DIABETIC RETINOPATHY DETECTION

Mini Project - Report

Submitted by

APEKSHA S SHETTY (4NM21CS031)

ANVITHA SHETTY (4NM21CS29)

6th Semester B.E.

Under the Guidance of

DR. SARIKA HEGDE
Professor

MS JAYAPADMINI KANCHAN
Assistant Professor Gd III

In partial fulfillment of the requirements for the award of

the Degree of

Bachelor of Engineering in Computer Science and Engineering

Department of Computer Science and Engineering
NMAM Institute of Technology, Nitte - 574110
(Deemed to be University)

MAY 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Mini project work entitled

" Diabetic Retinopathy Detection"

is a bonafide work carried out by

APEKSHA S SHETTY (4NM21CS031)

ANVITHA SHETTY (4NM21CS029)

of 6th Semester B.E. in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree in **Mechanical Engineering**

prescribed by **Visvesvaraya Technological University, Belagavi**

during the year ***2023-2024***.

____________________ ____________________

Signature of the Guide ***Signature of the HOD***

Viva Voce Examination

Name of the Examiners	**Signature with Date**
1. ____________________	____________________
2. ____________________	____________________

ABSTRACT

Diabetic retinopathy (DR) is a leading cause of blindness among working-age adults worldwide. Early detection and timely intervention are crucial for preventing vision loss in diabetic patients. In this project, we propose a solution for automated diabetic retinopathy detection using deep learning techniques. We preprocess retinal images to enhance features relevant to diabetic retinopathy. Next, we employ transfer learning, leveraging pre-trained convolutional neural networks (CNNs) such as Xception, to extract high-level features from retinal images. These features are then fed into a custom classification model to classify images into different stages of diabetic retinopathy. To assess the effectiveness of our model, we utilize a comprehensive dataset containing a considerable number of retinal images depicting four distinct stages of diabetic retinopathy: mild, moderate, severe, and proliferative. We conduct both training and validation phases using this dataset, implementing methodologies such as data augmentation to enhance the model's ability to generalize and maintain robustness. The proposed solution aims to provide a user-friendly interface for healthcare professionals to upload retinal images for automated diabetic retinopathy screening. Our system enables early detection of diabetic retinopathy, facilitating prompt intervention and management to prevent irreversible vision loss in diabetic patients.

ACKNOWLEDGEMENT

The completion of this project on Diabetic Retinopathy Detection has been made possible through the collective efforts of numerous individuals and organizations, and we extend our heartfelt appreciation to all those who have contributed to its success. We are sincerely thankful for the support and guidance we have received throughout this journey.

We wish to express our deepest gratitude to **Dr. Niranjan N Chiplunkar**, Principal of NMAMIT, Nitte, for granting us this valuable opportunity. Additionally, we extend our thanks to **Dr. Jyothi Shetty**, Professor and Head of the Department of Computer Science and Engineering at NMAMIT, Nitte.

Special thanks to our guides **Dr. Sarika Hegde** and **Ms. Jayapadmini Kanchan** for their invaluable guidance and provision of essential information throughout the project.

We are extremely grateful to our parents and friends who guided us throughout this project by providing immense support. We are also appreciative of the assistance provided by the staff and laboratory personnel, whose support allowed us to utilize the laboratory facilities effectively throughout the project duration.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 BRIEF OVERVIEW OF THE PROJECT

Diabetic retinopathy (DR) is a common complication of diabetes and a leading cause of vision loss among working-age adults. It occurs when high blood sugar levels damage blood vessels in the retina, leading to vision impairment and potentially blindness if left untreated. Early detection and intervention are crucial in managing DR and preventing vision loss.

In the case of DR, AI can be used to analyze retinal images and identify signs of the disease with high accuracy and efficiency. By leveraging deep learning techniques such as convolutional neural networks (CNNs)[3] and transfer learning, researchers and developers can create automated systems capable of detecting DR from digital retinal images. These systems can learn to recognize subtle patterns and features indicative of DR, enabling early diagnosis and timely intervention.

The project aims to create a deep learning model to accurately detect diabetic retinopathy (DR) stages in retinal images[2]. This involves improving image quality and extracting key features using transfer learning methods. The model will be trained and tested with a large dataset to ensure accuracy. Additionally, a user-friendly interface will be developed for healthcare professionals to upload images and receive quick DR screening results. By enabling early detection, the system can support timely interventions, reducing irreversible vision loss risks in diabetic patients.

1.2 IMPORTANCE/RELEVANCE OF THE PROJECT

A diabetic retinopathy detection system is crucial for preventing early vision loss and offering preventive measures.it enhances images, leverages transfer learning for feature extraction, and uses a custom model for classification. The system offers a user-friendly interface for quick dr screening by healthcare professionals, aiding in early intervention and reducing irreversible vision loss risks in diabetic patients. This technology not only preserves vision but also streamlines healthcare processes, integrates advanced diagnostic tools, reduces costs, and contributes to ongoing medical ai research and innovation.

Chapter 2

LITERATURE SURVEY

A literature survey is an essential part of any research paper or project, where the author reviews and summarizes existing relevant literature and research studies related to the topic of interest. The purpose of a literature survey is to provide context, background information, and a comprehensive understanding of the current state-of-the-art in the field.

2.1 A Deep Learning Ensemble Approach For Diabetic Retinopathy Detection [1]

This paper focuses on developing a deep learning ensemble approach for the detection and classification of diabetic retinopathy (DR) stages in colour fundus images. It addresses the limitations of existing methods in accurately identifying early stages of DR, which is crucial for timely treatment and prevention of vision loss. The authors utilize a combination of five deep convolutional neural network (CNN) models trained on a kaggle dataset of retina images to improve classification accuracy. The proposed ensemble model outperforms state-of-the-art methods and demonstrates the potential for further enhancing early-stage detection accuracy through specific model training and ensemble techniques.

2.2 Deep Learning Approach To Diabetic Retinopathy Detection [2]

In this paper a deep learning approach using an ensemble of three CNN architectures for automatic detection of the stage of diabetic retinopathy from fundus images. Transfer learning is employed to enhance generalization and reduce variance, resulting in high and stable performance. Future work includes extending the method with SHAP calculation for the entire ensemble and exploring meta-learning with the models.

2.3 Diabetic retinopathy detection using deep convolutional neural networks [3]

This research paper presents the design, architecture, and implementation of deep convolutional neural networks (CNNs) for automatic detection and classification of diabetic retinopathy from colour fundus retinal images. The paper discusses the use of the quadratic kappa metric to evaluate prediction results and showcases the ensemble of three major CNN models to achieve the best score of 0.3996 in diabetic retinopathy classification.

2.4 Automated detection of diabetic retinopathy using machine learning classifiers [4]

The research paper focuses on utilizing machine learning classifiers for the automated detection of diabetic retinopathy (DR). The study conducted in Almajmaah, Saudi Arabia, involved 327 diabetic patients and explored various machine learning algorithms, highlighting ranger random forest as the top performer in DR classification, achieving an accuracy of 86% on test data. The paper emphasizes the importance of integrating ophthalmology and machine learning for enhanced disease diagnosis and clinical decision-making support.

2.4 A lightweight Diabetic Retinopathy Detection Model Using a Deep-Learning Technique [5]

This paper focuses on a a lightweight deep-learning model which is used for diabetic retinopathy (DR) detection, leveraging image pre-processing, yolo v7 for feature extraction, and a tailored quantum marine predator algorithm for feature selection. The model, based on mobilenet v3, achieves high accuracy and f1 scores on the aptos and eyepacs datasets, demonstrating its efficiency for remote healthcare applications.

Chapter 3

PROBLEM STATEMENT

Diabetic retinopathy (DR) remains a significant cause of blindness among working-age adults globally, highlighting the critical need for early detection and intervention. Traditional methods of DR screening often lack efficiency and timeliness, leading to delays in diagnosis and subsequent management. Healthcare professionals face challenges in accurately categorizing DR stages from retinal images, which can result in missed opportunities for early intervention and increased risks of irreversible vision loss in diabetic patients. There is a pressing need for an automated system that can effectively preprocess retinal images, extract relevant features, and classify images into distinct DR stages with high accuracy. Such a system must also provide a seamless and user-friendly interface for healthcare professionals to facilitate efficient DR screening, enabling timely interventions and reducing vision loss risks in diabetic individuals.

OBJECTIVES

- Develop a user-friendly web application for uploading retinal images for DR screening.
- Integrate a pre-trained machine learning model for classifying retinal images into different DR severity levels.
- Enable real-time processing of uploaded images to provide immediate DR predictions.
- Design an intuitive interface accessible to users with varying technical proficiency.
- Optimize the ML model for high accuracy and sensitivity, minimizing false positives/negatives.
- Provide educational resources within the application to raise awareness about DR and its risk factors.
- Implement robust data privacy and security measures to protect user data.
- Conduct evaluations to assess the clinical utility and effectiveness of the system in supporting early DR detection and improving access to screening services.

Chapter 4
SOFTWARE AND HARDWARE REQUIREMENTS

Hardware requirements:

- 1.6hz or faster processor
- At least 8 GB of RAM

Software requirements:

- Programming Language: Python

Web Development Framework:

- Flask

Integrated Development Environment (IDE):

- Google collab
- PyCharm

Web Browser:

- Google chrome, Mozilla Firefox or Microsoft Edge

Operating system:

- Windows or Linux

Chapter 5

METHODOLOGY

Methodology refers to the systematic, structured approach or set of procedures used to conduct research or solve a problem in a scientific or systematic manner. It outlines the steps, techniques, tools, and procedures employed to gather data, analyze information, and draw conclusions in a research study or project. A well-defined methodology is crucial as it ensures the validity, reliability, and replicability of research findings.

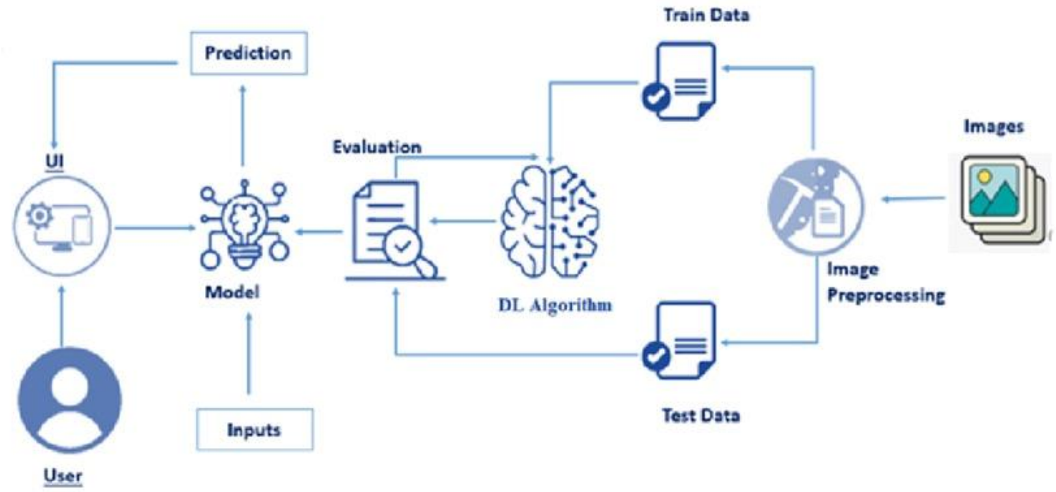

Figure 1: System Diagram

5.1 Dataset and Kaggle API

The dataset, found on Kaggle, is 10GB in size. It contains 4,396 images categorized into five classes. These classes are: "No Diabetic Retinopathy", "Mild DR", "Moderate DR", "Severe DR", and "Proliferative DR". The dataset is split into two parts for training and testing purposes, with 3,662 images for training and 734 for testing.

The dataset used in this project is relatively large, approximately 10GB in size. To facilitate efficient data loading and avoid potential issues with manually downloading and uploading such a large dataset, the Kaggle API was utilized. The Kaggle API provides a convenient way to access and download datasets directly from the Kaggle platform within the code environment. This approach ensures seamless data retrieval and eliminates the need for manual data handling, which can be cumbersome and time-consuming for datasets of this size.

5.2 Transfer Learning and Convolutional Neural Network (CNN)

The code utilizes transfer learning by leveraging the pre-trained Xception model from the TensorFlow Keras Applications. The Xception model is a deep convolutional neural network trained on the ImageNet dataset for image classification tasks. The pre-trained weights of the Xception model are loaded, and the top layers are replaced with custom layers for the specific task of diabetic retinopathy detection. Transfer learning is a popular and effective technique in deep learning, especially when dealing with limited data or computational resources. By leveraging a pre-trained model like Xception, the model can benefit from the knowledge learned on a large dataset (ImageNet), reducing the need for extensive training from scratch and potentially achieving better performance with fewer training examples. The Xception model is a CNN architecture that automatically learns relevant features from the input images. CNNs are highly effective for image classification tasks due to their ability to capture spatial and local patterns in the data. CNNs are the state-of-the-art models for image classification tasks and are well-suited for extracting hierarchical features from images, making them an excellent choice for the diabetic retinopathy detection task, which relies on identifying patterns and abnormalities in retinal images.

5.3 Dense Layers

We added a Flatten layer to the output of the Xception model, followed by a Dense layer with a softmax activation function. This Dense layer serves as the classification layer, mapping the extracted features from the CNN to the desired output classes (in this case, five classes representing different levels of diabetic retinopathy). The added Dense layer with a softmax activation function is a common approach for multi-class classification problems. It maps the extracted features from the CNN to the desired output classes, allowing the model to make predictions about the level of diabetic retinopathy.

5.4 Data Augmentation:

Then we employed data augmentation techniques using the ImageDataGenerator class from TensorFlow Keras. Specifically, it applies rescaling, shearing, zooming, and horizontal flipping to the training data. Data augmentation helps to increase the diversity of the training data and can improve the model's generalization ability. Data augmentation techniques help to increase the diversity and variation of the training data, which can improve the model's generalization performance and prevent overfitting. These techniques are particularly useful when working with limited training data, as is often the case in medical imaging applications.

5.5 Model Training: The model is compiled with categorical cross-entropy as the loss function and the Adam optimizer. The training process involves feeding the augmented training data and validation data to the model using the fit() function. The model is trained for a specified number of epochs (18 in the provided code). The categorical cross-entropy loss function is a suitable choice for multi-class classification problems, as it measures the performance of the model in correctly classifying instances into multiple classes. The Adam optimizer is a popular choice for deep learning models due to its computational efficiency and ability to handle sparse gradients.

5.6 Flask Web Application:

For deployment and user interaction, a Flask-based web application was developed using PyCharm as the IDE. Flask is a lightweight Python web framework that allows for the creation of web applications and APIs.

This web application provides a user interface where users can upload retinal images, which are then processed by the trained model for diabetic retinopathy detection and classification. Flask is a lightweight and flexible web framework that is well-suited for building web applications and APIs. It offers simplicity and ease of use, making it an ideal choice for creating a user-friendly interface for the diabetic retinopathy detection system.

Chapter 6

IMPLEMENTATION

The implementation of the diabetic retinopathy detection model involves several key steps, which are discussed in detail below.

6.1. Loading Dataset and Preprocessing

First, we import the necessary libraries, such as TensorFlow for deep learning, Pandas for data manipulation, ZipFile and BytesIO for handling compressed files, and the Google Colab files module for uploading and managing files in the Colab environment.

Then using our Kaggle API token present in the json file we load the diabetic-retinopathy-level-detection dataset from Kaggle. We created a directory where the zipped dataset is loaded, then it is unzipped to extract the entire 10GB dataset.

```
[13] !kaggle datasets download -d arbethi/diabetic-retinopathy-level-detection

     Dataset URL: https://www.kaggle.com/datasets/arbethi/diabetic-retinopathy-level-detection
     License(s): CC0-1.0
     Downloading diabetic-retinopathy-level-detection.zip to /content/kaggle
     100% 9.65G/9.66G [01:56<00:00, 103MB/s]
     100% 9.66G/9.66G [01:56<00:00, 88.9MB/s]
```

Fig 2: Downloading Dataset From Kaggle Using Kaggle API

Before proceeding with model development, the code specifies the image size of 299 x 299 pixels for the model input. This image size is compatible with the pre-trained Xception model, which will be used as the base for the diabetic retinopathy detection model. The paths for the training and testing datasets are defined as well.The pre-trained Xception model from TensorFlow is then loaded, and its weights are frozen by setting layer.trainable = False for all layers. This step ensures that the pre-trained weights are not updated during the training process, and only the newly added layers will be trained**.**

```
xception = Xception(input_shape=imageSize + [3], weights='imagenet',include_top=False)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83683744/83683744 [==============================] - 1s 0us/step
```

Fig 3: Loading the Xception Model

6.2. Model Architecture

Then we defined the additional layers to be added on top of the pre-trained Xception model. First, the output of the Xception model is flattened using the Flatten layer. This step is necessary because the output of the pre-trained model is a multi-dimensional tensor, and the subsequent layers expect a one-dimensional input.

Next, a Dense layer with 5 output units and a softmax activation function is added. The softmax activation function is commonly used for multi-class classification problems, which is suitable for the diabetic retinopathy detection task, where the output can be one of five possible classes (e.g., no retinopathy, mild, moderate, severe, or proliferative).

The model object is then created by specifying the input (Xception model input) and output (the newly added Dense layer) layers using the Model class from TensorFlow.

Finally, the model is compiled with the categorical cross-entropy loss function, Adam optimizer, and accuracy metric. The categorical cross-entropy loss is commonly used for multi-class classification problems, and the Adam optimizer is a popular choice for its efficiency and ability to handle sparse gradients.

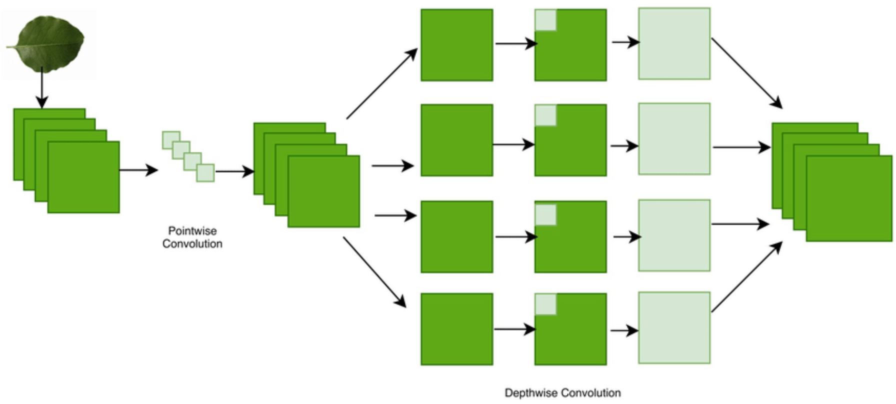

Fig 4: Xception Model Architecture

6.3. Data Augmentation

To increase the diversity of the training data and improve the model's generalization capability, data augmentation techniques are applied. We created two data generator objects: one for the training dataset (train_datagen) and another for the testing dataset (test_datagen).

The training data generator applies various augmentation techniques, such as rescaling (dividing pixel values by 255), shearing (applying a shear transformation), zooming (randomly zooming the images), and horizontal flipping (randomly flipping the images horizontally). These augmentation techniques help the model learn invariance to different transformations and variations in the input data.

The testing data generator only applies rescaling, as it is essential to maintain the integrity of the testing data for accurate evaluation.

6.4. Model Training and Saving

With the data generators set up, the code loads the training and testing datasets using the flow_from_directory method of the data generators. This method automatically loads the images from the specified directories and applies the specified augmentation techniques (if any) during the training process.

The model is then trained using the model.fit() function, which takes the following parameters:

training_set: The training data generator object.

validation_data: The testing data generator object, used for validation during training.

epochs: The number of iterations over the entire training dataset (set to 20 in this case).

steps_per_epoch: The number of batches of data to be processed in each epoch (set to the length of the training set).

validation_steps: The number of batches of data to be processed for validation in each epoch (set to the length of the testing set).

After training, the code saves the trained model to a file named ibmdr.h5 using the model.save() function. This step allows the trained model to be loaded and used for inference or prediction later, without the need for retraining.

6.5. Creating a User Interface

Next, we created a Flask web application enabling users to interact with the pre-trained diabetic retinopathy detection model. Upon uploading an image via the interface, the image is processed, and a prediction is made using the loaded model. The predicted result is then displayed to the user. The application consists of routes for the home page, an about page, an index page, and a result page. The result page handles image uploads, preprocesses the image for prediction, makes a prediction using the model, and renders the prediction result back to the user.

Chapter 7

RESULTS

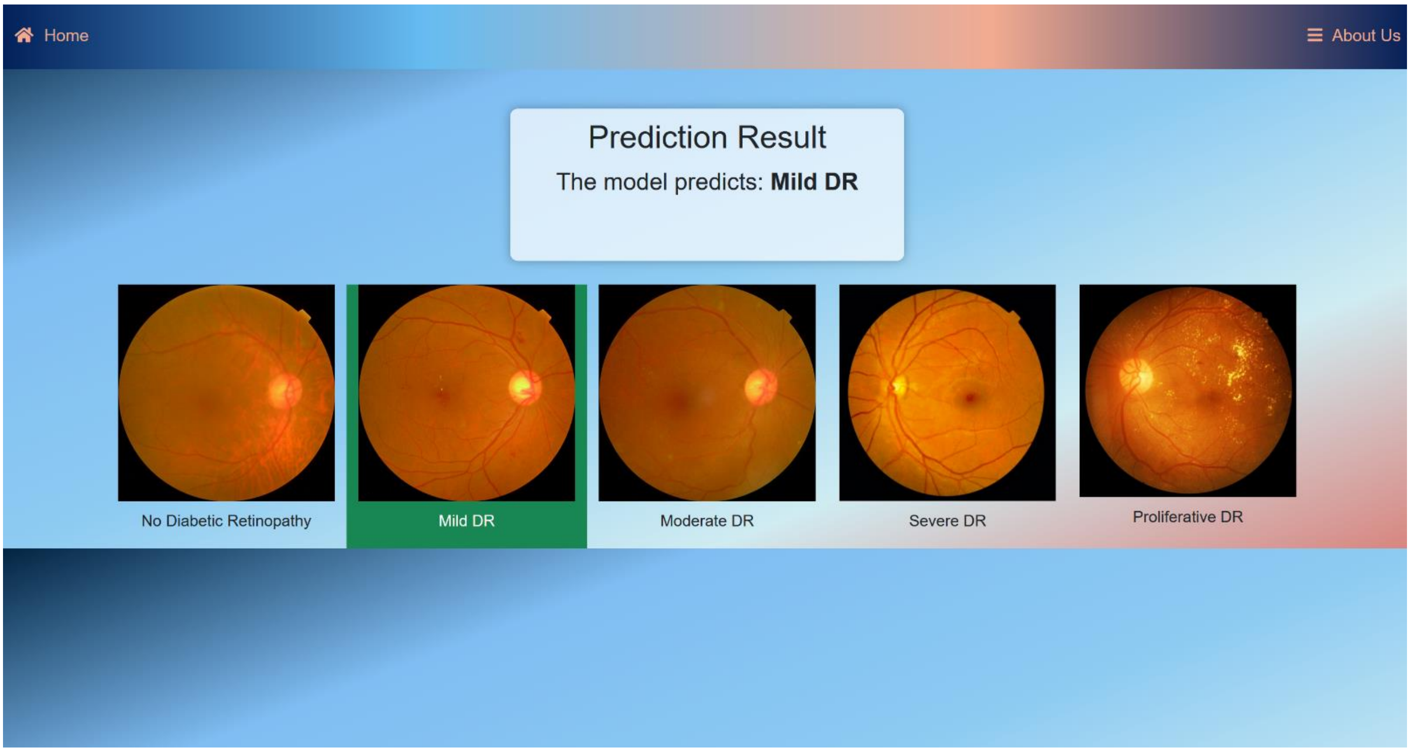

Figure 5: This page will give the predicted stage of DR

```
Epoch 1/18
115/115 [==============================] - 572s 5s/step - loss: 4.3950 - accuracy: 0.6554 - val_loss: 2.4889 - val_accuracy: 0.7575
Epoch 2/18
115/115 [==============================] - 545s 5s/step - loss: 3.7705 - accuracy: 0.7133 - val_loss: 3.3188 - val_accuracy: 0.7112
Epoch 3/18
115/115 [==============================] - 547s 5s/step - loss: 3.9174 - accuracy: 0.7288 - val_loss: 2.4390 - val_accuracy: 0.7480
Epoch 4/18
115/115 [==============================] - 551s 5s/step - loss: 3.5029 - accuracy: 0.7471 - val_loss: 1.8136 - val_accuracy: 0.7847
Epoch 5/18
115/115 [==============================] - 556s 5s/step - loss: 2.7892 - accuracy: 0.7635 - val_loss: 2.0840 - val_accuracy: 0.8106
Epoch 6/18
115/115 [==============================] - 550s 5s/step - loss: 2.9793 - accuracy: 0.7695 - val_loss: 3.6408 - val_accuracy: 0.7112
Epoch 7/18
115/115 [==============================] - 554s 5s/step - loss: 3.2032 - accuracy: 0.7758 - val_loss: 1.9877 - val_accuracy: 0.8270
Epoch 8/18
115/115 [==============================] - 550s 5s/step - loss: 2.5736 - accuracy: 0.8086 - val_loss: 3.8747 - val_accuracy: 0.6948
Epoch 9/18
115/115 [==============================] - 552s 5s/step - loss: 2.7630 - accuracy: 0.7979 - val_loss: 3.5623 - val_accuracy: 0.7425
Epoch 10/18
115/115 [==============================] - 550s 5s/step - loss: 2.8998 - accuracy: 0.7987 - val_loss: 2.2882 - val_accuracy: 0.8093
Epoch 11/18
115/115 [==============================] - 547s 5s/step - loss: 2.5975 - accuracy: 0.8056 - val_loss: 3.7630 - val_accuracy: 0.8065
Epoch 12/18
115/115 [==============================] - 543s 5s/step - loss: 2.4061 - accuracy: 0.8154 - val_loss: 1.9302 - val_accuracy: 0.8379
Epoch 13/18
115/115 [==============================] - 544s 5s/step - loss: 2.7107 - accuracy: 0.8091 - val_loss: 2.6957 - val_accuracy: 0.8038
Epoch 14/18
115/115 [==============================] - 549s 5s/step - loss: 2.4436 - accuracy: 0.8233 - val_loss: 2.0552 - val_accuracy: 0.8392
Epoch 15/18
115/115 [==============================] - 541s 5s/step - loss: 2.7371 - accuracy: 0.8162 - val_loss: 2.8302 - val_accuracy: 0.7956
115/115 [==============================] - 541s 5s/step - loss: 2.7371 - accuracy: 0.8162 - val_loss: 2.8302 - val_accuracy: 0.7956
Epoch 16/18
115/115 [==============================] - 598s 5s/step - loss: 2.3349 - accuracy: 0.8362 - val_loss: 2.2705 - val_accuracy: 0.8011
Epoch 17/18
115/115 [==============================] - 539s 5s/step - loss: 1.9826 - accuracy: 0.8514 - val_loss: 2.4027 - val_accuracy: 0.8270
Epoch 18/18
115/115 [==============================] - 541s 5s/step - loss: 2.0581 - accuracy: 0.8474 - val_loss: 2.2299 - val_accuracy: 0.8229
```

Figure 2: Model Training and Accuracy

CONCLUSION

In conclusion, the development of an automated system for diabetic retinopathy (DR) detection using deep learning techniques represents a significant advancement in modern healthcare. Through the integration of image preprocessing, transfer learning, and custom classification models, our project addresses the critical need for early detection and intervention in dr, a leading cause of blindness among working-age adults globally.

The system's ability to accurately categorize retinal images into distinct DR stages - mild, moderate, severe, and proliferative - enables healthcare professionals to make informed decisions promptly, leading to timely interventions and reduced risks of irreversible vision loss in diabetic patients. The user-friendly interface further streamlines the screening process, empowering healthcare providers with efficient tools for dr diagnosis and management.

By leveraging advanced technologies such as deep learning and image analysis, our project not only improves patient outcomes but also contributes to ongoing research and innovation in medical artificial intelligence (AI). The project's success underscores the importance of interdisciplinary collaboration between healthcare professionals, data scientists, and software engineers in tackling complex healthcare challenges with cutting-edge solutions.

Looking ahead, continued refinement and validation of the system using diverse datasets and real-world clinical scenarios will further enhance its accuracy, reliability, and scalability. The deployment of such automated dr detection systems in healthcare settings holds immense promise for improving healthcare accessibility, reducing healthcare costs, and ultimately, preserving vision and enhancing quality of life for diabetic individuals worldwide.

REFERENCES

[1]	A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection (https://ieeexplore.ieee.org/document/8869883) – Ensemble Methods
[2]	Deep Learning Approach To Diabetic Retinopathy Detection https://arxiv.org/pdf/2003.02261 - Deep Learning Approach
[3]	Diabetic retinopathy detection using deep convolutional neural networks (https://ieeexplore.ieee.org/document/7914977) – Convolutional Neural Networks
[4]	Automated detection of diabetic retinopathy using machine learning classifiers (https://pubmed.ncbi.nlm.nih.gov/33577010/) - Machine Learning Classifiers
[5]	A lightweight Diabetic Retinopathy Detection Model Using a Deep-Learning Technique (https://pubmed.ncbi.nlm.nih.gov/37835861/) – Deep Learning Techniques

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MINI PROJECT LAB(21CS605)

WEEKLY PROGRESS REPORT

DATE	DESCRIPTION
21/02/2024-27/02/2024	• Conducted thorough research on diabetic retinopathy, understanding the causes and symptoms. • Acquired knowledge of Flask, setting the stage for developing the application
28/02/2024-05/03/2024	• Data Collection : Researched and obtained a suitable dataset for our project from Kaggle, focusing on diabetic retinopathy. • Dataset Link : https://www.kaggle.com/datasets/arbethi/diabetic-retinopathy-level-detection
06/03/2024-12/03/2024	• Data Preparation: utilized the kaggle API to download the dataset in google collab to overcome size limitations.
13/03/2024-12/03/2024	• We developed a basic deep learning model for diabetic retinopathy detection having seven epochs.
20/03/2024-27/03/2024	• Trained the deep learning model for diabetic retinopathy detection having 18 epochs and checked the variation in accuracy.
28/03/2024-03/04/2024	• Developed a basic frontend using flask • Worked on home page creation, prediction page as well as an upload page that for users to easily upload their retina images
04/04/2024-10/04/2024	• Added more features to the website like about us page using CSS and JavaScript