



Video Based Face Recognition On Real Time Data

A Report for the Evaluation 3 of Project 2

Submitted by –

SYED ABBAS ALI

17SCSE104096

1713104094

BCA

Sem. – 6

In partial fulfillment for the award of the degree

Of

BACHELOR OF COMPUTER APPLICATION

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

**Dr. Saurabh Kumar Srivastava, Ph.D.,
Asst. Professor**

APRIL / MAY - 2020



SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report “**Video Based Face Recognition on Real Time Data**” is the Bonafide work of “**SYED ABBAS ALI**” who carried out the project work under my supervision.

Signature of the Head of the Department

Dr. THIRUNAVUKKARASU KANNAPIRAN,
Ph.D

HEAD OF THE DEPARTMENT

**School of Computing Science &
Engineering**

Signature of the Supervisor

Dr. SAURABH KUMAR SRIVASTAVA,
Ph.D

SUPERVISOR

Asst. Professor

**School of Computing Science &
Engineering**

ACKNOWLEDGEMENT

This project bears the imprint of many persons who are either directly or indirectly involved in successful completion of this project report.

This project would remain incomplete without making acknowledgement to all those who gave me invaluable help and cooperation to hundreds of people whom I interacted during the project.

I am thankful to my project guide Dr. SAURABH KUMAR SRIVASTAVA for his support and cooperation. It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior and acts during the course of project.

Lastly, I would like to thank the almighty and my parents for their moral support and my friends with whom I shared my experience and received lot of suggestions that improved quality of work.

Thank You

ABSTRACT

Video Based Face Recognition on Real-Time Data is a method implementing a search on real time data in order to detect and recognize the face of a person which is previously stored in the system database. The whole process works with the interface of software encoded on a camera (hardware) via different programming languages and library.

The aim of this project to automate the process of face recognition to the simplest level, so that it can be easily available for all kind security purposes and other identity recognizing unit, with the use of similar languages and libraries. The complete process works on the parameters of training and detection of images and then recognizing it.

TABLE OF CONTENT

1. List of symbols and abbreviations
2. List of Tables
3. List of Figures
4. Introduction
5. Identification of need and feasibility
6. Requirement Analysis
7. Proposed Work
8. Hardware and software specification
9. Design of Video-based face recognition on Real-Time data
 - Flow Chart
 - Data Flow Diagram
 - Entity Relationships Diagrams
10. Result and Snapshots
11. Conclusion
12. References
13. Important Codes

1.

LIST OF SYMBOL AND ABBREVIATIONS

Abbreviations	Description
OpenCV	Open Source Computer Vision
DFD	Data Flow Diagram
ER	Entity Relationship
RAM	Random Access Memory
GB	Giga Byte

2.

LIST OF TABLES

- Table 'People' Schema
- Table 'People' Record

3.

LIST OF FIGURES

- The home page
- The home page with instruction
- The Login Page
- The main page
- The main page with instructions
- Face Detection
- Face dataset generation
- Training Phase
- Recognition Phase

4.

INTRODUCTION

Facial recognition is a category of biometric software that maps an individual's facial features mathematically and stores the data as a face-print. The software uses deep learning algorithms to compare a live capture or digital image to the stored face-print in order to verify an individual's identity.

Automatic recognition of human faces is one of the most difficult and important problems in the areas of pattern recognition and computer vision. It can be used as a security mechanism to replace metal key, plastic card, and password or PIN number. It can also be used in criminology to find out possible criminals. However, most face recognition systems require the input face to be free of background and the size to be nearly unchanged. These constraints greatly hinder the usefulness of the system. As we know, a successful face detection process is the prerequisite to facilitate later face recognition task. If we do not have a successful face detection method, we cannot develop a successful face recognition system. Therefore, we should not treat face detection process merely as a preprocessing of a face recognition system.

This project deals with the process of simplified face recognition, with the programming concepts of Python 3.8 and library of programming functions that mainly aims at real-time computer vision that is OpenCV 4.2.0, this library reduces the efforts taken to extract the feature from the face of a person and cut the time from training the extracted images to be stored in the database, hence making the process of face recognition quite simpler and easier. The software used for maintaining the database is SQLite 3.2.1, which is light weight DBMS software that works freely with python. The combination these three things save time and effort.

5.

IDENTIFICATION OF NEED

Positive identification of individuals is a very basic societal requirement. In small tribes and villages, everyone knew and recognized everyone else. You could easily detect a stranger or identify a potential breach of security. In today's larger, more complex society, it isn't that simple. In fact as more interactions take place electronically, it becomes even more important to have an electronic verification of person's identity. Until recently, electronic verification took one of two forms:

- It was based on something the person had in their possession, like a magnetic swipe card
- Something they knew, like a password.

The problem is, these forms of electronic identification aren't very secure, because they can be given away, taken away, or lost and motivated people have found ways to forge or circumvent these credentials. So, the ultimate form of electronic verification of a person's identity is biometrics; using a physical attribute of the person to make a positive identification.

In many applications like surveillance and monitoring, say, of a public place, the traditional biometric techniques will fail as for obvious reasons we cannot ask everyone to come and put his/her thumb on slide or something similar. So we need a system which is similar. So we need a system which is similar to the human face to identify a person. To cater this need and using the observations of human psychophysics, face recognition as a field emerged.

➤ **Feasibility Study** –

After doing the project Video Based Face Recognition on Real-Time Data and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project.

Feasibility study includes consideration of all the possible ways to provide a solution to the given project. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily be done on the future upcoming requirements.

- **Economic Feasibility**

This project is quite economically feasible, the technologies that we have used are all open sourced and easily available on internet. Even the lectures used for explaining these technologies are available free of cost, so that any new user can easily work with it.

- **Technical Feasibility**

This includes the study of function, performance and constraints that may affect the ability to achieve an acceptable system. The hardware configurations are quite feasible and are in range.

- **Operational Feasibility**

The full system is GUI based and is user reliable, user can see his work getting done live in front of them.

6.

REQUIREMENT ANALYSIS

Video Based Face Recognition on Real-Time Data as mentioned to be the work reducing the time of extraction and training before in the report, so to implement it more frequently, the mentioned below are the functional items which specifically required:-

- Working web camera or any external camera
- Python environment se up on the work station
- Availability of OpenCV libraries.
- A DBMS software(SQLite specifically)

The other non functional attributes that can improve the quality of the image stored are:

- Proper Lighting Conditions
- No background disturbances
- Photos should be clicked at different angles

7.

REVIEW OF PREVIOUS WORK

Face Recognition App initially a form of computer application, it has seen wider uses in recent times on mobile platforms and in other forms of technology, such as robotics.

It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. Recently, it has also become popular as a commercial identification and marketing tool

Social media platforms have adopted facial recognition capabilities to diversify their functionalities in order to attract a wider user base amidst stiff competition from different applications.

Few of the previous work done in Face Recognition Technology are –

- The U.S. Department of State operates one of the largest face recognition systems in the world with a database of 117 million American adults, with photos typically drawn from driver's license photos. Although it is still far from completion, it is being put to use in certain cities to give clues as to who was in the photo. The FBI uses the photos as an investigative tool not for positive identification.
- DeepFace is a deep learning facial recognition system created by a research group at Facebook. It identifies human faces in digital images.
- Apple introduced Face ID on the flagship iPhone X as a biometric authentication successor to the Touch ID, a fingerprint based system.

8.

PROPOSED WORK

Modules –

The modules used in this software are as follows:

- **Face Detection:**

This is the first module of the whole process, which detects the faces available on real-time video, this uses the pre defined Cascade classifier of OpenCV , ‘haarcascade_frontalface_default.xml’ which is acronym from classifying the frontal face features and with help of a rectangle.

- **Face Dataset Creator**

This module takes the input from the video, detects the face using above module than takes 30 photo shots, name and unique ID of the input person and converts them to gray scale image and hence storing it in the dataset folder

- **Training**

The training module takes the images stored in the dataset according to the unique ID and extract the features from it so that it can use that classifiers to match faces of similar person, it also uses the library of OpenCV specifically the’ LBPHFaceRecognizer’.

- **Face Recognition**

In this matching module, in real-time we extract features of the input face using the same cascade classifier and recognizer and hence match them with the stored database, to give the result by identifying the face and replying with the proper name of the person

- **Login Module**

This is type of filter module so that no unauthorized user can access this particular software

9. HARDWARE AND SOFTWARE SPECIFICATION

Software Requirements:

- IDLE(Python GUI) 3.8
- OpenCV 4.2.0
- SQLite 3.2.1

Hardware Requirements:

- Processor: core i3 or above
- RAM: 4GB or above
- Hard disk: 50 GB or above
- Web Camera or External camera

Language Used:

- Front End: Python, OpenCV libraries
- Back End: SQLite 3.2.1

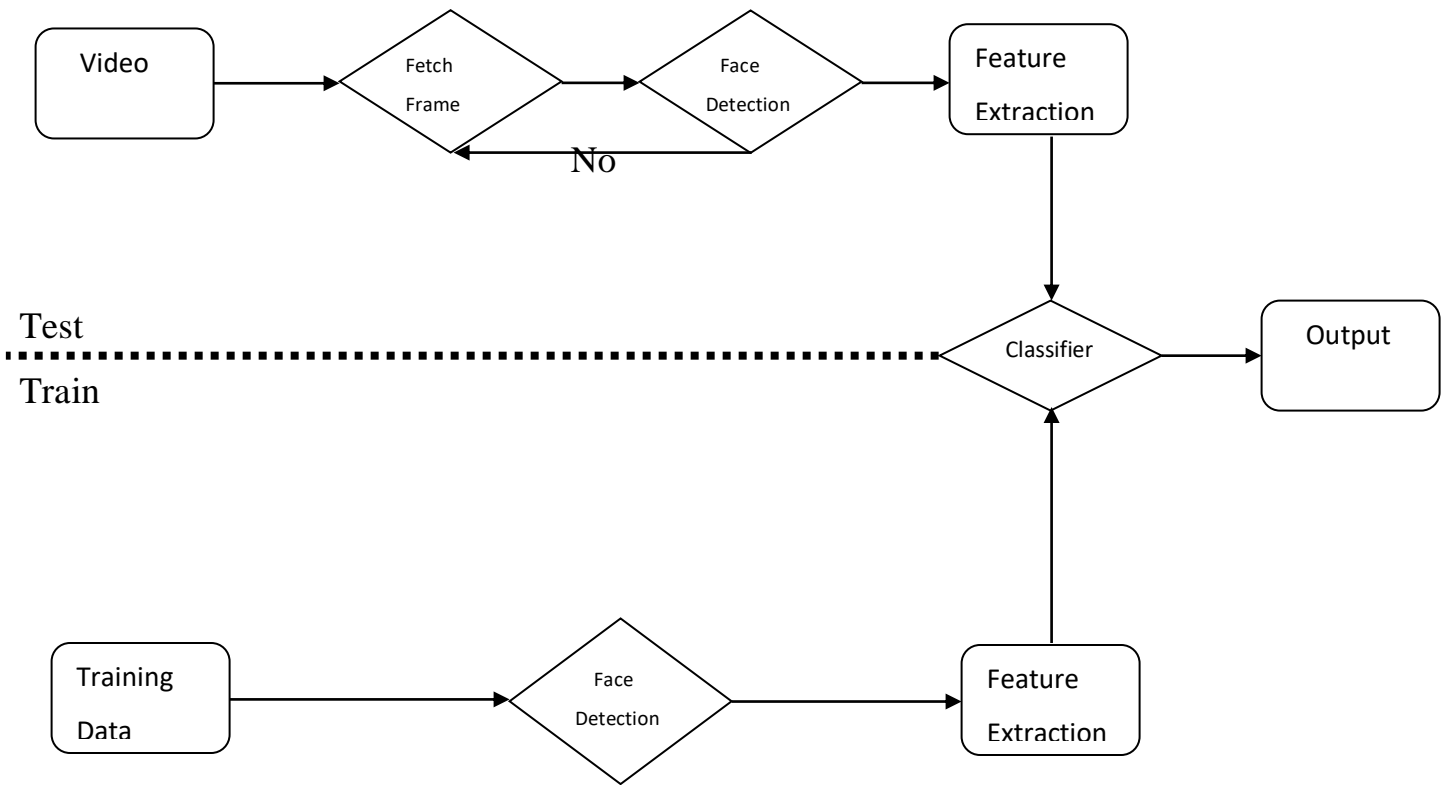
Supported Operating System:

- Windows

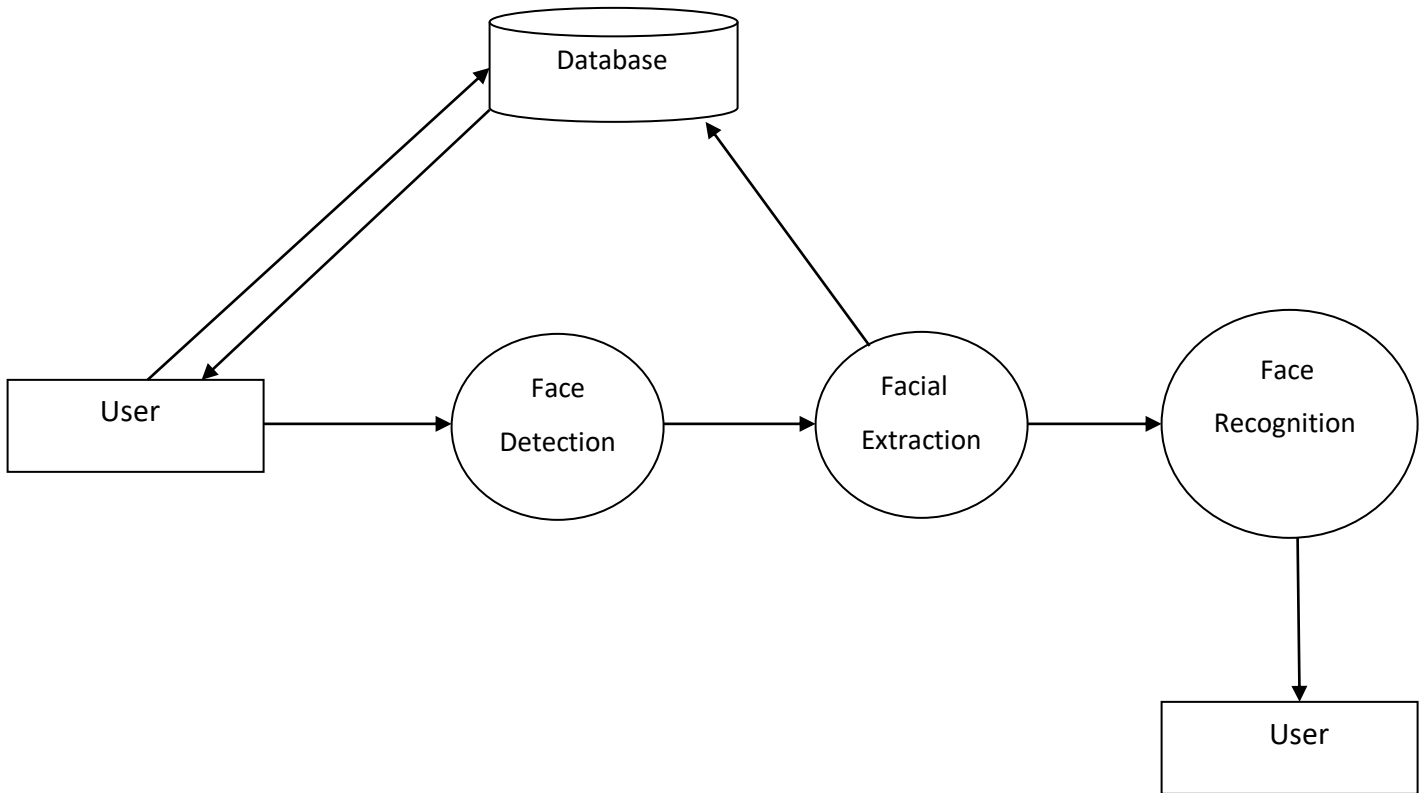
10.

DESIGN

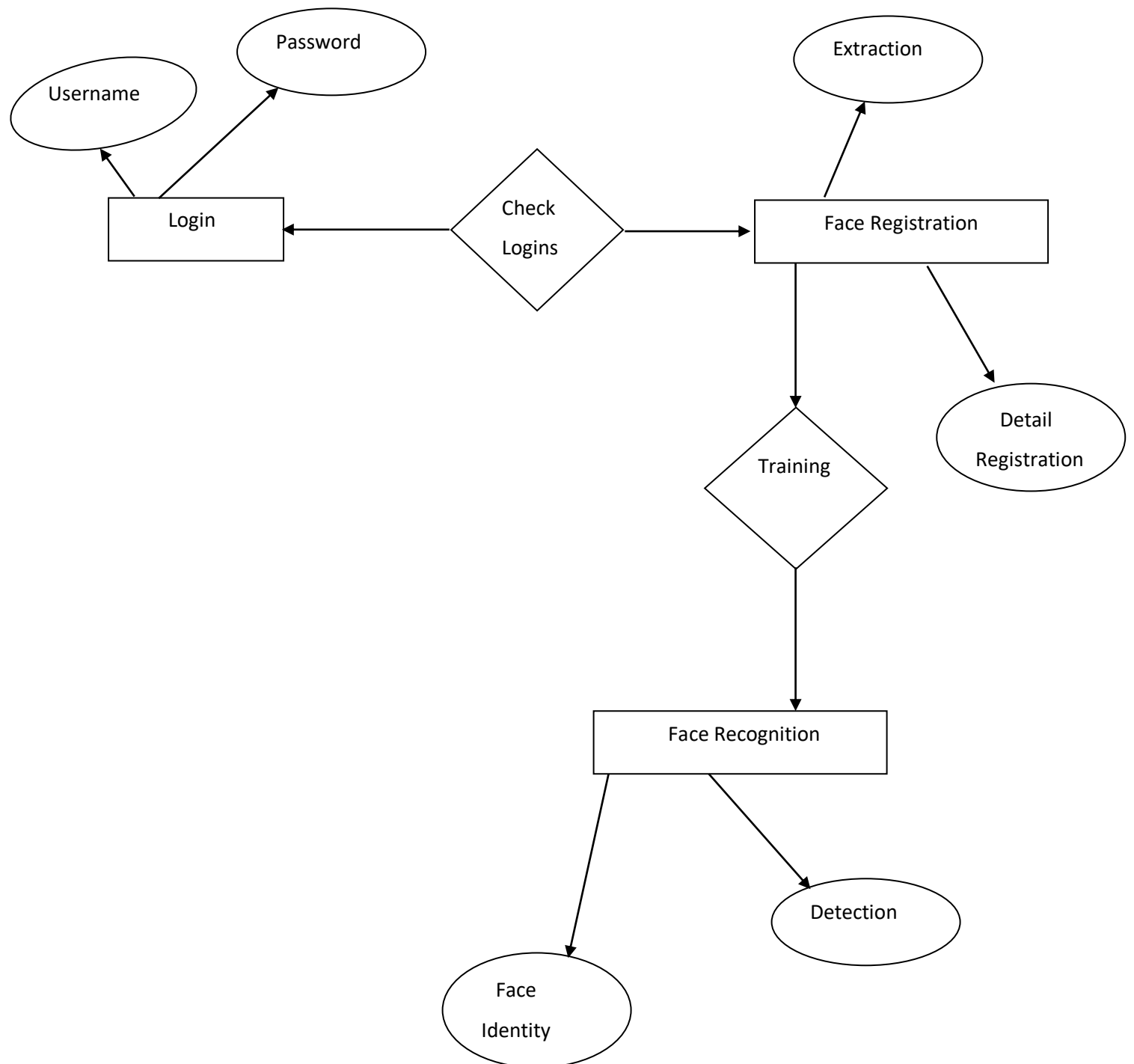
- **Flow Chart**



- **Data Flow Diagram**



- **Entity Relationship Diagram**



11.

RESULT

The Video Based Face Recognition On Real Time Data, was completed successfully, as we enter an input through the web cam, it is detected then processed for data set generation, in which 30 pictures of the input face is taken at one loop, hen in training the feature is extracted from the image and it is compared in the recognition phase. Hence giving us the correct result

12.

SNAPSHOTS

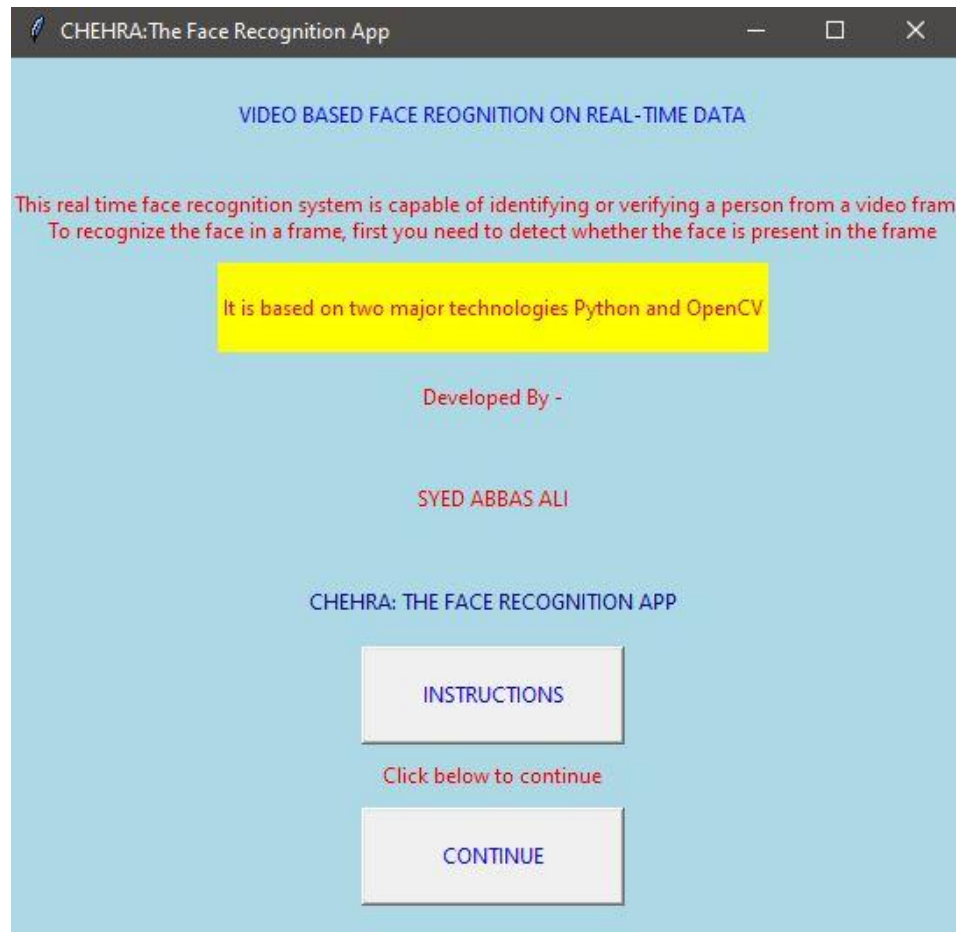


Figure 1: The Home Page

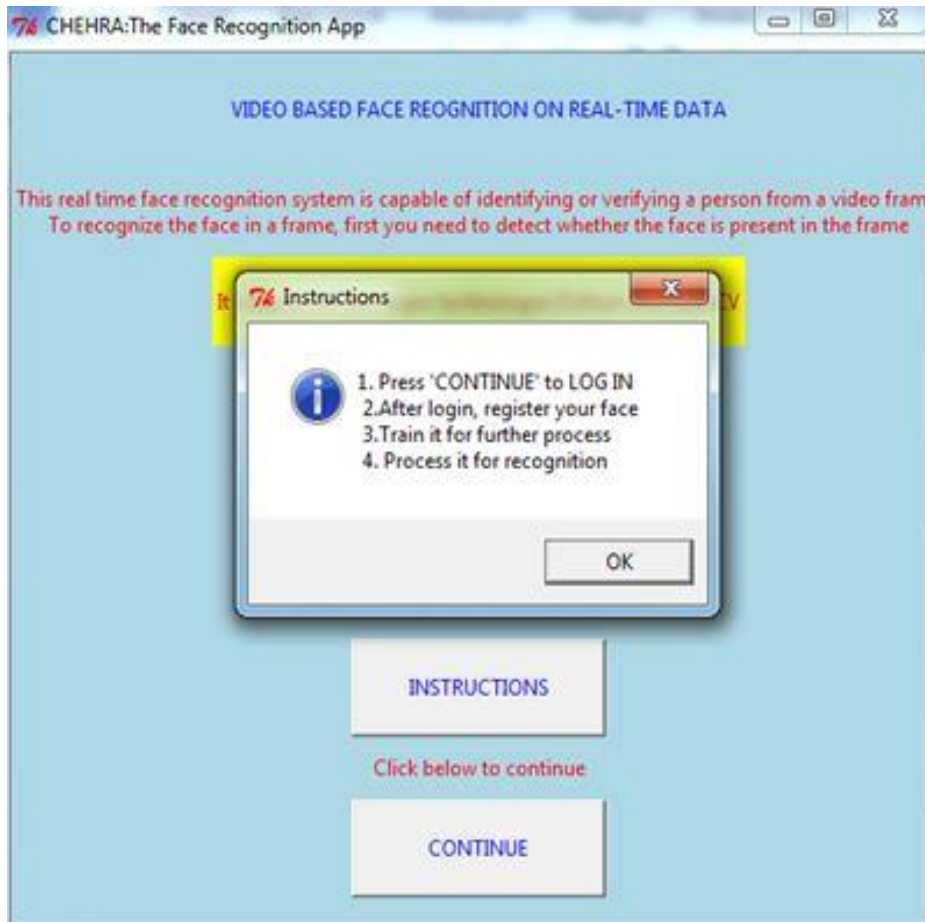


Figure 2: The Home Page with Instructions



Figure 3: Login Page

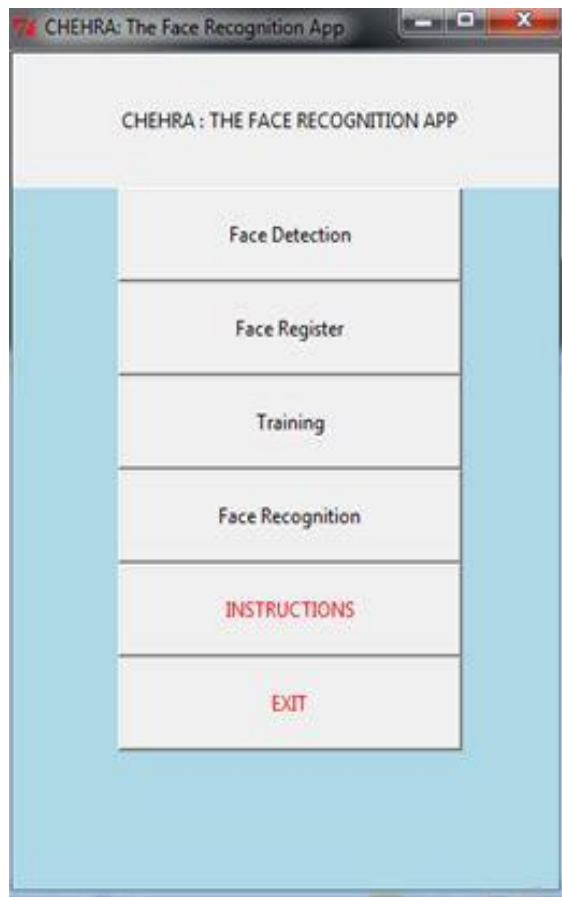


Figure 4: The Main Page

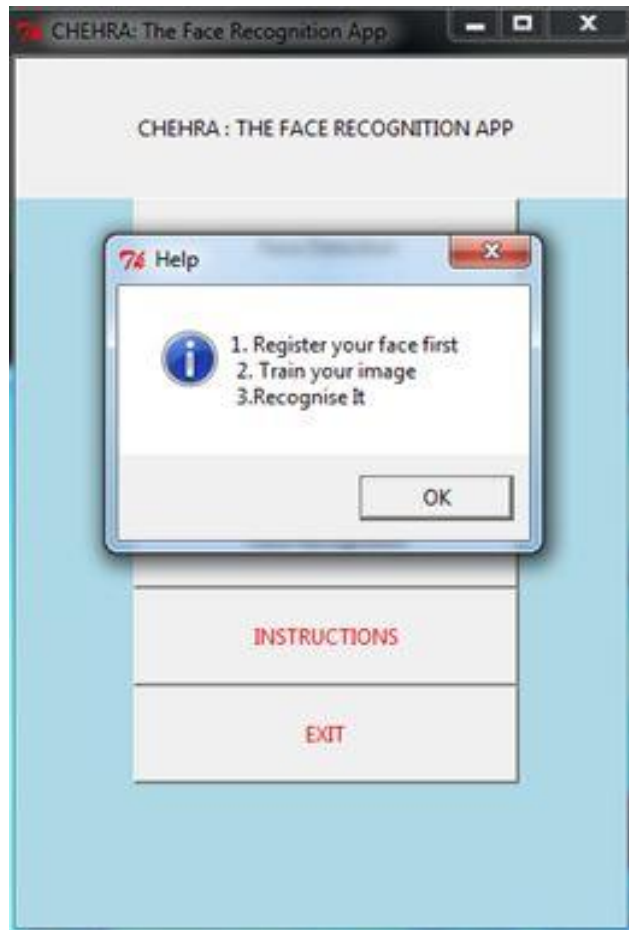


Figure 5: The Main Page with Instruction

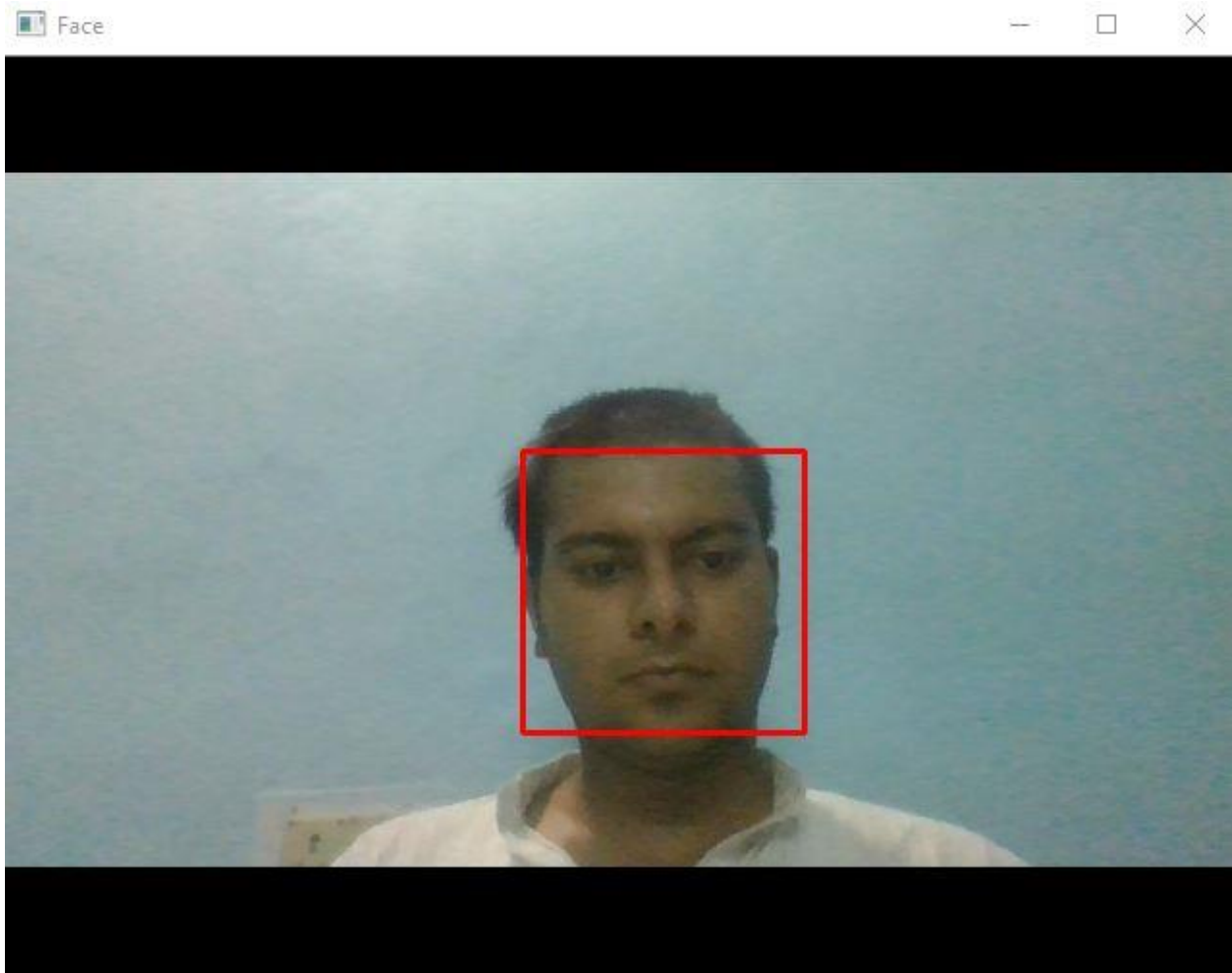


Figure 6: The Face Detection



Figure 7: Training Phase

Table name: WITHOUT ROWID




	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	ID	INTEGER							NULL
2	Name	STRING							NULL

Table 1: The People Schema

	ID	Name	Aqe	Gender	Criminal Record
1	1	SYED ABBAS ALI	22	M	Unknown
2	2	M S DHONI	35	M	Unknown

Table 2: The People Record

13.

CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also been adopted. This project shall prove to be a powerful package in satisfying all the requirements of university. At the end it is concluded we have made effort on following points:

- A description of the background and context of the project and its relation to work already done in this area
- Made statement of aims and objectives of the project
- We defined the problem on which we worked in the project
- We described the requirement specification of the system and the actions that can be done on these things

14.

REFERENCES

- www.tutorialspoint.com/python/
- www.learnpython.org
- <https://www.superdatascience.com/opencv-face-recognition/>
- <https://opencv.org/>
- <https://www.sqlite.org>
- <https://www.tutorialspoint.com/sqlite/>

15.

IMPORTANT CODES

#Dataset Creator

```
import cv2
import numpy as np

faceDetect = cv2.CascadeClassifier('haarcascade_frontalface_default.xml');
cam = cv2.VideoCapture(0);

id= input('enter user id')
sampleNum=0;

while(True):
    ret,img=cam.read();
    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces=faceDetect.detectMultiScale(gray,1.3,5);
    for(x,y,w,h) in faces:
        sampleNum=sampleNum+1;
        cv2.imwrite("dataSet/User."+str(id)+ "."+str(sampleNum)+".jpg",gray[y:y+h,x:x+w])
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
        cv2.waitKey(100);
    cv2.imshow("Face",img);
    cv2.waitKey(1);
    if(sampleNum>20):
        break;

cam.release()
cv2.destroyAllWindows()
```

#Trainer

```
import os
import cv2
import numpy as np
from PIL import Image

recognizer=cv2.face.LBPHFaceRecognizer_create();
path='dataSet'

def getImagesWithID(path):
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    faces=[]
    IDs=[]
    for imagePath in imagePaths:
        faceImg=Image.open(imagePath).convert('L');
        faceNp=np.array(faceImg,'uint8')
        ID=int(os.path.split(imagePath)[-1].split('.')[1])
        faces.append(faceNp)
        print ("ID")
        IDs.append(ID)
        cv2.imshow("training",faceNp)
        cv2.waitKey(10)
    return IDs, faces

Ids,faces=getImagesWithID(path)
recognizer.train(faces,np.array(Ids))
recognizer.save('recognizer/trainingData.yml')
cv2.destroyAllWindows()
```


#Recognizer

```
import cv2
import numpy as np

faceDetect = cv2.CascadeClassifier('haarcascade_frontalface_default.xml');
cam = cv2.VideoCapture(0);
rec=cv2.face.LBPHFaceRecognizer_create();
rec.read("recognizer\\trainingData.yml")
id=0

font=cv2.FONT_HERSHEY_COMPLEX_SMALL,5,1,0,4
while(True):
    ret,img=cam.read();
    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces=faceDetect.detectMultiScale(gray,1.3,5);
    for(x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
        id, conf = rec.predict(gray[y:y+h,x:x+w])
        if(id==1):
            id="Syed"
        elif(id==2):
            id="Dhoni"
        cv2.putText(im, str(id),(x,y+h),font,255);

    cv2.imshow("Face",img);
    if(cv2.waitKey(1)==ord('q')):
        break;

cam.release()
cv2.destroyAllWindows()
```

#The Main Page

```
from tkinter import *
import cv2
import sqlite3
import numpy as np
from PIL import Image
import os
import tkinter.messagebox

root = Tk();
root.title("CHEHRA:The Face Recognition App")
root.configure(background='lightblue')
root.geometry("540x500")

def line():
    execfile("login.py")
def ins():
    tkMessageBox.showinfo("Instructions", "1. Press 'CONTINUE' to LOG IN \n 2.After login,
register your face \n 3.Train it for further process \n 4. Process it for recognition")

b = Button(root, text="CONTINUE", command = line,height=3,width=20,fg="blue")

lb1 = Label(root, text="VIDEO BASED FACE REOGNITION ON REAL-TIME
DATA",height=4,fg='blue',bg='lightblue')
```

```
lb2 = Label(root, text="This real time face recognition system is capable of identifying or  
verifying a person from a video frame.\n\nTo recognize the face in a frame, first you need to detect  
whether the face is present in the frame",height=3,fg='red',bg='lightblue')
```

```
lb3 = Label(root, text="It is based on two major technologies Python and  
OpenCV",height=3,fg='red',bg='yellow')
```

```
lb4 = Label(root, text="Developed By -",height=3,fg='red',bg='lightblue')
```

```
lb5 = Label(root, text="SYED ABBAS ALI",height=4,fg='red',bg='lightblue')
```

```
lb6 = Label(root, text="CHEHRA: THE FACE RECOGNITION  
APP",height=3,fg='DARKBLUE',bg='lightblue')
```

```
lb7 = Label(root, text="Click below to continue",height=2,fg='red',bg='lightblue')
```

```
b1 = Button(root, text="INSTRUCTIONS", command = ins,height=3,width=20,fg="blue")
```

```
lb1.grid()
```

```
lb2.grid()
```

```
lb3.grid()
```

```
lb4.grid()
```

```
lb5.grid()
```

```
lb6.grid()
```

```
b1.grid()
```

```
lb7.grid()
```

```
b.grid()
```

```
root.mainloop()
```