A

Major Project Report

On

Diabetic Retinopathy Detection using CNN

Submitted to **CMREC, HYDERABAD**

In Partial Fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING -DATA SCIENCE

Submitted By

Y. VAISHNAVI (218R1A67C9)
P. VYSHNAVI (218R1A67B8)
K. SUSHEEL KUMAR (218R1A6794)
M. KOUSHIK PRASANNA (218R1A67A8)

Under the Esteemed guidance of

Mr. V. KIRAN KUMAR (Ph.D.)

Assistant Professor, Department of CSE (Data Science)

Department of Computer Science & Engineering-Data Science

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, Medchal Dist,
Hyderabad -501 401

2024-2025

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Accredited by NBA Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, Hyderabad-501 401

Department of Computer Science & Engineering-Data Science

CERTIFICATE

This is to certify that the project entitled **"Diabetic Retinopathy Detection using CNN"** is a Bonafide work carried out by

Y. VAISHNAVI (218R1A67C9)
P. VYSHNAVI (218R1A67B8)
K. SUSHEEL KUMAR (218R1A6794)
M. KOUSHIK PRASANNA (218R1A67A8)

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this Major project have been verified and are found to be satisfactory. The results embodied in this Major project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	Major Project Coordinator	Head of the Department	External Examiner
Mr. V. Kiran Kumar (Ph.D.)	**Mr. B. Kumara Swamy** (Ph. D, JNTUH)	**Dr. M. Laxmaiah**	
Assistant Professor CSE (Data Science), CMREC, Hyderabad	Assistant Professor CSE (Data Science), CMREC, Hyderabad	Professor & H.O.D CSE (Data Science), CMREC, Hyderabad	

DECLARATION

This is to certify that the work reported in the present Major project entitled " **Diabetic Retinopathy Detection using CNN"** is a record of Bonafide work done by us in the Department of Computer Science and Engineering (Data Science), CMR Engineering College, UGC Autonomous, JNTU Hyderabad. The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this Major project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

Y. VAISHNAVI (218R1A67C9)
P. VYSHNAVI (218R1A67B8)
K. SUSHEEL KUMAR (218R1A6794)
M. KOUSHIK PRASANNA (218R1A67A8)

ACKNOWLEDGMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. M. Laxmaiah,** HOD**, Department of CSE (Data Science), CMR Engineering College** for their constant support.

We are extremely thankful to **Mr. V. Kiran Kumar** (Ph. D.), Assistant Professor, Internal Guide, Department of CSE(DS), for his/ her constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if We do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We thank **Mr. B .Kumara Swamy** (Ph. D, JNTUH) Assistant Professor ,CSE(DS) Department , Major Project Coordinator for his constant support in carrying out the project activities and reviews.

We express our thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, We are very much thankful to our parents who guided me for every step.

Y. VAISHNAVI (218R1A67C9)
P. VYSHNAVI (218R1A67B8)
K. SUSHEEL KUMAR (218R1A6794)
M. KOUSHIK PRASANNA (218R1A67A8)

ABSTRACT

Diabetic Retinopathy (DR) is a common diabetes-related complication that damages the retina and can result in vision loss if not detected and treated early. Traditional DR detection methods are often subjective, time-consuming, and resource-intensive, making early detection challenging. Developing an automated system for DR detection using fusion-based Convolutional Neural Networks (CNNs) can address these limitations by classifying fundus images into five categories: no DR, mild, moderate, severe, and proliferative DR. Such a system enables timely intervention, reducing the risk of vision impairment and enhancing patient outcomes. Automated DR detection improves efficiency by alleviating the burden of manual interpretation, allowing ophthalmologists to prioritize critical cases and allocate resources effectively. Leveraging deep learning techniques, particularly CNNs, offers the ability to identify intricate patterns in fundus images, facilitating accurate classification of DR severity. The proposed system achieves high performance, with the Xception model demonstrating superior accuracy and robustness in classifying DR stages. This approach supports cost-effective, scalable, and precise DR screening, ultimately improving healthcare delivery and patient quality of life.

Key words : Diabetic Retinopathy, Deep Learning, Convolutional Neural Network, Xception, ResNet50, DiaNet, InceptionV3.

CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1. INTRODUCTION

Diabetic Retinopathy (DR) is a retinal vascular disease that affects individuals with diabetes, and its prevalence is expected to double in the coming years. The primary factor contributing to the development of DR is the duration of diabetes, as prolonged high blood sugar levels increase the risk of retinal damage. Despite this, many diabetic patients are often unaware of the potential for DR, leading to delays in diagnosis and treatment. Traditionally, DR diagnosis relies on the analysis of digital color fundus images, which must be reviewed by qualified clinical professionals. This process is time-consuming and can result in delayed outcomes, meaning patients may not receive the necessary follow-up care in a timely manner. As a result, the condition may worsen, leading to complications such as macular edema, vision loss, and in some cases, permanent blindness. Non-proliferative diabetic retinopathy (NPDR), characterized by retinal swelling, small blood vessel leakage, and blood vessel closure, can impair vision if left untreated. Additionally, macular ischemia and exudate formation are other types of NPDR that further contribute to vision deterioration. Currently, the manual testing performed by ophthalmologists is resource-intensive and inefficient, creating a need for automated systems that can improve the speed and accuracy of DR detection.

1.1 Objective:

Diabetic Retinopathy (DR) is a leading cause of vision impairment and blindness among diabetic patients. Early detection is crucial for timely intervention and treatment, reducing the risk of severe vision loss. This project aims to develop an accurate and efficient DR detection system using fusion-based Convolutional Neural Network (CNN) approaches, integrating deep learning and advanced image processing techniques. The proposed system leverages a fusion-based CNN model that combines multiple deep learning architectures to enhance feature extraction and classification accuracy. By integrating complementary features from different CNN architectures, the model improves sensitivity and specificity in detecting various DR stages, including mild, moderate, severe, and proliferative retinopathy.

Key innovations include multi-scale feature fusion, attention mechanisms for improved focus on pathological regions, and optimized preprocessing techniques to enhance image clarity. Additionally, the model is trained on diverse datasets to improve generalization across different demographic groups and imaging conditions.

The automated DR detection system offers a scalable solution for mass screening, enabling early diagnosis and reducing dependency on specialist ophthalmologists. By integrating into telemedicine platforms, it facilitates remote screening in underserved regions, improving access to quality eye care. The implementation of such a system can significantly reduce the burden on healthcare systems while ensuring timely and accurate diagnosis for diabetic patients.

This project contributes to the advancement of AI-driven medical diagnostics, reinforcing the role of deep learning in healthcare. By developing a robust and efficient DR detection system, it aims to improve patient outcomes, prevent vision loss, and enhance overall public health.

1.2 Problem Statement:

- Diabetic Retinopathy (DR) is caused by prolonged high blood sugar levels damaging blood vessels in the retina, leading to vision loss if undetected or untreated early.
- Traditional DR detection methods are time-consuming, subjective, and require extensive resources, resulting in delayed diagnoses and treatment, which can worsen the condition and increase healthcare costs.
- DR primarily affects diabetic patients, especially those with uncontrolled blood sugar levels. It poses a significant risk to individuals aged 40 and above, leading to vision impairment and blindness.
- The lack of early detection leads to irreversible vision damage, reduced quality of life, increased healthcare costs, and places a significant burden on healthcare systems globally.
- We propose developing an automated DR detection system using fusion-based CNN techniques to enhance early diagnosis, improve accuracy, and reduce the burden on healthcare professionals and resources.

1.2 Applications

1. **Automated Screening**

   - Enables large-scale, cost-effective DR screening in hospitals and clinics.
   - Reduces dependency on manual screening, improving early detection rates.
   - Minimizes the need for frequent hospital visits, benefiting both patients and healthcare providers.

2. **Telemedicine & Remote Diagnosis**

   - Facilitates DR screening in rural and underserved areas with limited access to ophthalmologists.
   - Allows remote evaluation of retinal images, ensuring timely diagnosis and treatment.
   - Enhances accessibility for diabetic patients, reducing geographical barriers to healthcare.

3. **Improved Accuracy**

   - Fusion-based CNNs combine multiple feature extraction techniques, improving detection precision.
   - Reduces false positives and false negatives, leading to reliable DR classification.
   - Outperforms traditional manual examinations by identifying subtle retinal abnormalities at an early stage.

4. **Treatment Monitoring**

- Tracks disease progression over time by analysing sequential retinal images.
- Helps ophthalmologists evaluate treatment effectiveness and make necessary adjustments.
- Reduces the chances of disease worsening due to late intervention.

5. **Reduced Workload for Ophthalmologists**

- AI-powered systems prioritize high-risk cases, enabling specialists to focus on urgent patients.
- Enhances workflow efficiency by automating preliminary screenings, reducing fatigue among doctors.
- Frees up medical resources for more complex cases requiring human expertise.

6. **Better Patient Outcomes**

- Ensures timely intervention, significantly reducing the risk of severe vision impairment or blindness.
- Encourages proactive healthcare management among diabetic patients.
- Improves overall healthcare system efficiency, benefiting both patients and providers.

By integrating AI-driven DR detection into healthcare systems, medical professionals can improve accuracy, accessibility, and efficiency, ultimately leading to better patient care and outcomes.

CHAPTER-2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 A Faster RCNN-Based Diabetic Retinopathy Detection Method Using Fused Features From Retina Images:

https://ieeexplore.ieee.org/abstract/document/10305557

ABSTRACT: Early identification of diabetic retinopathy (DR) is critical as it shows few symptoms at the primary stages due to the nature of its gradual and slow growth. DR must be detected at the early stage to receive appropriate treatment, which can prevent the condition from escalating to severe vision loss problems. The current study proposes an automatic and intelligent system to classify DR or normal condition from retina fundus images (FI). Firstly, the relevant FIs were pre-processed, followed by extracting discriminating features using histograms of oriented gradient (HOG), Shearlet transform, and Region-Based Convolutional Neural Network (RCNN) from FIs and merging them as one fused feature vector. By using the fused features, a machine learning (ML) based faster RCNN classifier was employed to identify the DR condition and DR lesions. An extended experiment was carried out by employing binary classification (normal and DR) from three publicly available datasets. With a testing accuracy of 98.58%, specificity of 97.12%, and sensitivity of 95.72%, this proposed faster RCNN deep learning technique with feature fusion ensured a satisfactory performance in identifying the DR compared to the relevant state-of-the-art works. By using a generalization validation strategy, this fusion-based method achieved a competitive performance with a detection accuracy of 95.75%.

2.2 Hybrid Methods for Fundus Image Analysis for Diagnosis of Diabetic Retinopathy Development Stages Based on Fusion Features:

https://www.mdpi.com/2075-4418/13/17/2783

ABSTRACT: Diabetic retinopathy (DR) is a complication of diabetes that damages the delicate blood vessels of the retina and leads to blindness. Ophthalmologists rely on diagnosing the retina by imaging the fundus. The process takes a long time and needs skilled doctors to diagnose and determine the stage of DR. Therefore, automatic techniques using artificial intelligence play an important role in analyzing fundus images for the detection of the stages of DR development. However, diagnosis using artificial intelligence techniques is a difficult task and passes through many stages, and the extraction of representative features is important in reaching satisfactory results. Convolutional Neural Network (CNN) models play an important and distinct role in extracting features with high accuracy. In this study, fundus images were used for the detection of the developmental stages of DR by two proposed methods, each with two systems. The first proposed method uses GoogLeNet with SVM and ResNet-18 with SVM. The second method uses Feed-Forward Neural Networks (FFNN) based on the hybrid features extracted by first using GoogLeNet, Fuzzy color histogram (FCH), Gray Level Co-occurrence Matrix (GLCM), and Local Binary Pattern (LBP); followed by ResNet-18, FCH, GLCM and LBP. All the proposed methods obtained superior results. The FFNN network with hybrid features of ResNet-18, FCH, GLCM, and LBP obtained 99.7% accuracy, 99.6% precision, 99.6% sensitivity, 100% specificity, and 99.86% AUC.

2.3 Identification of Diabetic Retinopathy Using Weighted Fusion Deep Learning Based on Dual-Channel Fundus Scans:

https://www.mdpi.com/2075-4418/12/2/540

ABSTRACT: It is a well-known fact that diabetic retinopathy (DR) is one of the most common causes of visual impairment between the ages of 25 and 74 around the globe. Diabetes is caused by persistently high blood glucose levels, which leads to blood vessel aggravations and vision loss. Early diagnosis can minimise the risk of proliferated diabetic retinopathy, which is the advanced level of this disease, and having higher risk of severe impairment. Therefore, it becomes important to classify DR stages. To this effect, this paper presents a weighted fusion deep learning network (WFDLN) to automatically extract features and classify DR stages from fundus scans. The proposed framework aims to treat the issue of low quality and identify retinopathy symptoms in fundus images. Two channels of fundus images, namely, the contrast-limited adaptive histogram equalization (CLAHE) fundus images and the contrast-enhanced canny edge detection (CECED) fundus images are processed by WFDLN. Fundus-related features of CLAHE images are extracted by fine-tuned Inception V3, whereas the features of CECED fundus images are extracted using fine-tuned VGG-16. Both channels' outputs are merged in a weighted approach, and softmax classification is used to determine the final recognition result. Experimental results show that the proposed network can identify the DR stages with high accuracy. The proposed method tested on the Messidor dataset reports an accuracy level of 98.5%, sensitivity of 98.9%, and specificity of 98.0%, whereas on the Kaggle dataset, the proposed model reports an accuracy level of 98.0%, sensitivity of 98.7%, and specificity of 97.8%. Compared with other models, our proposed network achieves comparable performance.

2.4 Hemorrhage Detection Based on 3D CNN Deep Learning Framework and Feature Fusion for Evaluating Retinal Abnormality in Diabetic Patients:

https://www.mdpi.com/1424-8220/21/11/3865

ABSTRACT: Diabetic retinopathy (DR) is the main cause of blindness in diabetic patients. Early and accurate diagnosis can improve the analysis and prognosis of the disease. One of the earliest symptoms of DR are the hemorrhages in the retina. Therefore, we propose a new method for accurate hemorrhage detection from the retinal fundus images. First, the proposed method uses the modified contrast enhancement method to improve the edge details from the input retinal fundus images. In the second stage, a new convolutional neural network (CNN) architecture is proposed to detect hemorrhages. A modified pre-trained CNN model is used to extract features from the detected hemorrhages. In the third stage, all extracted feature vectors are fused using the convolutional sparse image decomposition method, and finally, the best features are selected by using the multi-logistic regression controlled entropy variance approach. The proposed method is evaluated on 1509 images from HRF, DRIVE, STARE, MESSIDOR, DIARETDB0, and DIARETDB1 databases and achieves the average accuracy of 97.71%, which is superior to the previous works. Moreover, the proposed hemorrhage detection system attains better performance, in terms of visual quality and quantitative analysis with high accuracy, in comparison with the state-of-the-art methods.

2.5 Automated detection and classification of fundus diabetic retinopathy images using synergic deep learning model:

https://www.sciencedirect.com/science/article/abs/pii/S0167865520300714

ABSTRACT: In recent days, the incidence of Diabetic Retinopathy (DR)has become high, affecting the eyes because of drastic increase in the glucose level in blood. Globally, almost half of the people under the age of 70 gets severely affected by diabetes. In the absence of earlier recognition and proper medication, the DR patients tend to lose their vision. When the warning signs are tracked down, the severity level of the disease has to be validated so to take decisions regarding appropriate treatment further. The current research paper focuses on the concept of classification of DR fundus images on the basis of severity level using a deep learning model. This paper proposes a deep learning-based automated detection and classification model for fundus DR images. The proposed method involves various processes namely preprocessing, segmentation and classification. The methods begins with preprocessing stage in which unnecessary noise that exists in the edges is removed. Next, histogram-based segmentation takes place to extract the useful regions from the image. Then, Synergic Deep Learning (SDL) model was applied to classify the DR fundus images to various severity levels. The justification for the presented SDL model was carried out on Messidor DR dataset. The experimentation results indicated that the presented SDL model offers better classification over the existing models.

CHAPTER-3

EXISTING SYSTEM

3.1 EXISTING SYSTEM:

Diabetic Retinopathy (DR) is a severe complication of diabetes that affects the retina and can lead to blindness if left untreated. Early detection and classification of DR are crucial for effective treatment and management. Existing DR detection systems primarily rely on machine learning techniques and traditional image processing methods to classify DR from retina fundus images (FIs). These methods involve pre-processing the images to enhance quality, feature extraction to identify distinguishing patterns, and classification using machine learning models.

While these techniques have demonstrated promising accuracy levels, they present several limitations in real-world clinical applications. The reliance on traditional feature extraction methods and manual classification restricts their effectiveness, robustness, and scalability. This document highlights the key disadvantages of existing DR detection systems and explores the need for a more advanced deep learning-based approach to improve performance and reliability.

3.1.1. Existing DR Detection Systems

Most existing DR detection methods use a combination of machine learning and image processing techniques. These approaches typically involve the following steps:

1. **Pre-Processing**: Enhancing fundus image quality to remove noise and improve clarity.

Feature Extraction: Identifying patterns associated with DR using methods such as:

- Histograms of Oriented Gradients (HOG)
- Shearlet Transform
- Region-Based Convolutional Neural Networks (RCNN)

Feature Fusion: Combining multiple extracted features into a single vector to improve classification accuracy.

Classification: Employing machine learning classifiers such as Faster RCNN to detect lesions and classify images into normal or DR.

Validation: Evaluating system performance using publicly available datasets.

While these systems have achieved high accuracy in binary classification (normal vs. DR), they still face challenges in handling real-world clinical data, differentiating DR stages, and improving generalization.

3.2 Disadvantages of Existing Systems

3.2.1 Limitation in Feature Extraction

1. **Traditional methods like HOG, Shearlet Transform, and Faster RCNN may not effectively capture complex patterns**:

- DR manifests through intricate retinal changes such as microaneurysms, haemorrhages, and exudates, which may not be effectively represented by manually engineered features.
- Deep learning models, particularly Convolutional Neural Networks (CNNs), can automatically learn hierarchical representations, making them more suitable for DR detection.

2. **Manual feature engineering limits adaptability**:

- Traditional feature extraction methods require domain expertise and manual tuning, making them less scalable to diverse datasets.
- Variations in fundus images due to different imaging devices, patient conditions, and illumination levels can impact model performance, requiring constant re-adjustments.

3.2.2 Reduced Robustness in Large-Scale Data Handling

1. **Machine learning classifiers like Faster RCNN are less effective in handling large-scale, complex datasets**:

- Traditional classifiers require well-defined features, which may not be sufficient for large datasets with high variability.
- CNN-based approaches, on the other hand, automatically learn relevant features from vast amounts of data, improving scalability and accuracy.

2. Poor generalization to unseen data:

- Many traditional models are trained on publicly available datasets but fail to perform well on new, unseen clinical images due to domain shifts.
- This reduces the real-world applicability of existing systems, necessitating retraining with diverse data.

3.2.3 Limited Classification Capabilities

1. Binary classification (normal vs. DR) restricts clinical usefulness:

- DR progresses through multiple stages, requiring a system that can differentiate between mild, moderate, severe, and proliferative DR.
- Current systems lack the ability to provide a granular analysis of DR severity, which is critical for treatment planning.

2. Inability to detect specific retinal abnormalities:

- Traditional methods may fail to distinguish between different lesion types (microaneurysms, haemorrhages, exudates) accurately.
- Misclassification can lead to incorrect treatment decisions, potentially endangering patients.

3.2.4 High Dependence on Manual Feature Engineering

1. Time-consuming and resource-intensive process:

- Feature extraction and manual classification require expert intervention, increasing processing time.
- Deep learning models can automate feature extraction, reducing the need for extensive manual tuning and speeding up diagnosis.

2. Limited real-time applicability:

- Many traditional methods are computationally expensive, making them impractical for real-time screening in clinical settings.
- AI-based models can be optimized for faster inference, enabling real-time DR detection.

3.2.5 Challenges in Clinical Deployment

1. **Lack of integration with hospital systems**:

- Most existing models do not integrate seamlessly with Electronic Health Records (EHR), making it difficult for clinicians to access results efficiently.
- AI-based approaches can be designed to integrate with hospital workflows, improving usability.

2. **Insufficient validation in diverse patient populations**:

- Many existing studies use limited datasets that do not represent the full spectrum of DR variations across different populations.
- This reduces the model's reliability in real-world clinical scenarios.

3.2.6 Need for an Advanced Deep Learning-Based Approach

To overcome the limitations of traditional DR detection systems, there is a need to adopt deep learning-based approaches that offer improved performance, scalability, and generalization.

3.3 Advantages of Deep Learning for DR Detection

1. **Automated Feature Learning**: CNNs can automatically learn features from raw images, eliminating the need for manual feature engineering.

2. **Multi-Class Classification**: Advanced models can classify DR into five stages, enabling better clinical decision-making.

3. **Improved Robustness**: Deep learning models generalize better across diverse datasets, improving real-world applicability.

4. **Real-Time Screening**: AI-powered models can process images in real time, making them suitable for large-scale screening programs.

5. **Integration with Telemedicine**: AI-based DR detection can be deployed in remote areas, enabling early diagnosis for underserved populations.

Existing DR detection systems, while effective in controlled environments, face significant challenges in real-world clinical applications. The reliance on traditional feature extraction methods, limited classification capabilities, and reduced generalization restrict their usefulness in large-scale DR screening.

To address these limitations, deep learning-based models such as fusion-based CNNs offer a promising alternative by improving accuracy, scalability, and real-time applicability. Future research should focus on developing robust AI models that integrate seamlessly with clinical workflows, ensuring better patient outcomes and more efficient DR management.

By transitioning to AI-driven approaches, the healthcare industry can significantly enhance early DR detection, reduce the burden on ophthalmologists, and prevent vision loss in diabetic patients worldwide.

CHAPTER-4

PROPOSED SYSTEM

4.1 Overview :

The proposed system aims to develop an automated Diabetic Retinopathy (DR) detection model using fusion-based Convolutional Neural Networks (CNNs) to enhance classification accuracy. By leveraging deep learning techniques, the system will classify fundus images into five DR categories: no DR, mild, moderate, severe, and proliferative DR. The system will integrate multiple models, including ResNet50, InceptionV3, and the DiaNet ensemble (comprising ResNet and InceptionV3) for feature extraction and analysis, along with the Xception model for final classification. The fusion of these models will help capture a wide range of features, improving the robustness and accuracy of the DR detection system. The system will be trained and evaluated on publicly available datasets, to ensure scalability and generalizability. This approach will enable timely detection, allowing for early intervention and more efficient resource allocation in healthcare systems, ultimately improving patient outcomes and reducing the burden on ophthalmologists.

4.1.1 Advantages of proposed system:

1. **Enhanced Classification Accuracy** – The fusion of multiple CNN models improves overall accuracy in detecting Diabetic Retinopathy (DR).
2. **Multi-Class Classification** – Unlike traditional binary classification, this system classifies DR into five severity levels, providing detailed diagnoses.
3. **Robust Feature Extraction** – Combining ResNet50, InceptionV3, and Xception captures a broader range of retinal features.
4. **Reduced False Positives and False Negatives** – The ensemble approach minimizes misclassifications, ensuring reliable diagnosis.
5. **Improved Generalization** – Training on diverse datasets ensures adaptability to different imaging conditions and patient demographics.
6. **Early Detection of DR** – The system detects early-stage DR, allowing timely intervention to prevent vision loss.
7. **Scalability for Large-Scale Screening** – The automated approach can process thousands of images efficiently, supporting mass screening programs.

8. **Reduction in Ophthalmologists' Workload** – Automating DR detection minimizes the burden on specialists, allowing them to focus on critical cases.
9. **Time-Efficient Diagnosis** – AI-driven analysis significantly reduces the time required for DR assessment compared to manual methods.
10. **Improved Resource Allocation in Healthcare** – Faster diagnosis helps prioritize patients needing urgent medical attention.
11. **Pretrained Model Utilization** – Leveraging pretrained models (ImageNet-trained CNNs) accelerates training and improves performance.
12. **Enhanced Contrast and Visibility** – Preprocessing techniques like CLAHE improve image clarity, aiding better feature extraction.
13. **Data Augmentation for Better Generalization** – Techniques like rotation, flipping, and zooming help the model adapt to various fundus image variations.
14. **Higher Sensitivity and Specificity** – The system effectively distinguishes between different DR stages, reducing misdiagnosis.
15. **Adaptability to Real-World Clinical Settings** – The model can be integrated into hospitals and telemedicine platforms for real-time diagnosis.
16. **Potential for Mobile and Cloud Integration** – The system can be deployed on cloud-based platforms, making it accessible via mobile or web applications.
17. **Supports Personalized Treatment Plans** – Detailed DR classification helps ophthalmologists tailor treatment strategies for each patient.
18. **Minimal Human Intervention Required** – The automated pipeline reduces dependency on manual grading, ensuring consistent diagnosis.
19. **Interoperability with Existing Healthcare Systems** – The model can be integrated with Electronic Health Records (EHR) for seamless data management.
20. **Future Scope for Further Optimization** – The architecture allows integration with advanced techniques like attention mechanisms and GAN-based image enhancement for even better performance.

4.2 SYSTEM ARCHITECTURE:

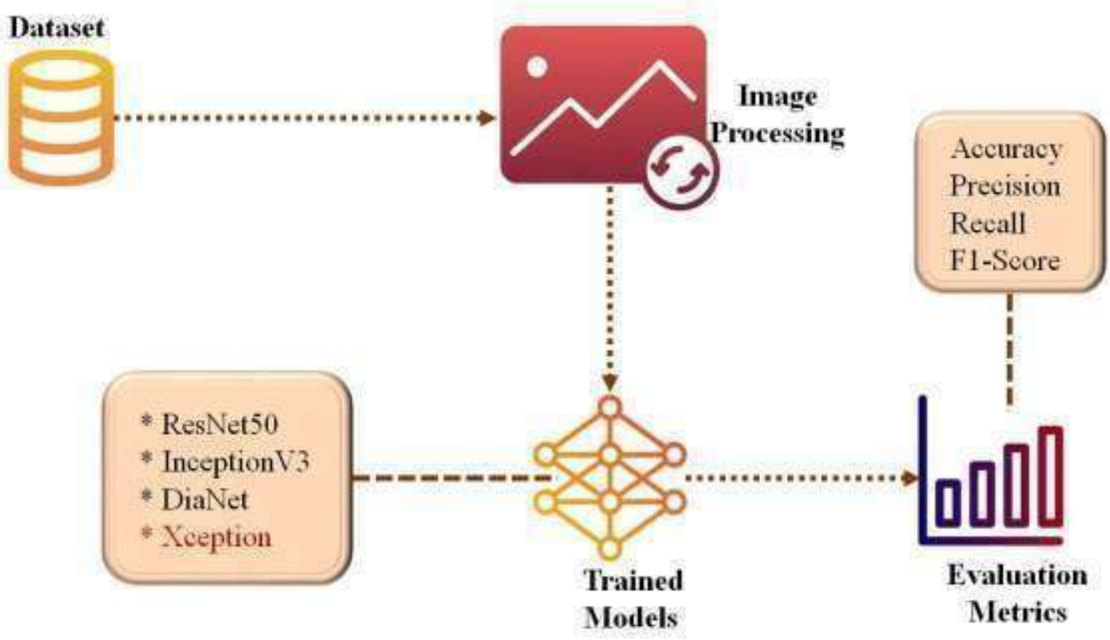


Fig.4.2.1 System architecture

The proposed system aims to develop an automated Diabetic Retinopathy (DR) detection model using fusion-based Convolutional Neural Networks (CNNs) to enhance classification accuracy. By leveraging deep learning techniques, the system will classify fundus images into five DR categories: no DR, mild, moderate, severe, and proliferative DR. The system will integrate multiple models, including ResNet50, InceptionV3, and the DiaNet ensemble (comprising ResNet and InceptionV3) for feature extraction and analysis, along with the Xception model for final classification. The fusion of these models will help capture a wide range of features, improving the robustness and accuracy of the DR detection system. The system will be trained and evaluated on publicly available datasets, to ensure scalability and generalizability. This approach will enable timely detection, allowing for early intervention and more efficient resource allocation in healthcare systems, ultimately improving patient outcomes and reducing the burden on ophthalmologists. This is diagram illustrates a machine learning workflow for image classification. A dataset is processed, then fed into trained models (ResNet50, InceptionV3, DiaNet, Xception). The models' performance is evaluated using metrics like accuracy, precision, recall, and F1-score. The process visualizes the flow from data input to model training and performance assessment.

4.2.1 Dataset Collection:

The dataset for diabetic retinopathy (DR) detection consists of retinal fundus images captured using high-resolution fundus cameras. These images represent different severity levels of DR, including No DR, Mild, Moderate, Severe, and Proliferative DR. The dataset includes diverse retinal images from multiple sources, ensuring variations in image quality, illumination, and patient demographics. Expert ophthalmologists annotate the images with ground truth labels to train and evaluate deep learning models for automated DR classification. The dataset covers a wide range of retinal abnormalities, such as microaneurysms, hemorrhages, and neovascularization, which are critical for accurate DR diagnosis. The inclusion of both normal and pathological images ensures a balanced representation for robust model training and evaluation.

4.2.2 Image Processing:

Image processing involves preparing retinal fundus images for efficient analysis and model train ing. Techniques such as re-scaling adjust pixel intensity values to normalize the dataset, improving model convergence. Shear transformation is applied to introduce slight distortions, enhancing the model's ability to generalize across variations. Zooming helps focus on specific retinal features, aiding in the detection of fine-grained abnormalities. Horizontal flipping augments the dataset by creating mirrored versions of images, reducing bias and improving robustness. Images are reshaped to a consistent dimension to ensure compatibility with deep learning architectures. These preprocessing steps enhance image quality, standardize input dimensions, and increase dataset diversity, contributing to improved performance in diabetic retinopathy classification.

4.2.3 Algorithms:

ResNet50 is a deep convolutional neural network designed to address vanishing gradient issues by using residual learning. It efficiently captures hierarchical image features, enabling accurate classification. The architecture's depth allows it to perform well in complex tasks by learning and fine-tuning relevant features from large image datasets.

InceptionV3 is a deep CNN that optimizes the use of computational resources by utilizing inception modules, which combine various filter sizes to capture multi-scale features. This model is particularly effective in image classification tasks, improving accuracy by identifying complex patterns and relationships within input data, especially in large datasets.

DiaNet is an ensemble model combining Res Net and InceptionV3 architectures to leverage the strengths of both networks. It enhances feature extraction by learning complex features from diverse image patterns and structures. The ensemble approach allows for robust analysis and improved classification accuracy by integrating different feature extraction techniques.

Xception is an advanced CNN based on depth wise separable convolutions, which improve efficiency by reducing the number of parameters while preserving model performance. It excels in extracting fine-grained features from images and is particularly effective in classification tasks that require high accuracy, such as complex image recognition.

4.2.4 Model Generation

Models used: ResNet50, InceptionV3, DiaNet (Ensemble of Res Net + InceptionV3), CNN, Xception. Feature extraction and analysis using deep learning techniques. Performance evaluation metrics (accuracy, precision, recall, F1-score) calculated for each model.

4.2.5 User Signup & Login

Secure user authentication for registration and login.

Ensures privacy and authorized access to the system.

4.2.6 User Input

Users upload retinal images for analysis. The system processes the input image and extracts relevant features.

4.2.7 Prediction

The trained model classifies the image based on disease severity. Final predicted result is displayed to the user. Helps in early detection, reducing workload for ophthalmologists.

CHAPTER -5

UML DIAGRAMS

5. UML DIAGRAMS

Unified Modeling Language (UML) is a standardized visual language used to model software systems. It encompasses a variety of diagram types, including use case, class, activity, sequence, and component diagrams. UML helps in designing, describing, and documenting system architecture and behavior, providing clarity and communication between developers, stakeholders, and end-users. In DR detection, UML can model the entire system's architecture, interactions, and processes, ensuring smooth integration and identifying potential improvements in system design. By representing workflows, class structures, and object interactions, UML helps stakeholders gain insights into how data flows and how components interact with each other. It is instrumental in ensuring system reliability, scalability, and performance, offering an organized approach to development and analysis.

Goals of UML

1. Clearly define system structure through visual representation of components, classes, and interactions for better understanding.
2. Provide a standardized language to facilitate communication between all stakeholders, ensuring alignment on system functionalities.
3. Identify and resolve potential issues early in the design process by visualizing interactions and data flow.
4. Model system behaviour to simulate scenarios and predict outcomes, enabling informed decision-making in system development.
5. Assist in system documentation, providing a reference for future modifications, updates, and maintenance tasks.
6. Improve system architecture by capturing relationships between components and ensuring appropriate scalability and flexibility.
7. Foster collaboration between developers and non-technical stakeholders, enhancing the effectiveness of the overall development process.

5.1 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system. It illustrates how input data is transformed through processes, stored in data repositories, and eventually produces outputs. DFDs use different levels of abstraction, with Level 0 showing a high-level overview and lower levels breaking down processes into more detailed steps.In the context of DR detection, a DFD will model how fundus images are captured, processed, and classified by the system. It will highlight the flow of image data from the input (e.g., a camera or database) through various stages of pre-processing, feature extraction, classification, and output generation. It helps visualize the interactions between users, processes, data repositories, and external systems.The primary purpose of a DFD in this context is to help design and understand the flow of data in the DR detection system.

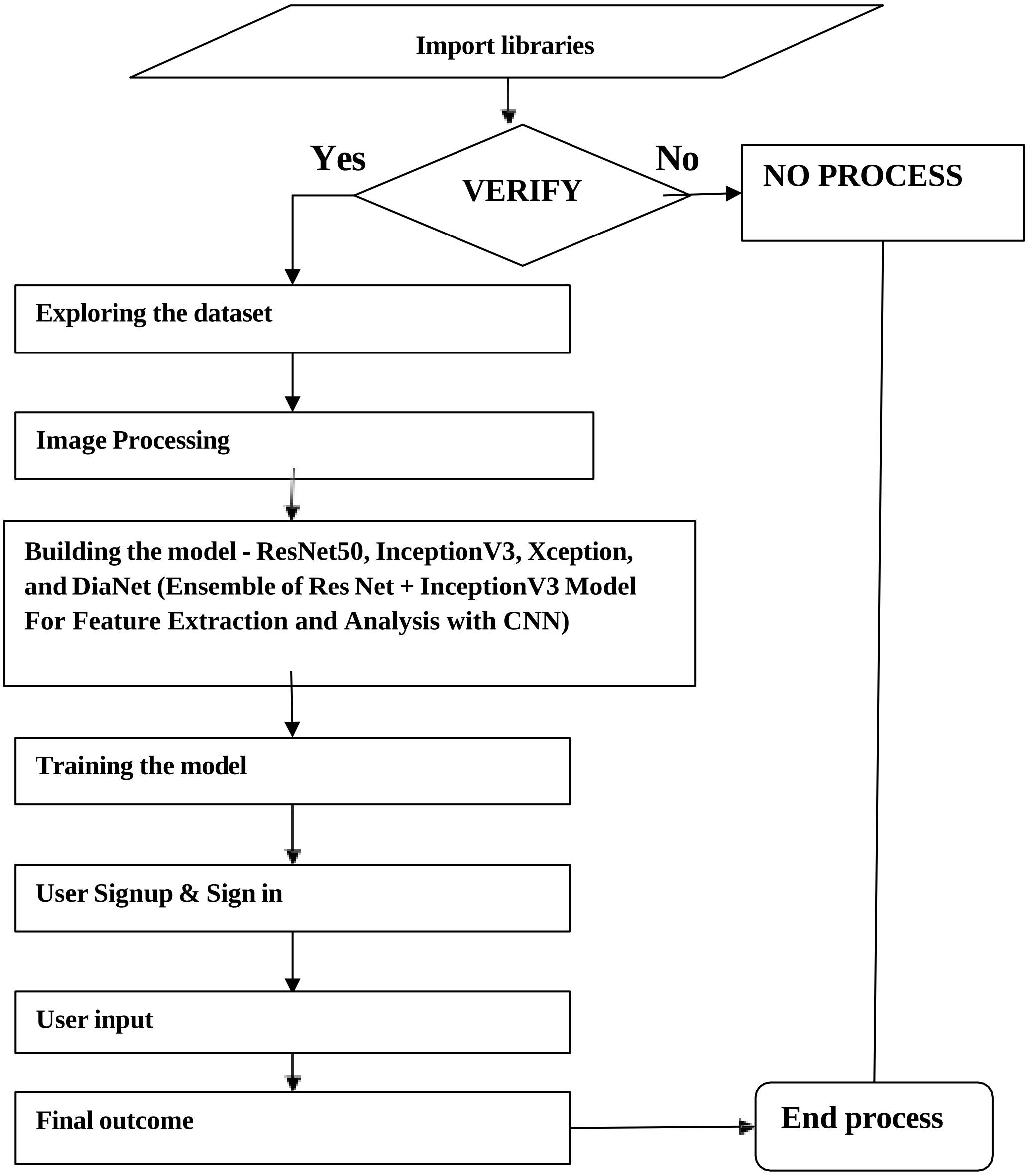


Fig 5.1 Data Flow Diagram

5.2 Use Case Diagram

A use case diagram visualizes interactions between users (or actors) and the system, highlighting key functionalities. It helps in identifying system requirements, user expectations, and system boundaries, ensuring that all stakeholder needs are captured and addressed during the development process.

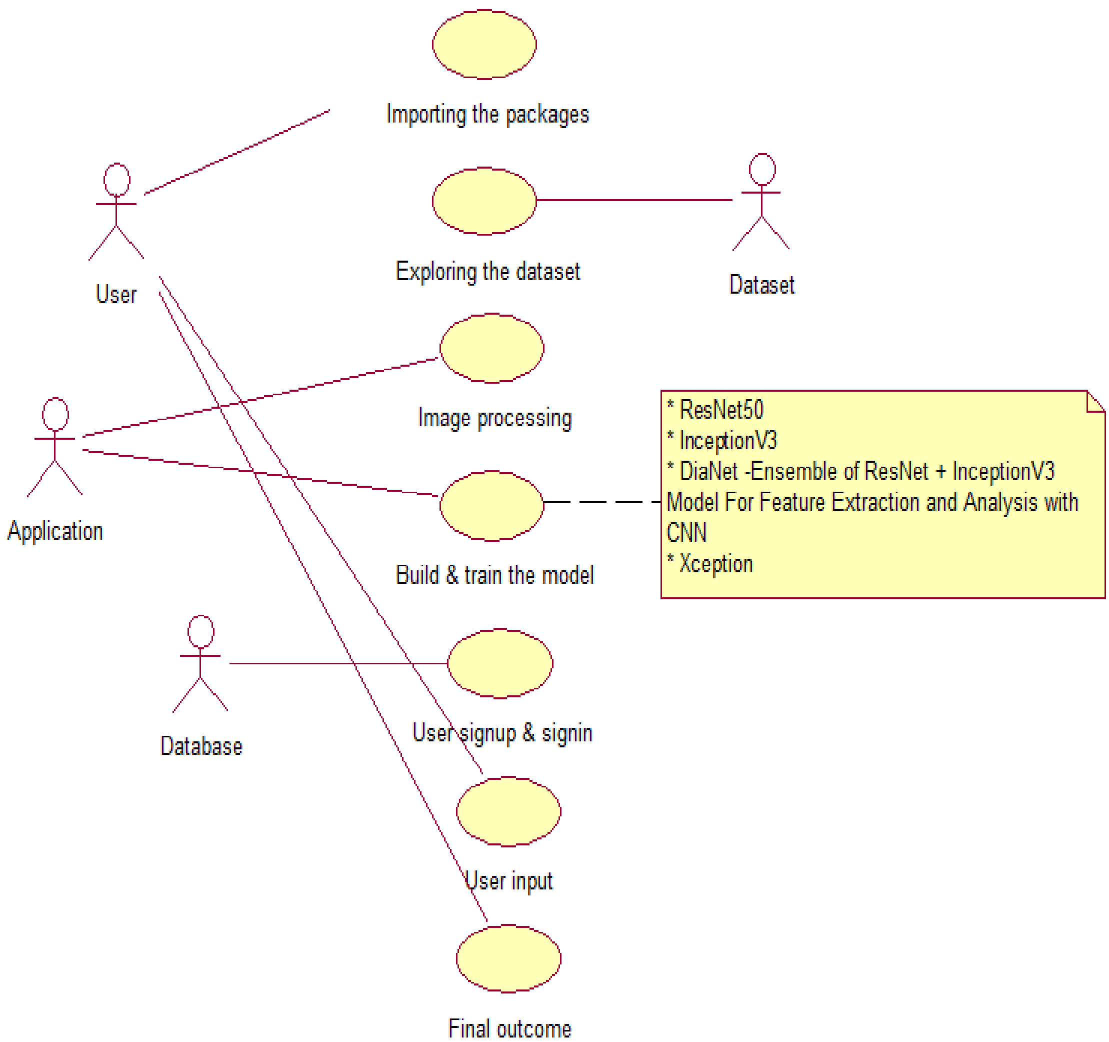


Fig 5.2 Use Case Diagram

5.3 Class Diagram

A class diagram outlines the system's classes, their attributes, methods, and relationships. It provides a clear structure of the system, helping to model the static structure and define how components interact and are organized within the system, which is crucial for coding and maintenance.

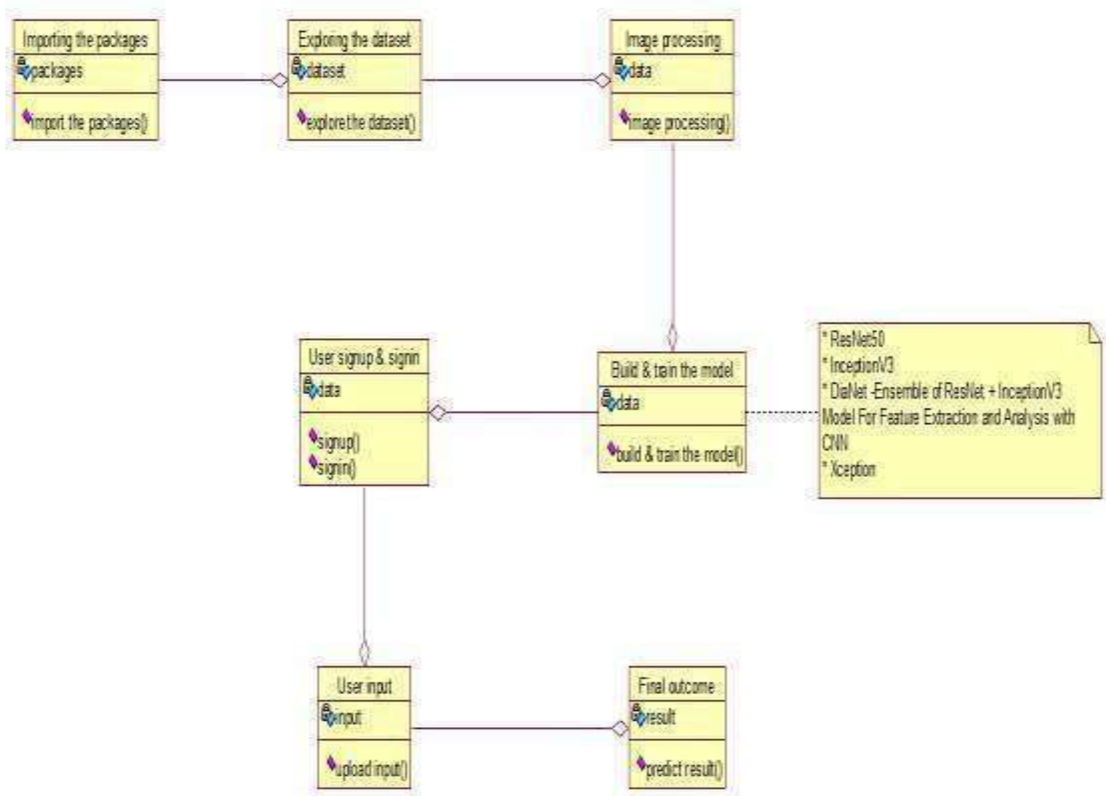


Fig 5.3 Class Diagram

5.4 Activity Diagram

An activity diagram represents the workflow of a system, illustrating the sequence of operations, decisions, and events. It helps in understanding the system's dynamic behavior and the steps involved in achieving specific tasks, ensuring smooth execution and proper task management.

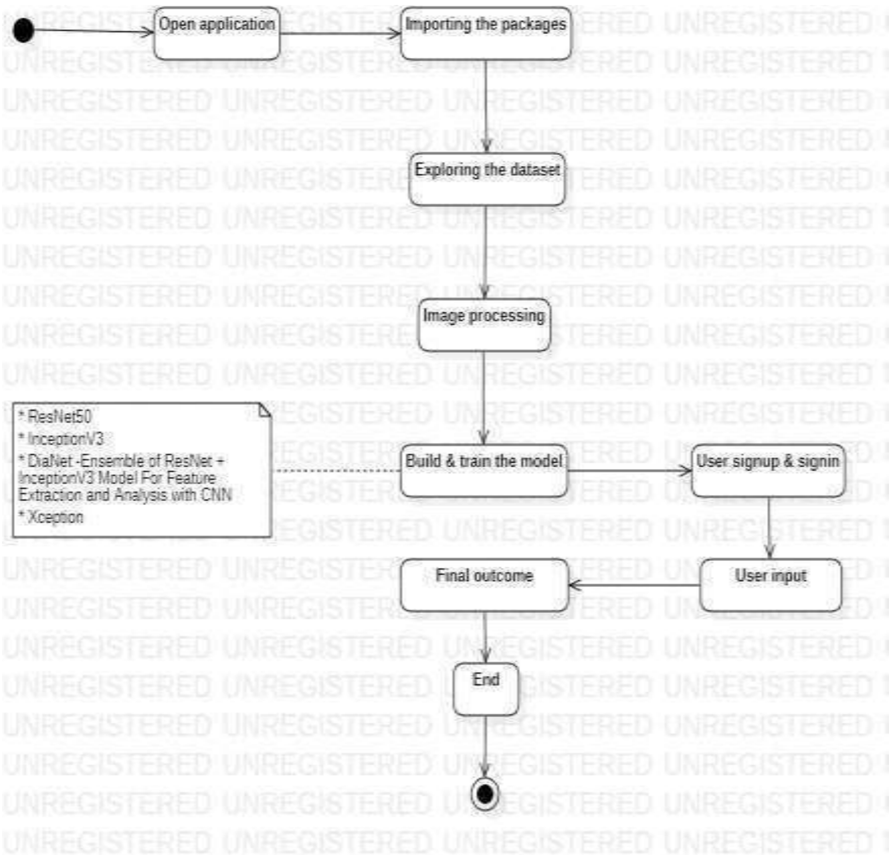


Fig 5.4 Activity Diagram

5.5 Sequence Diagram

A sequence diagram models the interaction between objects in a specific sequence, showing how data flows over time. It is useful for visualizing the process flow, communication between system components, and identifying potential bottlenecks or inefficiencies in the execution of tasks.

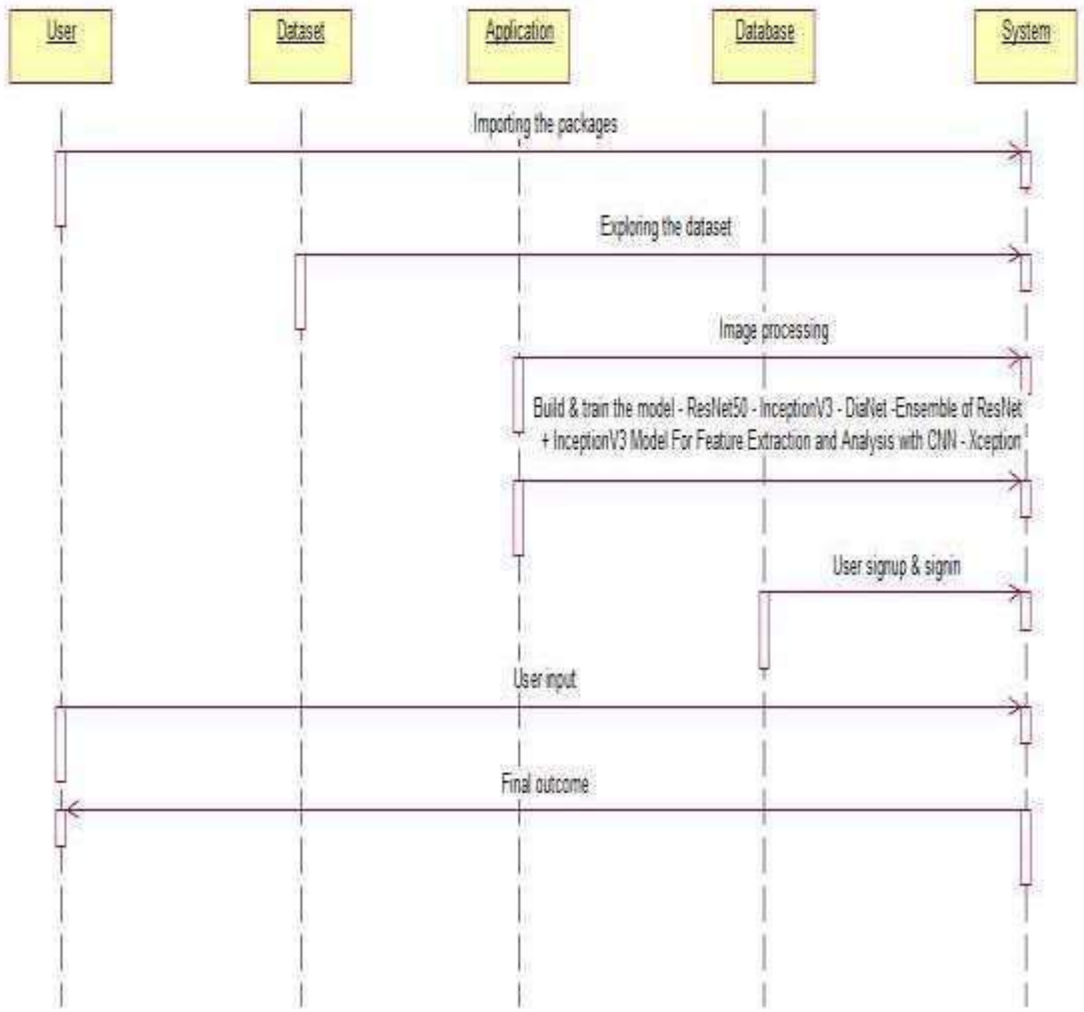


Fig 5.5 Sequence Diagram

5.6 Collaboration Diagram

A collaboration diagram highlights the relationships and interactions between system components or objects. It focuses on how different parts of the system cooperate to achieve tasks, assisting in optimizing interactions and ensuring that the system components work effectively together.

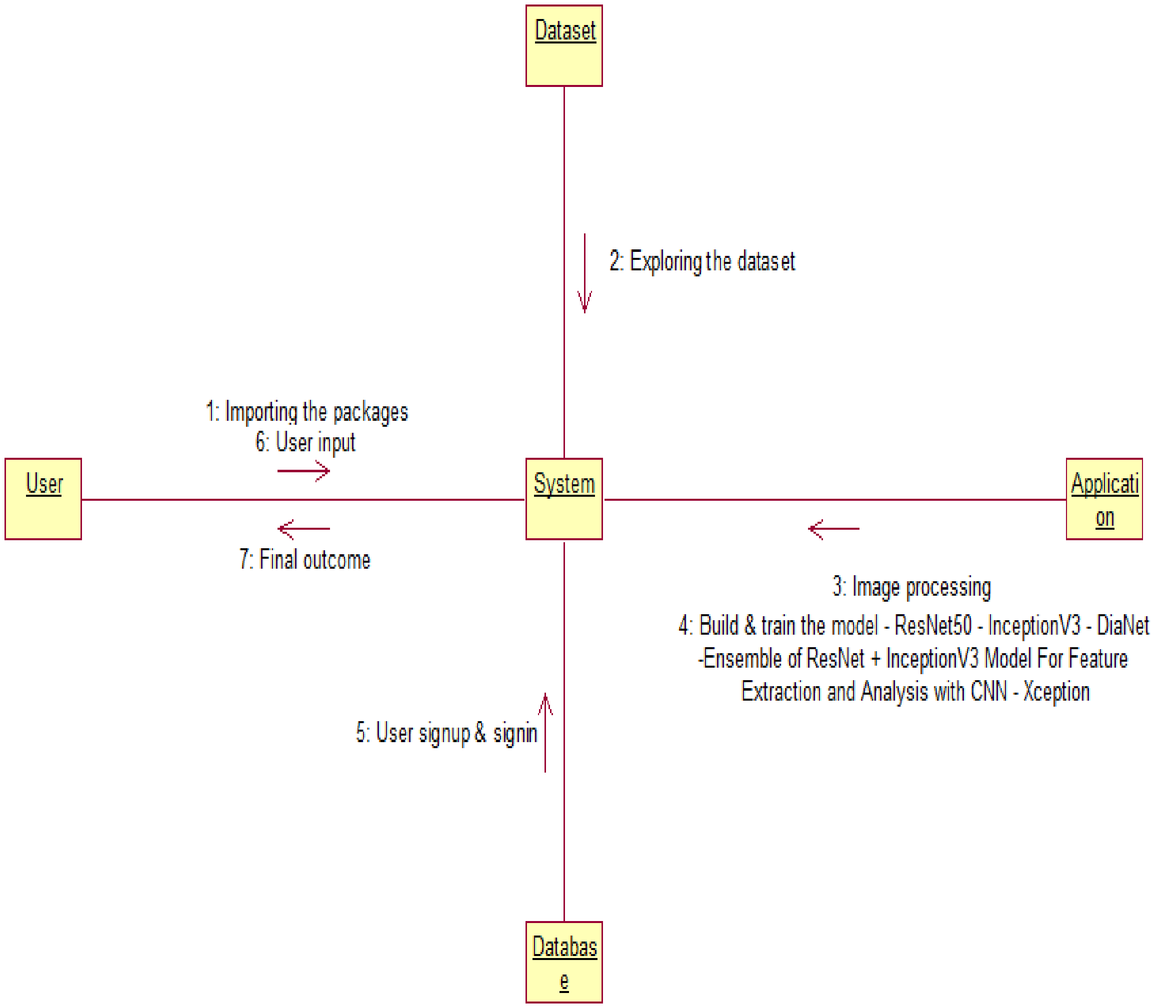


Fig 5.6 Collaboration Diagram

5.7 Component Diagram

A component diagram represents the physical components of the system and their dependencies. It helps in modeling the system's modular structure, ensuring efficient separation of concerns, and providing insight into how components interact, facilitating system design, scalability, and maintainability.

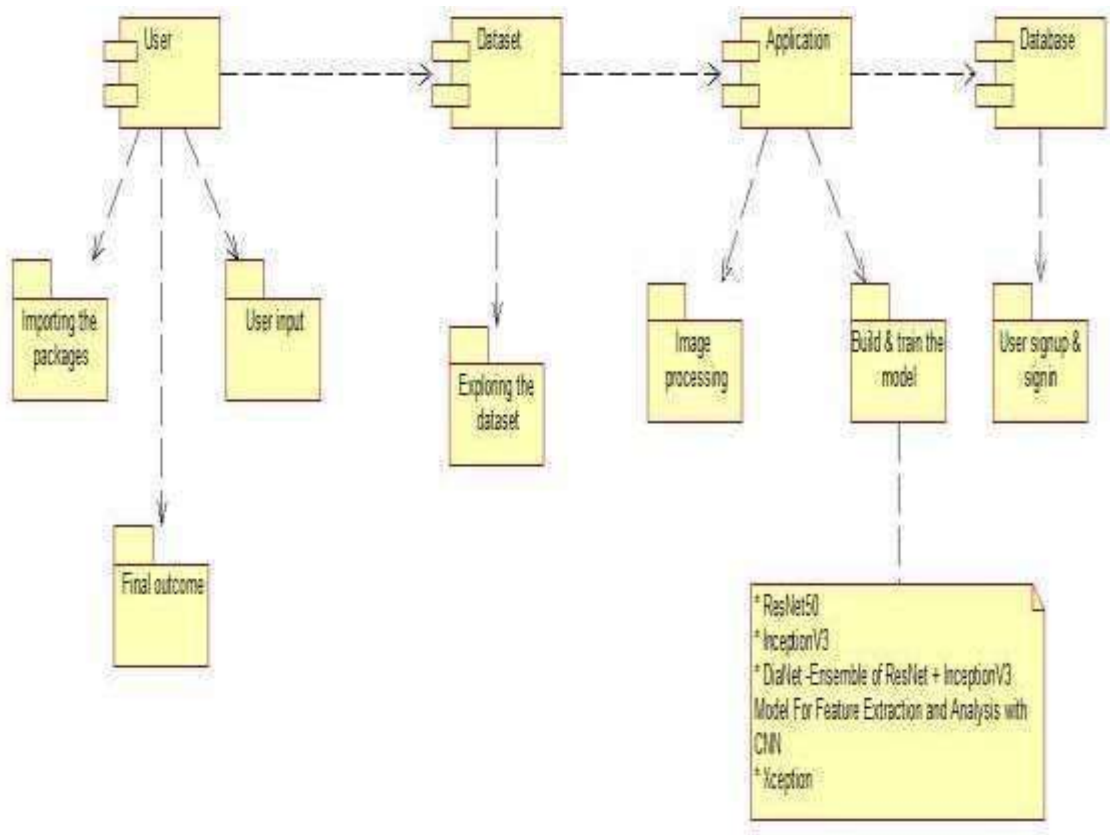


Fig 5.7 Component Diagram

5.8 Deployment Diagram

A deployment diagram illustrates the physical deployment of software components on hardware nodes. It helps in visualizing the architecture of the system's hardware and software environment, ensuring proper resource allocation, optimizing performance, and identifying potential infrastructure requirements or constraints.

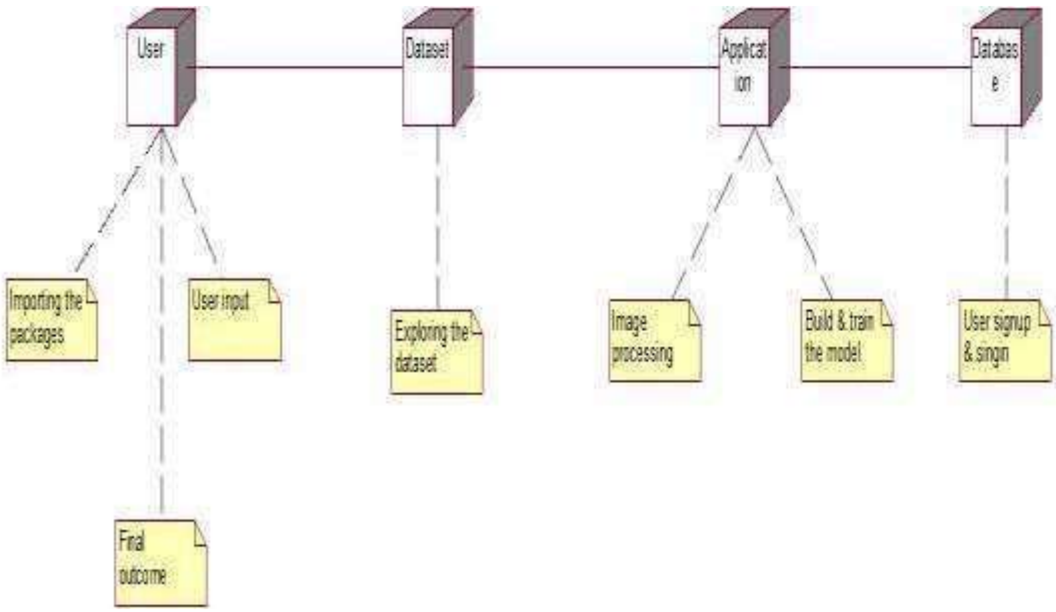


Fig 5.8 Deployment Diagram

CHAPTER-6

SOFTWARE ENVIRONMENT

6.1 Deep Learning:

Deep learning is the branch of machine learning that is based on artificial neural network architecture. An artificial neural network or ANN uses layers of interconnected nodes called neurons that work together to process and learn from the input data.

In a fully connected Deep Neural Network, there is an input layer and one or more hidden layers connected one after the other. Each neuron receives input from the previous layer neurons or the input layer. The output of one neuron becomes the input to other neurons in the next layer of the network, and this process continues until the final layer produces the output of the network. The layers of the neural network transform the input data through a series of nonlinear transformations, allowing the network to learn complex representations of the input data.

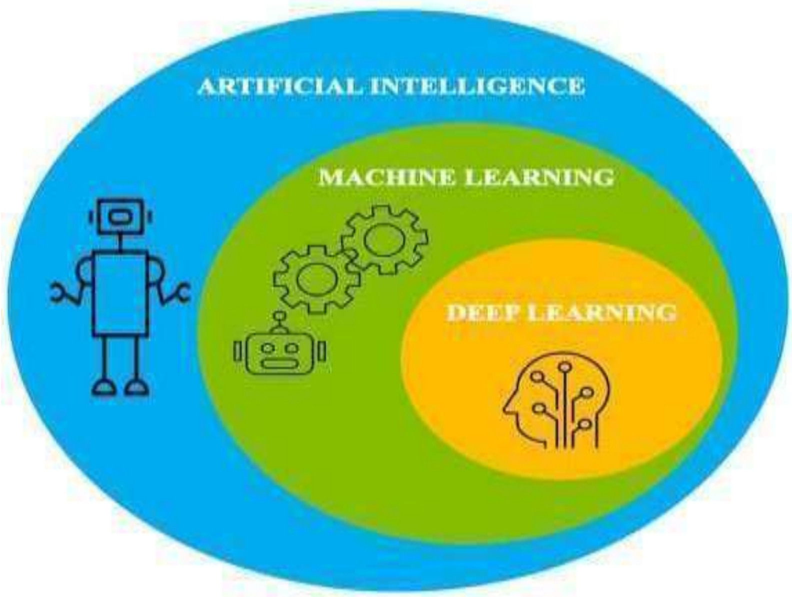


Fig 6.1 Deep Leaning

6.2 Artificial neural networks

Artificial neural networks are built on the principles of the structure and operation of human neurons. It is also known as neural networks or neural nets. An artificial neural network's input layer, which is the first layer, receives input from external sources and passes it on to the hidden layer, which is the second layer. Each neuron in the hidden layer gets information from the neurons in the previous layer, computes the weighted total, and then transfers it to the neurons in the next layer. These connections are weighted, which means that the impacts of the inputs from the preceding layer are more or less optimized by giving each input a distinct weight. These weights are then adjusted during the training process to enhance the performance of the model.

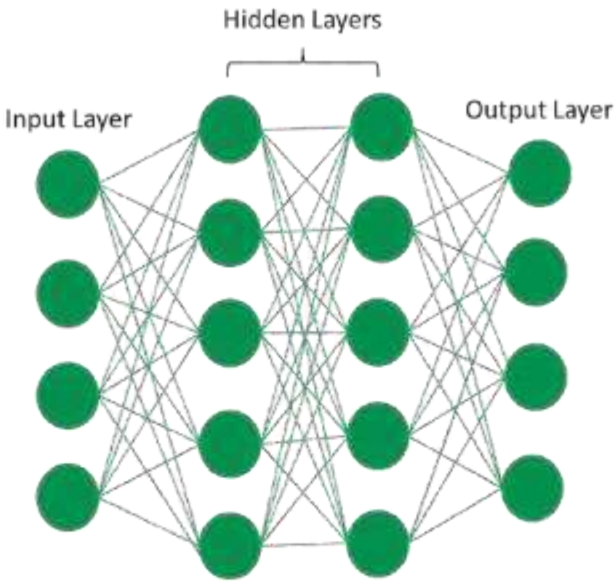


Fig 6.2 Artificial neural networks

Artificial neurons, also known as units, are found in artificial neural networks. The whole Artificial Neural Network is composed of these artificial neurons, which are arranged in a series of layers.

6.2.1 Types of neural networks

Deep Learning models are able to automatically learn features from the data, which makes them well-suited for tasks such as image recognition, speech recognition, and natural language processing. The most widely used architectures in deep learning are feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs).

- Feedforward neural networks (FNNs) are the simplest type of ANN, with a linear flow of information through the network. FNNs have been widely used for tasks such as image classification, speech recognition, and natural language processing.
- Convolutional Neural Networks (CNNs) are specifically for image and video recognition tasks. CNNs are able to automatically learn features from the images, which makes them well-suited for tasks such as image classification, object detection, and image segmentation.
- Recurrent Neural Networks (RNNs) are a type of neural network that is able to process sequential data, such as time series and natural language. RNNs are able to maintain an internal state that captures information about the previous inputs, which makes them well-suited for tasks such as speech recognition, natural language processing, and language translation.

6.2.2 Advantages of Deep Learning:

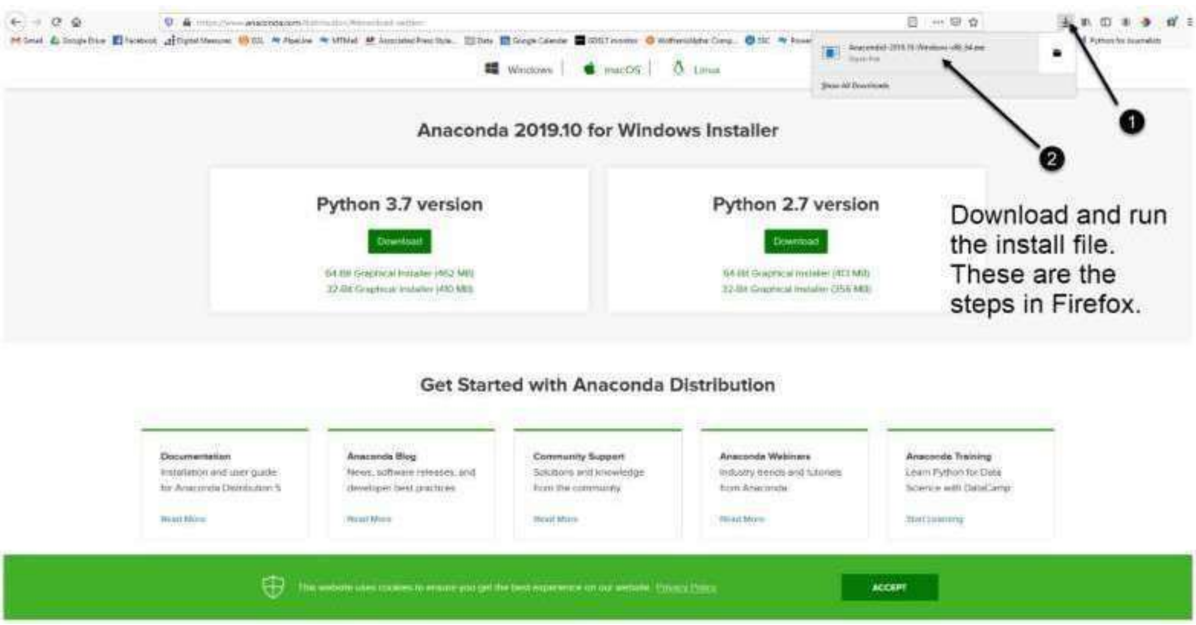
1. High accuracy: Deep Learning algorithms can achieve state-of-the-art performance in various tasks, such as image recognition and natural language processing.
2. Automated feature engineering: Deep Learning algorithms can automatically discover and learn relevant features from data without the need for manual feature engineering.
3. Scalability: Deep Learning models can scale to handle large and complex datasets, and can learn from massive amounts of data.
4. Flexibility: Deep Learning models can be applied to a wide range of tasks and can handle various types of data, such as images, text, and speech.
5. Continual improvement: Deep Learning models can continually improve their performance as more data becomes available.

6.2.3 Disadvantages of Deep Learning:

1. High computational requirements: Deep Learning AI models require large amounts of data and computational resources to train and optimize.
2. Requires large amounts of labeled data: Deep Learning models often require a large amount of labeled data for training, which can be expensive and time- consuming to acquire.
3. Interpretability: Deep Learning models can be challenging to interpret, making it difficult to understand how they make decisions.
4. Overfitting: Deep Learning models can sometimes overfit to the training data, resulting in poor performance on new and unseen data.

6.3 Python Anaconda Installation

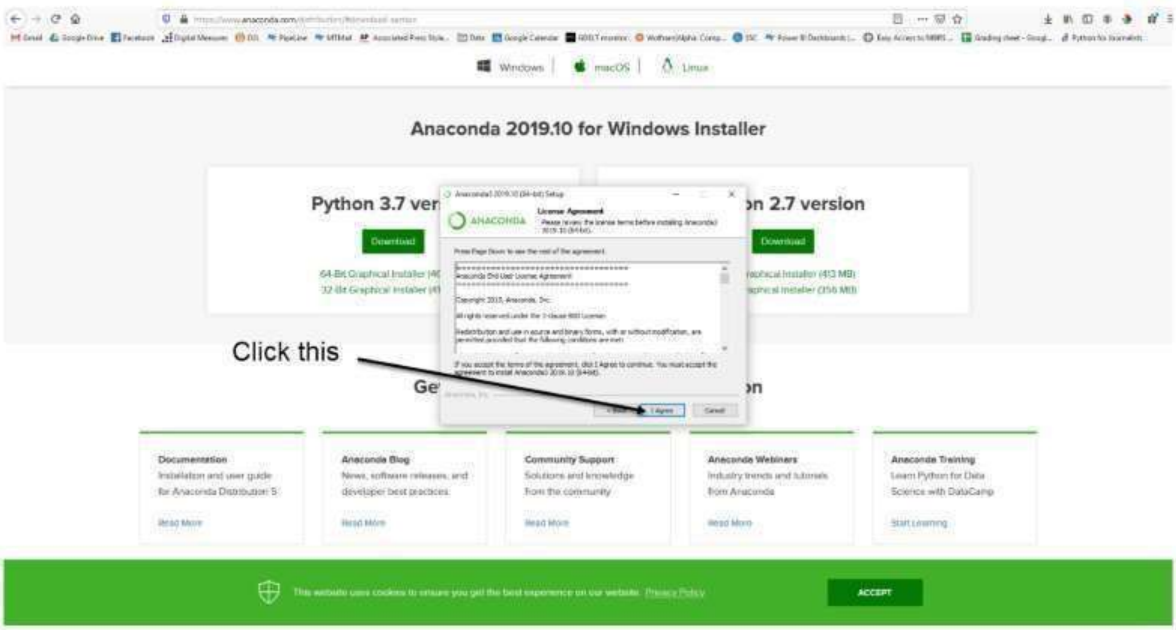
1. Go to this link and download Anaconda for Windows, Mac, or Linux: – Download anaconda



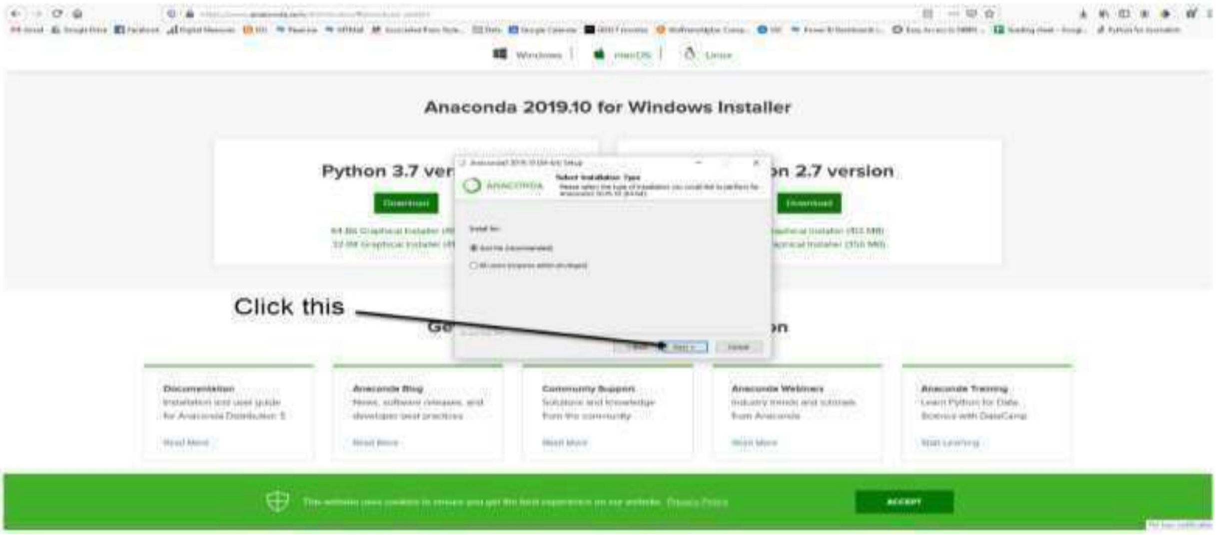
You can download the installer for Python 3.7 or for Python 2.7 (at the time of writing). And you can download it for a 32-bit or 64-bit machine.

2. Click on the downloaded .exe to open it. This is the Anaconda setup. Click next.

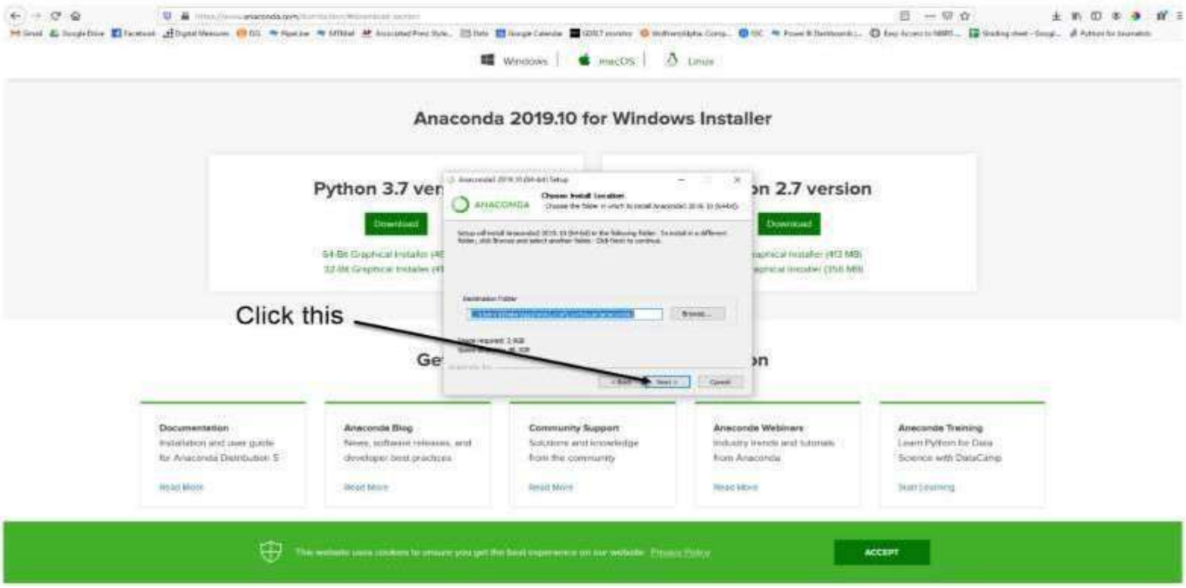
3. Now, you’ll see the license agreement. Click on ‘I Agree’.



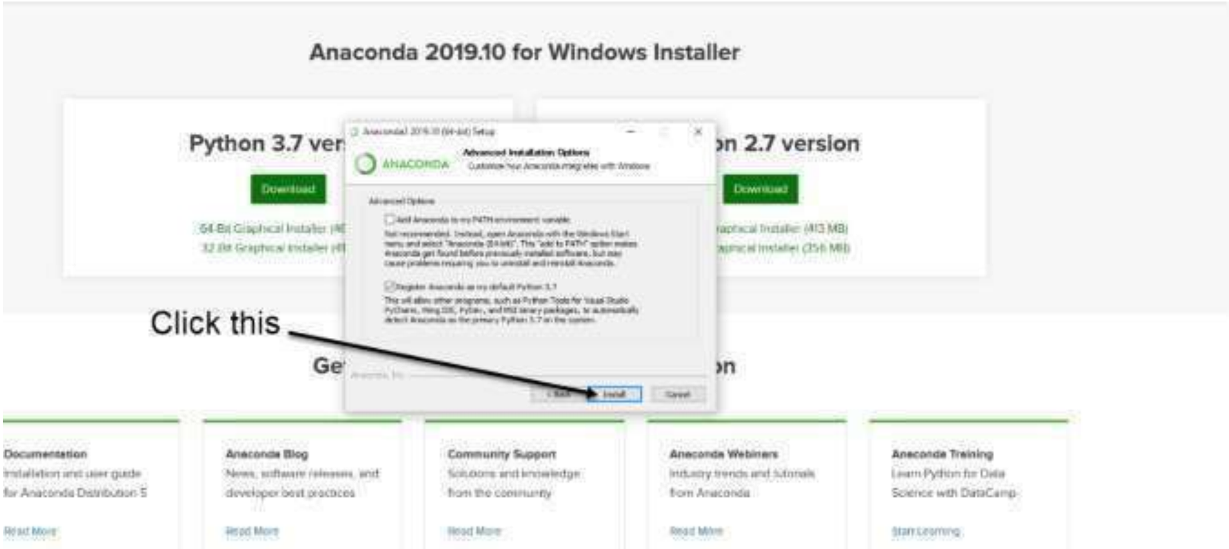
4. You can install it for all users or just for yourself. If you want to install it for all users, you need administrator privileges.



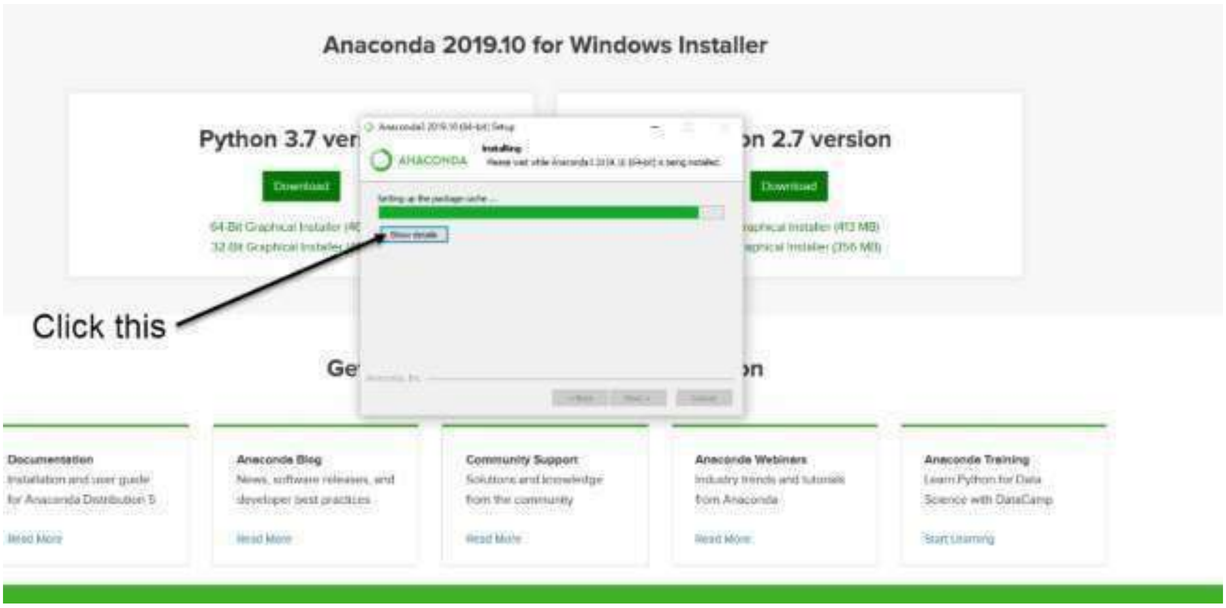
5. Choose where you want to install it. Here, you can see the available space and how much you need.



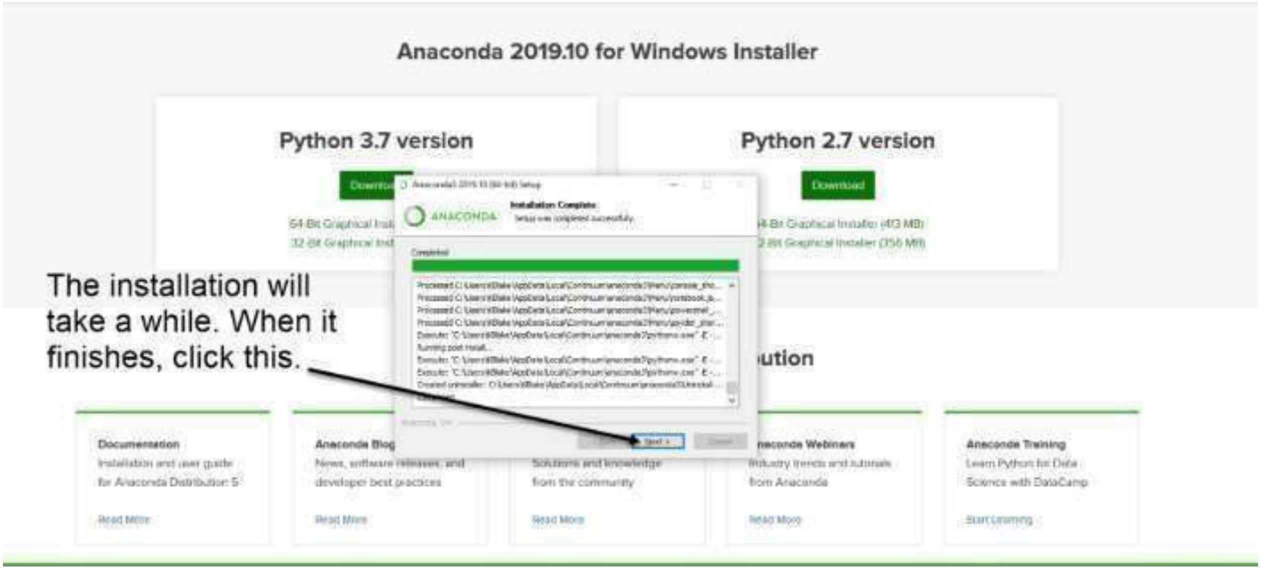
6. Now, you'll get some advanced options. You can add Anaconda to your system's PATH environment variable, and register it as the primary system Python 3.7. If you add it to PATH, it will be found before any other installation. Click on 'Install'.



7. It will unpack some packages and extract some files on your machine. This will take a few minutes.



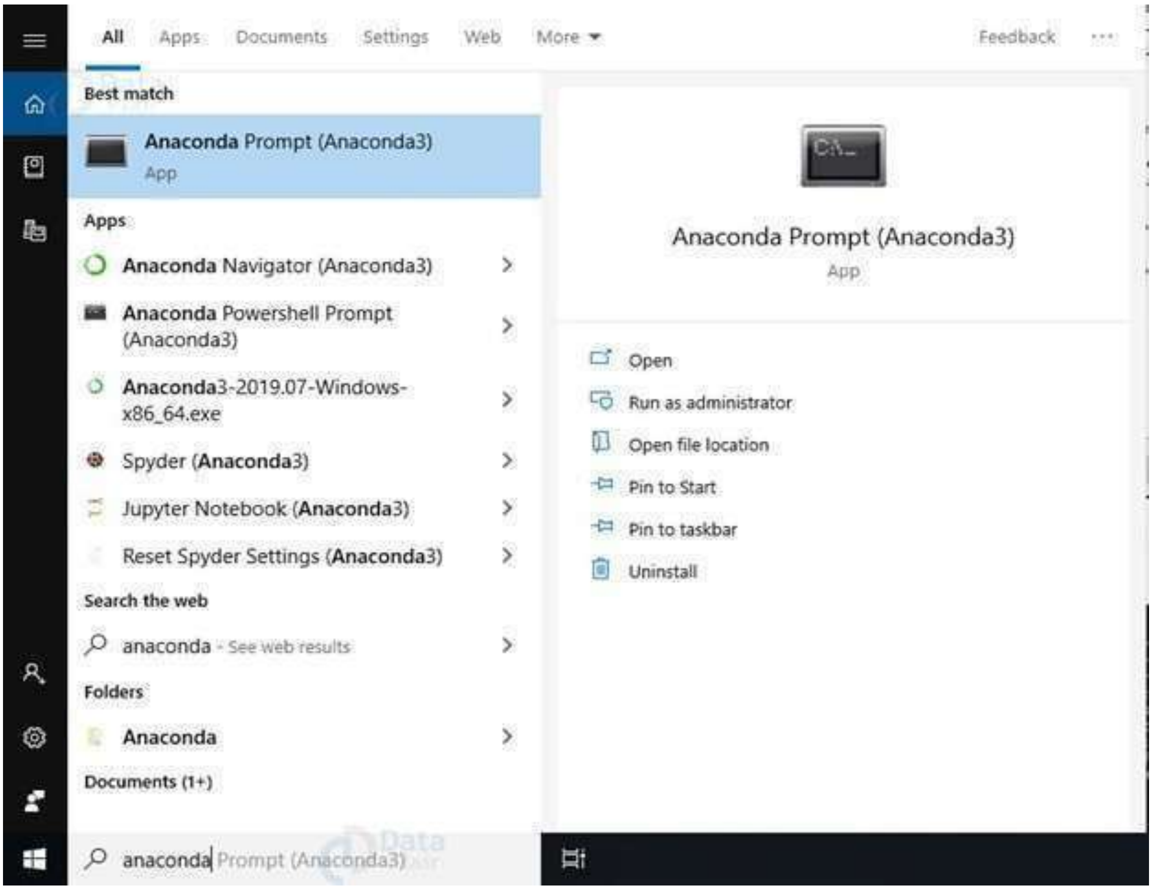
8. The installation is complete. Click Next.

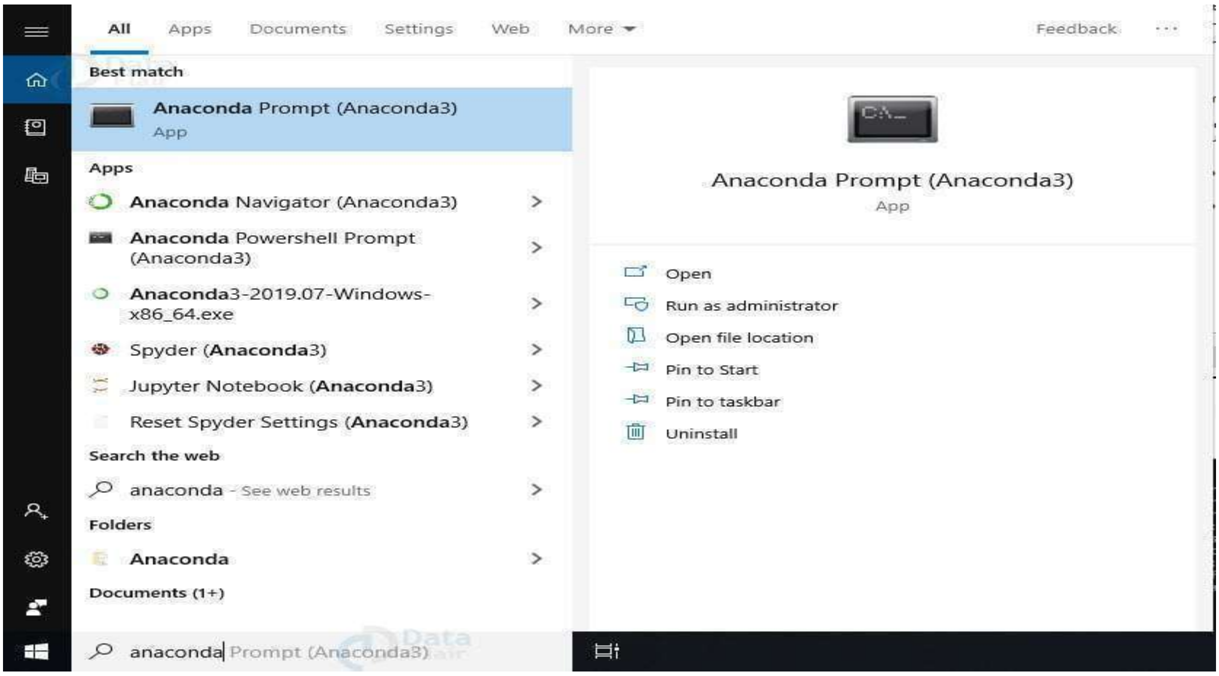


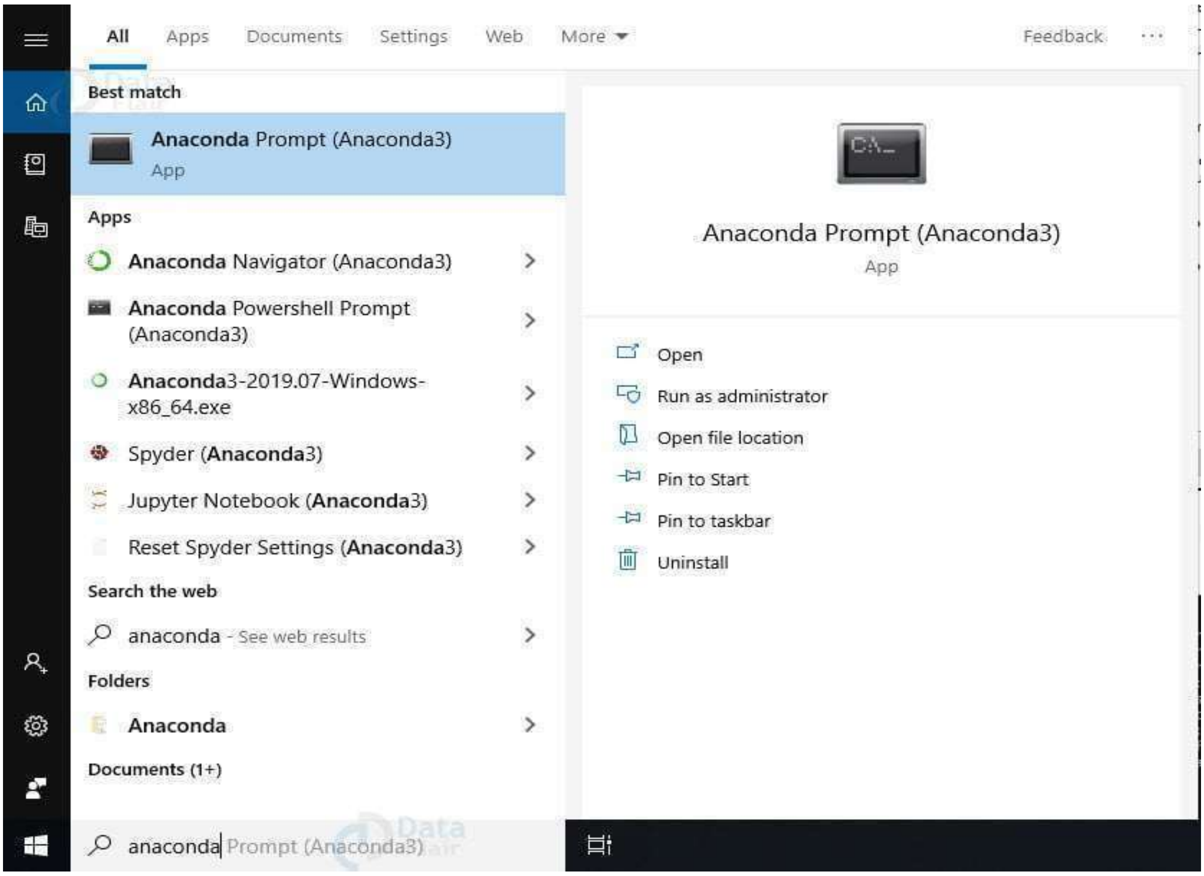
9. This screen will inform you about PyCharm. Click Next.

10. The installation is complete. You can choose to get more information about Anaconda cloud and how to get started with Anaconda. Click Finish.

11. If you search for Anaconda now, you will see the following options:




All
Apps
Documents
Settings
Web
More
Feedback
Best match
Anaconda Prompt (Anaconda3)
App
Apps
Anaconda Navigator (Anaconda3)
Anaconda Powershell Prompt (Anaconda3)
Anaconda3-2019.07-Windows-x86_64.exe
Spyder (Anaconda3)
Jupyter Notebook (Anaconda3)
Reset Spyder Settings (Anaconda3)
Search the web
anaconda - See web results
Folders
Anaconda
Documents (1+)
anaconda Prompt (Anaconda3)
Anaconda Prompt (Anaconda3)
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall


All
Apps
Documents
Settings
Web
More
Feedback
Best match
Anaconda Prompt (Anaconda3)
App
Apps
Anaconda Navigator (Anaconda3)
Anaconda Powershell Prompt (Anaconda3)
Anaconda3-2019.07-Windows-x86_64.exe
Spyder (Anaconda3)
Jupyter Notebook (Anaconda3)
Reset Spyder Settings (Anaconda3)
Search the web
anaconda - See web results
Folders
Anaconda
Documents (1+)
anaconda Prompt (Anaconda3)
Anaconda Prompt (Anaconda3)
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall

6.3.1 LIBRARIES/ PACKAGES :-

TensorFlow:

TensorFlow is an open-source library developed by Google Brain Team for high-level numerical computations. It is widely used in machine learning (ML) and deep learning (DL) applications, including image recognition, natural language processing, and AI-driven automation. TensorFlow operates on tensors, which are multi-dimensional arrays that facilitate efficient mathematical computations. One of its key advantages is its graph-based computation, which optimizes execution by structuring data flow as a directed graph. TensorFlow provides scalability by supporting both CPU and GPU processing, significantly improving the performance of complex models. It also includes TensorFlow Extended (TFX) for deploying ML models in production environments

Matplotlib:

This library is responsible for plotting numerical data. And that’s why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc. Matplotlib supports multiple chart types, including line graphs, scatter plots, bar charts, histograms, and pie charts. Its extensive customization options allow users to modify colours, labels, and figure sizes. One of its key advantages is its ability to integrate seamlessly with NumPy and Pandas, making it ideal for analysing large datasets. Researchers and data analysts use Matplotlib to create publication-quality visualizations that help interpret complex data trends.

Pandas:

Pandas are an important library for data scientists. It is an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools. It eases data analysis, data manipulation, and cleaning of data. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc.

Numpy:

The name “Numpy” stands for “Numerical Python”. It is the commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use Numpy internally to perform several operations on tensors. Array Interface is one of the key features of this library. One of its core features is vectorization, which allows operations on entire arrays without using explicit loops, making computations significantly faster. NumPy also offers advanced linear algebra functions, Fourier transformations, and random number generation, making it indispensable for numerical analysis

SciPy:

The name “SciPy” stands for “Scientific Python”. It is an open-source library used for high-level scientific computations. This library is built over an extension of Numpy. It works with Numpy to handle complex computations. While Numpy allows sorting and indexing of array data, the numerical data code is stored in SciPy. It is also widely used by application developers and engineers. Due to its high performance and efficiency, SciPy is widely used in developing applications requiring mathematical modeling and data processing.

Scikit-learn:

It is a famous Python library to work with complex data. Scikit-learn is an open-source library that supports machine learning. It supports variously supervised and unsupervised algorithms like linear regression, classification, clustering, etc. This library works in association with Numpy and SciPy. It also includes tools for model selection, feature extraction, and hyperparameter tuning, making it essential for building high-performance ML models. Due to its ease of use and efficiency, Scikit-learn is widely adopted in academia and industry for developing machine learning applications.

CHAPTER 7

SYSTEM REQUIREMENTS SPECIFICATIONS

7.1 SOFTWARE REQUIREMENTS

1) Software: Anaconda

2) Primary Language: Python

3) Frontend Framework: Flask

4) Back-end Framework: Jupyter Notebook

5) Database: Sqlite3

6) Front-End Technologies: HTML, CSS, JavaScript and Bootstrap4

7.2 HARDWARE REQUIREMENTS

1) Operating System: Windows Only

2) Processor: i5 and above

3) Ram: 8GB and above

4) Hard Disk: 25 GB in local drive

7.3 FUNCTIONAL REQUIREMENTS

1. Data Collection – Gather relevant data from various sources, ensuring accuracy and completeness.

2. Data Processing – Clean, transform, and organize raw data for further analysis.

3. Training and Testing – Split the dataset to train models and evaluate their performance.

4. Modelling – Apply machine learning or statistical techniques to identify patterns and relationships.

5. Predicting – Use trained models to generate insights or forecasts based on new data.

7.4 NON- FUNCTIONAL REQUIREMENTS

Performance: The system must process fundus images efficiently, providing real-time or near-real-time classification without significant delays. This is crucial for enabling timely diagnosis and intervention, especially in high-volume clinical settings.

Scalability: The system should be capable of handling an increasing number of fundus images without a decrease in performance. As the number of users or images grows, the system must maintain its ability to classify new data accurately and efficiently.

Usability: The user interface should be intuitive and easy to navigate for medical professionals with varying levels of technical expertise. Clear visualizations and straightforward interaction should allow ophthalmologists to interpret results quickly and effectively.

Reliability: The system should operate consistently with minimal downtime. It must be stable in production environments and handle unexpected errors or failures gracefully without compromising the accuracy of DR classification.

Security: Given the sensitive nature of healthcare data, the system must implement strict security measures, such as encryption and access control, to ensure patient data privacy and comply with relevant healthcare regulations (e.g., HIPAA).

Maintainability: The system should be easy to maintain, with modular architecture that allows for regular updates, bug fixes, and integration of new features. Proper documentation should be provided to assist in troubleshooting and future upgrades.

7.5. FEASIBILITY STUDY

Feasibility study examines the viability or sustainability of an idea, project, or business. The study examines whether there are enough resources to implement it, and the concept has the potential to generate reasonable profits. In addition, it will demonstrate the benefits received in return for taking the risk of investing in the idea.

Types of Feasibility Study

There are several different kinds of feasibility studies. Understanding the types of feasibility studies and the technicalities of the concept is important for any business. They are elaborated below:

Technical Feasibility

Technical feasibility study checks for accessibility of technical resources in the organization. In case technological resources exist, the study team will conduct assessments to check whether the technical team can customize or update the existing technology to suit the new method of workings for the project by properly checking the health of the hardware and software. Many factors need to be taken into consideration here, like staffing requirements, transportation, and technological competency.

Financial Feasibility

Financial feasibility allows an organization to determine cost-benefit analysis. It gives details about the investment that has to go in to get the desired level of benefit (profit). Factors such as total cost and expenses are considered to arrive simultaneously. With this data, the companies know their present state of financial affairs and anticipate future monetary requirements and the sources from which the company can acquire them. Investors can largely benefit from the economic analysis done. Assessing the return on investment of a particular asset or acquisition can be a financial feasibility study example.

Market Feasibility

Market feasibility assesses the viability of a product or service within a specific industry by analysing market demand, competition, and growth potential. It examines industry trends, consumer behaviour, and existing marketing strategies while identifying areas for improvement. Understanding the competitive landscape helps businesses position themselves effectively. A crucial aspect is forecasting sales projections based on market data, historical trends, and potential customer base. This study ensures that a business idea aligns with market needs and has the potential for profitability. By evaluating risks and opportunities, companies can refine their strategies, optimize resource allocation, and enhance their market entry success.

Organization Feasibility

Organizational feasibility evaluates whether a business has the necessary structure, legal framework, and management expertise to successfully implement its idea. It examines the company’s leadership, workforce capabilities, and operational efficiency. Key aspects include legal compliance, organizational hierarchy, and resource availability. A strong management team with relevant experience enhances decision-making and business sustainability. This study also identifies potential gaps in governance, staffing, or legal requirements that may hinder operations. By assessing internal strengths and weaknesses, businesses can make informed adjustments to ensure smooth execution, mitigate risks, and improve overall operational effectiveness for long-term success.

CHAPTER-8

SYSTEM TESTING

System testing, also referred to as system-level testing or system integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application.

System testing verifies that an application performs tasks as designed. It's a type of black box testing that focuses on the functionality of an application rather than the inner workings of a system, which white box testing is concerned with.

8.1 Types of system testing

With system testing, a QA team gauges if an application meets all of its technical, business and functional requirements. To accomplish this, the QA team might utilize various types of software testing techniques that determine the overall test coverage for an application and help catch critical defects that hamper an application's core functionalities before release.

The following are the common types of system testing techniques:

Performance testing: Performance testing measures the speed, average load time, stability, reliability and peak response times of the system under various conditions. It's typically coupled with stress testing and may include both hardware and software testing tools.

Usability testing: These are tests to evaluate if a system is easy to use and functional for the end user. Metrics, including user error rates, task success rates, the time it takes a user to complete a task and user satisfaction, are used during testing.

Load testing: This is testing to determine how a system or software performs under a real-life extreme load and test scenarios. Metrics, such as throughput, number of users and latency, are measured through this testing.

Regression testing: Also known as sanity testing, it ensures that all changes introduced into an application or code during system testing, recent code changes or updates haven't caused any new bugs or issues. Regression testing is responsible for the functionality of the existing features of a system or software.

Migration testing: This is conducted to ensure smooth migration of legacy systems to new systems without disruptions, data loss or downtimes.

Scalability testing: This measures an application's or system's capability to scale up or down when trying to meet the changing user requirements.

Functionality testing: This is conducted to validate a system's functionality against its functional and business requirements.

Recovery testing: This is a type of nonfunctional testing done to ensure that a system is capable of recovering from certain system errors, crashes and failures.

Fig 8.1 System Testing

Phases of system testing

System testing examines every component of an application to make sure that they work as a complete and unified whole. A QA team typically conducts system testing after it checks individual modules with functional or user story testing and then each component through integration testing.

If a software build achieves the desired results in system testing, it gets a final check via acceptance testing before it goes to production, where users consume the software. An app dev team logs all defects and establishes what kinds and numbers of defects are tolerable.

Typically, system testing goes through the following stages:

Test environment: In this initial stage, a test server is set up for creating a testing environment, which enables a tester to run a set of predefined test cases and test scripts.

Test case: This stage generates the test case for the testing process.

Test data: At this stage, the data to be tested is generated.

Test case execution: Once the test case and test data are generated, test cases are executed.

Reporting of defects: This is the stage where defects in the system are identified.

Regression testing: This is done to see if any problems were introduced into the development process by the previous stages.

Defect logging: All the identified defects are fixed at this stage.

Retest: A test is repeated if it's unsuccessful.

8.2 TEST CASES:

S.NO	INPUT	If available	If not available
1	User signup	User get registered into the application	There is no process
2	User signin	User get login into the application	There is no process
3	Enter input for prediction	Prediction result displayed	There is no process

CHAPTER-9

METHODOLOGY

9. Software Development Life Cycle (SDLC) – Umbrella Model

The Software Development Life Cycle (SDLC) is a structured approach followed in software development to ensure the creation of high-quality software that meets user requirements. The Umbrella Model encompasses all stages of software development, ensuring systematic progress from requirement gathering to maintenance.

In the case of Diabetic Retinopathy (DR) detection using Convolutional Neural Networks (CNNs), the SDLC model is crucial for developing a robust and efficient system that can accurately analyse retinal images, detect DR at different stages, and provide valuable insights.

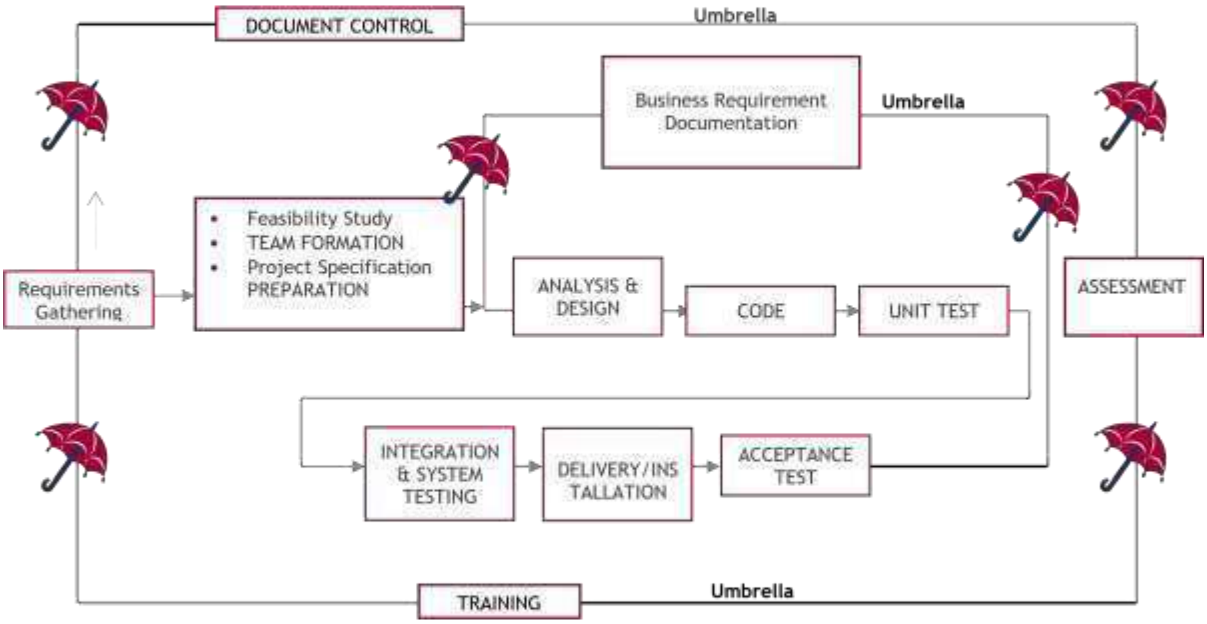


Fig 9.1 SDLC – Umbrella Model

9.1 Requirements Gathering Stage

The requirement gathering process involves defining the key functionalities and data requirements of the system. In this stage, the project's goals and objectives are refined into detailed requirements that outline the system's capabilities.

For Diabetic Retinopathy detection, the major requirements include:

- Input Data: Retinal fundus images collected from medical databases.
- Preprocessing Needs: Image enhancement, noise removal, and contrast adjustments.
- Feature Extraction: Identification of lesions, microaneurysms, hemorrhages, and exudates.
- Model Selection: CNN-based deep learning model for classification.
- Output Data: DR classification (e.g., No DR, Mild, Moderate, Severe, Proliferative).

Deliverables:

- Requirements Document – Describes the functional and non-functional system requirements.
- Requirements Traceability Matrix (RTM) – Ensures that every requirement is mapped to a project goal.
- Project Plan – Defines the timeline and resource allocation for development.

9.2 Feasibility Study and Project Specifications

The feasibility study ensures that the project is technically, financially, and operationally viable. The key aspects include:

- Technical Feasibility: Evaluates if CNN-based models can process high-resolution retinal images efficiently.
- Operational Feasibility: Ensures the system is user-friendly and accessible to ophthalmologists.

- Economic Feasibility: Estimates the cost of computing resources, data collection, and model training.

Project specifications define:

- System inputs: Retinal images and patient metadata.
- Processing methods: Image preprocessing, feature extraction, and CNN-based classification.
- Outputs: DR classification results with confidence scores.

9.3 Analysis Stage

The analysis phase provides a high-level view of the project's structure, risks, and strategies. This stage identifies:

- High-level system requirements: Image acquisition, preprocessing, feature extraction, and classification.
- Potential challenges: Variability in image quality, class imbalance, and computational requirements.
- Risk management: Implementing data augmentation and transfer learning to address limited training data.

Deliverables:

- Configuration Management Plan – Ensures version control of the software.
- Quality Assurance Plan – Defines testing and validation strategies.
- Project Plan and Schedule – Details milestones and estimated timelines.

9.4 Designing Stage

The design stage involves defining the system architecture and technical specifications. Key design elements include:

- Functional Hierarchy Diagrams: Show the interaction between image preprocessing, CNN model, and output generation.

- Screen Layout Diagrams: Illustrate user interfaces for data input and result visualization.
- Business Process Diagrams: Define the workflow of image processing and classification.
- Entity-Relationship (ER) Diagram: Represents database structure for storing retinal images and results.

Deliverables:

- Design Document – Contains detailed system architecture.
- Updated RTM – Maps design elements to requirements.

9.5 Development (Coding) Stage

The development phase involves implementing the system components. The CNN-based DR detection model is built using TensorFlow/ Keras with key steps:

1. Data Collection: Gather retinal images from public medical datasets (e.g., Kaggle APTOS, Messidor).
2. Preprocessing:
    - Resize images to a fixed dimension (e.g., 224x224 pixels).
    - Apply contrast enhancement and noise reduction.
    - Normalize pixel values for CNN input.
3. Model Development:
    - Implement CNN layers: Convolution, Pooling, Fully Connected layers.
    - Use pre-trained models (e.g., Res Net, Efficient Net) for transfer learning.
    - Optimize model parameters using techniques like dropout and batch normalization.
4. Training & Evaluation:
    - Split dataset into training, validation, and test sets.

    - Use accuracy, precision, recall, and F1-score as evaluation metrics.
    - Train using Adam optimizer and cross-entropy loss function.

Deliverables:

- CNN model architecture and code.
- Model checkpoints and logs for tracking performance.

9.6 Integration & Testing Stage

During this phase, individual software components are integrated and tested in a separate test environment. The system undergoes:

- Unit Testing: Verifies each module (preprocessing, feature extraction, classification) independently.
- Integration Testing: Ensures that different modules work together correctly.
- Performance Testing: Measures model inference time and accuracy.
- Validation Testing: Compares results with ophthalmologist diagnoses.

Deliverables:

- Test Reports – Document accuracy, errors, and performance benchmarks.
- Finalized Production Initiation Plan – Outlines dataset requirements and deployment strategies.

9.7 Installation & Acceptance Testing

This stage involves deploying the software in a real-world environment. Key steps include:

- Deploying the model on a cloud or local server.
- Integrating a user-friendly interface for image upload and result display.
- Conducting acceptance tests with ophthalmologists to validate results.

Acceptance criteria include:

- Model correctly classifies at least 85% of test images.

- System provides interpretability features (e.g., heatmaps highlighting affected areas).

Deliverables:

- Deployment report – Documents system setup and configurations.
- User acceptance test results – Confirms model reliability.

9.8 Maintenance Phase

After deployment, the maintenance team ensures system performance and accuracy by:

- Monitoring model drift – Regularly updating the model with new retinal images.
- Fixing software bugs – Addressing issues in preprocessing and classification.
- Improving accuracy – Fine-tuning model parameters and adding new features.

Ongoing improvements include:

- Training on larger datasets to enhance robustness.
- Incorporating Explainable AI (XAI) for better interpretability.

CHAPTER 10

SOURCE CODE

```
from _future_ import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np

# Keras
from keras.models import load_model

from PIL import Image as im
# Flask utils
from flask import Flask, redirect, url_for, request, render_template

import sqlite3

import numpy as np
import pickle
import sqlite3
import random

import smtplib
from email.message import EmailMessage


import sqlite3


from flask import Flask, render_template, request, redirect, Response



app = Flask(_name_)

UPLOAD_FOLDER = 'static/uploads/'

# allow files of a specific type
ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])

# function to check the file extension
def allowed_file(filename):
    return '.' in filename and \
            filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
```

```python
model_path1 = 'xception.h5' # load .h5 Model

from keras import backend as K

def recall_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = true_positives / (possible_positives + K.epsilon())
    return recall

def precision_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    return precision

def f1_m(y_true, y_pred):
    precision = precision_m(y_true, y_pred)
    recall = recall_m(y_true, y_pred)
    return 2*((precision*recall)/(precision+recall+K.epsilon()))




model1 = load_model(model_path1, custom_objects={'f1_score' : f1_m,
'precision_score' : precision_m, 'recall_score' : recall_m}, compile=False)


from keras.preprocessing.image import load_img, img_to_array

def model_predict1(image_path,model):
    print("Predicted")
    image = load_img(image_path,target_size=(128,128))
    image = img_to_array(image)
    image = image/255
    image = np.expand_dims(image,axis=0)

    result = np.argmax(model.predict(image))
    print(result)
    #prediction = classes2[result]
    if result == 0:
        return "Mild DR - Non-proliferative diabetic retinopathy (NPDR) is the
early stage of the disease in which symptoms will be mild or
nonexistent.","result.html"
    elif result == 1:
```

```
        return "Moderate nonproliferative diabetic retinopathy,the tiny blood
vessels further swell up, blocking blood flow to the retina and preventing
proper nourishment.","result.html"
    elif result == 2:
        return "No Diabetic Retinopathy","result.html"
    elif result == 3:
        return "Proliferate DR - is the advanced stage where abnormal new
blood vessels grow on the surface of the retina!","result.html"
    elif result == 4:
        return "Severe DR - is an eye condition that can cause vision loss and
blindness in people who have diabetes. ","result.html"




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


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


@app.route('/logon')
def logon():
    return render_template('signup.html')

@app.route('/login')
def login():
    return render_template('signin.html')


@app.route('/index')
def index():
    return render_template('index.html')


@app.route('/predict',methods=['GET','POST'])
def predict():
    print("Entered")

    print("Entered here")
    file = request.files['file'] # fet input
    filename = file.filename
    print("@@ Input posted = ", filename)
```

```
    file_path = os.path.join(UPLOAD_FOLDER, filename)
    file.save(file_path)

    print("@@ Predicting class...... ")
    pred, output_page = model_predict1(file_path,model1)

    return render_template(output_page, pred_output = pred,
img_src=UPLOAD_FOLDER + file.filename)



@app.route("/signup")
def signup():
    global otp, username, name, email, number, password
    username = request.args.get('user','')
    name = request.args.get('name','')
    email = request.args.get('email','')
    number = request.args.get('mobile','')
    password = request.args.get('password','')
    otp = random.randint(1000,5000)
    print(otp)
    msg = EmailMessage()
    msg.set_content("Your OTP is : "+str(otp))
    msg['Subject'] = 'OTP'
    msg['From'] = "manojtruprojects@gmail.com"
    msg['To'] = email


    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
    s.login("manojtruprojects@gmail.com", "qvhanvuuxyogomze")
    s.send_message(msg)
    s.quit()
    return render_template("val.html")

@app.route('/predict_lo', methods=['POST'])
def predict_lo():
    global otp, username, name, email, number, password
    if request.method == 'POST':
        message = request.form['message']
        print(message)
        if int(message) == otp:
            print("TRUE")
            con = sqlite3.connect('signup.db')
            cur = con.cursor()
            cur.execute("insert into info (user,email, password,mobile,name)
VALUES (?, ?, ?, ?, ?)",(username,email,password,number,name))
            con.commit()
            con.close()
```

```
            return render_template("signin.html")
    return render_template("signup.html")

@app.route("/signin")
def signin():

    mail1 = request.args.get('user','')
    password1 = request.args.get('password','')
    con = sqlite3.connect('signup.db')
    cur = con.cursor()
    cur.execute("select user, password from info where user = ? AND password =
?",(mail1,password1,))
    data = cur.fetchone()

    if data == None:
        return render_template("signin.html")

    elif mail1 == str(data[0]) and password1 == str(data[1]):
        return render_template("index.html")
    else:
        return render_template("signin.html")

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


if _name_ == '_main_':
    app.run(debug=False)
# coding=utf-8 from _future_ import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np

# Keras
from keras.models import load_model

from PIL import Image as im
# Flask utils
from flask import Flask, redirect, url_for, request, render_template

import sqlite3

import numpy as np
import pickle
```

```python
import sqlite3
import random

import smtplib
from email.message import EmailMessage


import sqlite3


from flask import Flask, render_template, request, redirect, Response



app = Flask(_name_)

UPLOAD_FOLDER = 'static/uploads/'

# allow files of a specific type
ALLOWED_EXTENSIONS = set(['png', 'jpg', 'jpeg'])

# function to check the file extension
def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS


model_path1 = 'xception.h5' # load .h5 Model

from keras import backend as K

def recall_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = true_positives / (possible_positives + K.epsilon())
    return recall

def precision_m(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    return precision

def f1_m(y_true, y_pred):
    precision = precision_m(y_true, y_pred)
    recall = recall_m(y_true, y_pred)
    return 2*((precision*recall)/(precision+recall+K.epsilon()))
```

```
model1 = load_model(model_path1, custom_objects={'f1_score' : f1_m,
'precision_score' : precision_m, 'recall_score' : recall_m}, compile=False)


from keras.preprocessing.image import load_img, img_to_array

def model_predict1(image_path,model):
    print("Predicted")
    image = load_img(image_path,target_size=(128,128))
    image = img_to_array(image)
    image = image/255
    image = np.expand_dims(image,axis=0)

    result = np.argmax(model.predict(image))
    print(result)
    #prediction = classes2[result]
    if result == 0:
        return "Mild DR - Non-proliferative diabetic retinopathy (NPDR) is the
early stage of the disease in which symptoms will be mild or
nonexistent.","result.html"
    elif result == 1:
        return "Moderate nonproliferative diabetic retinopathy,the tiny blood
vessels further swell up, blocking blood flow to the retina and preventing
proper nourishment.","result.html"
    elif result == 2:
        return "No Diabetic Retinopathy","result.html"
    elif result == 3:
        return "Proliferate DR - is the advanced stage where abnormal new
blood vessels grow on the surface of the retina!","result.html"
    elif result == 4:
        return "Severe DR - is an eye condition that can cause vision loss and
blindness in people who have diabetes. ","result.html"




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


@app.route('/')
@app.route('/home')
def home():
    return render_template('home.html')
```

```python
@app.route('/logon')
def logon():
    return render_template('signup.html')

@app.route('/login')
def login():
    return render_template('signin.html')


@app.route('/index')
def index():
    return render_template('index.html')


@app.route('/predict',methods=['GET','POST'])
def predict():
    print("Entered")

    print("Entered here")
    file = request.files['file'] # fet input
    filename = file.filename
    print("@@ Input posted = ", filename)

    file_path = os.path.join(UPLOAD_FOLDER, filename)
    file.save(file_path)

    print("@@ Predicting class...... ")
    pred, output_page = model_predict1(file_path,model1)

    return render_template(output_page, pred_output = pred,
img_src=UPLOAD_FOLDER + file.filename)



@app.route("/signup")
def signup():
    global otp, username, name, email, number, password
    username = request.args.get('user','')
    name = request.args.get('name','')
    email = request.args.get('email','')
    number = request.args.get('mobile','')
    password = request.args.get('password','')
    otp = random.randint(1000,5000)
    print(otp)
    msg = EmailMessage()
    msg.set_content("Your OTP is : "+str(otp))
    msg['Subject'] = 'OTP'
```

```
    msg['From'] = "manojtruprojects@gmail.com"
    msg['To'] = email


    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
    s.login("manojtruprojects@gmail.com", "qvhanvuuxyogomze")
    s.send_message(msg)
    s.quit()
    return render_template("val.html")

@app.route('/predict_lo', methods=['POST'])
def predict_lo():
    global otp, username, name, email, number, password
    if request.method == 'POST':
        message = request.form['message']
        print(message)
        if int(message) == otp:
            print("TRUE")
            con = sqlite3.connect('signup.db')
            cur = con.cursor()
            cur.execute("insert into info (user,email, password,mobile,name)
VALUES (?, ?, ?, ?, ?)",(username,email,password,number,name))
            con.commit()
            con.close()
            return render_template("signin.html")
    return render_template("signup.html")

@app.route("/signin")
def signin():

    mail1 = request.args.get('user','')
    password1 = request.args.get('password','')
    con = sqlite3.connect('signup.db')
    cur = con.cursor()
    cur.execute("select user, password from info where user = ? AND password =
?",(mail1,password1,))
    data = cur.fetchone()

    if data == None:
        return render_template("signin.html")

    elif mail1 == str(data[0]) and password1 == str(data[1]):
        return render_template("index.html")
    else:
        return render_template("signin.html")

@app.route("/notebook")
```

```
def notebook():
    return render_template("Notebook.html")


if _name_ == '_main_':
    app.run(debug=False)
```

CHAPTER-11

RESULTS AND DISCUSSION

11.1 MODULES:

1. **Data loading:** using this module we are going to import the dataset.

2. **Image Processing:** Image processing involves techniques like rescaling, shear transformation, zooming, horizontal flipping, and reshaping to enhance image quality. These steps help prepare the dataset for feature extraction using pre-trained CNN architectures.

3. **Model generation:** Model building - ResNet50, InceptionV3, DiaNet -Ensemble of ResNet + InceptionV3 Model For Feature Extraction and Analysis with CNN, Xception. Performance evaluation metrics for each algorithm is calculated.

4. **User signup & login:** Using this module will get registration and login

5. **User input**: Using this module will give input for prediction

6. **Prediction:** final predicted displayed

11.2 SCREENSHOTS:

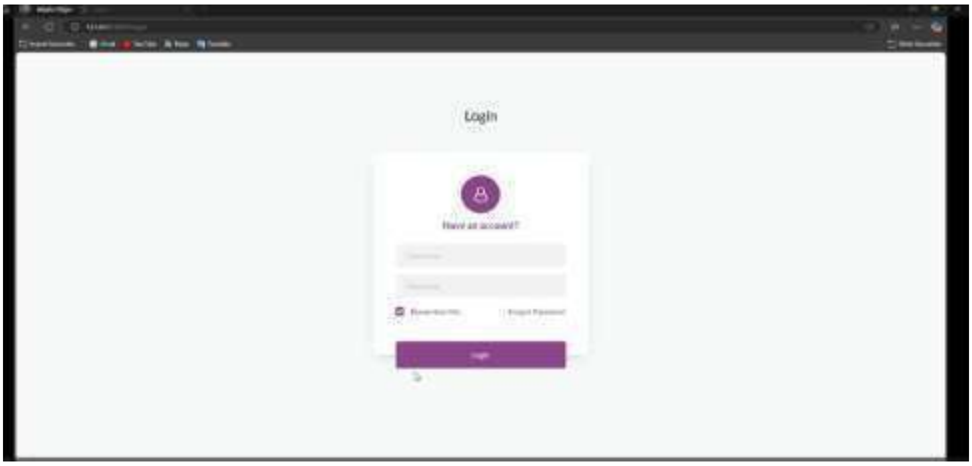


Fig 11.2.1 Login Page

The image shows a login page of a web application running on localhost (127.0.0.1:5000) with fields for username, password, and options for "Remember Me" and "Forgot Password."

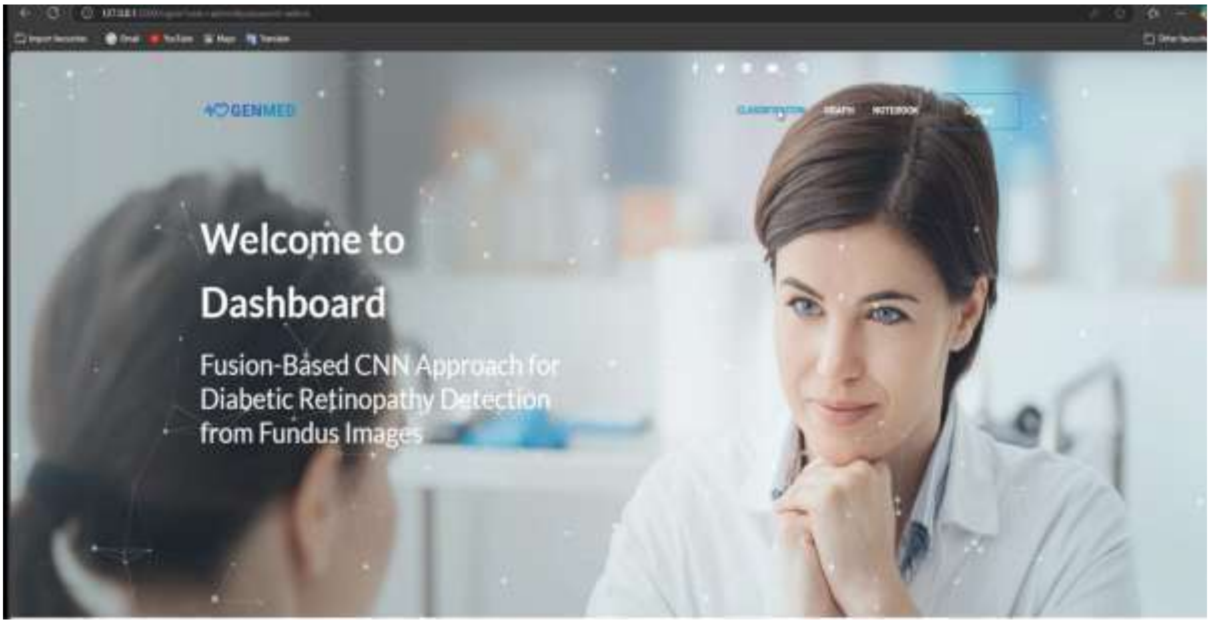


Fig 11.2.2 Cover Page

The image shows a dashboard for a Fusion-Based CNN Approach used for Diabetic Retinopathy Detection from Fundus Images, with options for classification, graph, notebook, and sign-out on a localhost (127.0.0.1:5000) web application.

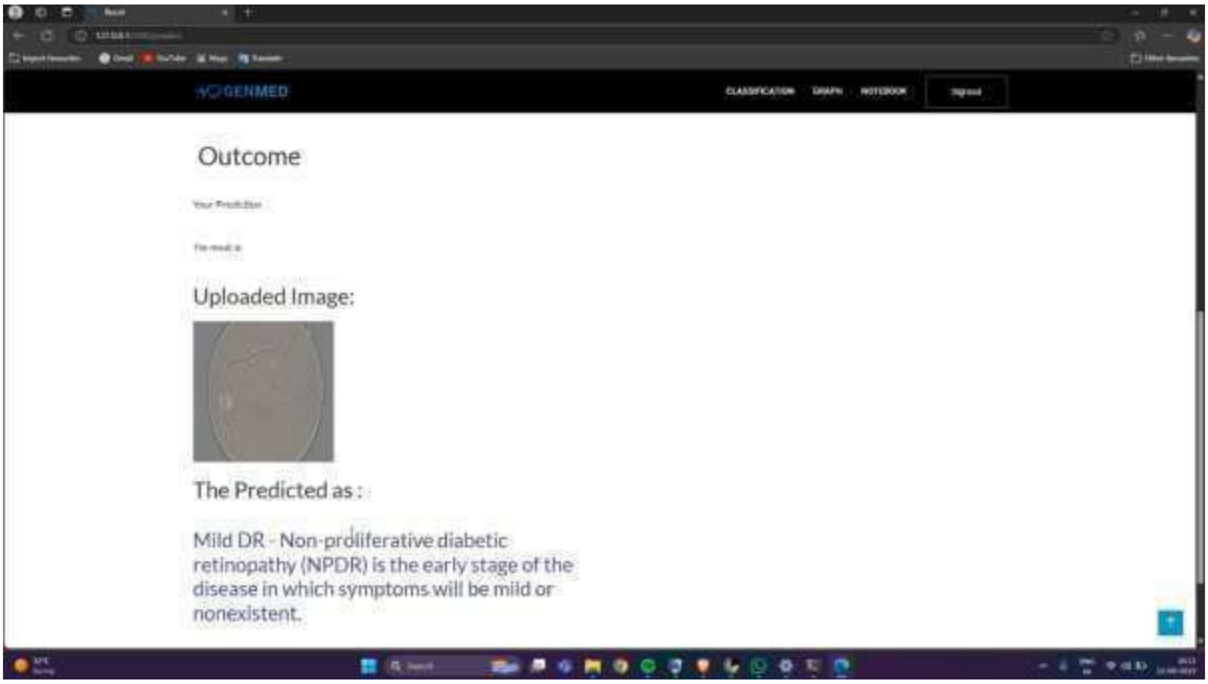


Fig 11.5.3 Output Page-1

The image shows a prediction result page for a diabetic retinopathy detection system, classifying the uploaded fundus image as Mild DR - Non-proliferative diabetic retinopathy (NPDR) on a localhost (127.0.0.1:5000/predict) web application.

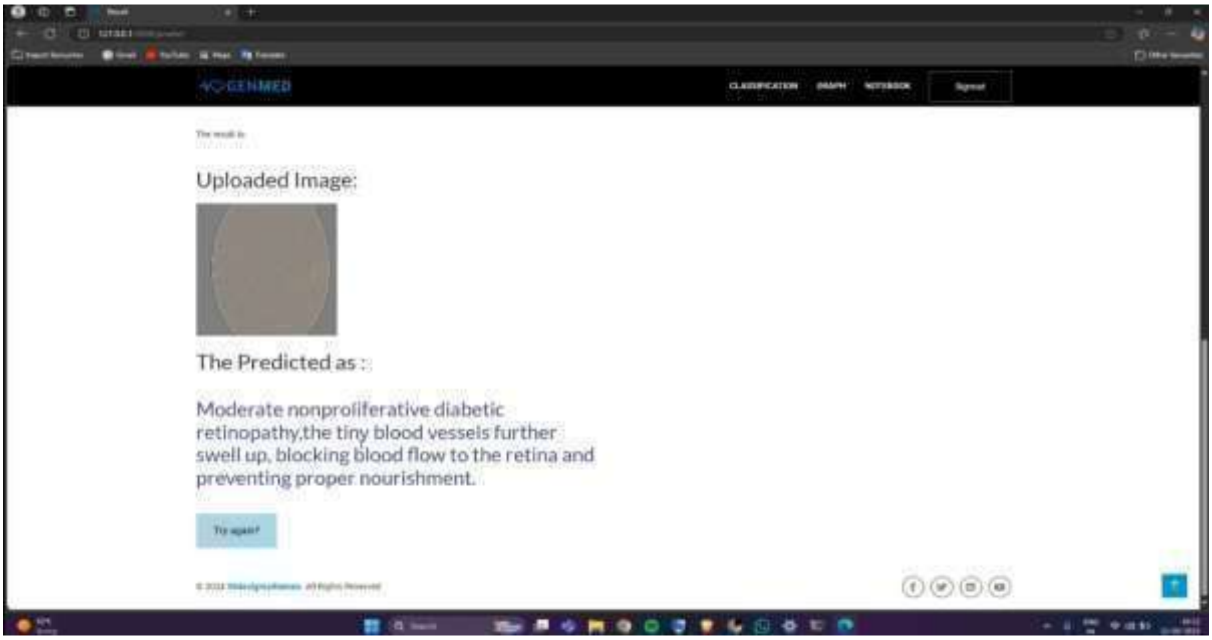


Fig 11.5.4 Output Page-2

The image shows the diabetic retinopathy classification result from a localhost web application (127.0.0.1:5000/predict).The uploaded fundus image has been classified as Moderate Non proliferative Diabetic Retinopathy (NPDR). This stage is characterized by swelling of tiny blood vessels, blocking blood flow to the retina, and preventing proper nourishment.

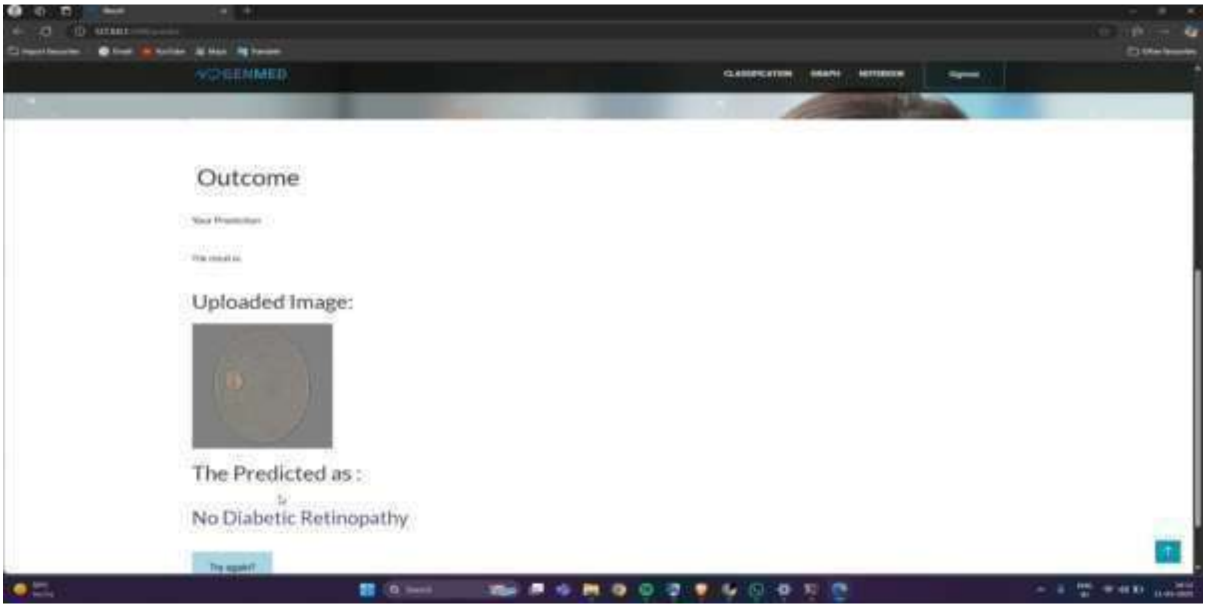


Fig 11.5.5 Output Page-3

The image displays a diabetic retinopathy detection result, where the uploaded fundus image is classified as No Diabetic Retinopathy on a localhost web application (127.0.0.1:5000/predict)

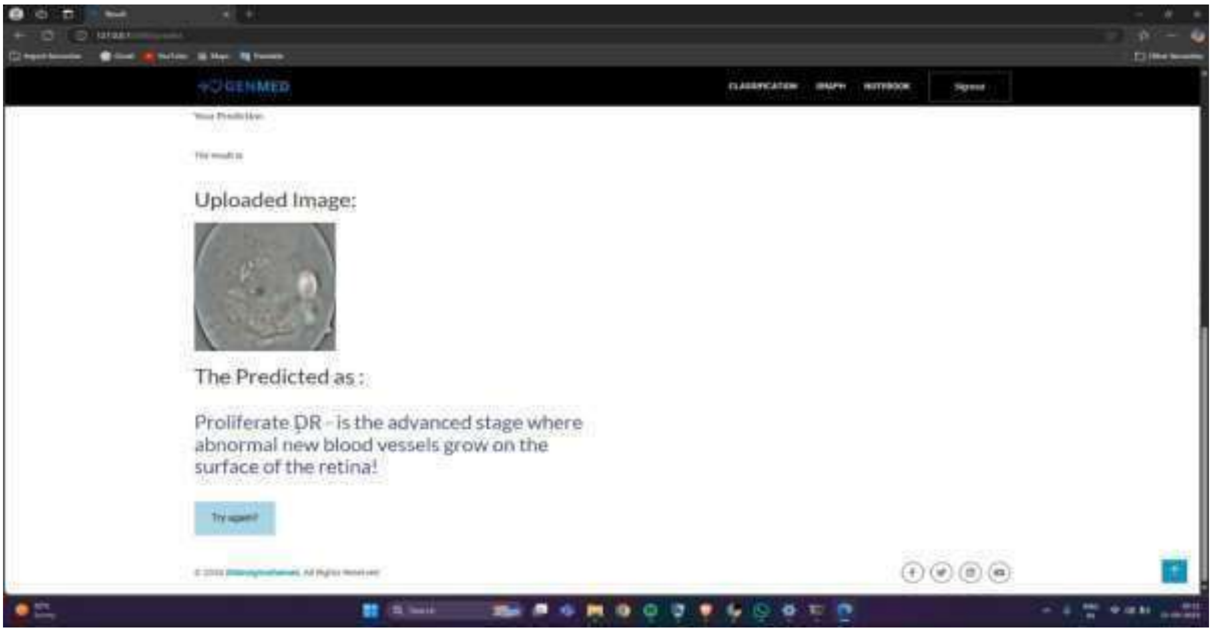


Fig 11.5.6 Output Page-4

The image shows a diabetic retinopathy detection result from a localhost web application (127.0.0.1:5000/predict). The uploaded fundus image has been classified as Proliferative Diabetic Retinopathy (PDR), which is an advanced stage characterized by the growth of abnormal new blood vessels on the retina's surface.

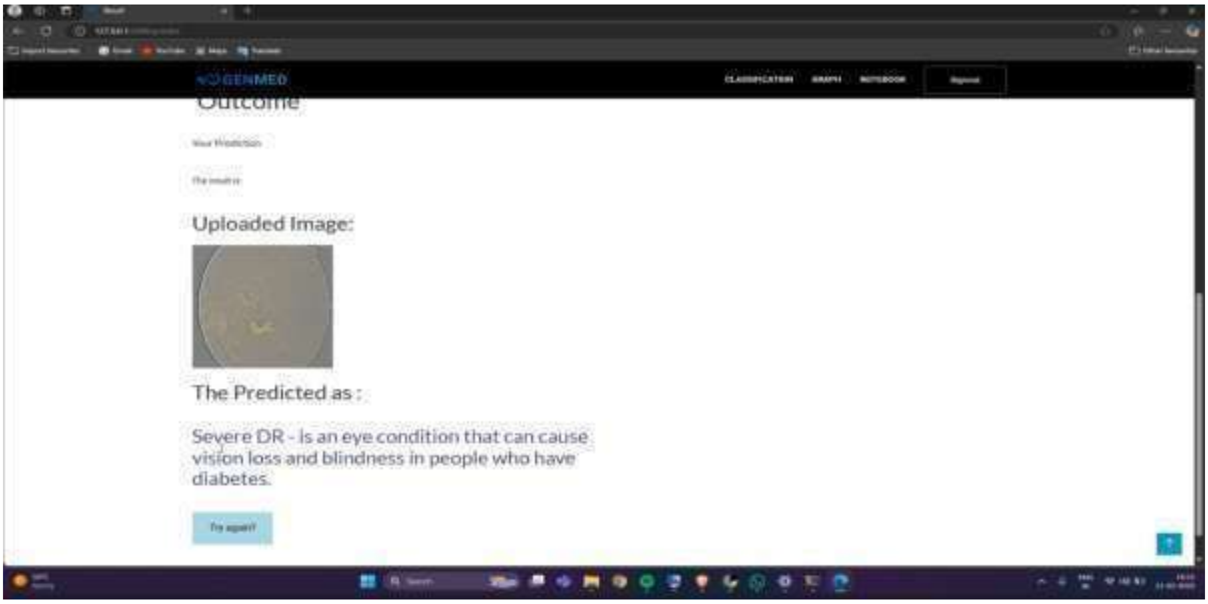


Fig 11.5.7 Output Page-5

The screenshot shows a web interface and displays the results of an image classification. It indicates that the uploaded image, a circular object with text, is predicted as "Severe DR - is an eye condition that can cause vision loss and blindness in people who have diabetes".

11.3 PERFORMANCE COMPARISON AND ACCURACY GRAPH

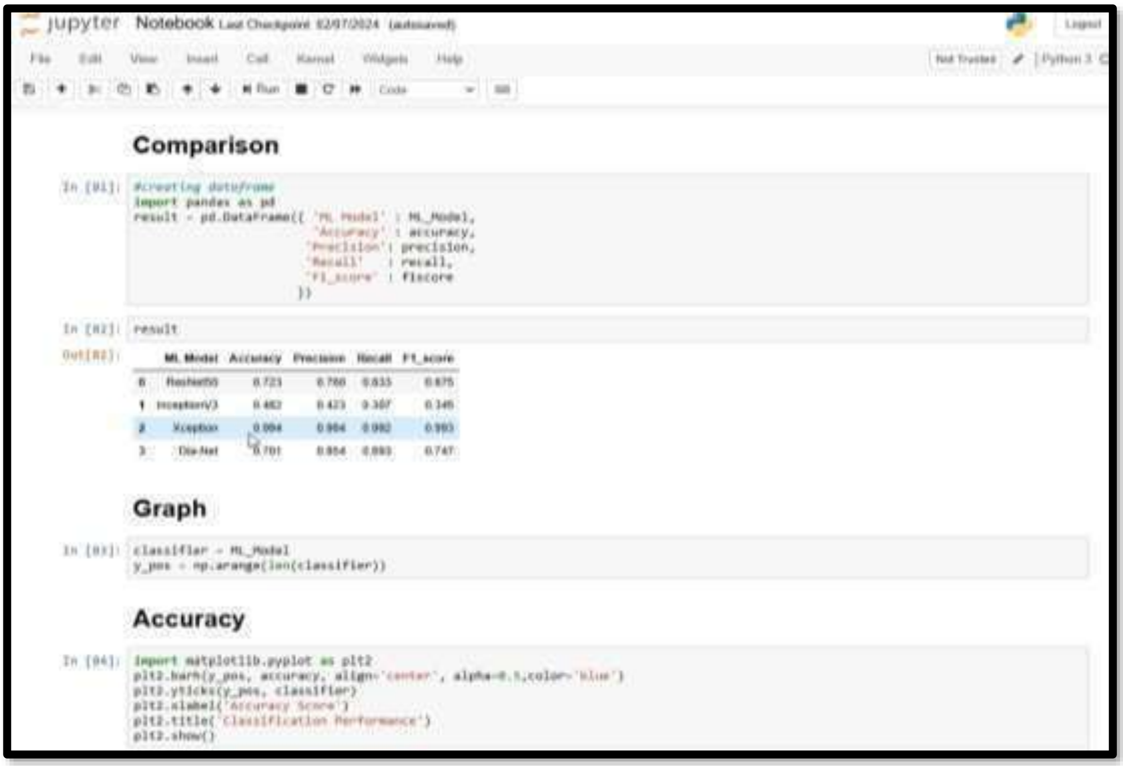


Fig 11.3.1 Comparison of ML model

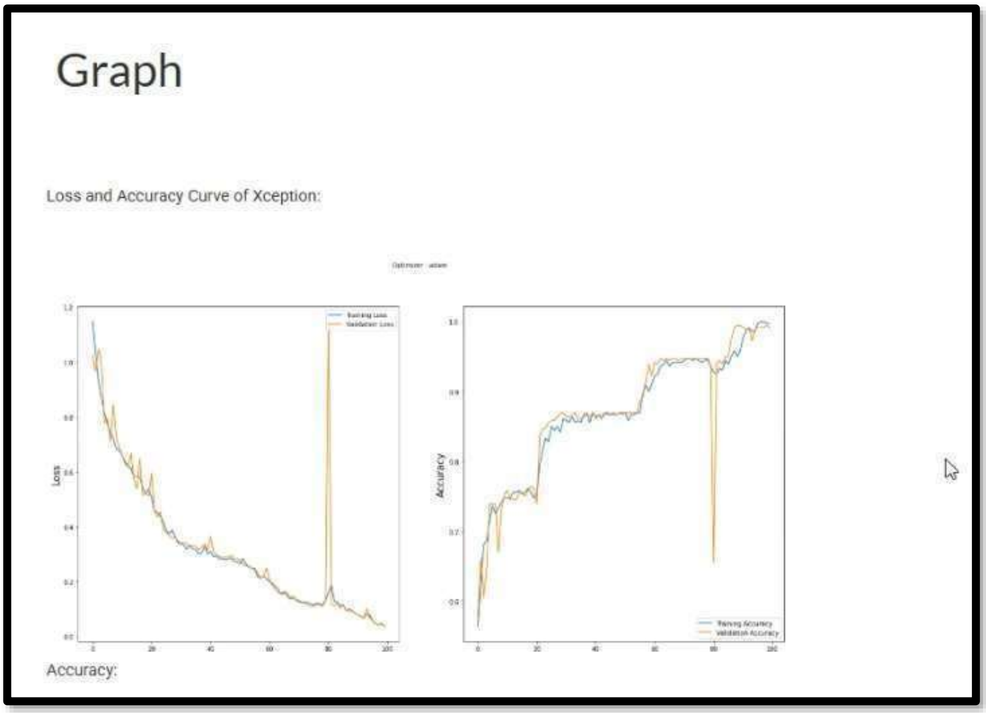


Fig 11.3.2 Loss and Accuracy Graph of Xception

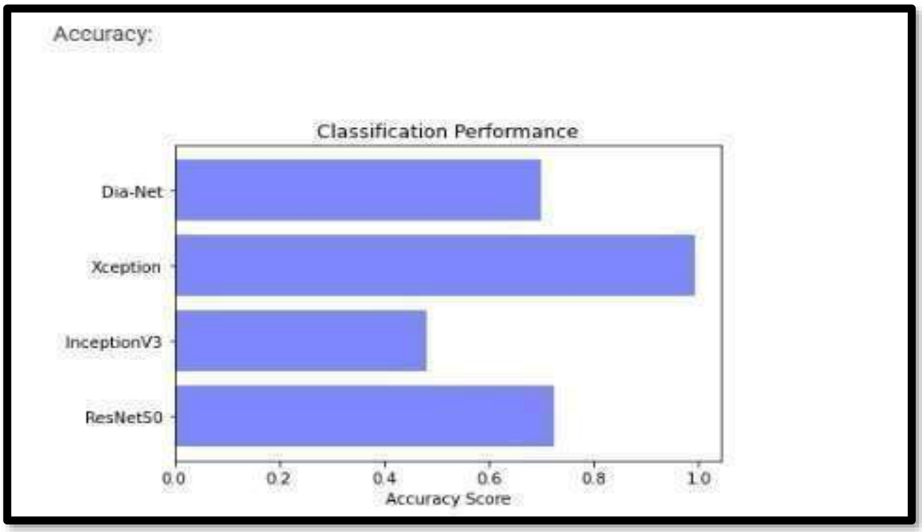


Fig 11.3.3 Accuracy Comparison

Accuracy: A test's accuracy refers to its capability to correctly distinguish between patients and healthy individuals. To determine this accuracy, it is essential to compute the rate of true positives and true negatives among all assessed cases. This can be mathematically expressed as:

Accuracy = (TP + TN) / (TP + FP + TN + FN) (1)

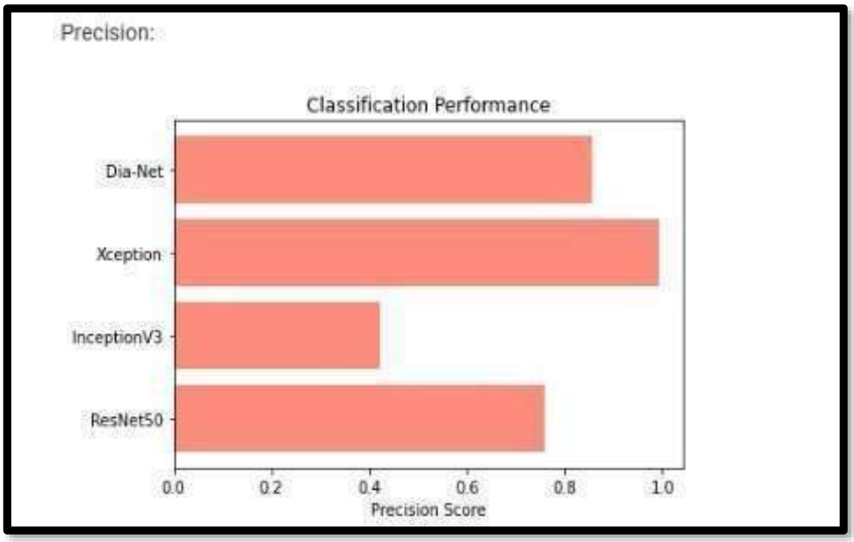


Fig 11.3.4 Precision Comparison

Precision: This metric assesses the share of instances that were accurately classified among those identified as positive. Therefore, precision can be calculated using the following formula:

Precision = (True Positive) / (True Positive + False Positive) (2)

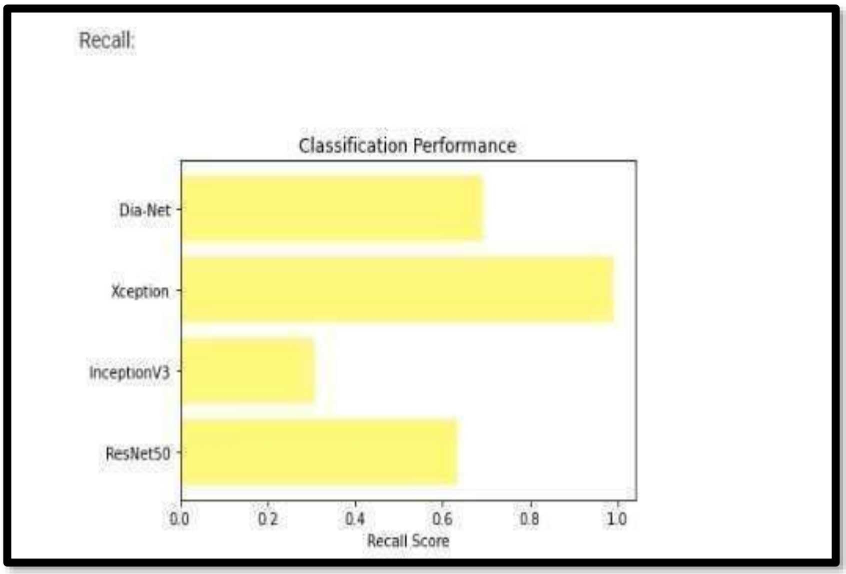


Fig 11.3.5 Recall Comparison

Recall: Recall is a measure in machine learning that gauges how well a model can recognize all relevant instances of a specific category. It represents the ratio of correctly predicted positive outcomes to the total actual positives, giving insights into the model’s thoroughness in capturing instances of a certain class.

Recall = TP / (TP + FN) (3)

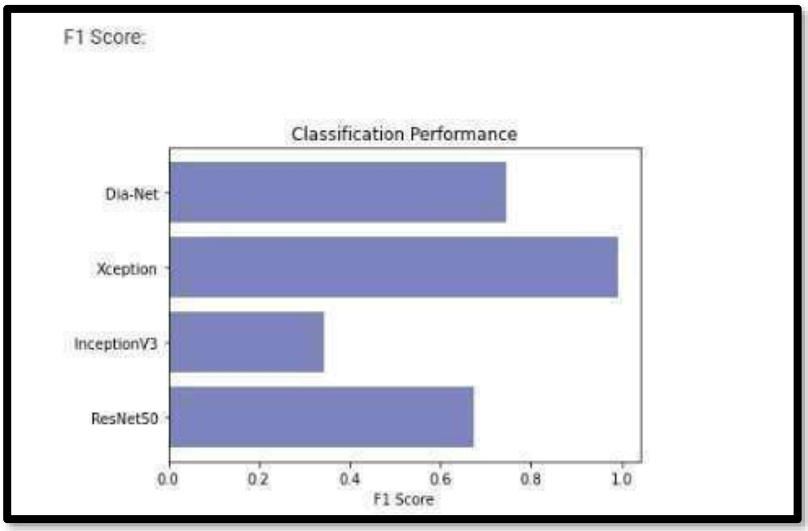


Fig 11.3.6 F1 Score Comparison

F1 Score: The F1 score serves as a metric to evaluate a model's accuracy in machine learning. It merges the precision and recall scores. The accuracy measurement assesses how many correct predictions a model has made throughout the whole dataset.

CHAPTER 12

CONCLUSION AND FUTURE SCOPE

12.1 CONCLUSION

In conclusion, the proposed system demonstrates the effectiveness of using fusion-based Convolutional Neural Networks (CNNs) for the automated detection of Diabetic Retinopathy (DR) in fundus images. By classifying DR into five distinct categories, the system ensures a more comprehensive assessment of the disease, enabling timely interventions that can prevent severe vision loss. The use of deep learning techniques, particularly CNNs, provides the ability to capture complex patterns and features from fundus images, facilitating accurate classification of DR severity. Among the various models experimented with, the Xception algorithm has shown exceptional performance, offering high accuracy and robustness in classifying the different stages of DR. Its superior feature extraction and classification abilities make it a highly reliable choice for automated DR detection. This approach not only streamlines the detection process but also improves the efficiency of ophthalmologists by prioritizing critical cases. The high performance of the Xception model ensures precise and effective DR screening, enhancing healthcare delivery and contributing significantly to the overall quality of life for patients.

Future work can explore the integration of additional deep learning models, such as Transformers and attention-based networks, to further enhance the accuracy and robustness of DR detection. Additionally, incorporating multimodal data, such as patient demographics and clinical records, could improve classification outcomes. Expanding the dataset diversity and addressing class imbalance through advanced techniques like synthetic data generation would strengthen the model's generalizability. Furthermore, real-time implementation in clinical settings could be explored for immediate detection and intervention.

12.2 FUTURE SCOPE

- **Integration of Advanced Models**: Future research can incorporate cutting-edge deep learning architectures like Transformers and attention-based networks to enhance model accuracy and robustness in diabetic retinopathy (DR) detection.
- **Multimodal Data Utilization**: Combining fundus images with patient demographics**,** medical history, and clinical records could lead to more precise and personalized diagnosis outcomes.
- **Improved Dataset Diversity**: Expanding datasets to include images from diverse populations and conditions can improve model reliability and performance across different demographics.
- **Class Imbalance Solutions**: Implementing techniques like synthetic data generation (e.g., GANs) **or** data augmentation can help address class imbalance and improve the learning process for minority classes.
- **Real-Time Clinical Integration**: Developing systems for real-time DR detection can support immediate diagnosis and treatment in clinical environments, improving patient care and reducing workload on ophthalmologists.
- **Mobile and Cloud Deployment**: Future enhancements could involve deploying the system on mobile or cloud platforms for wider accessibility and scalability.

CHAPTER- 13

REFERENCES

[1] Nur-A-Alam, M., Nasir, M. M. K., Ahsan, M., Based, M. A., Haider, J., & Palani, S. (2023). A faster RCNN-based diabetic retinopathy detection method using fused features from retina images. IEEE Access, 11, 124331-124349.

[2] Alshahrani, M., Al-Jabbar, M., Senan, E. M., Ahmed, I. A., & Saif, J. A. M. (2023). Hybrid Methods for Fundus Image Analysis for Diagnosis of Diabetic Retinopathy Development Stages Based on Fusion Features. Diagnostics, 13(17), 2783.

[3] Nneji, G. U., Cai, J., Deng, J., Monday, H. N., Hossin, M. A., & Nahar, S. (2022). Identification of diabetic retinopathy using weighted fusion deep learning based on dual-channel fundus scans. Diagnostics, 12(2), 540.

[4] Maqsood, S., Damaševičius, R., & Maskeliūnas, R. (2021). Hemorrhage detection based on 3D CNN deep learning framework and feature fusion for evaluating retinal abnormality in diabetic patients. Sensors, 21(11), 3865.

[5] M. Bilal, G. Ali, M. W. Iqbal, M. Anwar, M. S. A. Malik, and R. A. Kadir, ‘‘Auto prep: Efficient and automated data preprocessing pipeline,’’ IEEE Access, vol. 10, pp. 107764–107784, 2022.

[6] Madhavi, K. Reddy, Padmavathi Kora, L. Venkateswara Reddy, Janagaraj Avanija, K. L. S. Soujanya, and Prabhakar Telagarapu. "Cardiac arrhythmia detection using dual tree wavelet transform and convolutional neural network." Soft Computing 26, no. 7 (2022): 3561-3571.

[7] Reddy Madhavi, K., A. Vinaya Babu, G. Sunitha, and J. Avanija. "Detection of concept-drift for clustering time-changing categorical data: An optimal method for large datasets." In Data Engineering and Communication Technology: Proceedings of 3rd ICDECT-2K19, pp. 861-871. Springer Singapore, 2020.

[8] M. J. Iqbal, M. W. Iqbal, M. Anwar, M. M. Khan, A. J. Nazimi, and M. N. Ahmad, ‘‘Brain tumor segmentation in multimodal MRI using U-Net layered structure,’’ Comput., Mater. Continua, vol. 74, no. 3, pp. 5267–5281, 2023.

[9] M. Anwar, F. Masud, R. A. Butt, S. M. Idrus, M. N. Ahmad, and M. Y. Bajuri, ‘‘Traffic priority-aware medical data dissemination scheme for IoT based WBASN healthcare applications,’’ Comput., Mater. Continua, vol. 71, no. 3, pp. 4443–4456, 2022.

[10] S. Naseem, A. Alhudhaif, M. Anwar, K. N. Qureshi, and G. Jeon, ‘‘Artificial intelligence-based behavior detection using correlates for tracking online harms,’’ Pers. Ubiquitous Comput., vol. 27, no. 1, pp. 119–137, Feb. 2023.

[11] Manjushree, R., Bhoomika, D., Nair, R. R., & Babu, T. (2022, December). Automated detection of diabetic retinopathy using deep learning in retinal fundus images: analysis. In *2022 3rd International Conference on Communication, Computing and Industry 4.0 (C2I4)* (pp. 1-6). IEEE.

[12] Kanakaprabha, S., Radha, D., & Santhanalakshmi, S. (2021, April). Diabetic retinopathy detection using deep learning models. In *International Conference on Ubiquitous Computing and Intelligent Information Systems* (pp. 75-90). Singapore: Springer Nature Singapore.

[13] Rai, B. K., Ojha, H., & Srivastava, I. (2024, March). Diabetic Retinopathy Detection using Deep Learning Model ResNet15. In *2024 2nd International Conference on Disruptive Technologies (ICDT)* (pp. 1361-1366). IEEE.

[14] Sharmila, S., Thejas, V. N., Supriya, C., Sumukh, S., & Chethana, H. T. (2022). A review on detection of diabetic retinopathy. *Advances in Data and Information Sciences: Proceedings of ICDIS 2022*, 161-171.

[15] Goel, N., & Singh, S. K. (2022, December). Diabetic Retinopathy Image Analysis Using Deep Learning Techniques. In *2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (pp. 1-6). IEEE.

[16] Caicho, J., Chuya-Sumba, C., Jara, N., Salum, G. M., Tirado-Espín, A., Villalba-Meneses, G., ... & Almeida-Galárraga, D. A. (2021, December). Diabetic retinopathy: detection and classification using AlexNet, GoogleNet and ResNet50 convolutional neural networks.

In *International conference on smart technologies, systems and applications* (pp. 259-271). Cham: Springer International Publishing.

[17] Bhimavarapu, U., Chintalapudi, N., & Battineni, G. (2023). Automatic detection and classification of diabetic retinopathy using the improved pooling function in the convolution neural network. *Diagnostics*, *13*(15), 2606.

[18] Jwad, F. J., & Abdulmunem, A. A. (2024, May). Diabetic Retinopathy Identification Depend On Deep Learning Techniques: A comparative Study. In *2024 International Conference on Intelligent Systems and Computer Vision (ISCV)* (pp. 1-6). IEEE.

[19] Rahhal, D., Alhamouri, R., Albataineh, I., & Duwairi, R. (2022, June). Detection and classification of diabetic retinopathy using artificial intelligence algorithms. In *2022 13th International Conference on Information and Communication Systems (ICICS)* (pp. 15-21). IEEE.

[20] Saranya, P., Devi, S. K., & Bharanidharan, B. (2022, March). Detection of diabetic retinopathy in retinal fundus images using densenet based deep learning model. In *2022 international mobile and embedded technology conference (MECON)* (pp. 268-272). IEEE.