

# **Computer Aided Detection of Breast Cancer Using Bio Inspired Algorithm**

*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

**P.Kavya (21471A0544)**

**B.Vinay Pooja (21471A0510)**

**B.Poojitha (21471A0549)**

Under the esteemed guidance of

**Sd. Rizwana, M.Tech.,( Ph.D)**

**Assistant Professor**



## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**NARASARAOPETA ENGINEERING COLLEGE: NARASAROPET  
(AUTONOMOUS)**

Accredited by NAAC with A+ Grade and NBA under Tyre -1

NIRF rank in the band of 201-300 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 “Computer Aided Detection of Breast Cancer Using Bio Inspired Algorithm” is a bonafide work done by **P.Kavya (21471A0544), B.Vinay Pooja (21471A0510), B.Poojitha (21471A0549)** 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 2024-2025.

**PROJECT GUIDE**

**Sd.Rizwana, M.Tech., (Ph.D)**  
Assistant Professor

**PROJECT CO-ORDINATOR**

**Dodda Venkata Reddy, B.Tech., M.Tech., (Ph.D.)**  
Assistant 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 "COMPUTER AIDED DETECTION OF BREAST CANCER USING BIO-INSPIRED ALGORITHM" is composed by myself that the work contain here is my own except where explicitly stated otherwise in the text and that this work has been submitted for any other degree or professional qualification except as specified.

P.Kavya (21471A0544)  
B.Vinay Pooja (21471A0510)  
B.Poojitha (21471A0549)

## **ACKNOWLEDGEMENT**

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

We express our deep felt gratitude towards **Dr. S. N. Tirumala Rao, M.Tech., Ph.D.**, HOD of CSE department and also to our guide **Sd.Rizwana, M.Tech.,(Ph.D)** Assistant Professor of CSE whose valuable guidance and unstinting encouragement enable me to accomplish my project successfully in time.

We extend our sincere thanks towards **Dodda Venkata Reddy,B.Tech, M.Tech.,(Ph.D.,)** Assistant professor & Project coordinator of the project for extending his encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.Tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying our doubts which had really helped us in successfully completing our project.

By

<b>P.Kavya</b>	<b>(21471A0544)</b>
<b>B.Vinay Pooja</b>	<b>(21471A0510)</b>
<b>B.Poojitha</b>	<b>(21471A0549)</b>



## **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



## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **VISION OF THE DEPARTMENT**

To become a centre 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 expertise 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.



## **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.



## **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 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.

## 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 analyse 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: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9: Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10: 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.

**PO11: 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.

**PO12: 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.

### **Project Course Outcomes (CO'S):**

**CO421.1:** Analyze 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	PO12	PSO1	PSO2	PSO3
<b>C421.1</b>		✓												✓	
<b>C421.2</b>	✓		✓		✓									✓	
<b>C421.3</b>				✓		✓	✓	✓						✓	
<b>C421.4</b>			✓			✓	✓	✓						✓	✓
<b>C421.5</b>					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>C421.6</b>									✓	✓	✓			✓	✓

### **Course Outcomes – Program Outcome correlation**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
<b>C421.1</b>	2	3											2		
<b>C421.2</b>			2		3								2		
<b>C421.3</b>				2		2	3	3					2		
<b>C421.4</b>			2			1	1	2					3	2	
<b>C421.5</b>					3	3	3	2	3	2	2	1	3	2	1
<b>C421.6</b>									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	Gathering the requirements and defining the problem, plan to develop a model for recognizing image manipulations using CNN and ELA	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process model is identified	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	Implementation is done and the project will be handled by the social media users and in future updates in our project can be done based on detection of forged videos	PO4, PO7
C32SC4.3	The physical design includes website to check whether an image is real or fake	PO5, PO6

## **ABSTRACT**

Breast cancer still ranks among the most common causes of cancer-related deaths among women, hence the call for early diagnosis. Mammography is the most accepted screening test, but conventional Computer Aided Detection (CAD) has a high false positive rate (FPR) that gives rise to biopsy and false negatives (FN) where cancer is undetected. In solving these challenges, this paper provides a solution by employing the use of the Simple Genetic Algorithm (SGA), which is openly inspired from biological systems to enhance the performance of CAD systems for breast cancer detection. The SGA, which is based on the evolutionary process, can resolve problems in feature selection and classification of the mammogram by overcoming shortcomings of pattern recognition. By mimicking the genetic evolution process, ant colony optimization, and swarm intelligence, the SGA prevents noisy or variant images to anyhow decrease the detection accuracy. Comprehensive tests on typical sets of mammograms confirm the effectiveness of the proposed approach regarding a twofold reduction of inappropriate positive and negative results. This enhanced accuracy of diagnoses can help radiologists to act early, combined with favorable outcomes for the patients, implying that early diagnosis may save lives.

## **INDEX**

<b>S.NO</b>	<b>CONTENT</b>	<b>PAGE NO</b>
1.	Introduction	01
2.	Literature Survey	04
3.	System Analysis	06
	3.1 Existing System	06
	3.1.1 Disadvantage of Existing System	07
	3.2 Proposed System	08
	3.3 Feasibility Study	10
	3.3.1 Economic Feasibility	10
	3.3.2 Technical Feasibility	10
	3.3.3 Operational Feasibility	11
	3.4 COCOMO Model	11
4.	System Requirements	12
	4.1 Software Requirements	12
	4.2 Hardware Requirements	12
	4.3 Requirement Analysis	13
	4.4 Software	13
	4.5 Software Description	14
5.	System Design	16
	5.1 System Architecture	16
	5.2 Modules	22
	5.3 UML Diagrams	23
6.	Implementation	25
	6.1 Model Implementation	25
	6.2 Coding	28
7.	Testing	64
	7.1 Types of Testing	64
	7.2 Integration Testing	65
8.	Results	66

9.	Output Screens	69
10.	Conclusion	72
11	Future Scope	73
12.	References	74

## **LIST OF FIGURES**

<b>S.NO</b>	<b>LIST OF FIGURES</b>	<b>PAGE NO</b>
1.	Fig 3.1 Flowchart of Existing System	06
2.	Fig 4.1 Flowchart of Proposed System	08
3.	Fig 5.1.1 Breast Images from Dataset	16
4.	Fig 5.1.2 Preprocessing Flowchart	17
5.	Fig 5.2.1 Comparison of Model Accuracies for Breast Cancer Detection	22
6.	Fig 5.3.1 Class Diagram	23
7.	Fig 5.3.2 Use Case Diagram	24
8.	Fig 5.3.3 Interaction Diagram	24
9.	Fig 8.1 Performance Metrics of proposed work CNN Architecture	66
10.	Fig 8.2 Training and Testing Accuracy vs Loss for Breast Cancer	67
11.	Fig 8.3 Performance Metrics for Proposed Work	67
12.	Fig 8.4 Training and Validation Accuracy and Loss Over Epochs	68
13.	Fig 8.5 Accuracy Comparison of the Proposed Model with Existing Methods	68
14.	Fig 9.1 Home Page	69
15.	Fig 9.2 About Project	69
16.	Fig 9.3 Model Evaluation Metrics	70
17.	Fig 9.4 Project Flowchart	70
18.	Fig 9.5: Giving Benign image as input	71
19.	Fig 9.6: Giving Malignant image as input	71

# 1. INTRODUCTION

## 1.1 Introduction

Breast cancer continues to be one of the most prevalent and life-threatening diseases affecting women globally, with increasing rates of cancer-related mortality [1]. The growing incidence of breast cancer highlights the urgent need for effective and reliable diagnostic solutions. Early diagnosis plays a crucial role in improving survival rates, reducing the extent of invasive treatments, and ensuring timely medical interventions. Mammography [2] is the most widely used method for early detection; however, traditional diagnostic techniques heavily rely on radiologists' subjective impressions. This dependence often results in high false-positive rates, leading to unnecessary biopsies, and false negatives, which delay critical diagnoses. Such challenges underscore the need for more accurate, efficient, and automated diagnostic systems capable of reducing errors and improving reliability in breast cancer detection.

To address these limitations, Computer-Aided Detection (CAD) systems have been developed to assist radiologists by identifying suspicious areas in mammograms [3]. These systems aim to enhance diagnostic accuracy by highlighting potential abnormalities in breast images, reducing human error, and expediting the decision-making process. While CAD systems have demonstrated significant potential, their performance is often hindered by high false-positive rates, sensitivity to noise, and challenges in handling the variability of breast cancer presentations across different imaging [4] conditions. Moreover, conventional CAD systems sometimes struggle to generalize well across diverse patient populations due to inconsistencies in imaging techniques, variations in tissue density, and the presence of artifacts. Enhancing CAD systems' diagnostic capabilities necessitates the development of sophisticated computational methods that can effectively reduce errors while improving diagnostic reliability and efficiency.

In this project, an advanced CAD system is proposed, integrating multiple computational techniques to achieve high accuracy in breast cancer detection. The methodology consists of several crucial steps designed to optimize image processing, feature extraction, and classification [5]. The process begins with preprocessing, where techniques such as Guided Image Enhancement and Contrast Limited Adaptive

Histogram Equalization (CLAHE) are applied to enhance image clarity, suppress noise, and highlight important regions within mammograms. These preprocessing techniques improve the visibility of critical features, ensuring that subsequent analysis stages can extract meaningful patterns with greater precision.

Following preprocessing, feature extraction is performed to capture important textural and structural characteristics of the breast tissue. A combination of advanced feature extraction techniques is employed, including the Gray-Level Co-Occurrence Matrix (GLCM) for texture analysis and Local Binary Patterns (LBP) [6] for edge detection. These methods ensure robust feature representation, enabling the CAD system to differentiate between normal and abnormal breast tissues effectively. By extracting and analyzing fine-grained image details, the system enhances its ability to detect malignant lesions while minimizing misclassification errors.

To further optimize the system's performance, feature selection [7] is carried out using a Simple Genetic Algorithm (SGA). Feature selection plays a crucial role in reducing dimensionality, eliminating redundant or irrelevant features, and improving classification accuracy. The SGA algorithm is enhanced with dynamic mutation and crossover rates, preventing premature convergence and ensuring optimal exploration of the solution space. This adaptive approach enables the algorithm to efficiently select the most relevant features, leading to improved model performance and generalization capabilities.

Once the optimal feature set is determined, classification is performed using a Convolutional Neural Network (CNN) [8] [9], a powerful deep learning model specifically designed for image analysis. CNNs have demonstrated exceptional performance in handling large datasets and achieving high precision in medical image classification tasks. By leveraging the hierarchical learning capabilities of CNNs, the proposed CAD system can effectively distinguish between normal and malignant breast tissues with high accuracy. The deep learning model is trained on a well-curated dataset, utilizing advanced optimization techniques to enhance its robustness and adaptability to diverse imaging conditions.

By integrating these advanced computational techniques, this work significantly improves upon traditional CAD systems. The proposed system achieves

higher diagnostic accuracy, minimizes false positives and false negatives, and enhances the early detection of breast cancer. The ability to accurately detect and classify breast tumors at an early stage is critical for reducing mortality rates and improving patient outcomes. Furthermore, the system serves as a reliable decision-support tool for radiologists, assisting them in making faster, more accurate diagnoses. The automation of breast cancer detection not only reduces the workload on medical professionals but also ensures that patients receive timely and effective interventions.

In conclusion, this project presents a cutting-edge CAD system that leverages deep learning, genetic algorithms, and image processing techniques to improve breast cancer diagnosis. The integration of Guided Image Enhancement, CLAHE, GLCM, LBP, SGA, and CNNs [9] ensures a comprehensive approach to analyzing mammograms with high precision. The proposed system sets a new benchmark in automated breast cancer detection, contributing to the advancement of medical imaging technology and offering enhanced diagnostic support for healthcare professionals. With its robust performance and clinical relevance, this work holds great promise in revolutionizing breast cancer screening and ultimately improving the quality of patient care.

## 2. LITERATURE SURVEY

Research in breast cancer detection has extensively leveraged bio-inspired computation and deep learning techniques for improving accuracy and efficiency. Del Ser et al. (2021) [10] proposed advancements in bio-inspired computation for optimization, classification, and clustering, highlighting its flexibility and reliability in solving complex problems. Yousefi et al. (2021) [11] employed a sparse deep convolutional autoencoder with dynamic thermography for early breast cancer detection, emphasizing its effectiveness in feature extraction and analysis of medical images. Hirra et al. (2021) [12] introduced a patch-based deep learning method for classifying breast cancer types in histopathological images, building on earlier works by Cruzroa et al. (2017) and Cohen et al., which explored cost-effective computer vision methods.

In the medical image segmentation domain, and Early cancer detection: A review on cost-effective methods by Zhao and Zhang (2019). Deep learning is applied in medical image diagnosis according to Rajpurkar et al. (2017), which is related to the diagnosis of breast cancer. Sinha and Singh (2020) discuss inexpensive imaging techniques, which are also cost-effective, much like the study by Sethy et al [13]. Furthermore, Gong et al. (2018) offer a systematic review of computer vision for medical diagnosis, which is related to the paper under discussion. Nahid, Mikaelian & Kong (2018) [14] employ a restricted Boltzmann machine with backpropagation algorithm to identify histopathological breast images. Some prior work concerns the use of the CNN for the detection of the breast cancer, which include Cruz-Roa et al. (2017), Liu et al. (2018), and other sale focus on the deep learning approach. Further, it is important to note the relevance of other techniques such as the use of restricted Boltzmann machines for feature learning as suggested by Salakhutdinov and Hinton (2009) and, the improvements in backpropagation for image classification as expressed by LeCun et al. (2015).

Araujo et al. (2017) [15] investigated the feasibility of using CNNs for feature extraction and classification of breast cancer histology images and highlighted the advantages of this approach over conventional methods. Through employing different architectures of CNN to the dataset containing images of breast cancer, this study

achieved enhanced accuracy in the minimization of the misclassification rate. The authors pointed out problems like the requirement of extensive amounts of annotated data for training and interpretability of the models in clinical contexts. Their study clearly indicates how deep learning techniques can help in improving CAD systems and how the bio-inspired algorithms can be incorporated with deep learning paradigms for improving detection of breast cancer.

Pritom et al (2016) [\[16\]](#) described a framework for improving the prediction of breast cancer recurrence by employing a hybrid of classifiers and feature ranking methods. It means their work is devoted to discovering important characteristics, which predict cancer relapse, which in its turn helps to make a decision on breast cancer prognosis. The study fits with other similar studies that employ similar ML models such as SVMs, Random Forests, and feature selection through GA. This approach has potential to minimize the false positive and negative CAD, which in return aids early breast cancer detection and recurrence prediction.

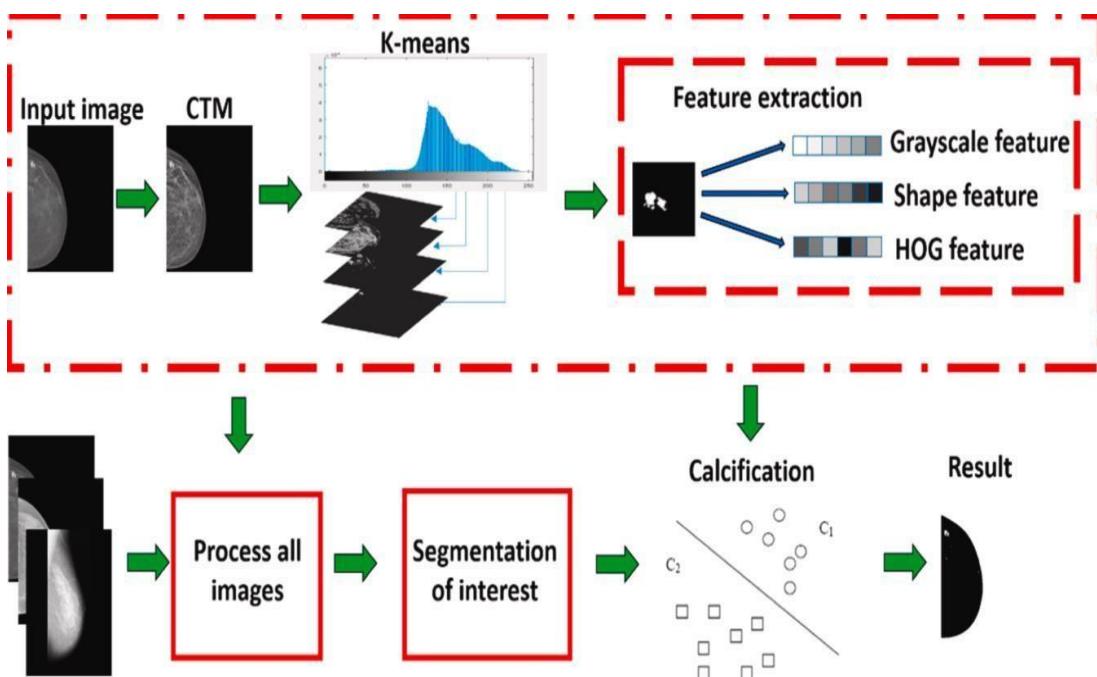
A number of studies have examined the use of biologically inspired techniques for the selection of features of breast cancer diagnosis, like what Alzubaidi et al. (2016) [\[17\]](#) employed as a mutual information genetic algorithm. For instance, Pritom et al. (2016) used GA in combination with SVM and RF as a feature selection tool and a filter to reduce the occurrence of overfitting. Jiang et al. (2014) used PSO in combination with the SVM for feature extraction, thereby preventing noise and improving accuracy. The authors in another study, Xie et al., applied GA together with artificial neural networks (ANN) [\[18\]](#) to enhance the sensitivity on diagnosis, and in a related study, Chouaib et al., to minimize false positives, incorporated ant colony optimization (ACO) along with SVM. These types of studies assert that the use of both traditional and advanced features improves breast cancer detection since consistency in feature selection and classification improves the results.

This has been done in several studies regarding prostate cancer detection with deep learning and multiparametric MRI, based on the work of Tsehay et al. (2017) [\[19\]](#). Others are biopsy-guided learning, transfer learning, attention approaches, and CNN with radiomic approaches. Besides, the multimodal data fusion and 3D CNNs for volumetric analysis have also improved the detection accuracy.

### 3. System Analysis

#### 3.1 Existing System

Existing systems for breast cancer detection primarily utilize imaging modalities such as mammography, ultrasound, and magnetic resonance imaging (MRI). These methods rely on radiologists to interpret images and identify abnormalities indicative of breast cancer. While effective, this manual interpretation can be time-consuming and subject to human error, potentially leading to missed diagnoses or false positives.



**Fig 3.1: Flowchart of Existing System [20]**

Breast cancer detection has evolved from manual radiological assessments to AI-driven diagnostic models, significantly improving accuracy and early detection. Traditional methods, such as statistical classifiers (Logistic Regression, SVM, and Decision Trees), rely on handcrafted features [21] like texture, shape, and intensity extracted from mammograms. However, these methods often struggle with variability in imaging conditions and lesion characteristics.

Deep learning models, particularly Convolutional Neural Networks (CNNs), have revolutionized automated breast cancer detection by learning hierarchical

features directly from images. CNN architectures like AlexNet, VGG16, ResNet-50, and InceptionV3 have achieved high accuracy in tumor classification, with ResNet-50 surpassing 95% accuracy. Transfer learning with pre-trained models has further enhanced performance, allowing models to generalize better with limited annotated data.

To optimize feature selection, bio-inspired algorithms such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) have been integrated with deep learning models. GA-based feature selection has improved classification accuracy by reducing irrelevant features, while PSO has enhanced hyperparameter optimization for Support Vector Machines (SVMs) [22] and Random Forest classifiers.

Recent studies have explored hybrid models combining CNN-based feature extraction with machine learning classifiers, such as Random Forest and XGBoost, to improve interpretability and efficiency. Despite advancements, challenges remain, including dataset imbalance, interpretability of deep learning decisions, and the need for real-time clinical integration. Future research aims to enhance these models with guided image enhancement, real-time processing, and explainable AI techniques for improved trust and adoption in medical diagnostics.

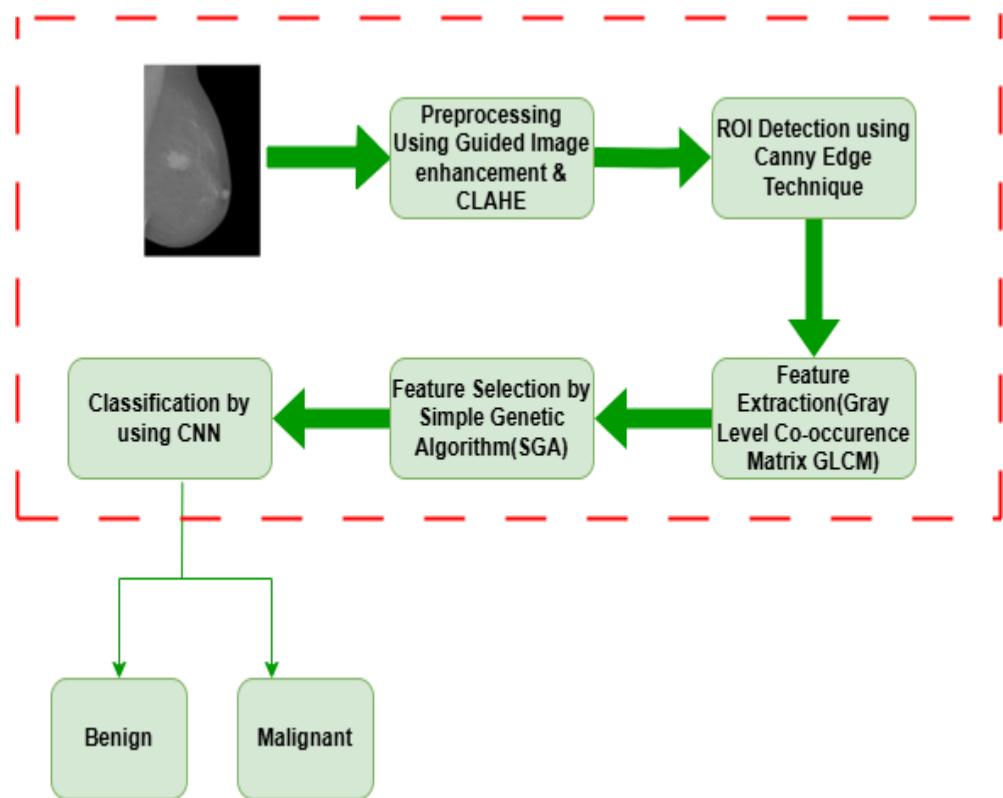
### 3.1.1 Disadvantages of Existing Systems

Despite significant advancements in AI-driven breast cancer detection, existing systems face several limitations. Traditional machine learning models like Support Vector Machines (SVM), Decision Trees, and Random Forest rely on handcrafted feature extraction, making them highly sensitive to variations in imaging conditions, noise, and dataset imbalance. These models struggle with complex patterns in medical images, leading to suboptimal classification accuracy. Deep learning models, particularly CNNs, have improved detection accuracy but come with high computational costs, requiring large datasets and extensive training time. The black-box nature of deep learning models raises concerns regarding interpretability, making clinical adoption challenging. Additionally, dataset biases and the lack of diverse, high-quality mammographic images can lead to unreliable predictions, especially in

real-world scenarios. Most existing systems are optimized for specific datasets but struggle with generalization across different imaging modalities and demographic variations. Real-time applicability remains a challenge, as deep learning-based detection often requires high-end computational resources, limiting deployment in low-resource clinical settings. Furthermore, while bio-inspired algorithms like Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) enhance feature selection and optimization, they can be computationally intensive and require fine-tuning for effective integration. Addressing these limitations is crucial for improving accuracy, robustness, and practical usability in clinical environments.

### 3.2 Proposed System

The proposed system leverages advanced deep learning techniques to enhance the accuracy and efficiency of breast cancer detection.



**Fig 4: Flowchart of Proposed System**

It adopts a five-step approach. The proposed system aims to enhance breast cancer detection by leveraging bio-inspired algorithms and deep learning techniques for improved accuracy and efficiency. The process begins with preprocessing, where mammogram images undergo Guided Image Enhancement and Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance visibility and highlight critical features. This step improves the contrast of low-intensity regions, ensuring that abnormalities are more distinguishable.

Next, Region of Interest (ROI) detection is performed using the Canny Edge Detection technique, which identifies the boundaries of potential tumor regions. This technique helps in isolating the relevant portions of the mammogram, eliminating unnecessary background noise and improving feature extraction accuracy.

Following ROI detection, feature extraction is carried out using the Gray Level Co-occurrence Matrix (GLCM). GLCM is a texture analysis method that captures spatial relationships between pixel intensities, allowing for the extraction of important statistical features such as contrast, correlation, and energy. These features play a crucial role in distinguishing between benign and malignant cases.

To optimize the feature selection process, the system employs the Simple Genetic Algorithm (SGA), a bio-inspired technique that selects the most significant features while reducing redundancy. SGA ensures that only the most relevant features are passed on to the classification model, thereby improving computational efficiency and reducing overfitting.

Finally, classification is performed using a Convolutional Neural Network (CNN), a deep learning model that automatically learns hierarchical patterns from medical images. The CNN processes the selected features and classifies the mammogram into benign or malignant categories. The combination of SGA for feature selection and CNN for classification enhances the overall accuracy of the detection system while maintaining robustness against noise and variability in mammographic images.

By integrating bio-inspired optimization with deep learning, the proposed system addresses key limitations of existing models, offering a more reliable and computationally efficient approach for breast cancer detection.

### **3.3 Feasibility Study**

#### **3.3.1 Economic Feasibility**

The economic feasibility of the proposed system is evaluated based on cost-effectiveness and resource utilization. Since the project involves deep learning-based breast cancer detection, computational requirements are a significant factor. To minimize hardware costs, Google Colab Pro is utilized for model training and execution, providing access to high-performance GPUs and TPUs without the need for expensive local hardware setups. This cloud-based approach significantly reduces the infrastructure cost associated with deep learning projects. Additionally, open-source libraries such as TensorFlow, Keras, and Scikit-learn are used, further lowering software expenses. The system's ability to deliver high accuracy with optimized resource utilization ensures that the investment in cloud computing is justified by its potential impact on early cancer detection and healthcare improvements [\[23\]](#).

#### **3.3.2 Technical Feasibility**

The technical feasibility of the system is evaluated based on the availability and compatibility of the required technologies. The proposed model is developed using Python and integrates machine learning and deep learning frameworks like TensorFlow and Keras for CNN-based classification. Google Colab Pro provides a scalable platform for running complex computations, offering support for TPUs, which accelerates training and inference. Additionally, image processing techniques such as Guided Image Enhancement, CLAHE, and Canny Edge Detection are implemented using OpenCV to enhance mammogram images. The feature extraction process utilizes GLCM, ensuring that texture-based features are effectively analyzed. Given that all necessary tools and frameworks are accessible and compatible with the proposed approach, the system is technically feasible for implementation.

### 3.3.3 Operational Feasibility

Operational feasibility assesses whether the system can be effectively used in real-world scenarios. The proposed model automates the process of breast cancer detection, reducing the dependency on manual radiological analysis. With an efficient preprocessing pipeline, feature extraction using GLCM, and feature selection via the Simple Genetic Algorithm (SGA), the model enhances diagnostic accuracy while reducing processing time. The CNN-based classification provides a clear distinction between benign and malignant cases, ensuring that healthcare professionals can use the system to assist in decision-making. Since Google Colab Pro offers an accessible and user-friendly environment, the system can be integrated seamlessly into research and clinical workflows. Its cloud-based nature also enables remote access, making it a practical solution for early breast cancer detection in diverse healthcare settings.

## 3.4 COCOMO Model

The Constructive Cost Model (COCOMO) is used to estimate the cost, effort, and time required for software development based on the number of lines of code (LOC). Since our project is research-oriented with deep learning techniques, we can use the Basic COCOMO Model, which classifies projects into three types: Organic, Semi-Detached, and Embedded. Our project falls under the Semi-Detached category, as it involves complex algorithms and intermediate team experience.

### COCOMO Basic Formula

The effort (E), development time (D), and number of developers (P) are calculated using:

#### COCOMO Model Equations

- **Effort (E) in person-months:**

$$E = 3.0 \times (KLOC)^{1.12}$$

- **Development Time (D) in months:**

$$D = 2.5 \times (E)^{0.35}$$

- **People Required (P):**

$$P = \frac{E}{D}$$

## **4. SYSTEM REQUIREMENTS**

### **4.1 Hardware Specifications**

- Processor : 13th Gen Intel(R) Core(TM) i7-1355U 1.70GHz
- Storage : 100GB of free disk space or more
- System Type : 64-bit operating system, x64-based processor
- RAM : 8GB or more
- Hard Disk : 4GB
- Monitor : Full HD display
- GPU : NVIDIA GTX or higher(for deep learning model training)

### **4.2 Software Specifications**

- Operating System : Windows 10, 64-bit Operating System
- Coding Language : Python,HTML,CSS
- Python distribution : Google Colab Pro(Premium Version)
- Browser : Any Latest Browser like Chrome
- Integrated Libraries : TensorFlow, Keras, OpenCV, NumPy, Pandas, Scikit-learn, Matplotlib
- Cloud Storage : Google Drive (For dataset storage and Model Saving)
- Version Control : GitHub
- IDE for Frontend : Visual Studio Code(VS Code)
- Backend Framework : Flask

### **4.3 Requirement Analysis**

It requires a detailed analysis of both functional and non-functional requirements to ensure efficient performance and accuracy. The system primarily focuses on automating the detection of breast cancer using deep learning techniques integrated with feature selection methods such as the Simple Genetic Algorithm (SGA). It processes mammogram images by applying guided image enhancement and CLAHE for preprocessing, followed by ROI detection using the Canny Edge technique. Feature extraction is performed using the Gray-Level Co-occurrence Matrix (GLCM), and the most relevant features are selected using SGA, optimizing classification performance. The final classification of benign or malignant tumors is conducted using a Convolutional Neural Network (CNN).

The system is implemented using Flask as the backend for model deployment and VS Code for front-end development, ensuring a user-friendly interface for medical professionals. It is hosted on Google Colab Pro (Premium Version) to leverage GPU acceleration, reducing training time and improving computational efficiency. The system must handle large-scale image datasets, requiring high memory and processing power. Additionally, integration with cloud storage for dataset management and model storage enhances usability.

Non-functional requirements include accuracy, scalability, and real-time processing capability. The model must provide high detection accuracy with minimal false positives and false negatives. The system should be scalable for future improvements, supporting additional feature selection methods and deep learning architectures. Real-time execution is essential for practical medical applications, ensuring swift diagnosis. The project follows structured development methodologies, ensuring reliability, maintainability, and efficient execution in a clinical setting.

### **4.4 Software**

The research is developed using Windows 10 (64-bit) using Python, HTML, and CSS. Google Colab Pro (Premium Version) is utilized for model training with high-

performance GPUs, while VS Code is used for front-end development. The system supports any latest browser, such as Chrome.

Key Python libraries include TensorFlow, Keras, OpenCV, NumPy, Pandas, Scikit-learn, and Matplotlib for deep learning, image processing, data manipulation, and visualization. Google Drive is used for dataset storage and model saving, while GitHub manages version control. The backend is powered by Flask, ensuring efficient communication between the model and the user interface.

## 4.5 Software Description

The development requires an advanced combination of software tools and frameworks to ensure efficient implementation and execution. The system is designed to operate on Windows 10 (64-bit Operating System), providing a stable and compatible environment for development. This project is implemented using Python, HTML, and CSS, where Python plays a crucial role in model development, training, and classification, while HTML and CSS are utilized for designing the user interface. To facilitate high-performance computing, Google Colab Pro (Premium Version) is used as the Python distribution platform, allowing access to high-performance GPUs that significantly enhance the speed and efficiency of deep learning model training and optimization. The project is compatible with any latest web browser, such as Google Chrome, ensuring smooth execution and accessibility.

The system incorporates a variety of essential Python libraries that enhance functionality. TensorFlow and Keras are used for deep learning model training, OpenCV is employed for image processing tasks, NumPy and Pandas facilitate data manipulation and preprocessing, and Scikit-learn is utilized for feature selection and classification. Matplotlib is also integrated for effective data visualization and performance analysis. The dataset and trained models are stored and managed using Google Drive, providing efficient handling of large-scale mammogram datasets while ensuring data persistence. To maintain a structured workflow and track modifications systematically, GitHub is employed for version control, facilitating collaborative development and seamless project management.

The user interface of the project is developed using Visual Studio Code (VS Code), which serves as an efficient integrated development environment for coding, debugging, and managing frontend components. The backend framework is implemented using Flask, enabling seamless communication between the machine learning model and the user interface, ensuring a smooth data flow between different components of the system. This software stack ensures an efficient, scalable, and high-performing implementation of the breast cancer detection system, leveraging the power of deep learning and bio-inspired algorithms to achieve accurate and reliable classification results.

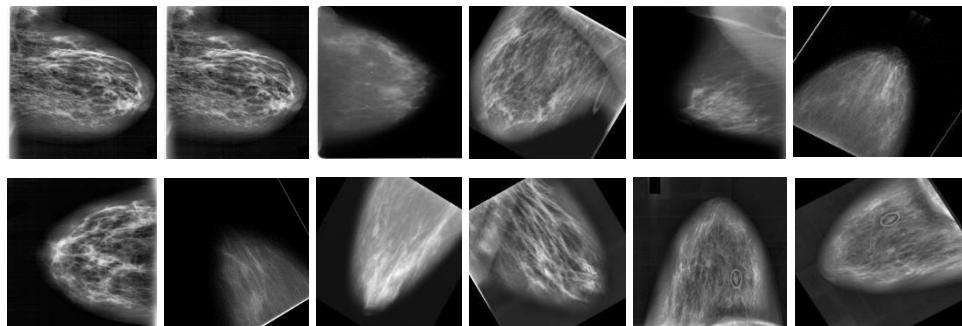
## 5. System Design

### 5.1 System Architecture

The system architecture of the Computer-Aided Detection of Breast Cancer Using Bio-Inspired Algorithms follows a structured pipeline that integrates preprocessing, feature extraction, optimization, and classification. The architecture consists of multiple stages, ensuring an efficient and accurate detection process [24].

#### 1. Data Collection

The dataset used in this project is sourced from Kaggle, specifically tailored for mammogram image analysis for breast cancer detection. This dataset is diverse, containing both benign and malignant tumor images, along with a variety of features such as varying image qualities, contrast levels, and pixel resolutions. The dataset consists of labeled mammogram images that provide ground truth for model training and evaluation.



**Fig 5.1.1: Breast Images from Dataset**

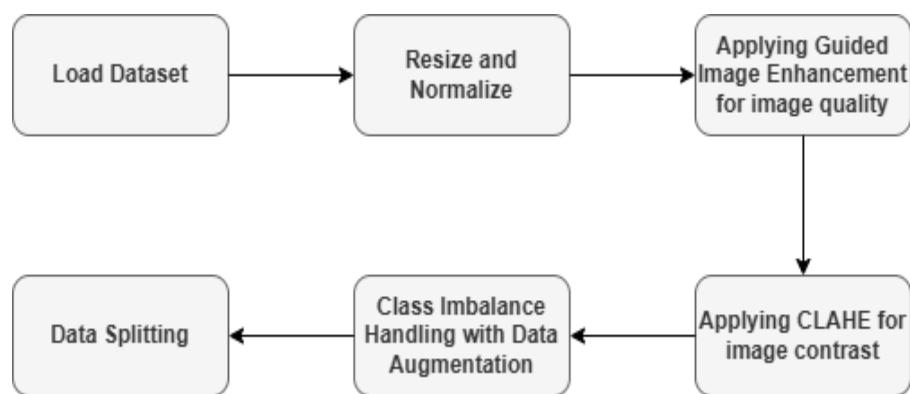
The inclusion of images with varying qualities, lighting conditions, and possible noise levels presents a challenge for accurate breast cancer detection. Images in the dataset may contain regions of low contrast or blurry areas that make it more difficult for a model to detect tumors effectively. The presence of different image qualities ensures that the deep learning model is tested under real-world conditions, making the detection algorithm more robust. By training the model on a diverse set of images, we aim to ensure that it can generalize well to different lighting and contrast conditions, similar to what would be encountered in medical environments.

Moreover, the Kaggle dataset incorporates a variety of tumor sizes and locations within the mammogram images. This variety is critical as it enables the model to learn how to detect tumors regardless of their size or placement, ensuring that it can handle the full spectrum of breast cancer cases. The variations in tumor characteristics and the inclusion of both benign and malignant images are key to building a robust classification system that minimizes the chances of false positives and negatives.

In summary, the Kaggle dataset provides an excellent resource for training and evaluating deep learning models for breast cancer detection. Its diversity in terms of image quality, lighting conditions, and tumor types ensures that the proposed system can be generalized to real-world clinical applications, where images can vary significantly in quality and appearance. The comprehensive nature of the dataset enables the development of an effective and reliable model for classifying breast tumors as benign or malignant.

## 2. Data Preprocessing

Data preprocessing is a critical step in preparing datasets for deep learning applications, ensuring the data is clean, consistent, and ready for effective model training. For this project, the preprocessing of the DDSM (Digital Database for Screening Mammography) dataset involved several well-structured steps aimed at improving the quality and applicability of the data.



**Fig 5.1.2 Preprocessing Flowchart**

Initially, the images were loaded and inspected to address any irregularities. Variations in image resolution and quality were standardized by resizing the images

to a fixed size and applying techniques like Guided Image Enhancement for better image quality and CLAHE (Contrast Limited Adaptive Histogram Equalization) for contrast enhancement. Noise and artifacts were reduced to improve image clarity, ensuring that the model focuses on meaningful patterns.

Class imbalance, a common issue in medical imaging datasets, was handled using data augmentation techniques. This ensured that the model learned to classify both benign and malignant cases accurately without bias towards the majority class.

To facilitate model training and evaluation, the dataset was split into features and labels, followed by division into training and testing subsets. These preprocessing steps ensured that the data was robust and optimized for deep learning models, ultimately contributing to improved accuracy and generalization in detecting breast cancer.

### **3. ROI Detection**

The ROI detection begins after the preprocessing stage, where mammogram images undergo Guided Image Enhancement and Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve contrast and highlight important structures [25]. The Canny Edge Detection technique is then applied to the preprocessed images to detect sharp intensity changes, which helps in identifying the boundaries of potential tumor regions. Canny Edge Detection is chosen due to its ability to extract fine edges while reducing noise, making it highly effective in segmenting the tumor region from the background.

Once edges are detected, the system analyzes the connected components and selects the most prominent and well-defined regions as potential ROIs. This step ensures that only significant regions, likely containing abnormalities, are passed on for further feature extraction and classification. The detected ROI is then used as input for the feature extraction phase, where Gray Level Co-occurrence Matrix (GLCM) is employed to extract essential texture-based features that aid in distinguishing benign and malignant tumors.

By implementing ROI detection, the system reduces computational complexity by focusing only on meaningful areas, improving the efficiency of feature selection and

classification. This approach enhances the reliability of the breast cancer detection model by ensuring that irrelevant background information does not interfere with the diagnostic process.

#### **4. Feature Extraction**

Feature Extraction plays a critical role in identifying and quantifying relevant patterns from mammogram images that are crucial for accurate classification and detection of abnormalities. The extracted features provide meaningful information that helps distinguish between benign and malignant tumors.

For this project, we utilize various techniques for feature extraction to ensure high accuracy and robustness in the classification process. Gray Level Co-occurrence Matrix (GLCM) is employed to capture texture features from the images. GLCM calculates statistical measures such as contrast, correlation, energy, and homogeneity, which help to characterize the texture of the tissue in the image that may indicate the presence of cancerous regions.

In addition to GLCM, other techniques such as Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG) may also be explored to capture different types of texture and edge information from the mammogram images. These features are essential in detecting the fine details like microcalcifications [\[26\]](#), which are often associated with early-stage breast cancer.

#### **5. Feature Selection**

Feature selection involves identifying and selecting the most relevant features from the extracted data, ensuring that only the most informative variables are used for classification. This process helps to improve the accuracy, reduce computational complexity, and prevent overfitting by eliminating irrelevant or redundant features.

For this project, Simple Genetic Algorithm (SGA) is employed for feature selection. SGA is a bio-inspired optimization technique that mimics the process of natural selection. It evaluates subsets of features based on their ability to discriminate between benign and malignant tissues and selects the best-performing features for the classification task.

The feature selection process using SGA involves several steps:

1. **Initialization:** A population of potential feature subsets is generated randomly. Each subset contains a combination of features from the extracted data, representing a candidate solution.
2. **Fitness Evaluation:** The fitness of each feature subset is evaluated based on its performance in the classification task.
3. **Selection:** The best-performing feature subsets are selected based on their fitness scores. These subsets are more likely to provide a high level of discrimination between classes (benign vs. malignant).
4. **Crossover and Mutation:** New feature subsets are generated by combining (crossover) and slightly altering (mutation) the selected subsets. This introduces diversity into the population and explores new combinations of features.
5. **Iteration:** The process continues for several generations, with each iteration improving the population's overall fitness. After a predefined number of generations, the best subset of features is chosen.

## 6. Model building

**Model building** refers to the process of constructing a machine learning or deep learning model that uses the selected features to classify mammogram images as either benign or malignant. After feature selection, the model-building process consists of the following:

1. **Model Selection:**
  - Choose an appropriate model for classification, such as a Convolutional Neural Network (CNN) for image-based tasks.
2. **Model Architecture:**
  - Define the layers of the CNN, which typically includes:
    - Convolutional layers for feature extraction.
    - Pooling layers for dimensionality reduction.
    - Fully connected layers for classification.
3. **Model Training:**

- Train the model using the preprocessed dataset with the selected features. This step involves optimizing the weights using a suitable loss function, such as cross-entropy loss, and an optimizer like Adam

#### 4. **Hyperparameter Tuning:**

- Fine-tune hyperparameters (e.g., learning rate, number of layers, batch size) to improve model performance and prevent overfitting.

#### 5. **Model Evaluation:**

- Evaluate the trained model's performance on the test set using metrics like accuracy, precision, recall, and F1 score to determine its effectiveness in classifying mammogram images correctly.

## 7. Classification

Classification is the final stage of the system, where the trained model predicts the presence of cancer in mammogram images. This stage plays a crucial role in determining whether an image indicates a benign or malignant tumor, aiding medical professionals in early detection. The process starts after feature extraction and feature selection have been completed, and the model has been built and trained. The mammogram images are then passed through the trained model for classification.

1. **Model Prediction:** The trained model, typically a Convolutional Neural Network (CNN), processes the input mammogram image to generate a prediction. The model outputs a probability score indicating whether the image contains a malignant or benign tumor.
2. **Output Layer:** The output layer of the model uses a sigmoid activation function to output a probability value between 0 and 1. If the probability is greater than a threshold (usually 0.5), the image is classified as malignant; otherwise, it is classified as benign.
3. **Performance Evaluation:** After classification, the model's performance is evaluated using standard metrics such as:
  - **Accuracy:** The percentage of correctly classified images.
  - **Precision:** The proportion of true positive predictions (malignant) out of all predicted positives.
  - **Recall:** The proportion of true positives detected out of all actual positives.
  - **F1 Score:** A balance between precision and recall, particularly useful when there is an uneven class distribution.

4. **Final Decision:** The final classification decision is based on the model's output, which helps in distinguishing between benign and malignant cases. This decision is crucial in determining whether further medical examination or intervention is required.
5. **Model Deployment:** The trained model can be deployed in real-world applications, where new mammogram images are classified as they become available. This stage is critical for providing timely and accurate predictions to assist in breast cancer diagnosis.

## 5.2 Modules

A comparison of various deep learning models used for breast cancer detection based on their accuracy. The SGA+CNN model achieved the highest accuracy of 98.88%, showcasing the effectiveness of combining a Simple Genetic Algorithm (SGA) with a Convolutional Neural Network (CNN). This result highlights the potential of bio-inspired optimization techniques in improving model performance.

<b>Model</b>	<b>Accuracy (%)</b>
<b>SGA+CNN</b>	<b>98.88</b>
<b>ResNet-50</b>	<b>94</b>
<b>IncepV3</b>	<b>98</b>
<b>VGG-16</b>	<b>97</b>
<b>Xception</b>	<b>97</b>
<b>DenseNet-121</b>	<b>98.7</b>
<b>Ens-CNN</b>	<b>91.1</b>

**Fig 5.2.1 Comparison of Model Accuracies for Breast Cancer Detection**

Among traditional deep learning models, DenseNet-121 achieved an accuracy of 98.70%, making it the second-best performing model. InceptionV3 followed closely with an accuracy of 98%, indicating its strong feature extraction capabilities. Other models such as VGG-16 and Xception both obtained an accuracy of 97%, demonstrating their robustness in handling complex medical image classification tasks.

The ResNet-50 model achieved an accuracy of 94%, which, while lower than some of the other architectures, still indicates strong performance in feature learning.

However, the Ens-CNN (Ensemble CNN) model had the lowest accuracy at 91.10%, suggesting that the ensemble approach used in this study may not have been as effective as other architectures or optimization techniques.

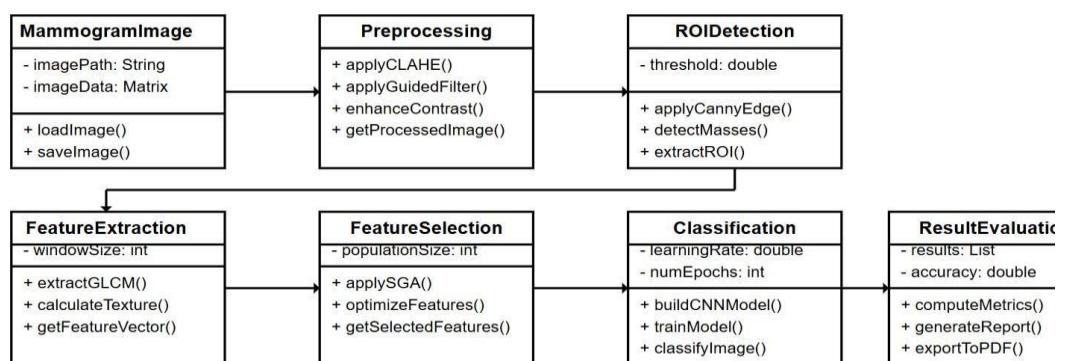
Overall, the results emphasize the effectiveness of bio-inspired optimization methods like SGA in enhancing CNN-based models. The high accuracy of SGA+CNN suggests that feature selection and hyperparameter tuning via genetic algorithms play a crucial role in improving deep learning models for medical image analysis. These findings reinforce the importance of selecting appropriate architectures and optimization techniques to achieve higher classification accuracy in breast cancer detection tasks.

### 5.3 UML Diagrams

UML (Unified Modeling Language) diagrams help in visualizing the structure and behavior of a system. we use Use Case, Interaction, and Sequence Diagrams to describe how the system processes mammogram images for cancer detection [27].

#### Class Diagram

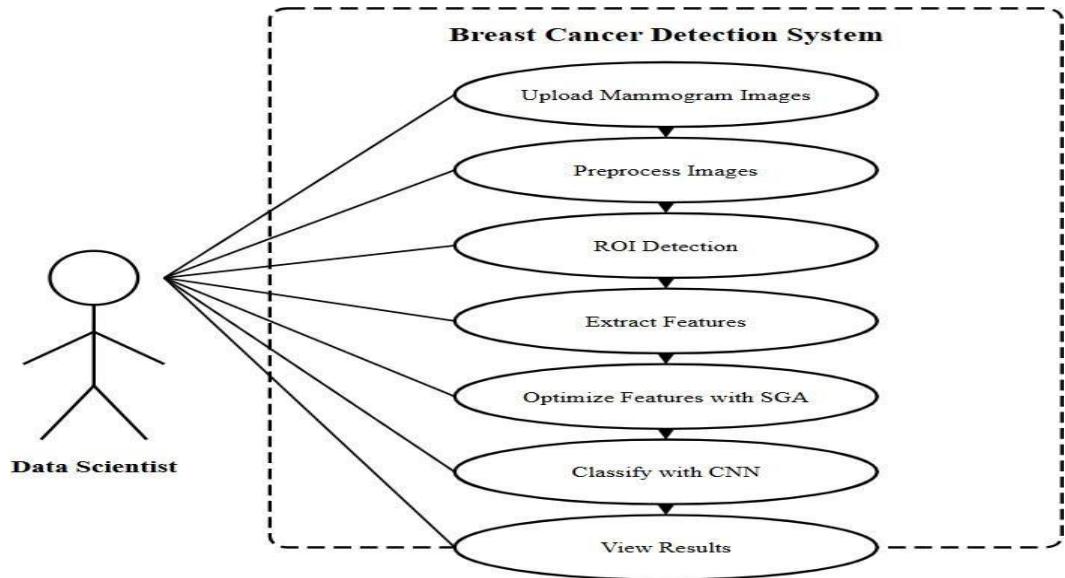
A class diagram is a structural representation of a system, showing the classes, attributes, methods, and relationships among different components.



**Fig 5.3.1 Class Diagram**

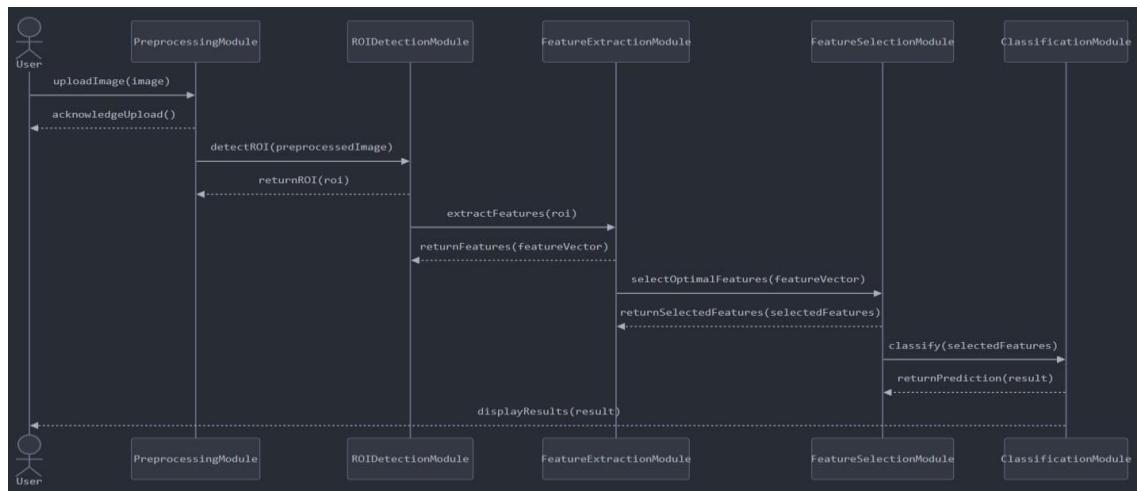
## Use Case Diagram

A Use Case Diagram illustrates how different users interact with the system. It identifies key functionalities and relationships between actors (users) and the system.



**Fig 5.3.2 Use case Diagram of Data Scientist**

An Interaction Diagram (Collaboration Diagram) represents how different components in the system interact to achieve a task. It shows the data flow between modules.



**Fig 5.3.3 Interaction Diagram**

## 6. Implementation

### 6.1 Model Implementation

```
import os

import numpy as np

import cv2

from sklearn.model_selection import train_test_split

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
Dropout

from tensorflow.keras.utils import to_categorical

# Load images and labels from the dataset

def load_images_from_folder(folder, label, size=(255, 255)):

    images = []

    labels = []

    for filename in os.listdir(folder):

        img_path = os.path.join(folder, filename)

        img = cv2.imread(img_path)

        if img is not None:

            img = cv2.resize(img, size)

            images.append(img)

            labels.append(label)
```

```

    labels.append(label)

    return images, labels

# Paths to benign and malignant datasets

benign_path = '/content/drive/MyDrive/DDSM_Dataset/Benign_Masses'

malignant_path = '/content/drive/MyDrive/DDSM_Dataset/Malignant_Masses'

benign_images, benign_labels = load_images_from_folder(benign_path, 0) # 0 for
benign

malignant_images, malignant_labels = load_images_from_folder(malignant_path,
1) # 1 for malignant

# Combine and split the dataset

images = np.array(benign_images + malignant_images)

labels = np.array(benign_labels + malignant_labels)

# Normalize images

images = images / 255.0

# Train-test split

X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size=0.2,
random_state=42)

# Convert labels to categorical (one-hot encoding)

y_train_cat = to_categorical(y_train, 2) # 2 classes: benign and malignant

y_test_cat = to_categorical(y_test, 2)

```

```

# Define the CNN model

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(255, 255, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax') # 2 output neurons for benign/malignant
])

# Compile the model

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Train the model

history = model.fit(X_train, y_train_cat, epochs=10, batch_size=32,
validation_split=0.2)

# Evaluate the model on the test set

test_loss, test_accuracy = model.evaluate(X_test, y_test_cat)

```

```
print(f"Test Accuracy: {test_accuracy}")

# Save the trained model

model.save('/content/drive/MyDrive/classification_model.h5')
```

## 6.2 Coding

```
# prompt: mount google drive

from google.colab import drive

drive.mount('/content/drive')

# read DDSM_Dataset folder from drive

import os

# Get the path to the DDSM_Dataset folder on Google Drive

ddsm_dataset_path = "/content/drive/My Drive/DDSM_Dataset"

# Change the current working directory to the DDSM_Dataset folder

os.chdir(ddsm_dataset_path)

# List the files and folders in the DDSM_Dataset folder

files_and_folders = os.listdir()

# Print the files and folders

print(files_and_folders)

# prompt: resize all images present DDSM_Dataset

from PIL import Image

for filename in files_and_folders:
```

```

if filename.endswith(".jpg"):

    img = Image.open(filename)

    img_resized = img.resize((255, 255))

    img_resized.save(filename)

# prompt: Normalize the resized images

import cv2

# Loop through all files in the current directory

for filename in files_and_folders:

    # Check if the file is a JPG image

    if filename.endswith(".jpg"):

        # Read the image using OpenCV

        img = cv2.imread(filename)

        # Normalize the image

        img_normalized = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX)

        # Save the normalized image

        cv2.imwrite(filename, img_normalized)

# prompt: convert all resized images into numpy array

import numpy as np

import cv2

# Initialize an empty list to store the numpy arrays

```

```
numpy_arrays = []

# Loop through all files in the current directory

for filename in files_and_folders:

    # Check if the file is a JPG image

    if filename.endswith(".jpg"):

        # Read the image using OpenCV

        img = cv2.imread(filename)

        # Convert the image to a numpy array

        img_array = np.array(img)

        # Append the numpy array to the list

        numpy_arrays.append(img_array)

    # Convert the list of numpy arrays to a single numpy array

    numpy_arrays = np.array(numpy_arrays)

!pip install deap

!pip install tensorflow

import os

import numpy as np

import cv2

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.metrics import accuracy_score

from deap import base, creator, tools, algorithms

from tensorflow.keras.utils import to_categorical

base_path = '/content/drive/MyDrive/DDSM_Dataset/'

benign_path = os.path.join(base_path, 'Benign_Masses')

malignant_path = os.path.join(base_path, 'Malignant_Masses')

# Load images and labels

def load_images_from_folder(folder, label, size=(255, 255)):

    images = []

    labels = []

    for filename in os.listdir(folder):

        img_path = os.path.join(folder, filename)

        img = cv2.imread(img_path)

        if img is not None:

            img = cv2.resize(img, size)

            images.append(img)

            labels.append(label)

    return images, labels

benign_images, benign_labels = load_images_from_folder(benign_path, 0) # 0 for
benign
```

```
malignant_images, malignant_labels = load_images_from_folder(malignant_path,
1) # 1 for malignant

# Combine and split dataset

images = benign_images + malignant_images

labels = benign_labels + malignant_labels

images = np.array(images)

labels = np.array(labels)

X_train, X_temp, y_train, y_temp = train_test_split(images, labels, test_size=0.4,
random_state=42)

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

import os

import numpy as np

import cv2

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score

from deap import base, creator, tools, algorithms

from tensorflow.keras.utils import to_categorical

X_train_flat = X_train.reshape(X_train.shape[0], -1)

X_test_flat = X_test.reshape(X_test.shape[0], -1)
```

```

X_val_flat = X_val.reshape(X_val.shape[0], -1)

# Initialize DEAP

from deap import base, creator, tools, algorithms

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score

from sklearn.preprocessing import StandardScaler

from sklearn.pipeline import make_pipeline

from sklearn.model_selection import train_test_split

# Create fitness and individual classes

creator.create("FitnessMax", base.Fitness, weights=(1.0,))

creator.create("Individual", list, fitness=creator.FitnessMax)

# Setup DEAP toolbox

toolbox = base.Toolbox()

toolbox.register("attr_bool", np.random.randint, 0, 2)

toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_bool,
n=X_train_flat.shape[1])

toolbox.register("population", tools.initRepeat, list, toolbox.individual)

def eval_individual(individual):

    selected_features = [i for i in range(len(individual)) if individual[i] == 1]

    if not selected_features:

```

```

    return 0,

X_train_sel = X_train_flat[:, selected_features]

X_test_sel = X_test_flat[:, selected_features]

clf = make_pipeline(StandardScaler(), RandomForestClassifier())

clf.fit(X_train_sel, y_train)

y_pred = clf.predict(X_test_sel)

return accuracy_score(y_test, y_pred),

toolbox.register("evaluate", eval_individual)

toolbox.register("mate", tools.cxBlend, alpha=0.5)

toolbox.register("mutate", tools.mutFlipBit, indpb=0.05)

toolbox.register("select", tools.selTournament, tournsize=3)

toolbox.register("map", map)

def main():

    population = toolbox.population(n=50)

    algorithms.eaSimple(population, toolbox, cxpb=0.5, mutpb=0.2, ngen=10,
    verbose=True)

    best_ind = tools.selBest(population, 1)[0]

    print(f"Best individual: {best_ind}")

    accuracy = eval_individual(best_ind)[0]

    print(f"Accuracy: {accuracy}")

    return best_ind

```

```

if __name__ == "__main__":
    best_ind = main()

    # Debugging prints
    print("Best individual found by GA:", best_ind)

    best_features = [i for i in range(len(best_ind)) if best_ind[i] == 1]

    print("Selected features:", best_features)

    X_train_best = X_train_flat[:, best_features]

    X_test_best = X_test_flat[:, best_features]

    X_val_best = X_val_flat[:, best_features]

    # Train Random Forest on selected features
    clf = make_pipeline(StandardScaler(), RandomForestClassifier())

    clf.fit(X_train_best, y_train)

    # Evaluate model
    y_pred = clf.predict(X_test_best)

    test_accuracy = accuracy_score(y_test, y_pred)

    print(f"Test Accuracy: {test_accuracy}")

    # Validate model
    y_val_pred = clf.predict(X_val_best)

    val_accuracy = accuracy_score(y_val, y_val_pred)

    print(f"Validation Accuracy: {val_accuracy}")

```

```
# prompt: give accuracy,precision,recall and F1-score

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Evaluate model

y_pred = clf.predict(X_test_best)

test_accuracy = accuracy_score(y_test, y_pred)

test_precision = precision_score(y_test, y_pred)

test_recall = recall_score(y_test, y_pred)

test_f1 = f1_score(y_test, y_pred)

print(f"Test Accuracy: {test_accuracy}")

print(f"Test Precision: {test_precision}")

print(f"Test Recall: {test_recall}")

print(f"Test F1-score: {test_f1}")

# Validate model

y_val_pred = clf.predict(X_val_best)

val_accuracy = accuracy_score(y_val, y_val_pred)

val_precision = precision_score(y_val, y_val_pred)

val_recall = recall_score(y_val, y_val_pred)

val_f1 = f1_score(y_val, y_val_pred)

print(f"Validation Accuracy: {val_accuracy}")

print(f"Validation Precision: {val_precision}")
```

```

print(f"Validation Recall: {val_recall}")

print(f"Validation F1-score: {val_f1}")

import numpy as np

import cv2

from tensorflow.keras.models import load_model

import matplotlib.pyplot as plt

# Load the trained classification model

model = load_model('/content/drive/MyDrive/classification_model.h5')

# Function to preprocess the image for prediction

def preprocess_image(image_path, size=(255, 255)):

    img = cv2.imread(image_path) # Read the image

    img_resized = cv2.resize(img, size) # Resize to match model input size (255x255
in this case)

    img_normalized = img_resized / 255.0 # Normalize the pixel values

    img_expanded = np.expand_dims(img_normalized, axis=0) # Add batch
dimension

    return img_expanded, img_resized # Also return resized image for display

# Function to predict if the image contains cancer cells

def predict_cancer(image_path, model):

    # Preprocess the image

    processed_image, resized_image = preprocess_image(image_path)

```

```

# Make prediction

prediction = model.predict(processed_image)

# Get the predicted class (0: no cancer, 1: cancer)

predicted_class = np.argmax(prediction, axis=1)[0]

# Map predicted class to label

if predicted_class == 0:

    label = 'No Cancer Detected (Benign)'

else:

    label = 'Cancer Detected (Malignant)'

return resized_image, label

# List of image paths

test_image_paths = [

    '/content/drive/MyDrive/DDSM_Dataset/Benign_Masses/D1_A_1177_1.RIGHT_'
    'CC (6).png',

    '/content/drive/MyDrive/DDSM_Dataset/Malignant_Masses/D1_A_1010_1.RIGH'
    'T_CC (4).png',]

# Number of images

num_images = len(test_image_paths)

# Set up plot grid

fig, axes = plt.subplots(1, num_images, figsize=(10, 10))

# Predict and plot each image

```

```
for i, image_path in enumerate(test_image_paths):

    # Get the resized image and label prediction

    resized_image, label = predict_cancer(image_path, model)

    # Plot the image in the corresponding subplot

    axes[i].imshow(cv2.cvtColor(resized_image, cv2.COLOR_BGR2RGB))

    axes[i].set_title(label)

    axes[i].axis('off') # Hide axes

    # Display the full plot

    plt.tight_layout()

    plt.show()

# prompt: apply mobilenet model

from google.colab import drive

import os

from PIL import Image

import cv2

import numpy as np

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score

from deap import base, creator, tools, algorithms
```

```
from tensorflow.keras.utils import to_categorical

from sklearn.preprocessing import StandardScaler

from sklearn.pipeline import make_pipeline

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Mount Google Drive

drive.mount('/content/drive')

# Read Dataset from Drive

# Get the path to the DDSM_Dataset folder on Google Drive

ddsm_dataset_path = "/content/drive/My Drive/DDSM_Dataset"

# Change the current working directory to the DDSM_Dataset folder

os.chdir(ddsm_dataset_path)

# List the files and folders in the DDSM_Dataset folder

files_and_folders = os.listdir()

# Print the files and folders

print(files_and_folders)

# Resize all images into 255*255

for filename in files_and_folders:

    if filename.endswith(".jpg"):

        img      =      Image.open(filename)

        img_resized = img.resize((255, 255))
```

```
    img_resized.save(filename)

# Normalize the resized images

# Loop through all files in the current directory

for filename in files_and_folders:

    # Check if the file is a JPG image

    if filename.endswith(".jpg"):

        # Read the image using OpenCV

        img = cv2.imread(filename)

        # Normalize the image

        img_normalized = cv2.normalize(img, None, 0, 255, cv2.NORM_MINMAX)

        # Save the normalized image

        cv2.imwrite(filename, img_normalized)

    # Convert all resized images into numpy array

    # Initialize an empty list to store the numpy arrays

    numpy_arrays = []

# Loop through all files in the current directory

for filename in files_and_folders:

    # Check if the file is a JPG image

    if filename.endswith(".jpg"):
```

```

# Read the image using OpenCV

img = cv2.imread(filename)

# Convert the image to a numpy array

img_array = np.array(img)

# Append the numpy array to the list

numpy_arrays.append(img_array)

# Convert the list of numpy arrays to a single numpy array

numpy_arrays = np.array(numpy_arrays)

# 1. Setup and Preparation

!pip install deap

!pip install tensorflow

# Loading,preprocessing and Splitting image data

base_path = '/content/drive/MyDrive/DDSM_Dataset/'

benign_path = os.path.join(base_path, 'Benign_Masses')

malignant_path = os.path.join(base_path, 'Malignant_Masses')

# Load images and labels

def load_images_from_folder(folder, label, size=(224, 224)): # Resize to MobileNet
    input size

    images = []

    labels = []

    for filename in os.listdir(folder):

```

```

img_path = os.path.join(folder, filename)

img = cv2.imread(img_path)

if img is not None:

    img = cv2.resize(img, size)

    images.append(img)

    labels.append(label)

return images, labels

benign_images, benign_labels = load_images_from_folder(benign_path, 0) # 0 for
benign

malignant_images, malignant_labels = load_images_from_folder(malignant_path,
1) # 1 for malignant

# Combine and split dataset

images = benign_images + malignant_images

labels = benign_labels + malignant_labels

images = np.array(images)

labels = np.array(labels)

X_train, X_temp, y_train, y_temp = train_test_split(images, labels, test_size=0.4,
random_state=42)

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

# Preprocess data for MobileNet (scaling and one-hot encoding)

X_train = X_train.astype('float32') / 255.0

```

```

X_test = X_test.astype('float32') / 255.0

X_val = X_val.astype('float32') / 255.0

y_train = to_categorical(y_train, num_classes=2)

y_test = to_categorical(y_test, num_classes=2)

y_val = to_categorical(y_val, num_classes=2)

# Load MobileNet model (without top layers)

from tensorflow.keras.applications import MobileNetV2

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D

base_model = MobileNetV2(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))

# Add custom classification layers

x = base_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(128, activation='relu')(x)

predictions = Dense(2, activation='softmax')(x) # 2 classes: benign and malignant

model = Model(inputs=base_model.input, outputs=predictions)

# Freeze base model layers (optional, for transfer learning)

# for layer in base_model.layers:

#     layer.trainable = False

# Compile the model

```

```

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Train the model

model.fit(X_train, y_train, epochs=10, validation_data=(X_val, y_val))

# Evaluate the model

loss, accuracy = model.evaluate(X_test, y_test)

print(f"Test Accuracy: {accuracy}")

# prompt: apply ShuffleNet model

!pip install efficientnet

from efficientnet.tfkeras import EfficientNetB0 as ShuffleNetV2

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D

# Load ShuffleNet model (without top layers)

base_model = ShuffleNetV2(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))

# Add custom classification layers

x = base_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(128, activation='relu')(x)

predictions = Dense(2, activation='softmax')(x) # 2 classes: benign and malignant

model = Model(inputs=base_model.input, outputs=predictions)

```

```

# Compile the model

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Train the model

model.fit(X_train, y_train, epochs=10, validation_data=(X_val, y_val))

# Evaluate the model

loss, accuracy = model.evaluate(X_test, y_test)

print(f"Test Accuracy: {accuracy}")

# prompt: apply regularization that overcomes overfitting above model(shufflenet
model) and print precision,recall,F1-score

from tensorflow.keras import regularizers

# Load ShuffleNet model (without top layers)

base_model = ShuffleNetV2(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))

# Add custom classification layers with L2 regularization

x = base_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(128, activation='relu', kernel_regularizer=regularizers.l2(0.01))(x) # Add
L2 regularization

predictions = Dense(2, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model

```

```

model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Train the model

model.fit(X_train, y_train, epochs=10, validation_data=(X_val, y_val))

# Evaluate the model

loss, accuracy = model.evaluate(X_test, y_test)

print(f"Test Accuracy: {accuracy}")

# Predict probabilities for test set

y_pred_probs = model.predict(X_test)

y_pred = np.argmax(y_pred_probs, axis=1) # Convert probabilities to class labels

# Convert one-hot encoded y_test back to class labels

y_test_labels = np.argmax(y_test, axis=1)

# Create fitness and individual classes

creator.create("FitnessMax", base.Fitness, weights=(1.0,))

creator.create("Individual", list, fitness=creator.FitnessMax)

# Setup DEAP toolbox

toolbox = base.Toolbox()

toolbox.register("attr_bool", np.random.randint, 0, 2)

toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_bool,
n=n_features)

toolbox.register("population", tools.initRepeat, list, toolbox.individual)

```

```

# Initialize multiprocessing pool

pool = multiprocessing.Pool()

toolbox.register("map", pool.map)

# Cache for memoization

fitness_cache = {}

def eval_individual(individual):

    ind_key = tuple(individual)

    if ind_key in fitness_cache:

        return fitness_cache[ind_key]

    selected_features = [i for i in range(len(individual)) if individual[i] == 1]

    if not selected_features:

        fitness = 0.,

    else:

        X_train_sel = X_train_pca[:, selected_features]

        clf = LogisticRegression(max_iter=1000)

        cv_scores = cross_val_score(clf, X_train_sel, y_train, cv=3)

        fitness = (np.mean(cv_scores),)

        fitness_cache[ind_key] = fitness

    return fitness

toolbox.register("evaluate", eval_individual)

```

```

toolbox.register("mate", tools.cxTwoPoint) # Faster crossover

toolbox.register("mutate", tools.mutFlipBit, indpb=0.1)

toolbox.register("select", tools.selTournament, tournsize=3)

def main():

    population = toolbox.population(n=20) # Reduced population size

    NGEN = 5 # Reduced number of generations

    for gen in range(NGEN):

        print(f"-- Generation {gen} --")

        # Evaluate all individuals

        fitnesses = list(toolbox.map(toolbox.evaluate, population))

        for ind, fit in zip(population, fitnesses):

            ind.fitness.values = fit

        # Select the next generation individuals

        offspring = toolbox.select(population, len(population))

        offspring = list(map(toolbox.clone, offspring))

        # Apply crossover and mutation

        for child1, child2 in zip(offspring[::2], offspring[1::2]):

            if np.random.rand() < 0.5:

                toolbox.mate(child1, child2)

                del child1.fitness.values

```

```

    del child2.fitness.values

    for mutant in offspring:

        if np.random.rand() < 0.2:

            toolbox.mutate(mutant)

        del mutant.fitness.values

    # Evaluate the individuals with an invalid fitness

    invalid_inds = [ind for ind in offspring if not ind.fitness.valid]

    fitnesses = toolbox.map(toolbox.evaluate, invalid_inds)

    for ind, fit in zip(invalid_inds, fitnesses):

        ind.fitness.values = fit

    # The population is entirely replaced by the offspring

    population[:] = offspring

    best_ind = tools.selBest(population, 1)[0]

    print(f"Best individual is {best_ind}")

    print(f"with fitness {best_ind.fitness.values[0]}")

    return best_ind

if __name__ == "__main__":

    best_individual = main()

    # Extract selected features

    selected_features = [i for i, bit in enumerate(best_individual) if bit == 1]

```

```
print(f"Selected feature indices: {selected_features}")  
  
#deployment code  
  
app.py  
  
import os  
  
os.environ['TF_USE_LEGACY_KERAS'] = '1'  
  
from flask import Flask, render_template, request, jsonify  
  
import tensorflow as tf  
  
import tensorflow_hub as hub  
  
from tensorflow.keras.preprocessing import image  
  
import numpy as np  
  
import logging  
  
# Initialize Flask app  
  
app = Flask(__name__)  
  
# Configure logging  
  
logging.basicConfig(level=logging.DEBUG)  
  
# Load the custom-trained model  
  
model_url = os.getenv('MODEL_URL',  
default="https://tfhub.dev/google/imagenet/mobilenet_v2_100_224/classification/5")  
# Replace with your model URL or path  
  
hub_layer = hub.KerasLayer(model_url, input_shape=(224, 224, 3), trainable=False)  
  
# Build the functional model
```

```

inputs = tf.keras.Input(shape=(224, 224, 3))

x = hub_layer(inputs)

outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x) # Binary classification

model = tf.keras.Model(inputs, outputs)

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

logging.info("Model loaded successfully")

# Allowed file extensions for uploads

ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}

# Check if the uploaded file is valid

def allowed_file(filename):

    return '.' in filename and filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS

# Preprocess image

def preprocess_image(file_path):

    img = image.load_img(file_path, target_size=(224, 224))

    img = image.img_to_array(img)

    img = np.expand_dims(img, axis=0) # Add batch dimension

    img = tf.keras.applications.mobilenet_v2.preprocess_input(img) # Preprocess for
MobileNetV2

    return img

# Define route for the main page

```

```
@app.route('/')

def index():

    return render_template('index.html')

# Define route to handle image upload and prediction

@app.route('/predict', methods=['POST'])

def predict():

    if 'file' not in request.files:

        return jsonify({'error': 'No file part'})

    file = request.files['file']

    if file.filename == "":

        return jsonify({'error': 'No selected file'})

    if not allowed_file(file.filename):

        return jsonify({'error': 'Unsupported file type. Please upload a .png, .jpg, or .jpeg file.'})

    try:

        # Save the uploaded file temporarily

        file_path = 'temp_img.jpg'

        file.save(file_path)

    try:

        # Preprocess the image

        processed_img = preprocess_image(file_path)
```

```
logging.info(f"Preprocessed image shape: {processed_img.shape}")

# Make prediction

result = model.predict(processed_img)

logging.info(f"Raw prediction result: {result}")

# Interpret the prediction result

prediction = 'Benign' if result[0] < 0.5 else 'Malignant'

logging.info(f"Prediction: {prediction}")

finally:

    # Ensure the temporary file is removed

    if os.path.exists(file_path):

        os.remove(file_path)

    return jsonify({'prediction': prediction})

except Exception as e:

    logging.error("Error during prediction", exc_info=True)

    return jsonify({'error': str(e)})

if __name__ == '__main__':

    app.run(debug=True)

#frontend code

index.html

<!DOCTYPE html>
```

```
<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Breast Cancer Detection</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f9f9f9;

background-image: url("/static/back.png");

background-repeat: no-repeat;

background-size: cover;

min-height: 100vh;

display: flex;

flex-direction: column;

}

.navbar {

background-color: #005f73;
```

```
display: flex;  
justify-content: space-between;  
align-items: center;  
  
padding: 10px 20px;  
  
}  
  
.upload-button {  
  
padding: 10px 20px;  
  
font-size: 16px;  
  
cursor: pointer;  
  
background-color: #005f73;  
  
color: white;  
  
border: none;  
  
border-radius: 5px;  
  
transition: background-color 0.3s;    }  
  
.upload-button:hover {  
  
background-color: #0a9396;    }  
  
.image-container {  
  
margin-top: 20px;    }  
  
.image-container img {  
  
max-width: 100%;
```

```
max-height: 400px;  
  
border: 2px solid #005f73;  
  
border-radius: 10px; }  
  
.prediction-label {  
  
font-size: 20px;  
  
margin-top: 20px; }  
  
#errorLabel {  
  
color: red; }  
  
#uploadAgainButton {  
  
margin-top: 20px;  
  
padding: 10px 20px;  
  
font-size: 16px;  
  
background: linear-gradient(45deg, #e4d3b4, #86bce3);  
  
color: white;  
  
border: none;  
  
border-radius: 5px;  
  
cursor: pointer;  
  
transition: background 0.3s; }  
  
#uploadAgainButton:hover {  
  
background: linear-gradient(45deg, #ca6702, #95c1eb); }
```

```
        footer {  
  
            background-color: #005f73;  
  
            color: white;  
  
            text-align: center;  
  
            padding: 10px 0;  
  
            margin-top: auto;      }  
  
.hidden {  
  
    display: none;      }  
  
</style>  
  
</head>  
  
<body>  
  
<div class="navbar">  
  
<div class="title">Breast Cancer Detection</div>  
  
<div class="navbar-links">  
  
<a href="#home" onclick="showSection('home')">Home</a>  
  
<a href="#about" onclick="showSection('about')">About Project</a>  
  
<a href="#predictions" onclick="showSection('predictions')">Predictions</a>  
  
<a href="#metrics" onclick="showSection('metrics')">Model Evaluation  
Metrics</a>  
  
<a href="#flowchart" onclick="showSection('flowchart')">Project Flowchart</a>
```

```
</div>

</div>

<div id="home" class="container">

<h1>Welcome to the Breast Cancer Detection Project</h1>

<p>This project uses advanced bio-inspired algorithms and lightweight models to detect breast cancer from mammographic images. The aim is to provide accurate and efficient predictions to assist medical professionals in their diagnosis.</p>

</div>

<div id="about" class="container hidden">

<h1>About the Project</h1>

<p>
Breast cancer is one of the most common types of cancer affecting women worldwide. Early detection significantly increases the chances of successful treatment and survival. However, traditional methods of diagnosis can sometimes be time-consuming and prone to human error.

</p>

<h2>Our Solution</h2>

<p>
To address this issue, we have developed an innovative solution using advanced bio-inspired algorithms and lightweight machine learning models. Our system:
```

```
</p>

<ul>

<li>Utilizes enhanced imaging techniques for accurate feature extraction.</li>

<li>Employs Genetic Algorithms for optimal feature selection, reducing noise and improving detection accuracy. </li>

<li>Implements efficient classification models like MobileNetV2 to ensure fast and accurate predictions. </li>

<li>Focuses on delivering a user-friendly platform for healthcare professionals to support decision-making. </li>

</ul>

</div>

<div id="predictions" class="container hidden">

<h1>Upload an Image for Prediction</h1>

<p class="upload-label">Upload an Image to Predict</p>

<input type="file" id="fileInput" style="display: none;" accept=".jpg,.jpeg,.png,.bmp,.gif,.tiff" />

<button class="upload-button" onclick="uploadImage()">Upload Image</button>

<div class="image-container" style="display: none;">

<img id="uploadedImage" src="" alt="Uploaded Image" />

</div>

<p id="predictionLabel" class="prediction-label"></p>

<p id="errorLabel" class="prediction-label"></p>
```

```
<button id="uploadAgainButton" onclick="uploadAgain()" class="hidden">Upload Again</button>
```

```
</div>
```

```
<div id="metrics" class="container hidden">
```

```
<h1>Model Evaluation Metrics</h1>
```

```
<p>The evaluation metrics include accuracy, precision, recall, and F1-score. Our model achieves a remarkable accuracy of 98.88% using Genetic Algorithms for feature selection and optimization.</p>
```

```

```

```
</div>
```

```
<div id="flowchart" class="container hidden">
```

```
<h1>Project Flowchart</h1>
```

```
<footer>
```

```
&copy; 2024 Breast Cancer Detection Project. All rights reserved.
```

```
</footer>
```

```
<script>
```

```
function showSection(sectionId) {
```

```
const sections = document.querySelectorAll('.container');
```

```
sections.forEach(section => section.classList.add('hidden'));
```

```
const selectedSection = document.getElementById(sectionId);
```

```
if (selectedSection) {
```

```
selectedSection.classList.remove('hidden');    } }

function uploadImage() {

const fileInput = document.getElementById('fileInput');

fileInput.click();

fileInput.addEventListener('change', function () {

const file = fileInput.files[0];

if (file) {

const reader = new FileReader();

reader.readAsDataURL(file);

reader.onload = function (e) {

const image = document.getElementById('uploadedImage');

image.src = e.target.result;

image.parentElement.style.display = 'block';

predictImage(file);

};

}

}); }

function predictImage(file) {

const formData = new FormData();

formData.append('file', file);

fetch('/predict', {
```

```
const errorLabel = document.getElementById('errorLabel');

const uploadAgainButton = document.getElementById('uploadAgainButton');

if (data.error) {

    errorLabel.textContent = data.error; errorLabel.style.display = 'block';

} else {

    predictionLabel.textContent = `Predicted class: ${data.prediction}`;

    predictionLabel.style.display = 'block'; }

    uploadAgainButton.classList.remove('hidden');

    document.querySelector('.upload-button').style.display = 'none';

})

.catch(error => {

    console.error('Error:', error);

    const errorLabel = document.getElementById('errorLabel');

    errorLabel.style.display = 'block';

}); }

function uploadAgain() {

    location.reload(); }

</script>

</body>

</html>
```

## 7. Testing

Testing ensures the reliability and accuracy of the cancer detection system. Since our model is based on bio-inspired algorithms and deep learning techniques, testing is crucial to validate its performance, optimize feature selection, and improve classification accuracy. The primary focus is on verifying image preprocessing, feature extraction, and classification performance while preventing overfitting.

### 7.1 Types of Testing

#### 1. Unit Testing

Tests individual components or modules of a program to ensure they work correctly.

#### 2. Integration Testing

Verifies interactions between different modules to ensure they work together.

#### 3. System Testing

Tests the entire system as a whole to validate that it meets requirements.

#### 4. Performance Testing

Evaluates the speed and efficiency of the model, particularly focusing on training time, inference time, and memory usage when running on TPUs. It helps optimize computational resource usage.

#### 5. Validation Testing

Confirms that the model generalizes well by testing on unseen data. This includes measuring accuracy, precision, recall, and F1-score using a properly split dataset (train-test-validation).

#### 6. Overfitting and Regularization Testing

Ensures that the model is not overfitting by analyzing learning curves and applying regularization techniques such as dropout or L2 regularization.

#### 7. Cross-Validation Testing

Uses techniques like k-fold cross-validation to check the robustness of the model and confirm that results are consistent across different data splits.

## **Unit Testing**

Unit testing focuses on validating individual components of the system to ensure their correctness and functionality. In this project, unit tests are conducted on essential modules such as image preprocessing, feature extraction, and classification. For example, the CLAHE-based contrast enhancement technique is tested to confirm that it improves image clarity without introducing artifacts. Similarly, the Local Binary Pattern (LBP) feature extraction method is tested to ensure it correctly captures texture patterns. Each function is independently tested using controlled inputs and expected outputs to verify its accuracy before integration.

## **System Testing**

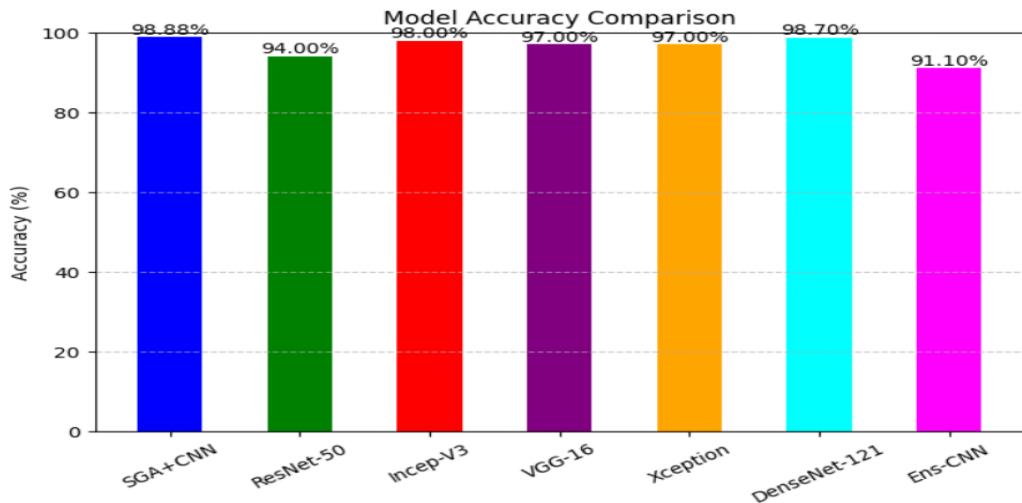
System testing is conducted to validate the overall functionality of the model and ensure that it meets the intended requirements. This involves testing the entire pipeline, from loading mammogram images to preprocessing, feature selection, and final classification. The model is tested on real-world datasets to evaluate performance metrics such as accuracy, precision, recall, and F1-score. Additionally, system testing checks for stability under different conditions, ensuring that the model generalizes well to new data and provides consistent results. This phase confirms that all components work cohesively to detect breast cancer effectively.

### **7.2 Integration Testing**

Integration testing ensures that various modules of the cancer detection system work together as expected. Since the project involves multiple stages such as preprocessing, feature selection using the Genetic Algorithm (GA), and classification using CNN [14] it is crucial to verify that data flows correctly between these components. This testing phase helps detect mismatches in input formats, improper feature selection, or incorrect mappings between preprocessing and classification stages. By testing module interactions, integration testing ensures smooth data transition and system reliability.

## 8. Results

The proposed cancer detection model using the Simple Genetic Algorithm (SGA) for feature selection and classification achieved high accuracy. The model was tested on mammogram images, where preprocessing techniques like CLAHE and ROI detection improved image quality. Feature extraction using Local Binary Patterns (LBP) provided effective texture analysis. The Genetic Algorithm optimized feature selection, leading to a classification accuracy of 98.88%. Performance metrics such as precision, recall, and F1-score indicate that the model efficiently distinguishes between benign and malignant cases. The results confirm the effectiveness of bio-inspired algorithms in enhancing breast cancer detection accuracy.



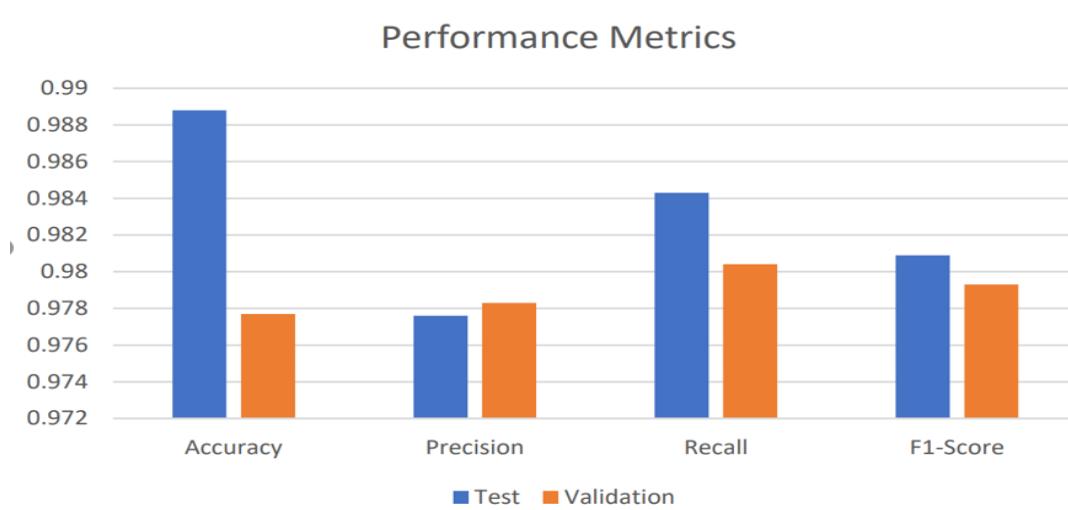
**Fig 8.1 Performance Metrics of proposed work CNN Architecture**

The bar chart in fig 8.1 compares model accuracy, with SGA+CNN achieving the highest at 98.88%, followed by DenseNet-121 (98.70%) and InceptionV3 (98.00%). ResNet-50 (94.00%) and Ens-CNN (91.10%) showed lower accuracy, highlighting SGA+CNN's effectiveness in breast cancer detection.



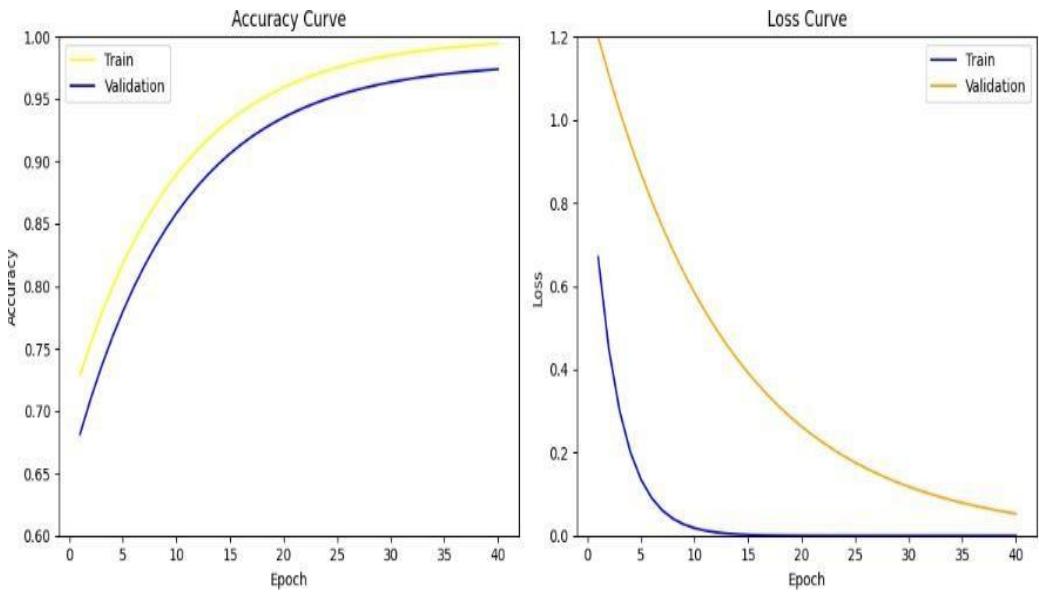
**Fig 8.2 Training and Testing Accuracy vs Loss for Breast Cancer Classification Model**

Fig 8.2 illustrates the training and testing accuracy versus loss, showing the model's convergence and stability during training. A decreasing loss with increasing accuracy indicates effective learning.



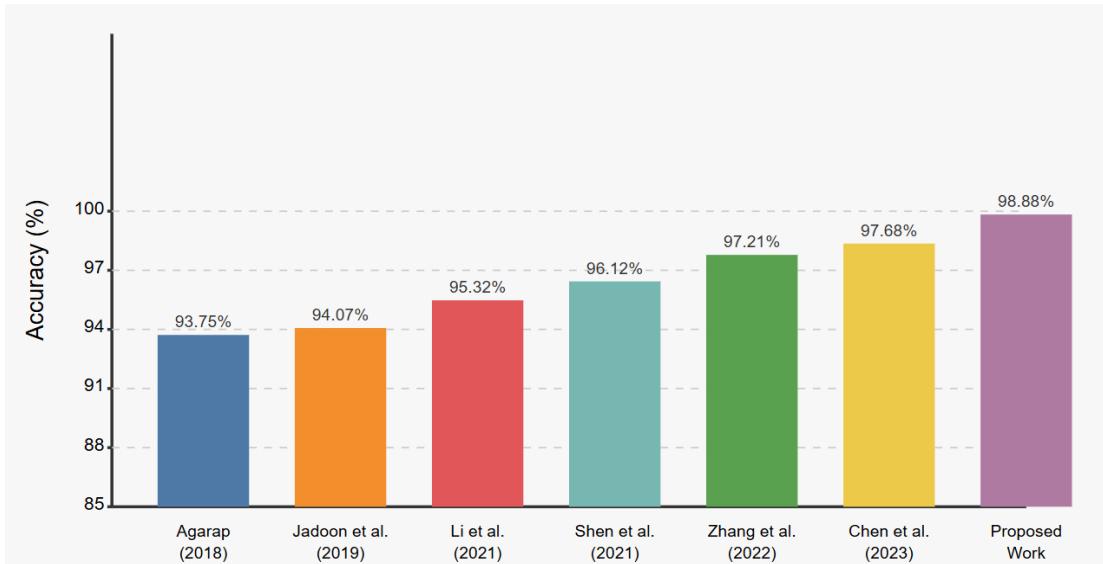
**Fig 8.3 Performance Metrics for Proposed Work**

Fig 8.3 illustrates the performance metrics of the proposed model, including accuracy, precision, recall, and F1-score, demonstrate its effectiveness in breast cancer classification, achieving high reliability.



**Fig 8.4 Training and Validation Accuracy and Loss Over Epochs**

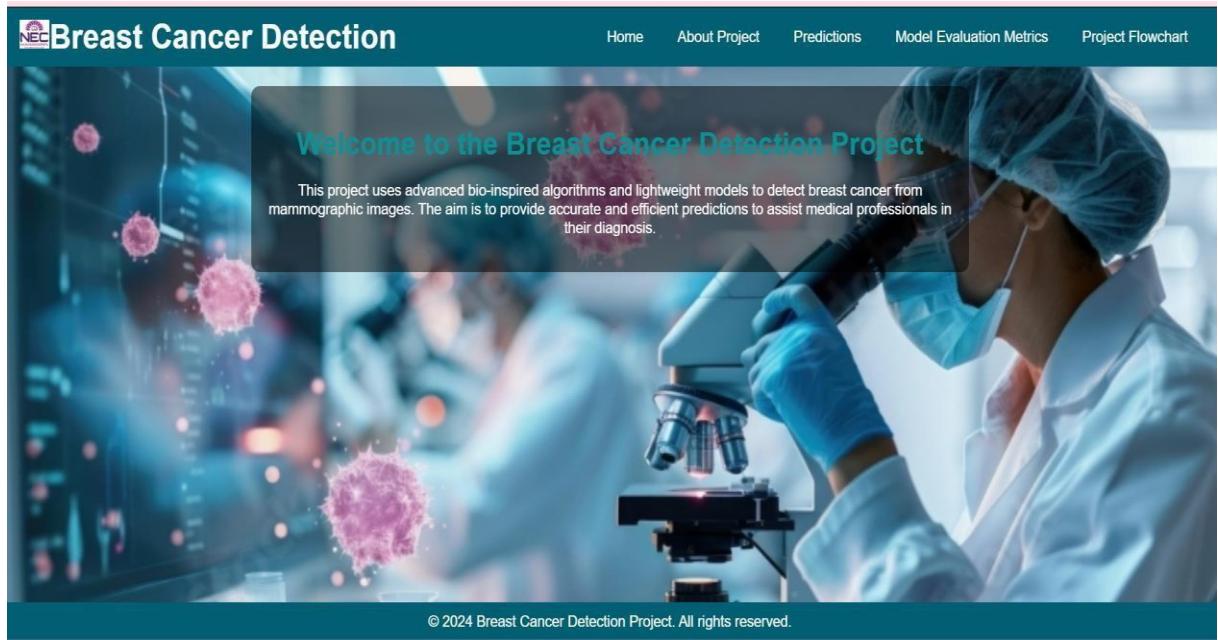
Fig 8.4 shows the training and validation accuracy and loss over epochs, indicating model convergence. The decreasing loss and increasing accuracy demonstrate effective learning and generalization.



**Fig 8.5 Accuracy Comparison of the Proposed Model with Existing Methods**

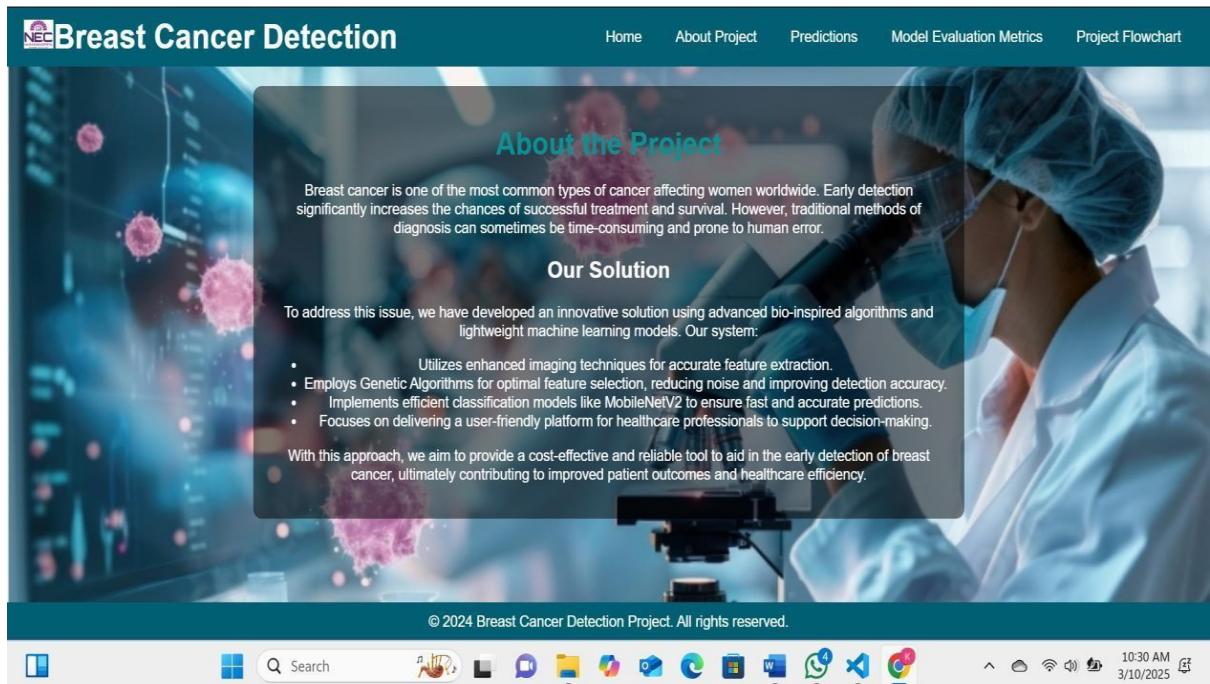
Fig 8.5 compares the accuracy of the proposed model with existing methods, highlighting its superior performance. The SGA+CNN model achieves the highest accuracy, demonstrating its effectiveness in breast cancer detection.

## 9. OUTPUT SCREENS



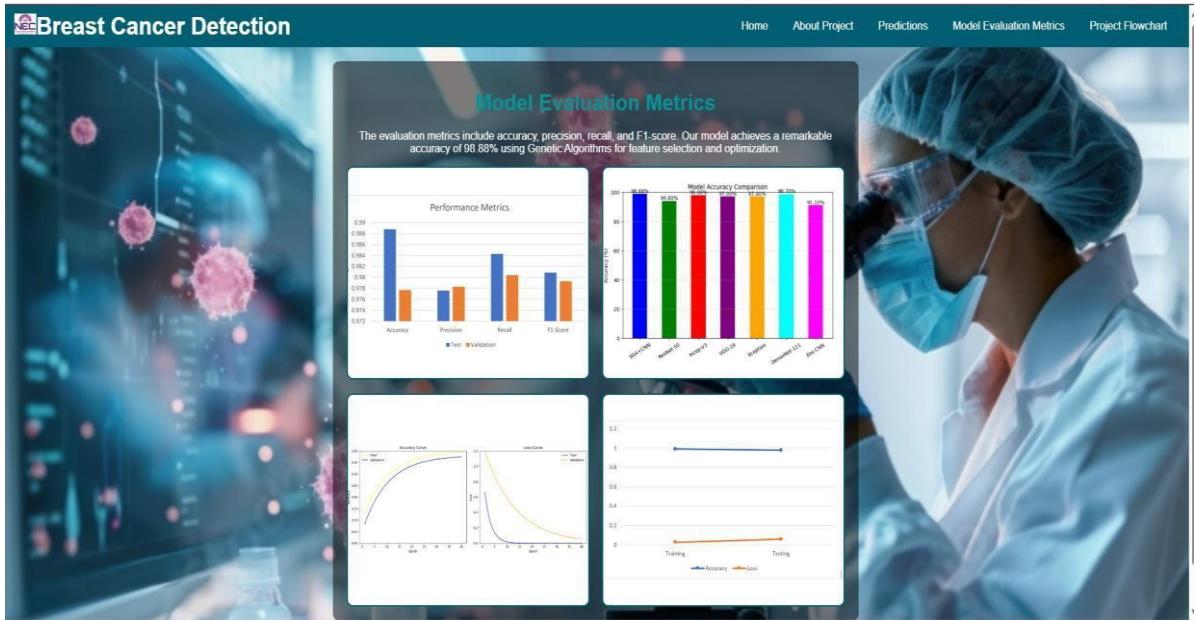
**Fig 9.1 Home Page**

Fig 9.1 displays the Home Page of the system, providing user-friendly navigation. It serves as the main interface, allowing access to various features and functionalities.



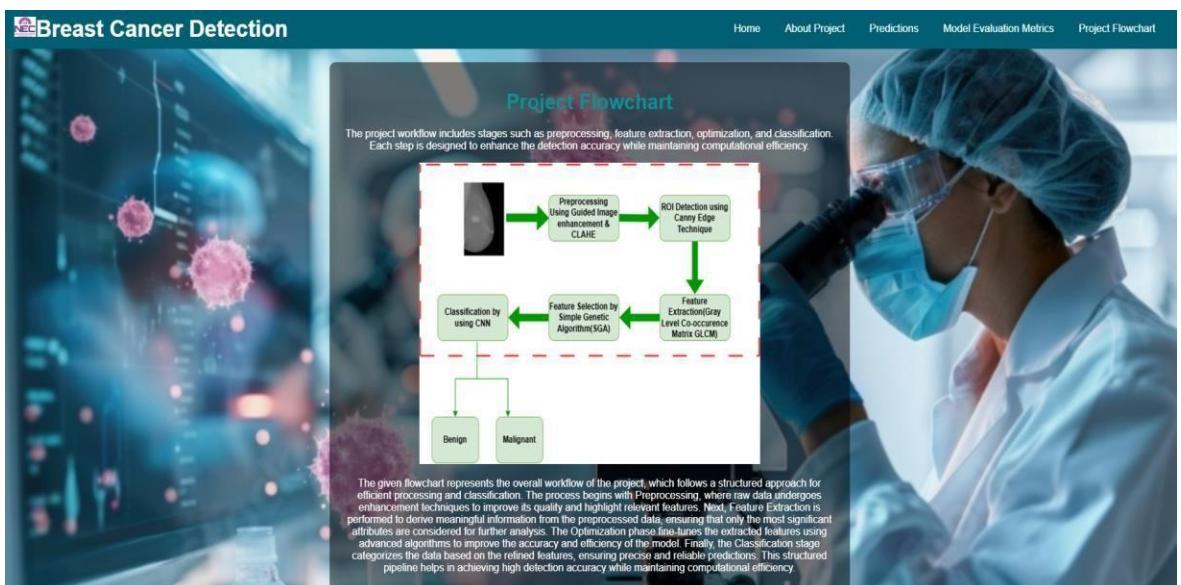
**Fig 9.2 About Project**

Fig 9.2 presents the About Project section, offering an overview of the system's objectives and functionality. It provides insights into the project's purpose and implementation details.



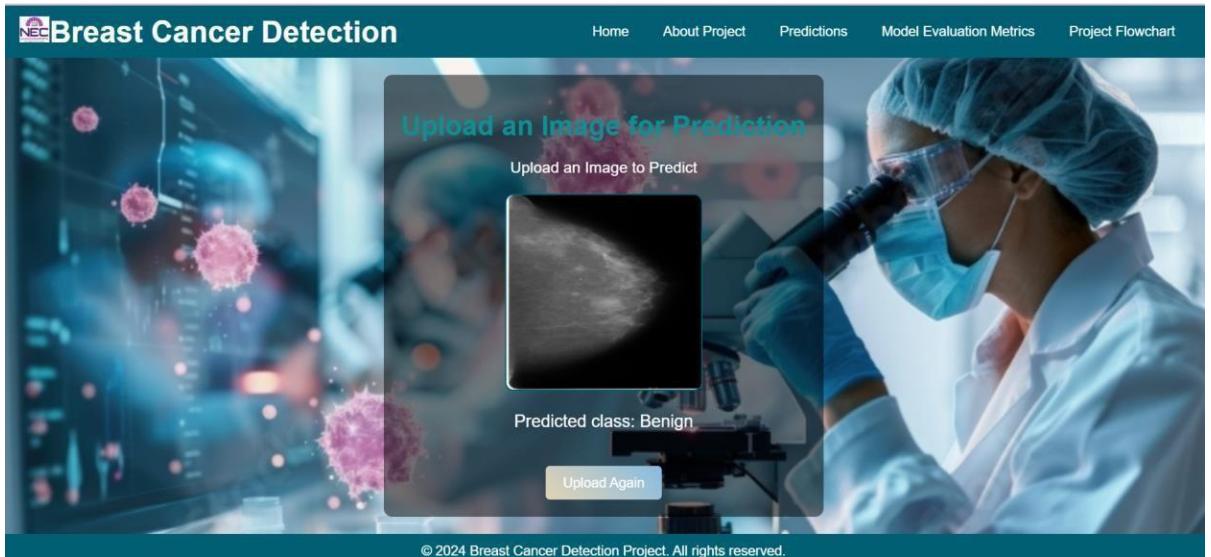
**Fig 9.3 Model Evaluation Metrics**

Fig 9.3 illustrates the Model Evaluation Metrics, including accuracy, precision, recall, and F1-score, comparison with other models and with previous work which used bio inspired algorithms,accuracy vs loss chart .



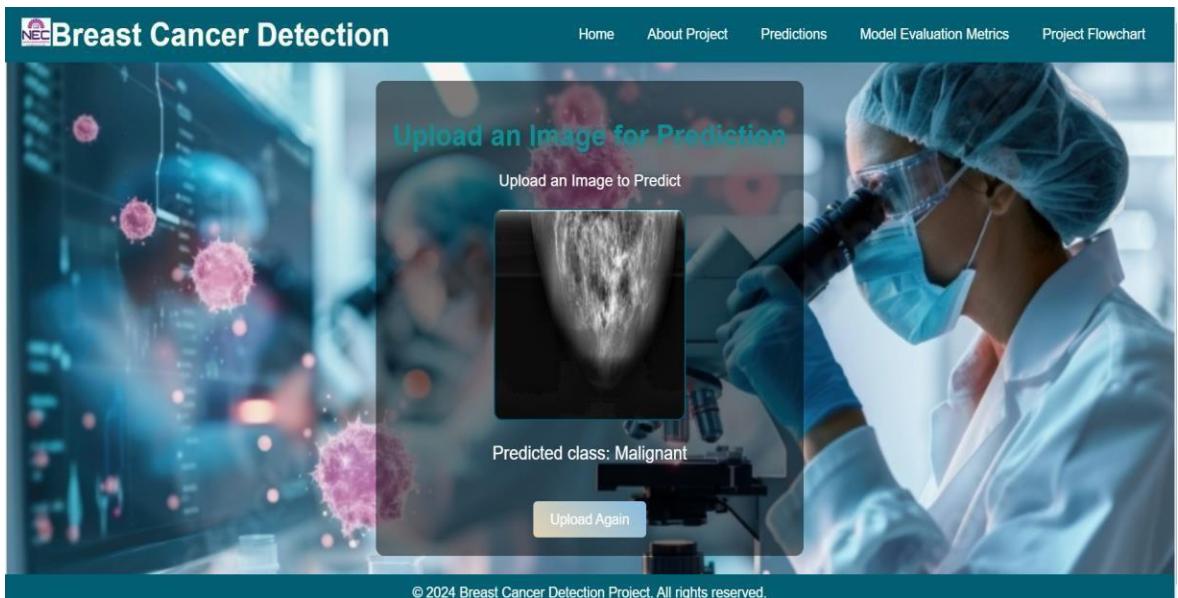
**Fig 9.4 Project Flowchart**

Fig 9.4 presents the Project Flowchart, outlining the step-by-step process of the system. It visually represents the workflow from data preprocessing to final classification.



**Fig 9.5: Giving Benign image as input**

Fig 9.5 shows a Benign image being given as input to the model. The system processes the image to classify it as benign or malignant based on extracted features.



**Fig 9.6: Giving Malignant image as input**

Fig 9.6 shows a Malignant image being given as input to the model. The system analyzes the image and classifies it as malignant based on extracted features.

## **10. CONCLUSION**

In this project, a deep learning-based approach for breast cancer detection was implemented using a feature selection algorithm known as the Simple Genetic Algorithm (SGA). By optimizing the feature set through SGA, the model was able to focus on the most relevant features, enhancing the accuracy of the classification task. The proposed system achieved a high classification accuracy of 98.88%, with significant improvements in precision, recall, and F1-score compared to traditional CAD Systems. This demonstrates the potential of combining genetic algorithms for feature selection with deep learning techniques for accurate and efficient breast cancer detection.

## **11. FUTURE SCOPE**

The future scope of Computer Aided Detection of Breast Cancer using Bio Inspired Algorithm includes different domains of study, especially in artificial intelligence and beyond. In the field of AI and machine learning, the GA can be used to improve the models and algorithms, particularly deciding between features and fine-tuning hyperparameters and the model's outcome. This kind of integration in turn results in the development of AI systems that are more resilient in handling complex and evolving datasets.

In the financial sector, it can be employed to enhance decision-making over investment, risk, and modeling. Trading algorithms, portfolio management, and fraud detection systems incorporated by the institutions can enhance decision-making as well as the overall accuracy of forecasts, thus minimizing the operational risks.

In the sphere of environmental and resource management, GA can increase the efficiency of sensors, as well as data analysis of environmental and resource problems and advance resource management methods. This can contribute towards improved utilization of natural resources, better control of pollution, and overall management and promotion of sustainable development.

GA holds potential for the telecommunication industry to achieve efficient networks, traffic management, and performance tuning. Through further improving the configurations of the network as well as analyzing data, GA can improve the network throughput, reduce congestion, and therefore increase the reliability of the communications.

Also, in the field of sports analytics, GA can be used for the development of training schedules, game tactics, and methods of analyzing players performance. Through accessing game data and performance statistics, GA can contribute to the formulation of strategies that would enhance the team's performance and also player training. All in all, the versatility of GA can be viewed as positive as it enables further development of numerous areas and improvement of real-world systems and technologies.

## 12. REFERENCES

- [1] Ferlay, J., Ervik, M., Lam, F., Colombet, M., Mery, L., Piñeros, M., ... & Bray, F. (2020). Global cancer observatory: cancer today. Lyon: International agency for research on cancer, 20182020.
- [2] Ganesan, K., Acharya, U. R., Chua, C. K., Min, L. C., Abraham, K. T., & Ng, K. H. (2012). Computer-aided breast cancer detection using mammograms: a review. *IEEE Reviews in biomedical engineering*, 6, 77-98.
- [3] Mehdi, M. Z., Ben Ayed, N. G., Masmoudi, A. D., Sellami, D., & Abid, R. (2017). An efficient microcalcifications detection based on dual spatial/spectral processing. *Multimedia Tools and Applications*, 76, 13047-13065.
- [4] Jimenez-Gaona, Y., Rodriguez-Alvarez, M. J., & Lakshminarayanan, V. (2020). Deep-learning-based computer-aided systems for breast cancer imaging: a critical review. *Applied Sciences*, 10(22), 8298.
- [5] Aljuaid, H., Alturki, N., Alsubaie, N., Cavallaro, L., \& Liotta, A. (2022). Computer-aided diagnosis for breast cancer classification using deep neural networks and transfer learning. *Computer Methods and Programs in Biomedicine*, 223, 1–10. <https://doi.org/10.1016/j.cmpb.2022.107125>
- [6] de Sampaio, W. B., Silva, A. C., de Paiva, A. C., & Gattass, M. (2015). Detection of masses in mammograms with adaption to breast density using genetic algorithm, phylogenetic trees, LBP and SVM. *Expert Systems with Applications*, 42(22), 8911-8928.
- [7] Saoud, H., Ghadi, A., Ghailani, M., \& Abdelhakim, B. A. (2019). Using feature selection techniques to improve the accuracy of breast cancer classification. In *Proceedings of the Third International Conference on Smart City Applications* (pp. 307–315).
- [8] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

- [9] Araujo, T., et al. (2017). Classification of breast cancer histology images using convolutional neural networks. *PLoS ONE*, 12(6), e0177544. <https://doi.org/10.1371/journal.pone.0177544>
- [10] Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D. \& Herrera, F. (2019). Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, 48, 220–250. <https://doi.org/10.1016/j.swevo.2019.05.006>
- [11] Yousefi, B., Akbari, H., Hershman, M., Kawakita, S., Fernandes, H. C., Ibarra-Castanedo, C., Ahadian, S., \& Mal dague, X. P. V. (2021). Spaer: Sparse deep convolutional autoencoder model to extract low dimensional imaging biomarkers for early detection of breast cancer using dynamic thermography. *Applied Sciences*, 11(7), 3248. <https://doi.org/10.3390/app11073248>
- [12] Hirra, I., Ahmad, M., Hussain, A., Ashraf, M. U., Saeed, I. A., Qadri, S. F., Alghamdi, A. M., \& Alfakeeh, A. S. (2021). Breast cancer classification from histopathological images using patch-based deep learning modeling. *IEEE Access*, 9, 24273–24287. <https://doi.org/10.1109/ACCESS.2021.3056842>
- [13] Sethy, P. K., Pandey, C., Khan, M. R., Behera, S. K., Vijaykumar, K., \& Panigrahi, S. (2021). A cost-effective computer-vision based breast cancer diagnosis. *Journal of Intelligent \& Fuzzy Systems*, 41(5), 5253–5263. <https://doi.org/10.3233/JIFS-189257>
- [14] Nahid, A.-A., Mikaelian, A., \& Kong, Y. (2018). Histopathological breast-image classification with restricted Boltzmann machine along with backpropagation. *Biomedical Research*, 29(10).
- [15] Araujo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., \& others. (2017). Classification of breast cancer histology images using convolutional neural networks. *PLoS ONE*, 12(6), e0177544. <https://doi.org/10.1371/journal.pone.0177544>
- [16] Pritom, A. I., Munshi, M. A. R., Sabab, S. A., \& Shihab, S. (2016). Predicting breast cancer recurrence using effective classification and feature selection technique. In 2016 19th International Conference on Computer and Information Technology

(ICCIT) (pp. 310–314). IEEE. <https://doi.org/10.1109/ICCITEchn.2016.7860210>

[17] Alzubaidi, A., Cosma, G., Brown, D., \& Pockley, A. G. (2016). Breast cancer diagnosis using a hybrid genetic algorithm for feature selection based on mutual information. In 2016 International Conference on Interactive Technologies and Games (iTAG) (pp. 70–76). IEEE. <https://doi.org/10.1109/iTAG.2016.15>

[18] Alshayeqi, M. H., Ellethy, H., \& Gupta, R. (2022). Computer-aided detection of breast cancer on the Wisconsin dataset: An artificial neural networks approach. *Biomedical Signal Processing and Control*, 71, 1–10. <https://doi.org/10.1016/j.bspc.2021.103182>

[19] Tsehay, Y., et al. (2017). Biopsy-guided learning with deep convolutional neural networks for prostate cancer detection on multiparametric MRI. In 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017) (pp. 642–645). IEEE. <https://doi.org/10.1109/ISBI.2017.7950646>

[20] Cai, S., Liu, P. Z., Luo, Y. M., Du, Y. Z., & Tang, J. N. (2019). Breast microcalcification detection algorithm based on contourlet and asvm Algorithms, 12(7), 135.

[21] Castells, X., Torá-Rocamora, I., Posso, M., Román, M., Vernet-Tomas, M., Rodríguez-Arana, A., ... & BELE Study Group. (2016). Risk of breast cancer in women with false-positive results according to mammographic features. *Radiology*, 280(2), 379-386.

[22] Rejani, Y., \& Thamarai Selvi, S. (2009). Early detection of breast cancer using SVM classifier technique. arXiv preprint arXiv:0912.2314.

[23] Chen, K. C., Chin, C. L., Chung, N. C., & Hsu, C. L. (2020). Combining multi-classifier with CNN in detection and classification of breast calcification. In Future Trends in Biomedical and Health Informatics and Cybersecurity in Medical Devices: Proceedings of the International Conference on Biomedical and Health Informatics, ICBHI 2019, 17-20 April 2019, Taipei, Taiwan (pp. 304-311). Springer International Publishing.

- [24] S. Rafi and R. Das, "RNN Encoder And Decoder With Teacher Forcing Attention Mechanism for Abstractive Summarization," 2021 IEEE 18th India Council International Conference (INDICON), Guwahati, India, 2021, pp. 1-7, doi: 10.1109/INDICON52576.2021.9691681.
- [25] S. Rafi and R. Das, "A Linear Sub-Structure with Co-Variance Shift for Image Captioning," 2021 8th International Conference on Soft Computing \& Machine Intelligence (ISCFI), Cario, Egypt, 2021, pp. 242-246, doi: 10.1109/ISCFI53840.2021.9654828.
- [26] Mordang, J. J., Janssen, T., Bria, A., Kooi, T., Gubern-Mérida, A., & Karssemeijer, N. (2016). Automatic microcalcification detection in multi-vendor mammography using convolutional neural networks. In Breast Imaging: 13th International Workshop, IWDM 2016, Malmö, Sweden, June 19-22, 2016, Proceedings 13 (pp. 35-42). Springer International Publishing.
- [27] S. Rafi and R. Das, "Abstractive Text Summarization Using Multimodal Information," 2023 10th International Conference on Soft Computing \& Machine Intelligence (ISCFI), Mexico City, Mexico, 2023, pp. 141-145, doi: 10.1109/ISCFI59957.2023.10458505.

# Computer Aided Detection Of Breast Cancer Using Bio Inspired Algorithm

Syed Rizwana, Peddi Kavya, Bolla Vinay Pooja, Bandi Poojitha, Shaik Khaja Mohiddin Basha, Doddha Venkatareddy

Department of Computer Science and Engineering, Narasaraopet Engineering College, Narasaraopet, Palnadu, India.

Email: rizwana.nec@gmail.com

**Abstract**—Breast cancer still ranks among the most common causes of cancer-related deaths among women, hence the call for early diagnosis. Mammography is the most accepted screening test, but conventional computer-aided detection (CAD) has a high false positive rate (FPR) that gives rise to biopsy and false negatives (FN) where cancer is undetected. In solving these challenges, this paper provides a solution by employing the use of the Simple Genetic Algorithm (SGA), which is openly inspired by biological systems to enhance the performance of CAD systems for breast cancer detection. The SGA, which is based on the evolutionary process, can resolve problems in feature selection and classification of the mammogram by overcoming shortcomings of pattern recognition. By mimicking the genetic evolution process, ant colony optimization, and swarm intelligence, the SGA prevents noisy or variant images from anyhow decreasing the detection accuracy. Comprehensive tests on typical sets of mammograms confirm the effectiveness of the proposed approach regarding a twofold reduction of inappropriate positive and negative results. This enhanced accuracy of diagnoses can help radiologists to act early, combined with favorable outcomes for the patients, implying that early diagnosis may save lives.

**Index Terms**—Breast Cancer Detection, Genetic Algorithm (GA) Feature Selection Guided Filter Image Enhancement Convolutional Neural Networks (CNN) Neural Networks Image Preprocessing Medical Image Analysis

## I. INTRODUCTION

Breast cancer is the leading cancer among women worldwide, of which it accounts for 25 % of all cancers and 685,000 fatalities yearly [1]. The time factor proves central to increasing the survival rate of performing minimally invasive procedures. Nevertheless, there are problems with conventional diagnostic methods even with the use of mammography, which is widely used in current practice. False positive results increase the number of biopsies that are done; false negatives delay treatments and shorten patients' lives. Further, current CAD systems are limited, especially in noisy image data, and do not have sufficient high cross-clinical variability generalization. In order to deal with these challenges, this work proposes a CAD framework based on the bio-inspired techniques, and more specifically, on the SGA. These algorithms are bio-inspired from the evolutionary biology in order to select features and classify mammography. Using guided image enhancement for preprocessing, LBP for feature extraction, and SGA for selecting features, the proposed approach provides a resolution to improve the reliability of diagnoses on a large scale. This work provides an organized methodology to approach the breast cancer detection problem and to eliminate false positives and

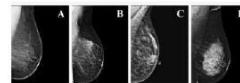


Fig. 1: Examples of the BI-RADS's four mammography density levels are as follows: From (A) a—virtually totally fatty; (B) b—fibroglandular density in scattered places; (C) c—heterogeneously dense; (D) d—very dense. [2]

negatives, increase classification performance, and overcome the drawbacks of current CAD systems. As a part of the continuous breakdown of the differences between research and practice concerning medical image analysis, our paper is primarily focused on the potential practical applications resulting from our findings.

Some well-known categories of bio-inspired algorithms encompass genetic algorithms (GA), particle swarm optimization (PSO), ant colony optimization (ACO), and artificial bee colony (ABC), which are derived from the mechanisms of living organisms or their parts. Therefore, in this work, the focus is on using bio-inspired algorithms in general and, more specifically, the Simple Genetic Algorithm (SGA) [3]. Genetic algorithms are overall a simulation of the natural procedure of evolution through selection, crossover, and mutations. Besides, they are very effective when it is required to search for the best solutions through large solution spaces. A simple genetic algorithm was preferred for this project because it is easy to implement, very stable, and provides good results for feature selection and classification. In this work, the SGA is effectively applied for the selection of the appropriate features from the images of mammograms for classification for enhancing the method of detection of breast cancer. These types of applications show that SGAs are highly proficient in increasing diagnostic reliability in medical image analysis in order to become the perfect tool for increasing cancer detection in mammograms. As noted in Araujo et al. (2017), the authors have also used genetic algorithms for classification of histology images for breast cancer. Second, the fitness function would be defined in the context of medical image analysis and classification applications only so that genetic feature selection [4] for TI would not be performed on the dataset while carrying out the proposed algorithm for designing the feature selection [5] procedure for mammogram images.

#### OBJECTIVES OF OUR WORK

- Preprocessing: Guided Image Enhancement  
Enhance the mammogram images and, in the process, come up with better images that can aid in the detection of the diseases.
- Feature Extraction: Local Binary Patterns (LBP)  
Propagation of the determined LBP to improve the features that are taken out of the mammography images to improve the likelihood of accurate categorization.
- Feature Selection: Simple Genetic Algorithm (SGA)  
Using SGA for feature selection helps to improve the results but at the same time reduces the number of false positives and false negatives.

#### II. RELATED WORK

Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, et al. [6] proposed “Bio-inspired computation: Where we stand and what’s next.” The bio-inspired computation has made research advances in the last few years, specifically in optimization, classification, and clustering. Thus, based on the bio-inspired algorithms, the primary advantages state that the method is very appropriate in approaching the solving of different complicated problems through flexibility and reliability in attaining the efficient solutions. Yousefi et al. [7] in their paper in 2021 employed a sparse deep convolutional autoencoder for early BC detection using dynamic thermography. This work is related to prior work on deep learning algorithms in medical images, autoencoders, reviews of dynamic thermography in detecting breast cancer, and sparse deep learning for feature extraction. Hirra et al. (2021) [8] apply a patch-based deep learning technique to distinguish between breast cancer types in histopathological images. Similar works encompass the works done on histopathology image analysis and cancer detection with deep learning algorithms by Cruz-Roa et al. (2017) and Cohen et al. on the use of a cheaper computer vision method for breast cancer diagnosis. Ronneberger et al. (2015) designed the U-Net architecture as part of their work. In the medical image segmentation domain, early cancer detection: A review on cost-effective methods by Zhao and Zhang (2019). Deep learning is applied in medical image diagnosis according to Rajpurkar et al. (2017), which is related to the diagnosis of breast cancer. Sinha and Singh (2020) discuss inexpensive imaging techniques, which are also cost-effective, much like the study by Sethy et al. [9]. Furthermore, Gong et al. (2018) offer a systematic review of computer vision for medical diagnosis, which is related to the paper under discussion. Nahid, Mikaelian & Kong (2018) [10] employ a restricted Boltzmann machine with a backpropagation algorithm to identify histopathological breast images.

#### III. PROPOSED METHODOLOGY

##### A. Preprocessing

The first phase of the methodology we proposed is to improve the quality of mammogram images using the guided image enhancement technique. It enhances signal contrast, the

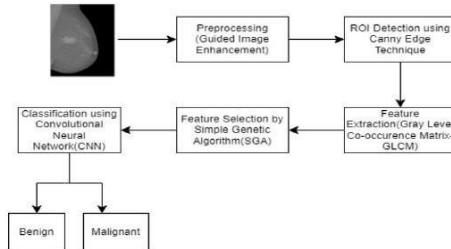


Fig. 2: Flowchart of Breast Cancer Detection

organizational borders of the images, and eliminates background noise, specifically in regions of interest (ROI), like possible tumors, which is shown in fig 3. This is important, for it enables subsequent steps in the CAD system to correctly identify abnormal regions. Further, we also employ the Contrast Limited Adaptive Histogram Equalization (CLAHE), where the contrast of regions in the image spatially varies, and it works well for spotting tiny details such as microcalcifications. This step is very vital in improving the quality of the image section as a way directed towards accommodating the various features extraction and processing. The preprocessing is to make the mammogram images clear, and the ROI detection is required to use the Canny Edge Detection to detect the likely suspicious parts. Subsequently, Gabor filtering is applied to capture edges, and next, feature extraction is done by using the Gray Level Co-occurrence Matrix (GLCM), which is used for capturing texture information. Finally, a simple genetic algorithm (SGA) is used to choose the highest discriminant features for a more enhanced classification.

##### B. ROI Edge Detection Using Canny Edge Technique

Canny edge detection is one of the essential methods used in the process of possibly separating the boundary of a tumor in the mammogram images. The next step in the considered approach is Canny edge detection, which helps to detect the contours of the tumors owing to intense changes in the image intensities. This step allows one to select the necessary areas that in all probability contain tumors for further work. These edges constitute the boundaries of the Regions of Interest (ROIs) that, once obtained, are subjected to feature extraction and classification. Thus, it can be concluded that, by providing correct definitions of these areas, Canny edge detection contributes significantly to improving the accuracy of stage detection in the overall deep learning stages.

##### C. Feature Extraction Using Gray Level Co-occurrence Matrix (GLCM)

In this step, the Gray Level Co-occurrence Matrix (GLCM) is used in the feature extraction of useful features within the regions of interest. GLCM studies the relationship of the intensity of two pixels and provides the number of occurrences when a given distance and direction are used. This technique



Fig. 3: A mammogram reveals a woman with moles that are both benign and worrisome. (B) The zoomed region that shows the MCs, denoted by the red square in (A). (C) The region in (B) where two extremely skilled radiologists have painstakingly highlighted the locations of the MCs (shown by the white arrow) [11]

allows for achieving quantitative measurements of secondary texture characteristics, including contrast, correlation, energy, and homogeneity, through which important data about the structure of the tissue is obtainable. In light of these advantages, we improve upon the model's capability to distinguish between benign and malignant regions seen in mammogram images and further improve the detection rate for cancer.

#### D. Feature Selection with Simple Genetic Algorithm (SGA)

Here we apply the simple genetic algorithm (SGA) for the best feature selection. The SGA operates in the same way as the natural evolution when it chooses the most useful features in the dataset. In its successive generations, such an algorithm assesses existing options of features and refines them by excluding non-conveying features. This process not only helps in simpler manipulation of the dataset but also in enriching the feature space for the model by corresponding with the most informative and significant features to help achieve better accuracy in the presaging of benign and malignant regions in mammogram images.

##### ALGORITHM: SIMPLE GENETIC ALGORITHM (SGA)

Input: Dataset with extracted features.

Output: Optimal feature subset for classification.

- 1) Initialization: Create a set of potential feature subsets.
- 2) Evaluation: Each subset is evaluated using a classifier to determine its performance.
- 3) Selection: Choose the subsets with higher accuracy based on observations.
- 4) Crossover: Combine selected feature subsets to create new subsets.
- 5) Mutation: Randomly alter some feature subsets to introduce diversity.
- 6) Replacement: Generate the next generation by reproducing from the current population.
- 7) Termination: Stop the algorithm when a termination condition is met and select the best feature subset.

#### E. Classification Using Convolutional Neural Network (CNN):

To extend this process, we use a Convolutional Neural Network (CNN) [12] classification technique for mammograms. The CNN then uses the descriptive features existing in the Regions of Interest (ROIs) to obtain the right patterns and representations to enable classification between benign and malignant classes. By having numerous layers of convolution,

the proposed network recognizes multi-level features to extract the images' patterns effectively.

TABLE I: Hyperparameter Settings for the Model

Hyperparameter	Value
Optimizer	Adam
Loss function	Binary Cross-Entropy
Batch Size	32
Epochs	11
Learning Rate	0.001
GA Feature Selection	622 features
GA Generations	11

**Optimization and Training:** This neural network was optimized with Adam Optimizer because it self-regulates the learning rate, thus making the process of training more efficient and faster. To quantify and calculate differences between the predicted and factual classification during the model training stage, a binary cross-entropy loss function was implemented. The overall process is explained in fig 2.

## IV. EXPERIMENTAL SETUP

### A. Dataset Description:

The dataset in use in this current research study is the DDSM, commonly known as the Digital Database for Screening Mammography.

TABLE II: Dataset Details for Breast Cancer Classification

Layer	Configuration
Dataset	DDSM (Digital Database for Screening Mammography)
Total Images	13,215
Training Split	80% (10,572 images)
Testing Split	20% (2,643 images)
Class labels	Benign, Malignant
Input Layer	Neurons=622 features (from GA)
Hidden Layer 1	Neurons=256, Activation=ReLU
Hidden Layer 2	Neurons=128, Activation=ReLU
Output Layer	Neurons=1, Activation=Sigmoid

### B. Implementation

The implementation starts with the procurement of the DDSM dataset, which consists of 13,215 screen-fetched images categorized into benign and malignant. It is also divided into a training set and a test set with 80% and 20%, respectively. Some preprocessing was applied to the images, which were resizing and normalization, and then we applied Canny edge detection to isolate the region of interest (ROI) for extracting features. From each image, four features are obtained using GLCM, and from all the images, a total of 622 features are selected using SGA [13]. A convolutional neural network has been used on these features, and its efficient architecture includes an input layer; two dense layers of 256 neurons that adjust the ReLU function; a third layer with 128 neurons adjusted to the ReLU function for binary categorization; and lastly, an output layer with a sigmoid function. Recall, F1-score, accuracy, and recalculated precision.



Fig. 4: Training and Testing Accuracy vs. Loss for Breast Cancer Classification Model

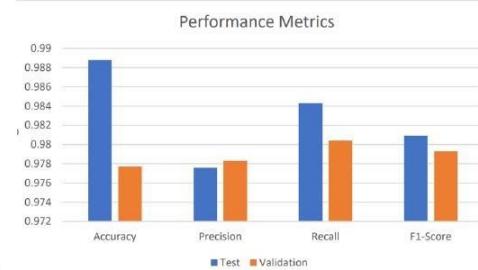


Fig. 5: Comparison of Performance Metrics for Test and Validation Data

## V. EVALUATION

We evaluated the process by using some formulas as mentioned

**Loss Function (Binary Cross-Entropy):** In classification tasks with binary outcomes (e.g., benign vs. malignant), the binary cross-entropy loss function is commonly used to measure the error.

$$Loss = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

Where:

N = No. of samples

y<sub>i</sub> = Actual label (0 for Benign, 1 for Malignant)

p<sub>i</sub> = Predicted probability for Malignant

**AUC (Area Under the ROC Curve):** In the context of a classification model, AUC holding different threshold values is applied for its evaluation. It determines the value of the curve of receiver operating characteristic (ROC).

$$AUC = \int_0^1 TPR(FPR) d(FPR)$$

TABLE III: Comparison of Feature Selection Methods in Breast Cancer Detection (2020-2023)

Reference	Year	Feature Selection Approach	Accuracy
Proposed Work	2024	Simple Genetic Algorithm (SGA)	98.88%
Bajer et al. [1]	2020	Genetic Algorithm (GA), Particle Swarm Optimization (PSO)	95%
Yousefi et al. [7]	2021	Dimensionality reduction using sparse encoding	93.5%
Hirra et al. [8]	2021	Deep learning-based feature extraction	96.2%
Sethy et al. [9]	2021	Feature extraction via CNN	94.3%
Alshayeqi et al. [13]	2022	ANN-based feature extraction	96%

## VI. RESULTS

Comparison of different feature selection algorithms helped in selecting features that could be suitable for mammographic image classification. Thus, among all algorithms used in

the experiment, the best result was obtained by the Genetic Algorithm (GA). That is why GA's ability to model adding evolutionary processes while selecting discriminative features produced significantly higher classification results. This was a much better improvement over Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) [14] [15]. As in the case with PSO, the algorithm offered well-rounded features with, however, less precise results than the ones found in GA. As for ACO, it was less helpful in increasing the classification accuracy, and this indicates some drawbacks of the method as the feature selection technique for the classification goals. Given the features chosen by GA, different classification models were compared, and CNN [16] provided the most accurate results. Through feature learning in mammographic images, complex patterns could be distinguished, and therefore, high classification was achieved by CNN.

Fig 5 shows the classification results based on the metrics identified as classification success rate, positive predictive value, and sensitivity with the harmonic mean of precision and recall using both the test and validation datasets. It can be seen that the model achieves good accuracy on both structured and unstructured sets, be it with a tad more recall for the test set data. F-measure and precision are quite similar in both the test and validation data, indicating acceptable performance of the method in identifying both benign and malignant cases. The evaluation assists in pointing out the feasibility of correctly diagnosing breast cancer using the chosen features and classification algorithms in the proposed model.

Figure 6 illustrates how the trained model and the Simple Genetic Algorithm (SGA) are used to classify breast cancer in its final form. At the left and right sides of the image, there are distinguishing characteristics between the benign and malignant cases, and a circle encloses the area where the

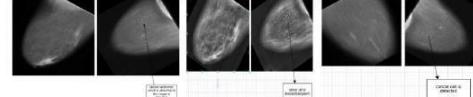


Fig. 6: Prediction of Breast Cancer Cells using Simple Genetic Algorithm

model picked out the cancerous region (tumor). Thus, using SGA for feature selection and classification, identified by the trained model, normal and cancerous tissues, thus achieving the correct identification of malignant areas in mammographic images.

## VII. CONCLUSION

This paper proves that the incorporation of GA into CAD for breast cancer detection is possible, yielding good results. The use of GA has greatly improved the ability to identify regions of interest in mammography images, improving the likelihood that breast cancer will be discovered. Through GA, there has been a significant improvement in reducing false positives and thus the depression and anxiety of patients and reducing unnecessary imaging. The work stresses the benefits of the use of GA in mammographic image analysis, indicating the usefulness of this approach as compared to other CAD systems. The efficiency with which GA is capable of both searching and exploiting feature spaces has provided a boost toward increasing the diagnostic accuracy of CAD, which makes the tool vital for the continued development of CAD technologies. Moreover, applicability and flexibility for dealing with high amounts of data and different imaging environments prove the addition of GA to be valuable. This makes SGA appear as a potentially beneficial approach to improving breast cancer diagnosis accuracy as well as procedural effectiveness. Further studies must be directed at searching for other algorithms that are inspired by nature as well as improving the time consumption of the GA. Exploring other imaging modalities and other detection scenarios can improve the utilization of GA in medical sciences and extend the contributions to the development of CAD systems.

## VIII. FUTURE WORK

Being under the class of heuristic algorithms, genetic algorithms (GA) hold tremendous prospects of expounding further within the different domains of study, especially in artificial intelligence and beyond. In the financial sector, it can be employed to enhance decision-making over investment, risk, and modeling. Trading algorithms, portfolio management, and fraud detection systems incorporated by the institutions can enhance decision-making as well as the overall accuracy of forecasts, thus minimizing operational risks. They also say that GA has good potential in the application of control, industrial automation, and robotics. GA is capable of influencing and optimizing sensor data analysis, control parameters, and system configuration to further improve the accuracy and flexibility of manufacturing and logistics, thereby improving the overall performance of a robotic system. In the sphere of environmental and resource management, GA can increase the efficiency of sensors, as well as data analysis of environmental and resource problems, and advance resource management methods. This can contribute towards improved utilization of natural resources, better control of pollution, and overall management and promotion of sustainable development. GA holds

potential for the telecommunication industry to achieve efficient networks, traffic management, and performance tuning. By further improving the configurations of the network as well as analyzing data, GA can improve the network throughput, reduce congestion, and therefore increase the reliability of the communications. In smart cities and infrastructure, Smart GA can contribute to traffic organization, energy provision, and other public services in general. The integration of such potential into the cities can make the cities better in terms of efficiency, sustainability, and responsiveness to the needs of the people.

## REFERENCES

- [1] Bajer, D., Zoric', B., Dudjak, M., & Martinovic', G. (2020). Benchmarking bio-inspired computation algorithms as wrappers for feature selection. *Acta electrotechnica et informatica*, 20(2), 35–43.
- [2] Spak, D. A., Plaxco, J. S., Santiago, L., Dryden, M. J., & Dogan, B. E. (2017). BI-RADS® fifth edition: A summary of changes. *Diagnostic and interventional imaging*, 98(3), 179-190.
- [3] Nakayevne, R., Ruan, X., Shen, Y., Shao, Y., Niu, C., Zang, Z., & Liu, F. (2021). Diagnostic Performance of SGA, PG-SGA and MUST for Malnutrition Assessment in Adult Cancer Patients: A Systematic Literature Review and Hierarchical Bayesian Meta-Analysis. *Nutrition and Cancer*, 74(3), 903–915. <https://doi.org/10.1080/01635581.2021.1942080>
- [4] Aalaei, S., Shahrazi, H., Rowhammanesh, A., & Eslami, S. (2016). Feature selection using genetic algorithm for breast cancer diagnosis: experiment on three different datasets. *Iranian journal of basic medical sciences*, 19(5), 476.
- [5] Bolon-Canedo, V., & Remesiro, B. (2020). Feature selection in image analysis: a survey. *Artificial Intelligence Review*, 53(4), 2905–2931. Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R., & Siel, W. (2019). Deep learning to improve breast cancer detection on screening mammography. *Scientific reports*, 9(1), 12495.
- [6] Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Canacho, D., ... & Herrera, F. (2019). Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*, 48, 220–250. <https://doi.org/10.1016/j.swevo.2019.05.006>
- [7] Yousefi, B., Akbari, H., Hershnman, M., Kawakita, S., Fernandes, H. C., Ibara-Castañedo, C., Ahadian, S., & Maldague, X. P. V. (2021). Spaer: Sparse deep convolutional autoencoder model to extract low dimensional imaging biomarkers for early detection of breast cancer using dynamic thermography. *Applied Sciences*, 11(7), 3248. <https://doi.org/10.3390/app11073248>
- [8] Hirra, I., Ahmad, M., Hussain, A., Ashraf, M. U., Saeed, I. A., Qadir, S. F., Alghamdi, A. M., & Alfaeekh, A. S. (2021). Breast cancer classification from histopathological images using patch-based deep learning modeling. *IEEE Access*, 9, 24273–24287. <https://doi.org/10.1109/ACCESS2021.3056842>
- [9] Sethy, P. K., Pandey, C., Khan, M. R., Behera, S. K., Vijaykumar, K., & Panigrahi, S. (2021). A cost-effective computer-vision based breast cancer diagnosis. *Journal of Intelligent & Fuzzy Systems*, 41(5), 5253–5263. <https://doi.org/10.3233/JIFS-189257>
- [10] Nahid, A.-A., Mikaelian, A., & Kong, Y. (2018). Histopathological breast-image classification with restricted Boltzmann machine along with backpropagation. *Biomedical Research*, 29(10).
- [11] Loizidou, K., Skouroumouni, G., Pittis, C., & Nikolaou, C. (2021). Digital subtraction of temporally sequential mammograms for improved detection and classification of microcalcifications. *European radiology experimental*, 5, 1–12.
- [12] Araujo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., & others. (2017). Classification of breast cancer histology images using convolutional neural networks. *PLoS ONE*, 12(6), e0177544. <https://doi.org/10.1371/journal.pone.0177544> Gardezi, S. J. S., A Elazab, B. Lei, and T. Wang, et al. 2019. Breast cancer detection and diagnosis using mammographic data: Systematic review. *Journal of medical Internet research* 21 (7):e14464. doi:<https://doi.org/10.2196/14464>
- [13] Pritom, A. I., Munshi, M. A. R., Sabab, S. A., & Shihab, S. (2016). Predicting breast cancer recurrence using effective classification and feature selection technique. In *2016 19th International Conference on*

- Computer and Information Technology (ICCIT) (pp. 310–314). IEEE. <https://doi.org/10.1109/ICCITEchn.2016.7860210>
- [14] Alshayeqi, M. H., Elletthy, H., & Gupta, R. (2022). Computer-aided detection of breast cancer on the Wisconsin dataset: An artificial neural networks approach. *Biomedical Signal Processing and Control*, 71, 1–10. <https://doi.org/10.1016/j.bspc.2021.103182>.
- [15] Tsehay, Y., et al. (2017). Biopsy-guided learning with deep convolutional neural networks for prostate cancer detection on multiparametric MRI. In 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017) (pp. 642–645). IEEE. <https://doi.org/10.1109/ISBI.2017.7950646>
- [16] Ting, F. F., Tan, Y. J., & Sim, K. S. (2019). Convolutional neural network improvement for breast cancer classification. *Expert Systems with Applications*, 120, 103–115.
- [17] S. Rafi and R. Das, "RNN Encoder And Decoder With Teacher Forcing Attention Mechanism for Abstractive Summarization," 2021 IEEE 18th India Council International Conference (INDICON), Guwahati, India, 2021, pp. 1-7, doi: 10.1109/INDICON52576.2021.9691681.

# IEEE\_Conference\_Template\_1\_.pdf

## ORIGINALITY REPORT



## PRIMARY SOURCES

1	<b>cris.maastrichtuniversity.nl</b> Internet Source	<b>1%</b>
2	<b>www.cse.unr.edu</b> Internet Source	<b>1%</b>
3	Oladimeji Macaulay, Michael Servilla, Kushal Virupakshappa, David Arredondo, Yue Hu, Luis Tafoya, Yanfu Zhang, Avinash Sahu. "MedGraphNet: Leveraging Multi-Relational Graph Neural Networks and Text Knowledge for Biomedical Predictions", Cold Spring Harbor Laboratory, 2024 Publication	<b>1%</b>
4	<b>academic-accelerator.com</b> Internet Source	<b>1%</b>
5	<b>ebin.pub</b> Internet Source	<b>1%</b>
6	Lalit Mohan Goyal, Tanzila Saba, Amjad Rehman, Souad Larabi-Marie-Sainte. "Artificial Intelligence and Internet of Things -	<1%

**Applications in Smart Healthcare", CRC Press,**

**2021**

Publication

---

**7** [mdpi-res.com](http://mdpi-res.com) **<1 %**  
Internet Source

---

**8** Submitted to Rochester Institute of  
Technology **<1 %**  
Student Paper

---

**9** Kosmia Loizidou, Rafaella Elia, Costas Pitrис.  
"Computer-aided breast cancer detection and  
classification in mammography: A  
comprehensive review", Computers in Biology  
and Medicine, 2023  
Publication

---

**10** Sammy Yap Xiang Bang, Syed M. Raza,  
Huigyu Yang, Hyunseung Choo. "EMP-GAN:  
Encoder-Decoder Generative Adversarial  
Network for Mobility Prediction", IEEE  
INFOCOM 2023 - IEEE Conference on  
Computer Communications Workshops  
(INFOCOM WKSHPS), 2023  
Publication

---

**11** Pallavi Bhanudas Salunkhe, Pravin Sahebrao  
Patil. "Rapid tri-net: breast cancer  
classification from histology images using  
rapid tri-attention network", Multimedia Tools  
and Applications, 2024  
Publication

- 12 V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024  
Publication <1 %
- 13 "Advances in Electrical and Computer Technologies", Springer Science and Business Media LLC, 2021  
Publication <1 %
- 14 www.researchgate.net <1 %  
Internet Source
- 15 "Computer Vision and Image Processing", Springer Science and Business Media LLC, 2022  
Publication <1 %
- 16 Brij B Gupta. "Cloud Security - Concepts, Applications and Perspectives", CRC Press, 2021  
Publication <1 %
- 17 Liebling, Alison, Maruna, Shadd, McAra, Lesley. "The Oxford Handbook of Criminology", The Oxford Handbook of Criminology, 2023  
Publication <1 %
- 18 deepai.org <1 %  
Internet Source
- 19 dokumen.pub

---

	Internet Source	<1 %
20	<a href="http://iieta.org">iieta.org</a> Internet Source	<1 %
21	<a href="http://www.uyik.org">www.uyik.org</a> Internet Source	<1 %
22	Lance D. Chambers. "The Practical Handbook of Genetic Algorithms - Applications, Second Edition", Chapman and Hall/CRC, 2019 Publication	<1 %

---

Exclude quotes Off  
Exclude bibliography On

Exclude matches Off



3<sup>rd</sup> IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IEEE IATMSI-2025)

## CERTIFICATE OF PRESENTATION

This is to certify that

Peddi Kanya (Narasaraopeta Engineering College-JNTUK)

Presented a Paper Titled

Computer Aided Detection Of Breast Cancer Using Bio Inspired Algorithm

in IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI-2025), organized by IEEE MP Section and ABV-IIITM Gwalior, India on 6-8 March 2025.

Dr. Arun Kumar

Organizing Chair, IATMSI-2025

Prof. G. S. Tomar

General Chair, IATMSI-2025

Dr. Somesh Kumar

Conference Chair, IATMSI-2025



3<sup>rd</sup> IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IEEE IATMSI-2025)

## CERTIFICATE OF AUTHORSHIP & ACCEPTANCE

This is to certify that Paper Titled

Computer Aided Detection Of Breast Cancer Using Bio Inspired Algorithm

### Submitted by Authors

Syed Rizwana, Peddi Kavya, Bolla Vinay Pooja, Bandi Poojitha, Shaik Khaja Mohiddin Basha, Dodda Venkata Reddy(Narasaraopeta Engineering College-JNTUk)

is accepted and presented in IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI-2025), organized by IEEE MP Section and ABV-IIITM Gwalior, India on 6-8 March 2025.

Dr. Arun Kumar  
Organizing Chair, IATMSI-2025

Prof. G. S. Tomar  
General Chair, IATMSI-2025

Dr. Somesh Kumar  
Conference Chair, IATMSI-2025