PATIENT IDENTIFICATION AND DISEASE PREDICTION

A report submitted to

RAMAIAH INSTITUTE OF TECHNOLOGY

Bengaluru

ISP SENIOR PROJECT

as partial fulfillment of the requirement for

Bachelor of Engineering (B.E) in Information Science and Engineering

By

Mohammed Salmaan Khan (USN - 1MS17IS065)
Ahmed Uddin Khalid Syed (USN - 1MS17IS008)
Ayush Agrawal (USN - 1MS17IS031)
Ayush Kala (USN - 1MS17IS032)

under the guidance of

DR. SIDDESH G M

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY

May 2021

Department of Information Science and Engineering

Ramaiah Institute of Technology

Bengaluru – 54

CERTIFICATE

This is to certify that Ahmed Uddin Khalid Syed (USN-1MS17IS008), Mohammed Salmaan Khan (USN- 1MS17IS065), Ayush Agrawal (USN-1MS17IS031) And Ayush Kala (USN-1MS17IS032) who were working for their ISP, SENIOR PROJECT under my guidance, have completed the work as per my satisfaction with the topic PATIENT IDENTIFICATION AND DISEASE PREDICTION. To the best of my understanding the work to be submitted in dissertation does not contain any work, which has been previously carried out by others and submitted by the candidates for themselves for the award of any degree anywhere.

(Guide)
Dr. Siddesh G M
Associate Professor, Dept. of ISE

(Head of the Department)
Dr. Yogish H K
Professor & Head, Dept. of ISE

(Examiner 1)

Name

Signature

(Examiner 2)

Department of Information Science and Engineering

Ramaiah Institute of Technology

Bengaluru - 54

DECLARATION

We hereby declare that the entire work embodied in this ISP SENIOR PROJECT report has been carried out by us at Ramaiah Institute of Technology under the supervision of **Dr. Siddesh G M**. This project report has not been submitted in part or full for the award of any diploma or degree of this or any other University.

MOHAMMED SALMAAN KHAN (USN- 1MS17IS065)

AHMED UDDIN KHALID SYED (USN- 1MS17IS008)

AYUSH AGRAWAL (USN- 1MS17IS031)

AYUSH KALA (USN- 1MS17IS032)

Acknowledgements

The Successful completion of this project work was made possible through the valuable contribution of a number of people. To say thank you to all of them is not even enough to express our gratitude. We are overwhelmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put these ideas, well above the level of simplicity and into something concrete.

Our first debt of gratitude must go to our guide Dr. Siddesh GM , Associate professor of Dept. of Information Science and Engineering, MSRIT who gave us the golden opportunity to do this wonderful project on the topic "Patient Identification and Disease Prediction" , which also helped us in doing a lot of Research and we came to know about so many new things. We are thankful to him for his valuable and inspiring guidance throughout the course of this work. He has driven us to an everlasting debt of gratitude through his valuable guidance and support in bringing out this work.

We take this platform to thank the Head of the Department, Dr. Yogish H K for his valuable suggestions and support to carry out the entire project work in the facilities of the Department.
Our special thanks goes to our parents for their blessings and they have been a great source of inspiration for us. Any attempt at any level can’t be satisfactorily completed without the support and guidance of friends. We’d like to thank them for helping us in gathering different information, collecting data and guiding us from time to time in making this project.

Abstract

Since the onset of the Covid pandemic, hospitals have become chaotic, overcrowded and disorganized. The healthcare infrastructure is unable to keep up with the sudden surge of patients. Patients, covid and non-covid alike have to undergo the tedious and time consuming process of filling physical forms that fail to accurately reflect their health history. This leads to a lot of inconsistencies during treatment as the doctors in charge aren't aware of their patients' health histories. This project provides a contact-less registration process for patients using facial recognition which is fast and non-cumbersome. The centralized record system also helps keep a consistent track record of the patient's health history/treatments and reduces the chance of misdiagnosis. This Project also provides two disease prediction models, namely Diabetes Prediction and Heart disease Prediction that help patients perform self-assessment and doctors confirm their medical diagnosis.

Table of Contents

1. **Introduction** **1**

1.1 Introduction 1

1.2 Motivation 2

1.3 Constraint and Requirements 2

1.4 Problem Statement 3

1.5 Scope and Objectives 3

1.6 Proposed Model 4

1.7 Organization of Report 5

2. Literature Survey **6**

2.1 Literature Survey 6

2.2 Outcome of the Literature Survey 8

3. System Analysis and Design **10**

3.1 Hardware Requirements 10

3.2 Software Requirements 11

3.2.1 Django Framework 11

3.2.2 Colaboratory 12

3.2.3 SQLite 12

3.3 Functional Requirements 12

3.4 Non-Functional Requirements 13

3.5 System Design 14

3.5.1 Proposed System 15

3.5.2 Use Case Diagram 16
3.5.3 Sequence Diagram 17
3.6 Datasets 18
3.6.1 Pima Indian Diabetes Dataset 18
3.6.2 Heart Disease UCL 19
4. Modeling and Implementation 20
4.1 Structure of the System 20
4.2 Modules and Algorithms 21
4.2.1 MTCNN 21
4.2.2 Face Net 22
4.2.3 Random Forest Algorithm 23
4.2.4 Support Vector Machine Algorithm 25
4.3 Outlier Detection 26
4.4 Oversampling 28
4.5 Detailed Workflow 29
4.5.1 Algorithm 29
5. Testing, Results and Discussion 31
5.1 Testing 31
5.2 Results and Discussion 35
5.2.1 Face Identification and Recognition 35
5.2.2 Diabetes Prediction 37
5.2.3 Heart Disease Prediction 41
6. Conclusion and Future Work 45
6.1 Conclusion 45
6.2 Future Work 45
7. Bibliography 46

List of Figures

Chapter 3

3.1 Use Case Diagram

3.2 Sequence Diagram

Chapter 4

4.1 Structure of the System

4.2 Outliers Detection and Removal for Glucose Attribute

4.3 Outliers Detection and Removal for Blood Pressure Attribute

4.4 Outliers Detection and Removal for BMI Attribute

4.5 Bar plots of Oversampling

4.6 Detailed WorkFlow

Chapter 5

5.1 Home Page

5.2 Information Form

5.3 Face Recognition Results

5.4 Diabetes Prediction Form and Result

5.5 Heart Disease Prediction Form and Result

5.6 Comparison between face detection algorithm to process 699 frames

5.7 Comparison between face detection algorithm w.r.t time

5.8 Accuracy of Diabetes Model

5.9 F1-Score of Diabetes Model

5.10 Precision of Diabetes Model

5.11 Recall of Diabetes Model

5.12 Confusion Matrix for Diabetes Model

5.13 Precision of Heart Disease Prediction Model

5.14 F1-Score of Heart Disease Prediction Model

5.15 Accuracy of Heart Disease Prediction Model

5.16 Recall of Heart Disease Prediction Model

5.17 Confusion Matrix for Heart Disease Prediction Mode

List of Tables

Table 1: Comparison between Diabetes Prediction Models

Table 2: Comparison between Heart Disease Prediction Models

Chapter 1

Introduction

1.1 Introduction

The entire process of patient registration at a hospital is cumbersome, repetitive and requires contact. Each time a patient visits a hospital, they are required to fill a form and update their information. Furthermore, doctors do not have a centralized record system from which they can access patient history. Also, there is no way a patient can take a self-analysis test to find out whether they have diabetes or heart disease. This project hopes to reduce the time it takes for registration and help maintain a consistent record of patient history. This project also hopes to serve as a self-analysis test for people to understand if they're at risk of diabetes and heart disease and also serves as a confirmatory test for doctors to check whether they got their diagnosis with regards to diabetes and heart disease right.

A facial recognition system is a technology capable of matching a human face from a video frame against a database of faces and is typically employed to authenticate users through ID verification services. It works by pinpointing and measuring facial features from a given image.

Machine Learning is an emerging approach that helps in prediction and diagnosis of a disease. Medical science has large amounts of data growth per year. Due to an increased amount of data growth in the medical and healthcare fields, accurate analysis on medical data is seen to benefit early patient care. Machine Learning finds hidden pattern information in the huge amounts of medical data.

Diabetes is a disease that occurs when your blood glucose, also called blood sugar, is too high. Blood glucose is your main source of energy and comes from the food you eat. Insulin, a hormone made by the pancreas, helps glucose from food get into your cells to be used for energy.

Heart disease refers to any condition affecting the heart. There are many types, some of which are preventable. Unlike cardiovascular disease, which includes problems with the entire circulatory system, heart disease affects only the heart.

1.2 Motivation

The inefficient and traditional methods of registration and storing relevant medical documents at hospitals was the inspiration for this project. This inefficiency was brought to light during the Covid situation when hospitals were chaotic and disorganized. This project looks to fix that inefficiency by allowing a faster registration process at hospitals and providing a consistent view of personal details and medical history. Diabetes and Heart disease affect roughly 11.8% and 5.8% of the population respectively. A lot of these cases go undiagnosed due to misdiagnosis or general negligence, we hope to change that with this project. This project aims at providing patients with an easy self-diagnosis that does not require a visit to the hospital. It also provides doctors with a fast and easy confirmatory test to support their medical diagnosis.

1.3 Constraints and Requirements

- Unavailability of high-quality cameras leading to poor facial recognition. We need to convince hospitals to invest in high quality cameras.

- Inaccuracy of facial recognition algorithm leading to mis-matched patient records. The current accuracy of our system is roughly 81%, we need to drastically improve this before real world application.

- Changing appearances of patients between visits could lead to low accuracy facial recognition.

- Inaccuracy of disease prediction algorithm could lead to the wrong diagnosis of Diabetes and Heart Disease. Furthermore, the model relies on patients to enter the right details for diagnosis.

1.4 Problem Statement

Since the past couple of years, due to the pandemic, we have seen a lapse in our healthcare system, right from the admission process to the treatment in between and eventually the discharge. Patients are required to undergo the tedious process of filling physical forms that fail to accurately reflect their health history. This leads to a lot of time consuming and error inducing processes that directly impact the treatment process. The unavailability of doctors during the pandemic was especially hard for non-covid patients who couldn't get their symptoms diagnosed by a doctor or were too worried to go to the hospital, as they were scared of catching the infection, because of this, people with genuine problems such as diabetes and heart disease that needed to be diagnosed and treated immediately couldn't get the medical attention necessary.

1.5 Scope and Objectives

Face Recognition is becoming the new norm in the healthcare sector. This project looks to improve the existing healthcare infrastructure by helping hospitals deal with excessive patient load in a fast, organized manner and allow doctors to get a consistent view of the patient's history. Machine learning methods are widely used in predicting diabetes and heart disease but the existing methods are too complex. The project also aims to allow patients/doctors to predict diabetes and heart disease based on the patient's basic details and health history.

The key functions performed include:

- Facial recognition using Face net.
- Diabetes prediction using Random Forest Classifier.
- Heart disease prediction using SVM.

Objectives:

- To design a system that recognizes facial patterns.
- To have the system match the facial patterns to the ones stored in the database.
- To have the system display the stored patient details upon recognition.
- To accurately predict diabetes and heart disease based on patient history.
- To make the proposed system cost effective, efficient and reliable.

1.6 Proposed Model

We use Face Net for facial recognition in this project. Face Net is one of the new methods in face recognition technology. This method is based on a deep convolutional network and triplet loss training to carry out training data, but the training process requires complex computing and a long time. By integrating the TensorFlow learning machine and pre-trained model, the training time needed is drastically reduced. Through this project, we also aim to develop a system which can perform early prediction of diabetes for a patient with higher accuracy by using the Random Forest algorithm in machine learning techniques. Random Forest algorithms are often used for each classification and regression tasks and also it is a type of ensemble learning method. The accuracy level is greater when compared to other algorithms. The proposed model gives the best results for diabetic prediction and the result showed that the prediction system is capable of predicting the diabetes disease effectively, efficiently and most importantly, instantly. The system is also capable of predicting heart disease in patients. We use Support Vector Machine which is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs.

1.7 Organization of Report

Chapter 1: It is an introduction to the project. It lists the motivation behind why this project was chosen, what the scope of the project is, what outputs we hope to get, what challenges the model faces and the basic gist of the model that we hope to develop.

Chapter 2: We review several papers that worked on projects that are similar to ours, so that we have an idea on how to go about designing and implementing our model and understand what the problems with the already existing models are.

Chapter 3 : We discuss the design of the system that would be necessary to implement the project. The descriptions include hardware, software and the storage that have been used by us. We also discuss all the functional and non-functional requirements that the project would be expected to meet along with the necessary diagrams that explain the architecture and working of our model.

Chapter 4: We describe how the implementation of the project took place and how it fulfills the expectations set in the design phase. We list the step-by-step procedure that helps navigate our model from start to finish and define the algorithms that have been used to obtain the desired results.

Chapter 5: We examine the outputs produced by the model and analyze the results obtained to check whether it does everything that we expect it to do with an acceptable accuracy and within the desired timeframe. We also analyze a few key metrics that help give an insight into the functioning of the model.

Chapter 6: We summarize the report and discuss the key findings from this project. We also discuss the limitations of the model and how modifying it slightly can give it multiple real-life applications, thereby exponentially improving the healthcare industry

Chapter 2
Literature Survey

Literature review is the process of analyzing, summarizing, organizing and presenting novel conclusions from the results of technical review of a large number of recently published scholarly articles. The results of the literature review can contribute to the body of knowledge when peer reviewed and published as survey articles.

2.1 Literature Survey

K. VijiyaKumar [1] proposed a system which can perform an early prediction of diabetes for a patient with a higher accuracy by using the Random Forest algorithm in machine learning techniques. Random Forest algorithms are often used for classification and regression tasks and also it is a type of ensemble learning method. The accuracy level is greater when compared to other algorithms. The proposed model gives the best results for diabetic prediction and the result showed that the prediction system is capable of predicting the diabetes disease effectively, efficiently and most importantly, instantly.

Harshit Jindal, Sarthak Agrawal, Rishabh Khera, Rachna Jain and Preeti Nagrath [2] focused on which patient is more likely to have a heart disease based on various medical attributes. They prepared a heart disease prediction system to predict whether the patient is likely to be diagnosed with a heart disease or not using the medical history of the patient. They used different algorithms of machine learning such as logistic regression and KNN to predict and classify the patient with heart disease. The strength of the proposed model was quite satisfying and was able to predict evidence of having a heart disease in a particular individual by using KNN and Logistic Regression which showed a good accuracy in comparison to the previously used classifier such as naive bayes etc. The Given heart disease prediction system enhances medical care and reduces the cost.

Muhammad Azeem Sarwar, Nasir Kamal, Wajeeha Hamid and Munam Ali Shah [3] discuss the predictive analytics in healthcare by using six different machine learning algorithms. Performance and accuracy of the applied algorithms is discussed and compared. Comparison of the different machine learning techniques used reveals which algorithm is best suited for prediction of diabetes. This aims to help doctors and practitioners in early prediction of diabetes using machine learning techniques.

Ivan William, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawanto, Heru Agus Santoso and Christy Atika Sari [4] aimed to conduct surveys, test performance, and compare the accuracy of the FaceNet method with various other methods that have been developed previously. They integrated the Tensorflow learning machine and pre-trained model, and found the training time to be shorter. Implementation of the FaceNet method used two types of pre-trained models, namely CASIA-Web Face and VGGFace2, and tested on various data sets of standard face images that have been widely used before. From the results, FaceNet showed excellent results and was superior to other methods. By using VGGFace2 pre-trained models, FaceNet was able to touch 100% accuracy.

Madhura Patil, Rima Jadhav, Vishakha Patil, Aditi Bhawar and Mrs.Geeta Chillarge [5] provide a system which can help for prediction of heart disease by considering risk factors associated with heart disease. The Support Vector Machine algorithm is applied on historical information/data of the patient and it provides features like Age, Sex, Smoking, Overweight, Alcohol Intake, Bad Cholesterol, Blood Pressure and Heart Rate to make the prediction.

Ning Zhang, Wuqi Gao and Junmin Luo [6] deeply analyzed MTCNN and its implementation principle was introduced in detail. The real effect of MTCNN in the task of face detection is verified by experiments. The results of the model are compared with those of the Yolov3 model in the wider face dataset.

2.2 Outcome of the Literature Survey

The entire review can be summarized in the following few points:

- Based on comparison with other methods, the FaceNet method is proven to have the best performance. The accuracy of the FaceNet method is also strongly influenced by the pre-trained data model where VGGFace2 produces better average recognition accuracy. But in one data set only found an accuracy of about 77%, this is possible because each face label has many differences, while the training method on FaceNet uses triplet loss that will minimize the gap of anchor and positive, also maximize the gap of anchor and negative image, where is the difference a very significant face can be considered a negative image.

- The accuracy level of Random Forest is greater when compared to other algorithms in predicting diabetes. Random Forest uses the foundations of each indiscriminately created decision tree to predict the result and stores the anticipated outcome at intervals the target place. It also calculates the votes for each predicted target and ultimately, admits the high voted predicted target as a result of the ultimate prediction from the random forest formula.

- By applying the Support Vector Machine classifier, we can predict if the patient will get heart disease or not. SVM is a strong classifier which can identify two classes.SVM classifies the test image to the class which has the maximum distance to the closest point in the training.

- Classification Accuracy, sensitivity, and specificity of the SVM has been found to be high. The SVM is a learning algorithm for classification. It tries to find the optimal separating hyperplane such that the expected classification error for unseen patterns is minimized.

- MTCNN can output the position of faces in the image accurately when the number of faces in the image is small. When MTCNN processes an image, it first performs the image resizing operation to scale the original image to different scales to generate an image pyramid. Then the images of different scales are sent to the three sub networks for training in order to detect different sizes of human faces and realize multi-scale target detection.

- When there are many faces in a picture, MTCNN can get the position of the face more accurately. Yolov3 has a high rate of missing detection. The R-Net or refined network reduces the number of candidates by performing calibration with bounding box regression and employs non-maximum suppression to merge overlapping candidates. Due to this, MTCNN can get the position of faces more accurately.

- In extreme cases, MTCNN can still recognize most of the faces. Yolov3 algorithm can hardly recognize the faces in the image when there are many faces with small targets in a picture. Since MTCNN uses five-point facial landmarks.

Chapter 3

System Analysis and Design

The chapter aims to introduce the requirements of the project. To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as guidelines opposed to an absolute rule. Most software defines two sets of software requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirements tend to increase over time. Requirements are typically classified into types produced at different stages in a development progression, with the taxonomy depending on the overall model being used. The proposed system requirements are classified into four categories that are as follows: Functional requirements, Non-functional requirements, Hardware requirements and Software requirements. The functional requirements specify what the product must do. They relate to the actions that the product must carry out in order to satisfy the fundamental reasons for its existence. Non-functional requirements are the properties that your product must have. Think of these properties as the characteristics or qualities that make the product most attractive, usable, fast, reliable. These properties are not required because they are fundamental properties of the product such as computations, manipulating data and so on, but because the client wants these fundamental activities to be performed in a certain number, these requirements also play an important role in analysing the system.

3.1 Hardware Requirements

The most common set of requirements defined for a software application is the physical computer resources, also known as Hardware Requirements. The proposed system has the following Hardware Requirements:

1. Laptop / Desktop - To run the project
2. High Quality Camera/Webcam - To perform facial recognition efficiently

3. Memory: 2-3 GB - To store the project and datasets
4. High Performing CPU and GPU - To increase the efficiency of facial recognition

3.2 Software Requirements

A computing platform describes some sort of framework, either in hardware or software, which allows software to run. Typical platforms include a computer's architecture, operating systems, or programming languages and their run time libraries. Software deals with defining software requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Operation environment:

1. Platforms: Django 3.2.3, Colaboratory, SQLite
2. Languages Used: Python, HTML
3. Operating system: Linux Ubuntu 20.04.2 LTS

3.2.1 Django Framework

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source. Django was designed to help developers take applications from concept to completion as quickly as possible. Django takes security seriously and helps developers avoid many common security mistakes. Some of the busiest sites on the Web leverage Django’s ability to quickly and flexibly scale. This framework was used to create the application of this project.

3.2.2 Colaboratory

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited

to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

3.2.3 SQLite

SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program. SQLite generally follows PostgreSQL syntax. SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity. This means that one can, for example, insert a string into a column defined as an integer. SQLite will attempt to convert data between formats where appropriate, the string "123" into an integer in this case, but does not guarantee such conversions and will store the data as-is if such a conversion is not possible. SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

3.3 Functional Requirements

Functional requirements define a function of a system or its components. A function is described as a set of inputs, the behavior and the outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Following are the functional requirements for the developed system:

- The system shall be able to add the patient and his/her medical details.
- The system shall be able to create a dataset of images of the patient for the identification process.
- The system shall be able to use facial recognition to recognize the patients based on comparison with the database.
- The system shall be able to show the history of the patient.

- The system shall be able to let the doctors and patients check for the possibility of being a Diabetic patient based on input parameters.
- The system shall be able to let the doctors and patients check for the possibility of having a heart disease based on input parameters.

3.4 Non-functional Requirements

A Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements. The developed system deals with satisfying following non-functional requirements

- **Confidence**: Confidence scores are a critical component of face detection and comparison systems. These systems make predictions of whether a face exists in an image or matches a face in another image, with a corresponding level of confidence in the prediction.

- **Precision**: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate.

  Precision = TP/TP+FP

  TP=True Positive, FP = False Positive

- **Accuracy**: Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observations to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positives and false negatives are almost the same. Therefore, you have to look at other parameters to evaluate the performance of your model.

  Accuracy = TP+TN/TP+FP+FN+TN

- **F1 score**: F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven

class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

- **Recall**: Recall is the ratio of correctly predicted positive observations to the all observations in actual class yes.

Recall = TP/TP+FN

- **Support**: Support may be defined as the number of samples of the true response that lies in each class of target values.

3.5 System Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. Systems design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created? From the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system. Systems design then overlaps with systems analysis, systems engineering and systems architecture System designing in terms of software engineering has its own value and importance in the system development process as a whole. To mention it may though seem as simple as anything or simply the design of systems, but in a broader sense it implies a systematic and rigorous approach to design such a system which fulfills all the practical aspects including flexibility, efficiency and security. System designing leads to ensure that the system is created in such a way that it fulfills the need of the users and keep them at ease being user-oriented. During the system design the overall structure and style is decided. It decides the high-level design of the system.

Here, the proposed system provides a smooth flow right from taking the input details of the Patient and storing it in the database to displaying them upon facial recognition. The system also provides

an interface for the Patient and the Doctors to predict heart disease and Diabetes based on the parameters.

3.5.1 Proposed System

The proposed system aims to make an application using the Django Framework through which:

- **Patient details can be added:**

  The details consist of the personal details and medical history of the patient. These details along with images of the Patient's face are stored in the database.

- **The Model can be trained:**

  The model is trained using the images of the Patients that are stored in the database.

- **The Patient's details can be retrieved:**

  Upon successful facial recognition, the details of the Patient are displayed.

- **Diabetes Prediction can be done:**

  Based on the input parameters, the system can be used for predicting Diabetes.

- **Heart Disease Prediction can be done:**

  Based on the input parameters, the system can be used for Heart Disease Prediction.

  The below diagram is a flowchart of the proposed system. It shows the proposed system in an easy and understanding manner.

3.5.2 Use Case Diagram of the Proposed System

In software and systems engineering, a use case is a list of action or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system, to achieve a goal. The actor can be a human, an external system, or time. The actors that have played a major role in the proposed system are the Patient, the Admin and the Doctor.

Fig 3.1 Use Case Diagram

Figure 3.5.2 indicates the interactions between the system and the actors. As shown, the Patient will be involved in actions such as adding images, patient details, medical history, details for the Diabetes and Heart Disease Prediction.

The admin is involved in actions such as the Identification process of the patient, Training the model with image datasets and training the models for the Diabetes and Heart Disease prediction.

The Doctor is involved in actions such as Identifying the patient, entering the details for Disease Prediction, Reviewing the Disease Prediction and Reviewing the Medical records of the Patients.

3.5.3 Sequence Diagram of the Proposed System

Sequence diagrams are simple subsets of interaction diagrams. They map out sequential events in an engineering or business process to streamline activities. The sequence diagram is used to represent the interactions and data flow between the system and the various users in our proposed system.

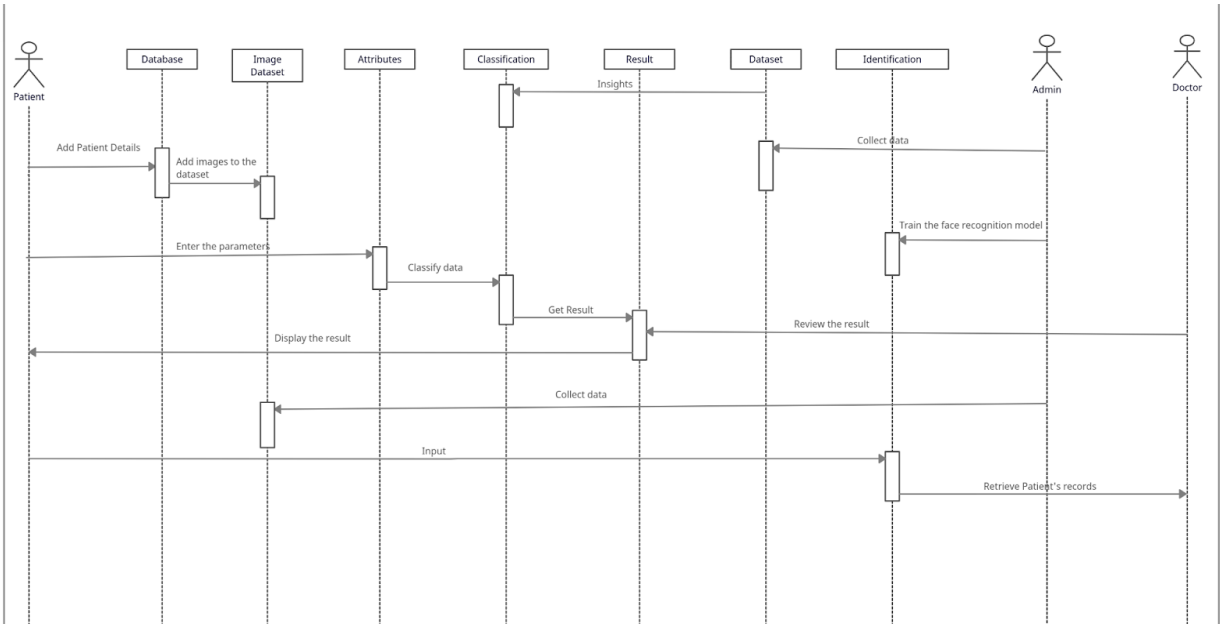

Fig 3.2 Sequence Diagram

As per the above sequence diagram, we can see that the actors are the Patient, the Admin and the Doctor. The Patient adds his/her details in the database and those images are used and added to the image dataset. The patient also provides the parameters which would serve as the attributes in the case of heart disease and Diabetes prediction. This data is then classified and the result is obtained.

The result is then shown to the Patient. Lastly, the patient's face is used for the process of identification. The admin collects the data in the form of images from the Image Dataset for the purpose of the facial recognition process. The admin also collects the data from the Heart Disease Prediction and Diabetes Prediction datasets respectively which in turn take insights from the data which was classified. The admin also trains the Face Recognition Model.

The Doctor reviews the results of the Heart Disease Prediction and the Diabetes Prediction models. The Doctor also retrieves the Patient's records and medical history.

3.6 Datasets

A dataset, or data set, is simply a collection of data. The simplest and most common format for datasets you'll find online is a spreadsheet or CSV format — a single file organized as a table of rows and columns. But some datasets will be stored in other formats, and they don't have to be just one file. Sometimes a dataset may be a zip file or folder containing multiple data tables with related data.

3.6.1 Pima Indians Diabetes Database

Link: https://www.kaggle.com/uciml/pima-indians-diabetes-database

License: CC0: Public Domain

Context: This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database.

Attributes:

1. Glucose - Plasma glucose concentration 2 hours in an oral glucose tolerance test
2. Blood Pressure - Diastolic blood pressure (mm Hg)
3. Skin Thickness - Triceps skinfold thickness (mm)

4. Insulin - 2-Hour serum insulin (mu U/ml)
5. BMI - Body mass index (weight in kg/ (height in m) ^2)
6. Diabetes PedigreeFunction - Diabetes pedigree function
7. Age - Age (years)
8. Outcome - Class variable (0 or 1)

3.6.2 Heart Disease UCI

Link: https://www.kaggle.com/ronitf/heart-disease-uci

License: Reddit API Terms

Context: The dataset used is the Cleveland Heart Disease dataset taken from the UCI repository.

Attributes:

1. Age - age in years
2. Sex - (1 = male; 0 = female)
3. CP - chest pain type
4. Trestbps - resting blood pressure (in mm Hg on admission to the hospital)
5. Chol - serum cholesterol in mg/dl
6. FBS - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. Restecg - resting electrocardiographic results
8. Thalach - maximum heart rate achieved
9. Exang - exercise induced angina (1 = yes; 0 = no)
10. Oldpeak - ST depression induced by exercise relative to rest
11. Slope - the slope of the peak exercise ST segment
12. CA - number of major vessels (0-3) colored by fluoroscopy
13. Thal - 3 = normal; 6 = fixed defect; 7 = reversible defect
14. Target - 1 or 0

Chapter 4
Modeling and Implementation

4.1 Structure of the System:

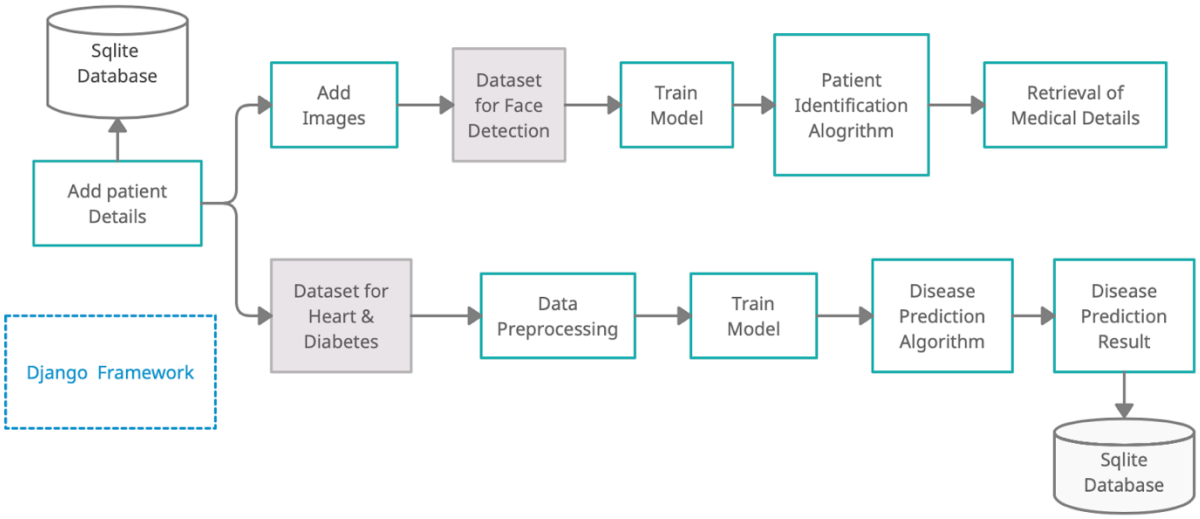

Fig 4.1 Structure of the System

The structure of the system, as shown in the figure, has a first step where patients enter the personal and medical details. These details are then stored in the sql database. Then the patient adds his images. These faces are the input of the next step, that performs the feature extraction and trains the recognition model. This feature extraction consists of obtaining the information that can be useful for identification and recognition, from each of the faces, such as the aperture of the eyes or the shape of the nose. This information or features are then fed to the recognition model, which transforms the feature information into digital information.

In the Second step, the datasets on heart disease and diabetes are used as input for the model. The preprocessing of data is performed to clean the data. Then the model for prediction is trained. Then the entered details are classified by the model on the basis of information learned from the training of the dataset. Then the results are stored in sql database and displayed to the user.

4.2 Modules and Algorithms:

4.2.1 MTCNN:

MTCNN or Multi-Task Cascaded Convolutional Neural Networks is a neural network which detects faces and facial landmarks on images. MTCNN is used to detect faces for training the model and it is also used in recognition. It consists of 3 neural networks connected in a cascade. This model has three convolutional networks (P-Net, R-Net, and O-Net) and is able to outperform many face-detection benchmarks while retaining real-time performance.

Stage 1: The Proposal Network (P-Net)

This first stage is a fully convolutional network (FCN). The difference between a CNN and a FCN is that a fully convolutional network does not use a dense layer as part of the architecture. This Proposal Network is used to obtain candidate windows and their bounding box regression vectors.

Stage 2: The Refine Network (R-Net)

The R-Net further reduces the number of candidates, performs calibration with bounding box regression and employs non-maximum suppression (NMS) to merge overlapping candidates. The R-Net outputs whether the input is a face or not, a 4-element vector which is the bounding box for the face, and a 10-element vector for facial landmark localization.

Stage 3: The Output Network (O-Net)

This stage is similar to the R-Net, but this Output Network aims to describe the face in more detail and output the five facial landmarks' positions for eyes, nose and mouth.

Face classification is a binary classification problem that uses cross-entropy loss:

$$L_i^{det} = -(y_i^{det} log(P_i) + (1 - y_i^{det})(1 - log(P_i)))$$

Bounding box regression is the learning objective of the regression problem. For each candidate window, the offset between the candidate and the nearest ground truth is calculated. Euclidean loss is employed for this task.

$$L_i^{box} = \left|\left|\hat{y}_i^{box} - y_i^{box}\right|\right|_2^2$$

Facial Landmark localization of facial landmarks is formulated as a regression problem, in which the loss function is Euclidean distance:

$$L_i^{landmark} = \left|\left|\hat{y}_i^{landmark} - \hat{y}_i^{landmark}\right|\right|_2^2$$

4.2.2 FaceNet:

FaceNet is a deep neural network used for extracting features from an image of a person's face. FaceNet takes an image of the person's face as input and outputs a vector of 128 numbers which represent the most important features of a face. In machine learning, this vector is called embedding. FaceNet takes a person's face and compresses it into a vector of 128 numbers. Ideally, embeddings of similar faces are also similar. Embeddings are vectors and we can interpret vectors as points in the Cartesian coordinate system. One possible way of recognizing a person on an unseen image would be to calculate its embedding, calculate distances to images of known people and if the face embedding is close enough to embeddings of person A, we say that this image contains the face of person A.

FaceNet learns in the following way:

- Randomly selects an anchor image.
- Randomly selects an image of the same person as the anchor image (positive example).
- Randomly selects an image of a person different from the anchor image (negative example).
- Adjusts the FaceNet network parameters so that the positive example is closer to the anchor than the negative example.

The intuition behind the triplet loss function is that we want our anchor image (image of a specific person A) to be closer to positive images (all the images of person A) as compared to negative images (all the other images). Triplet loss function can be formally defined as –

$$\sum_{i}^{N} \left[\left|\left| f(x_i^{\alpha}) - f(x_i^{p}) \right|\right|_2^2 - \left|\left| f(x_i^{\alpha} - f(x_i^{n})) \right|\right|_2^2 + \alpha \right]$$

4.2.3 Random Forest Algorithm: -

- It is an ensemble classifier using many decision trees models; it can be used for regression as well as classification.
- Accuracy and variable importance information can be provided with the results.
- A random forest is the classifier consisting of a collection of tree structured classifiers k, where the k is independently, identically distributed random trees and each random tree consists of the unit of vote for classification of input.
- Random forest uses the Gini index for the classification and determining the final class in each tree.
- The final class of each tree is aggregated and voted by the weighted values to construct the final classifier.
- The working of random forest is, A random seed is chosen which pulls out a random, a collection of samples from the training datasets while maintaining the class distribution.

Implementation in Scikit-learn

For each decision tree, Scikit-learn calculates a nodes importance using Gini Importance, assuming only two child nodes (binary tree):

$$ni_j = w_j c_j - w_{left(j)} c_{left(j)} - w_{right(j)} c_{right(j)}$$

- ni_j= the importance of node j
- w_j = weighted number of samples reaching node j
- c_j= the impurity value of node j
- left(j) = child node from left split on node j
- right(j) = child node from right split on node j

The importance for each feature on a decision tree is then calculated as:

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

- fi_j= the importance of feature i
- ni_j= the importance of node j

These can then be normalized to a value between 0 and 1 by dividing by the sum of all feature importance values:

$$normfi_i = \frac{fi_i}{\sum_{j \in all\ features} fi_j}$$

The final feature importance, at the Random Forest level, is its average over all the trees. The sum of the feature's importance value on each tree is calculated and divided by the total number of trees:

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T}$$

- $RFfi_i$= the importance of feature i calculated from all trees in the Random Forest model
- $normfi_{ij}$= the normalized feature importance for i in tree j
- T = total number of trees

4.2.4 Support Vector Machine Algorithm: -

Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the kernel trick. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e., it converts non separable problem to separable problem. It is mostly useful in non-linear separation problems. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

SVM can be of two types.

- **Linear SVM**: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM**: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

4.3 Outlier Detection:

An outlier is an observation that lies in an abnormal distance from other values in a random sample from a population. It is important to remove outliers to build a prediction model with a good accuracy score. Some Outliers were present in the Diabetes dataset.

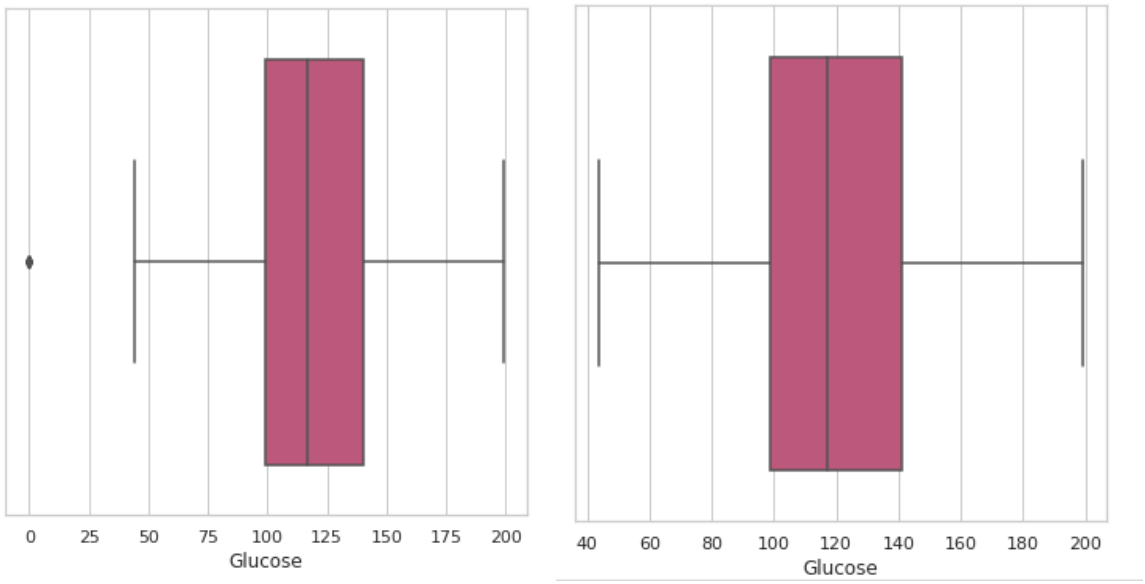

Fig 4.2 Outliers Detection and Removal for Glucose Attribute

For the glucose attributes, there is an outlier present having the value 0, furthermore the glucose concentration of a person cannot be zero, hence we remove it.

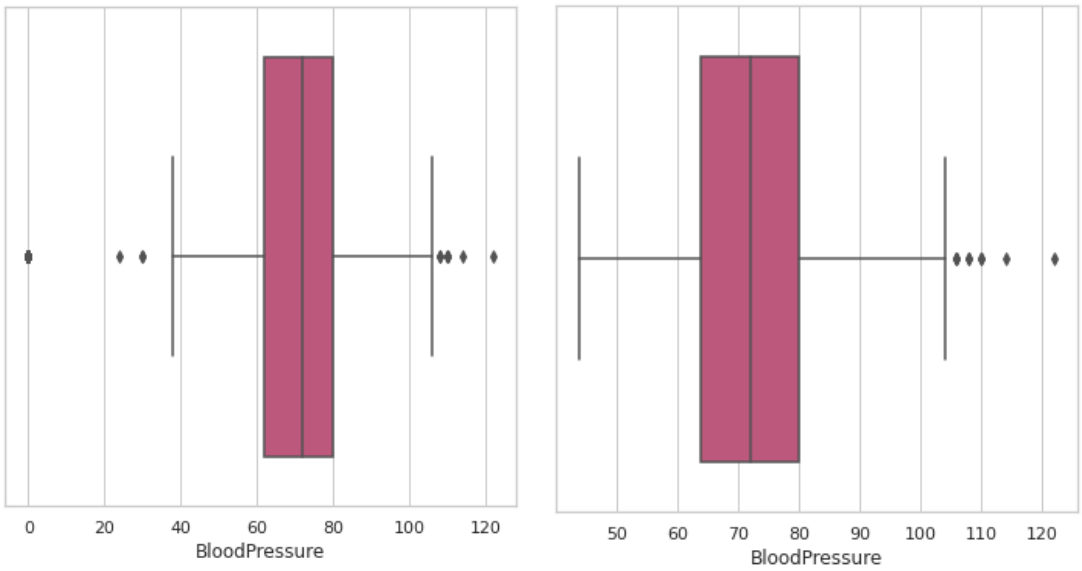

Fig 4.3 Outliers Detection and Removal for Blood Pressure Attribute

For the Blood Pressure attributes, the boxplot shows few outliers present in the range from 0-40 mm Hg, hence they are removed.

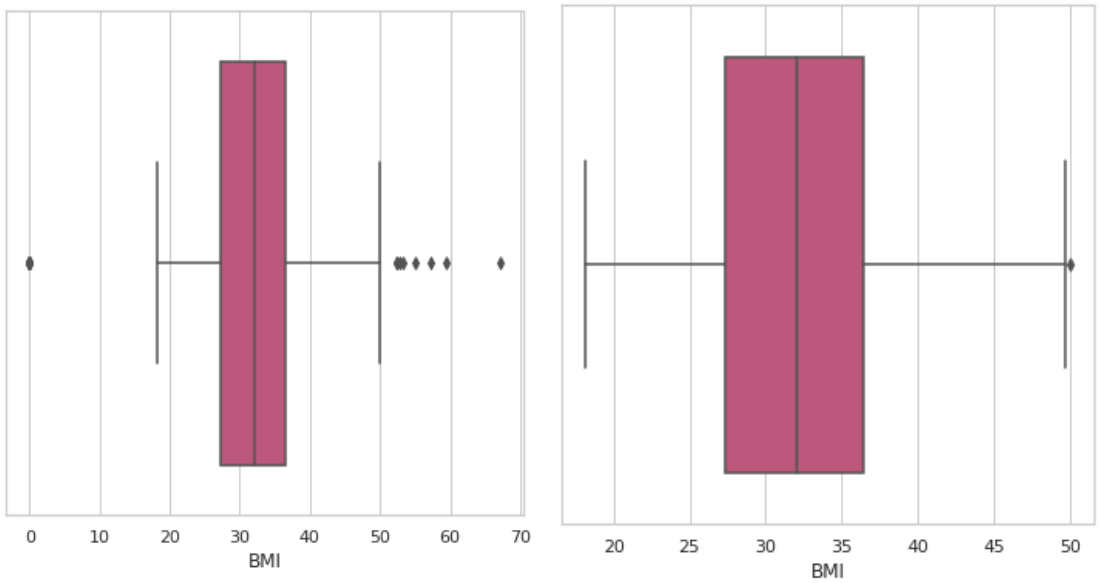

Fig 4.4 Outliers Detection and Removal for BMI Attribute

For the BMI attributes, the boxplot shows few outliers present such as 0, hence they are removed.

4.4 Oversampling:

There are a number of methods available to oversample a dataset used in a typical classification problem. In the diabetes dataset, the number of diabetic patients is lower than the number of non-diabetic people. To oversample, the team used SMOTE oversampling technique.

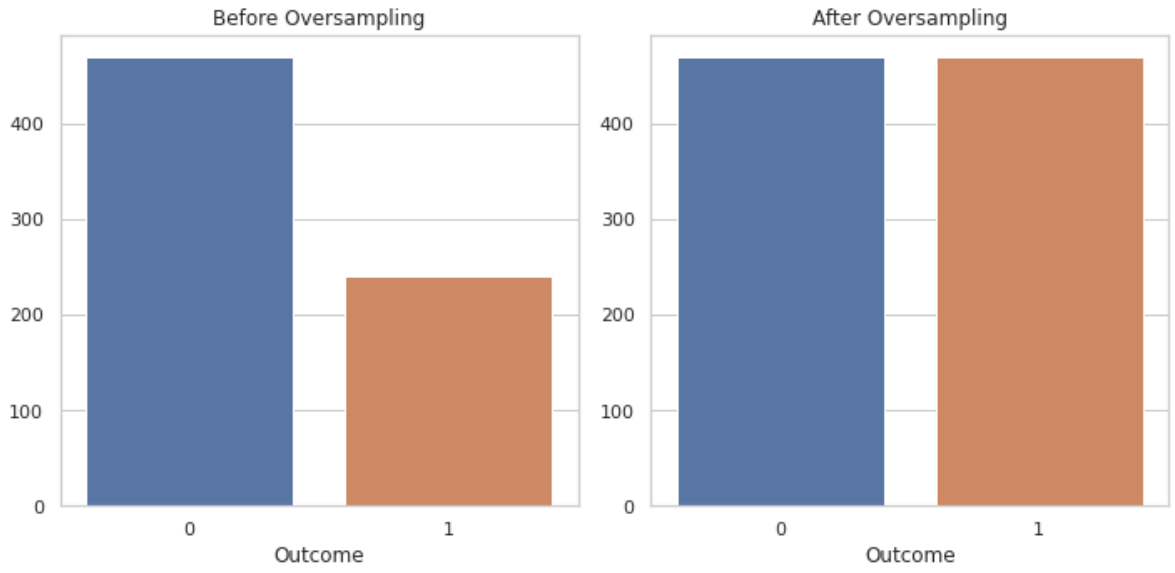

Fig 4.5 Bar plots of Oversampling

The figure above shows the distribution which was changed before training and comparing the prediction model.

4.5 Detailed Workflow:

The diagram shown below is a flowchart which shows the step-by-step detailed working of the patient identification and disease prediction application.

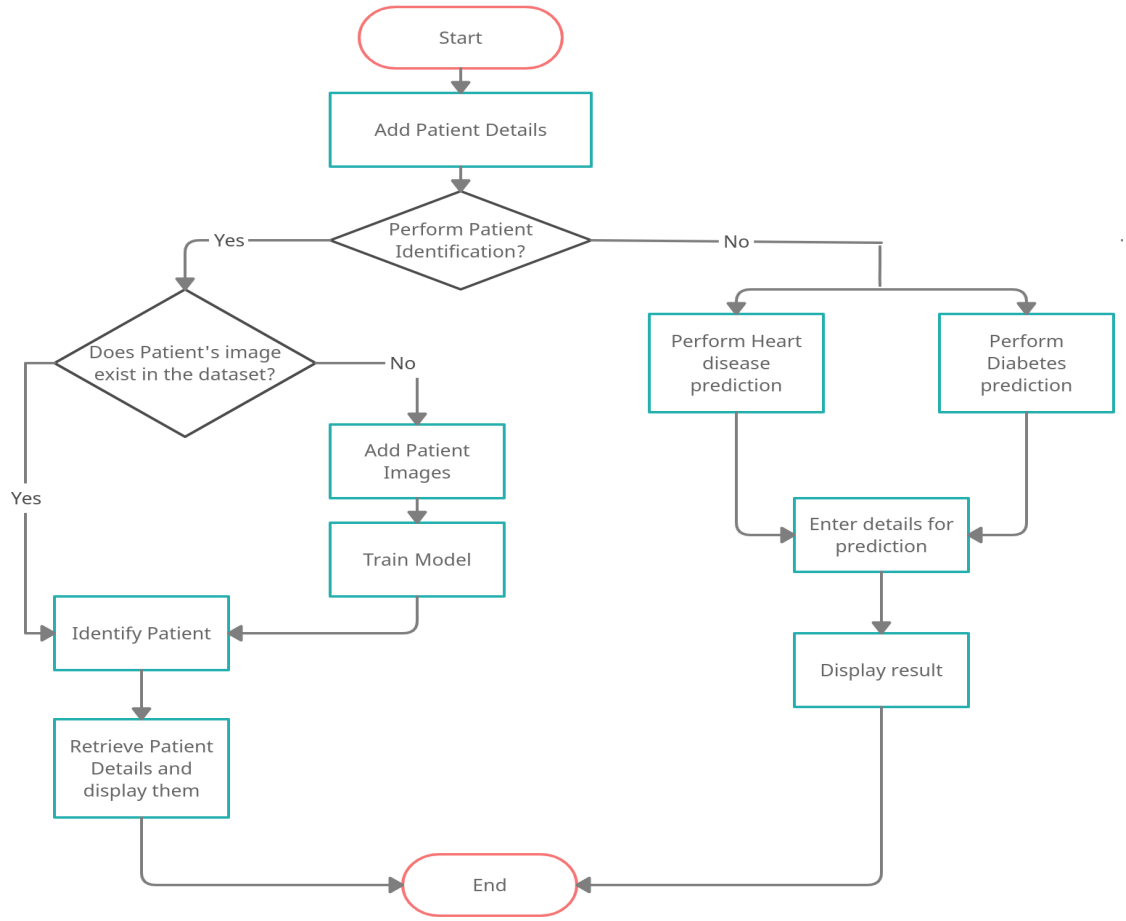

Fig 4.6 Detailed WorkFlow

4.5.1 Algorithm:-

Step 1: In the first step, the Django application is launched and the homepage of the web application is requested. In this home page there is a navigation bar which helps navigate through different services in the application.

Step 2: Adding a Patient is the next step in this application. Where the patient can enter his/her personal details and medical details.

Step 3: After a Patient is added his facial images are added to the image dataset.

Step 4: Then the models are trained for face recognition, heart disease prediction and diabetes prediction. These models should be continuously updated throughout the lifetime of the application.

Step 5: On the homepage of the application, there are multiple services from which a user can choose. These services are face recognition, heart disease prediction and diabetes prediction. If the user selects face recognition go to step 6. If the user selects heart disease prediction go to step 7. If the user selects diabetes prediction go to step 8.

Step 6: In the face recognition model a video frame is opened (which requires a webcam/camera). In this video frame the patient is identified and his name and confidence level are displayed on the screen. Once the Admin confirms the identity of the person, he can retrieve the patient's details and medical records. After the service is completed, the user can either terminate the application in step 9 or the user can go to the selecting menu in step 5.

Step 7: In the heart disease prediction model, a form is used to take the inputs from the user. This input is used to predict if a patient has more chances of getting the heart disease or not. After the service is completed, the user can either terminate the application to step 9 or the user can go to the selecting menu in step 5.

Step 8: In the diabetes prediction model, a form is used to take the inputs from the user. This input is used to predict if a patient has more chances of getting diabetes or not. After the service is completed, the user can either terminate the application to step 9 or the user can go to the selecting menu in step 5.

Step 9: The application is terminated.

Chapter 5

Testing, Results and Discussion

5.1 Testing :-

The first step for testing is to run the code written in Python to generate a Django web application Figure 5.1 shows the web page created by our code.

After the code is successfully compiled, the html file generates a front-end web page which is the home page of the web application, that comprises of the services that the application provides. It also displays all the options that allows the users to navigate through all the features offered by the application.

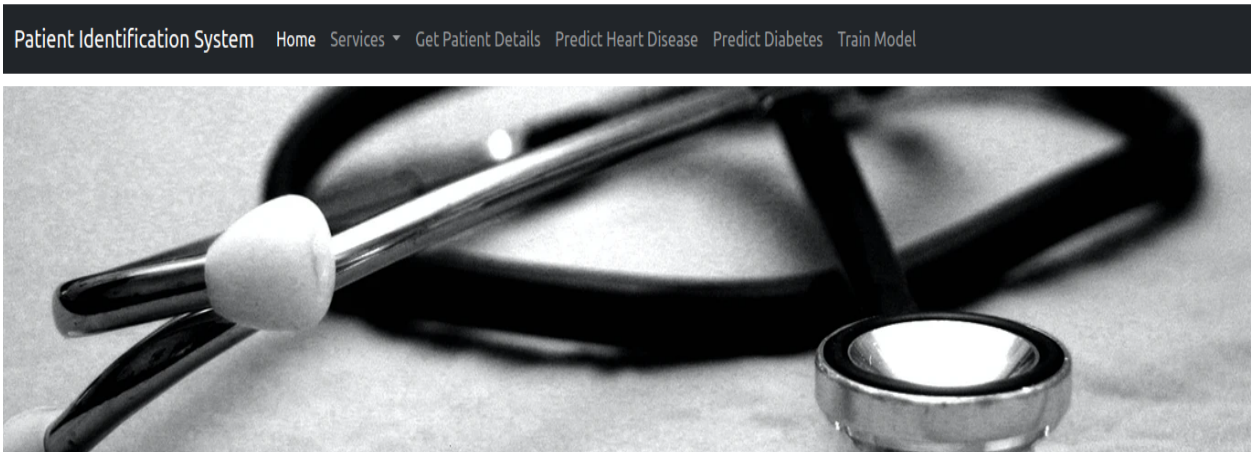

Fig 5.1 Home Page

As the home page is created, testing can be started for the services offered by the web application. There are three types of services offered by the web application :-

- Patient Identification System
- Diabetes Prediction
- Heart Disease Prediction

First, testing is done on the patient identification system by entering the personal details and medical details of the patient and submitting it. On clicking the submit button, the details get stored or updated in the SQLite database, which is then used for the patient identification process.

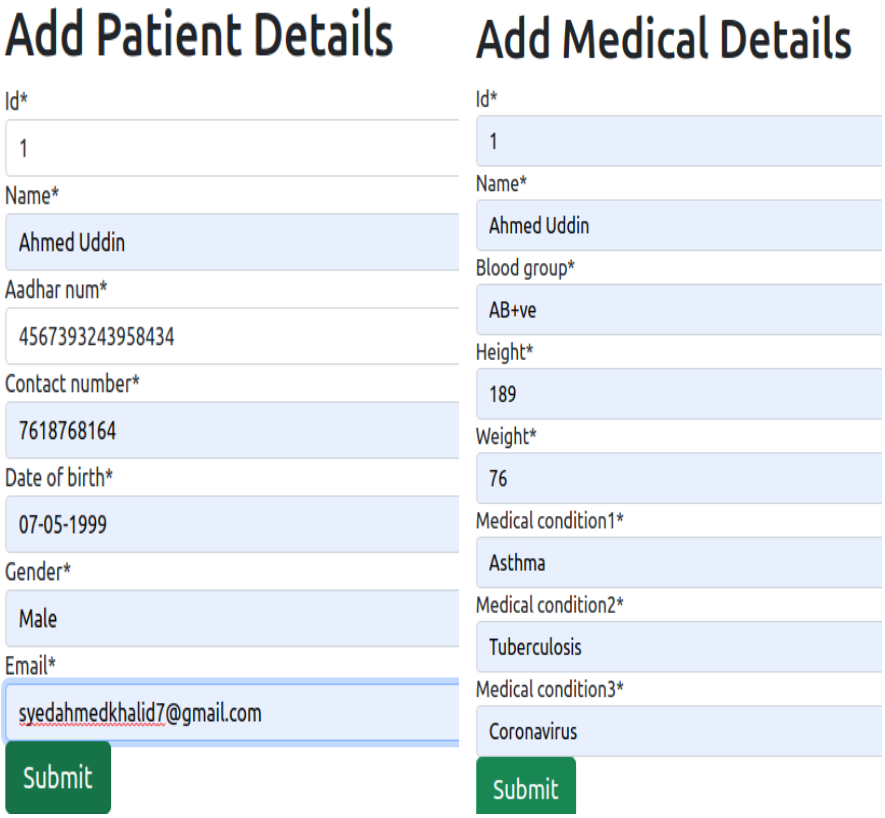

Fig 5.2 Information Form

As the required details are uploaded in the database, a minimum of 15 to 20 photos are uploaded for each person which will be added in the dataset. Then the model is trained for facial recognition. MTCNN is applied for face detection and Face Net is applied for facial recognition.

When the identification is run, it will detect and capture the face image and then find the match in the dataset and if the match is found, the linked data record from the database is retrieved.

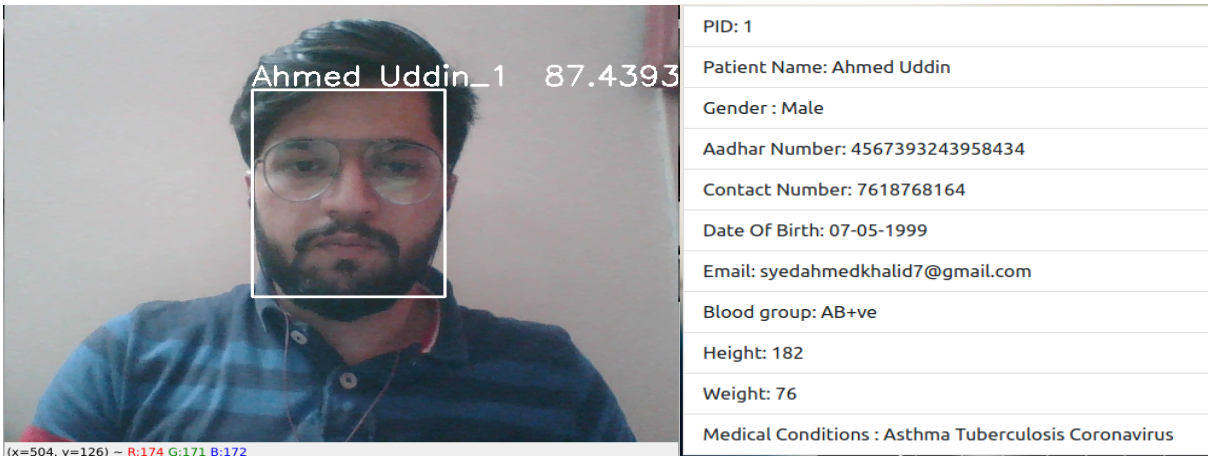

Fig 5.3 Face Recognition Results

As for the Diabetic prediction system, the model is built using the Random Forest Algorithm and then the model is trained for the Pima Indians Diabetes Database from UCI Machine Learning. The required details should be entered by the Patient or the Doctor and then the model will process the data and give the result.

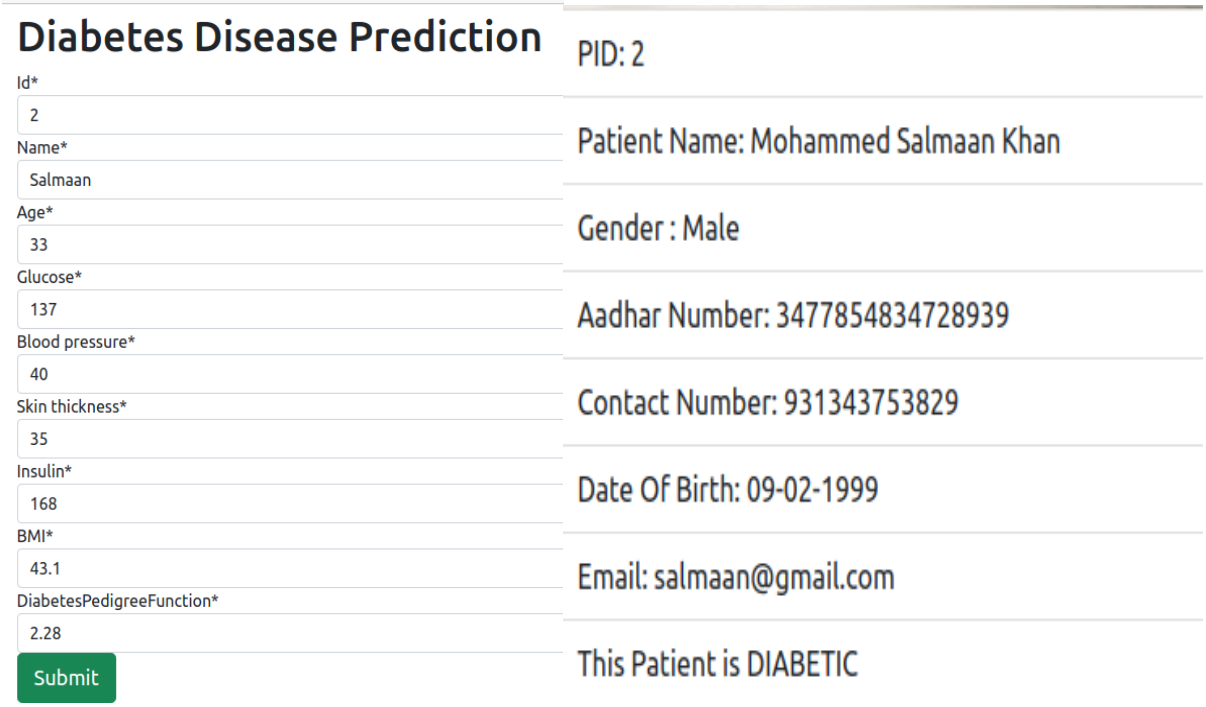

Fig 5.4 Diabetes Prediction Form and Result

In the case of the Heart Disease Prediction System, the model is built using the Support Vector Machine Algorithm and then the model is trained for the Heart Disease UCI from UCI Machine Learning. The required details should be entered by the Patient or the Doctor and then the model will process the data and give the result.

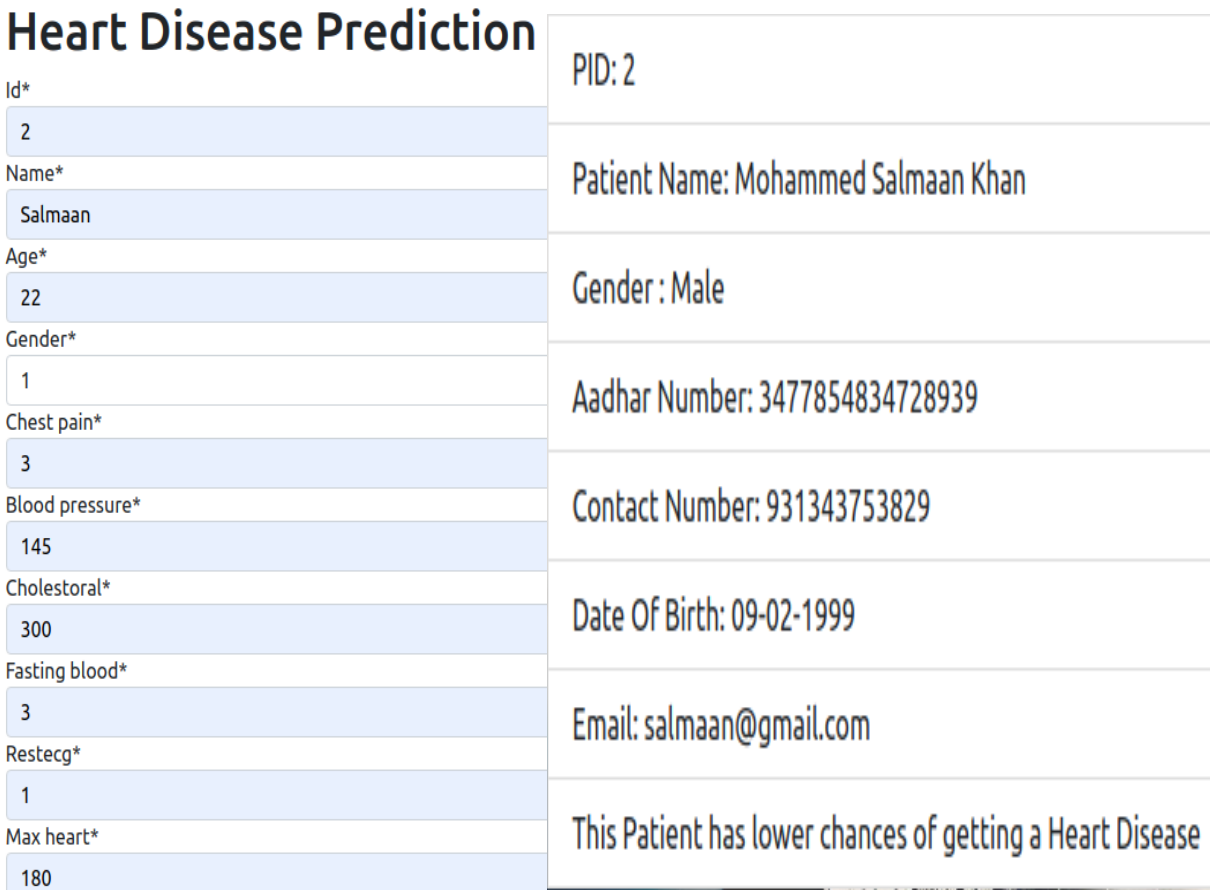

Fig 5.5 Heart Disease Prediction Form and Result

5.2 Results and Discussion :

5.2.1 Face Identification and Recognition :-

One of the most important characteristics is the speed of the algorithm. Hence when the team wanted to choose between the face detection algorithms, depending on their application, execution time is crucial . In the following graph, the team compared the total time that the algorithms needed to process the video. From the graph one can see that the Dlib algorithm needed the shortest time to process. On the other hand, MTCNN took the longest.

We have seen how many detections each algorithm made in addition to their execution times. To conclude, if a fast face detection algorithm is needed, one should use Dlib. On the other hand, if the goal is to select an algorithm to detect a large number of faces, one's choice can be Face net or Mtcnn.

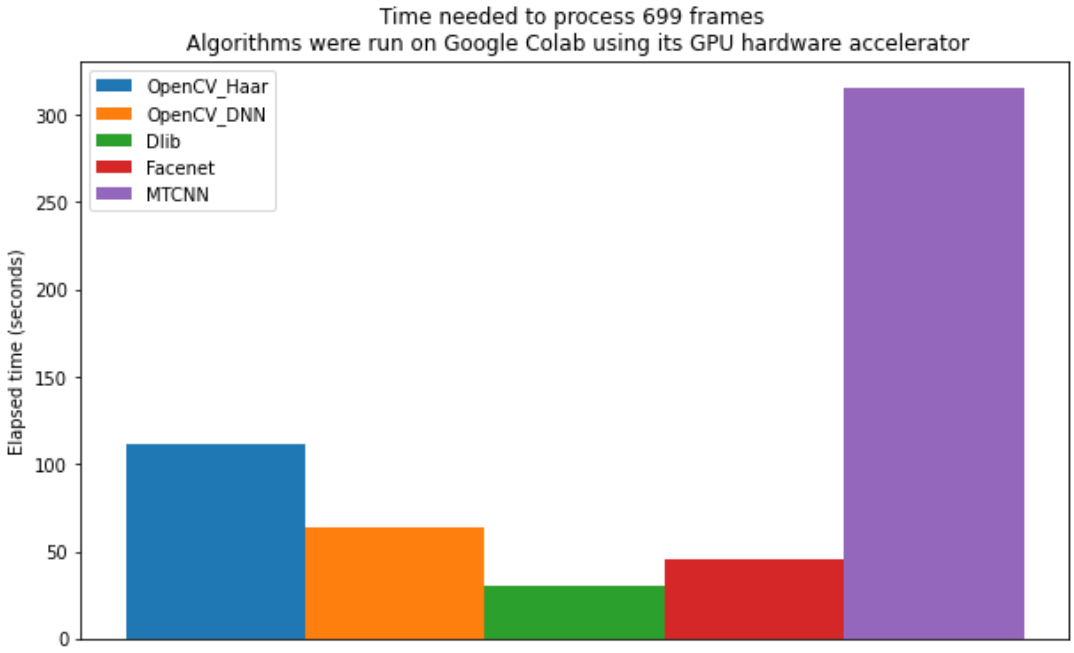

Fig 5.6 Comparison between face detection algorithm to process 699 frames

But The Face Net algorithm performed better among the others as it had a maximum accuracy of 99.63%. So, the team chose to implement the Face Net algorithm designed for low-power devices. The Face Net model offered better speed with high accuracy over others as is evident from the graph shown below.

The reason behind the fast performance is that the global average pooling layer has been replaced by a depth-wise convolutional layer which improves performance on face detection and recognition.

That's why the team used the combination of Mtcnn and Face Net. Mtcnn for face detection and Face Net for face recognition. Which gives them better accuracy with good overall execution time.

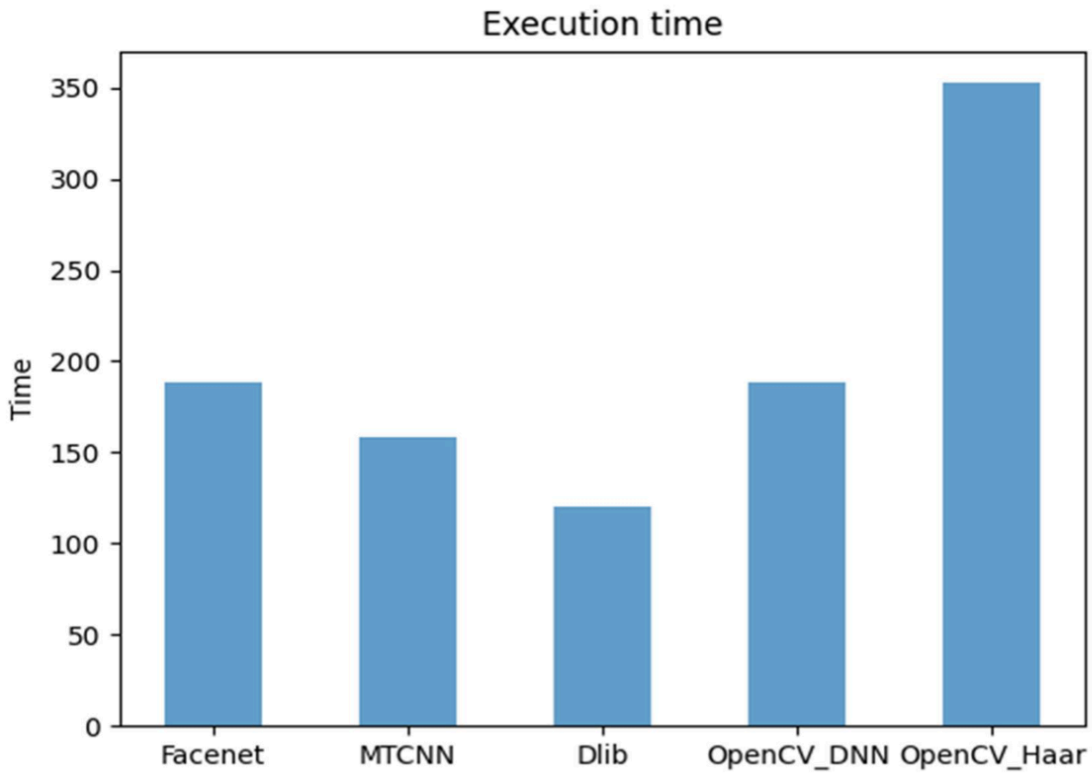

Fig 5.7 Comparison between face detection algorithm w.r.t time

5.2.2 Diabetes Prediction :-

Here one can see in the table, that the Random Forest Classifier gives a better Accuracy score of 83%. Whereas the rest of the algorithms give below 75%. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Name	Precision_0	Precision_1	Recall_0	Recall_1	F1-score_0	F1-score_1	Support_0	Support_1	Accuracy
LinearDiscriminantAnalysis	0.71	0.77	0.79	0.69	0.75	0.73	139	143	0.737588652482269
RandomForestClassifier	0.87	0.81	0.79	0.88	0.83	0.85	139	143	0.836879432624114
KneighborsClassifier	0.76	0.71	0.67	0.8	0.71	0.75	139	143	0.734042553191489
DecisionTreeClassifier	0.72	0.74	0.74	0.71	0.73	0.73	139	143	0.726950354609929
GaussianNB	0.69	0.75	0.77	0.66	0.73	0.7	139	143	0.712765957446808
SVC	0.69	0.74	0.76	0.66	0.72	0.7	139	143	0.712765957446808

Table 1: Comparison between Diabetes Prediction Models

Visualisation of attributes from above table of diabetes prediction:-

Precision and recall are two extremely important model evaluation metrics. While precision refers to the percentage of your results which are relevant, recall refers to the percentage of total relevant results correctly classified by your algorithm. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. Accuracy refers to how close measurements are to the "true" value.

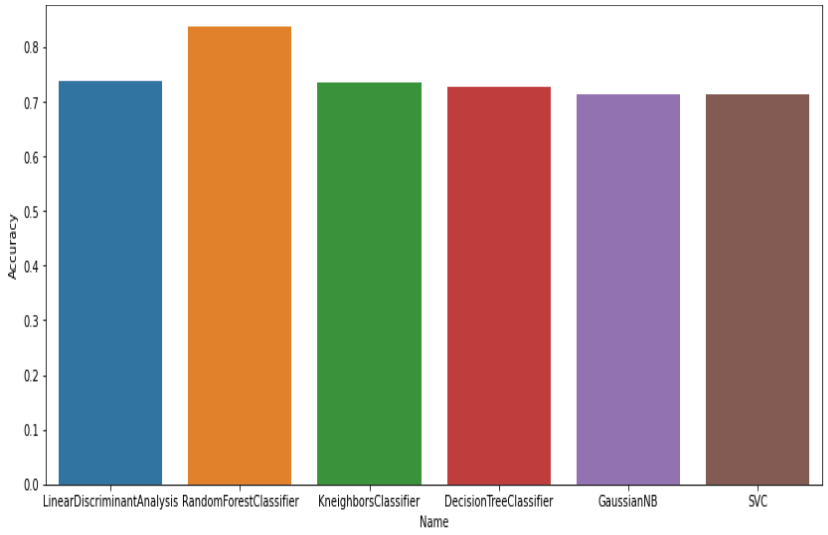

Fig 5.8 Accuracy of Diabetes Model

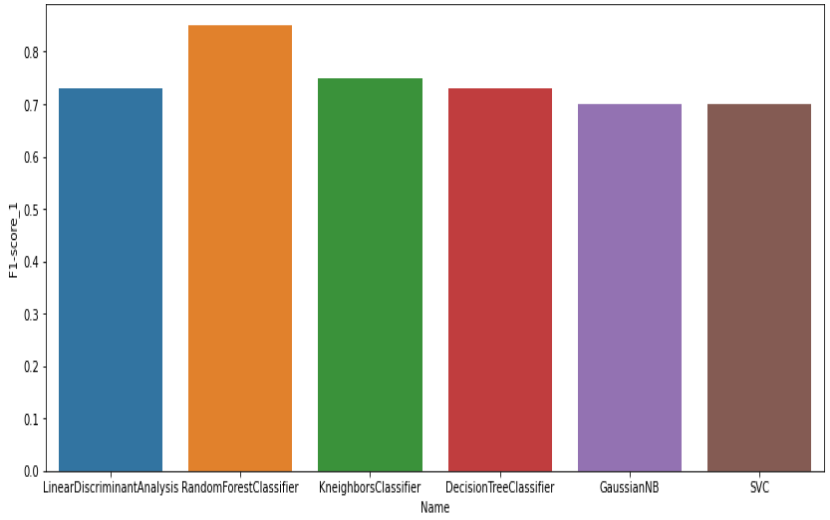

Fig 5.9 F1-Score of Diabetes Model

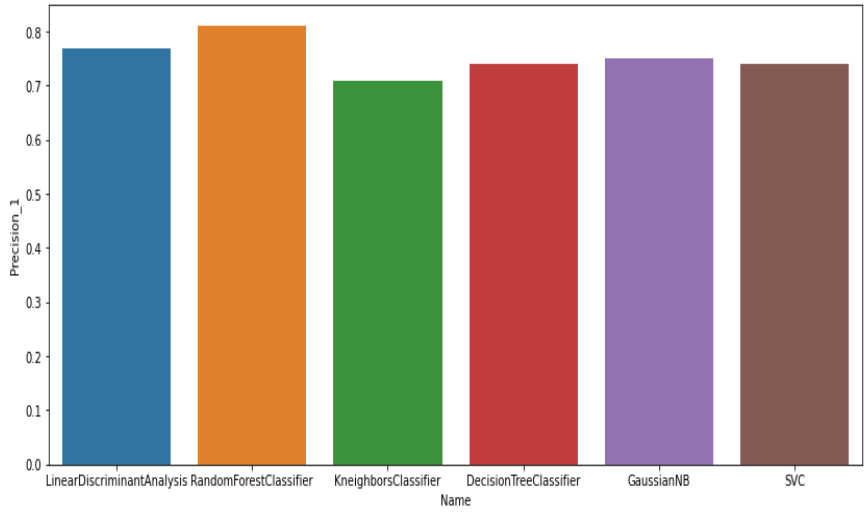

Fig 5.10 Precision of Diabetes Model

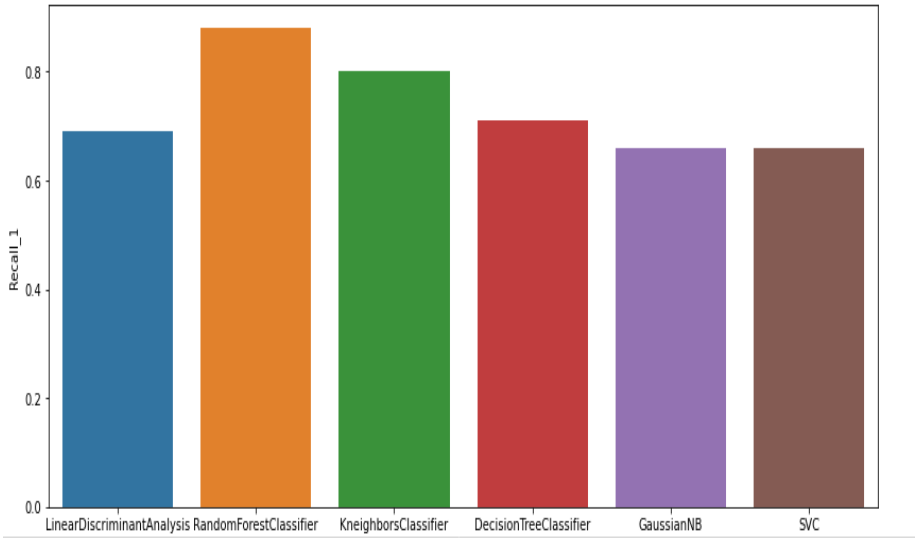

Fig 5.11 Recall of Diabetes Model

The confusion matrix for this Random Forest model :

In this confusion matrix one can see that there are 128 True positives and 15 false positives. In this situation, false positives are dangerous as these would actually be diabetic but the model predicted them wrong. But this number is far lower than other values in the matrix.

Therefore the team concluded that the Random Forest Classifier can be used to predict diabetes with good accuracy.

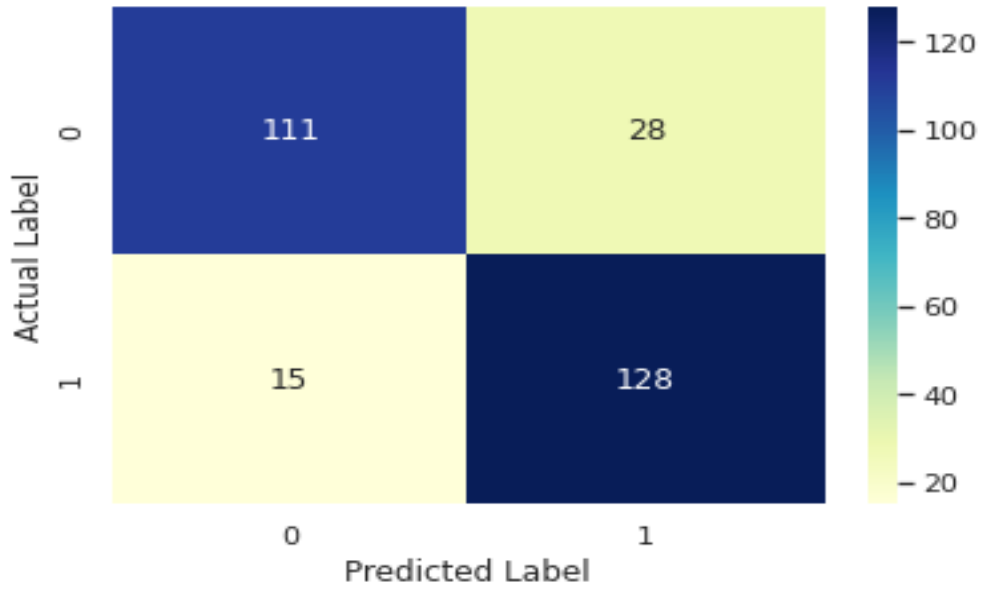

Fig 5.12 Confusion Matrix for Diabetes Model

5.2.3 Heart Disease Prediction :-

Here one can see in the table, the Support Vector Machine Classifier and Random Forest both give an Accuracy score of 88%. But SVC gives better recall value. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. It is a supervised machine learning algorithm capable of performing classification, regression and even outlier detection.

Name	Precision_0	Precision_1	Recall_0	Recall_1	F1-score_0	F1-score_1	Support_0	Support_1	Accuracy
LinearDiscriminantAnalysis	0.88	0.86	0.81	0.91	0.85	0.89	27	34	0.868852459016393
RandomForestClassifier	0.86	0.91	0.89	0.88	0.87	0.9	27	34	0.885245901639344
KneighborsClassifier	0.8	0.9	0.89	0.82	0.84	0.86	27	34	0.852459016393443
DecisionTreeClassifier	0.72	0.81	0.78	0.76	0.75	0.79	27	34	0.770491803278688
GaussianNB	0.85	0.88	0.85	0.88	0.85	0.88	27	34	0.868852459016393
SVC	0.88	0.89	0.85	0.91	0.87	0.9	27	34	0.885245901639344

Table 2: Comparison between Heart Disease Prediction Models

Visualisation of attributes from the table shown above for Heart Disease Prediction:-

The similar metrics which were used for diabetes prediction visualisation have been used for heart disease prediction also.

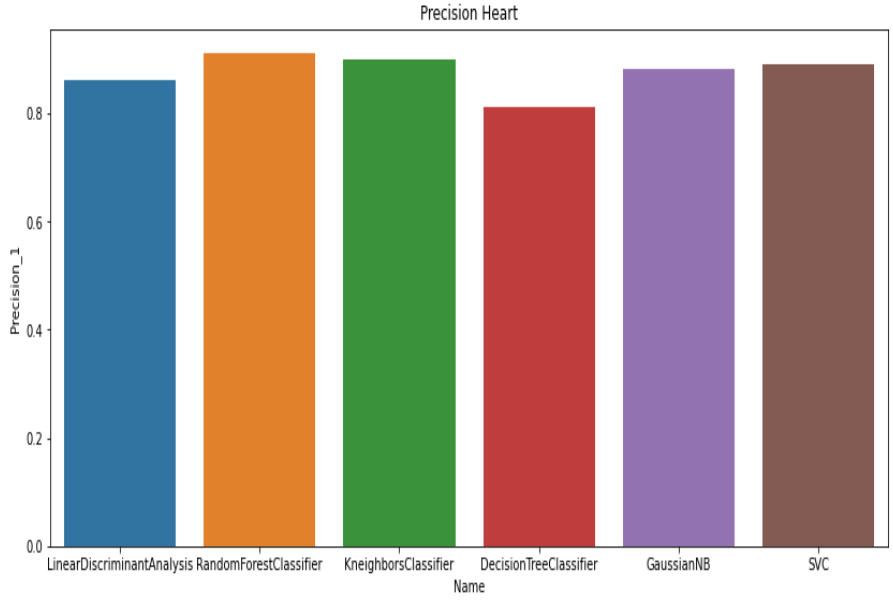

Fig 5.13 Precision of Heart Disease Prediction Model

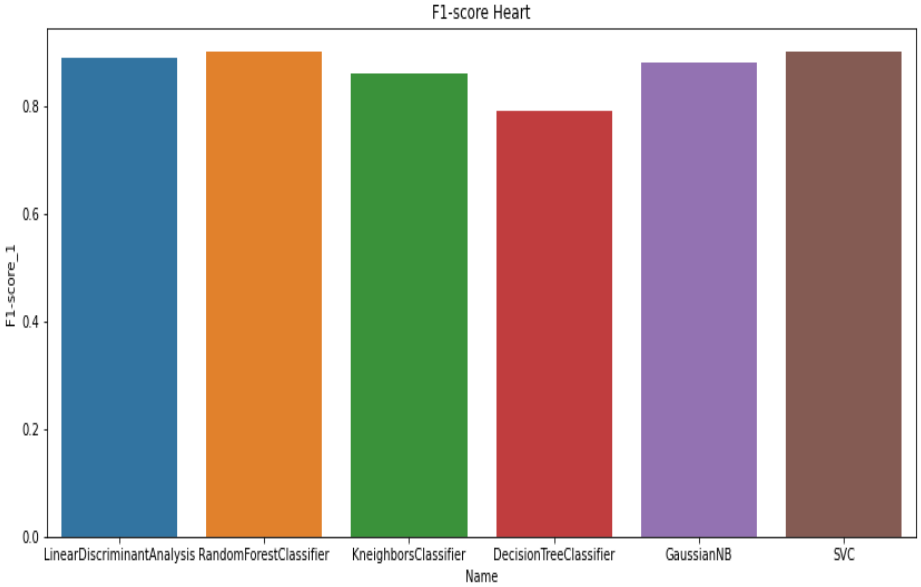

Fig 5.14 F1-Score of Heart Disease Prediction Model

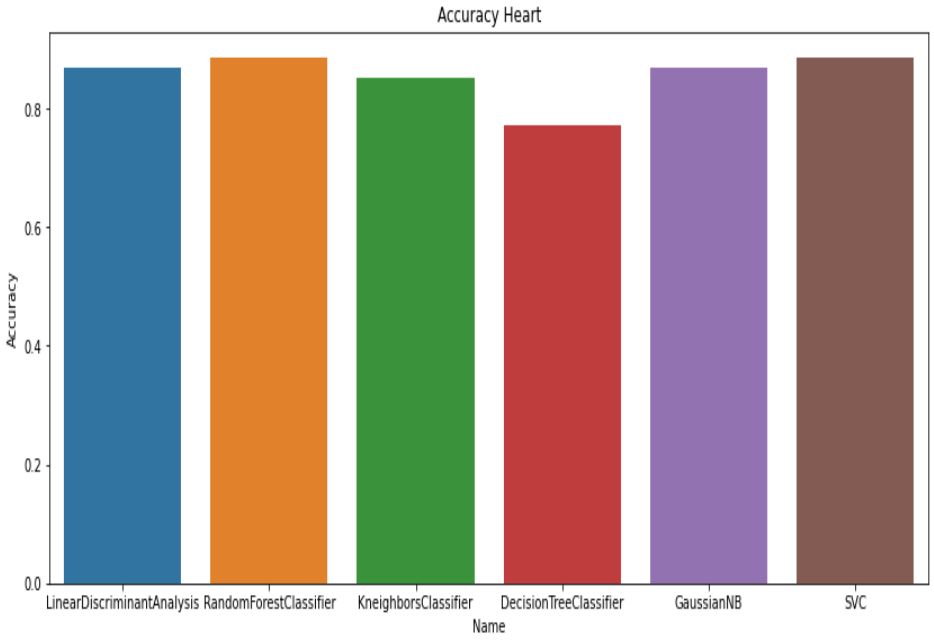

Fig 5.15 Accuracy of Heart Disease Prediction Model

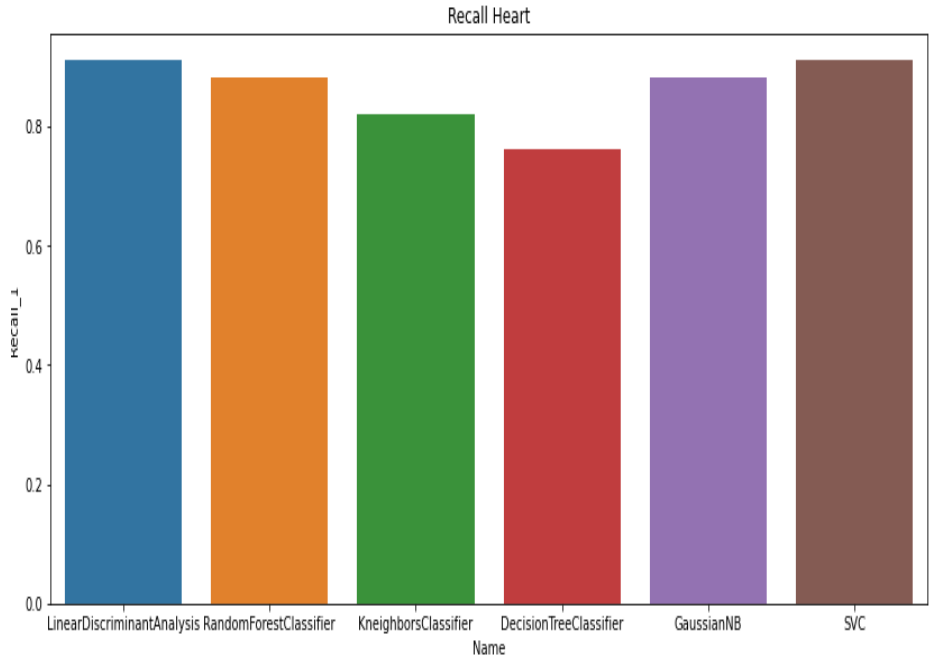

Fig 5.16 Recall of Heart Disease Prediction Model

The confusion matrix for this Support Vector Machine model :

In this confusion matrix one can see that there are 31 True positives and 3 false positives. In this situation, false positives are dangerous as these would actually be a heart patient but the model predicted them wrong. But this number is far lower than other values in the matrix.

Therefore the team concluded that the Support Vector Machine Classifier can be used to predict heart disease with good accuracy.

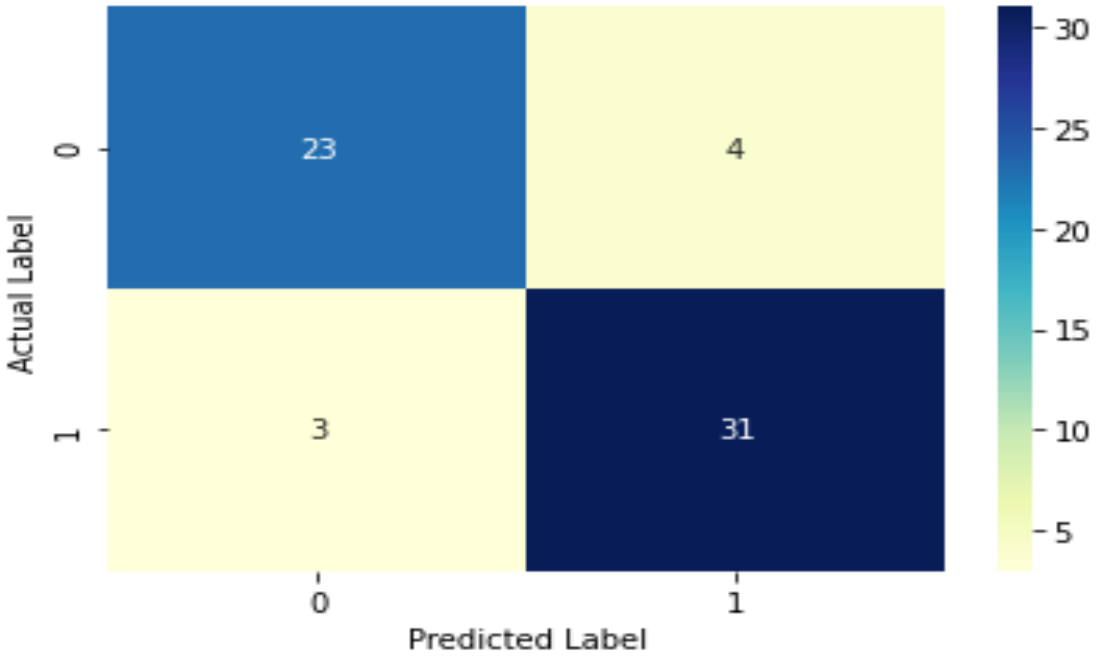

Fig 5.17 Confusion Matrix for Heart Disease Prediction Model

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This project performed to our satisfaction and solved all the problems we aimed to solve. The facial recognition process is fast, efficient and makes it easy to identify a patient and access their personal details and medical history. This will greatly speed up the hospital registration process and help doctors maintain and access patient medical records in a seamless manner. This could also make receptionists at hospitals insignificant in the future. The model also accurately predicted Diabetes and Heart Disease in patients. This could have many real life applications and could serve as a self-analysis test since the easy to use interface only requires patients to enter basic details about themselves and in return tells them if they are at risk of Diabetes and Heart Disease. This model could also serve as a confirmatory test for doctors to see if the results produced by the model coincide with their diagnosis. The accuracy of facial recognition on average was 81%. Diabetes and heart disease prediction accuracy were 84% and 88% respectively.

6.2 Future Work

- Additional models such as brain tumour prediction and classification , Covid-19 detection from x-ray or CT-Scans using images as inputs can be integrated into this project.
- A decentralized record system can be maintained to keep track of patient history and prescriptions in a secure manner. This can help eliminate the need for multiple registrations at different hospitals thereby eliminating time consuming registration processes and medical inconsistencies.
- Facial Recognition can be used in pharmacies to give patients the drugs that are prescribed by the doctor after accessing the hospital's records.

Chapter 7

Bibliography

Research Papers:-

[1] Random Forest Algorithm for the Prediction of Diabetes by Vijayakumar , 2019

[2] Heart disease prediction using machine learning algorithms by Harshit Jindal, Sarthak Agrawal, Rishabh Khera, Rachna Jain and Preeti Nagrath , 2021

[3] Prediction of Diabetes using machine learning Algorithms in Healthcare by Muhammad Azeem Sarwar, Nasir Kamal, Wajeeha Hamid and Munam Ali Shah , 2018

[4] Face Recognition using Face Net (Survey, Performance Test, and Comparison) by Ivan William ,De Rosal Ignatius Moses Setiadi ,Eko Hari Rachmawanto ,Heru Agus Santoso and Christy Atika Sari , 2019

[5] Research on Face Detection Technology based on MTCNN by Ning Zhang , Junmin Luo and Wuqi Gao , 2020

Links:-

https://www.kaggle.com/uciml/pima-indians-diabetes-database

https://towardsdatascience.com/face-detection-using-mtcnn-a-guide-for-face-extraction-with-a-focus-on-speed-c6d59f82d49

https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/

https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/

https://www.kaggle.com/ronitf/heart-disease-uci

https://github.com/davidsandberg/facenet

https://www.cdc.gov/diabetes/basics/quick-facts.html

https://towardsdatascience.com/heart-disease-prediction-73468d630cfc