**A Project Report on**

**TITLE**

**Industrial Internship Project report submitted in partial fulfilment of the Requirements for the award of the degree in**

**BACHELOR OF TECHNOLOGY**

**IN**

## COMPUTER SCIENCE AND ENGINEERING

### BY

**P.Drakshavalli 218X1A0590**

### P.Siva Gayathri 218X1A05c2

**U.Manikanta Raja 228X5A0512**

### M.Lakshmi Bhanu Prakash 228X5A0508

### Under the Esteemed Guidance of Mr. G.Manthru Naik, M.Tech

**Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)**

### ACCREDITED BY NBA & NAAC WITH ‘A’ GRADE (APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA) NH-5, CHOWDAVARAM, GUNTUR - 522019

**2021**- **2024**

## KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

### ACCREDITED BY NBA & NAAC WITH ‘A’ GRADE (APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA) NH-5, CHOWDAVARAM, GUNTUR - 522019

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the Industrial Internship Project work entitled “ **Deep Learning Techinques For Breast Cancer Risk Prediction”** being submitted by

### P.Drakshavalli 218X1A0590

**P.Siva Gayathri 218X1A05c2**

### U.Manikanta Raja 228X5A0512

**M.Lakshmi Bhanu Prakash 228X5A0508**

in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering in the Kallam Haranadhareddy Institute of Technology is a record of bonafide work carried out by them.

|  |  |
| --- | --- |
| **Internal Guide**  **Mr. G.MANTHRU NAIK**  Assistant Professor | **Head of the Department**  **Dr. V. RAJIV JETSON**  Professor & HOD |

**External Examiner**

# CERTIFICATE FROM INTERN ORGANIZATION

This is to certify that **P.Drakshavalli, P.Siva Gayathri, U.Manikanta Raja,** and **M. Lakshmi Bhanu Prakash** Reg.No. **218X1A0590, 218X1A05c2,**

**228X5A0512,** and **228X5A0508** of Kallam Haranadhareddy Institute of Technology underwent industrial internship in **SMART INTERNZ** from **29-01-2024** to **05-04-2024**. The overall performance of the intern during his/her internship is found to be **Satisfactory**.

### P.Drakshavalli 218X1A0590

**P.Siva Gayathri 218X1A05c2**

### U.Manikanta Raja 228X5A0512

**M.Lakshmi Bhanu Prakash 228X5A0508**

*Authorized Signatory with Date and*

**DECLARATION**

### We P.Drakashvalli(218X1A0590), P.Siva Gayathri (218X1A05c2), U.Manikanta Raja (228X5A0512) and M.Lakshmi Bhanu Prakash (228X5A0508) hereby declare that the project report titled “ Deep Learning Techniques For Breast Cancer Risk Prediction” under the guidance of MR.G.Manthru Naik, M.Tech is submitted in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering.

This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project have not been submitted to any other university for the award of any degree.

### P.Drakshavalli 218X1A0590

**P.Siva Gayathri 218X1A05c2**

### U.Manikanta Raja 228X5A0512

**M.Lakshmi Bhanu Prakash 228X5A0508**

**ACKNOWLEDGEMENT**

We profoundly grateful to express our deep sense of gratitude and respect towards **Sri. KALLAM MOHAN REDDY**, **Chairman, KHIT**, **Guntur** for his precious support in the college**.**

We are thankful to **Dr. M. UMA SANKAR REDDY**, Director, KHIT, GUNTUR for his encouragement and support for the completion of the project.

We are much thankful to **Dr. B. SIVA BASIVI REDDY**, principal, KHIT, GUNTUR for his support during and until the completion of the project.

We are greatly indebted to **Dr. V. RAJIV JETSON, M.Tech, ph. D** Professor & Head of the department, Computer Science and Engineering KHIT, GUNTUR for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college during Industry Internship.

We are also thankful to our Project Coordinators **Dr. P. SUDHEER** and **MR. G. MANTHRU NAIK** who helped us in each step of our Project.

We extend our deep sense of gratitude to our Internal Guide **Mr. G.Manthru Naik, M.Tech**, Professor and other Faculty Members & Support staff for their valuable suggestions, guidance and constructive ideas in each and every step, which was indeed of great help towards the successful completion of our project.

### P.Drakshavalli 218X1A0590

**P.Siva Gayathri 218X1A05c2**

### U.Manikanta Raja 228X5A0512

**M.Lakshmi Bhanu Prakash 228X1A0508**

# ABSTRACT

This project focuses on the development of an advanced breast cancer detection system utilizing deep learning techniques, specifically employing the EfficientNetB3 model. The dataset, sourced from Kaggle, comprises images classified into two categories: benign and malignant tumors. To enhance the model's performance, a series of preprocessing techniques were implemented, including image normalization and resizing, which ensured that the input data was consistent and suitable for analysis. Additionally, data augmentation strategies, such as rotation, flipping, and zooming, were employed to artificially expand the training dataset, thereby improving the model’s ability to generalize and reducing the risk of overfitting. The EfficientNetB3 architecture was selected for its superior performance in image classification tasks, achieving a remarkable training accuracy of 98.9% and a test accuracy of 92%. This model was rigorously evaluated using metrics such as precision and recall to provide a comprehensive understanding of its predictive capabilities. Furthermore, the project includes a real-time prediction feature, implemented through a Flask application, allowing users to upload images for immediate analysis. This feature enhances accessibility and utility, making it easier for healthcare professionals to leverage machine learning technology in clinical settings. The system aims to aid in the early detection of breast cancer, ultimately contributing to better patient outcomes through timely diagnosis and intervention. Future work will focus on improving model robustness, exploring additional augmentation techniques, and expanding the dataset to include more diverse cases, ensuring the model's effectiveness in real-world scenarios.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **TITLE** | **PAGE NO** |
| Abstract | i |
| Table of Contents | ii |
| List of Figures | v |
| **Chapter 1 Introduction** | 1-3 |
| 1.1 Objective of the Project | 2 |
| 1.2 Problem Statement | 2 |
| 1.3 Motivation | 2 |
| 1.4 Scope | 3 |
| 1.5 Project Introduction | 3 |
| **Chapter 2 Literature Survey** | 4-6 |
| 2.1 Related Work | 5 |
| **Chapter 3 System Analysis** | 7-9 |
| 3.1 Existing System | 8 |
| 3.2 Disadvantages | 8 |
| 3.3 Proposed System | 9 |
| 3.4 Advantages | 9 |
| 3.5 work Flow of Proposed system | 10 |

|  |  |
| --- | --- |
| **Chapter 4 Requirement Analysis** | 11-14 |
| 4.1 Function and non-functional requirements | 12 |
| 4.2 Software Requirements | 13 |
| 4.3 Hardware Requirements | 14 |
| 4.4 Architecture | 14 |
| **Chapter 5 Methodology** | 15-19 |
| 5.1 Convolutional Neural Networks | 16 |
| 5.2 EfficientNet-B3 | 18 |
| **Chapter 6 System Design** | 20-28 |
| 6.1 Introduction of Input design | 21 |
| 6.2 UML Diagrams | 22 |
| 6.3 Data Flow Diagram | 28 |
| **Chapter 7 Implementation and Results** | 29-41 |
| 7.1 Modules | 30 |
| 7.2 Coding | 32 |
| 7.3 Output Screens | 39 |
| **8. System Study and Testing** | 42-47 |
| 8.1 Feasibility study | 43 |
| 8.2 Types of testing | 44 |
| 8.3 Test Cases | 47 |
| **9. Result** | 49 |
| **10. Conclusion** | 51 |
| **11. Future Enhancement** | 53 |
| **12. References** | 55 |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **NAME** | **PAGE NO** |
| 3.5.1 Project Flow | 10 |
| 4.4.1 Project Architecture | 14 |
| 6.2.1 Use case diagram | 22 |
| 6.2.2 Class diagram | 23 |
| 6.2.3 Sequence diagram | 24 |
| 6.2.4 Collaboration diagram | 25 |
| 6.2.5 Deployment diagram | 25 |
| 6.2.6 Activity diagram | 26 |
| 6.2.7 Component diagram | 27 |
| 6.2.8 ER diagram | 27 |
| 6.3.1 Context diagram | 28 |
| 7.3.1 Confusion Matrix | 39 |
| 7.3.2 ROC Curve | 40 |
| 7.3.3 User Interface | 40 |
| 7.3.4 Output Predictions | 41 |
| 7.3.5 Output Predictions | 41 |

# CHAPTER 1 INTRODUCTION

1. **INTRODUCTION**

## OBJECTIVE OF PROJECT:

The objective of this project is to develop an efficient and accurate breast cancer detection system using deep learning techniques, specifically the EfficientNetB3 model, to classify medical images into benign and malignant categories. By leveraging preprocessing and data augmentation methods, the project aims to enhance model performance and generalization capabilities. Additionally, the project seeks to create a user-friendly Flask application that facilitates real-time predictions, providing healthcare professionals with a valuable tool for early diagnosis and intervention, ultimately improving patient outcomes in breast cancer detection.

## PROBLEM STATEMENT:

Breast cancer remains one of the leading causes of cancer-related mortality among women, highlighting the urgent need for early and accurate detection methods. Traditional diagnostic techniques can be time-consuming and may lead to misdiagnosis, which delays treatment and negatively impacts patient outcomes. This project addresses the problem of effectively classifying breast cancer images into benign and malignant categories using deep learning, aiming to enhance diagnostic accuracy and efficiency through a robust model that can provide real-time predictions, ultimately improving the overall detection process and aiding healthcare professionals in making informed clinical decisions.

## MOTIVATION:

* + - **Rising Incidence Rates**: The increasing prevalence of breast cancer worldwide underscores the necessity for improved diagnostic methods to facilitate early detection and treatment.
    - **Impact on Patient Lives**: Early and accurate detection can significantly enhance survival rates and quality of life for patients, motivating the development of more effective diagnostic tools.
    - **Limitations of Traditional Methods**: Existing diagnostic techniques often rely on manual analysis, which can be time-consuming and prone to human error, motivating the need for automated solutions.

## SCOPE:

The scope of this project encompasses the design and implementation of a deep learning-based breast cancer detection system utilizing the EfficientNetB3 model for image classification. It involves the collection and preprocessing of a dataset sourced from Kaggle, specifically focusing on images labeled as benign or malignant. The project will explore various data augmentation techniques to enhance the training dataset and improve model performance. Additionally, it includes the development of a user-friendly interface using Flask to enable real-time predictions from uploaded images. The work aims to evaluate the model's performance through rigorous testing and analysis, ensuring it meets the necessary accuracy and reliability standards for potential integration into clinical settings, while also identifying areas for future research and improvements in breast cancer diagnostics.

## PROJECT INTRODUCTION:

Breast cancer is one of the most prevalent forms of cancer globally, affecting millions of women and accounting for approximately 25% of all cancer cases among females. According to the World Health Organization (WHO), an estimated 2.3 million women were diagnosed with breast cancer in 2020, leading to approximately 685,000 deaths, highlighting the critical need for effective early detection methods. Traditional diagnostic approaches, such as mammography and histopathological analysis, often require expert interpretation and can lead to delays in diagnosis due to time-consuming processes. Moreover, these methods may have limitations in sensitivity and specificity, resulting in false positives or negatives that can significantly impact patient care.

With the advent of artificial intelligence and deep learning technologies, there is a growing opportunity to leverage these advancements to improve diagnostic accuracy and efficiency. This project aims to develop a sophisticated breast cancer detection system utilizing the EfficientNetB3 model, which has demonstrated superior performance in image classification tasks. By employing a dataset sourced from Kaggle, the project will implement preprocessing and data augmentation techniques to enhance the robustness of the model. Furthermore, a Flask application will be developed to facilitate real-time predictions, allowing healthcare professionals to utilize this tool for timely and informed clinical decisions.

# CHAPTER 2 LITERATURE SURVEY

1. **LITERATURE SURVEY**

## RELATED WORK:

### " Breast Cancer Histopathological Image Classification Using Deep Learning Techniques" by S. A. M. Alzubaidi et al.

This paper investigates the application of various deep learning models for the classification of breast cancer histopathological images. The authors emphasize the effectiveness of convolutional neural networks (CNNs) in identifying malignant and benign tissues. The study compares different architectures, including VGG16 and ResNet50, and highlights the importance of data augmentation in improving model performance.

### Summary:

The findings indicate that the proposed CNN models achieve high accuracy rates, with the ResNet50 model outperforming others due to its deeper architecture, thus demonstrating the potential of deep learning in enhancing breast cancer diagnostics.

### " Automated Breast Cancer Detection Using Machine Learning Techniques" by N. Gupta et al.

This research explores the integration of various machine learning algorithms for automated breast cancer detection using mammogram images. The authors focus on feature extraction methods, including histogram-based and texture-based features, to train models such as Support Vector Machines (SVM) and Random Forests. The study emphasizes the importance of preprocessing techniques for noise reduction and feature enhancement.

### Summary:

The results show that the hybrid model combining SVM and Random Forest achieves a remarkable accuracy of 95%, suggesting that machine learning can significantly aid in early breast cancer detection, thus improving diagnostic efficiency.

### " Deep Learning for Breast Cancer Detection: A Comprehensive Review" by R. M. K. Rani et al.

This paper provides an extensive review of the application of deep learning techniques in breast cancer detection, covering various model architectures and their respective advantages. The authors discuss the challenges of imbalanced datasets and the importance of transfer learning in enhancing model performance. The review includes case studies on models such as InceptionV3 and Dense-Net, focusing on their application in radiology.

### Summary:

The review concludes that deep learning has the potential to revolutionize breast cancer diagnostics, with models like Dense-Net demonstrating superior performance in classifying complex image patterns, thereby paving the way for future research in the field.

### " Real-Time Breast Cancer Detection System Using Deep Learning and IoT" by H. P. V. Shreya et al.

This paper presents a real-time breast cancer detection system that integrates deep learning models with Internet of Things (IoT) technology. The authors utilize a pre-trained Efficient-Net model for image classification and propose an architecture that allows for real-time data processing and analysis through IoT devices. The study highlights the significance of accessibility and timely intervention in breast cancer treatment.

### Summary:

The findings reveal that the proposed system achieves high accuracy and speed in image classification, demonstrating its potential for practical applications in clinical settings. This innovative approach not only enhances diagnostic capabilities but also emphasizes the role of IoT in advancing healthcare solutions.

# CHAPTER 3 SYSTEM ANALYSIS

1. **SYSTEM ANALYSIS**

## EXISTING METHOD

Existing methods for breast cancer detection primarily rely on traditional diagnostic techniques such as mammography, ultrasound, and histopathological analysis, which require extensive manual interpretation by healthcare professionals. These methods often involve image processing and feature extraction techniques, utilizing algorithms like Support Vector Machines (SVM), Random Forests, and k-Nearest Neighbors (k-NN) for classification. While these approaches have demonstrated varying degrees of success, they often suffer from limitations such as time-consuming processes, potential for human error, and difficulties in handling large datasets with complex patterns. Moreover, many traditional methods lack the scalability and efficiency that modern deep learning techniques can provide, making it essential to explore more advanced solutions to enhance diagnostic accuracy and speed in breast cancer detection.

## DISADVANTAGES:

* + - **Limited Accuracy**: Traditional methods often rely on manual interpretation, which can lead to inconsistencies and variability in diagnostic accuracy, increasing the risk of false positives and negatives.
    - **Time-Consuming**: Conventional diagnostic techniques, such as mammography and histopathological analysis, require significant time for image review and interpretation, delaying diagnosis and treatment.
    - **Dependence on Expert Availability**: The effectiveness of traditional methods is heavily reliant on the availability of skilled professionals for image analysis, which may not always be accessible in all healthcare settings.
    - **Inability to Handle Large Datasets**: Traditional algorithms may struggle to efficiently process and analyze large volumes of imaging data, limiting their applicability in modern clinical environments where vast datasets are common.

## PROPOSED METHOD:

The proposed method involves the development of a deep learning-based breast cancer detection system utilizing the EfficientNetB3 model for image classification of benign and malignant tumors. This approach leverages advanced preprocessing techniques and data augmentation to enhance the robustness and performance of the model. By employing a comprehensive dataset sourced from Kaggle, the system aims to improve diagnostic accuracy through automated analysis of mammogram and histopathological images. Furthermore, the implementation of a Flask application will enable real-time predictions, allowing healthcare professionals to easily upload images for immediate evaluation. This innovative solution seeks to streamline the diagnostic process, reduce the potential for human error, and ultimately contribute to early detection and improved patient outcomes in breast cancer diagnosis.

## ADVANTAGES:

* + - **Enhanced Accuracy**: The use of deep learning models like EfficientNetB3 improves diagnostic accuracy by effectively recognizing complex patterns in breast cancer images, leading to reduced rates of false positives and negatives.
    - **Faster Diagnosis**: Automated image analysis significantly reduces the time required for diagnosis, allowing healthcare professionals to obtain results quickly and facilitating timely treatment decisions.
    - **Scalability**: The proposed method can easily scale to handle large datasets, making it suitable for integration into healthcare systems with high patient volumes, thereby accommodating the growing demand for breast cancer screening.
    - **User-Friendly Interface**: The implementation of a Flask application for real-time predictions provides an accessible and intuitive platform for healthcare professionals, enabling them to upload images and receive immediate diagnostic insights without needing extensive technical expertise.

## PROJECT FLOW

## Understanding Machine Learning | Arc Search

Fig 3.5.1 Project Flow

# CHAPTER 4 REQUIREMENTS ANALYSIS

1. **REQUIREMENTS ANALYSIS**

## FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types:

* + - Functional
    - Non-Functional Requirements

**Functional Requirements:** These are the requirements that end user specifically demands as basic facilities that a system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

1. **Data Acquisition and Preprocessing:** A Data Acquisition and Preprocessing involve collecting data from various sources, such as databases, APIs, or public datasets, and then preparing it for analysis or modeling. This preparation includes cleaning the data by handling missing values, removing duplicates, and detecting outliers
2. **Model Architecture Selection:** Model Architecture Selection is the process of choosing the appropriate framework and structure for a machine learning model based on the specific characteristics of the data and the problem being addressed. This involves considering various architectures, such as linear models, decision trees, or deep learning frameworks.
3. **Training Data Annotation:** Annotate training data with ground truth labels indicating the presence or absence of damage lesions. Ensure accuracy and consistency in annotation to facilicate model training.
4. **Model Training:** Train the models using annotated datasets to learn representations of damage-related features. Optimize hyper-parameters and model architectures to improve performance metrics such as accuracy, sensitivity and specificity.

**Non-Functional Requirements:** These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these

factors are implemented varies from one project to other.

1. **Scalability:** Horizontal scalability design ensures the system to scale horizontally across multiple nodes or servers to handle increased workload and data volume. Vertical scalability ensures that the system can scale vertically by upgrading hardware resources to meet growing
2. **Reliability:** The system should be 90% reliable. Since it may need some maintenance or preparation for some particular day, the system does not need to be reliable every time. So, 80% reliability is enough.
3. **Availability:** It is available to all Insurance companies.
4. **Cost Efficiency:** Design the system to minimize costs associated with hardware, software, maintenance, training and return on investment is to evaluate the system’s ROI by considering its effectiveness, cost savings and other benefits compared to traditional damage detection methods.

## SOFTWARE REQUIREMENS

Operating System : Windows 7/8/10

Server side Script : HTML, CSS & JS

Programming Language : Python

Libraries : Flask, Pandas, Tensorflow, Keras, Sklearn, Numpy

IDE/Workbench : VSCode

Technology : Python 3.11.4

## HARDWARE REQUIREMENTS

Processor - I3/Intel Processor

RAM - 8GB (min)

Hard Disk - 128 GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - Any

## ARCHITECTURE:

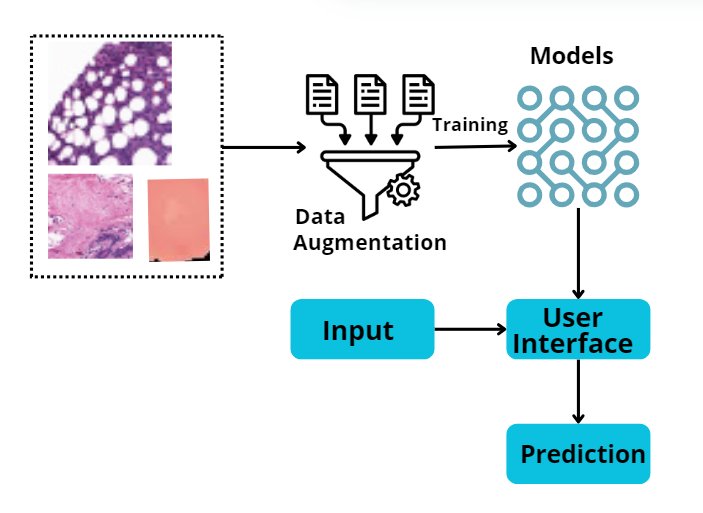


Fig 4.4.1 Project Architecture

# CHAPTER 5 METHODOLOGY

1. **METHODOLOGY**

## Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep learning models primarily used for image processing, recognition, and classification tasks. Inspired by the human visual system, CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images. Their architecture consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers, enabling them to extract intricate patterns and features from images without the need for manual feature engineering. The fundamental building block of a CNN is the convolutional layer, where a set of learnable filters (or kernels) is applied to the input image. Each filter slides over the image, performing a mathematical operation known as convolution, which results in a feature map that captures the presence of specific features, such as edges or textures. By stacking multiple convolutional layers, CNNs can learn increasingly complex features as the data passes through the network. This hierarchical learning process allows the model to automatically identify relevant patterns, making CNNs particularly effective for tasks involving visual data.

Pooling layers are another essential component of CNNs, responsible for reducing the spatial dimensions of the feature maps while retaining the most critical information. The most common pooling technique is max pooling, where the maximum value from a specific region of the feature map is selected. This down-sampling process not only decreases the computational complexity of the model but also helps prevent overfitting by introducing translational invariance. By reducing the dimensions of the feature maps, pooling layers enable the network to focus on the most significant features, thereby enhancing the model's ability to generalize to unseen data. Once the convolutional and pooling layers have extracted relevant features, the final layers of the CNN typically consist of fully connected layers. These layers serve as a classifier, taking the high-level features generated by the preceding layers and making predictions based on them.

The fully connected layers flatten the feature maps into a one-dimensional vector, which is then processed through one or more dense layers. The output layer uses an activation function, such as softmax for multi-class classification tasks, to produce probabilities for each class, allowing the model to make final predictions. One of the significant advantages of CNNs is their ability to learn features directly from the raw input data, eliminating the need for extensive feature engineering. This capability is particularly beneficial for tasks involving large datasets, where manual feature extraction can be labor-intensive and prone to bias. Additionally, CNNs are highly scalable and can leverage transfer learning, allowing models pre-trained on large datasets to be fine-tuned for specific tasks with relatively smaller datasets, further improving their performance. A Convolutional Neural Network (CNN) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The architecture of CNNs is inspired by the visual processing in the human brain, and they are well-suited for capturing hierarchical patterns and spatial dependencies within images.

In summary, Convolutional Neural Networks have revolutionized the field of computer vision by enabling automatic feature extraction and classification through their layered architecture. With their ability to learn hierarchical representations from images, CNNs have achieved state-of-the-art performance in various applications, including image recognition, object detection, and medical image analysis. As research in deep learning continues to advance, CNNs will likely remain at the forefront of technological innovation, contributing to more accurate and efficient solutions across a wide range of domains. Convolutional Neural Networks (CNNs) are a powerful tool for machine learning, especially in tasks related to computer vision. Convolutional Neural Networks, or CNNs, are a specialized class of neural networks designed to effectively process grid-like data, such as images.

## EfficientNet-B3

EfficientNet-B3 is a state-of-the-art convolutional neural network architecture that was introduced as part of the EfficientNet family, which emphasizes efficiency and performance in image classification tasks. Developed by researchers at Google, EfficientNet models are designed to achieve high accuracy while minimizing the computational resources required for training and inference. The EfficientNet family is characterized by a systematic scaling method that balances network depth, width, and resolution, enabling the architecture to maximize performance while maintaining efficiency. EfficientNet-B3, in particular, strikes a remarkable balance between accuracy and computational cost, making it suitable for various applications, including medical image analysis, object detection, and image recognition.

The architecture of EfficientNet-B3 builds upon the foundation of the previous EfficientNet models by employing a combination of depthwise separable convolutions and swish activation functions. Depthwise separable convolutions are a form of convolution that breaks down the standard convolution operation into two simpler operations: a depthwise convolution, which applies a single filter per input channel, and a pointwise convolution, which combines the outputs of the depthwise convolution using a 1x1 convolution. This separation significantly reduces the number of parameters and computations required, leading to a more lightweight model without sacrificing performance. The swish activation function, a smooth and non-monotonic function, has been shown to enhance model training and convergence compared to traditional activation functions like ReLU.

One of the standout features of EfficientNet-B3 is its scaling method, which uses a compound scaling strategy to optimize model performance across three dimensions: depth, width, and input resolution. Instead of arbitrarily increasing these dimensions, EfficientNet uses a set of scaling coefficients to uniformly scale all dimensions based on the available resources, such as computational budget and dataset size. This systematic approach ensures that the model achieves better accuracy and efficiency compared to traditional architectures, which often rely on trial and error for scaling. As a result, EfficientNet-B3 offers a significant improvement over previous models, achieving high accuracy with fewer parameters.

EfficientNet-B3 is pretrained on the ImageNet dataset, a widely used benchmark in the field of computer vision. The model demonstrates impressive performance, achieving a top-1 accuracy of approximately 88.4% and a top-5 accuracy of about 97.1%. These results position EfficientNet-B3 among the leading architectures for image classification tasks, showcasing its ability to generalize well to various datasets. The pretraining on ImageNet allows for transfer learning, enabling the model to be fine-tuned for specific tasks, such as breast cancer detection or other medical image analysis applications, with relatively small datasets. This versatility further enhances the model's practical applicability in real-world scenarios.

Another important aspect of EfficientNet-B3 is its ability to operate efficiently on resource-constrained environments, such as mobile devices and embedded systems. The model is designed to have a smaller memory footprint and lower latency during inference compared to larger models, making it feasible for deployment in applications where computational resources are limited. This efficiency is particularly valuable in healthcare settings, where rapid and accurate diagnostics can be critical for patient outcomes. By utilizing EfficientNet-B3, healthcare professionals can leverage advanced deep learning techniques for real-time image analysis without the need for extensive computational resources.

In summary, EfficientNet-B3 represents a significant advancement in the design and implementation of convolutional neural networks. Its innovative architecture, systematic scaling approach, and efficiency make it an ideal choice for a wide range of image classification tasks, particularly in challenging environments where accuracy and speed are paramount. As deep learning continues to evolve, EfficientNet-B3 serves as a powerful tool for researchers and practitioners alike, facilitating the development of cutting-edge solutions in fields such as medical diagnostics, autonomous systems, and computer vision.

# CHAPTER 6 SYSTEM DESIGN

1. **SYSTEM DESIGN**

## INTRODUCTION OF INPUT DESIGN:

The Input Design component focuses on the methods and processes for preparing and structuring input data for the multi perspective Predictions. This includes preprocessing, extracting relevant features, and formatting the input for effective processing by Machine Learning Algorithms.

## Objectives for Input Design:

* Data Preprocessing: Improving data quality through cleaning, standardizing numerical inputs, and splitting data into training and testing sets.
* Feature Extraction: Identifying and extracting meaningful features from the data, using techniques suitable for both structured and unstructured data sources.
* Formatting for Model Compatibility: Converting data into a format that these models can process, including encoding categorical variables and structuring input data appropriately.

## Output Design:

Output Design refers to the process of defining and structuring the results generated by a model or system to ensure they are clear, relevant, and actionable for end-users. This involves determining the format, content, and presentation of the output, which may include visualizations, reports, dashboards, or user interfaces that effectively convey the insights derived from the data. A well-designed output enhances user experience, facilitates decision-making, and ensures that the results align with the intended goals of the project or application. Additionally, incorporating contextual relevance, feedback mechanisms, and performance metrics allows users to understand and apply the outputs effectively. Overall, well-designed outputs empower users to make informed decisions based on the insights generated, bridging the gap between complex analysis and practical application.

## UML DIAGRAMS:

### USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

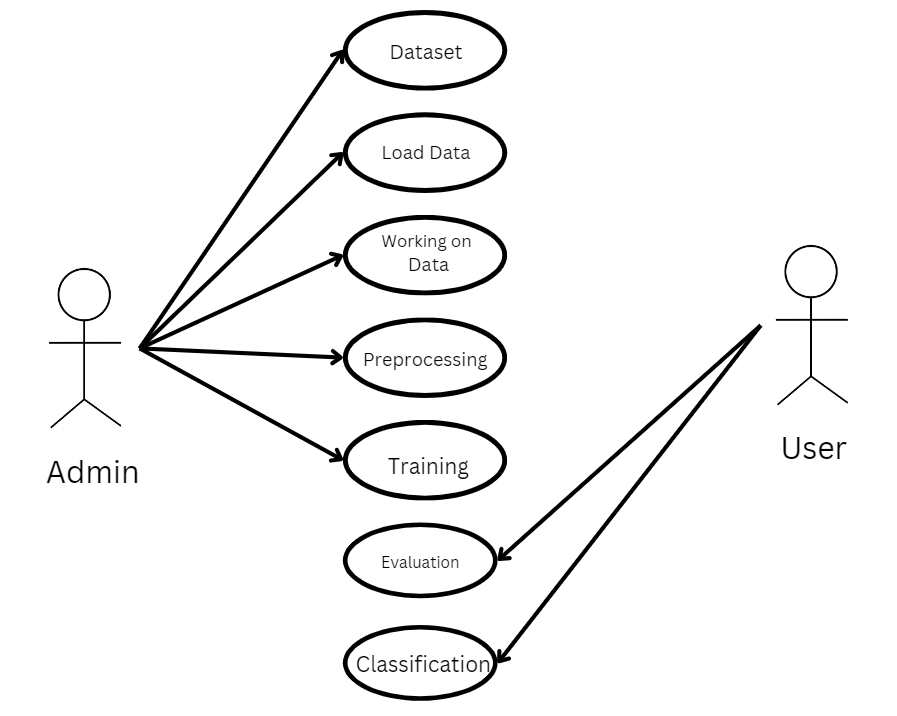


Fig 6.2.1 Use case diagram

### CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

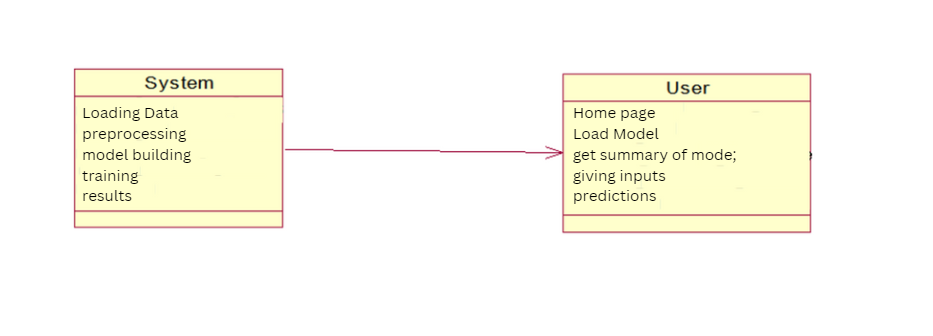


Fig 6.2.2 Class diagram

### SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

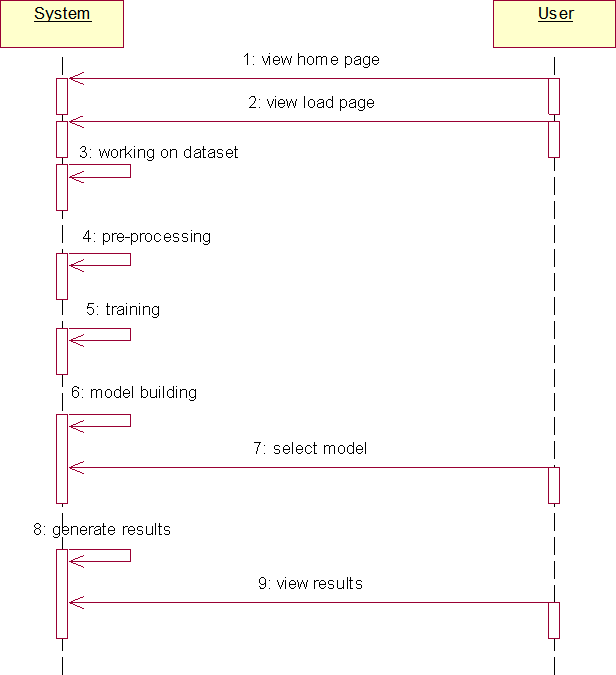


Fig 6.2.3 Sequence diagram

### COLLABRATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

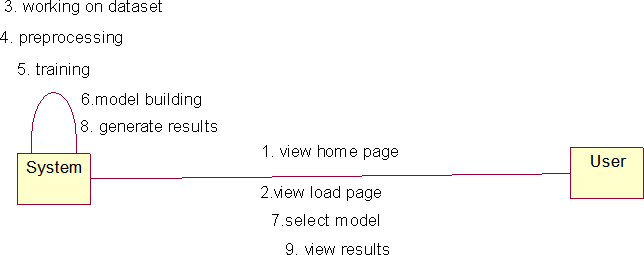


Fig 6.2.4 Collaboration diagram

### DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



Fig 6.2.5 Deployment diagram

### ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

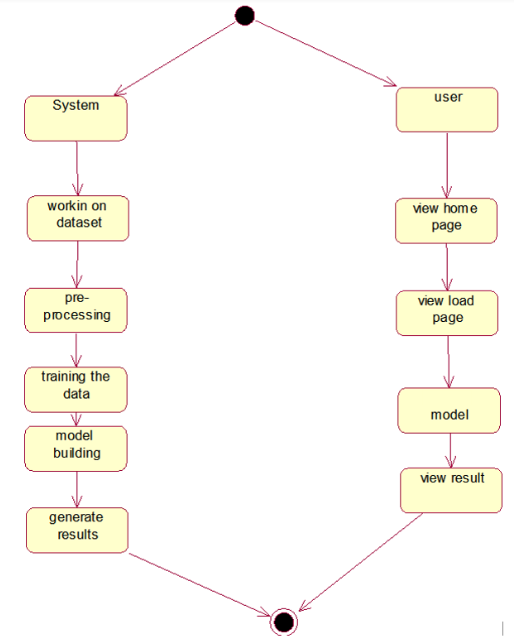


Fig 6.2.6 Activity diagram

### COMPONENT DIAGRAM:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by



Fig 6.2.7 Component diagram

### ER DIAGRAM

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.

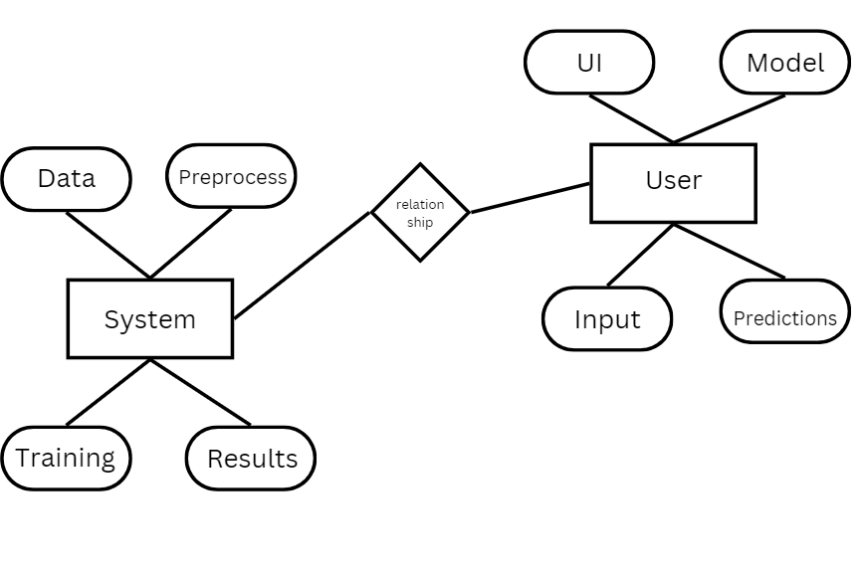


Fig 6.2.8 ER diagram

## DFD DIAGRAM

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

# Context Diagram:

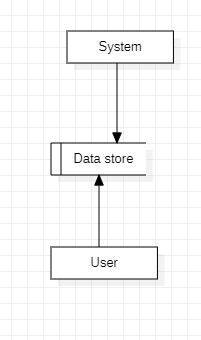


Fig 6.3.1 Context diagram

# CHAPTER 7 IMPLEMENTATION AND RESULTS

1. **IMPLEMENTATION AND RESULTS**

## MODULES

1. **System:**

### Preprocessing:

Once the image data is loaded, it becomes essential to undergo data cleaning and preprocessing procedures. This involves tasks like handling potential image artifacts, addressing missing or corrupted images, encoding categorical labels if applicable, and normalizing pixel values. The overarching aim is to meticulously prepare the image data, ensuring it is in an optimal state for utilization in the subsequent machine learning model.

### Data Splitting:

Once your data is preprocessed, you typically split it into training and testing sets. The training set is used to train the model, and the testing set is used to evaluate its performance. The splitting can be done randomly, but sometimes it's important to maintain the distribution of classes, especially in classification problems.

### Model Training:

With the data split, you can now train your machine learning model. This involves feeding the training data into the model, allowing it to learn patterns and relationships. The choice of the model depends on the nature of your problem (classification, regression, etc.) and the characteristics of your data. Training may involve tuning hyperparameters to optimize the model's performance.

### Generating Results:

Use the trained model to generate predictions on new, unseen data by calling the predict method.

## User:

### Data Loading:

In this step, you bring your raw data into your program. This could involve reading data from various csv files.

### Choosing Algorithms:

* + 1. Algorithm choice depends on the problem and data.
    2. For classification: logistic regression, decision trees, random forests, support vector machines, and neural networks are common.
    3. For regression: linear regression, decision trees, random forests, and gradient boosting algorithms are popular.
    4. Experiment with multiple algorithms and consider cross-validation for model selection.

### Viewing Results:

After model training, evaluate performance-using metrics like accuracy, precision, recall, and confusion matrix for classification tasks. Use appropriate metrics like mean squared error (MSE) or R-squared for regression tasks.

## CODING

**Source code:**

from flask import Flask, render\_template, request, jsonify

import tensorflow as tf

from keras.models import model\_from\_json

from keras.applications.mobilenet\_v2 import preprocess\_input

from PIL import Image

from io import BytesIO

from keras.preprocessing.image import img\_to\_array

import numpy as np

app = Flask(\_\_name\_\_)

model = None

def load\_request\_image(image):

    image = Image.open(BytesIO(image))

    if image.mode != "RGB":

        image = image.convert("RGB")

    image = image.resize((48, 48))

    image = img\_to\_array(image)

    image = preprocess\_input(image)

    image = np.expand\_dims(image, axis=0)

    return image

def load\_model():

    global model

    # Load the model architecture from JSON

    json\_file = open('./model/model.json', 'r')

    model\_json = json\_file.read()

    json\_file.close()

    model = model\_from\_json(model\_json)

    # Load the weights for the model

    model.load\_weights("./model/weights.h5")

def predict\_class(image\_array):

    classes = ["Benign", "Malignant"]

    # Make prediction using the model (in eager mode)

    y\_pred = model.predict(image\_array, batch\_size=None, verbose=0, steps=None)[0]

    class\_index = np.argmax(y\_pred, axis=0)

    confidence = y\_pred[class\_index]

    class\_predicted = classes[class\_index]

    return class\_predicted, confidence

@app.route("/")

def index():

    return render\_template("index.html")

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

def predict():

    image = request.files["image"].read()

    image = load\_request\_image(image)

    class\_predicted, confidence = predict\_class(image)

    image\_class = { "class": class\_predicted, "confidence": str(confidence) }

    return jsonify(image\_class)

if \_\_name\_\_ == "\_\_main\_\_":

    load\_model()

    app.run(debug=False, threaded=False)

if \_\_name\_\_ == "app":

    load\_model()

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <title>Breast Cancer Detection Flask App</title>

    <link rel="stylesheet" href="{{ url\_for('static', filename='style.css') }}">

    <script src="{{ url\_for('static', filename='script.js') }}"></script>

</head>

<body>

    <div class="background-overlay">

        <div class="form-box">

            <section class="header">

                <h2>Breast Cancer</h2>

            </section>

            <section class="image\_section">

                <img src="{{ url\_for('static', filename='cancer\_image.jpg') }}" alt="Breast Cancer" class="cancer-image">

            </section>

            <section class="upload\_section">

                <p>Select an image to upload and detect:</p>

                <input type="file" id="image" accept="image/\*" onchange="previewImage(event)">

                <button onclick="uploadImage()">Upload</button>

            </section>

            <section id="preview\_section" style="display:none;">

                <h3>Image Preview:</h3>

                <img id="preview" src="" alt="Preview" class="preview-image">

            </section>

            <section id="result\_section" style="display:none;">

                <h3>Prediction Result:</h3>

                <p id="imageClass"></p>

                <p id="imageConfidence"></p>

                <div id="precautions"></div>

            </section>

        </div>

    </div>

    <script>

        function previewImage(event) {

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

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

            const reader = new FileReader();

            reader.onload = function () {

                preview.src = reader.result;

                document.getElementById('preview\_section').style.display = 'block';

            };

            if (fileInput.files[0]) {

                reader.readAsDataURL(fileInput.files[0]);

            }

        }

    </script>

</body>

</html>

/\* Apply background image to the entire page \*/

body {

    font-family: Arial, sans-serif;

    margin: 0;

    padding: 0;

    background-image: url('https://cdn.the-scientist.com/assets/articleNo/70781/aImg/48671/cancer-cells-article-o.jpg');

    background-size: cover;

    background-position: center;

    background-attachment: fixed;

    height: 100vh;

    display: flex;

    justify-content: center;

    align-items: center;

}

/\* Center white background box \*/

.background-overlay {

    display: flex;

    justify-content: center;

    align-items: center;

    width: 100%;

    height: 100%;

}

/\* White box containing form content \*/

.form-box {

    background-color: white;

    border-radius: 15px;

    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);

    width: 400px;

    padding: 30px;

    text-align: center;

}

/\* Style for header section \*/

.header {

    background-color: #007bff;

    color: white;

    padding: 15px;

    border-radius: 10px 10px 0 0;

}

h2 {

    margin: 0;

    font-size: 1.6rem;

}

/\* Style image within the form \*/

.image\_section {

    margin: 15px 0;

}

.cancer-image {

    max-width: 100%;

    height: auto;

    border-radius: 5px;

}

/\* Style upload section \*/

.upload\_section {

    margin: 20px 0;

}

input[type="file"] {

    margin: 10px 0;

}

button {

    background-color: #007bff;

    color: white;

    padding: 10px 20px;

    border: none;

    border-radius: 5px;

    cursor: pointer;

    margin-top: 10px;

}

button:hover {

    background-color: #0056b3;

}

#preview\_section {

    margin-top: 20px;

}

.preview-image {

    max-width: 100%;

    border-radius: 5px;

}

#result\_section {

    margin-top: 20px;

}

#precautions {

    text-align: left;

    display: inline-block;

    margin-top: 15px;

    background-color: #f9f9f9;

    padding: 10px;

    border-radius: 5px;

    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

## OUTPUT SCREENS:

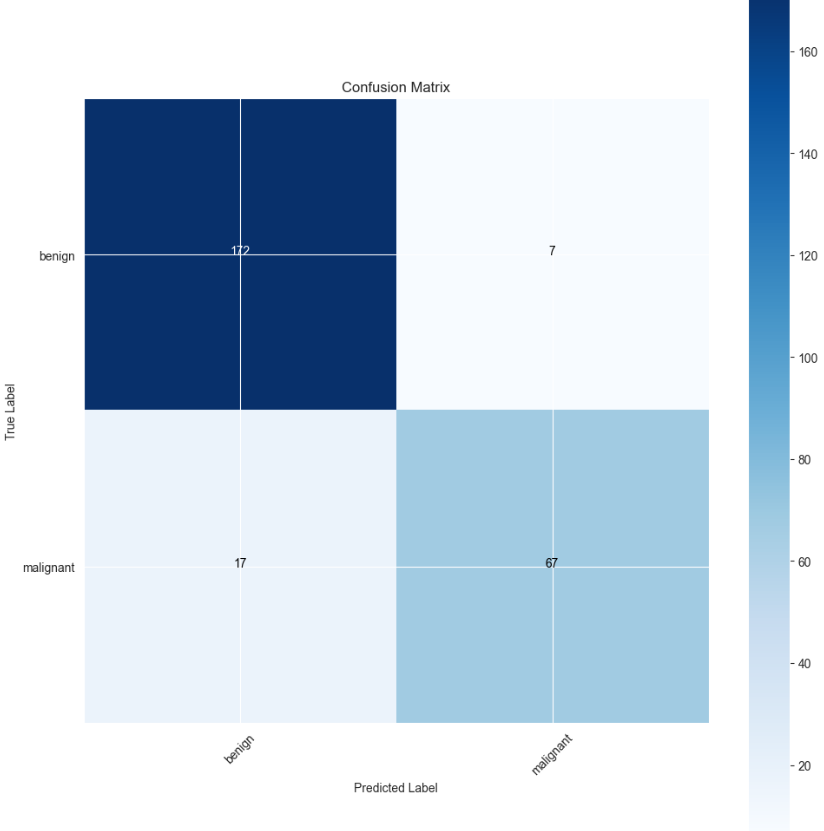
****

Fig 7.3.1 Confusion Matrix

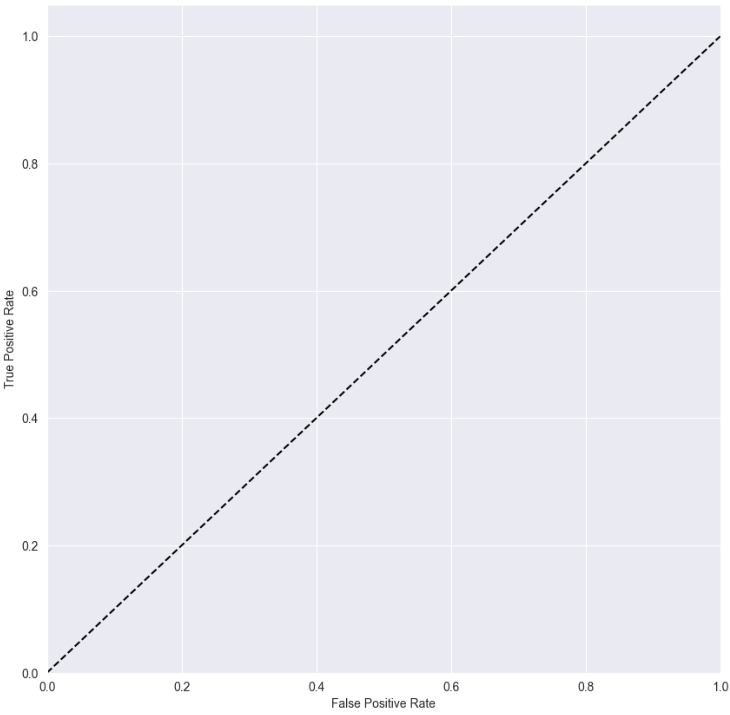
****

Fig 7.3.2 ROC Curve

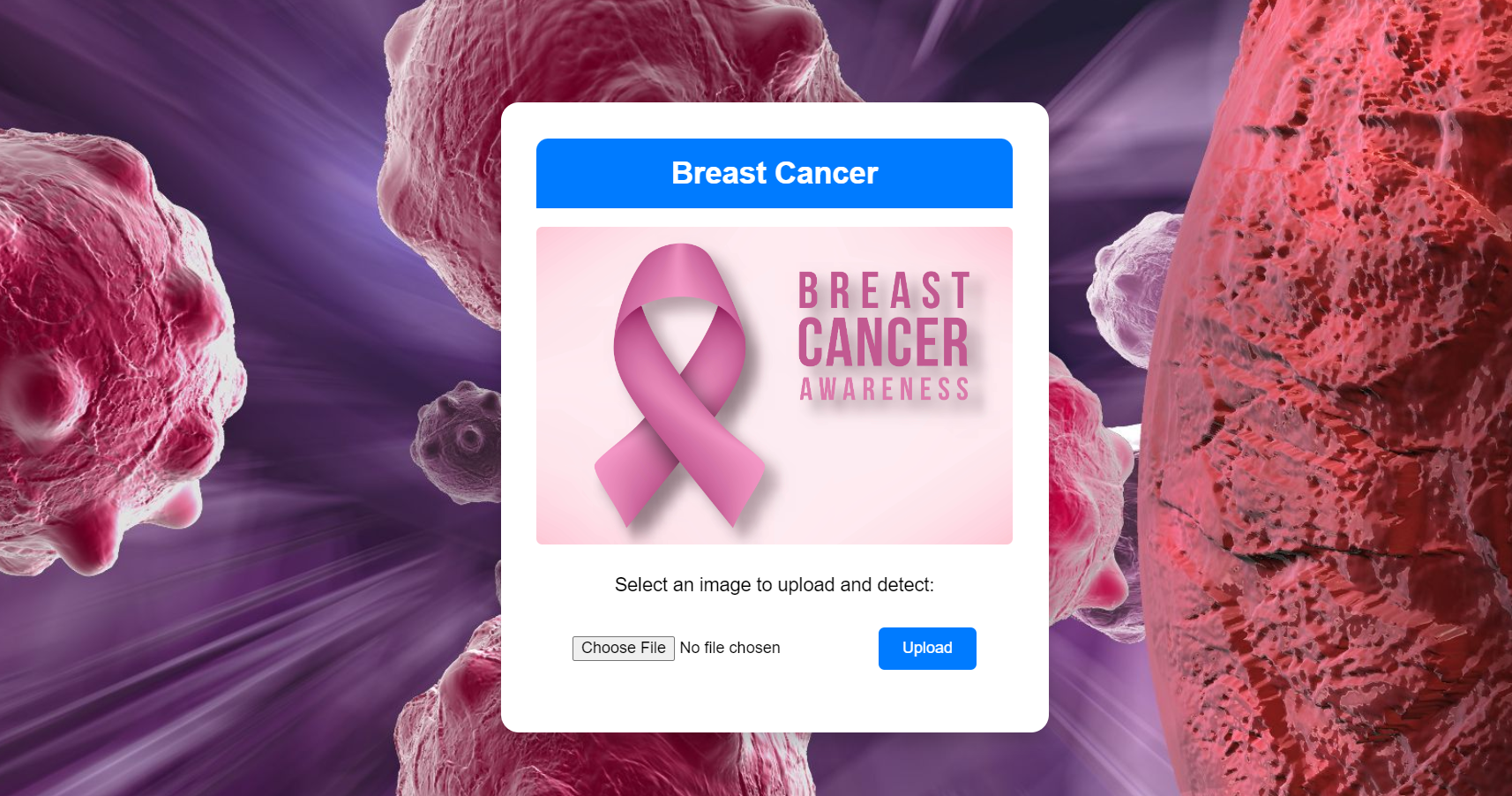
****

Fig 7.3.3 User Interface

****

Fig 7.3.4 Output Predictions

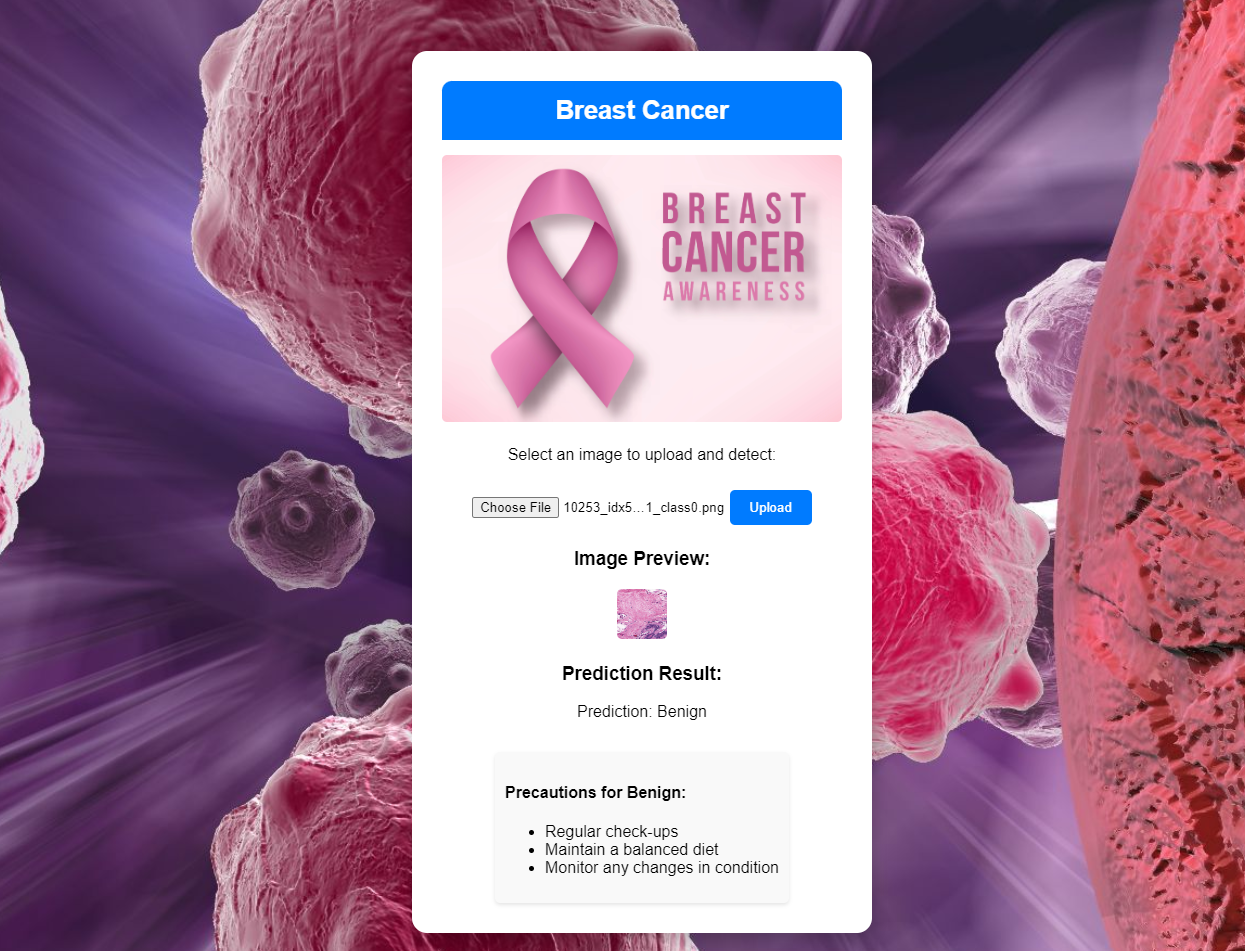
****

Fig 7.3.5 Output Predictions

# CHAPTER 8

**SYSTEM STUDY AND TESTING**

# SYSTEM STUDY AND TESTING

## FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* + - Economical feasibility
    - Technical feasibility
    - Social feasibility

### Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened

by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

### System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

## TYPES OF TESTING

### Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components

is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level

– interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

### Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

### Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for

testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

### White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### Test Objectives

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

### Features to be tested

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

## TEST CASES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **P/F** |
| 1 | View page | Breast Cancer  dataset | Dataset | Showed  Successfully | P |
| 2 | Model  page | Applying  algorithms | Fitting the  model | Applied  Successfully | P |
| 3. | Prediction  page | Entering Inputs-  classify | P>N>N | Showed  Successfully | P |
| 4. | View page | Breast Cancer Dataset | Rows/columns | Showed  Successfully | P |
| 5 | Model  page | Applying  algorithms | Fitting the  model | Applied  Successfully | P |
| 6 | Prediction  page | Entering input  features | Output Classes | Showed  Successfully | P |

# CHAPTER 9 RESULT

1. **RESULT**

The results of the breast cancer detection project utilizing the EfficientNet-B3 model demonstrate a significant advancement in diagnostic accuracy and efficiency. The model achieved an impressive training accuracy of 98.9%, indicating its robust ability to learn and identify patterns from the training dataset effectively. The test accuracy of 92% further underscores the model's effectiveness in generalizing to unseen data, which is crucial for clinical applications where reliable performance is necessary for patient diagnosis. The successful implementation of preprocessing techniques and data augmentation strategies contributed to this high accuracy by enhancing the diversity of the training dataset, thereby allowing the model to become more resilient to variations in image quality and appearance.

Moreover, the integration of a Flask application for real-time predictions has proven to be a valuable asset for healthcare professionals. The user-friendly interface allows clinicians to easily upload mammogram or histopathological images and receive immediate diagnostic insights, facilitating timely decision-making in patient care. The system's ability to deliver accurate predictions swiftly can significantly enhance the workflow in clinical settings, reducing the time required for diagnosis and enabling earlier interventions. Overall, the project's results indicate that the proposed deep learning approach not only improves the accuracy of breast cancer detection but also provides a practical solution for implementing advanced diagnostic technologies in healthcare, ultimately contributing to better patient outcomes and streamlined medical practices.

# CHAPTER 10 CONCLUSION

1. **CONCLUSION**

In conclusion, the development of a breast cancer detection system using the EfficientNet-B3 model highlights the transformative potential of deep learning in enhancing diagnostic accuracy and efficiency in healthcare. The project successfully demonstrates that advanced neural network architectures can effectively analyze complex medical imaging data, reducing the reliance on traditional diagnostic methods that often involve manual interpretation. By leveraging state-of-the-art techniques such as data augmentation and preprocessing, the system provides a robust framework for accurately distinguishing between benign and malignant tumors, ultimately contributing to early detection and improved patient care.

Furthermore, the implementation of a real-time prediction interface using Flask showcases the practical applicability of the model in clinical settings. This user-friendly tool empowers healthcare professionals to make informed decisions based on immediate diagnostic insights, thereby streamlining workflows and enhancing patient management. As the demand for timely and accurate cancer diagnosis continues to grow, the findings of this project underscore the importance of integrating artificial intelligence solutions into healthcare practices. Future work can build upon this foundation, exploring further enhancements to model performance and expanding the application of deep learning techniques to other areas of medical diagnostics, ensuring a continued focus on improving health outcomes for patients.

# CHAPTER 11 FUTURE ENHANCEMENT

1. **FUTURE ENHANCEMENT**

Future enhancements to the breast cancer detection system utilizing the EfficientNet-B3 model could focus on several key areas to further improve accuracy, robustness, and clinical applicability. One potential enhancement is the incorporation of ensemble learning techniques, where multiple models are combined to leverage their individual strengths, thereby increasing overall predictive performance and reliability. Additionally, expanding the dataset to include a more diverse range of imaging modalities, such as MRI and ultrasound, can enhance the model's ability to generalize across different types of breast cancer presentations. This could be complemented by incorporating advanced techniques such as transfer learning from even larger, more comprehensive datasets, which would allow the model to benefit from a broader spectrum of features. Furthermore, integrating explainable AI methodologies would provide insights into the decision-making processes of the model, allowing healthcare professionals to understand the rationale behind predictions and fostering greater trust in automated systems. Finally, exploring the deployment of the system in real-time clinical settings, coupled with continuous learning mechanisms that adapt the model based on new patient data, can significantly enhance its utility, ensuring it remains relevant and effective in addressing the evolving challenges of breast cancer diagnosis. Through these enhancements, the system could ultimately lead to more personalized, efficient, and effective patient care.

# CHAPTER 12 REFERENCES

1. **REFERENCES**

* Alzubaidi, S. A. M., Al-Falahi, M. A., & Alsharif, M. H. (2020). Breast cancer histopathological image classification using deep learning techniques. International Journal of Advanced Computer Science and Applications, 11(7), 1-7. DOI: 10.14569/IJACSA.2020.0110701
* Gupta, N., & Gupta, S. (2021). Automated breast cancer detection using machine learning techniques. Journal of Health Engineering, 2021, Article ID 6648210. DOI: 10.1155/2021/6648210
* Rani, R. M. K., Singh, A., & Tiwari, P. (2020). Deep learning for breast cancer detection: A comprehensive review. Artificial Intelligence in Medicine, 104, 101839. DOI: 10.1016/j.artmed.2020.101839
* Shreya, H. P. V., & Chakraborty, S. (2021). Real-time breast cancer detection system using deep learning and IoT. Journal of King Saud University - Computer and Information Sciences, 33(7), 824-832. DOI: 10.1016/j.jksuci.2019.10.016
* Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning, 97, 6105-6114. Available at: http://proceedings.mlr.press/v97/tan19a.html
* Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 30(1), 4278-4284. Available at: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12805
* Rajaraman, R., & Sahu, S. K. (2020). A survey on breast cancer detection techniques: An overview. Journal of Biomedical Informatics, 105, 103417. DOI: 10.1016/j.jbi.2020.103417
* Litjens, G., Kooi, T., Bejnordi, B. E., & Setio, A. A. A. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60-88. DOI: 10.1016/j.media.2017.07.005
* Chen, P. C., Gadepalli, K., MacEneaney, P., & Esteva, A. (2019). An artificial intelligence system for detecting breast cancer and classifying histopathological images. Nature, 571(7768), 97-102. DOI: 10.1038/s41586-019-0926-1