VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 018, KARNATAKA, INDIA

A

PROJECT REPORT

ON

"DIABETIC RETINOPATHY"

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

HAREESH NAIK	**1JS19CS061**
RAKSHITH B G	**1JS19CS132**
YASHAS K M	**1JS19CS190**

Under the guidance of

Mrs. Rashmi B N

Assistant Professor

Department of Computer Science and Engineering

JSSATE, Bengaluru

JSS ACADEMY OF TECHNICAL EDUCATION, BENGALURU

Department of Computer Science and Engineering

2022 – 2023

JSS MAHAVIDYAPEETHA, BENGALURU

JSS ACADEMY OF TECHNICAL EDUCATION

JSS Campus, Uttarahalli-Kengeri Main Road, Bengaluru - 560060

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Project work entitled **"DIABETIC RETINOPATHY"** is a bonafide work carried out by **Mr. RAKSHITH B G (1JS19CS132), Mr. YASHAS K M (1JS19CS190) and Mr. HAREESH NAIK (1JS19CS061)** in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the academic year 2022 - 2023. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report. The project has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Mrs. Rashmi B N .
Assistant Professor
Dept. of CSE JSSATE, Bengaluru

Dr. Mallikarjuna P B
Professor & Head
Dept. of CSE JSSATE, Bengaluru

Dr. Bhimasen Soragon
Principal
JSSATE, Bengaluru

External Viva

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

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So, with gratitude, we acknowledge all those whose guidance and encouragement crowned my effort with success.

First and foremost, we would like to express our pranamas and gratitude to his **Holiness Jagadguru Sri Sri Shivarathri Deshikendra Mahaswamiji- galavaru** and our beloved principal **Dr. Bhimsen Soregaon** for providing us an opportunity to carry out the major project as a part of our curriculum in thepartial fulfillment of the degree course.

We express our sincere gratitude for our beloved Professor & Head of the department, **Dr. Mallikarjuna P B**, for his co-operation and encouragement at all the moments of our approach.

It is our pleasant duty to place on record our deepest sense of gratitude to our respected guide **Mrs. Rashmi B N,** Assistant Professor**,** for the constant encouragement, valuable help and assistance in every possible way.

We would like to thank all **Teaching** and **non-teaching Faculty** for providingus with their valuable guidance and for being there at all stages of our work.

HAREESH NAIK (1JS19CS061)
RAKSHITH B G (1JS19CS132)
YASHAS K M (1JS19CS190)

… # ABSTRACT

A chronic and irreversible eye condition called diabetic retinopathy affects both vision and quality of life. In this study, we create a convolutional neural network-based deep learning (DL) architecture for automated Diabetic Retinopathy detection. Systemic deep learning,

For diagnostic purposes, techniques like convolutional neural networks (CNNs) can infer a hierarchical representation of pictures to distinguish between Diabetic Retinopathy and non-Diabetic Retinopathy patterns. Six learnt layers—four convolutional layers and two fully-connected layers—make up the proposed DL architecture. To further improve the effectiveness of Diabetic Retinopathy diagnosis, dropout and data augmentation procedures are used. The main cause of blindness among people in their working years worldwide is Diabetic Retinopathy.

Due to the complex grading system and the requirement that skilled doctors detect the existence and relevance of numerous tiny characteristics, diagnosing Diabetic Retinopathy with colour fundus pictures is a challenging and time-consuming task. Here, we provide a CNN method for Diabetic Retinopathy diagnosis. We create a network with a Convolutional Neural Network (CNN) architecture and data augmentation that can recognise the complex elements needed for the classification challenge, like tiny aneurysms, exudate, and haemorrhages on the retina.

Diabetes often leads to diabetic retinopathy, an illness that damages the retina, the light-sensitive tissue at the back of the eye. It occurs when brought on by inflammation of the retina's blood vessels, which, if unabated, can result in blindness and vision loss. A degenerative condition, diabetic retinopathy can range in severity from mild to severe. To stop or stop the disease's course, early detection and treatment are crucial. Regular eye checkups, blood sugar, blood pressure, and cholesterol control, as well as potential interventions like surgery or laser treatment, are all crucial to keeping track of diabetic retinopathy. Certainly! Millions folks worldwide the world grapple with the a fatal the wake of diabetes known as diabetic retinopathy. An open source Kaggle dataset is used as an input for Diabetic Retinopathy. Total number of 2664 images are used in Diabetic Retinopathy and the testing accuracy for Diabetic Retinopathy is 84%.

Table of Contents

LIST OF FIGURES

CHAPTER 1

INTRODUCITON

1.1 Overview

Millions of people throughout the world suffer from diabetes, which has several dangerous complications, including diabetic retinopathy. High blood sugar levels harm the tiny blood vessels in the retina, the light-sensitive tissue in the back of the eye, resulting in this disorder. If left untreated, this can eventually result in blindness and vision loss. A degenerative condition, diabetic retinopathy can range in severity from mild to severe. To stop or decrease the development of diabetic retinopathy, patients with diabetes must control their blood sugar levels and other risk factors.

In order to manage this condition and protect vision, early detection and treatment are also essential. In this situation, it's critical to comprehend the origins, signs, risks, and treatment options for diabetic retinopathy. You should also schedule routine eye exams to identify and treat the illness as early as feasible.

There are two types of diabetic retinopathy

1. Non-proliferative diabetic retinopathy (NPDR)
2. Proliferative diabetic retinopathy (PDR)

1. The early stage of diabetic retinopathy, known as non-proliferative diabetic retinopathy (NPDR), is brought on by destruction to the retina's small blood vessels. At this point, the retina may expand as a result of blood or fluid leaking from the blood vessels into the retina. Vision loss or blurriness may result from this.

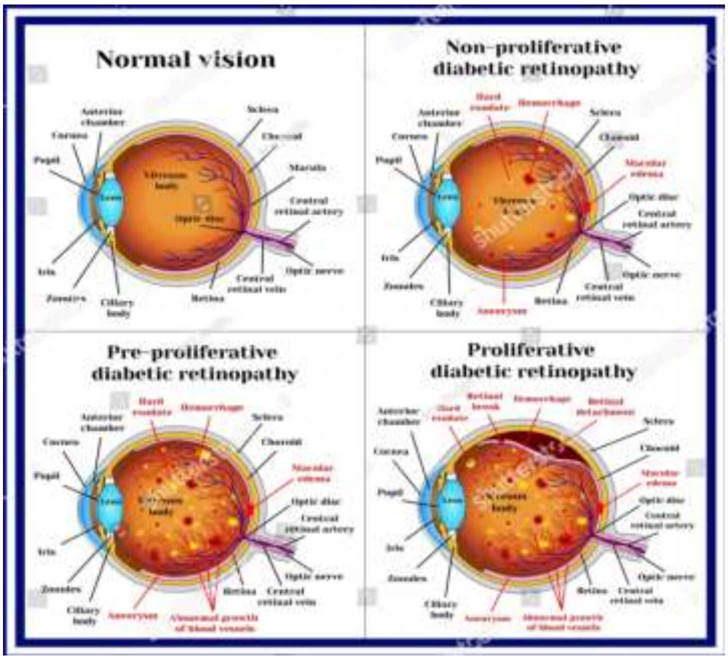


Fig 1.1: Types of Diabetic Retinopathy

The optic nerve of your eye is harmed by the illness known as Diabetic Retinopathy, which worsens over time. It frequently has to do with an increase in pressure inside your eye. Diabetic Retinopathy frequently runs in families and may not manifest until later in life. The optic nerve, which sends images to the brain, might be harmed by the elevated pressure, also known as intraocular pressure. Diabetic Retinopathy can result in irreversible vision loss if the damage is not treated. Diabetic Retinopathy, if left untreated, can result in total and irreversible blindness within a few years.

1.2 Existing System

Diabetic Retinopathy is frequently diagnosed too late because it usually goes unnoticed for years. In fact, even in countries with high sociosanitary standards, half of cases have moderate to advanced disease in the worse eye at the time of first presentation, which significantly increases the disease's financial and human toll on both the individual and the community. Measurement of intraocular pressure (IOP) is frequent during routine eye exams, although it cannot distinguish between normal.

Diabetic eyes: Many ocular hypertensives do not require treatment and will never develop Diabetic Retinopathy in their lifetimes. Up to 50% of Diabetic Retinopathy subjects may not show an increased IOP upon evaluation. According to several epidemiological studies (Baltimore Eye Survey, Rotterdam Eye Study, Blue Mountains Eye Study, Visual Impairment Project, Proyecto VER, and Latino Eye Study), Diabetic Retinopathy is not diagnosed in 50% of cases in the western world, with higher rates in certain ethnic groups, and in up to 90% of cases in developing countries (Aravind Eye Study). Diabetic Retinopathy, on the other hand, is ironically frequently overtreated: Many people receive treatment despite the fact that they are asymptomatic. This fervently urges a global improvement in the disease's precise detection.

The Early Treatment Diabetic Retinopathy Study (ETDRS) classification system serves as the foundation for the current approach for the diagnosis and grading of diabetic retinopathy (DR). This approach evaluates the severity of diabetic retinopathy using colour fundus pictures of the retina.

The ETDRS method assigns a score to various forms of diabetic retinopathy, ranging from 0 (no diabetic retinopathy) to 4 (the most severe form). The grades are determined by the degree of neovascularization, exudates, cotton wool patches, haemorrhages, and microaneurysms. In general, regular eye exams, glucose control, blood pressure management, and, occasionally, treatment with drugs or surgery are used to diagnose and treat diabetic retinopathy.

1.3 Problem Statement

The project's goal is to create a convolutional neural network for the recognition of deep learning for Diabetic Retinopathy. For Diabetic Retinopathy, an open source RIGA dataset is utilised. The eye ailment Diabetic Retinopathy, often known as intraocular pressure, damages the visual nerve. The eye's internal pressure increased. Our goal is to create a convolutional neural network with the following layers: convolution, max pooling, rectified linear unit (ReLU), and fully connected.

The majority of research in the field of Diabetic Retinopathy has been centred on illness detection or manually extracting features, however this paper attempts to use deep learning to diagnose the disease into its many stages. in order to increase productivity.

The dataset contains images of 2 classes as follows:

- class 0 for Normal
- class 1 for Mild
- class 2 for Moderate
- class 3 for Proliferate
- class 4 for severe

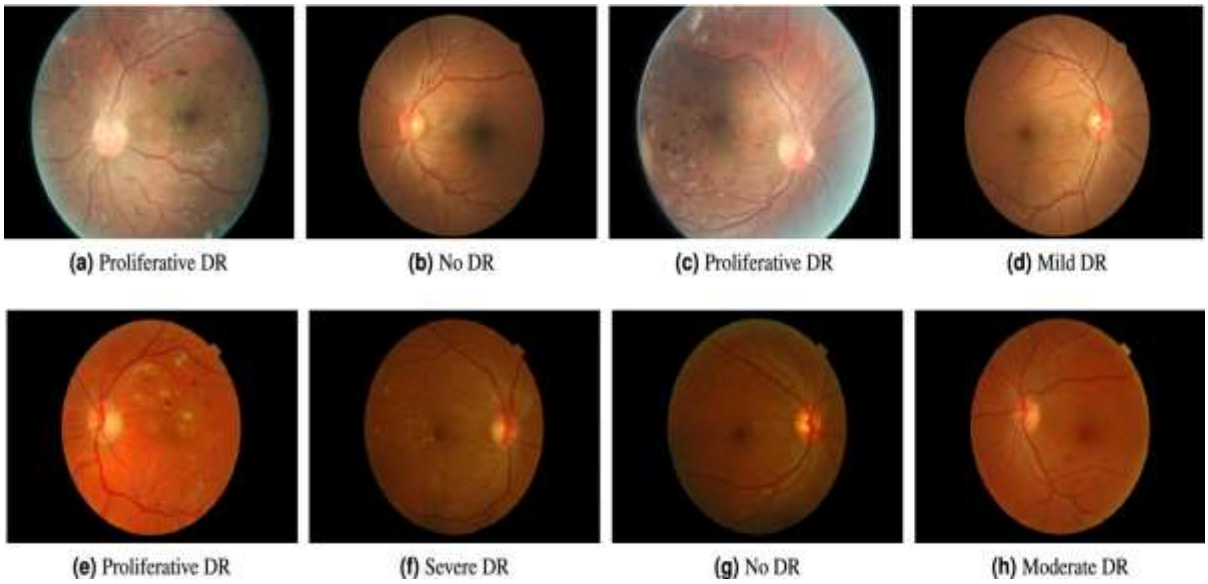


Fig 1.2: Classification of Diabetic Retinopathy

The majority of research in the field of Diabetic Retinopathy has been centred on illness detection or manually extracting features, however this paper attempts to use deep learning to diagnose the disease into its many stages. in order to increase productivity.

1.4 Motivations for Project:

Prevalence: Millions of people worldwide suffer from diabetes, and diabetic retinopathy is the main cause of blindness in working-age individuals.

Impact: The quality of life and ability to work of a person can be significantly impacted by vision loss and blindness caused by diabetic retinopathy.

Treatment options: There are treatments for diabetic retinopathy, but they are not always successful, and there is a need for more successful and widely available treatments. Early detection: Early diagnosis of diabetic retinopathy can stop or slow the disease's course and lower the chance of visual loss. For the purpose of identifying those who are at risk of developing diabetic retinopathy, new screening measures are required.

1.5 Proposed System

A disorder of the eyes called Diabetic Retinopathy can proceed to total blindness. Diabetic Retinopathy is a chronic condition whose development can only be prevented if precisely identified at an early stage. The proposed method offers a computer-aided automated Diabetic Retinopathy detection system that enables ophthalmologists to accurately and quickly diagnose Diabetic Retinopathy patients.

An algorithm uses a fundus image that has already been processed to extract the optic disc and cup, then calculates the Convolution Neural Network (CNN). To train and evaluate the classifier, intensity and textural information are taken from the image. The classification of the image as Diabetic Retinopathy, non-Diabetic Retinopathy, or suspect is the result of the combination of characteristics and CNN for Diabetic Retinopathy detection.

1.6 Advantages of Proposed System

- This model is more accurate than previous models, which leads to better performance.
- The temporal complexity has been significantly reduced.
- Patients or their Parents can take the test whenever they choose, without needing to visit a hospital.
- It is cost-effective for any of the scans performed.
- The number of employees on the work might be drastically reduced.

CHAPTER 2
LITERATURE SURVEY

2.1 Lee - Prevalent insomnia Concerns.

The title of Lee et al.'s article from 2020 is "Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images from Multiethnic Populations with Diabetes." The retinal pictures from multiethnic populations with diabetes are used in this work to propose a deep learning system for the diagnosis of diabetic retinopathy (DR) and related eye illnesses. The incidence of DR and the significance of early detection and treatment are discussed by the authors in the opening paragraphs. Additionally, they go over the drawbacks of the available diagnostic tools and the potential advantages of using deep learning algorithms to increase precision and effectiveness.

The convolutional neural network (CNN) architecture used by the proposed deep learning system is trained on a sizable dataset of retinal pictures from numerous ethnic communities with diabetes. In order to broaden the dataset's diversity and quantity and boost the model's effectiveness, the authors additionally employ data augmentation approaches. With an area under the curve (AUC) of 0.975 for DR detection and 0.991 for referable eye disorders, the study's findings demonstrate that the suggested deep learning system obtained great accuracy in detecting DR and related eye ailments. The system's applicability to patients who are African Americans, Hispanic, and Asian is also demonstrated by the authors.

2.2 Mottet – Diabetic Retinopathy detection using Deep Learning.

The authors start off by going over the frequency, significance, and limitations of DR as well as the existing diagnostic techniques. The authors then conduct a literature review of the state-of-the-art deep learning-based DR detection systems, concentrating on the various deep learning architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid architectures.

The authors also look at the various retinal imaging modalities used for training and testing, such as fluorescein angiography images, optical coherence tomography (OCT) images, and cooler fundus photographs. Each image type's benefits and drawbacks are discussed, as well as how they affect the effectiveness of the deep learning models.

The systematic review's findings demonstrate the great accuracy and sensitivity of deep learning-based DR detection systems, with some studies obtaining AUC values of above 0.99. The authors also draw attention to the potential advantages of applying deep learning algorithms, including lessening the workload for ophthalmologists, expanding the reach of screening programs, and lowering healthcare expenses. The requirement for vast and varied datasets, the possibility of bias and overfitting, and the lack of transparency and interpretability of deep learning models are just a few of the difficulties and restrictions that the authors also mention with regard to deep learning-based DR detection systems.

2.3 Hua-Deep learning based Automated diagnosis of DR using fundus images.

The relevance of DR screening and the possibility for deep learning algorithms to increase screening efficiency and accuracy are covered by the authors in the opening paragraphs. Then, focusing on the various deep learning architectures, including CNNs and hybrid architectures, they review the most recent research on deep learning-based DR diagnosis using fundus images.

The systematic review and meta-analysis findings demonstrate the high accuracy and sensitivity of deep learning-based automated diagnosis of DR using fundus pictures, with some studies reporting AUC values of > 0.98. The authors also draw attention to the potential advantages of applying deep learning algorithms, including lessening the workload for ophthalmologists, expanding the reach of screening programs, and lowering healthcare expenses.

The necessity for vast and varied datasets, the possibility of bias and overfitting, and the lack of transparency and interpretability of deep learning models are just a few of the difficulties and restrictions that the authors also mention with regard to deep learning-based DR diagnosis system.

2.4 Marques- Detection of ocular disease.

The necessity of early DR detection and treatment as well as the shortcomings of current diagnostic techniques are covered by the writers in the opening paragraphs. Then, focusing on the various deep learning architectures, including CNNs, RNNs, and hybrid architectures, they review the most recent research on deep learning-based DR detection systems. The authors also look at the many retinal imaging modalities, such as color fundus photos, OCT images, and others, that are utilized for training and assessment. Each image type's benefits and drawbacks are discussed, as well as how they affect the effectiveness of the deep learning models.

The results of the review show that deep learning-based DR detection systems have achieved high accuracy and sensitivity, with some studies reporting AUC values of over 0.99. The authors also highlight the potential benefits of using deep learning algorithms, such as reducing the burden on ophthalmologists, improving screening coverage, and reducing healthcare cost.

2.5 Kim-Efficient Classification of DIR using dual scale lesion aggregation with deep convolutional network.

In order to train their model using a transfer learning strategy, the authors use a sizable dataset of retinal pictures with DR grades. Two branches—one for the fine-grained scale and the other for the coarse-grained scale—make up the DCNN model. Lesion-level local features are extracted by the fine-grained branch, whilst image-level global features are recorded by the coarse-grained branch. The two branches' collected characteristics are then integrated with a fully linked layer and divided into several DR grades.

The authors also look at the various kinds of retinal pictures utilized for training and testing, such as color fundus photos, OCT images, and other imaging modalities. They talk about each image type's benefits and drawbacks as well as how they affect the effectiveness of deep learning models.

Overall, the study by Kim et al. (2021) offers a promising strategy for the quick and accurate categorization of DR using a deep convolutional neural network and a dual-scale lesion aggregation method. The model is a promising tool for assisting doctors in the diagnosis and management of DR due to its interpretability and performance.

CHAPTER 3

REQUIREMENTS

3.1 Functional Requirements

- To find Diabetic Retinopathy, a dataset of roughly 2500 records was gathered from the internet andother sources.
- Some of the attributes in this dataset had string values that were transformed to numeric data for more thorough analysis.
- After that, the data was analyzed using graphs on the basis of a variety of distinct parameters.
- The complete dataset has been split into two groups: Train and Test.
- We will train our datasets using CNN algorithms.
- The model will then be evaluated using the accuracy score, classification report, and confusion matrix

3.2 Non-Functional Requirements

Non-functional requirements describe how the system should work or, in other words, how the system or model should act. There are a variety of attributes that are considered non-functional system requirements. Performance, scalability, adaptability, portability, maintainability, and reliability are a few of them. Nonfunctional requirements are often known as system quality attributes. These are only attributes of a system in development, hence there are no codes to execute them. On the basis of client need, the traits might be prioritized. Only those requirements that are absolutely important for the project should be chosen.
Some Non-Functional Requirements are as follows:

- **Reliability**

  The model should be good enough that customers, or in this case, patients, can trust the model's predictions. If clients don't trust the model's expected outcome, the model isn't very useful. Customers should be able to trust the model if it makes accurate forecasts

- **Maintainability**

  Maintenance should not be complicated or tough because the model will be executed and run on a range of machines after deployment. Problem detection should be quick and easy if it creates issues or crashes at that time.

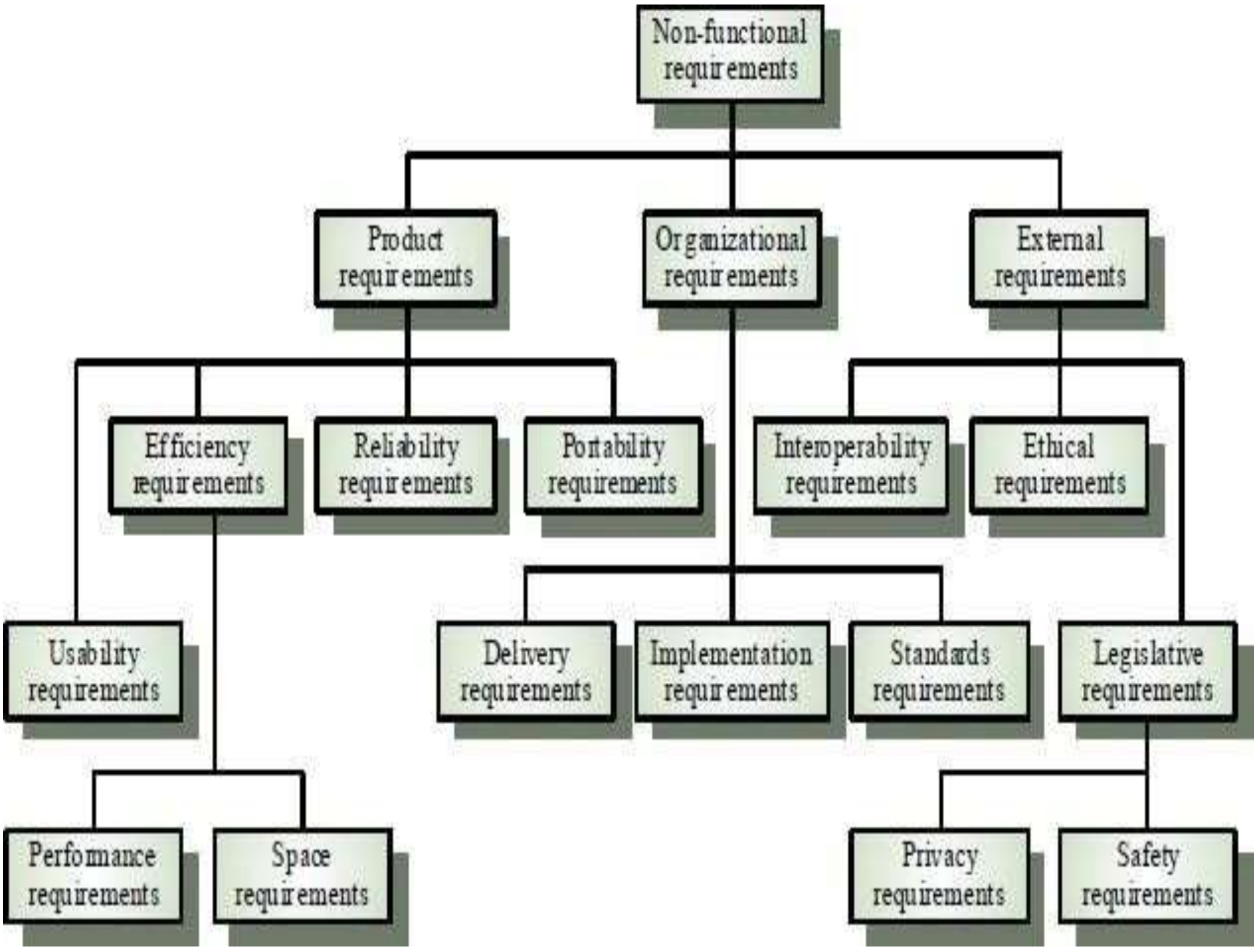


Fig 3.1 Non-Functional Requirements

- **Performance**

  The accuracy of the model, or how well it predicts the autistic disease, is what the model's performance is all about. When we apply the model to forecast real-world scenarios and evaluate it on a set of test cases, for example, If the majority of the model's predictions are true, we may say the model's performance is excellent; but, if the majority of test cases are erroneous, the model's performance is insufficient, and additional work on it should be done.

- **Portability**

  The model should not be restricted to a single or a small number of systems. It should also be available on other platforms. In this project, for example, we used the web server to make the model portable. When the needed server is setup, the model can be accessible via the internet platform.

- **Scalability**

  The structure should be versatile enough to allow for the inclusion of new features in the future. Clients who want to add new features may face difficulties if a model isn't versatile enough to allow for changes.

- **Flexibility**

  This non-functionality feature refers to a system's inability to cope with changing events and conditions, resulting in its ineffectiveness lack adapting to new rules and approaches. When a system is adaptable, it will have little trouble reorganizing itself to accommodate the necessary changes. When approaches change, the model's flexibility becomes a critical factor. As a result, it's critical for a system to be adaptable and flexible enough to meet the demands.

3.3 Hardware Requirements for the model

System Processor required	-	Core i3 / i5
Hard Disk required	-	500 GB
Ram required	-	4 GB

- ❖ Any desktop / Laptop system having these configurations is good

3.4 Software Requirements for the model

Operating system required	-	Windows / Linux
Programming Language used	-	HTML, CSS, JavaScript, Python
Framework needed	-	Anaconda
IDLE used	-	Jupyter Notebook, VS Co

3.5 Product Requirements:

Correctness:

It used a pre-trained classification model and adhered to a clear set of procedures and rules to converse with the user. Rigorous testing was also done to ensure that the data was accurate.

Modularity:

It used a pre-trained classification model and adhered to a clear set of procedures and rules to converse with the user. Rigorous testing was also done to ensure that the data was accurate.

Robustness:

The user can anticipate outcomes with the highest level of relevance and accuracy in a short amount of time because to the software's optimized overall performance. Non-functional requirements are often known as a system's characteristics.

CHAPTER 4

SYSTEM ARCHITECTURE AND DESIGN

4.1 System Design

The process of defining the components, modules, interfaces, and data for a system in order to meet specific criteria is known as system design. The process of building or altering systems, as well as the procedures, techniques, models, and methodologies required to do so, is known as system development.

The WebApp's regular hypermedia body is perceived by the System Engineering setup. The goals set for a WebApp, the item to be offered, the clients who will come, and the course hypothesis that has been developed are all connected to the building arrangement. Content creation is based on how substance questions are posed and structured for presentation and course. WebApp development is the process of creating an application to manage client relationships, manage internal planning tasks, sway course, and deliver content.

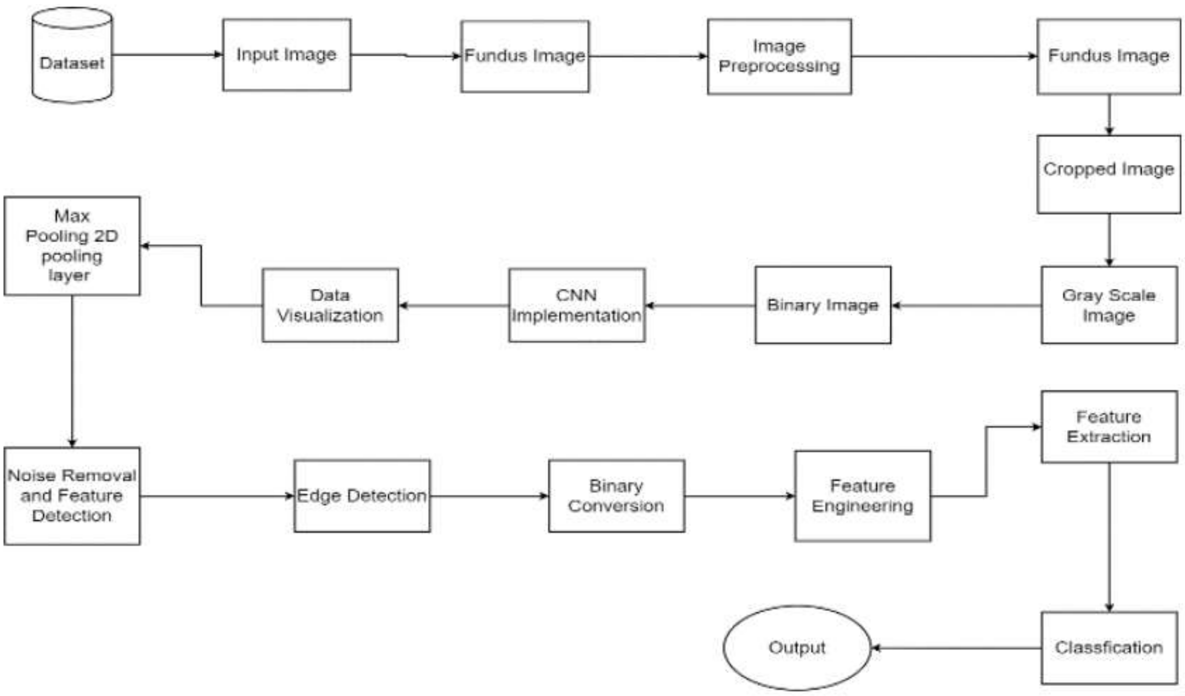


Fig 4.1: System Architecture Diagram

A software product's architectural framework is described in high-level design (HLD). The architecture diagram gives a comprehensive picture of a system by highlighting the key elements that would be created for the final product and their interfaces. In the HLD,

Possibly nontechnical to lightly technical language that system administrators should be able to comprehend. For programmers, however, low level design further reveals the logical precise design of each of these pieces. The design that is used to create the specifications for software is known as high level design. This chapter generates a full system design and demonstrates how the modules, submodules, and data flow between them are carried out and integrated. It demonstrates the implementation process and is made up of relatively straightforward parts.

The briefs on design considerations describe how the system behaves in boundary contexts and what steps should be done in the event of an anomalous scenario. Data collection, pre-processing techniques, classification, and prediction are a few design considerations. In order to help designers apply the universal accessible design principles and standards to buildings and facilities, design considerations have been developed. They can also be used to find weaknesses in current systems.

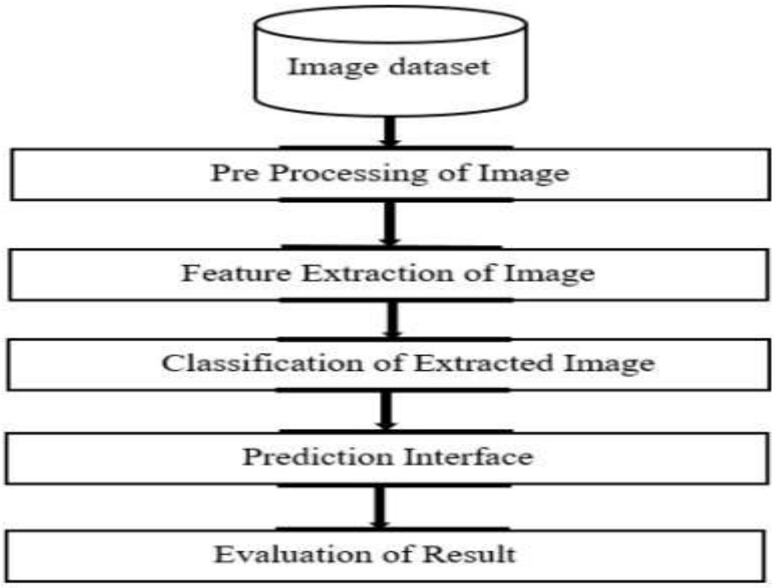


Fig 4.2: Image Classification

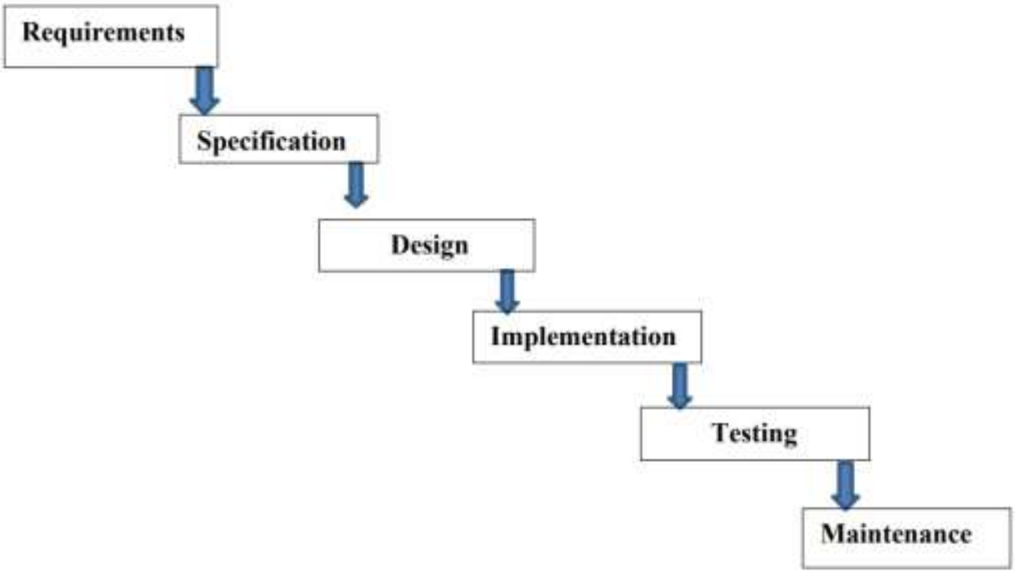


Fig 4.3 : waterfall model for DR

The proposed system has the following steps for weapon and militant detection

1. Image Pre-Processing.
2. Identification
3. Feature Extraction
6. Image Recognition

Image preprocessing: The goal of the image processing mechanism is to manipulate images in various ways to improve the image quality. Image processing techniques use images as their input and output. When analysing the transition of one image into another,

improvement of appearance. First, a grayscale image is created from an RGB image. It makes the process simpler and helps the image become less complex. the use of min-max Grey scale values are transformed into binary values using the scalar approach. The obtained binary values are used as the process's input. Consider the regions of one value in the produced binary matrix to be white, and the regions of zero value to be black. The region of interest can be located by using these variables. so that the values can be used to extract features and identify regions of interest.

Identification: At this stage, identifying the area that needs to move on to the next step involves determining the specific area of the image that will be used for further steps like feature extraction and image classification. The pre-processing step's result is used as the starting point for the identification procedure. The binary values obtained during the pre-processing stage serve as the foundation for this process. The area shown in black is thought to be of interest. the area of interest discovered through image pre-processing. It is believed that area to be the part of the image from which the weapon and the enemy would be identified. The feature extraction procedure receives the recognized weapon and militant photos.

Feature extraction: Extract the necessary characteristic from the selected region that was collected in the previous step in this stage. By converting the decreased size matrix to control over fitting, that area is condensed. The memory size of the photos is reduced by reducing the matrix size. The reduced matrix is next subjected to the flattening process, which turns it into a one-dimensional array that is used for the final detection.

4.2 Flow Chart Diagram

A graphical depiction of a computer program in terms of its function sequence (as distinct from the data it processes). A flowchart is one of the seven basic quality gadgets used on the adventure board, and it depicts the exercises required to fulfil the goals of a specific task in the most practical order. Also known as strategy maps, they are tools that display a progression of steps with expanding possibilities that depict at least one information source and converts it to yields.

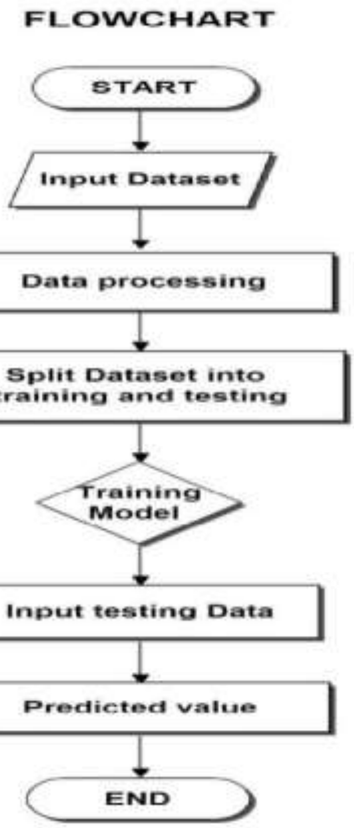


Fig 4.4: Flow Chart Diagram

4.3 Use Case Diagram

A use case is a written description of how people will use your website to accomplish activities. It describes how a system reacts to a request from the perspective of a user. Each use case is described as a series of basic actions, starting with the user's goal and ending with the achievement of that goal.

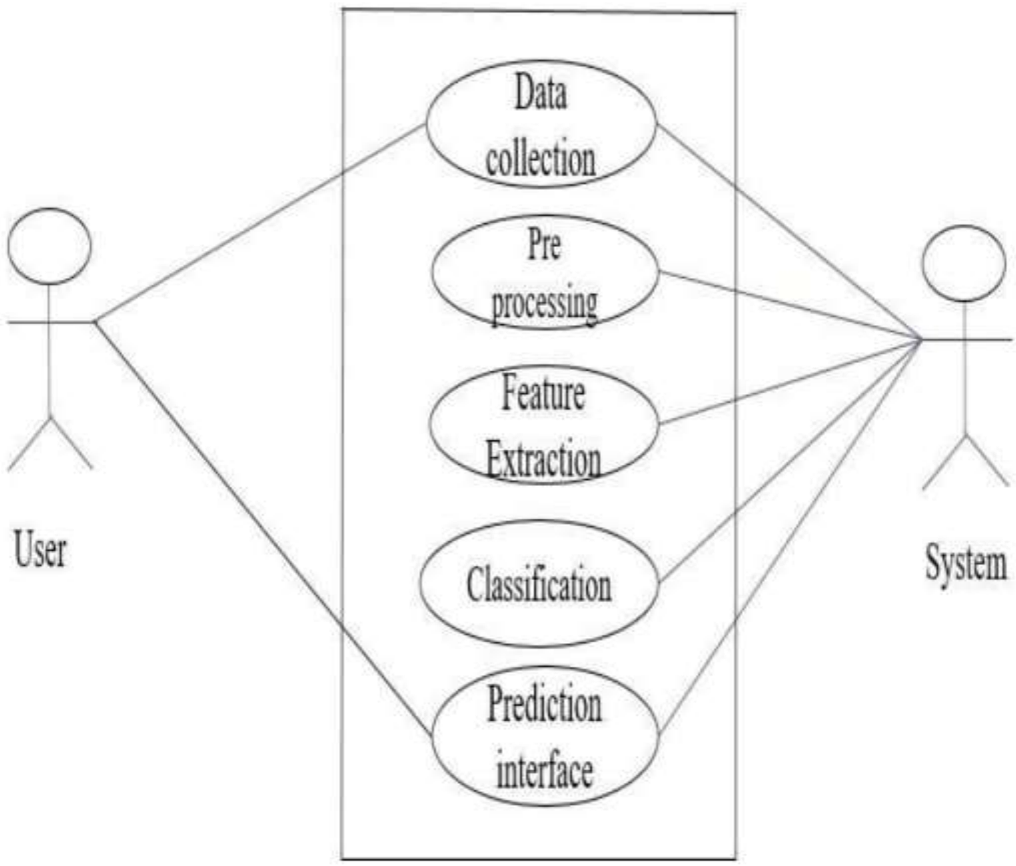


Fig 4.5 Use Case Diagram

4.4 Dataflow Diagram

- An information stream outline, also known as a DFD, is a realistic representation of a data framework's "stream" of information. A data stream diagram can also be used to depict data preparation (organized structure). Drawing a setting level DFD initially, which depicts the collaboration between the framework and external parts, is standard procedure for a designer. DFDs depict the flow of information from external sources into the framework, as well as how the information goes from one procedure to the nextand its proper storage. There are just four photos in all.

- Outside elements, such as sources and purposes of data entering and exiting the framework, are represented by squares.
- Rounded square shapes speaking to forms in various strategies may be referred to as Exercises, Activities, Techniques, Subsystems, and so on, which accept data as data, prepare it, and then yield it.
- Arrows pointing to information streams, which could be electronic data or actual objects. Information cannot be transferred from one information store to another without the use of a procedure, and outside components are not permitted to access information stores directly.
- The level three-sided square shape alludes to the need for data repositories to both receive and distribute data for storage and preparation.

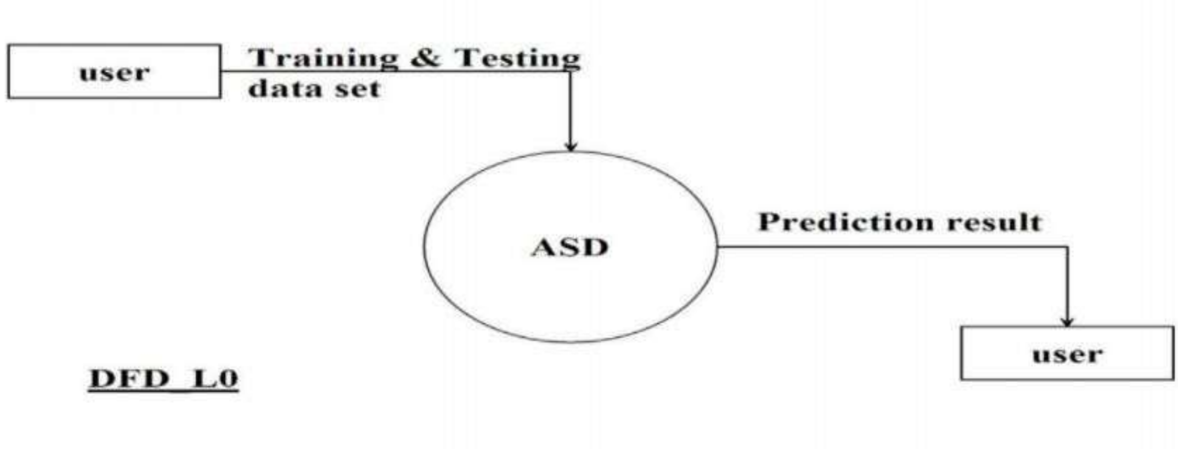


Fig 4.6 Dataflow Diagram L0

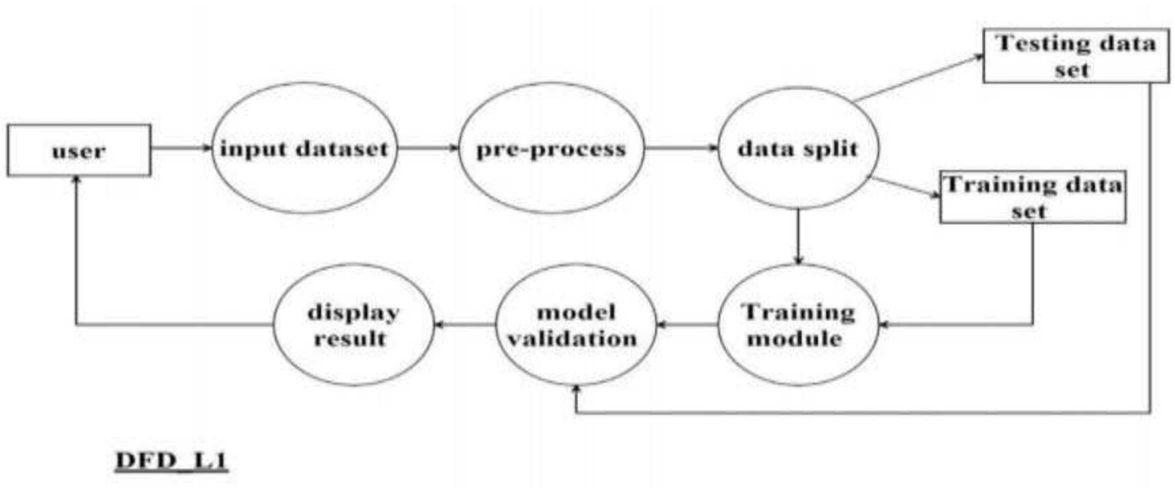


Fig 4.7 Dataflow Diagram L1

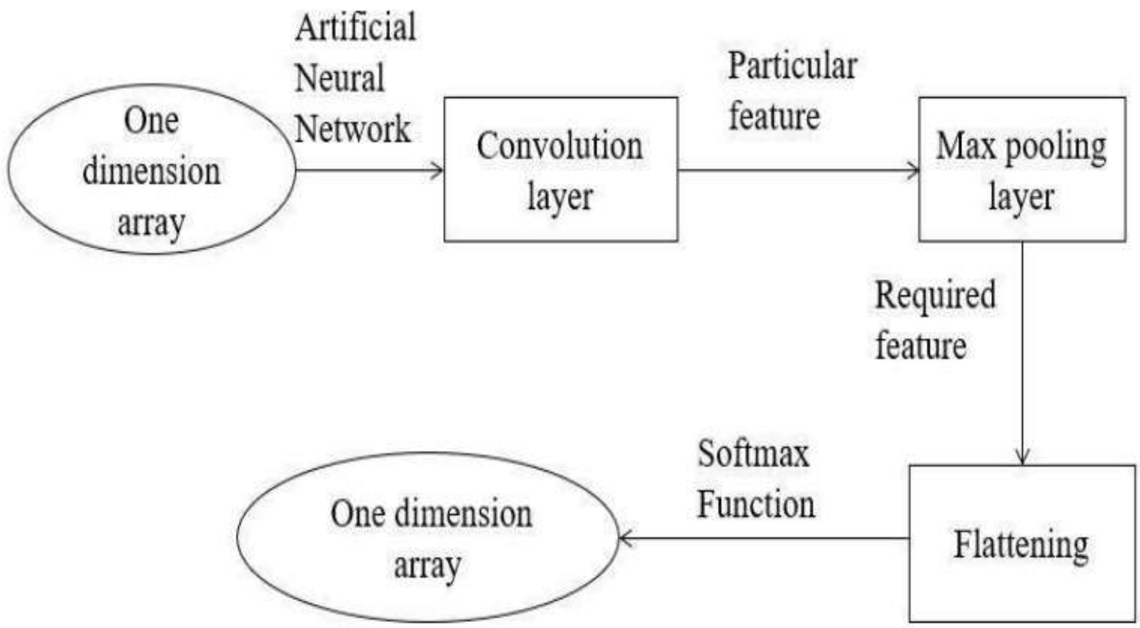


Fig 4.8 Dataflow Diagram L2

4.5 Module Description

4.5.1 Modules

- Data Acquisition and Preprocessing
- Feature Selection and Data Preparation
- Model Construction and Model Training
- Model Validation and Result Analysis

4.5.1.1 Data Acquisition and Pre-processing

Artificial intelligence needs two things to work, data (piles of it) and models. While verifying the data, make sure to populate your learning model with enough features (a section of data that might aid in the fulfillment of a wish, such as the outside of a house to estimate its value). With everything taken into account, the more data you have the better so make to go with enough segments.

The basic information gathered from web sources is still in the form of decrees, digits, and abstract phrases. Mistakes, bans, and anomalies can be found in the raw data. It requires changes after mindful researching the completed surveys. The going with progresses is locked in with the treatment of basic data. A massive amount of unprocessed data has gathered through field study need be collected. Data pre-processing is a data mining technique for transforming raw data into a format that is both usable and efficient. Figuratively speaking, at whatever point the data is collected from sources it is accumulated in rough association which isn't attainable for the assessment. Data Pre-processing is central because of the closeness of unformatted real data. Generally certifiable data is made out of -

- **Mistaken information (missing information)** –The data can have many irrelevant and missing parts. When some data in the data set is missing, this condition occurs. There are numerous explanations behind missing information, for example, information isn't persistently gathered, a mix-up in information section, specialized issues with biometrics and significantly more.

- **The nearness of noisy information (Wrong information and exceptions)** – Noisy data is a meaningless data that can’t be interpreted by machines. Poor data collection, data input issues, and other factors can all contribute to this. The purposes behind the presence of boisterous information could be an innovative issue of contraption that assembles information, a human misstep during information passage and substantially more.

- **Conflicting information** – The presence of duplication inside the information, human information transmission, containing botches in codes or names, i.e., infringement of information imperatives, and a variety of other factors contribute to the presence of irregularities.

4.5.1.2 Feature Selection and Data Preparation

Feature selection is a process of selecting out the most significant features from a given dataset. Feature selection enhance the performance of a machine learning model. Feature engineering is the process of constructing new features from existing data to train a machine learning model. Highlight Engineering is the largest AI workmanship, making a huge difference between a tolerable and a terrible model. Feature planning is the way to turn raw data into characteristics that better respond to farsighted models in order to achieveimproved model precision in the case of unnoticeable data.

The route toward sifting through data into social events and classes dependent on explicit qualities is known as the gathering of data. Gathering helps to build connections between the observation classes. It might be either according to numerical characteristics or as showed by attributes. Here, we must therefore imagine the prepared data to find out if the readiness data has the correct name, which is referred to as the purpose or target.

Next, we will cut a solitary informational collection into a training set and test set.

- **Training set -** It is a subset of dataset to train model.
- **Test set -** It is a subset to test the trained model.

4.5.1.3 Model Construction and Model Training

The path to ML model setup includes the use of an ML count (i.e., the learning figure), where ready data can be collected. The term ML model implies the doodad model which the readiness technique makes. The arrangement data must contain the correct answer, which is known as a goal or target trademark. The learning content finds plan for a map of the data credits (the correct reaction to be expected) in the arrangement information, and gives an ML model which gives you these models.

4.5.1.4 Model Validation and Result Analysis

In testing stage, the model is applied to new game plan of data. The arrangement and test data are two particular datasets. The goal in building an AI model is to have the model perform well. On the arrangement set, similarly as summarize well on new data in the test set. When the create model is attempted then we will sit back data for the estimate. At the point when desire is done then we will separate the respect find the critical information. Wemay measure the effectiveness of a model and its hyperparameters by applying it to some of the training data and comparing the prediction to the known value.

4.5.1.5 Never train on test data.

In the event that you are seeing shockingly acceptable outcomes on your assessment measurements, it may be an indication that you are incidentally preparing on the test set.

CHAPTER 5

ALGORITMS

5.1 INTRODUCTION:

A convolutional neural network is made up of a network of artificial neurons that are connected and have biases and weights that can be learned. These neurons communicate with one another. The connections' numerical weights are adjusted during training so that, when given a picture or pattern to recognize, a fully trained network will react appropriately. Multiple layers of neurons that can recognize features make up the network. Numerous neurons in each layer react to various combinations of inputs from the layers below.

The layers are constructed as indicated in Figure so that the first layer recognizes a set of basic patterns in the input, the second layer recognizes patterns of patterns, and the third layer recognizes patterns of those patterns.

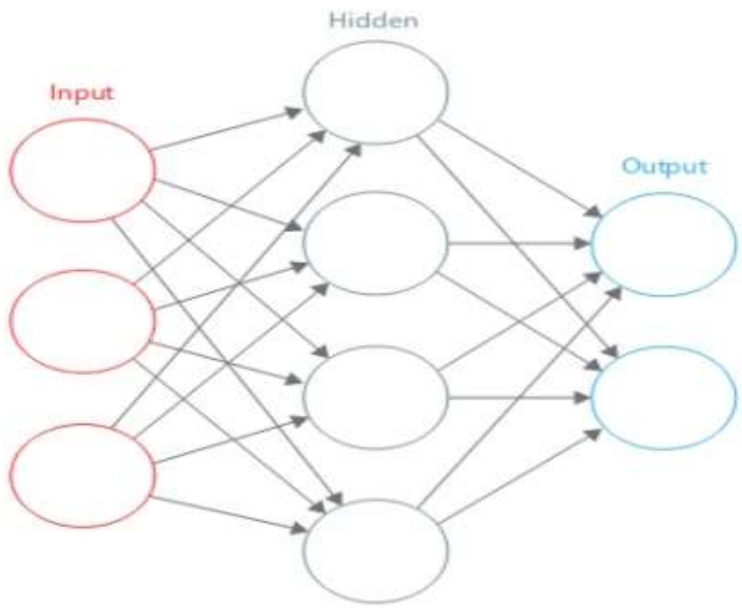


Figure 5.1: Artificial neural network

5.2 Layers of CNN:

Complex architectures are created for classification challenges by stacking various layers in a CNN. Convolution layer, pooling/subsampling layer, non-linear (ReLU) layer, and fully linked layer are the four different types of layers. The different layers of CNN are displayed in Figure 3.3. The convolution layer receives some input from the input image. The pooling layer receives the output of this layer after that. This is repeated after which a fully connected layer performing categorization is applied.

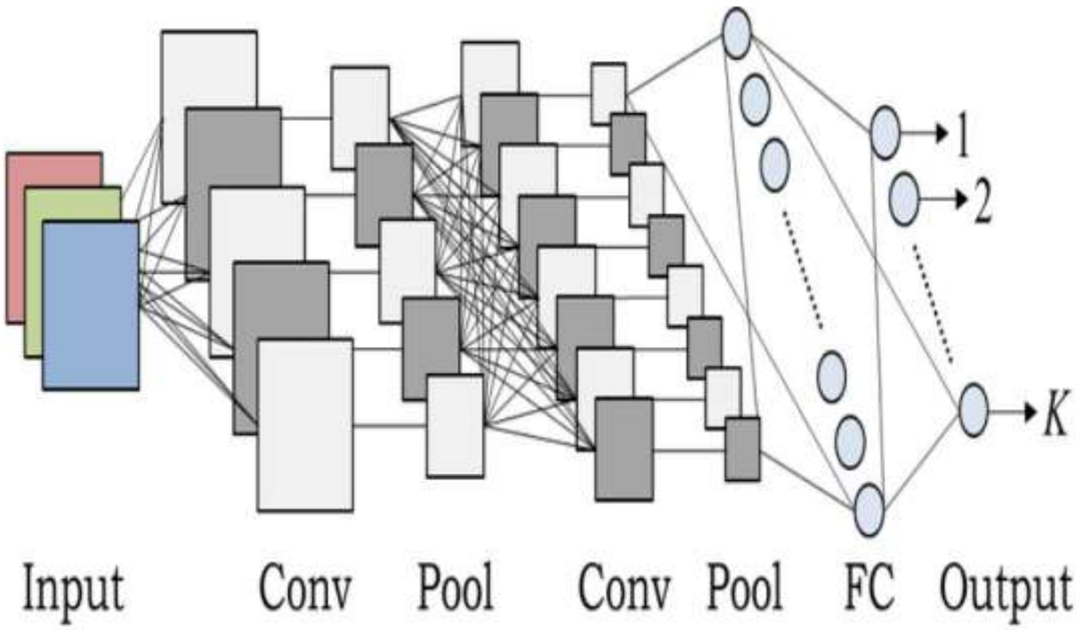


Figure 5.2: Architecture of CNN

5.3 Convolution layers

The convolution technique extracts many information from the input. the initial Low-level features like edges, lines, and corners are extracted using convolution layer. Higher-level layers get higher-level features out of the data. Figure 3.5 depicts the 3D convolution technique utilized in CNNs. The input is convolved with H kernels, each of size k x k x D individually. The input is of size N x N x D. Convolution of an input with a single kernel yields a single output feature, and independent convolution with H kernels yields H features.

Each kernel is moved one element at a time from the top-left corner to the bottom-right corner of the input, starting there. The kernel is moved one element at a time from left to right until it reaches the top-right corner, at which point it is moved one element downward. Up till the kernel reaches the bottom-right corner, this process is repeated.

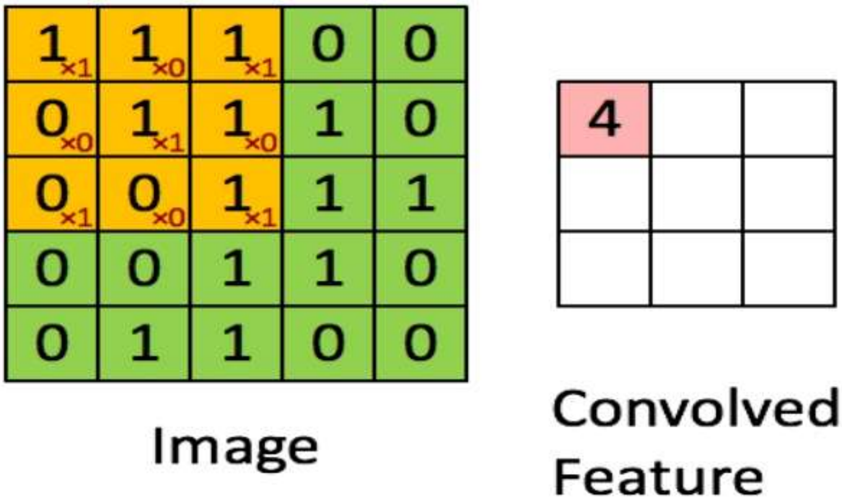


Figure 5.3: Representation of convolution network

For example, if N = 5 and k = 5 , there are 5 separate positions from left to right and 5 separate positions from top to bottom that the kernel can take. Corresponding to these positions, each feature in the output will contain 28x28 (i.e., (N-k+1) x (N-k+1)) elements. For each position of the kernel in a sliding window process, k x k x D elements of input and k x k x D elements of kernal are element-by-element multiplied and accumulated. So to create one element of one output feature, k x k x D multiply-accumulate operations are required.

5.4 Pooling Layer

The pooling (subsampling) layer aids in lowering the features' resolution. The features are resistant to distortion and noise. There are two methods for performing pooling operations: maximum pooling and average pooling. The input is segmented into separate, non-overlapping sections in both circumstances.

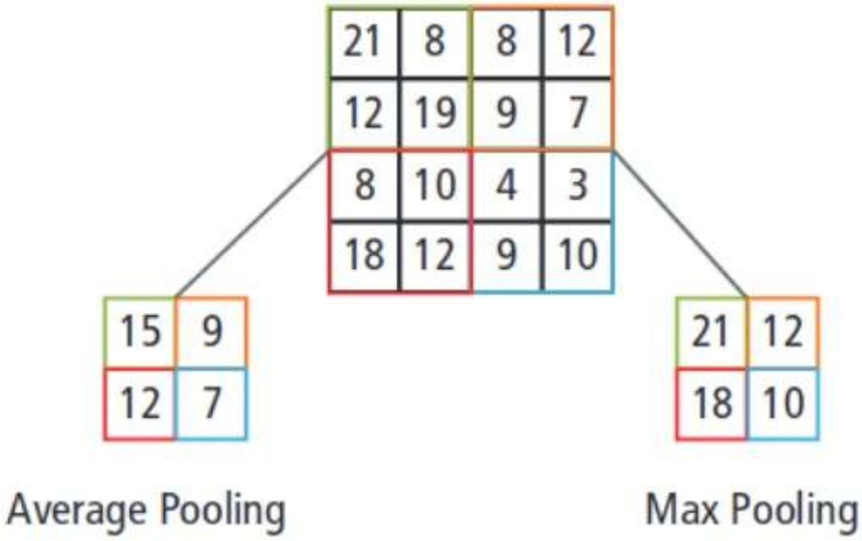


Figure 5.4: Representation of max pooling and avg pooling

The input, a 4x4 image, is separated into four 2x2 non-overlapping matrices, each shown as a square in one of four colors: green, yellow, red, or blue. The maximum value of the four values in the 2x2 matrix—21 from the green matrix, 12 from the yellow matrix, 18 from the red matrix, and 10 from the blue matrix—is used in the max pooling process and is used as the output. The average of the four values—15 from the green matrix, 9 from the yellow matrix, 12 from the red matrix, and 7 from the blue matrix—is instead used in the average pooling scenario. If the averaged result is a fraction, it must be rounded to the nearest whole number.

5.5 Advantages of CNN

The last 50 years have seen the development of neural networks and other pattern recognition techniques. Significant progress has been made in the field of convolutional neural networks recently.

Ruggedness to shifts and distortion in the image: The CNN is unaffected by distortions during detection, such as form changes brought on by the camera lens, various lighting situations, stances, the existence of partial occlusions, horizontal and vertical shifts, etc. because weight configurations are consistent across space. Shift invariants include CNNs.

Shift invariants can also be attained using completely connected layers. The result of the training in this instance is several units with identical weight patterns at various input points. To master these weight configurations and to cover the range of potential permutations, many training cases would be needed.

Fewer memory requirements: When a fully linked layer is utilized to extract the features, an order of 106 coefficients will be needed if the input image is 32x32 and the hidden layer has 1000 features. This calls for a lot of RAM. However, when we employ a convolutional layer, the same coefficients are applied throughout the space at various points. Thus, there is a significant reduction in the memory required.

Easier and better training: The common neural network, which has several parameters, is comparable to a CNN. The amount of training time would rise correspondingly. In CNN, there are significantly fewer parameters, which results in a shorter training period. With the premise of perfect training, a typical neural network can be developed whose performance is identical to CNN's. A conventional neural network like CNN would have more parameters for practical training. This causes the training process to add extra noise. As a result, a normal neural network's performance will never be as good as a CNN.

Better Accuracy: Convolutional neural networks are more accurate in solving image recognition issues. Comparing CNNs to various classification techniques, it has been discovered that they have higher accuracy means.

5.6 Application of CNN

Image Classification: Due to their capacity to combine feature and classifier learning, CNNs outperform other techniques in terms of classification accuracy on large datasets.

Speech Recognition: A recent application of Convolutional Neural Networks in Speech Recognition. In comparison to Deep Neural Networks (DNN), CNN produced greater outcomes. Researchers from Microsoft Corporation identified four niches where CNN outperforms DNN in 2015. they are

(1) Noise robustness

(2) Distant speech recognition

(3) Low-footprint models

(4) Channel-mismatched training-test conditions.

Face Recognition: The face identification method involves solving issues including focusing on each face despite poor lighting or a varied stance, recognizing all the faces in the photo, finding distinctive traits, comparing those features to those in the database, and recognizing the person's name. A complex, multidimensional visual stimuli that is represented by faces. It is demonstrated utilizing a hybrid neural network made of a convolutional neural network, a self-organizing map neural network, and local image sampling.

Scene Labelling: In scene labelling each pixel is labelled with the category of the object it belongs to, CNNs are very effective in scene labelling.

Action Recognition: The challenges in creating an action recognition system are the translations and distortions of features in various patterns that belong to the same action class. Utilizing CNNs will enable us to solve these challenges.

Human Pose Estimation: Recognition of human poses in computer vision has long been a challenge. This issue couldn't be resolved because of the high dimensionality of the input data and the high diversity of possible body positions. CNNs are now useful for estimating human stance.

Document Analysis: Traditional handwriting recognizers leverage the sequential character of the pen trajectory by encoding the input in the time domain. These representations are susceptible to irrelevant factors like writing speed and stroke order.

CHAPTER 6

IMPLEMENTATION TECHNOLOGIES

6.1 Introduction

The acknowledging of an application or execution of an arrangement, thought, model, plan, determination, standard, calculation, or strategy is known as usage. At the end of the day, a programmer part or any other PC framework through programming and arrangement recognizes a specialized determination or calculation. Numerous usages may exist for a given particular or standard.

Execution is one of the most significant periods of the Software Development Life Cycle (SDLC). It covers all procedures for the proper functioning of new programming or equipment, including set-up, design, execution, testing and implementation of important charges. It codes the framework using a certain language of programming and moves the structure into an actual framework.

6.2 Overview of System Implementation

This project is implemented considering the following aspects:

6.2.1 Usability aspect
6.2.2 Technical aspect

6.2.1 Usability Aspect

The usability aspect of implementation of the project is realized using following principles:

6.2.1.1 The project is implemented using PYTHON

Python is a high-level interpreted language of broadcasting. As an ABC programming language successor, Guido van Rossum first started working on Python in the late 1980s and was first published in 1991 as Python 0.9.0. Python 2.0 was launched in 2000 and introduced new features such as list comprises and a refund system. Python 3.0 was released in 2008 and has not been fully backwardly compatible with a major revision in the language and many Python 2 code are not updated on Python 3. In 2020 version 2.7.18 was discontinued.

In addition to Python's objectively oriented approach to help programmers to write clear logical code for large and small projects, Python is a language structure. Python is dynamically designed and collected with waste. It supports multiple paradigms of programming including structured programming (partly procedural), object-oriented and functional.

Python has a strong tape frame and the executives' programmed memory. It promotes a wide range of ideal programming models, including object-set, basic, practical and procedural models. It comes with a huge and comprehensive standard library.

For some working frameworks, Python mediators are available. CPython, the lead execution of the Python system, is a network-based improvement model and open-source programming. The not-for-profit Python Software Foundation is supervised by Python and CPython.

Python had been designed to be exceptionally extensible, as opposed to having all its usefulness incorporated into its centre. This minimal measured quality has made it famous especially as a way of adding to existing applications programmable interfaces. Van Rossum was dissatisfied with his dissatisfaction with ABC and adopted the opposite approach to his visions of a centrelanguage with a huge standard library and an effective mediator.

6.2.2 Technical Aspect

The technical aspect of the project's implementation is carried out according to the following principle:

Anaconda

The programming languages of Python and R (data science, apps for apprenticed learning, extensive data processing, and prediction analytics), etc. are distributed in Anaconda. This distribution includes Windows, Linux and MacOS-suited data-science packages. Developed and maintained by Anaconda, Inc., the distribution is automatically installed with 250 packages and over 7500 additional open-source packages, including the anaconda package and virtual environment manager from PyPI. In addition, this distribution offers more than 250 packages.

Anaconda Navigator is the Anaconda distribution's Graphical User Interface (GUI) desktop which allows users to launch applications and manage anacondas, environments and channels without the use of command-line commands. Navigator can find packages on Anaconda Cloud or in the Anaconda repository in a local environment, install and update packages in an environment. It comes with Windows, macOS and Linux. The bundle adaptations of the board framework anaconda are monitored.

6.2.2.1 NumPy

NumPy is a library that supplies the Python programming language with a large number of high-level math functions to function with the arrays, and adds a support for large multi-dimensional arrays and matrices. NumPy has a wide range of contributors and is open-source software. NumPy is the basic Python science computing package. Among others, it contains things

- Powerful array object N-dimensional
- Advanced (broadcast) features
- C/C++ and FORTRAN code integration tools
- Useful linear algebra, transforming Fourier and capability for random numbers

In addition, NumPy can also be used as a productive multi-dimensional container for non-exclusive information, apart from its undeniable logical uses. Subjective types of information can be characterized. This enables NumPy to flawlessly and expeditiously integrate a wide range of databases.

6.2.2.2 Matplotlib

Matplotlib is a Python 2D array plot visualization library. Built over NumPy arrays, Matplotlib is a multiplatform data display book to work with a wider SciPy stack. It was launched in 2002 by John Hunter. In Python, the Python, IPython and Jupyter shells, Jupyter's notebooks, web app servers and four graphical user interface toolkits, Matplotlib can be used. Matplotlib is composed of several tracks such as line, bar, dispersion, histogram etc.

A wide range of plots is available in Matplotlib. Plots help to understand and correlate trends and patterns. They are typically tools for quantitative information reasoning.

A MATLAB-like interface, in combination with IPython, is provided for simple plotting of the pilot module. You can use an object-oriented interface or a set of functions familiar to MATLAB users to control the line styles, fonts, axis properties etc.

6.2.2.3 Pandas

Pandas is a Python programming language software library designed for the handling and analysis of the data. Data structures and operations for the handling of numerical and time series tables are operated. The name comes from the term 'panel data,' which is an econometric term for data sets which includes observations for a single person over multiple periods.

Pandas offers a fast flexible and expressive data structure designed to work on structured (table, potentially multidimensional) and time series data. Easy as well as intuitive.

The objective of this package is to be the basic building block of high levels block to do Python data analysis on practical and real words. In addition, it has the wider objective of becoming open-source data analysis the most powerful and flexible.

Available in any language manipulation tool.

Pandas is well suited for many different types of data:

• Tabular information, like in a SQL or Excel table, with heterogeneously typed columns

• Unordered and ordered time series data (not necessarily fixed frequency).

• Arbitrary (homogeneous or heterogeneous) matrix information with row and column markings

The two primary data structures, the 1-dimensional series and the 2-dimensional data frame, handle the majority of typical cases of use in financial, statistical, social sciences, etc. And a lot of engineering fields. Data Frame offers for R users all of the data framework that R offers.

Pandas is built on NumPy and is well integrated with a number of other Third-Party Libraries within a scientific computing environment.

- Easy handling of missing data in floating point and non-floating-point data (NaN).
- Size Mutability: Columns and higher dimensional objects can be inserted and deleted from Data Frame.
- Automated, explicit data alignment: objects can be aligned to a number of labels explicitly, or user may just ignore labels and automatically let Series, Data Frame, etc.
- Strong, flexible group with function to perform split application combination activities on data sets, for the aggregation and transformation of data
- It is easy to convert rugged and otherwise indexed data into Data Frame objects in otherPython and NumPy data structures.
- Smart label-based cutting, sophisticated indexing, and sub-sets of large data sets
- Intuitive data combination and connection
- Flexible data set refurbishment and pivoting
- Hierarchical axis marking (possible to have multiple labels per tick)
- Robust IO file loader tools, Excel files, databases and ultrafast HDF5 data saving / loading formats
- Time series feature: generation of date and frequency conversion, shifting, dating, and lagging of moving window stats.

6.3 Implementation Support

6.3.1 Installation of Jupyter Notebook

The following are the requirements for the Windows Operating System installation of Jupyter Notebook:

6.3.1.1 Windows 7/8/10 from Microsoft (32-bit or 64-bit)

6.3.1.2 Minimum 2 GB RAM

6.3.1.3 Minimum disc space 1 GB available, Recommended 2 GB

6.3.1.4 Mind screen Resolution 1280x800

6.3.1.5 3.3 or higher or Python 2.7 Python

6.4 Pre-processing:

Step - 1 : Convolution

Step - 2 : Pooling

Step - 3 : Flattening

Step - 4 : Full connection

In step 1, we’ve imported Sequential from keras.models, to initialise our neural network model as a sequential network. There are two basic ways of initialising a neural network, either by a sequence of layers or as a graph.

In line 2 To conduct the convolution operation, or the first phase of a CNN, on the training images, we have imported Conv2D from keras.layers. Convolution 2-D is what we're using because we're dealing with photos, which are essentially 2-dimensional arrays; if you're dealing with films, where the third dimension is time, you might need to use Convolution 3-D.

In line 3 The second phase in the process of generating a CNN is the pooling operation, and we have imported MaxPooling2D from keras.layers. There are several sorts of pooling procedures, such as Min Pooling, Mean Pooling, etc., however we are utilising a Maxpooling function to construct this particular neural network. We require the highest value pixel from each region of interest in MaxPooling.

In line 4, we’ve imported Flatten from keras.layers, which is used for Flattening. Flattening is the process of converting all the resultant 2 dimensional arrays into a single long continuous linear vector.

And finally in line 5, we’ve imported Dense from keras.layers, which is used to perform the full connection of the neural network, which is the step 4 in the process of building a CNN.

Now, we will create an object of the sequential class below

Code:

```
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
# Initialising the CNN
classifier = Sequential()
# Step 1 - Convolution
classifier.add(Conv2D(32, (3, 3), input_shape = (256,256, 3), activation = 'relu'))
# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))
# Adding a second convolutional layer
classifier.add(Conv2D(32, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))
# Step 3 - Flattening
classifier.add(Flatten())
# Step 4 - Full connection
classifier.add(Dense(units = 128, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))
# Compiling the CNN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

6.5 Loading image Datasets:

The downloaded picture dataset must be fitted to our CNN. But first, in order to avoid over-fitting, we'll pre-process the photos. Overfitting is when nodes from one layer to another are overfitted, resulting in excellent training accuracy but very poor test accuracy.

Therefore, we must conduct some image augmentations on our images before fitting them to the neural network, which is essentially synthesizing the training data. For the synthesizing portion, as well as to prepare the training set and the test set of images that are present in properly structured directories, where the directory's name is taken as the label of all the images present in it, we will use the keras. preprocessing library. For instance: Keras will classify all of the photographs in the folder with the name "cats" as being cats.

Code:

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255,
shear_range = 0.2,
zoom_range = 0.2,
horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
training_set = train_datagen.flow_from_directory('/content/drive/My
Drive/Glaucoma/data/train',
target_size = (256,256),
batch_size = 32,
class_mode = 'binary')
test_set = test_datagen.flow_from_directory('/content/drive/My Drive/Glaucoma/data/test',
target_size = (256,256),
batch_size = 32,
class_mode = 'binary')
```

6.6 Performance Evolution:

You can find the explanation of what each of the above parameters do here, in the keras documentation page. Butwhat you need to understand as a whole of whats happening above is that we are creating synthetic data out of the same images by performing different type of operations on these images like flipping, rotating, blurring, etc. Now lets fit the data to our model !

```
score = classifier.evaluate_generator(test_set,811/batch_size)
print(" Total: ", len(test_set.filenames))
print("Loss: ", score[0], "Accuracy: ", score[1])
#print("Accuracy = ",score[1])
```

6.7 Dieses Detection :

The image that has to be tested on CNN is stored in the test_image. As the model only accepts images with a resolution of 64x64, we must first convert the test image to this format before sending it to the model. Then, in order to obtain the prediction, we use the predict() method on our classifier object. Given that the prediction will take the form of a binary number, we will either receive a 1 or a 0, which will signify the presence or absence of diabetic retinopathy, respectively.

Code:

```
import numpy as np
from keras.preprocessing import image
test_image = image.load_img('/content/drive/My
Drive/Glaucoma/data/test/class0/Im0002_ORIGA.jpg', target_size = (256,256))
test_image = image.img_to_array(test_image)

test_image = np.expand_dims(test_image, axis = 0)
result = model.predict(test_image)
training_set.class_indices

if result[0][0] == 0:
print("Diabetic Retinopathy")
else:
print("NON Diabetic Retinopathy")
```

6.8 plot graph:

Plotting the graph according to the accuracy and loss, with epoch as X axis and accuracy and loss as the Y axis. The graph is plotted for each image tested.

Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from PIL import Image
%matplotlib inline
plt.style.use('fivethirtyeight')
def plot_model_history(model_history):
fig, axs = plt.subplots(1,2,figsize=(15,5))
# summarize history for accuracy
axs[0].plot(range(1,len(model_history.history['acc'])+1),model_history.history['acc'])

axs[0].plot(range(1,len(model_history.history['val_acc'])+1),model_history.history['val_acc'])
axs[0].set_title('Model Accuracy')
axs[0].set_ylabel('Accuracy')
axs[0].set_xlabel('Epoch')

axs[0].set_xticks(np.arange(1,len(model_history.history['acc'])+1),len(model_history.history['
acc'])/10)
axs[0].legend(['train', 'val'], loc='best')
#summarize history for loss

axs[1].plot(range(1,len(model_history.history['loss'])+1),model_history.history['loss'])
axs[1].plot(range(1,len(model_history.history['val_loss'])+1),model_history.history['val_loss']
)
axs[1].set_title('Model Loss')
axs[1].set_ylabel('Loss')
```

```
axs[1].set_xlabel('Epoch')
axs[1].set_xticks(np.arange(1,len(model_history.history['loss'])+1),len(model_history.history['
loss'])/10)
axs[1].legend(['train', 'val'], loc='best')
plt.show()
```

CHAPTER 7

PSEUDO CODE

The term "pseudo code" is frequently used in programming and algorithmic domains. It is a programming paradigm that allows a programmer to represent an algorithm's implementation. Simply said, it's a representation of an algorithm that has been cooked up. Algorithms are frequently represented using pseudo codes because they can be comprehended by programmers regardless of their programming background or knowledge. Pseudo code, as the name implies, is a fictitious code or representation of code that can be understood by anyone with a basic understanding of programming. The objective of employing pseudocode is to make an algorithm's key concept more efficient. It is used to plan an algorithm by sketching out the program's structure before beginning the actual coding. Pseudocode is not a programming language in the traditional sense. As a result, it can't be turned into an executable program. Before being turned into a specific programming language, it uses short phrases or simple English language syntaxes to write code for programs. This is done to identify top-level flow faults and to comprehend the programming data flows that will be used in the final program. This saves time during real programming because conceptual problems have already been fixed.

7.1 Diabetic Retinopathy Pseudo Code

- Collect Diabetic eye fundus dataset for toddlers
- Read dataset
- Data preprocessing (Remove null entries, remove unnecessary columns, remove noisy data).
- Save clean toddler dataset into separate excel sheet.
- Read clean toddler’s dataset
- Split dataset into Training set and Testing set with the ratio of 0.8 : 0.2.
- Train all the five-machine learning model with training set.

- Check the accuracy score (training + testing), classification report, confusion matrix with the help of testing dataset.
- Compare testing vs predicted result to obtain parameters mentioned in previous line.
- Obtain the patient data from frontend as JSON.
- Convert the JSON response from frontend into Python Dictionary data structure.
- With the help of keys, select required parameter for ML models.
- Convert the obtained data into Dataframe, as models require data to be in Dataframe form to predict result.
- Obtain the result from Pre – Trained ML models.
- Combined_Result = $\sum$ ($testing\_accuracy\_score_i \times prediction\_of\_algo_i$) if $testing\_accuracy\_score_i < training\_accuracy\_score_i$
- Combined_Result will be sent to frontend via flask API.

CHAPTER 8

TESTING AND PREDICTION

8.1 Software Testing Introduction:

Software testing is a procedure used to assess the accuracy, comprehensiveness, and caliber of created computer software. The process of testing developed software is used to assess its quality. The act of testing involves running a program with the goal of identifying errors. Verification and validation are common names for software testing.

8.2 STLC(Software Testing Life Cycle):

Testing itself has many phases i.e. is called as STLC. STLC is part of SDLC

- Test Plan
- Test Development
- Test Execution
- Analyze Result
- Defect Tracking

8.2.1 Test plan:

The testing environment, purpose, scope, objectives, test strategy, schedules, milestones, testing tool, roles and responsibilities, risks, training, staffing, and who will test the application, what kinds of tests should be carried out, and how it will track the defects are all described in this document.

8.2.2 Test Development:

Preparing test cases, test data, Preparing test procedure, Preparing test scenario, Writing test script

8.2.3 Test execution:

In this phase we execute the documents those are prepared in test development phase

8.2.4 Analyze Result:

Once executed documents will get results either pass or fail. We need to analyze the results during this phase

8.2LEVEL OF TESTING USED IN PROJECT:

8.3.1 Unit testing:

Initialization testing is the first stage of dynamic testing, and developers are first in queue for responsibilities before test engineers. After the anticipated test findings are achieved or when differences are comprehensible/acceptable, unit testing is carried out.

8.3.2 Integration testing:

Every component of the application has been tested. To ensure that the interaction of two or more components yields results that meet functional requirements, integration testing is used.

8.3.3 System testing:

To test the complete system in terms of functionality and non-functionality. It is black box testing, performed by the Test Team, and at the start of the system testing the complete system is configured in a controlled environment

8.3.4 Functional testing:

The outgoing links from each page from the tested site. Check each internal link. Check for links that leap to the same pages. Verify the field default values. incorrect data entered into the form fields.

The final outcome of activities or events that are conducted qualitatively or quantitatively is referred to as a result. Performance analysis is a type of operational analysis that depicts the link between performance variables.

CHAPTER 9

Screen Shots

9.1.1 Home Page

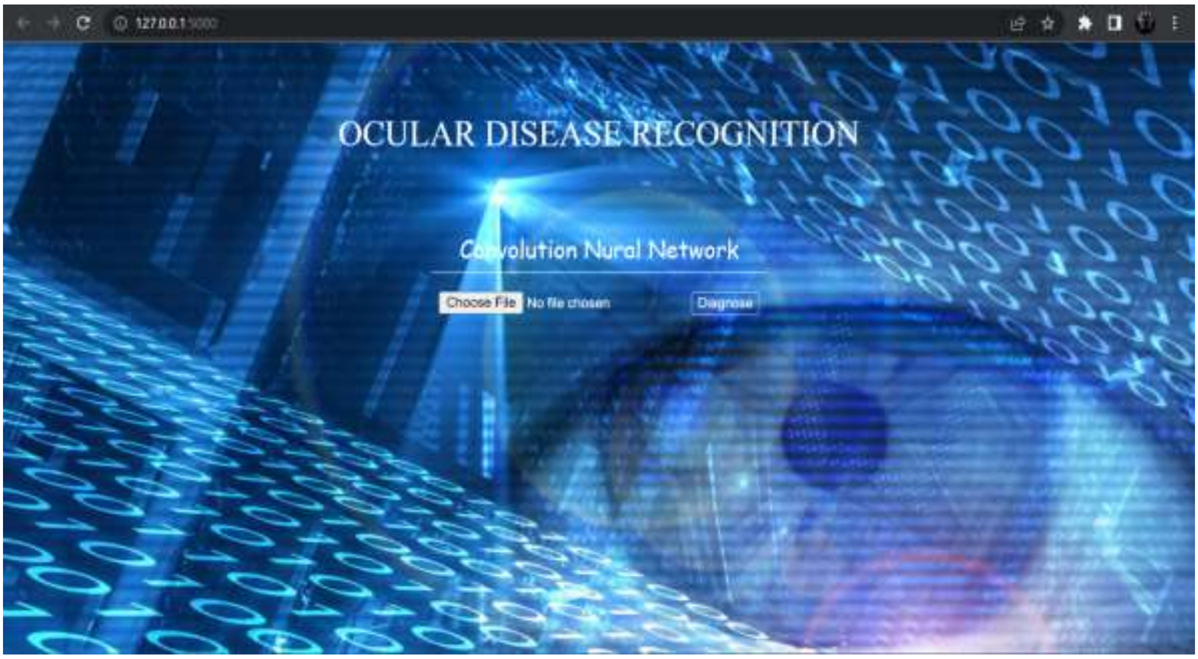


Fig 9.1: Input Button

Fig 9.2: Analysis Button

9.1.2 Selecting Input data from data set

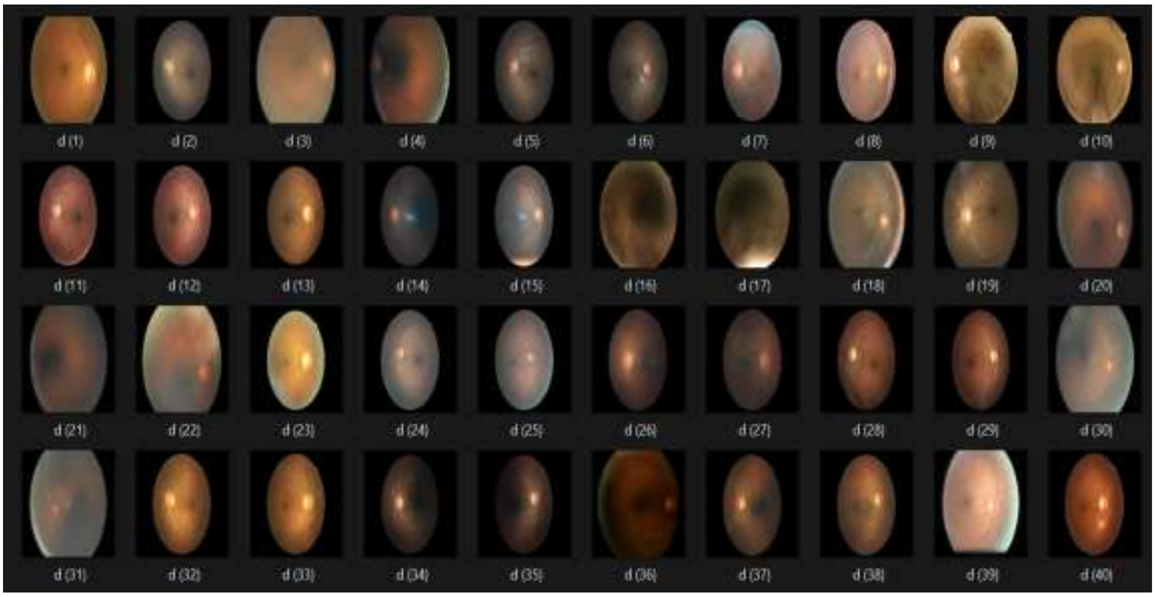


Fig 9.3: Fundus Image Dataset

9.1.3 Result Page

Fig 9.4: Moderate Fundus Image

Fig 9.5: Accuracy of Moderate Fundus Image

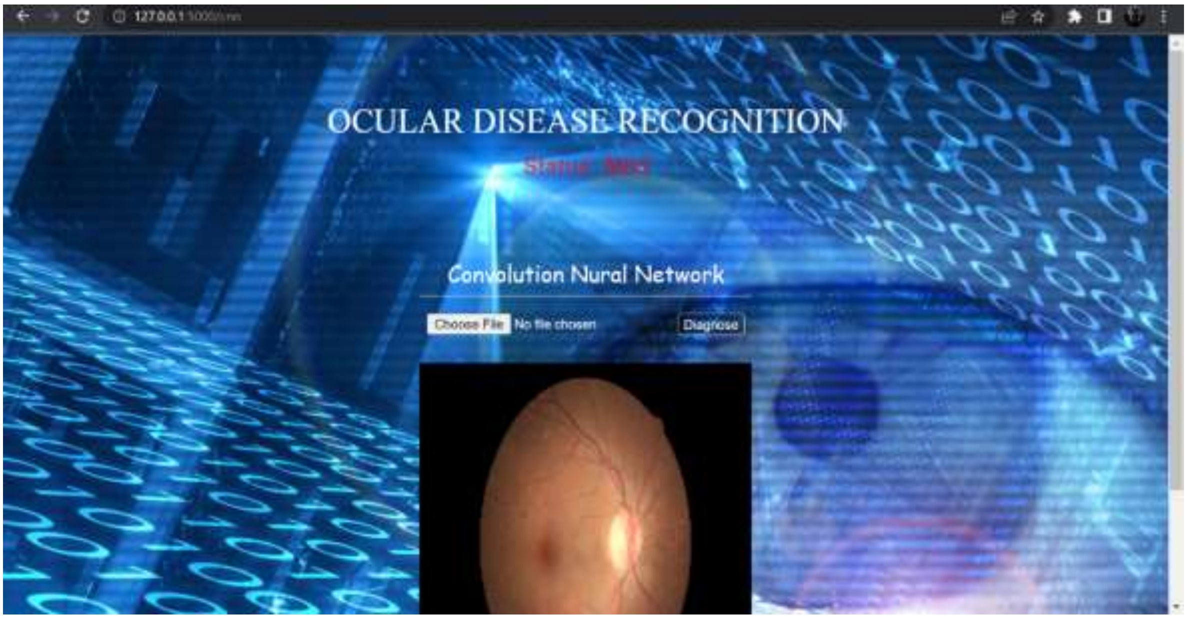


Fig 9.6: Mild Fundus Image

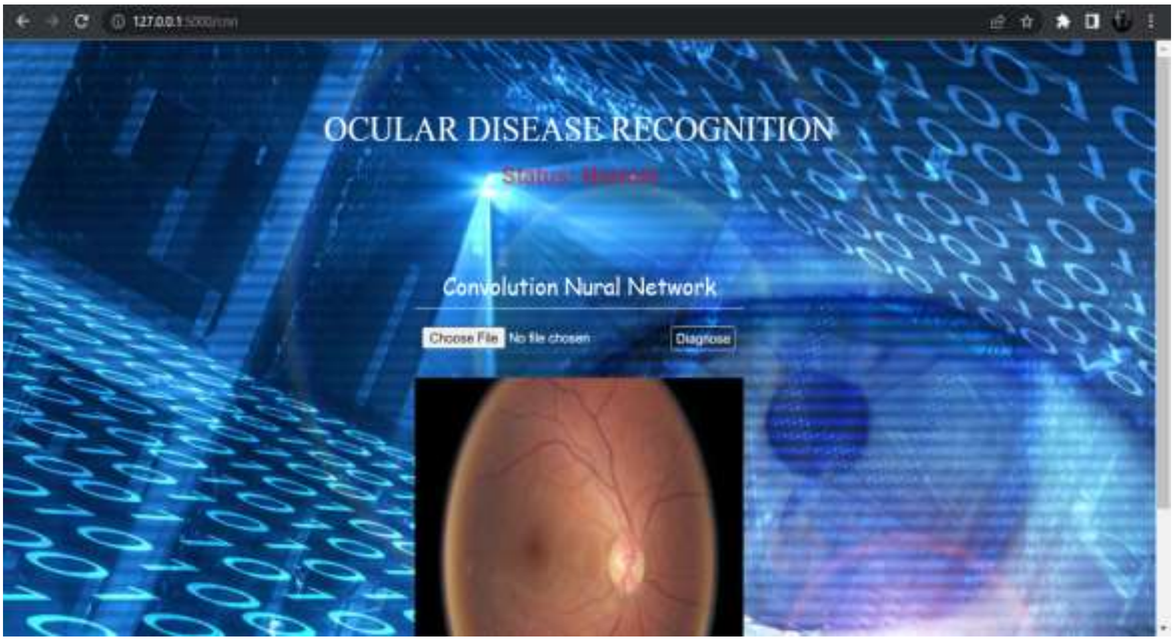


Fig 9.7: Normal Fundus Image

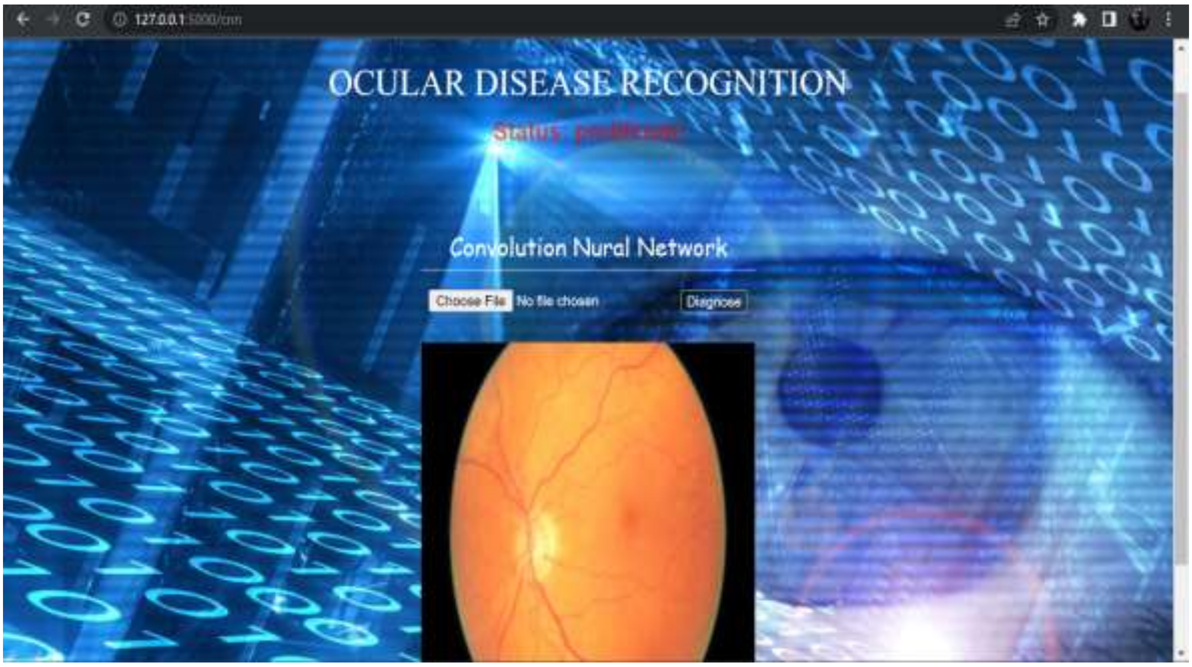


Fig 9.8: Proliferate Fundus Image

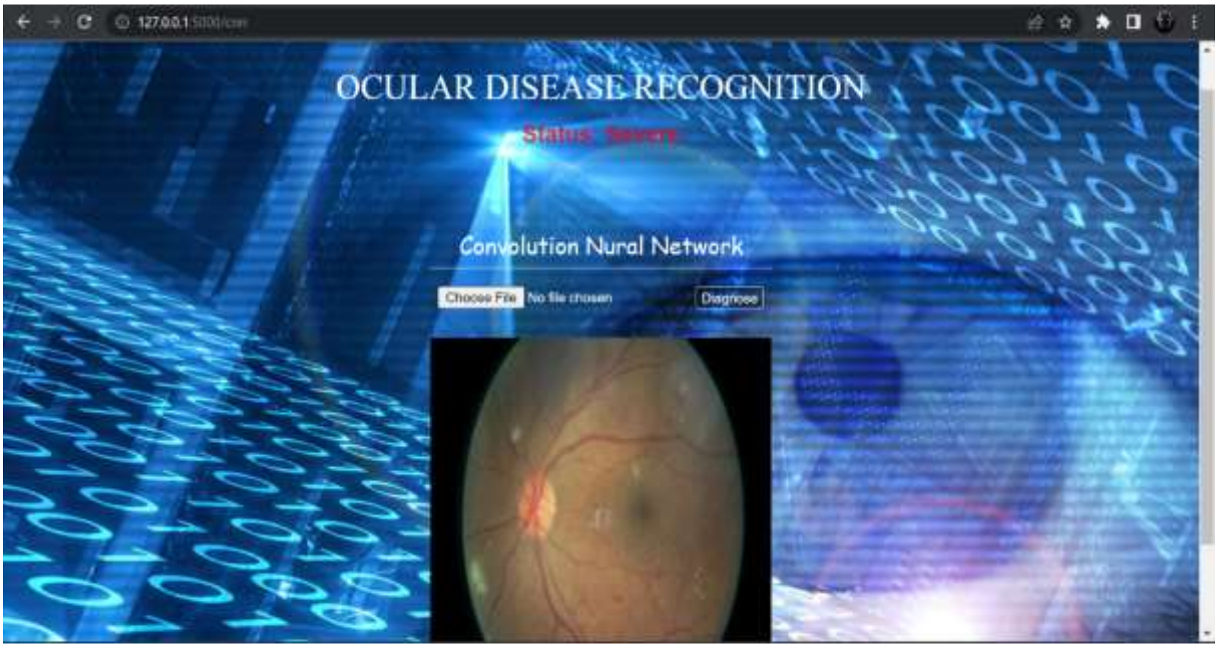


Fig 9.9: Severe Fundus Image

CHAPTER 10

CONCLUSION AND FUTURE ENHANCEMENTS

10.1 CONCLUSION

Our projects an analysis of a model to identify the severity of DR from Fundus Photographs. Our method performed well in comparison to other method. It is a fact that better and accurate the diagnosis, the more exact will be the treatment plan. So diagnostic measures should aim towards accuracy for an effective treatment regimen. In our study we were able to establish a good accuracy in the diagnosis results

10.2 FUTURE SCOPES

Although we have faced many problems dealing with the findings, feature we have picked may seem rational but they are efficient according to our study. So possible approach would be based on feature extracting. One can study further in order to detect hemorrhage more specifically. We would expect such feature engineering to make the method better to perform the whole 5-class classification perfectly. Moreover, one can be done in exploring more nuanced data normalization and denoising techniques.

REFERENCES

[1] Grading diabetic retinopathy from stereoscopic color fundus photographs an extension of the modified airlie house classification: Report number 10.

[2] Baxter H Forrester J et al Williams R, Airey M. Epidemiology of diabetic retinopathy and macular oedema: a systematic review. http://dx.doi.

[3] Ryo Kawasaki et al Joanne Yau, Sophie Rogers. Global prevalence and major risk factors of diabetic retinopathy. Diabetes Care http://care.diabetesjournals.org/ content/35/3/556.full.pdf, 2012.

[4] Hykin PG Fraser-Bell S, Kaines A. Update on treatments for diabetic macular edema. Current Opinion in Opthalmology, 2008.

[5] Linda Geiss et al. Engelgau, Michael. The evolving diabetes burden in the united states. Annals of Internal Medicine, 2004.

[6] J.M. Morel Buades, B. Coll. A non local algorithm for image denoising. http://dx.doi.org/10.

[7] GiriBabuKande, T.SatyaSavithri and P.V.Subbaiah," Segmentation of Vessels in Fundus Images using Spatially Weighted Fuzzy c-Means Clustering Algorithm," International Journal of Computer Science and Network Security, Vol.7 No.12, pp. 102-109. Dec 2007.

[8] Bob Zhang, B. V. K. Vijaya Kumar, David Zhang. Detecting diabetes mellitus and Non proliferative diabetic retinopathy using tongue color, Texture, and Geometry features. IEEE Transactions on Biomedical Engineering. 2014; 61(2), 491-501.

[9] Yuji Hatanaka, Toshiaki Nakagawa, Yoshinori Hayashi, Masakatsu Kakogawa, Akira Sawada, KazuhideKawase, Takeshi Hara and Hiroshi Fujita," Improvement of Automatic Hemorrhages Detection Methods using Brightness Correction on Fundus Images," In Proc of the international society of optics and photonics on medical imaging, Vol. 6915, 2008.

PROJECT PLAGIARISM RESULT

DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

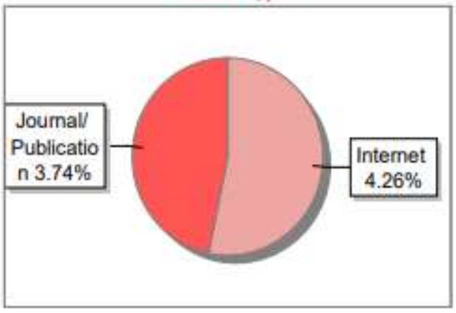
Submission Information

Author Name	Rakshith
Title	DIABETIC RETINOPATHY
Paper/Submission ID	740210
Submission Date	2023-05-12 15:25:13
Total Pages	64
Document type	Project Work

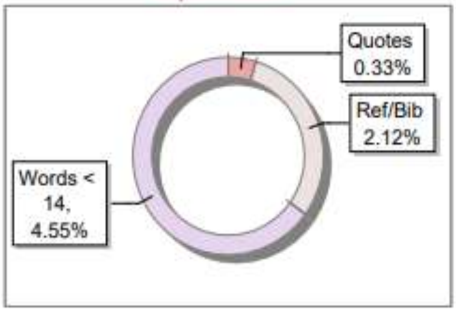
Result Information

Similarity **8 %**

Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words Similarity	Excluded
Excluded Source	**0 %**
Excluded Phrases	Excluded

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

8	22	A	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	bmsce.ac.in	1	Publication
3	www.ncbi.nlm.nih.gov	1	Internet Data
4	Thesis Submitted to Shodhganga Repository	1	Publication
5	www.nature.com	<1	Publication
6	Artificial Intelligence A Guide to Intelligent Systems	<1	Publication
7	worldwidescience.org	<1	Internet Data
8	www.ncbi.nlm.nih.gov	<1	Internet Data
9	www.dx.doi.org	<1	Publication
10	qdoc.tips	<1	Internet Data
11	erj.ersjournals.com	<1	Internet Data
12	towardsdatascience.com	<1	Internet Data
13	www.ijert.org	<1	Internet Data
14	moam.info	<1	Internet Data
15	drttit.edu.in	<1	Publication
17	www.epa.gov	<1	Publication
18	www.ncbi.nlm.nih.gov	<1	Internet Data
19	documents.newhorizonindia.edu	<1	Publication
20	qdoc.tips	<1	Internet Data
21	www.scribd.com	<1	Internet Data
24	repository.mut.ac.ke 8080	<1	Publication
26	www.ncbi.nlm.nih.gov	<1	Internet Data
39	www.scribd.com	<1	Internet Data

EXCLUDED PHRASES

1	**jssate**
2	**vtu**