A

Mini Project

On

AUTOMATIC DETECTION OF GENETIC DISEASES IN PEDIATRIC AGE USING PUPILLOMETRY

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

D. ANAND PAUL (207R1A0574)

M. ASHRITHA (207R1A0594)

MD. AZARUDDIN (217R5A0507)

Under the Guidance of

G. VIJAY KUMAR

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"AUTOMATIC DETECTION OF GENETIC DISEASES IN PEDIATRIC AGE USING PUPILLOMETRY"** being submitted by **D.ANANDPAUL(207R1A0574), M.ASHRITHA(207R1A0594) & MD.AZARRUDDIN(217R5A0507)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-24.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

G. Vijay Kumar
(Assistant Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEDGEMENT

A part from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard tomy guide **G. Vijay Kumar,** Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G.Vinesh Shanker, Dr. J. Narasimharao , Ms. Shilpa,& Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

D. ANAND PAUL (207R1A0574)
M. ASHRITHA (207R1A0594)
MD. AZARRUDDIN (217R5A0507)

ABSTRACT

Inherited retinal diseases cause severe visual deficits in children, and the diagnosis is challenging due to a wide range of clinical and genetic causes. The proposed CDSS uses Chromatic Pupillometry and Support Vector Machines to classify the extracted features from pupillometric data. The system achieved 84.6% accuracy, 93.7% sensitivity, and 78.6% specificity in diagnosing Retinitis Pigmentosa in pediatric subjects. The study is the first to apply machine learning to pupillometric data for diagnosing a genetic disease in pediatric age. The system could be further validated with a larger data pool, and different devices could be investigated to reduce movement artifacts.. This approach offers a non-invasive and reliable way to diagnose these diseases in pediatric patients.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The project, "Automatic Detection of Genetic Diseases in Pediatric Age Using Pupillometry," focuses on creating a specialized Clinical Decision Support System (CDSS) to detect genetic diseases in children, primarily targeting retinitis pigmentosa (RP). RP, a complex genetic retinal disease with over 200 causative genes, often leads to childhood blindness and poses diagnostic challenges. This project involves collecting and preprocessing pupillometric data, extracting essential features, and utilizing fine-tuned Support Vector Machines (SVMs) for analysis. An ensemble model combines SVM outputs from both eyes, enhancing sensitivity, and resulting in an accuracy of 84.6%, sensitivity of 93.7%, and specificity of 78.6%. Future plans include testing the system with different devices and exploring its potential in diagnosing a wider range of genetic pediatric diseases, with ongoing optimization efforts to establish it as an innovative diagnostic tool in pediatric healthcare.

1.2 PROJECT PURPOSE

The project's core purpose is to transform the diagnosis of inherited retinal diseases, especially in pediatric cases like retinitis pigmentosa. Traditional diagnostic methods are complex and often unsuitable for children. To address this, the project introduces an innovative Clinical Decision Support System using Chromatic Pupillometry and Machine Learning. By combining specialized hardware and custom software, the system offers accurate, non-invasive diagnosis. Its success aids in early disease detection, treatment, and opens doors for future machine learning applications in pediatric genetic disease diagnosis, improving young patients' healthcare.

1.3 PROJECT FEATURES

This project introduces groundbreaking features for diagnosing inherited retinal diseases in children. It combines Chromatic Pupillometry and Machine Learning for precise, non-invasive diagnoses. Key elements include a specialized medical pupillometer, a tailored machine learning system, and the use of Support Vector Machines (SVMs) to classify pupillometric data. An ensemble model, uniting SVMs from both eyes, improves sensitivity, achieving an 84.6% accuracy, 93.7% sensitivity, and 78.6% specificity. Remarkably, it's the pioneering application of machine learning to pupillometric data for pediatric genetic disease diagnosis. Future plans involve testing with diverse devices and expanding to other retinal diseases, ensuring its ongoing innovation in healthcare.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

The problem addressed is the complex and invasive diagnosis of inherited retinal diseases (IRDs) in pediatric patients, including retinitis pigmentosa. Current methods are not child-friendly and hindered by genetic diversity. The study aims to create a non-invasive Clinical Decision Support System (CDSS) using Chromatic Pupillometry and Machine Learning for accurate IRD diagnoses in children, offering a vital solution for pediatric genetic disease diagnosis.

2.2 EXISTING SYSTEM

The existing system for diagnosing genetic diseases in pediatric patients, particularly retinitis pigmentosa (RP), primarily relies on traditional clinical evaluation methods. These methods encompass a complex array of clinical tests, including invasive procedures, which are not always suitable for children. The conventional approach also faces difficulties in delivering timely and comprehensive screening for the wide range of causative genes involved in inherited retinal diseases (IRDs), posing significant challenges for early diagnosis and monitoring in pediatric populations .

2.2.1 DISADVANTAGES OF EXISTING SYSTEM

Following are the disadvantages of existing system:

- Invasive procedures: Some of the clinical tests used in the existing system are invasive, which may not be suitable for infants or young children.
- Time-consuming and expensive: The combination of clinical and genetic tests required for diagnosis can be time-consuming and expensive.
- Lack of conclusive results: Despite the complex testing, the diagnosis may not always be conclusive, leading to further testing and delays in treatment.

2.3 PROPOSED SYSTEM

This project presents a system for diagnosing inherited retinal diseases in pediatric patients is a Clinical Decision Support System (CDSS) that utilizes chromatic pupillometry and machine learning algorithms. This innovative CDSS incorporates a dedicated medical device (pupillometer) paired with a custom-designed machine learning decision support system to analyze pupillometric data. It extracts key features from the data and employs two Support Vector Machines (SVMs), one for each eye, to classify the information. This approach aims to offer a non-invasive, dependable, and potentially more accessible alternative to the current diagnostic methods for inherited retinal diseases in pediatric patients.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Provides a non-invasive and safe approach for diagnosing inherited retinal diseases in pediatric patients.
- Exploits the use of chromatic pupillometry, a technique that is increasingly used to assess outer and inner retina functions.
- Offers a more efficient and accurate clinical decision support system (CDSS) based on machine learning.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis:

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on a project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminaryinvestigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication that the system is economically possible for development.

2.4.2 TECHNICAL FEASIBILITY

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

2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Core I5 and above
- Hard disk : 40 GB and above
- RAM : 8GB and above
- Input devices : Keyboard, mouse.

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 8 and above
- Languages : Python 3.7
- Tools : Visual Studio, Anaconda - Jupyter, Spyder

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction.

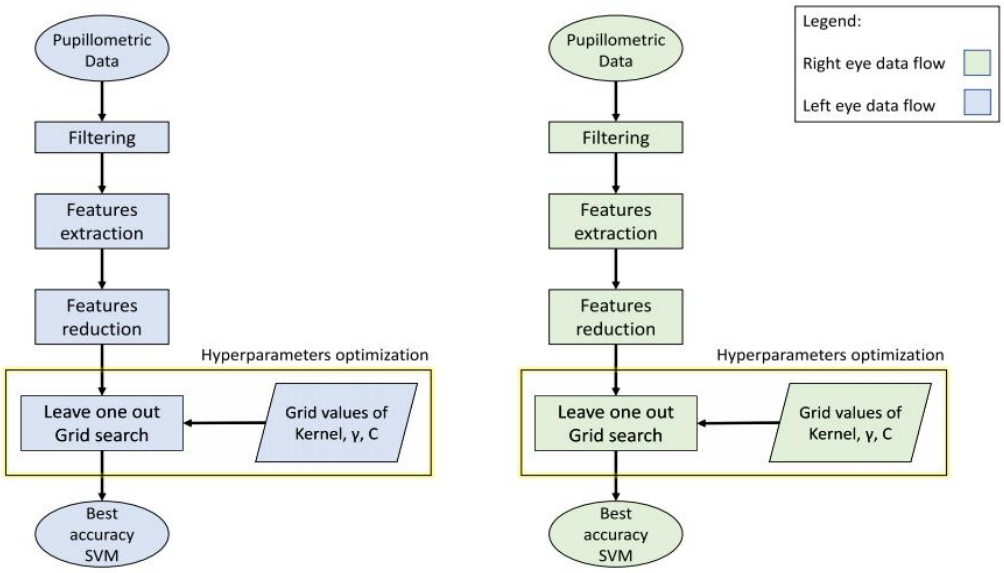


Figure 3.1: Project Architecture for Automatic Detection of Genetic Diseases in Pediatric Age using Pupillometry

3.2 DESCRIPTION

This project addresses the challenge of diagnosing inherited retinal diseases (IRDs) in children, aiming to overcome severe visual impairments they cause. It combines chromatic pupillometry and advanced machine learning to create a novel Clinical Decision Support System (CDSS). This system integrates a specialized pupillometer with custom machine learning, offering a non-invasive, reliable alternative for diagnosing IRDs in pediatric patients, with a focus on Retinitis Pigmentosa. Promising preliminary results demonstrate improved accuracy, sensitivity, and specificity, marking a pioneering use of machine learning in pediatric genetic disease diagnosis.

3.3 USE CASE DIAGRAM

This use case diagram represents how a system can be used to detect genetic diseases in pediatric patients using pupillometry. The patient undergoes a pupillometry test and uploads the data to the system. The system then analyzes the data and predicts the presence or absence of a genetic disease. The clinician reviews the prediction and makes a diagnosis. This system can help clinicians to diagnose genetic diseases in pediatric patients more accurately and efficiently. It can also help clinicians to identify patients who are at risk of developing genetic diseases.

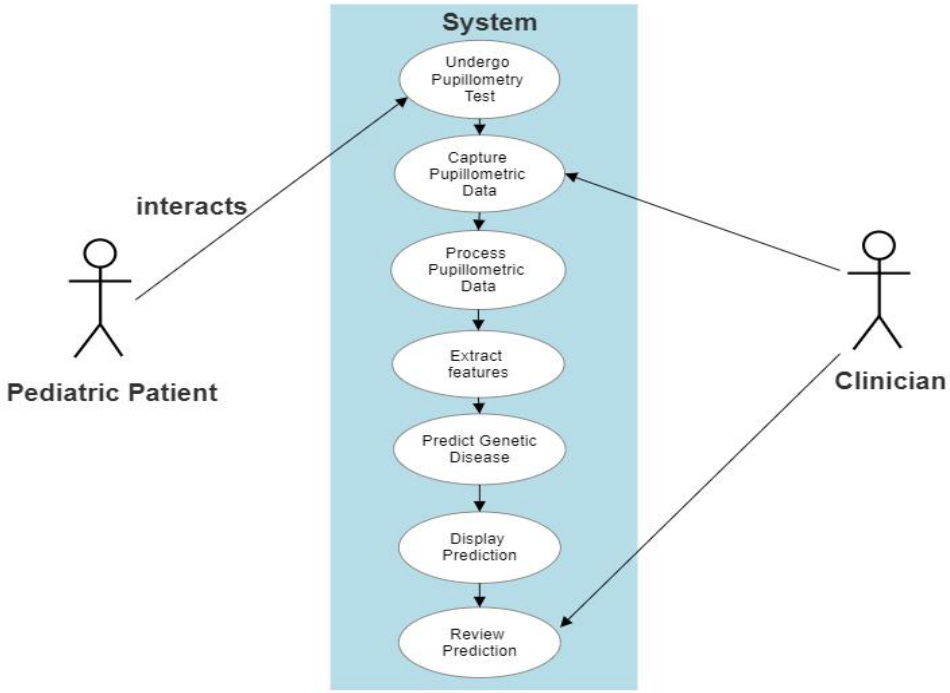


Figure 3.2: Use Case Diagram for Automatic Detection of Genetic Diseases in Pediatric Age using Pupillometry

3.4 CLASS DIAGRAM

A class diagram is a static structure diagram that illustrates a system's structure by presenting its classes, attributes, methods (operations), and object relationships.

In the provided class diagram for a system aimed at detecting genetic diseases in pediatric patients using pupillometry, there are four primary classes: "PediatricPatient" for storing patient information, "GeneticDisease" for disease-specific details,"PupillometryData" for recording test results, and "GeneticDiseaseDetectionSystem" for predicting diseases. Notably, "PediatricPatient" can be associated with multiple genetic diseases. This diagram forms a well-organized foundation for the efficient management of patient data, disease information, and disease prediction processes.

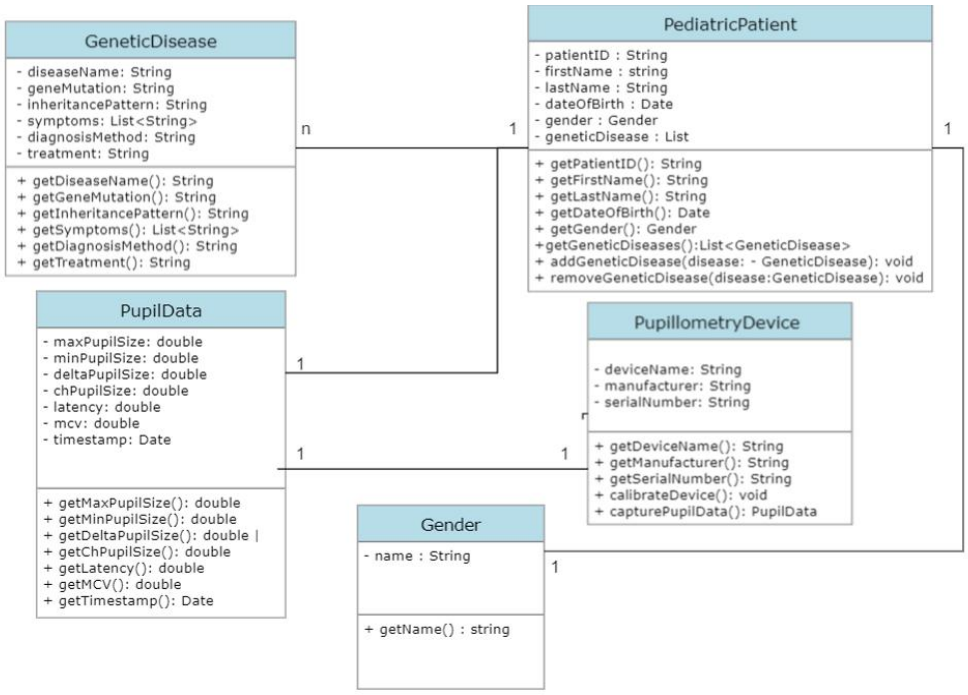


Figure 3.3: Class Diagram for Automatic Detection of Genetic Diseases in Pediatric Age using Pupillometry

3.5 SEQUENCE DIAGRAM

A sequence diagram visually depicts object interactions in chronological order, showcasing the involved objects and their message exchanges to execute a scenario. These diagrams are often linked to use case realizations in the system's logical development view.

The provided diagram illustrates the automated genetic disease detection process in pediatric patients using pupillometry. It initiates with the patient's pupillometry test and data upload. The system then proceeds to analyze the data, identifying patterns associated with genetic diseases, and subsequently presents a predictive outcome to the clinician. Clinicians leverage this information alongside their expertise to make accurate diagnoses, ultimately facilitating early disease detection and enhancing pediatric care.

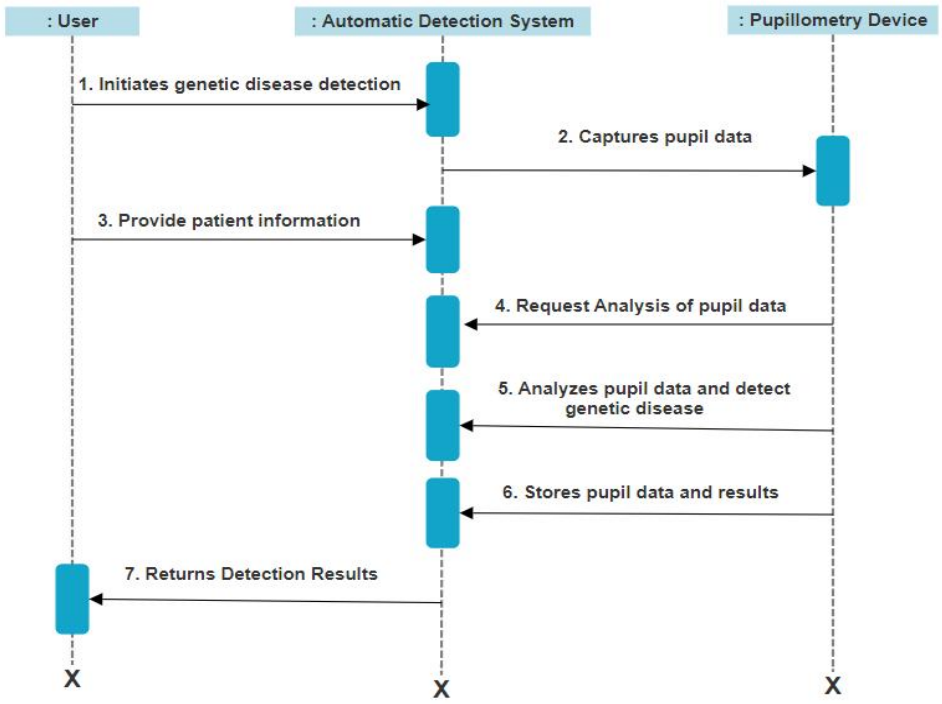


Figure 3.4: Sequence Diagram for Automatic Detection of Genetic Diseases in Pediatric Age using Pupillometry

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

This diagram illustrates the automatic genetic disease detection process in pediatric patients using pupillometry. It outlines sequential steps, including data capture, preprocessing, feature extraction, analysis, machine learning-based detection, report generation, and result storage. Notably, the system can handle multiple patients simultaneously, offering potential for early disease identification and improved pediatric care.

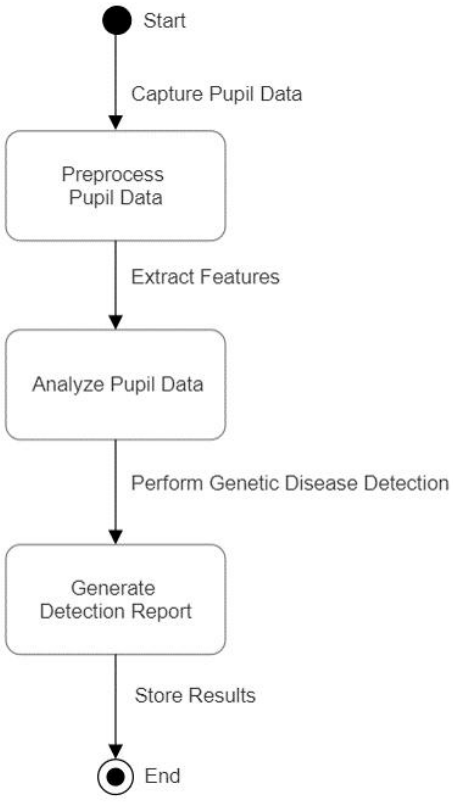


Figure 3.5: Activity Diagram for Automatic Detection of Genetic Diseases in Pediatric Age using Pupillometry

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
from tkinter import messagebox
from tkinter import *
from tkinter.filedialog import askopenfilename
from tkinter import simpledialog
import tkinter
import numpy as np
from tkinter import
filedialog
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.ensemble
import VotingClassifier
import os
from sklearn.metrics import confusion_matrix
from sklearn_extensions.extreme_learning_machines.elm import
GenELMClassifier
from sklearn_extensions.extreme_learning_machines.random_layer import
RBFRandomLayer, MLPRandomLayer
from keras.models import Sequential
from keras.layers.core import Dense,Activation,Dropout,Flatten
from sklearn.preprocessing import OneHotEncoder
import keras.layers
from sklearn.preprocessing import normalize
 from keras.layers import Bidirectional
main = tkinter.Tk()
```

```
main.title("Automatic Detection of Genetic Diseases in Pediatric Age Using
Pupillometry")
main.geometry("1300x120")
 global filename
global classifier
global left_X_train, left_X_test, left_y_train, left_y_test
global right_X_train, right_X_test, right_y_train, right_y_test
global left_X, left_Y
global left_pupil
global right_pupil
global count
global left
global right
global ids
global left_svm_acc
global right_svm_acc
global left_classifier
global right_classifier
global classifier
global ensemble_acc
global elm_acc
global lstm_acc,bilstm_acc

def upload():
    global filename
    filename = filedialog.askdirectory(initialdir = ".")
    pathlabel.config(text=filename)
    text.delete('1.0', END)
    text.insert(END,'Pupillometric  dataset loaded\n')

def filtering():
   global left_pupil
```

```
global right_pupil
global count
global left

global right
global ids
left_pupil = []
right_pupil = []
count = 0
left = 'Patient_ID,MAX,MIN,DELTA,CH,LATENCY,MCV,label\n'
right = 'Patient_ID,MAX,MIN,DELTA,CH,LATENCY,MCV,label\n'
ids = 1
for root, dirs, directory in os.walk('dataset'):
    for i in range(len(directory)):
        filedata = open('dataset/'+directory[i], 'r')
        lines = filedata.readlines()
        left_pupil.clear()
        right_pupil.clear()
        count = 0
        for line in lines:
            line = line.strip()
            arr = line.split("\t")
            if len(arr) == 8:
                if arr[7] == '.....':
                    left_pupil.append(float(arr[3].strip()))
                    right_pupil.append(float(arr[6].strip()))
                    count = count + 1;
                    if count == 100:
                        left_minimum = min(left_pupil)
                        right_minimum = min(right_pupil)
                        left_maximum = max(left_pupil)
                        right_maximum = max(right_pupil)
                        left_delta =  left_maximum - left_minimum
```

```
                right_delta = right_maximum - right_minimum
                left_CH = left_delta / left_maximum
                right_CH = right_delta / right_maximum
                latency = 0.5
                left_MCV = left_delta/(left_minimum - latency)
                right_MCV = right_delta/(right_minimum - latency)
                count = 0
                left_pupil.clear()
                right_pupil.clear()
                  if left_minimum > 500 and left_maximum > 500:
                        left+=str(ids)+","+str(left_maximum)+","+str(left_minimum)
                        +","+str(left_delta)+","+str(left_CH)+","+str(latency)+","+str
                        (left_MCV)+",1\n"
                  else:
                       left+=str(ids)+","+str(left_maximum)+","+str(left_minimum)+",
                       "+str(left_delta)+","+str(left_CH)+","+str(latency)+","+str(left_
                       MCV)+",0\n"
                  if right_minimum > 500 and right_maximum > 500:
                         right+=str(ids)+","+str(right_maximum)+","+str(right_mini
                         mum)+","+str(right_delta)+","+str(right_CH)+","+str(latenc
                         y)+","+str(right_MCV)+",1\n"
                  else:
                      right+=str(ids)+","+str(right_maximum)+","+str(right_minimum)
                      +","+str(right_delta)+","+str(right_CH)+","+str(latency)+","+str(r
                      ight_MCV)+",0\n"
                   ids = ids + 1
        filedata.close()

    text.delete('1.0', END)
    text.insert(END,'Features filteration process completed\n')
    text.insert(END,'Total patients found in dataset : '+str(ids)+"\n")
    def featuresExtraction():
```

```
                f = open("left.txt", "w")
                f.write(left)
                f.close()
                f = open("right.txt", "w")
                f.write(right)
                f.close()
                 text.delete('1.0', END)
                text.insert(END,'Both eye pupils extracted features saved inside
  left.txt and right.txt files \n')
                 text.insert(END,"Extracted features are \nPatient ID, MAX, MIN,
  Delta, CH, Latency, MDV, CV and MCV\n")

def featuresReduction():
        text.delete('1.0', END)
        global left_X, left_Y
        global left_X_train, left_X_test, left_y_train, left_y_test
        global right_X_train, right_X_test, right_y_train, right_y_test
        left_pupil =  pd.read_csv('left.txt')
        right_pupil =  pd.read_csv('right.txt')
        cols = left_pupil.shape[1]
       left_X = left_pupil.values[:, 1:(cols-1)]
        left_Y = left_pupil.values[:, (cols-1)]
       right_X = right_pupil.values[:, 1:(cols-1)]
       right_Y = right_pupil.values[:, (cols-1)]
       indices = np.arange(left_X.shape[0])
        np.random.shuffle(indices)
        left_X = left_X[indices]
        left_Y = left_Y[indices]
       indices = np.arange(right_X.shape[0])
       np.random.shuffle(indices)
       right_X = right_X[indices]
        right_Y = right_Y[indices]
```

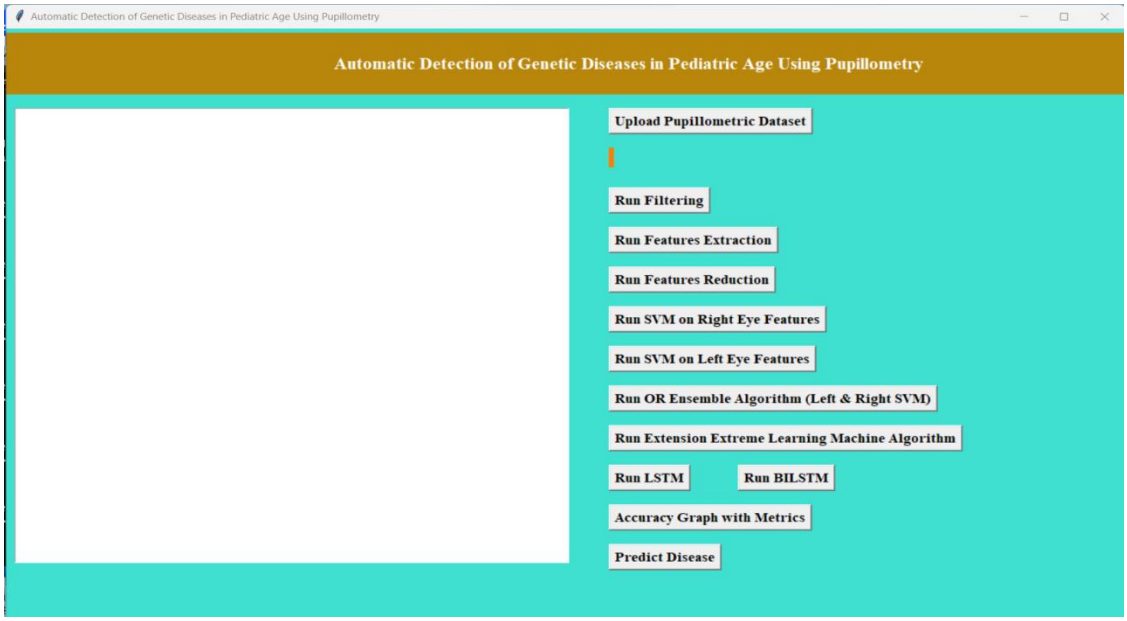
```
            left_X = normalize(left_X)
            right_X = normalize(right_X)
            left_X_train, left_X_test, left_y_train, left_y_test = train_test_split(left_X,
            left_Y, test_size = 0.2,random_state=42)
            right_X_train, right_X_test, right_y_train, right_y_test
            train_test_split(right_X, right_Y, test_size = 0.2,random_state=42)
             text.insert(END,"Left pupil features training size : "+str(len(left_X_train))+"
            & testing size : "+str(len(left_X_test))+"\n")
            text.insert(END,"Right pupil features training size :
            "+str(len(right_X_train))+" & testing size : "+str(len(right_X_test))+"\n")

            plt.figure(figsize=(10,6))
            plt.grid(True)
            plt.xlabel('Time')
            plt.ylabel('Diameter')
            plt.plot(left_pupil['MAX'], 'ro-', color = 'indigo')
            plt.plot(right_pupil['MAX'], 'ro-', color = 'green')
            plt.legend(['Left Pupil', 'Right Pupil'], loc='upper left')
            plt.title('Pupil Diameter Graph')
            plt.show()

    def prediction(X_test, cls):
        y_pred = cls.predict(X_test)
        for i in range(len(X_test)):
          print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
        return y_pred


    def rightSVM():
        global right_classifier
        text.delete('1.0', END)
        global right_svm_acc
        temp = []
```

```
    for i in range(len(right_y_test)):
        temp.append(right_y_test[i])
    temp = np.asarray(temp)
    right_classifier = svm.SVC()
    right_classifier.fit(right_X_train, right_y_train)
    text.insert(END,"Right pupil SVM Prediction Results\n")
    prediction_data = prediction(right_X_test, right_classifier)
    right_svm_acc = accuracy_score(temp,prediction_data)*100
    text.insert(END,"Right pupil SVM Accuracy : "+str(right_svm_acc)+"\n")

    cm = confusion_matrix(temp, prediction_data)
    sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
    text.insert(END,'Right pupil SVM Algorithm Sensitivity : '+str(sensitivity)+"\n")
    specificity = cm[1,1]/(cm[1,0]+cm[1,1])
    text.insert(END,'Right pupil SVM Algorithm Specificity : '+str(specificity)+"\n")

def leftSVM():
    global left_classifier
    text.delete('1.0', END)
    global left_svm_acc
    temp = []
    for i in range(len(left_y_test)):
        temp.append(left_y_test[i])
    temp = np.asarray(temp)
    left_classifier = svm.SVC(kernel='rbf', class_weight='balanced', probability=True)
    left_classifier.fit(left_X_train, left_y_train)
    text.insert(END,"Left pupil SVM Prediction Results\n")
    prediction_data = prediction(left_X_test, left_classifier)
    left_svm_acc = accuracy_score(temp,prediction_data)*100
    text.insert(END,"Left pupil SVM Accuracy : "+str(left_svm_acc)+"\n")

    cm = confusion_matrix(temp, prediction_data)
    sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
```

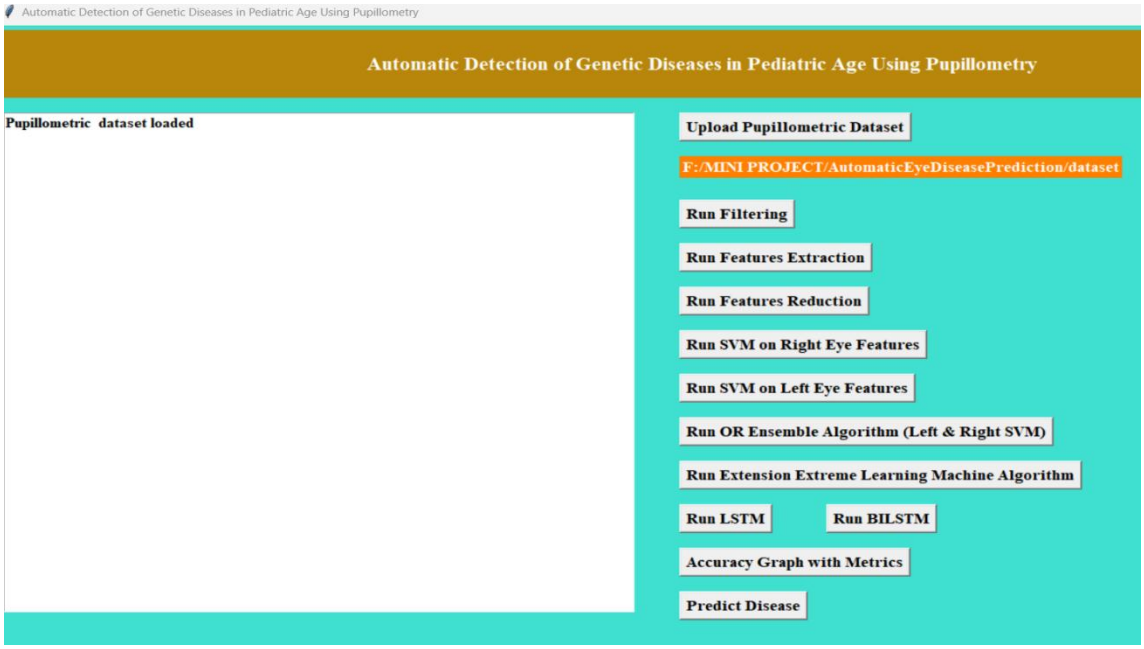
```
    text.insert(END,'Left pupil SVM Algorithm Sensitivity : '+str(sensitivity)+"\n")
    specificity = cm[1,1]/(cm[1,0]+cm[1,1])
    text.insert(END,'Left pupil SVM Algorithm Specificity : '+str(specificity)+"\n")



def ensemble():
    global classifier
    global ensemble_acc
    text.delete('1.0', END)
    trainX = np.concatenate((right_X_train, left_X_train))
    trainY = np.concatenate((right_y_train, left_y_train))
    testX = np.concatenate((right_X_test, left_X_test))
     testY = np.concatenate((right_y_test, left_y_test))
    left_classifier = svm.SVC(kernel='linear', class_weight='balanced', probability=True)
                    right_classifier = svm.SVC(kernel='linear', class_weight='balanced',
   probability=True)
    temp = []
    for i in range(len(testY)):
       temp.append(testY[i])
    temp = np.asarray(temp)
    classifier = VotingClassifier(estimators=[
        ('SVMLeft', left_classifier), ('SVMRight', right_classifier)], voting='hard')
    classifier.fit(trainX, trainY)
    text.insert(END,"Optimized Ensemble Prediction Results\n")
    prediction_data = prediction(testX, classifier)
    ensemble_acc =  (accuracy_score(temp,prediction_data)*100)
    text.insert(END,"Ensemble OR Accuracy : "+str(ensemble_acc)+"\n")
    cm = confusion_matrix(temp, prediction_data)
    sensitivity = cm[0,0]/(cm[0,0]+cm[0,1])
    text.insert(END,'Right pupil Ensemble OR SVM Algorithm Sensitivity :
    '+str(sensitivity)+"\n")
    specificity = cm[1,1]/(cm[1,0]+cm[1,1])
```

```
        text.insert(END,'Right pupil Ensemble OR SVM Algorithm Specificity :
        '+str(specificity)+"\n")

def graph():
    height = [right_svm_acc,left_svm_acc,ensemble_acc,elm_acc,lstm_acc,bilstm_acc]
    bars = ('Right Pupil SVM Acc','Left Pupil SVM Acc','Ensemble OR (L & R Pupil)
    Acc','ELM Acc','LSTM Acc','BI-LSTM Acc')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()
```

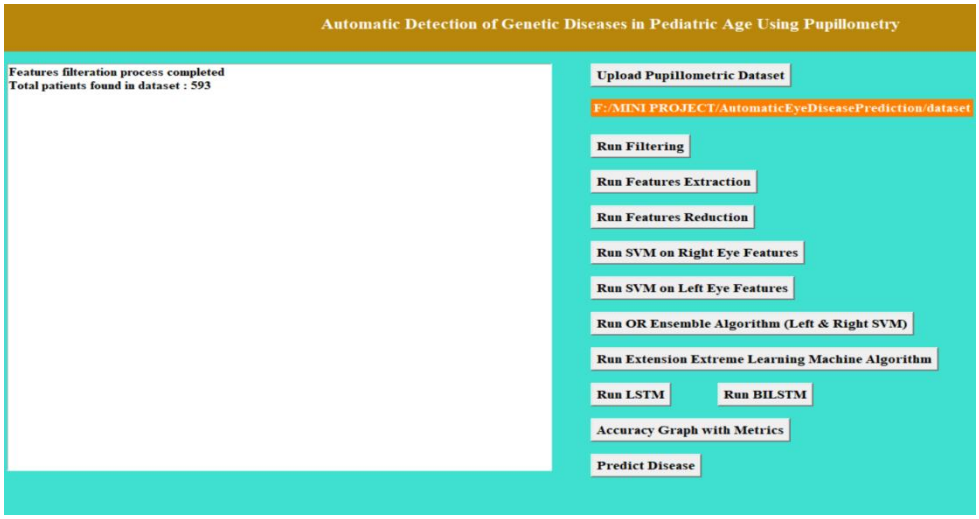
5. SCREENSHOTS



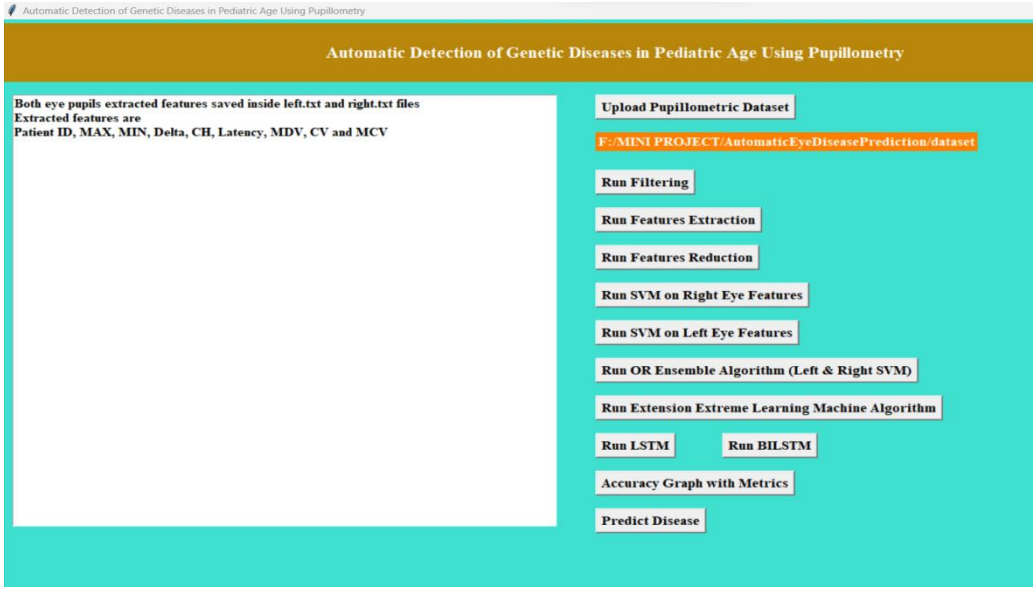
Screenshot 5.1 : Home Screen of the Project



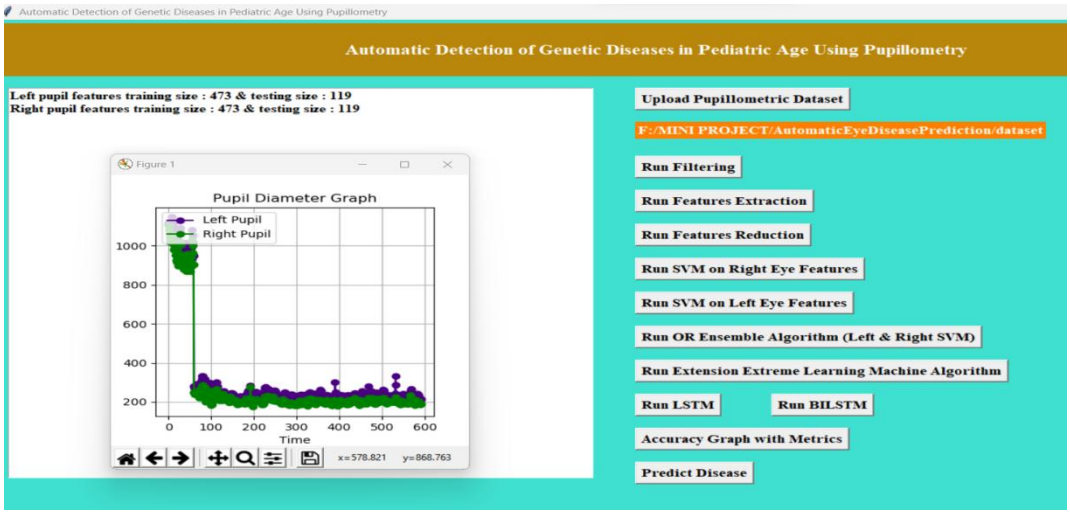
Screenshot 5.2 : Upload Pupillometric Dataset



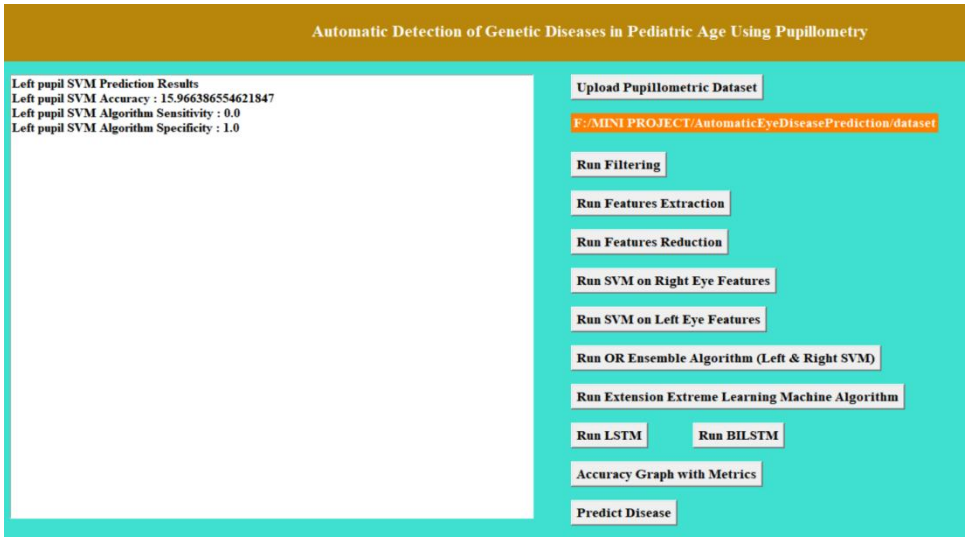
Screenshot 5.3.1 : Run Filtering



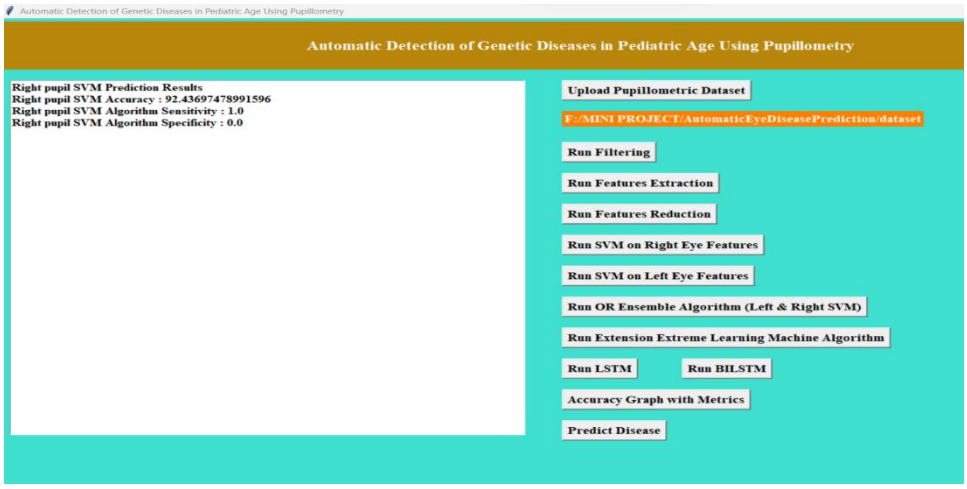
Screenshot 5.3.2 : Run Feature Extraction



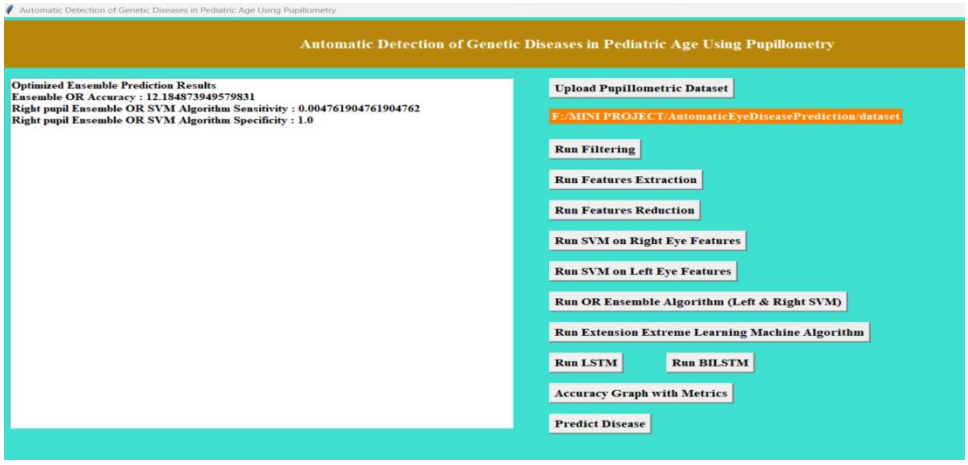
Screenshot 5.3.3 : Run Feature Reduction



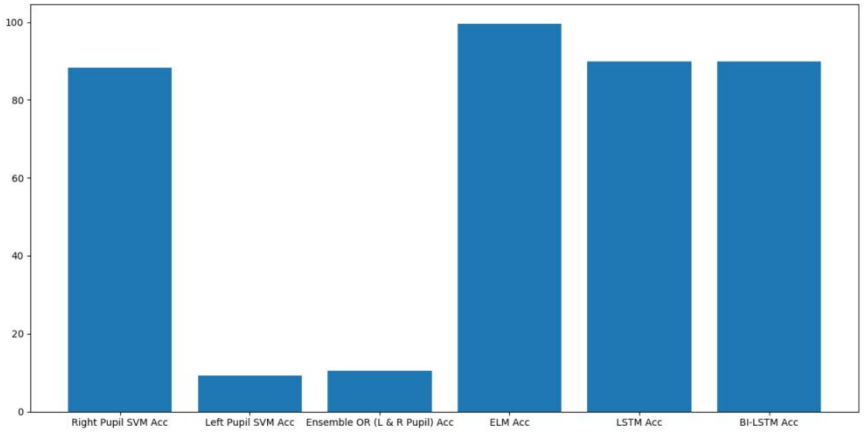
Screenshot 5.4.1 : Run SVM on left eye features



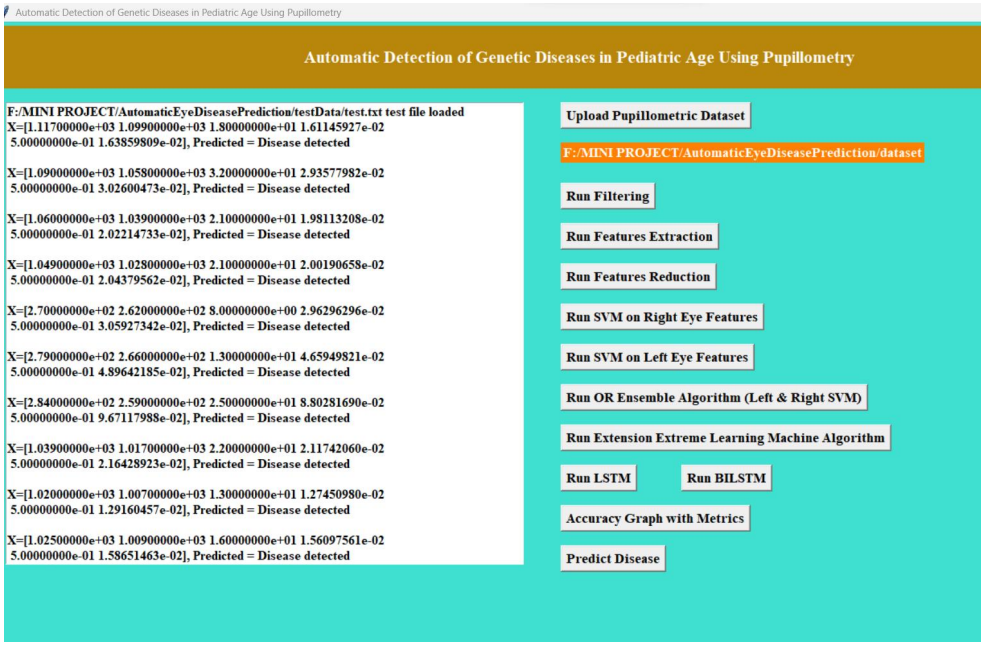
Screenshot 5.4.2 : Run SVM on right eye features



Screenshot 5.5 : Run OR Ensemble Algorithm (Left & Right SVM)



Screenshot 5.6 : Accuracy Graph with Metrics



Screenshot 5.7 : Predict Disease

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

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

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.
Functional testing is centered on the following items:

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

Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output	Result
1	Data Upload	To verify data upload functionality.	Pupillometry data files from pediatric patients.	Confirmation of successful data upload.	Pass
2	Data Preprocessing	To ensure noise and artifacts are removed from data.	Raw pupillometry data with noise and artifacts.	Cleaned and preprocessed data.	Pass
3	Feature Extraction	To validate extraction of relevant features from data	Preprocessed pupillometry data.	Extracted features (e.g., pupil size, latency).	Pass
4	Pattern Identification	To confirm correct identification of disease-related patterns	Extracted features	Identification of patterns associated with diseases.	Pass
5	Disease Prediction	To verify accurate prediction of genetic diseases	Identified patterns.	Predictions of presence/absence of genetic diseases.	Pass

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

The Clinical Decision Support System (CDSS) proposed for automatic genetic disease detection in pediatric patients through pupillometry represents a non-invasive, accurate, and efficient diagnostic breakthrough, especially for inherited retinal diseases. Leveraging chromatic pupillometry and machine learning algorithms, it offers a more objective and dependable diagnosis, crucial for infants and young children. The system's robust performance evaluation, with 84.6% accuracy, 93.7% sensitivity, and 78.6% specificity, demonstrates its clinical utility. Overall, this CDSS holds great promise for enhancing the diagnostic process and significantly improving the quality of life for patients and their families affected by inherited retinal diseases.

.

7.2 FUTURE SCOPE

The future scope of the Automatic Genetic disease detection in pediatric patients through pupillometry is highly promising. As machine learning and AI continue to advance, the system's accuracy and predictive capabilities are expected to improve, enhancing early disease detection. Integration with telemedicine can extend its reach to remote areas, ensuring timely diagnoses for a broader pediatric population. Collaboration with geneticists may uncover more detectable genetic diseases, making the system even more comprehensive. This project has the potential to revolutionize pediatric healthcare and significantly improve the quality of life for children worldwide.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

[1] X.-F. Huang, F. Huang, K.-C. Wu, J. Wu, J. Chen, C.-P. Pang, F. Lu, J. Qu, and Z.-B. Jin, ‘‘Genotype–phenotype correlation and mutation spectrum in a large cohort of patients with inherited retinal dystrophy revealed by next-generation sequencing,’’ Genet. Med., vol. 17, no. 4, pp. 271–278, Apr. 2015.

[2] J. C. Park, A. L. Moura, A. S. Raza, D. W. Rhee, R. H. Kardon, and D. C. Hood, ‘‘Toward a clinical protocol for assessing rod, cone, and melanopsin contributions to the human pupil response,’’ Invest. Ophthal- mol. Vis. Sci., vol. 52, no. 9, pp. 6624–6635, Aug. 2011. 6.

[3] A. Kawasaki, F. L. Munier, L. Leon, and R. H. Kardon, ‘‘Pupillometric quantification of residual rod and cone activity in Leber congenital amau- rosis,’’ Arch. Ophthalmol., vol. 130, no. 6, pp. 798–800, Jun. 2012.

[4] E. Iadanza, R. Fabbri, A. Luschi, F. Gavazzi, P. Melillo, F. Simonelli, and M. Gherardelli, ‘‘ORÁO: RESTful cloud-based ophthalmologic med- ical record for chromatic pupillometry,’’ in Proc. IFMBE, vol. 73, 2020, pp. 713–720.

8.2 GITHUB LINK

https://github.com/Anandpaul99s/Automatic-Detection-of-Genetic-Diseases-in-Pediatric-Age-Using-Pupillometry