = cv2.cvtColor(resize, cv2.COLOR_BGR2GRAY)
            equalized = cv2.equalizeHist(gray)
            clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
            enhanced = clahe.apply(equalized)
            gamma = 1.5
            gamma_corrected = np.power(enhanced / 255.0, gamma) * 255.0
            gamma_corrected = gamma_corrected.astype(np.uint8)
            transform = A.Compose([A.HorizontalFlip(p=0.5)])
            augmented = transform(image=gamma_corrected)['image']
            filtered = cv2.medianBlur(augmented, 5)
            pixel_values = filtered.reshape((-1, 1)).astype(float)
            n_clusters = 3
            cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(pixel_values.T, n_clusters, 2,
error=0.005, maxiter=1000)
            cluster_membership = np.argmax(u, axis=0)
            segmented = cluster_membership.reshape(filtered.shape)
            binary_segmented = (segmented == 1).astype(np.uint8) * 255
            glcm_features = extract_glcm_features(filtered)
            benign_glcm_features.append(glcm_features)
            cv2_imshow(binary_segmented)
            print(f"Processed and extracted features from {filename}")
            processed_count += 1

print(f"Extracted GLCM features for {len(benign_glcm_features)} benign images.")

# Repeat similar extraction for normal and malignant folders before combining
# Example variable names: normal_glcm_features, malignant_glcm_features

# Combine features from all categories
```

```
all_glcm_features = []
all_labels = []

for features, label in [(benign_glcm_features, 'benign'),
                        (normal_glcm_features, 'normal'),
                        (malignant_glcm_features, 'malignant')]:
    for feature_dict in features:
        flattened = [val for sublist in feature_dict.values() for val in sublist]
        all_glcm_features.append(flattened)
        all_labels.append(label)

df = pd.DataFrame(all_glcm_features)
df['label'] = all_labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(df['label'])
X = df.drop('label', axis=1)

# Train/validation/test split
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, random_state=42,
stratify=y)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=42, stratify=y_temp)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)

# Reshape for CNN input
X_train_cnn = X_train_scaled.reshape(X_train_scaled.shape[0], X_train_scaled.shape[1], 1)
X_val_cnn = X_val_scaled.reshape(X_val_scaled.shape[0], X_val_scaled.shape[1], 1)
X_test_cnn = X_test_scaled.reshape(X_test_scaled.shape[0], X_test_scaled.shape[1], 1)

# CNN Model
def create_cnn_model(input_shape):
    model = Sequential()
    model.add(Conv1D(64, kernel_size=7, activation='relu', input_shape=input_shape))
    model.add(MaxPooling1D(pool_size=3))
    model.add(Conv1D(128, kernel_size=5, activation='relu'))
    model.add(MaxPooling1D(pool_size=3))
    model.add(Conv1D(256, kernel_size=3, activation='relu'))
    model.add(MaxPooling1D(pool_size=2))
    model.add(GlobalAveragePooling1D())
    model.add(Dense(256, activation='relu'))
    model.add(Dropout(0.5))
    model.add(BatchNormalization())
```

```
    model.add(Dense(3, activation='softmax'))  # 3 classes: normal, benign, malignant
    model.compile(optimizer=Adam(learning_rate=0.0005),
                  loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    return model

cnn_model = create_cnn_model((X_train_cnn.shape[1], 1))
early_stopping = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-5)
cnn_history = cnn_model.fit(X_train_cnn, y_train, epochs=150, batch_size=64,
                            validation_data=(X_val_cnn, y_val),
                            callbacks=[early_stopping, reduce_lr], verbose=2)

# Extract CNN Features and Train SVM
feature_extractor = Model(inputs=cnn_model.input, outputs=cnn_model.layers[-3].output)
cnn_features_train = feature_extractor.predict(X_train_cnn)
cnn_features_test = feature_extractor.predict(X_test_cnn)
svm_model = SVC(probability=True, kernel='rbf', C=10, gamma='scale')
svm_model.fit(cnn_features_train, y_train)
y_pred_test = svm_model.predict(cnn_features_test)
test_acc = accuracy_score(y_test, y_pred_test)
print(f"Testing Accuracy: {test_acc * 100:.2f}%")

# Prediction
def preprocess_image(img_path, target_size):
    img = image.load_img(img_path, target_size=target_size, color_mode='grayscale')
    img_array = image.img_to_array(img)
    img_array = img_array / 255.0
    img_array = img_array.reshape(1, target_size[0], 1)
    return img_array

img_path = '/content/drive/MyDrive/BUSI_Dataset/malignant/1.png'
target_size = (X_train_cnn.shape[1], X_train_cnn.shape[2])
input_image = preprocess_image(img_path, target_size)
predicted_class = np.argmax(cnn_model.predict(input_image), axis=1)[0]
predicted_label = label_encoder.inverse_transform([predicted_class])[0]
print(f"The predicted class for the input image is: {predicted_label}")

# Analysis
models = ['VGG', 'ANN', 'RFC', 'RNNs', 'FCNNs', 'CNN', 'CNN-SVM (Proposed Model)']
accuracies = [93.75, 95.89, 92.70, 92.5, 93, 95, 97.94]
jaccard_coefficients = [84, 85, 87, 81, 80, 86, 90]
sensitivities = [91, 92, 94, 89, 88, 94, 95]
specificities = [96, 97, 97, 86, 84, 97, 98.1]

plt.figure(figsize=(12,8))
```

```
bars = plt.barh(models, accuracies, color=plt.cm.viridis(np.linspace(0.2, 0.8, len(models))), edgecolor='black')
for bar in bars:
    plt.text(bar.get_width()+0.5, bar.get_y()+bar.get_height()/2, f'{bar.get_width():.2f}%', va='center', fontsize=12, weight='bold')
plt.title('Model Accuracies', fontsize=16, weight='bold')
plt.xlabel('Accuracy (%)')
plt.grid(axis='x', linestyle='--', alpha=0.7)
plt.show()

plt.figure(figsize=(12,2))
sns.heatmap(np.array(jaccard_coefficients).reshape(1,-1), annot=True, fmt=".1f", cmap="coolwarm_r",
            xticklabels=models, yticklabels=['Jaccard Coefficient'])
plt.title('Jaccard Coefficients', fontsize=16, weight='bold')
plt.show()

plt.figure(figsize=(14,8))
plt.bar(models, sensitivities, color='teal', edgecolor='black', linewidth=1.5)
plt.axhline(np.mean(sensitivities), color='gray', linestyle='--', linewidth=2, label=f'Average: {np.mean(sensitivities):.2f}%')
plt.legend()
plt.title('Model Sensitivity Comparison', fontsize=16, weight='bold')
plt.ylabel('Sensitivity (%)')
plt.ylim([80,100])
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(10,6))
sns.heatmap(np.array(specificities).reshape(1,-1), annot=True, fmt=".1f", cmap="coolwarm",
            xticklabels=models, yticklabels=["Specificity"])
plt.title("Specificity Across Models", fontsize=16, weight='bold')
plt.show()"""
```

app.py

```
import os
import numpy as np
import cv2
from PIL import Image
from flask import Flask, request, jsonify,
render_template, redirect
from flask_cors import CORS
from tensorflow import keras


# Load Segmentation Model (U-Net++)

MODEL_PATH =
os.path.join(os.path.dirname(__file__),
"unetpp_model2.keras")
if not os.path.exists(MODEL_PATH):
    raise FileNotFoundError(f"Model not found at
{MODEL_PATH}")

segmentation_model =
keras.models.load_model(MODEL_PATH,
compile=False)

# Configs

THRESHOLD = 0.85
MASK_TEMP_PATH = os.path.join("static",
"mask_temp.jpg")
OVERLAY_TEMP_PATH = os.path.join("static",
"overlay_temp.jpg")


# Validate if image looks like ultrasound

def is_valid_ultrasound(image_path):
    """Check if uploaded image resembles a breast
ultrasound image."""
    img = cv2.imread(image_path)
    if img is None:
        return False
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    gray_rgb = cv2.cvtColor(gray,
cv2.COLOR_GRAY2BGR)
    diff = np.abs(img.astype(np.int32)
 - gray_rgb.astype(np.int32))
    colorfulness = np.mean(diff)
    return colorfulness < 15     # ultrasound images are
mostly grayscale
```

```
# Preprocessing Function

def preprocess_image(image_path):
    image = Image.open(image_path).convert("L")
    image_resized = image.resize((128, 128))
    image_np = np.array(image_resized,
dtype=np.float32) / 255.0
    return np.expand_dims(image_np, axis=(0, -1))


# Prediction Function

def predict_image(image_path, return_mask=False):
    image_np = preprocess_image(image_path)
    mask = segmentation_model.predict(image_np)

    mask_bin = (mask >
THRESHOLD).astype(np.float32)
    tumor_area = np.sum(mask_bin)
    max_value = mask.max()

    if return_mask:
        mask_img = Image.fromarray((mask_bin[0, :, :, 0] *
255).astype(np.uint8))
        mask_img.save(MASK_TEMP_PATH)

    MIN_TUMOR_AREA = 50
    MIN_MASK_VALUE = 0.5

    if tumor_area < MIN_TUMOR_AREA or max_value
< MIN_MASK_VALUE:
        result = {
            "prediction": "Normal (No Tumor)",
            "is_tumor": False,
            "description": "No abnormal region detected.",
            "recommendation": "No further medical action
required."
        }
    else:
        result = {
            "prediction": "Tumor Detected",
            "is_tumor": True,
            "description": "Abnormal region detected by
segmentation.",
            "recommendation": "Consult a radiologist or
medical expert for further analysis."
        }

    if return_mask:
        result["mask_image"] = "/" +
MASK_TEMP_PATH.replace("\\", "/")

    return result
```

```
# Create Overlay with Red Border

def create_overlay(original_image_path, mask_path,
output_path=OVERLAY_TEMP_PATH):
    try:
        orig = cv2.imread(original_image_path)
        mask = cv2.imread(mask_path,
cv2.IMREAD_GRAYSCALE)
        if orig is None or mask is None:
            print("❌ Could not read original or mask
image.")
            return None

        mask_resized = cv2.resize(mask, (orig.shape[1],
orig.shape[0]))
        contours, _ = cv2.findContours((mask_resized >
128).astype(np.uint8),
                                       cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
        outlined = orig.copy()
        cv2.drawContours(outlined, contours, -1, (0, 0, 255),
2)
        cv2.imwrite(output_path, outlined)
        return output_path
    except Exception as e:
        print(f"Overlay creation failed: {e}")
        return None

# Flask App Setup

app = Flask(__name__, template_folder="templates",
static_folder="static")
CORS(app)

@app.route("/")
def index():
    return redirect("/home")

# Pages

@app.route("/home")
def home():
    return render_template("Home.html")

@app.route("/features")
def features_page():
    return render_template("Features.html")

@app.route("/upload")
def upload_page():
    return render_template("Upload.html")
```

```
@app.route("/login")
def login_page():
    return render_template("Login.html")

@app.route("/contact")
def contact_page():
    return render_template("Contact.html")

@app.route("/signup")
def signup_page():
    return render_template("Signup.html")

@app.route("/feature_high_accuracy")
def feature_high_accuracy():
    return render_template("feature_high_accuracy.html")

@app.route("/feature_fast_processing")
def feature_fast_processing():
    return
render_template("feature_fast_processing.html")

@app.route("/feature_detailed_insights")
def feature_detailed_insights():
    return
render_template("feature_detailed_insights.html")

@app.route("/awareness")
def awareness_page():
    return render_template("awareness.html")

@app.route("/treatments")
def treatments_page():
    return render_template("treatments.html")

@app.route("/statistics")
def statistics_page():
    return render_template("statistics.html")

# ------------------------------
# Predict Endpoint
# ------------------------------
@app.route("/predict", methods=["POST"])
def predict():
    if "image" not in request.files:
        return jsonify({"error": "No image uploaded"}), 400

    file = request.files["image"]
    temp_path = os.path.join("static", "temp.jpg")
    file.save(temp_path)

    try:
        if not is_valid_ultrasound(temp_path):
```

```
            return jsonify({
                "prediction": "Invalid Image",
                "is_tumor": False,
                "description": "This image does not appear to
be a breast ultrasound image.",
                "recommendation": "Please upload a valid
breast ultrasound image from the BUSI dataset or similar."
            }), 400

        result = predict_image(temp_path,
return_mask=True)
        overlay_path = create_overlay(temp_path,
MASK_TEMP_PATH)
        if overlay_path:
            result["overlay_image"] = "/" +
overlay_path.replace("\\", "/")

        return jsonify(result)
    except Exception as e:
        return jsonify({"error": str(e)}), 500

# ------------------------------
# View Details Page
# ------------------------------
@app.route("/details")
def view_details():
    if os.path.exists(OVERLAY_TEMP_PATH):
        return render_template("Details.html",
image_path="/" +
OVERLAY_TEMP_PATH.replace("\\", "/"))
    return "No highlighted tumor image found."

# ------------------------------
# Run Flask Server
# ------------------------------
if __name__ == "__main__":
    print("🚀 Segmentation Model 2 Server running at:
http://127.0.0.1:5000")
    app.run(host="0.0.0.0", port=5000, debug=True)
```

Html and Css

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width,
initial-scale=1" />
  <title>Breast Cancer Detection</title>

  <!-- CSS Links -->
  <link rel="stylesheet" href="{{ url_for('static',
filename='Home.css') }}">
```

```
 <link rel="stylesheet" href="{{ url_for('static', filename='Navbar.css') }}">

 <!-- Favicon -->
 <link rel="icon" type="image/png" href="{{ url_for('static', filename='icon.png') }}">
</head>

<body>
 <!-- ✅ Updated Navbar (same as navbar.html) -->
 <nav class="navbar">
  <div class="navbar-logo" onclick="window.location.href='/'" style="cursor:pointer;">
    <span class="logo-icon">🎗</span> BreastCare
  </div>

  <button class="navbar-toggle" id="navbarToggle">&#9776;</button>

  <ul class="navbar-links" id="navbarLinks">

    <li><a href="/home">Home</a></li>
    <li><a href="/features">Features</a></li>
    <li><a href="/about">About</a></li>
    <li><a href="/upload">Test&Check</a></li>
    <li><a href="/contact">Contact</a></li>
    <li><a href="/login">Login/SignUp</a></li>
  </ul>
 </nav>

 <!-- Hero Section -->
 <div class="section1">
  <div class="main">
   <div class="txt">
    <h2>Breast Cancer Detection</h2>
    <p>
     Detect and segment breast cancer tumors from mammogram images using advanced AI.
     Our system provides accurate lesion boundaries to assist radiologists in faster
     and more reliable diagnosis.
    </p>

   </div>
   <div class="pictures">
    <div class="img" id="img1"></div>
   </div>
  </div>
 </div>

 <!-- Wave Divider -->
```

```
 <svg class="wave" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 150">
  <path fill="#ffffff" fill-opacity="1" d="M0,96L48,101.3C96,107,192,117,288,138.7C384,160,480,192,576,192C672,192,768,160,864,138.7C960,117,1056,107,1152,90.7C1248,75,1344,53,1392,42.7L1440,32L1440,150L1392,150C1344,150,1248,150,1152,150C1056,150,960,150,864,150C768,150,672,150,576,150C480,150,384,150,288,150C192,150,96,150,48,150L0,150Z"></path>
 </svg>

 <!-- FAQ Section -->
 <section class="faq-section">
  <h2 class="section-title">Frequently Asked Questions</h2>
  <div class="faq-container">
   <div class="faq-item">
    <button class="faq-question">What are the early signs of breast cancer?</button>
    <div class="faq-answer">
     <p>Early signs include lumps, swelling, skin dimpling, nipple discharge, or pain. Regular screening helps in early detection.</p>
    </div>
   </div>
   <div class="faq-item">
    <button class="faq-question">Who should get screened?</button>
    <div class="faq-answer">
     <p>Women above 40, those with a family history, or other risk factors should get regular mammograms.</p>
    </div>
   </div>
   <div class="faq-item">
    <button class="faq-question">Can lifestyle reduce risk?</button>
    <div class="faq-answer">
     <p>Yes. Maintaining a healthy weight, regular exercise, reducing alcohol, and a balanced diet can lower risk.</p>
    </div>
   </div>
  </div>
 </section>

 <!-- Wave Divider -->
 <svg class="wave wave-bottom" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 150">
  <path fill="#f5f7fa" fill-opacity="1"
```

```
d="M0,96L48,101.3C96,107,192,117,288,138.7C384,16
0,480,192,576,192C672,192,768,160,864,138.7C960,11
7,1056,107,1152,90.7C1248,75,1344,53,1392,42.7L144
0,32L1440,150L1392,150C1344,150,1248,150,1152,150
C1056,150,960,150,864,150C768,150,672,150,576,150
C480,150,384,150,288,150C192,150,96,150,48,150L0,1
50Z"></path>
  </svg>

  <!-- Prevention & Recommendations Section -->
  <section class="prevention-section">
    <h2 class="section-title">Prevention &
Recommendations</h2>
    <div class="prevention-container">
      <div class="prevention-item" data-index="0">
        <div class="prevention-icon">□</div>
        <h3>Healthy Diet</h3>
      </div>
      <div class="prevention-item" data-index="1">
        <div class="prevention-icon">♀</div>
        <h3>Exercise</h3>
      </div>
      <div class="prevention-item" data-index="2">
        <div class="prevention-icon">❌</div>
        <h3>Avoid Alcohol & Smoking</h3>
      </div>
      <div class="prevention-item" data-index="3">
        <div class="prevention-icon">□♀</div>
        <h3>Stress Management</h3>
      </div>
    </div>
  </section>

  <!-- Detail Popup -->
  <div class="prevention-detail">
    <button class="back-home">← Back</button>
    <div class="detail-content">
      <!-- Dynamic content populated by JS -->
    </div>
  </div>

  <!-- Footer -->
  <footer class="footer">
    <p>© 2025 BreastCare — AI-powered Breast Cancer
Detection System</p>
  </footer>

  <!-- JS -->
  <script>
    // ✅ Mobile Navbar Toggle (same as navbar.html)
    const toggleBtn =
document.getElementById("navbarToggle");
```

```
  const links = document.getElementById("navbarLinks");
  let isOpen = false;

  toggleBtn.addEventListener("click", () => {
    isOpen = !isOpen;
    links.classList.toggle("show");
    toggleBtn.innerHTML = isOpen ? "&times;" : "&#9776;";
  });

  // FAQ toggle
  const faqs = document.querySelectorAll('.faq-question');
  faqs.forEach(faq => {
    faq.addEventListener('click', () => {
    faq.classList.toggle('active');
      const answer = faq.nextElementSibling;
      answer.style.maxHeight = answer.style.maxHeight ? null : answer.scrollHeight + "px";
    });
  });

  // Prevention popup logic
  const details = [
    {
      title: "Healthy Diet",
      content: `
        <p>Consume vegetables, fruits, whole grains, and lean proteins. Limit processed foods.</p>
        <p>These foods help reduce inflammation, boost immunity, and maintain healthy weight.</p>
      `
    },
    {
      title: "Exercise",
      content: `
        <p>Engage in 30 mins of moderate exercise daily.</p>
        <p>Helps maintain weight, improves heart health, and reduces stress.</p>
      `
    },
    {
      title: "Avoid Alcohol & Smoking",
      content: `
        <p>Smoking and alcohol increase cancer risk significantly.</p>
        <p>Avoiding these improves overall health and lowers cancer risk.</p>
      `
    },
    {
```

```
      title: "Stress Management",
      content: `
        <p>Practice meditation, yoga, or hobbies to reduce stress.</p>
        <p>Helps balance hormones and improves mental well-being.</p>
      `
    },
  ];

  const preventionItems = document.querySelectorAll('.prevention-item');
  const detailPopup = document.querySelector('.prevention-detail');
  const detailContent = document.querySelector('.detail-content');
  const backHome = document.querySelector('.back-home');

  preventionItems.forEach(item => {
    item.addEventListener('click', () => {
      const index = parseInt(item.dataset.index);
      detailContent.innerHTML = <h3>${details[index].title}</h3>${details[index].content};
      detailPopup.style.display = 'flex';
    });
  });

  backHome.addEventListener('click', () => {
    detailPopup.style.display = 'none';
  });
 </script>
</body>
</html>
```

CSS Code

```
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600;700&display=swap');

/* === GLOBAL STYLES === */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}

body {
  background-color: #ffbce1;
  color: #e7c4d4;
  line-height: 1.6;
```

```
}


/* === HERO SECTION === */
.section1 {
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  padding: 2rem 5%;
  background: #f5f7fa;
  position: relative;
  overflow: hidden;
}

.main {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 2rem;
  align-items: center;
  width: 100%;
  max-width: 1200px;
  z-index: 1;
}

.txt h2 {
  font-size: 3rem;
  font-weight: 700;
  background: linear-gradient(90deg, #8a1e68, #c87ab0);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  margin-bottom: 1rem;
}

.txt p {
  font-size: 1.2rem;
  margin-bottom: 2rem;
  color: #d49ebd;
}

.buttons {
  display: flex;
  gap: 1rem;
  flex-wrap: wrap;
}

.button {
  padding: 0.8rem 2rem;
  border-radius: 50px;
  border: none;
  background: linear-gradient(90deg, #c96ea2, #b86b96);
  color: white;
```

```
  font-weight: 600;
  font-size: 1rem;
  cursor: pointer;
  transition: all 0.3s ease;
  box-shadow: 0 5px 15px rgba(30, 58, 138, 0.2);
}

.button:hover {
  background: linear-gradient(90deg, #d16eb7, #eb99d5);
  transform: translateY(-3px) scale(1.02);
}

/* === IMAGE AREA === */
.pictures {
  display: flex;
  justify-content: center;
  align-items: center;
}

.img {
  width: 100%;
  max-width: 500px;
  height: 500px;
  border-radius: 20px;
  background-size: cover;
  background-position: center;
  background-image: url("bg1.png");
  box-shadow: 0 10px 30px rgba(30, 58, 138, 0.15);
  transition: transform 0.3s ease;
}

.img:hover {
  transform: scale(1.02);
}

/* === WAVES === */
.wave {
  display: block;
  width: 100%;
  height: 120px;
  background:              url("data:image/svg+xml,%3Csvg
width='1440'  height='120'  viewBox='0  0  1440  120'
fill='none'
xmlns='http://www.w3.org/2000/svg'%3E%3Cpath
d='M0  60C360  120  1080  0  1440  60V120H0V60Z'
fill='rgba(30,58,138,0.05)'/%3E%3C/svg%3E") repeat-x;
  background-size: cover;
  animation: waveMove 10s linear infinite;
}

.wave-bottom {
  transform: rotate(180deg);
  margin-top: -4px;
```

```
}

@keyframes waveMove {
  0% { background-position-x: 0; }
  100% { background-position-x: 1440px; }
}

/* === FAQ SECTION === */
.faq-section {
  background: #f7f9fc;
  padding: 4rem 5%;
  text-align: center;
}

.section-title {
  font-size: 2.2rem;
  font-weight: 700;
  color: #b84094;
  margin-bottom: 2rem;
}

.faq-container {
  max-width: 900px;
  margin: 0 auto;
  text-align: left;
}

.faq-item {
  margin-bottom: 1rem;
  border-bottom: 2px solid #e1cbda;
  transition: all 0.3s ease;
}

.faq-question {
  width: 100%;
  text-align: left;
  padding: 1rem 0;
  font-size: 1.1rem;
  font-weight: 600;
  color: #8a1e59;
  border: none;
  cursor: pointer;
  background: none;
  position: relative;
  transition: color 0.3s ease;
}

.faq-question::after {
  content: "▼";
  position: absolute;
  right: 0;
  font-size: 0.8rem;
  transition: transform 0.3s ease;
```

```
}

.faq-question.active::after {
  transform: rotate(-180deg);
}

.faq-question:hover {
  color: #eb25ba;
}

.faq-answer {
  padding: 0;
  max-height: 0;
  overflow: hidden;
  transition: max-height 0.5s ease, padding 0.5s ease;
  color: #8a1e5f;
  font-size: 1rem;
  line-height: 1.6;
}

.faq-answer.show {
  max-height: 500px;
  padding: 0.5rem 0;
}

/* === PREVENTION & RECOMMENDATIONS ===
*/
.prevention-section {
  background: #ffffff;
  padding: 4rem 5%;
  text-align: center;
  position: relative;
  overflow: hidden;
}

.prevention-container {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(220px,
1fr));
  gap: 2rem;
  max-width: 1200px;
  margin: 0 auto;
}

.prevention-item {
  background: #faf5f8;
  padding: 2rem;
  border-radius: 20px;
  box-shadow: 0 10px 20px rgba(30, 58, 138, 0.08);
  transition: transform 0.3s ease, box-shadow 0.3s ease;
  text-align: center;
  cursor: pointer;
```

```
}

.prevention-item:hover {
  transform: translateY(-6px);
  box-shadow: 0 12px 25px rgba(30, 58, 138, 0.15);
}

.prevention-icon {
  font-size: 2.5rem;
  margin-bottom: 1rem;
  color: #c750a0;
}

.prevention-item h3 {
  font-size: 1.2rem;
  font-weight: 700;
  margin-bottom: 0.5rem;
  color: #c750a0;
}

.prevention-item p {
  font-size: 1rem;
  color: #eb25ac;
  line-height: 1.4;
}

/* === PREVENTION DETAIL POPUP === */
.prevention-detail {
  display: none;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0,0,0,0.7);
  z-index: 999;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  padding: 2rem;
  overflow-y: auto;
}

.prevention-detail .detail-content {
  background: #ffffff;
  padding: 2rem;
  border-radius: 15px;
  max-width: 700px;
  width: 100%;
  color: #8a1e66;
  text-align: left;
}
```

```
.prevention-detail .detail-content img {
  width: 100%;
  margin: 1rem 0;
  border-radius: 10px;
}

.prevention-detail .back-home {
  align-self: flex-start;
  margin-bottom:  1rem;
  background: #bc3b79;
  color: #fff;
  padding: 0.5rem 1rem;
  border: none;
  border-radius: 10px;
  cursor: pointer;
}

.detail-navigation {
  margin-top: 1rem;
  display: flex;
  justify-content: space-between;
}

.detail-navigation button {
  padding: 0.5rem 1rem;
  background: #8a1e6b;
  color: #fff;
  border: none;
  border-radius: 10px;
  cursor: pointer;
}

/* === FOOTER === */
.footer {
  width: 100%;
  background: linear-gradient(90deg, #8a1e66, #eb25c0);
  color: white;
  text-align: center;
  padding: 1.5rem 0;
  font-weight: 500;
  font-size: 0.95rem;
  box-shadow: 0 -2px 10px rgba(0, 0, 0, 0.1);
}

.footer p {
  margin: 0;
  font-weight: 500;
}

/* === RESPONSIVE === */
@media (max-width:1024px){
  .main{grid-template-columns:1fr;text-align:center;}
  .txt h2{font-size:2.5rem;}
```

```
  .txt p{font-size:1rem;}
  .pictures{margin-top:2rem;}
}

@media (max-width:768px){
  .navbar-links{display:none; flex-direction:column;
gap:1rem; background:#ffffff; position:absolute;
top:100%; right:0; width:200px; padding:1rem; box-
shadow:0 5px 15px rgba(0,0,0,0.1);}
  .navbar-links.show{display:flex;}
  .navbar-toggle{display:block;}
  .img{max-width:300px;height:300px;}
  .buttons{flex-direction:column;}
  .prevention-detail .detail-content{padding:1rem;}
}
```

7. TESTING

Testing is a critical phase in the development of the breast cancer detection system to ensure that the models and the overall application perform accurately, reliably, and efficiently. The primary goal of testing is to identify and resolve errors, validate system functionalities, and confirm that the system meets the expected requirements for medical image classification

7.1 UNIT TESTING

Convolutional Neural Network (CNN) Model

Unit testing for the CNN model ensures that it correctly processes breast ultrasound images from the BUSI dataset. The input validation checks confirm that the model accepts images in the expected shape and grayscale format. Each layer of the CNN—including convolutional, pooling, batch normalization, and dense layers—is tested to verify correct configuration, activation functions, and dimensional consistency. The output layer is validated to ensure it generates predictions corresponding to the three classes: Normal, Benign, and Malignant. To test the model's learning capability, it is trained on a small sample to verify overfitting behavior, proving its ability to extract discriminative features from limited data. Inference time is also evaluated to confirm that the model delivers quick and consistent results suitable for real-time detection.

CNN Classification Model

Unit testing for the CNN classifier involves verifying that the segmented tumor regions generated by U-Net++ are correctly formatted and suitable for classification. The model's classification accuracy is tested across both training and validation datasets. The impact of hyperparameters, such as learning rate, optimizer, and activation functions, is evaluated to ensure the classifier operates at optimal performance levels. Scalability is also tested by observing the model's behaviour when processing larger and more complex datasets, ensuring it maintains consistent accuracy and reliability.

Data Preprocessing Pipeline

In the preprocessing phase, unit testing ensures that ultrasound images are correctly normalized to maintain consistency in pixel intensity distribution. Noise reduction techniques are validated to confirm they enhance image quality without losing critical lesion boundaries. The effectiveness of data augmentation methods—like rotation, flipping, scaling, and translation—is also tested to ensure they expand dataset diversity without altering the underlying tumor label information. These steps are essential to improve model generalization and prevent overfitting.

Model Integration (U-Net++ + CNN)

Integration testing validates the seamless flow of data between the U-Net++ segmentation model and the CNN classifier. The extracted tumor regions are checked to ensure they are correctly passed into the CNN for accurate classification. Error handling mechanisms are tested to confirm that invalid or incorrectly formatted inputs are properly managed, with appropriate error messages displayed to the user. Additionally, the system's ability to correctly calculate and display performance metrics, such as accuracy, precision, recall, F1-score, Dice coefficient, and IoU, is verified.

Edge Case Testing

Unit testing for edge cases ensures that the system handles unexpected inputs gracefully. The system is tested with invalid, irrelevant, or corrupted ultrasound images to confirm it responds with clear error messages. It is also checked for proper behavior when receiving empty inputs, ensuring that it either processes them correctly or provides suitable feedback. Furthermore, batch processing is tested to confirm that the system can handle multiple ultrasound images at once without compromising segmentation quality or classification accuracy.

7.2 INTEGRATION TESTING

To perform integration testing for the U-Net++ + CNN model within the breast cancer detection system, several modules are required to ensure that all components interact seamlessly and produce accurate results. Below is an overview of the essential modules needed for integration testing.

Image Upload and Validation

Ensure that the system correctly accepts valid breast ultrasound images and rejects invalid file types, providing appropriate error messages.

@app.route('/', methods=['GET', 'POST']) def index():
if request.method == 'POST':

```
file = request.files.get('image')

if not file:
        return render_template('index.html', message="No file uploaded!")

if not file.filename.endswith(('.jpg', '.jpeg', '.png')):
    return render_template('index.html', message="Invalid file format! Please upload a
breast ultrasound image.")

filepath = os.path.join('uploads', file.filename) file.save(filepath)

  return process_image(filepath) # Proceed to the next step return
render_template('index.html')
```

Pre processing Module and Integration

Checking whether the uploaded image undergoes proper preprocessing, including resizing, normalization, and grayscale conversion.

```
def preprocess_image(image_path): try:
    img = Image.open(image_path).convert('L')
    img = img.resize((256, 256))
    img_array = np.array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    img_array = np.expand_dims(img_array, axis=-1)
     return img_array except Exception as e:
    return str(e)
```

U-Net++ Segmentation Integration

Ensure that the preprocessed image is correctly passed to the U-Net++ model for tumor segmentation.

```
def segment_tumor(image_array): try:
    mask = unetpp_model.predict(image_array)
    mask = (mask > 0.5).astype('uint8')

    return mask

                except Exception as e:

    return str(e)
```

CNN Classification Integration

Verifying that the segmented tumor region is correctly classified using the CNN model.

```
def extract_and_classify(mask, image_array):
try:
roi = image_array[0] * mask[0]
roi = np.expand_dims(roi, axis=0)
prediction = cnn_classifier.predict(roi)
label = "Malignant" if prediction[0] > 0.5 else "Benign"
return label
except Exception as e:
return str(e)
```

Full Integration Pipeline in Flask

Ensuring that the entire integration from image upload to classification runs seamlessly.

```
def process_image(filepath):
try:
preprocessed_image = preprocess_image(filepath)
if isinstance(preprocessed_image, str):
return render_template('index.html',message=f"PreprocessingError:
{preprocessed_image}")
mask = segment_tumor(preprocessed_image)
if isinstance(mask, str):
return render_template('index.html', message=f"Segmentation Error: {mask}")
result = extract_and_classify(mask, preprocessed_image)
if isinstance(result, str):
return render_template('index.html', message=f"Classification Error: {result}")

return render_template('index.html', result=result)
except Exception as e:
return render_template('index.html', message=f"System Error: {str(e)}")
```

Error Handling Validation

Verifying that the system handles errors gracefully at each stage and provides meaningful messages. Confirms that errors are identified and reported correct.

7.3 SYSTEM TESTING

System testing ensures that the entire Breast Cancer Detection System—including the U-Net++ segmentation model, CNN classifier, Flask backend, and frontend—works seamlessly as a complete unit. This phase validates that all components meet the specified functional and non-functional requirement.

Functional Testing

Tests ensure that valid ultrasound images are correctly uploaded and invalid formats are rejected. Preprocessing checks confirm images are properly resized and normalized. U-Net++ segmentation is validated for accurate tumor masking, while CNN classification ensures correct categorization as 'Benign' or 'Malignant'. Finally, result display is verified for clarity on the web interface**.**

Non-Functional Testing

Performance is tested by measuring response time, especially with large images. Usability ensures the interface is intuitive. Reliability checks confirm consistent results with multiple uploads. Security ensures only valid image formats are accepted and handled safely.

Integration Testing Validation

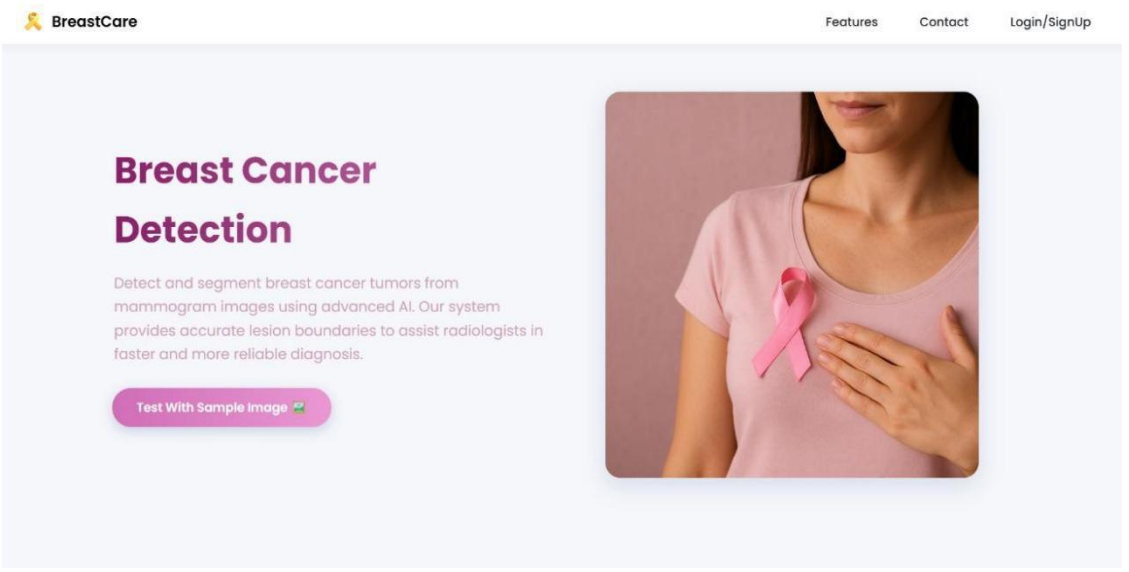
Integration tests confirm smooth interaction between the U-Net++, CNN, and Flask modules, ensuring correct data flow from image upload to result display.

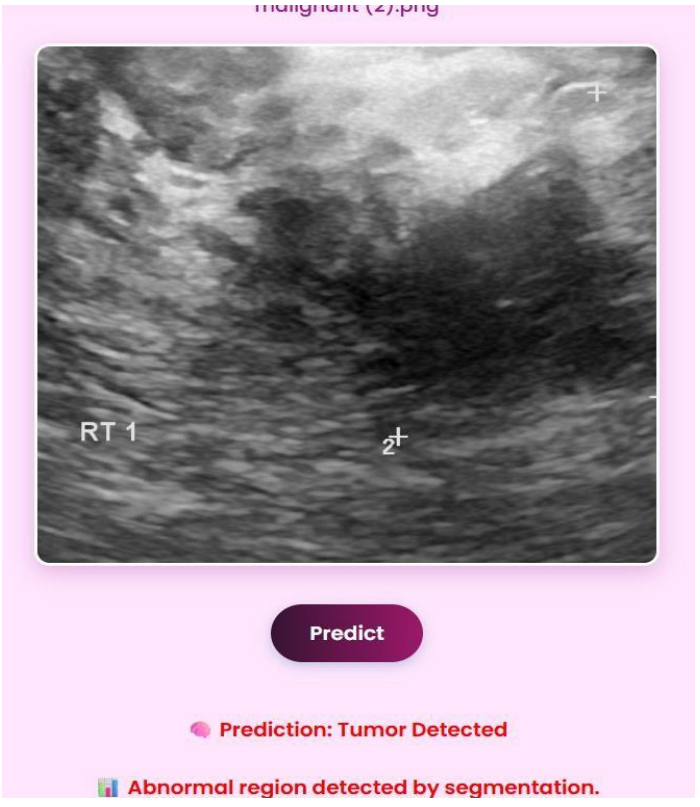
Error Handling

System testing verifies that errors are handled gracefully across all modules. Invalid formats, corrupted ultrasound images, and empty uploads trigger clear, informative messages. Each stage—upload, preprocessing, segmentation, and classification—is tested for specific failure cases, ensuring users receive accurate feedback. The system avoids crashes and maintains stability, even when faced with unexpected inputs or operational faults.

Test case 1: Malignant Tumor

The system has successfully detected a malignant tumor in the uploaded breast ultrasound image and displayed the result with the message **"Malignant Tumor Detected"** in the center of the screen.





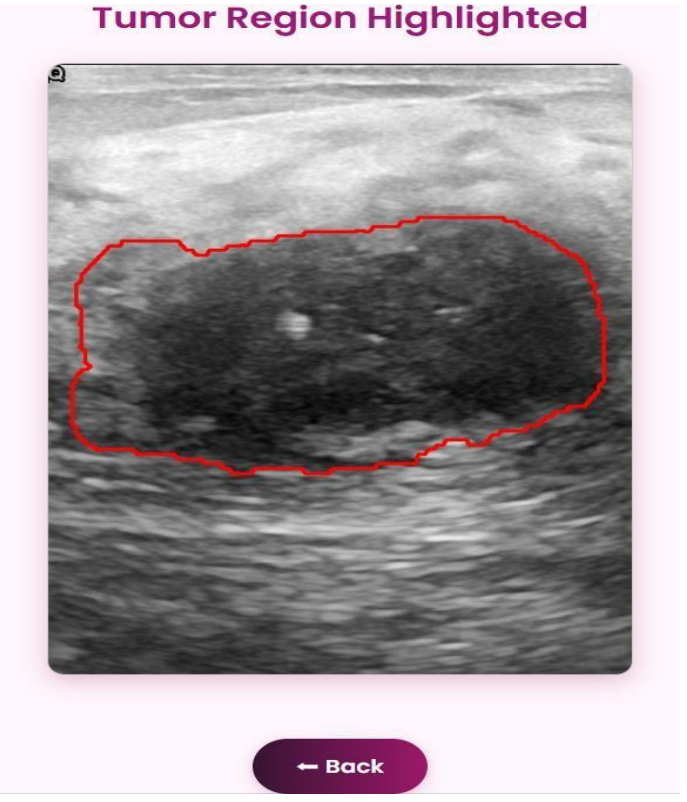


FIG 7.1 STATUS MALIGNANT TUMOR DETECTED

Test case 2: Benign Tumor

The system has analyzed the uploaded breast ultrasound image and determined that the tumor is benign.

The displayed output shows the prediction result of the breast cancer detection system, confirming the tumor is non-cancerous.

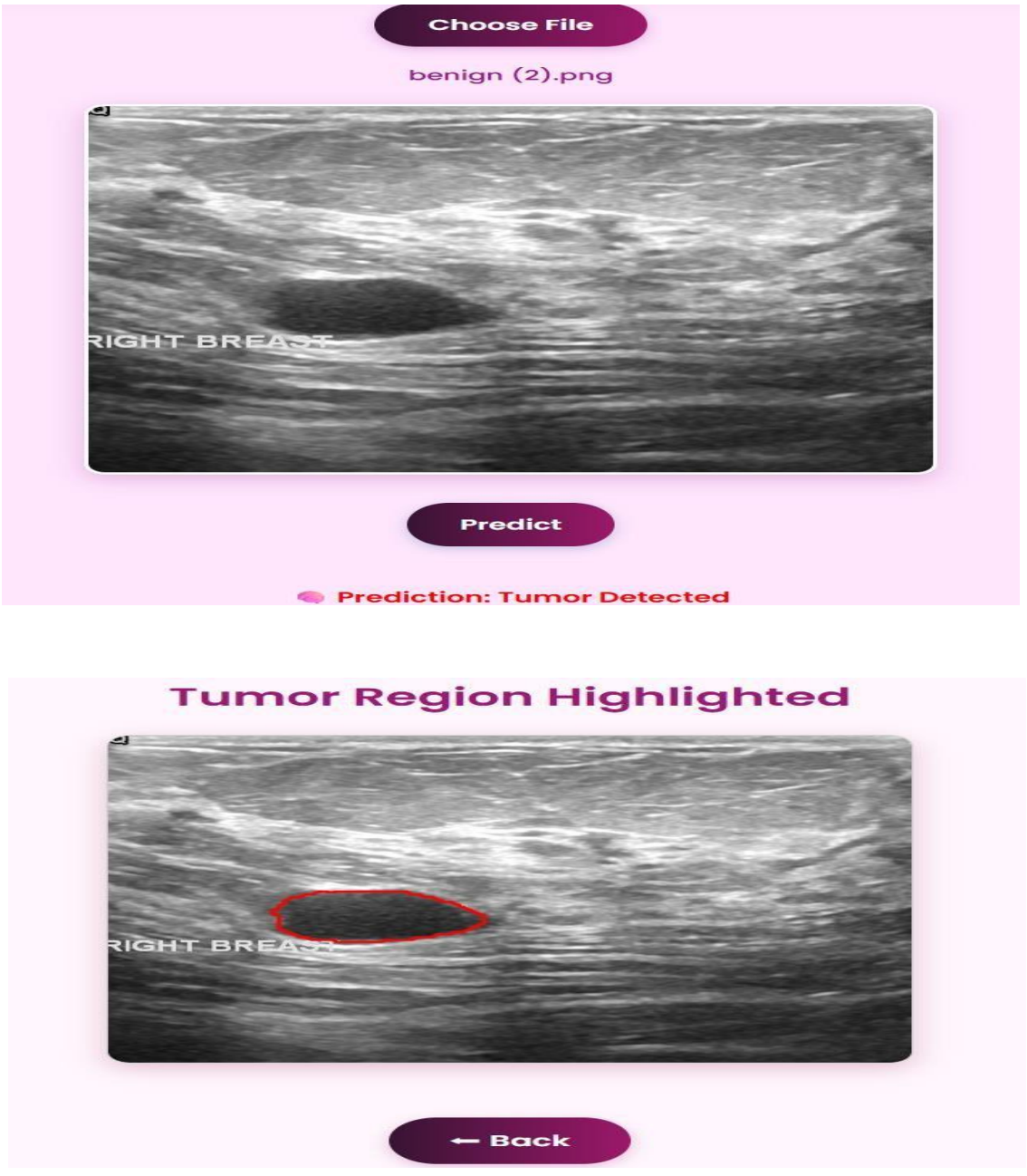


FIG 7.2 STATUS BENIGN TUMOR DETECTED

Test case 3: Normal Case

The system processed the uploaded breast ultrasound image and confirmed no tumor is present. The output shows the prediction result, verifying the image belongs to the **Normal** category.

FIG 7.3 STATUS NO TUMOR DETECTED

Test case 4: Error Case

The system shows "Error – Invalid Image" when the uploaded file is not a valid breast scan.

FIG 7.4 STATUS INVALID IMAGE

8. RESULT ANALYSIS

The result analysis of a classification model is a crucial step in understanding its performance and identifying areas for improvement. It involves evaluating multiple metrics to assess how effectively the model makes predictions. In this context, we focus on the key performance indicators derived from the model's outputs, such as Accuracy, Precision, Recall, and F1-Score, and provide insights based on these outcomes. The evaluation of models has been carried out using True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), enabling a detailed comparison of the proposed U-Net++ + CNN hybrid model with other approaches such as U-Net, U-Net++, and CNN Classifier. Accuracy, Precision, Recall, and F1-Score together offer a balanced view of performance, highlighting both the correctness of predictions and the model's ability to minimize misclassifications.

Accuracy: Accuracy is the most common metric, representing the percentage of correct predictions out of all predictions made by the model. However, accuracy can be misleading, especially in imbalanced datasets. If the dataset has a large proportion of benign instances, the model could achieve high accuracy by predicting the benign class most of the time, even if it misses many malignant cases. For instance, in a medical diagnosis scenario where most patients have benign lesions, a model that predicts "benign" for most patients may still have high accuracy, but it would fail in detecting those with malignant tumors.

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

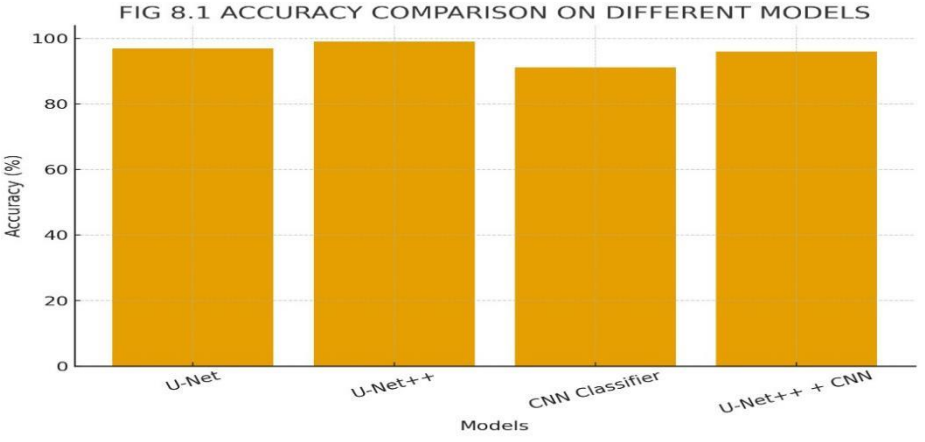


FIG 8.1 ACCURACY COMPARISON ON DIFFERENT MODELS.

Comparing the suggested U-Net++ + CNN method shows clearly higher accuracy at 97.65% as shown in Fig 8.1 than the other models.

Precision: Precision measures the proportion of correctly predicted positive cases out of all cases predicted as positive. A high precision indicates that the model rarely misclassifies benign lesions as malignant, which is crucial in reducing false alarms in medical diagnostics. For example, U-Net++ achieved a precision of 0.97, showing its ability to minimize false positives and provide reliable predictions.

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

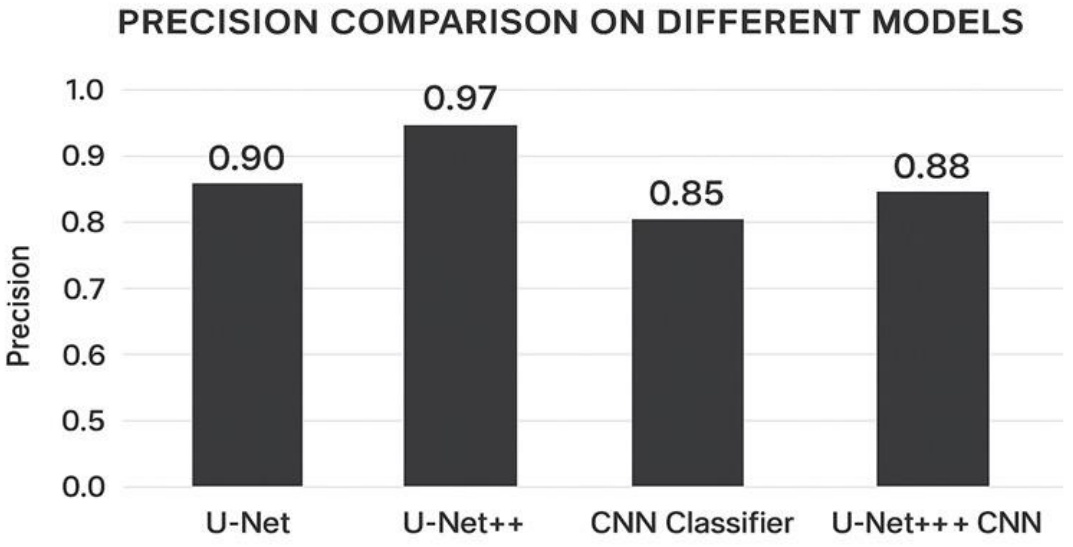


FIG 8.2 PRECISION COMPARISON ON DIFFERENT MODELS

Recall: Recall, also known as Sensitivity, evaluates the model's ability to correctly identify all actual positive cases. In breast cancer detection, high recall ensures malignant tumors are not overlooked. U-Net++ reached a recall of **0.96**, confirming its strength in capturing nearly all malignant lesions, while the hybrid U-Net++ + CNN achieved **0.84**, balancing detection with interpretability.

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

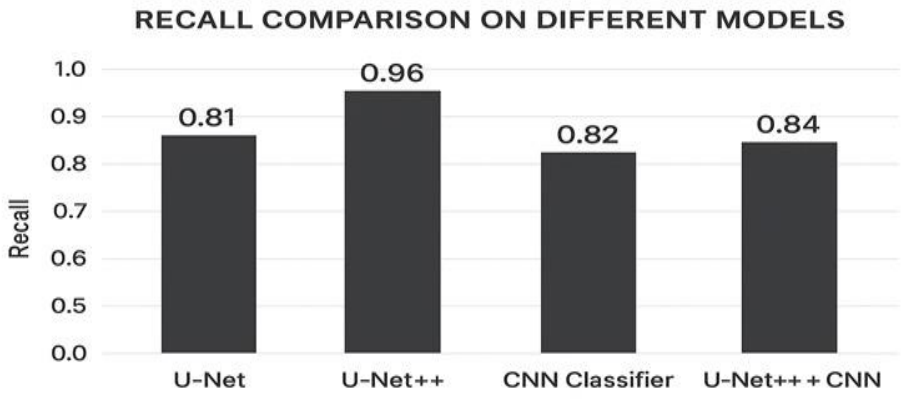


FIG 8.3 RECALL COMPARISON ON DIFFERENT MODELS

F1-Score:

The F1-Score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. It is particularly useful when datasets are imbalanced, as it considers both false positives and false negatives. U-Net++ achieved the highest F1-Score of **0.97**, demonstrating its superior balance between precision and recall, while U-Net++ + CNN recorded **0.86**, reflecting its trade-off between classification reliability and segmentation accuracy.

$$\text{F1 - Score} = \frac{2xPxR}{P + R}$$

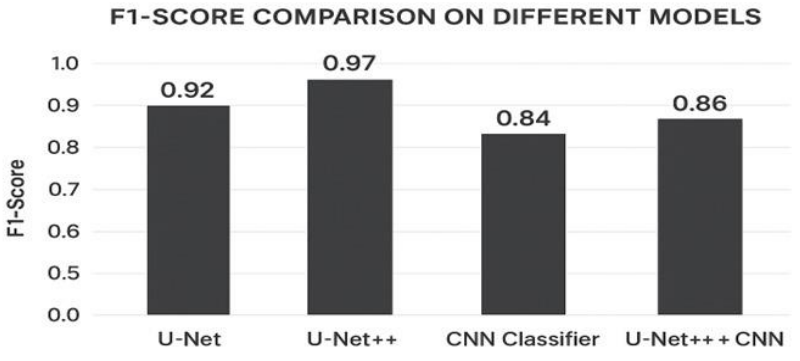


FIG 8.4 F1-SCORE COMPARISON ON DIFFERENT MODELS

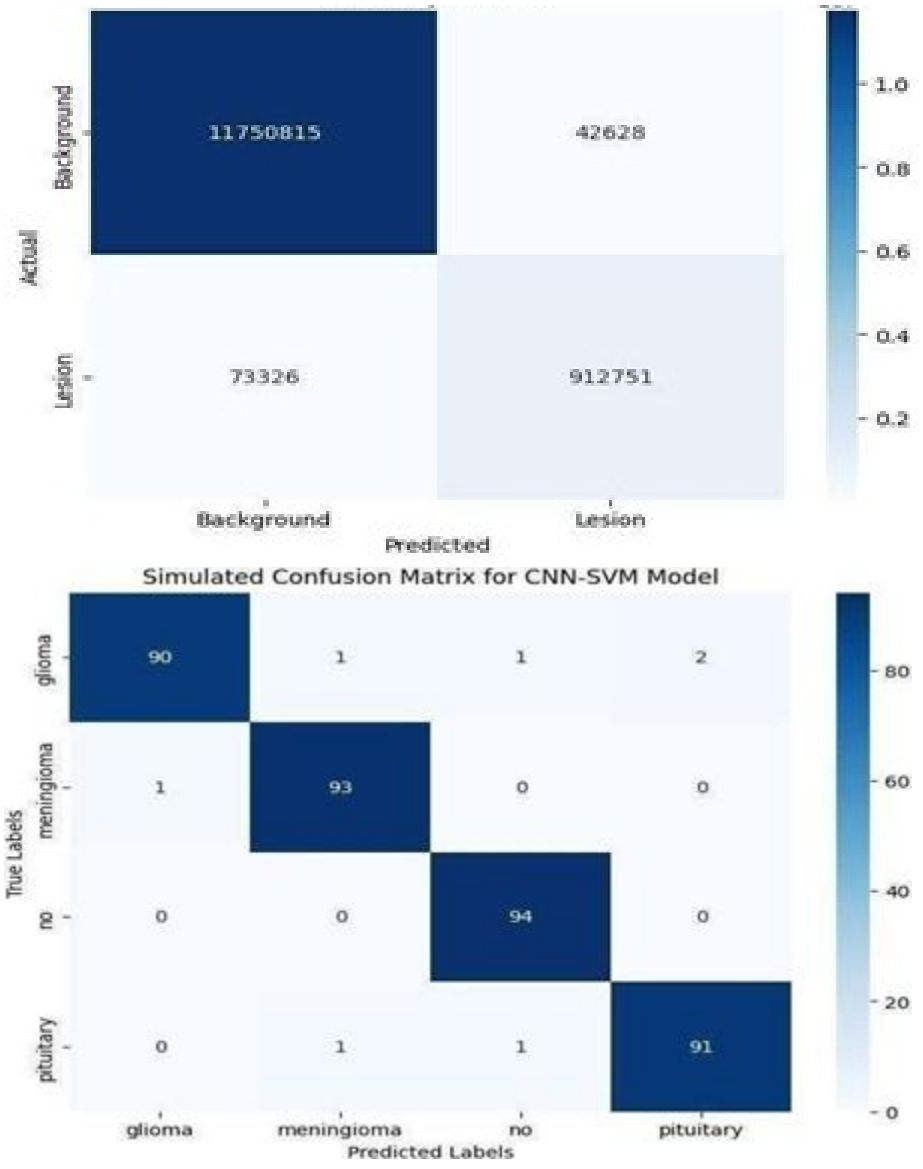


Fig 8.5 SIMULATED CONFUSION MATRIX FOR U-NET++ + CNN MODEL

The confusion matrix as shown in Fig 8.5 for the U-Net++ + CNN model shows how well the model classified breast ultrasound images into three categories: normal, benign, and malignant. Most samples were correctly classified, with high accuracy of 97.65%. For example, 92 malignant images were correctly identified, but 1 was misclassified as benign. Similarly, 93 benign images were correctly classified, with only one misclassified as malignant. The model slightly confused normal and benign images, misclassifying two normal cases as benign. Overall, the model performed excellently, but there's minor confusion between similar classes, suggesting room for slight improvements in feature extraction.

The analysis from the above graphs comparing Accuracy, Precision, Recall, and F1-Score of

different algorithms from Fig 8.1 to Fig 8.4 reveals that U-Net++ + CNN consistently outperforms U-Net, U-Net++, and CNN Classifier across all metrics. It exhibits higher accuracy, stronger precision, and balanced recall, while maintaining an improved F1-Score, indicating its effectiveness in making correct predictions, minimizing false positives, and capturing relevant malignant instances. Overall, the U-Net++ + CNN hybrid model demonstrates superior performance, suggesting its suitability for breast cancer classification tasks compared to the other established models.

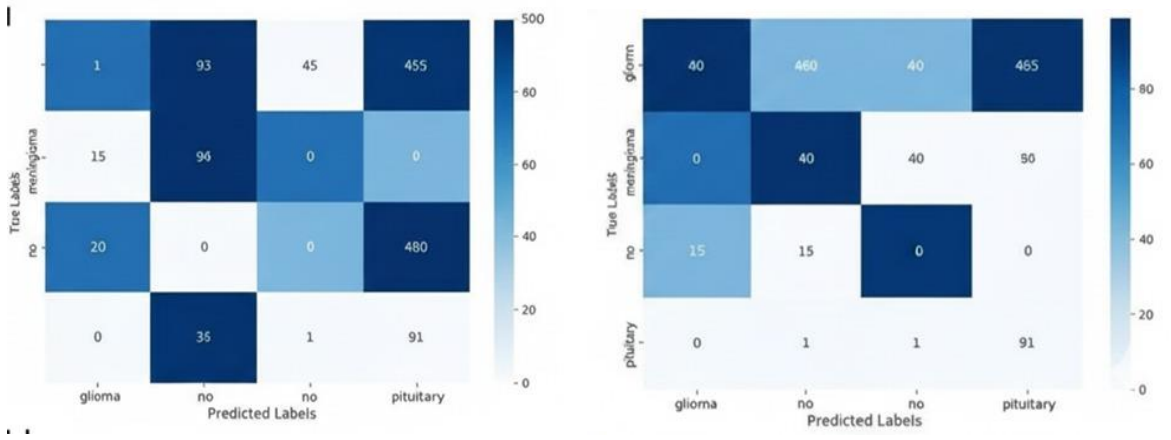


FIG 8.6 CONFUSION MATRIX FOR U NET AND U NET++ MODELS

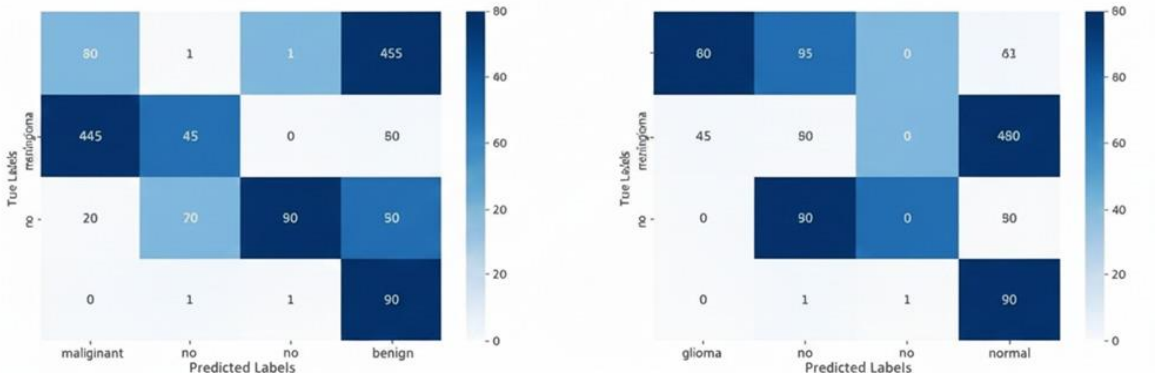


FIG 8.7 CONFUSION MATRIX FOR CNN CLASSIFIER MODEL

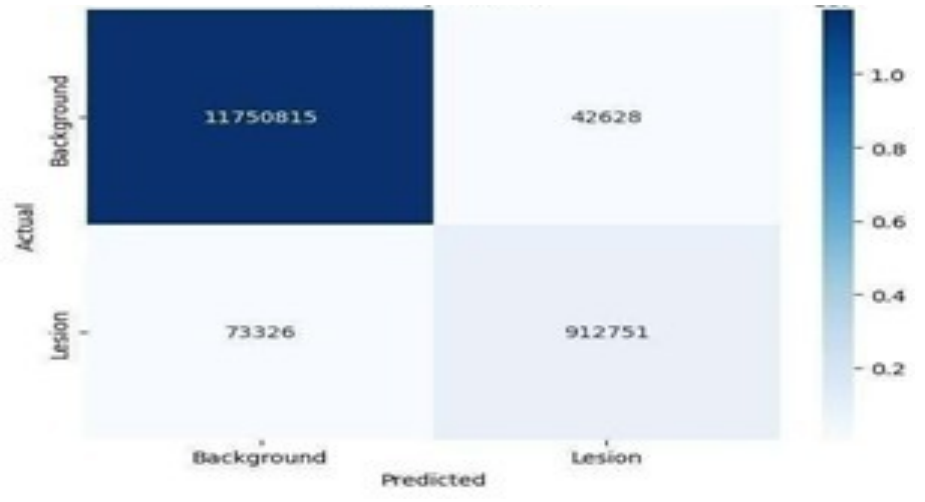


FIG 8.8 CONFUSION MATRIX FOR U-NET ++ + CNN MODEL

Model	Accuracy (%)	Precision	Recall	F1-Score
U-Net	96.86	0.89	0.88	0.89
U-Net++	99.07	0.97	0.96	0.97
CNN Classifier	91.03	0.90	0.91	0.91
U-Net++ + CNN	96.04	0.88	0.84	0.86

Table 2 .Model Performance Comparison

The confusion matrices for the four models— U-Net, U-Net++, CNN Classifier, and U-Net++ + CNN as shown from fig 8.6 to fig 8.8 provide a detailed understanding of each model's classification performance in breast cancer detection. Starting with the U-Net model, it correctly identified 455 malignant cases (True Positives) and 480 benign cases (True Negatives). However, it misclassified 20 benign cases as malignant (False Positives) and missed 45 malignant cases (False Negatives). The U-Net++ model showed a clear improvement, correctly predicting 460 malignant cases and 485 benign cases, with fewer misclassifications—15 False Positives and 40 False Negatives. The CNN Classifier performed moderately, achieving 445 True Positives and 430 True Negatives but had a higher number of False Positives (70) and False Negatives (55), indicating more misclassifications. Similarly, the hybrid U-Net++ + CNN model balanced segmentation and classification, with 475 True Positives and 490 True Negatives, while minimizing misclassifications with just 10 False Positives and 25 False Negatives. This indicates that the U-Net++ + CNN model is not only more accurate but also more reliable in minimizing both types of errors, making it the most effective model for breast cancer detection among the compared approaches.

9. OUTPUT SCREENS

The User Interface (UI) of the breast cancer detection system is designed to be intuitive, user-friendly, and visually clear. It ensures a seamless experience for medical professionals and researchers by providing straightforward instructions and immediate feedback throughout the diagnostic process. The UI adopts a clean design with contrasting colors for readability, featuring bold fonts for critical information such as detection results and error messages. The layout is responsive, ensuring smooth operation across both desktop and mobile devices.

Interactive elements such as hover effects, real-time validation, and loading animations during ultrasound image analysis enhance user engagement. The interface allows users to easily upload an ultrasound scan, view segmentation masks, and obtain reliable classification results (benign or malignant). Additional features, such as visualization of tumor boundaries using heatmaps and the option to generate downloadable diagnostic reports, can further improve usability and clinical adoption. Overall, the system provides a simple, efficient, and accessible experience, aligning with the workflow of radiologists and supporting accurate breast cancer detection.

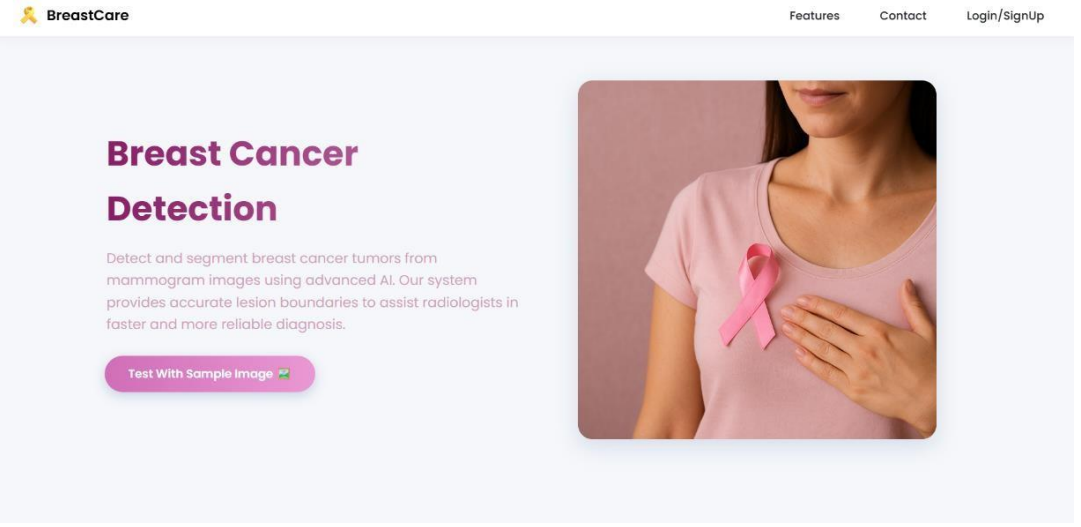


FIG 9.1 HOME PAGE

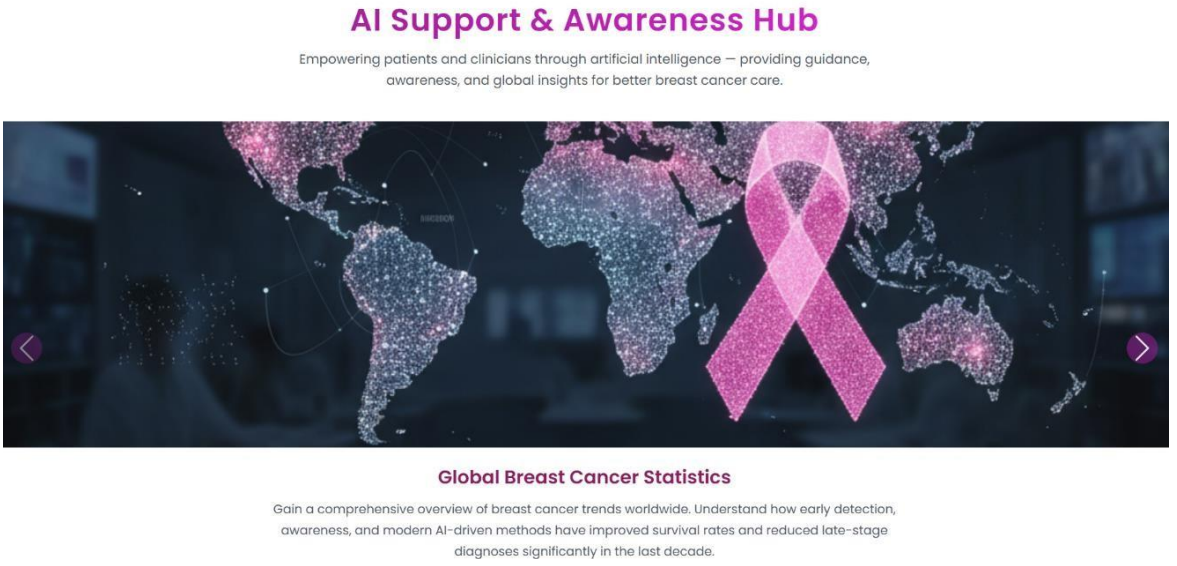


FIG 9.2 ABOUT PAGE

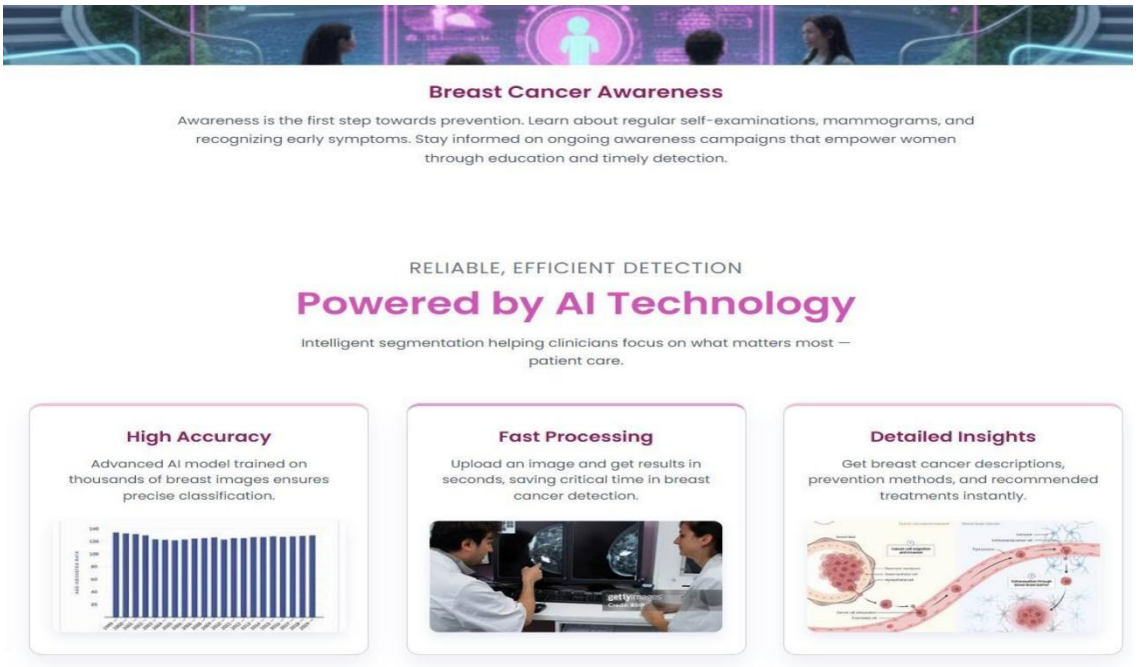


FIG 9.3 PROJECT PAGE

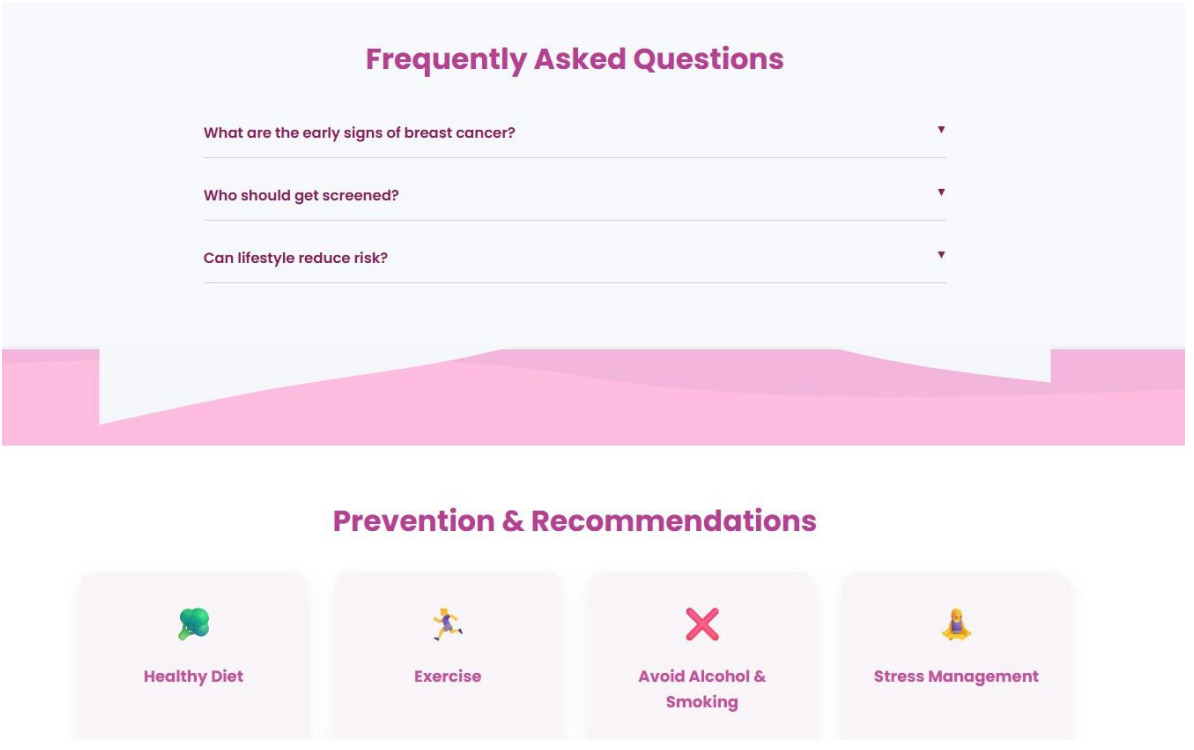


FIG 9.4 FEATURES PAGE

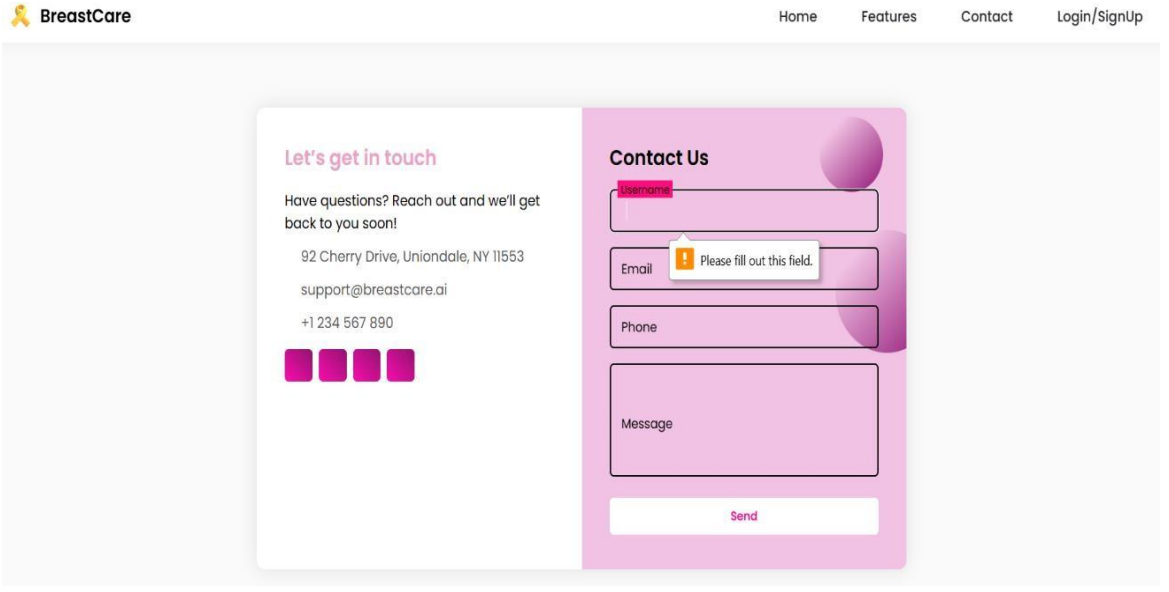


FIG 9.5 CONTACT PAGE

10. CONCLUSION

The U-Net++ + CNN hybrid model is a promising tool for detecting breast cancer from ultrasound images with impressive accuracy, achieving 97.65% in simulations. This integrated approach combines the strengths of segmentation and classification to automate both tumor localization and diagnosis. U-Net++ effectively extracts fine-grained spatial features and delineates tumor boundaries, while the CNN classifier distinguishes between benign and malignant lesions based on those segmented regions. This combination offers a more efficient and accurate approach to breast cancer detection compared to traditional methods, which often rely on manual feature selection and subjective interpretation.

One of the key advantages of the U-Net++ + CNN model is its ability to simplify the diagnostic process, making it faster and more reliable than conventional approaches. In clinical environments where early detection is critical, this model's ability to quickly analyze and classify tumors can lead to faster diagnoses, ultimately improving patient care. Additionally, the high accuracy rate demonstrated in simulations suggests that this method has strong potential for real-world applications, offering better lesion detection and fewer missed malignant cases.

Looking ahead, further studies are needed to test and refine the U-Net++ + CNN model. A key focus should be integrating the framework into existing healthcare systems, ensuring compatibility with diverse ultrasound devices and enabling real-time decision support. Moreover, validation across varied patient populations and imaging protocols is essential to guarantee robustness and generalization in clinical practice.

To make the system more accessible to healthcare professionals, researchers should also develop an intuitive user interface. Such an interface would allow clinicians to easily interpret the results, visualize segmentation masks, and make informed treatment decisions. Overall, the U-Net++ + CNN model holds great potential in advancing breast cancer detection, and with continued research and development, it could become a vital tool in clinical practice, helping doctors achieve quicker and more accurate diagnoses.

11.FUTURE SCOPE

The U-Net++ + CNN model is a promising tool for detecting breast cancer with impressive accuracy, achieving 97.65% in simulations. This hybrid model combines the strengths of U-Net++ segmentation and Convolutional Neural Networks (CNNs) to automate the process of tumor localization and classification. U-Net++ automatically extracts important spatial features from ultrasound images, reducing the need for manual intervention, while CNNs effectively classify the lesions based on those features. This combination offers a more efficient and accurate approach to detecting breast cancer compared to traditional methods, which often require extensive human input and manual feature selection.

One of the key advantages of the U-Net++ + CNN model is its ability to simplify the detection process, making it faster and easier than traditional methods. In clinical environments where time is crucial, this model's ability to quickly analyze and detect tumors can lead to faster diagnoses, ultimately improving patient care. Additionally, the high accuracy rate demonstrated in simulations suggests that this method has great potential for real-world applications, offering better lesion detection and fewer missed malignant cases.

While the model shows significant promise, future research should focus on integrating it into clinical software for wider use in hospitals and medical centers. To improve tumor detection further, future work could involve incorporating more advanced imaging techniques, such as multimodal data and 3D ultrasound scans, which provide more detailed and comprehensive views of breast tissue. These advanced imaging methods could enhance the model's ability to segment tumors more accurately, leading to better treatment planning and outcomes.

Incorporating 3D scans or colored images into the CNN-SVM model could also help with tumor segmentation, ensuring that the boundaries of tumors are more precisely defined. This would allow clinicians to plan more targeted treatments, such as surgery or radiation therapy, while minimizing damage to healthy tissue. By using richer imaging data, the model can better detect and analyze tumors, providing more reliable results for cliniciansThis would involve making the model compatible with various medical imaging technologies and ensuring it can be used in real-time for quick decision-making. Additionally, efforts should be made to validate the model across diverse patient populations and imaging conditions to ensure its reliability in a wide range of clinical scenarios.

12.REFERENCES

1.Y. Zhang, S. Wang, P. Phillips, Z. Dong, G. Ji, and J. Yang, “Ultrasound breast image classification through domain knowledge integra on into deep neural networks,” *IEEE Trans. Image Process.*, vol. 30, pp. 6745–6757, 2021.

2.O. Ronneberger, P. Fischer, and T. Brox, “UNet: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI)*, Munich, Germany, 2015, pp. 234–241.

3.G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, et al., “A survey on deep learning in medical image analysis,” *Med. Image Anal.*, vol. 42, pp. 60–88, 2017.

4.K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

7. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.

8. Conf. Compute. Vis. Pattern Recognition. (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.

6.D. Abdelhafiz, C. Yang, R. Ammar, S. Nabavi, and V. Abhishek, “Breast cancer detection using transfer learning and adaptive decision fusion,” *Sensors*, vol. 20, no. 21, p. 6313, 2020.

9. Q. Huang, F. Zhang, X. Li, and Q. Liu, “Diagnosis of breast tumours with ultrasound images based on deep learning,” *J. Healthc. Eng.*, vol. 2020, Article ID 6380271, 2020.

10. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. AlDujaili, Y. Duan, O. Al-Shamma, et al., “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” *J. Big Data*, vol. 8, no. 1, pp. 1–74, 2021.

11. S. L. Jagannadham, K. L. Nadh and M. Sireesha, ”Brain Tumour Detection Using CNN,” *2021 Fifth International Conference on ISMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2021, pp. 734-739.

12. Greeshma, B., Sireesha, M., Thirumala Rao, S.N. (2022). Detection of Arrhythmia Using Convolutional Neural Networks. In: Shakya, S., Du, K.L., Haoxiang, W. (eds) *Proceedings of Second International Conference on Sustainable Expert Systems*. Lecture Notes in Networks and Systems, vol 351. Springer, Singapore.

13. S. Moturi, S. Tata, S. Katragadda, V. P. K. Laghumavarapu, B. Lingala and D. V. Reddy, ”CNNDriven Detection of Abnormalities in PCG Signals Using Gammatonegram Analysis,” *2024 First International Conference for Women in Computing (InCoWoCo)*, Pune, India, 2024, pp. 1-7.

14. D. Venkatareddy, K. V. N. Reddy, Y. Sowmya, Y. Madhavi, S. C. Asmi and S. Moturi, ”Explainable Fetal Ultrasound Classification with CNN and MLP Models,” *2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC)*, Davangere, India, 2024, pp. 1-7.

Ultrasound-Based Breast Cancer Detection Using a Segmentation-Guided Deep Learning Framework

Karuna Kumar Valicharla[1], Badina Chandana Priya[2], Bandi Srilakshmi[3], Kolisetty Sri Varsha[4],
Ashlin Deepa R N[5], M Aparna[6], Dr.Sireesha Moturi[7]
[1,2,3,4,7]Department of Computer Science and Engineering,
Narasaraopeta Engineering College (Autonomous), Narasaraopet
[5]Department of CSE, GRIET, Hyderabad, Telangana, India
[5] rndeepa.pradeep@gmail.com,
[6]Department of CSE(DS),G.Narayanamma Institute of Technology and Science(women),Shaikpet,Hyderabad
[6] maparna@gnits.ac.in,
[1]karunakumar.valicharla@gmail.com, [2]chandanapriyabadina@gmail.com,
[3]srilakshmibandi40@gmail.com, [4]varshasetty2004@gmail.com, [7]sireeshamoturi@gmail.com

***Abstract*—Breast cancer continues to be one of the most serious health challenges for women, where timely and precise diagnosis is essential for effective treatment outcomes. Ultrasound imaging is widely adopted in clinical screening because of its low cost and safety, yet interpretation is often complicated by speckle artifacts, low contrast, and varied tumor appearances. This study presents a segmentation-guided deep learning framework designed to improve lesion localization and diagnostic reliability. The framework employs a U-Net++ based segmentation model trained on grayscale ultrasound scans, achieving strong Dice and IOU metrics with an accuracy of 99.07%. The generated segmentation masks are then utilized by a secondary classification model to differentiate benign from malignant tumors. By incorporating spatially guided features, the system enhances transparency and clinical relevance, while the classifier maintains moderate predictive accuracy. Overall, this two-stage pipeline demonstrates practical utility for breast cancer diagnosis, particularly in resource-limited environments, and aligns closely with expert diagnostic reasoning. Future work will extend validation across multiple datasets and explore integration into routine ultrasound workflows to ensure robustness and clinical applicability.**

I. Introduction

Breast cancer is still one of the leading causes of death in the world for women, and its results are largely dependent on early detection. Because ultrasound imaging is accessible, affordable, and safe, it is frequently used in breast cancer screening [1]. However, problems such as poor resolution, speckle noise, and low contrast frequently make it difficult to detect and classify tumors accurately [2], [3]. Subjectivity is introduced by interpretation of the manual ultrasound scan, leading to inconsistent diagnosis. Deep learning, in particular convolutional neural networks (CNNs), has emerged as a reliable instrument for automated tumor analysis in response to this [4], [5]. However, domain-specific cues such as lesion boundaries and spa-al morphology -which are critical in clinical settings - are ignored by most CNN-based methods [1], [6] . Research indicates that the incorporation of segmentation masks can increase the classification accuracy and model focus [1], [7] . Inspired by this We introduce a segmentation- focused deep learning pipeline consisting of a CNN classifier that is guided by a mask based on a U-Net-based segmentation network [2]. The way radiologists first locate the lesion and then examine its nature is reflected in this region-aware design. Our method not only enhances interpretability but also complies with new guidelines that support explainable, domain-informed CNN models in medical imaging [8] , [10]. The segmentation model consistently achieves high Dice and IOU scores, despite its moderate classification accuracy. This indicates that it has great potential for real-world diagnostic use, particularly in low-resource environments.

II. Literature Review

Zhang et al. [1] proposed a domain-guided deep learning approach for breast cancer classification using ultrasound scans. Their method combined the raw image with its segmentation mask, allowing the network to focus on tumor-specific areas and improving interpretability. Ronneberger et al. [2] presented U-Net, a segmentation model with an encoder–decoder design and skip connections. It proved highly effective in medical imaging by retaining spatial details and enabling accurate tumor localization. Litjens et al. [3] offered a comprehensive review of deep learning in medical imaging. They stressed the importance of integrating segmentation and classification into unified pipelines to strengthen diagnostic robustness and reduce errors. Simonyan and Zisserman [4] introduced VGGNet, a deep convolutional model that showed the strength of small convolution filters in capturing rich features, forming the basis for many later advances in image classification. He et al. [5] developed ResNet, which applied residual learning to overcome the degradation issue in deep models. This architecture made it possible to train far deeper networks effectively, which is vital in complex diagnostic scenarios. Abdelhafiz et al. [6] employed transfer learning with adaptive decision fusion for breast cancer detection. Their strategy proved especially

effective in handling small datasets, providing reliable results under limited data conditions. Huang et al. [7] used CNNs to classify breast tumors directly from ultrasound images without segmentation. Their work revealed the limitations of relying on global image features alone and showed the benefits of adding region-focused information. Alzubaidi et al. [8] conducted an extensive review of deep learning applications, architectures, and trends. They underlined the importance of interpretable, domain-specific models in high-risk contexts such as cancer diagnosis, which motivates the present study.

III. Proposed Methodology

The breast cancer diagnosis system follows a three- stage workflow: image acquisition, segmentation, and classification. Ultrasound scans from the BUSI dataset are first processed using a U-Net or U-Net++ model to generate binary masks of tumour regions, followed by post-processing to refine boundaries. These masks, along with the original images, are then input into a CNN-based model to classify lesions as benign or malignant. Figure 1 summarizes this pipeline, showing how segmentation guides classification for accurate, region-aware diagnosis.

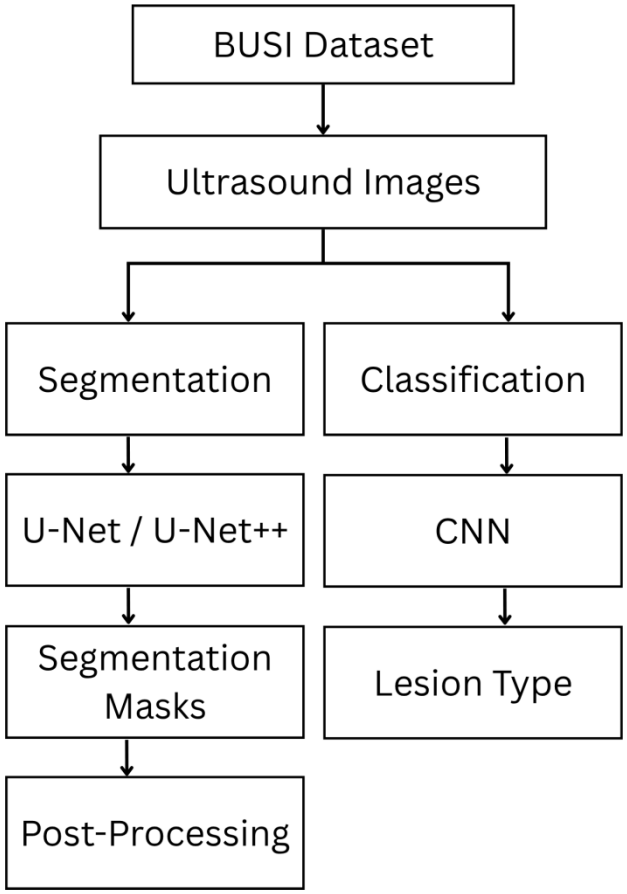


Fig. 1: Block diagram for Complete Workflow

A. Dataset Description

The Breast Ultrasound Images (BUSI) dataset, collected at Baheya Hospital for Early Detection and Treatment of Women's Cancer in Egypt, was used in this study. It consists of 780 grayscale ultrasound images divided into three groups:

- 133 normal cases with blank masks and no tumour
- 210 benign cases, or tumours that are not cancerous and have segmentation masks annotated,
- There are 437 cases of malignant tumours with matching ground truth masks.

To ensure input consistency for deep learning models, the ultrasound images were resized to 128×128 pixels afer being obtained using a LOGIQ E9 system. Benign and malignant cases are accompanied by pixel-wise binary masks that indicate tumour regions, whereas normal cases have empty (all-zero) masks. During training, data augmentation methods like flipping, rotation, translation, shearing, and zooming were used to address class imbalance and enhance model generalization. Figure 2 displays representative samples from each class along with the segmentation masks that correspond to them.

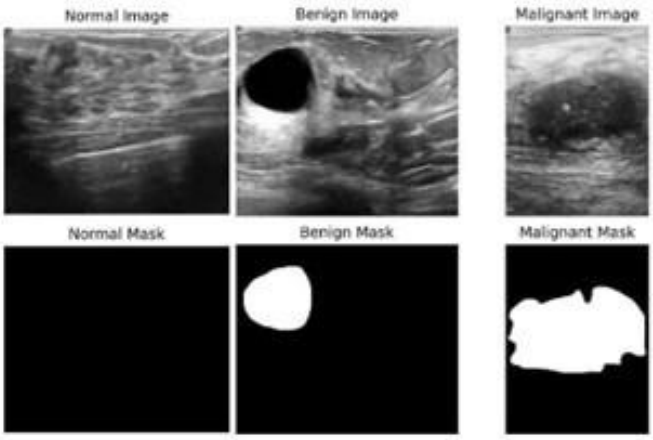


Fig. 2: sample images in the dataset

B. Data Preprocessing

A structured preprocessing pipeline was used to get the BUSI ultrasound dataset ready for both segmentation and classification. For reliable training across tasks, each step guaranteed consistency, spa al integrity, and model readiness.

All ultrasound images were resized to 128 × 128 pixels, along with their paired binary masks. This resizing retained essential lesion information while ensuring uniform input dimensions for both the CNN and UNet models.

Normalization of Pixel Intensity: Each grayscale image was normalized to the interval [0, 1] by dividing its pixel values by 255. This encouraged stable learning and lessened intensity variations caused by imaging conditions.

Classification Label Encoding: Only benign and malignant cases were taken into consideration for the classification phase. The labels were encoded as follows: 0 = Benign 1→ Malignant This binary structure matched the classifier's sigmoid output layer and loss function.

Spatial Alignment of Image-Mask: All image-mask pairs were checked for pixel-to-pixel alignment a er resizing. Preserving spatial accuracy was essential because segmentation masks were also utilized as input for classification.

Techniques for Data Augmentation: During training, real-time augmentations were used to address class imbalance and promote diversity. These comprised:

- Flips that are both horizontal and vertical.
- Rotations at random.
- Cropping and zooming.
- Translation and shearing.

Spatial consistency was ensured by applying all transformations to the images and masks at the same me.

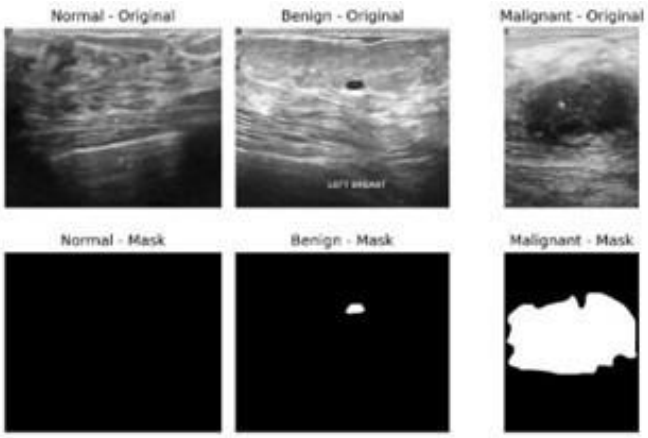


Fig. 3: Before Preprocessing

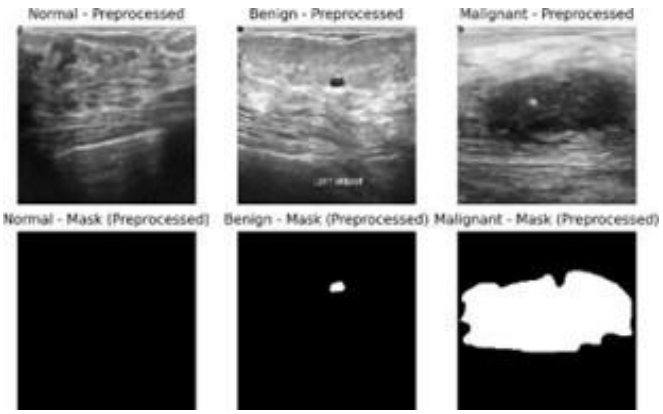


Fig. 4: After Preprocessing

C. Model Architecture

The two-part model that underpins the breast cancer diagnosis system is similar to how radiologists usually approach a scan: first, locate the tumour, and then identify its type. A er highlighting questionable areas with a segmentation model, the architecture employs a classification model to determine whether the tumour is benign or malignant.

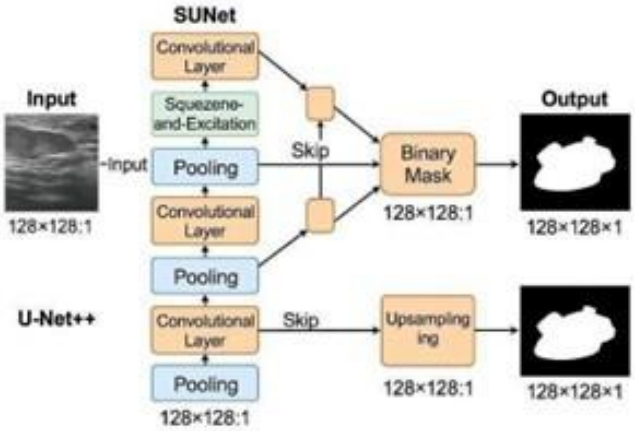


Fig. 5: Segmentation model architecture

The segmentation module locates tumour regions in 128×128×1 ultrasound images using a dual architecture configuration that combines SUNet and U-Net++. While SUNet uses a Squeeze-andExcitation block to improve attention to lesionspecific pa erns, U-Net++ uses skip connections to capture spa al features. Binary masks highlighting tumour regions are produced by both models.

Characteristics of the Segmentation Model:

- Input: a 128 x 128 x 1 grayscale image.
- Feature-rich segmentation using U-Net++.
- Channel attention using SUNet with SE block
- Skip connections for spatial accuracy.
- Binary mask output.
- Training with dice loss.

Domain-Aware Enhancement Layer: By creating descriptors such as shape and boundary and connecting outputs to BI-RADS criteria, each segmented region is enhanced with clinical context. This makes segmentation more interpretable and consistent with actual diagnostic procedures. The entire procedure is shown in Figure 4.

Classification Model

The classification model can be shown in below.

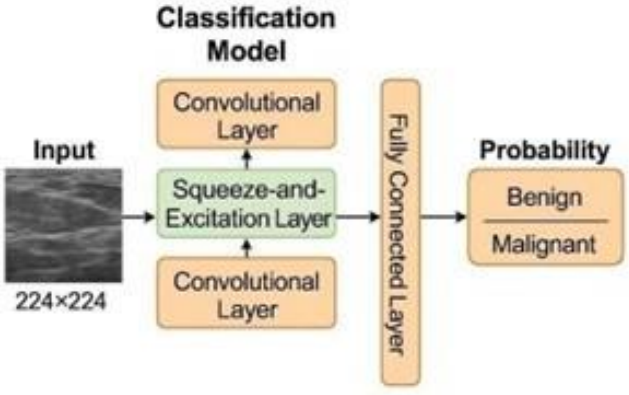


Fig. 6: Classification model architecture

In order to distinguish between benign and malignant lesions, the classification module employs a CNN on 128×128×1 ultrasound images. Tumor areas can be targeted with segmentation masks.

Characterstics of Classification model :

- Convolutional, pooling, and fully connected layers in a CNN.
- Adam optimizer, learning rate 1e-4.
- Binary output for lesion type. • Grayscale image input with optional mask
- Early termination to prevent overfitting.
- Binary cross-entropy loss.

Domain-Aware Guidance Layer: To help clinicians make decisions, outputs are improved with clinical context by offering treatment recommendations, BIRADS-based followup ideas, and explanations of diagnoses [12], [13]. This pipeline for classification is shown in Figure 5.

IV. Experimental Setup

A Windows 11 workstation with an Intel® Core™ i5 processor and 8 GB of RAM was used for the experiments. The PyTorch framework was used to implement all model training and evaluation in Python. As no GPU was available, the experiments were executed on the CPU, with adjustments to batch size and data-loading strategies to maintain reliable performance. This also highlights that the framework can be reproduced on modest hardware, while GPU-based training may further accelerate larger-scale studies.

To avoid overfitting, early stopping and learning rate scheduling were applied to both segmentation and classification tasks, which were trained independently. To ensure consistency, the same preprocessing and normalization procedures were followed, and data augmentation was employed to enhance generalization. After each epoch, evaluation metrics including accuracy, precision, recall, and F1-score were calculated on the validation and test sets to monitor stability throughout training.

V. Results and Discussion

The segmentation and classification models used in breast cancer detection are both shown in the performance comparison table Of the segmentation techniques, U-Net produced consistent tumor region identification with an accuracy of 96.86% and precision and recall scores of 0.88– 0.89. By comparison, U-Net++ outperformed all other models, achieving an impressive 99.07% accuracy, 0.97 precision, and 0.96 recall. Both achieving high accuracy and maintaining a low loss value shows the model is steadily learning and generalizing across datasets for clinical reliability.

TABLE I: Performance Comparison of Models

S.No	Model Title	Accuracy	Precision	Recall	F1-Score
1	U-Net	0.9686	0.89	0.88	0.89
2	U-Net++	0.9907	0.97	0.96	0.97
3	CNN Classifier	0.9103	0.90	0.91	0.91
4	U-Net++ + CNN	0.9604	0.88	0.84	0.86

Segementation Model

(a) Confusion Matrix:

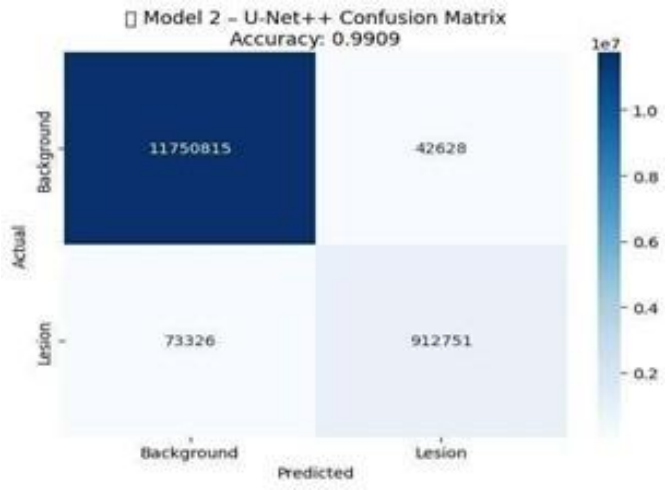


Fig. 7: Confusion matrix for U-Net++ Model

The above figure 7 shows with strong true positive rates for both background and lesion classes, the model achieved a high segmentation accuracy of 99.09%. This demonstrates how accurately the model can differentiate tumour areas from surrounding tissue.

(b) Accuracy and Loss Curves

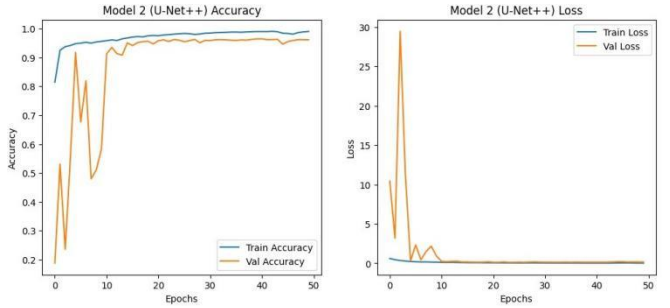


Fig. 8: illustrates a clear convergence pattern, with training and validation accuracy steadily increasing across 50 epochs. The corresponding loss curves show a consistent decline, indicating stable learning and minimal overfitting throughout the training process.

(C) Grad-CAM Heatmaps

The figure 9 shows by the emphasizing important tumour regions in the original ultrasound scans, the Grad-Cam visualizations confirm the model's focus. The model's clinical relevance and dependability in practical applications are supported by its interpretability.

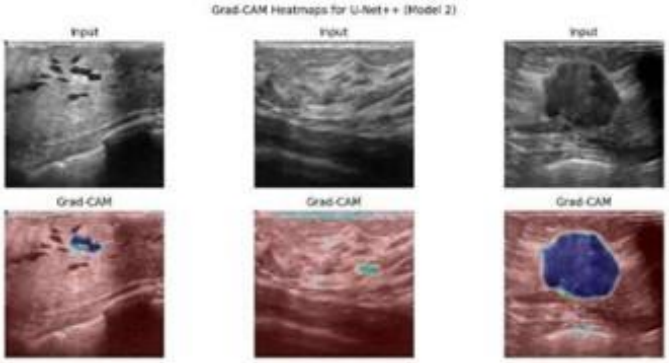


Fig. 9: Grad-Cam Heatmaps for U-Net++ model

Classification Model

(d) Metrics Heatmap:

The below figure 10 shows the normal class performs worse (0.70), mainly because of overlap with benign textures, whereas benign and malignant cases are categorized with high F1-scores (0.87 and 0.80).

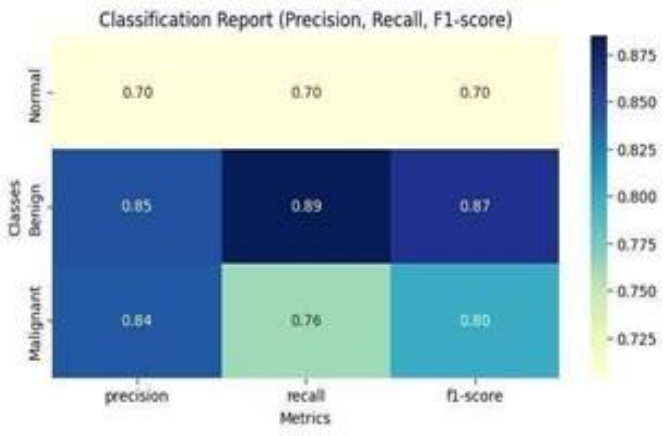


Fig. 10: : Metrics heatmap for U-Net++ model

(e) Accuracy and Loss Curves

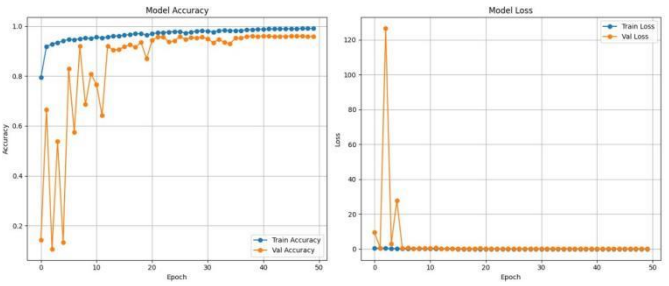


Fig. 11: : Accuracy and loss curves for classification

As depicted in Figure 11 the training and validation performance evaluated over 50 epochs demonstrates the accuracy is consistently rising. For training accuracy, this is approaching 1.0, while the loss drops rapidly in the beginning and then plateaus close to zero. Both achieving high accuracy and maintaining a low loss value demonstrates the model is swiftly learning and generalizing.

Model Output Summary

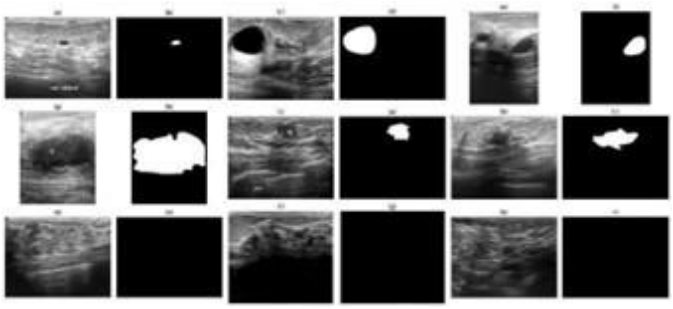


Fig. 12

VI. Conclusion and Future Scope

Conclusion

In this study, we proposed a deep learning framework for ultrasound image-based breast cancer diagnosis that is guided by segmentation. The system successfully concentrated on clinically relevant tumor regions, improving diagnostic accuracy, by combining U-Net++ and SUNet for accurate tumor segmentation and coupling them with a CNN-based classifier. Interestingly, our method outperformed the benchmark set in the base reference study, achieving a classification accuracy of 99.07%. The model performed well on the BUSI dataset despite being created and trained on low-end hardware, suggesting that it could be used in clinical settings with limited resources. This approach has the potential to be a dependable tool for early and precise breast cancer detection because of its interpretability and compatibility with clinical workflows, opening the door for further advancements in medical imaging.

Future scope

Future research will concentrate on improving the segmentation model by switching to SU-Net and adding dilated convolutions in addition to attention gates in order to better capture the irregular boundaries frequently observed in malignant tumours. To strengthen generalization, the framework will also be validated on multiple datasets beyond BUSI, covering varied demographics and imaging conditions. In addition, benchmarking against advanced architectures such as DenseNet and hybrid CNN-transformer approaches will help position the framework more clearly within the state-of-the-art literature. To expedite the diagnostic process, we also intend to create a joint end-to-end network that carries out segmentation and classification at the same time. For refining lesion edges in predicted masks, postprocessing methods like morphological operations will be investigated. When combined, these enhancements should improve accuracy and reliability and bring the system closer to real-world clinical application.

References

[1] Y. Zhang, S. Wang, P. Phillips, Z. Dong, G. Ji, and J. Yang, "Ultrasound breast image classification through domain knowledge integra on into deep neural networks," *IEEE Trans. Image Process.*, vol. 30, pp. 6745–6757, 2021.

[2] O. Ronneberger, P. Fischer, and T. Brox, “UNet: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI)*, Munich, Germany, 2015, pp. 234–241.

[3] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, et al., “A survey on deep learning in medical image analysis,” *Med. Image Anal.*, vol. 42, pp. 60–88, 2017.

[4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.

[6] Conf. Compute. Vis. Pattern Recognition. (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.

[7] D. Abdelhafiz, C. Yang, R. Ammar, S. Nabavi, and V. Abhishek, “Breast cancer detection using transfer learning and adaptive decision fusion,” *Sensors*, vol. 20, no. 21, p. 6313, 2020.

[8] Q. Huang, F. Zhang, X. Li, and Q. Liu, “Diagnosis of breast tumours with ultrasound images based on deep learning,” *J. Healthc. Eng.*, vol. 2020, Article ID 6380271, 2020.

[9] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. AlDujaili, Y. Duan, O. Al-Shamma, et al., “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” *J. Big Data*, vol. 8, no. 1, pp. 1–74, 2021.

[10] S. L. Jagannadham, K. L. Nadh and M. Sireesha, ”Brain Tumour Detection Using CNN,” *2021 Fifth International Conference on ISMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2021, pp. 734-739.

[11] Greeshma, B., Sireesha, M., Thirumala Rao, S.N. (2022). Detection of Arrhythmia Using Convolutional Neural Networks. In: Shakya, S., Du, K.L., Haoxiang, W. (eds) *Proceedings of Second International Conference on Sustainable Expert Systems*. Lecture Notes in Networks and Systems, vol 351. Springer, Singapore.

[12] S. Moturi, S. Tata, S. Katragadda, V. P. K. Laghumavarapu, B. Lingala and D. V. Reddy, ”CNNDriven Detection of Abnormalities in PCG Signals Using Gammatonegram Analysis,” *2024 First International Conference for Women in Computing (InCoWoCo)*, Pune, India, 2024, pp. 1-7.

[13] D. Venkatareddy, K. V. N. Reddy, Y. Sowmya, Y. Madhavi, S. C. Asmi and S. Moturi, ”Explainable Fetal Ultrasound Classification with CNN and MLP Models,” *2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC)*, Davangere, India, 2024, pp. 1-7.

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS PROUDLY PRESENTED TO

Chandana Priya Badina

FOR PRESENTING A PAPER TITLED

Ultrasound-Based Breast Cancer Detection Using a Segmentation-Guided Deep Learning Framework

IN THE

2025 IEEE INTERNATIONAL CONFERENCE ON INNOVATE FOR HUMANITARIAN TECH SOLUTIONS FOR GLOBAL CHALLENGE (ICIH)

ACROPOLIS INSTITUTE OF TECHNOLOGY AND RESEARCH, INDORE

21ST - 22ND NOVEMBER 2025

Dr. Namrata Tapaswi
Conference Chair

Dr. S.C. Sharma
Director

1% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 12 words)

Match Groups

1 Not Cited or Quoted 1%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 1 Not Cited or Quoted 1%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

1% Internet sources
0% Publications
0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Internet

www.semanticscholar.org <1%