A Thesis/Project/Dissertation Report

on

<u>FACE RECOGNITION SYSTEM</u>

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

B.Tech. - (CSE)

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr. Arvind Dagur

Submitted By

SAURABH TIWARI

19SCSE1010100, 19021011302

KARANJEET SINGH

19SCSE1050012, 19021050102

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT
OF COMPUTER SCIENCE AND ENGINEERING / B.Tech CSE
GALGOTIAS UNIVERSITY, GREATER NOIDA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"FACE RECOGNITION SYSTEM."** In partial fulfillment of the requirements for the award of the **Bachelor of Technology (B. Tech)** submitted in the School of Computing Science and Engineeringof Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name… Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

SAURABH TIWARI - 19SCSE1010100

This is to certify that the above statement made by the candidates is correct to the best of my knowledge

Dr. Arvind Dagur

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **SAURABH TIWARI - 19SCSE1010100** has been held on December, 2021 and his/her work is recommended for the award of Bachelor of Technology(B.Tech - CSE).

Signature of Examiner(s)   **Signature of Supervisor(s)**

Signature of Project Coordinator   **Signature of Dean**

Date: December, 2021

Place: Greater Noida

STATEMENT OF PROJECT REPORT PREPARATION

1. Thesis Title— Face Recognition System
2. Degree for which the report is submitted— Bachelor of Technology (B.Tech

– CSE).
3. Project Supervisor was referred to for preparing the report— Mr. Padmanabhan. P
4. Specifications regarding the thesis / report format have been closely
followed.
5. The contents of the thesis have been organised based on the guidelines
provided.
6. The report has been prepared without resorting to the plagiarism.
7. All the resources used have been cited appropriately.
8. The report has not been submitted elsewhere for any consideration / a degree.

SIGNATURE OF THE STUDENT:-

SAURABH TIWARI

19SCSE1010100

ACKNOWLEDGEMENT

I would like to express my special thank you to my Guide "**Dr. Arvind Dagur Sir**" and my Vice Chancellor "**Mrs. Preeti Bajaj Mam**" who gave me the golden opportunity to work on this wonderful project on the topic "**FACE RECOGNITION SYSTEM**", which helped me in doing a lot of Research in this field and I came to know about various new things and facts. I am really thankful to them. Secondly, but not the least I would like to thank my parents and as well as my friends who helped me a lot in finalizing this superb project within the limited time frame.

Date:- 22/12/2021.   **Name**:- *SAURABH TIWARI*

                                                                                          Admission No. :- *19SCSE1010100*

                                                                                          Enrollment No. :- *19021011302*

<u>ABSTRACT</u>

Face recognition is most demanding field over the past two decades. We can analyze the reason stems from the need for automated visual and monitoring systems which have power to replace all human efforts done manually. These researches involve the knowledge such as neuroscience, computer vision, pattern recognition, machine leaning and many other techniques.

Face recognition is better than other diagnostic procedures namely; fingerprint, iris scanning, signature etc. The face recognition system uses two techniques face detection and face recognitionand combine both to build a face recognition system. The first face detection process is performedon acquired images and it will find the object. The second face recognition process uses algorithm
to find details about a person's face. Thereafter these distinctive details, such as the distance betweenthe eyes, the distance from the mouth to the nose and eyes or the shape of chin and all the specific details is then converted into statistical representation and compared with the data stored in the facial recognition database.

TABLE OF CONTENTS

List of Table

S.No.	Caption	Page No.
1	**Chapter 1**	**12-15**
2	**Chapter 2**	**15-20**
3	**Chapter 3**	**21-32**
4	**Chapter 4**	**33-50**
5	**Chapter 5**	**51-53**

<u>ACRONYMS</u>

ML	**Machine Learning**
SML	**Supervised Machine Learning**
UML	**Unsupervised Machine Learning**
ER	**Entity Relationship Diagram**
UCD	**Use Case Diagram**

CHAPTER-1

<u>Introduction</u>

It is a process/method of verifying and identifying people using their face.

Face recognition system is used to identifying the various faces in video and photos in real-time situation.

The facial recognition program is a part of applications for the processing of facial images and their importance in the field of research is increasing recently. This system uses human biometricinformation and is easier to use instead of fingerprints, iris scanning, signature etc., because thesebiometric forms are not very suitable for those people who are not interested in physical interaction.These systems are commonly used and preferred by people with security cameras in city life. Facerecognition system can be used to prevent crime, video surveillance, personal security, this virtualsystem really change global era to very digitalized form where persons verification, and many other similar security activities provide assured results and it is great achievement and a next stepfor achievement of digital era.

This system has to resolve all the complexities which face during recognition. of individuals as there is lot of cases where some requirements are missing and not fulfilled mean image is not clearor any other traits are not visible , lighting condition, and lots of problem on the images with which software deals .This technology combine both techniques face detection and face matching in image analysis. The detection application is used to detect facial expression in given image. And the matching algorithm is used to distinguished well-constructed images. Face detection method detect face in the given image and Face recognition method is used to extract facial features fromthe given image

like how's the eye , lips, nose looks like . Which makes this algorithm more transparent and complexion than single verification or detection and that task performed by recognition algorithms. The first step we take with the face recognition program is to get a picturefrom the camera. The second step we have to take is face detection in the image we found. And the third step is to recognize the face from output of detection part. The last step is to retain a person's identity as a result of the recognition component.

Face recognition is one of the most significant and best researches in the field of pattern recognitionand computer vision due to its huge applications that apply to the biometric environment, information security control, law enforcement, surveillance system and smart card.

Biometric-based technology has emerged as the most promising option for personal identification in recent years since, instead of verifying people and allowing them access to   physical and text- based domains ,smart cards, and keys, these methods tese a person's personalityor behaviour to found and found who he is. Passwords and Pin's are hard to remember and can be stolen or guessed, cards, keys and the like can be misplaced, forgotten or created twice. However, living human characteristics cannot be misplaced, forgotten or stolen.

Face detection and recognition becomes the most biometrics authentication techniques from the past few years. Face recognition is the process of using the face properties in the biometric systems. Face recognition system has two main tasks: verification and identification. Face identification means a 1:N problem that compares a query face imageagainst all image templates in a face database. We perform face recognition, an extremely complex visual task, almost instantaneously and our own recognition ability is far more robust than any computer's can hope to be. We can recognize a familiar individual undervery adverse lighting conditions, from varying angles or view points. Scaling differences(a face being near or far away), different backgrounds do not affect our ability to recognize faces and we can even recognize individuals with just a fraction of their face visible or even after several years have past. Furthermore, we are able to recognize the faces of several thousand individuals whom we have met during our lifetime. In the Face recognition process the input image is

compared with the database. Theinput image is also called as probe and the database is called as gallery. Then it gives a match report and then the classification is done to identify the sub-population to which new observations belong.

There are basically three approaches for face recognition :

Feature base approach: In feature based approach the local features like nose, eyes aresegmented and it can be used as input data in face detection to easier the task of face recognition.

Holistic approach: In holistic approach the whole face taken as the input in the face detection system to perform face recognition.

Hybrid approach : Hybrid approach is combination of feature based and holisticapproach. In this approach both local and whole face is used as the input to face detection system.

i) **Face recognition using Artificial Intelligence:-** The current technology amazes people with amazing innovations that not only make life simple but also bearable. Face recognition has over time proven to be the least intrusive and fastest form of biometric verification. Facial Recognition is a category of biometric software that maps an individual's facial features and stores the data as a face print. The software uses deep learning algorithms to compare a live captured image to the stored face print to verify one's identity. Image processing and machine learning are the backbones of this technology. Face recognition has received substantial attention from researchers due to human activities found in various applications of security like an airport, criminal detection, face tracking, forensic, etc. Compared to other biometric traits like palm print, iris, fingerprint, etc., face biometrics can be non-intrusive. They can be taken even without the user's knowledge and further can be used for security-based applications like criminal detection, face tracking, airport security, and forensic surveillance systems. Face recognition involves capturing face images from a video or a surveillancecamera. They are compared with the stored database. Face recognition involvestraining known images, classify them with known classes, and then they are stored in the

database. When a test image is given to the system it is classified and compared with the stored database.

ii) **Image Processing and Machine Learning :** Image processing by computers involves the process of Computer Vision. It deals with the high- level understanding of digital images or videos. The requirement is to automate tasks that the human visual systems can do. So, a computer should be able to recognize objects such as that of a face of a human being or a lamppost or even a statue.

iii) **Image reading :** The computer reads any image as a range of values between 0 and 255. For any color image, there are 3 primary colors – Red, green, and blue. A matrix is formed for every primary color and later these matrices combine to provide a Pixel value for the individual R, G, B colors. Each element of the matrices provide data about the intensity of the brightness of the pixel.

iv) **OpenCV:** OpenCV is a Python library that is designed to solve computer vision problems. OpenCV was originally developed in 1999 by Intel but later supported by Willow Garage. OpenCV supports a variety of programming languages such as C++, Python, Java, etc. Support for multiple platforms including Windows, Linux, and macOS. OpenCV Python is a wrapper class for the original C++ library to be used with Python. Using this, all of the OpenCV array structures get converted to/from NumPy arrays. This makes it easier to integrate it with other libraries that use NumPy. For example, libraries such as SciPy and Matplotlib.

v) **Machine learning** : Every Machine Learning algorithm takes a dataset as input and learns from the data it basically means to learn the algorithm from the provided input and output as data. It identifies the patterns in the data and provides the desired algorithm. For instance, to identify whose face is present in a given image, multiple things can be looked at as a pattern:

Height/width of the face.:

Height and width may not be reliable since the image could be rescaled to a smaller faceor grid. However, even after rescaling, what remains unchanged are the ratios – the ratioof the height of the face to the width of the face won't change.

Color of the face.:

Width of other parts of the face like lips, nose, etc.

There is a pattern involved – different faces have different dimensions like the ones above. Similar faces have similar dimensions. Machine Learning algorithms only understand numbers so it is quite challenging. This numerical representation of a "face"(or an element in the training set) is termed as a feature vector. A feature vector comprisesof various numbers in a specific order.

PROBLEM FORMULATION

Face recognition programs identify people with their facial images. Face detection systems create an identity for an authorized person rather than just checking whether an identity document (ID) or key is used or whether the user knows the secret numbers (pins) or passwords. A standard facial vision statement can be made as follows photos provided of motion pictures or video of the group identifying or confirming one or more people in of the event we use a database of facial data Available details of such collateral as the age of race facial expressions and expressions can be used to reduce search improves the realization of problem solving including the division of face-to- face acquisitions from scenes that include feature from the facial region recognition orvalidation of input detection problems is anonymous face and system reports back to a specified id from the database of people who are known when the system verification problems need to be verified or declines the required ownership of the input face.

REQUIRED TOOLS

Digital images and with the help of some libraries:-
OpenCv,
face recognition,
numpy.
matplotlib and using python we can build a face recognition system.

CHAPTER - 2

LITERATURE SURVEY

This section is a basic overview of the major techniques used in the face recognition system thatapply mostly to the front face of the human being.

The methods include neural networks, hidden Markov model, face matching done geometrically and template matching. Eigenface is one of the most widely used methods in face recognition anddetection which are broadly called as the principle components in mathematical terms. The eigenvectors are ordered to represent different amounts of the variations in the faces.Neural networks are highly used in the face recognition and detection systems.

Template matching is one of the techniques through which the test image is represented as a two- dimensional array of values which can be compared using Euclidean distance with single templaterepresenting the whole face. This method can also use more than one face template from different points of view to represent an individual face.
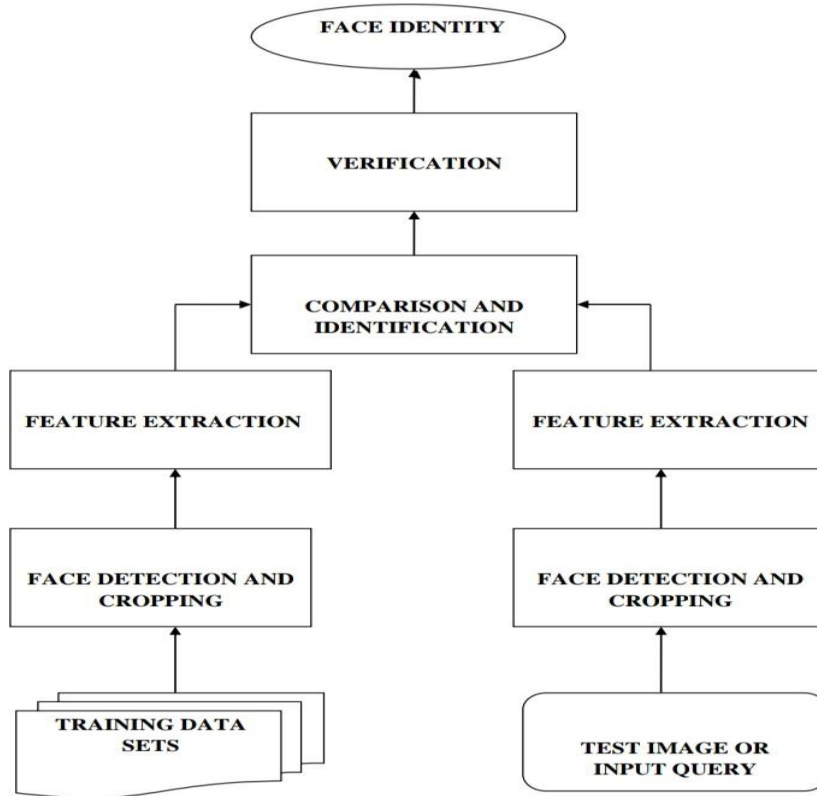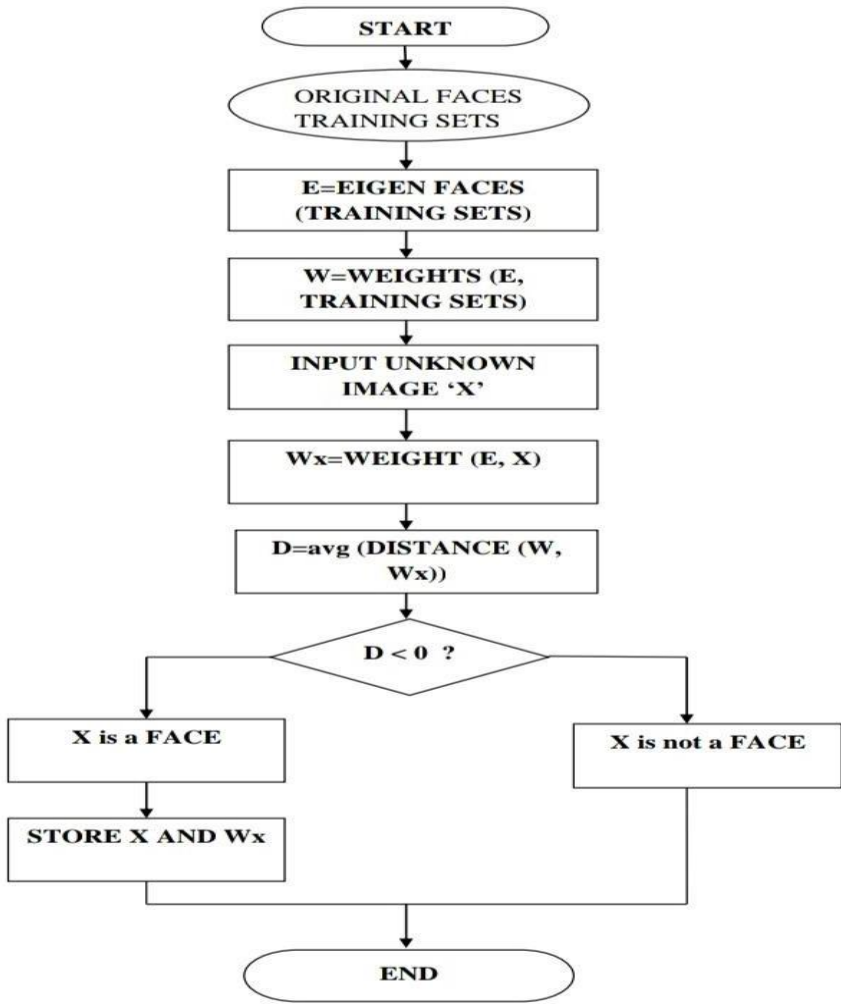
CHAPTER:- 3

PROJECT DESIGN

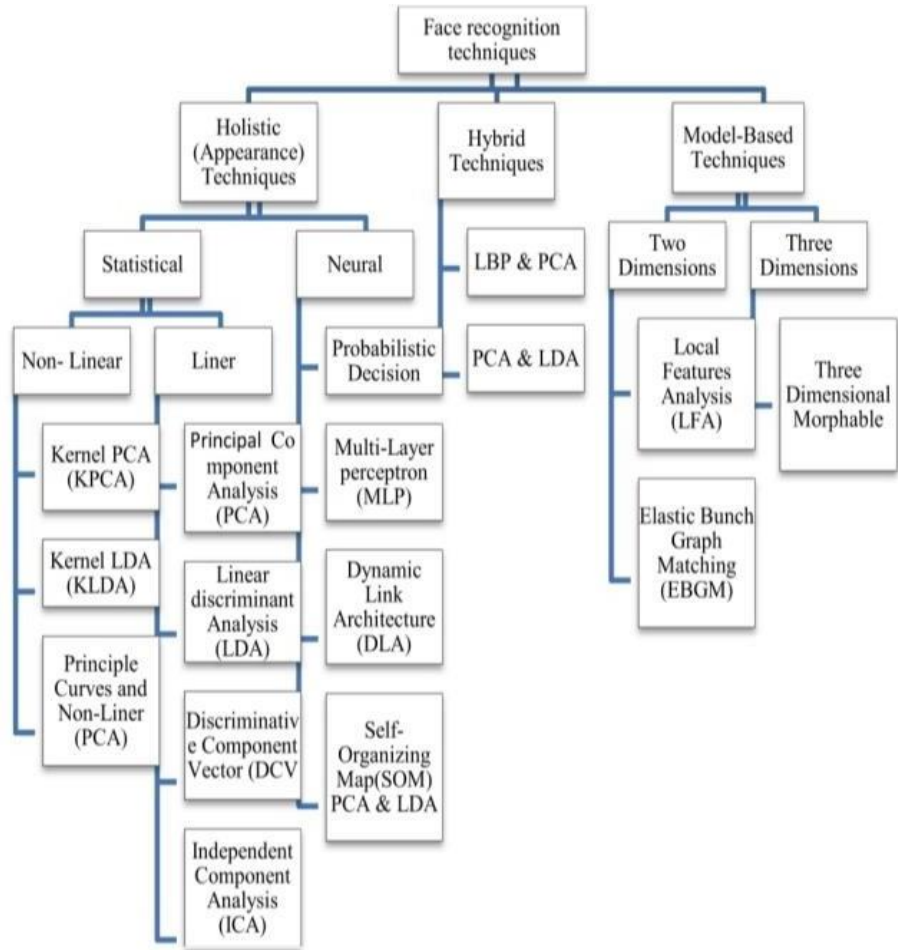**Fig . Architecture Diagram for Proposed Method**

```
                    ┌─────────────────┐
                    │      START      │
                    └─────────────────┘
                             │
                             ▼
                    ┌──────────────────────┐
                    │  ORIGINAL FACES      │
                    │  TRAINING SETS       │
                    └──────────────────────┘
                             │
                             ▼
                    ┌──────────────────────┐
                    │  E=EIGEN FACES       │
                    │  (TRAINING SETS)     │
                    └──────────────────────┘
                             │
                             ▼
                    ┌──────────────────────┐
                    │  W=WEIGHTS (E,       │
                    │  TRAINING SETS)      │
                    └──────────────────────┘
                             │
                             ▼
                    ┌──────────────────────┐
                    │  INPUT UNKNOWN       │
                    │  IMAGE 'X'           │
                    └──────────────────────┘
                             │
                             ▼
                    ┌──────────────────────┐
                    │  Wx=WEIGHT (E, X)    │
                    └──────────────────────┘
                             │
                             ▼
                    ┌──────────────────────┐
                    │  D=avg (DISTANCE (W, │
                    │  Wx))                │
                    └──────────────────────┘
                             │
                             ▼
                        ◇ D < 0 ? ◇
          ┌──────────────┘         └──────────────┐
          ▼                                        ▼
  ┌──────────────────┐                    ┌──────────────────┐
  │  X is a FACE     │                    │  X is not a FACE │
  └──────────────────┘                    └──────────────────┘
          │                                        │
          ▼                                        │
  ┌──────────────────┐                             │
  │ STORE X AND Wx   │                             │
  └──────────────────┘                             │
          └───────────────┬────────────────────────┘
                          ▼
                  ┌─────────────────┐
                  │      END        │
                  └─────────────────┘
```

Fig. Face Recognition approaches

CODE AND ITS IMPLEMENTATION:-

```python
# IMPORTING

import tkinter as tk

from tkinter import ttk

from tkinter import messagebox as mess

import tkinter.simpledialog as tsd

import cv2,os

import csv

import numpy as np

from PIL import Image

import pandas as pd

import datetime

import time


# FUNCTIONS

def assure_path_exists(path):

    dir = os.path.dirname(path)

    if not os.path.exists(dir):

        os.makedirs(dir)



def tick():

    time_string = time.strftime('%H:%M:%S')
```

```
    clock.config(text=time_string)

    clock.after(200,tick)



def contact():

    mess._show(title='Contact us', message="Please contact us on : 'xyz965@gmail.com' ")
```

SAVING OUR DETAILS IN CSV FILE ALSO :-

```
exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")

    if exists1:

        df = pd.read_csv("StudentDetails\StudentDetails.csv")

    else:

        mess._show(title='Details Missing', message='Students details are missing, please check!')

        cam.release()

        cv2.destroyAllWindows()

        window.destroy()

    while True:

        ret, im = cam.read()

        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)

        faces = faceCascade.detectMultiScale(gray, 1.2, 5)

        for (x, y, w, h) in faces:

            cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)

            serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
```
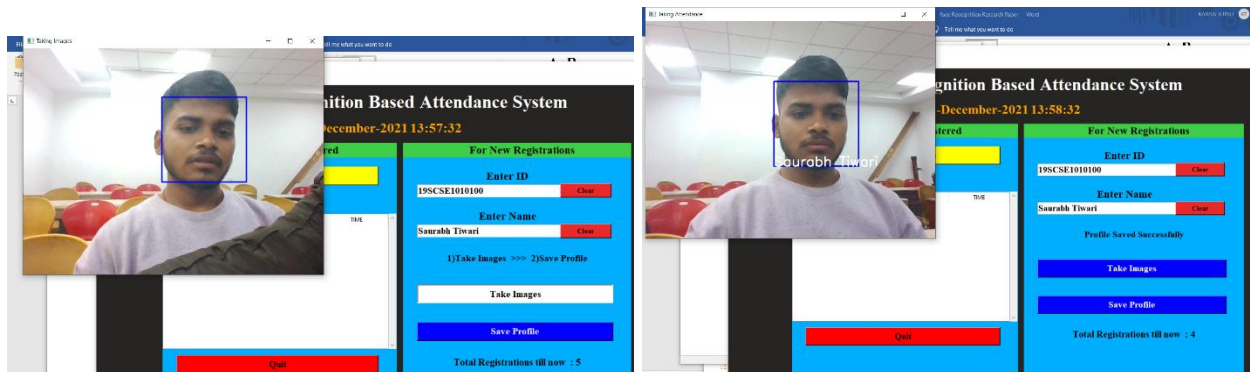
OUTPUT:-

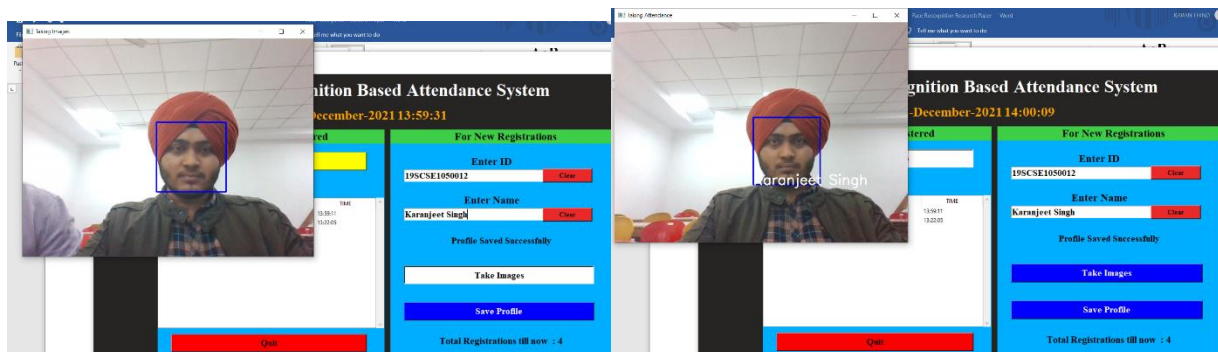## INSERTION OF THE IMAGE:-
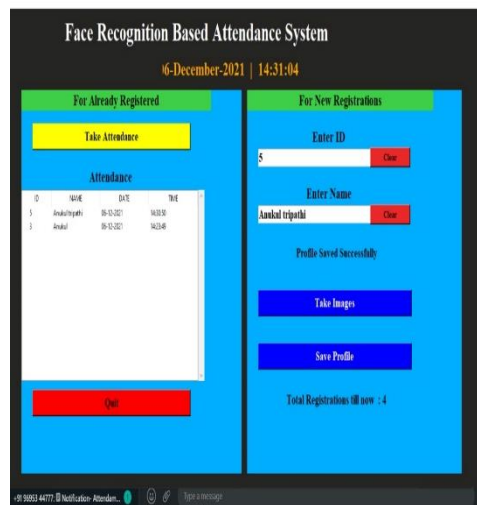## PERSON:-1

IMAGE

IDENTIFICATION:-

PERSON-1

PERSON-2

RECORD OUR STATUS:-
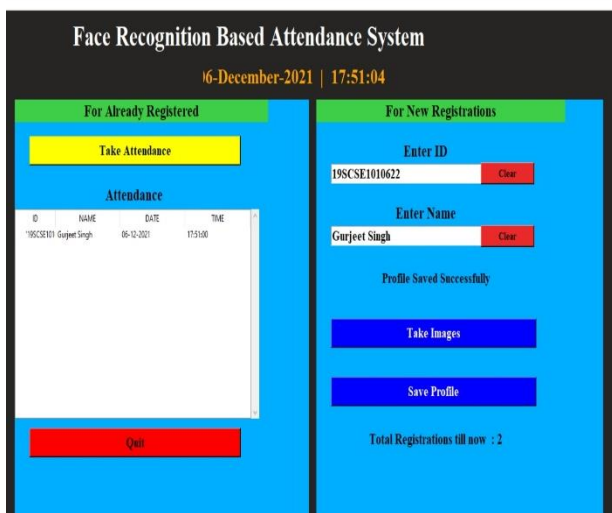
RECORD IN CSV FILE:-

Id		Name	Date		Time
56		Anukul tripathi	########		13:22:05
'19SCSE1010100'		Saurabh Tiwari	########		13:59:11
'19SCSE1050012'		Karanjeet Singh	########		14:00:34
'19SCSE1050012'		Karanjeet Singh	########		14:05:02
'54'		Anukul tripathi	########		15:31:56

CHAPTER-5

FUNCTIONALITY/WORKING OF PROJECT

Source code:-

IMPORTING

###

```python
import tkinter as tk

from tkinter import ttk

from tkinter import messagebox as mess

import tkinter.simpledialog as tsd

import cv2,os

import csv

import numpy as np

from PIL import Image

import pandas as pd

import datetime

import time
```

FUNCTIONS

##

```python
def assure_path_exists(path):
```

```python
    dir = os.path.dirname(path)

    if not os.path.exists(dir):

        os.makedirs(dir)
```

##
####

```python
def tick():

    time_string = time.strftime('%H:%M:%S')

    clock.config(text=time_string)

    clock.after(200,tick)
```

##
#####

```python
def contact():

    mess._show(title='Contact us', message="Please contact us on : 'xyz965@gmail.com' ")
```

##
#####

```python
def check_haarcascadefile():

    exists = os.path.isfile("haarcascade_frontalface_default.xml")
```

```python
    if exists:

        pass

    else:

        mess._show(title='Some file missing', message='Please contact us for help')

        window.destroy()
```

##
#####

```python
def save_pass():

    assure_path_exists("TrainingImageLabel/")

    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")

    if exists1:

        tf = open("TrainingImageLabel\psd.txt", "r")

        key = tf.read()

    else:

        master.destroy()

        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below',
show='*')

        if new_pas == None:

            mess._show(title='No Password Entered', message='Password not set!! Please try again')

        else:

            tf = open("TrainingImageLabel\psd.txt", "w")
```

```python
        tf.write(new_pas)

        mess._show(title='Password Registered', message='New password was registered

successfully!!')

        return

    op = (old.get())

    newp= (new.get())

    nnewp = (nnew.get())

    if (op == key):

        if(newp == nnewp):

            txf = open("TrainingImageLabel\psd.txt", "w")

            txf.write(newp)

        else:

            mess._show(title='Error', message='Confirm new password again!!!')

            return

    else:

        mess._show(title='Wrong Password', message='Please enter correct old password.')

        return

    mess._show(title='Password Changed', message='Password changed successfully!!')

    master.destroy()
```

##

#####

```python
def change_pass():

    global master

    master = tk.Tk()

    master.geometry("400x160")

    master.resizable(False,False)

    master.title("Change Password")

    master.configure(background="white")

    lbl4 = tk.Label(master,text='    Enter Old Password',bg='white',font=('times', 12, ' bold '))

    lbl4.place(x=10,y=10)

    global old

    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show='*')

    old.place(x=180,y=10)

    lbl5 = tk.Label(master, text='   Enter New Password', bg='white', font=('times', 12, ' bold '))

    lbl5.place(x=10, y=45)

    global new

    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold
'),show='*')

    new.place(x=180, y=45)

    lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))

    lbl6.place(x=10, y=80)

    global nnew

    nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold
'),show='*')
```

```python
    nnew.place(x=180, y=80)

    cancel=tk.Button(master,text="Cancel",
command=master.destroy ,fg="black"  ,bg="red" ,height=1,width=25 , activebackground =
"white" ,font=('times', 10, ' bold '))

    cancel.place(x=200, y=120)

    save1 = tk.Button(master, text="Save", command=save_pass, fg="black", bg="#3ece48",
height = 1,width=25, activebackground="white", font=('times', 10, ' bold '))

    save1.place(x=10, y=120)

    master.mainloop()


####################################################################################
#######

def psw():
    assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
        tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
        new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below',
show='*')
        if new_pas == None:
```

```python
        mess._show(title='No Password Entered', message='Password not set!! Please try again')

    else:

        tf = open("TrainingImageLabel\psd.txt", "w")

        tf.write(new_pas)

        mess._show(title='Password Registered', message='New password was registered
successfully!!')

        return

    password = tsd.askstring('Password', 'Enter Password', show='*')

    if (password == key):

        TrainImages()

    elif (password == None):

        pass

    else:

        mess._show(title='Wrong Password', message='You have entered wrong password')


##############################################################################
########


def clear():

    txt.delete(0, 'end')

    res = "1)Take Images  >>>  2)Save Profile"

    message1.configure(text=res)
```

```python
def clear2():

    txt2.delete(0, 'end')

    res = "1)Take Images  >>>  2)Save Profile"

    message1.configure(text=res)


###############################################################################
#########


def TakeImages():

    check_haarcascadefile()

    columns = ['SERIAL NO.', '', 'ID', '', 'NAME']

    assure_path_exists("StudentDetails/")

    assure_path_exists("TrainingImage/")

    serial = 0

    exists = os.path.isfile("StudentDetails\StudentDetails.csv")

    if exists:

        with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:

            reader1 = csv.reader(csvFile1)

            for l in reader1:

                serial = serial + 1

        serial = (serial // 2)

        csvFile1.close()
```

```python
else:
    with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:

        writer = csv.writer(csvFile1)

        writer.writerow(columns)

        serial = 1

    csvFile1.close()

Id = (txt.get())

name = (txt2.get())

if ((name.isalpha()) or (' ' in name)):

    cam = cv2.VideoCapture(0)

    harcascadePath = "haarcascade_frontalface_default.xml"

    detector = cv2.CascadeClassifier(harcascadePath)

    sampleNum = 0

    while (True):

        ret, img = cam.read()

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = detector.detectMultiScale(gray, 1.3, 5)

        for (x, y, w, h) in faces:

            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

            # incrementing sample number

            sampleNum = sampleNum + 1

            # saving the captured face in the dataset folder TrainingImage
```

```python
        cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' +
str(sampleNum) + ".jpg",

                gray[y:y + h, x:x + w])
        # display the frame
        cv2.imshow('Taking Images', img)
    # wait for 100 miliseconds
    if cv2.waitKey(100) & 0xFF == ord('q'):
        break
    # break if the sample number is morethan 100
    elif sampleNum > 100:
        break
cam.release()
cv2.destroyAllWindows()
res = "Images Taken for ID : " + Id
row = [serial, '', Id, '', name]
with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
    writer = csv.writer(csvFile)
    writer.writerow(row)
csvFile.close()
message1.configure(text=res)
else:
    if (name.isalpha() == False):
        res = "Enter Correct name"
```

```python
        message.configure(text=res)


##############################################################################
##########


def TrainImages():
    check_haarcascadefile()
    assure_path_exists("TrainingImageLabel/")
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, ID = getImagesAndLabels("TrainingImage")
    try:
        recognizer.train(faces, np.array(ID))
    except:
        mess._show(title='No Registrations', message='Please Register someone first!!!')
        return
    recognizer.save("TrainingImageLabel\Trainner.yml")
    res = "Profile Saved Successfully"
    message1.configure(text=res)
    message.configure(text='Total Registrations till now  : ' + str(ID[0]))
```

```python
###############################################################################
##############3


def getImagesAndLabels(path):
    # get the path of all the files in the folder
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faces = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
        ID = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(ID)
    return faces, Ids
```

```python
#################################################################################
#############

def TrackImages():

    check_haarcascadefile()

    assure_path_exists("Attendance/")

    assure_path_exists("StudentDetails/")

    for k in tv.get_children():

        tv.delete(k)

    msg = ''

    i = 0

    j = 0

    recognizer = cv2.face.LBPHFaceRecognizer_create()  # cv2.createLBPHFaceRecognizer()

    exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")

    if exists3:

        recognizer.read("TrainingImageLabel\Trainner.yml")

    else:

        mess._show(title='Data Missing', message='Please click on Save Profile to reset data!!')

        return

    harcascadePath = "haarcascade_frontalface_default.xml"

    faceCascade = cv2.CascadeClassifier(harcascadePath);


    cam = cv2.VideoCapture(0)
```

```python
font = cv2.FONT_HERSHEY_SIMPLEX

col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']

exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")

if exists1:

    df = pd.read_csv("StudentDetails\StudentDetails.csv")

else:

    mess._show(title='Details Missing', message='Students details are missing, please check!')

    cam.release()

    cv2.destroyAllWindows()

    window.destroy()

while True:

    ret, im = cam.read()

    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(gray, 1.2, 5)

    for (x, y, w, h) in faces:

        cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)

        serial, conf = recognizer.predict(gray[y:y + h, x:x + w])

        if (conf < 50):

            ts = time.time()

            date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

            timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

            aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values

            ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
```

```python
            ID = str(ID)

            ID = ID[1:-1]

            bb = str(aa)

            bb = bb[2:-2]

            attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]


        else:

            Id = 'Unknown'

            bb = str(Id)

        cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)

    cv2.imshow('Taking Attendance', im)

    if (cv2.waitKey(1) == ord('q')):

        break

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")

if exists:

    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:

        writer = csv.writer(csvFile1)

        writer.writerow(attendance)

    csvFile1.close()

else:

    with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
```

```python
        writer = csv.writer(csvFile1)

        writer.writerow(col_names)

        writer.writerow(attendance)

    csvFile1.close()

with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:

    reader1 = csv.reader(csvFile1)

    for lines in reader1:

        i = i + 1

        if (i > 1):

            if (i % 2 != 0):

                iidd = str(lines[0]) + '   '

                tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))

    csvFile1.close()

    cam.release()

    cv2.destroyAllWindows()


########################################### USED STUFFS

##############################################


global key

key = ''


ts = time.time()
```

```python
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')

day,month,year=date.split("-")


mont={'01':'January',
    '02':'February',
    '03':'March',
    '04':'April',
    '05':'May',
    '06':'June',
    '07':'July',
    '08':'August',
    '09':'September',
    '10':'October',
    '11':'November',
    '12':'December'
    }


######################################## GUI FRONT-END
#############################################


window = tk.Tk()
window.geometry("1280x720")
window.resizable(True,False)
```

```python
window.title("Attendance System")

window.configure(background='#262523')


frame1 = tk.Frame(window, bg="#00aeff")

frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)


frame2 = tk.Frame(window, bg="#00aeff")

frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)


message3 = tk.Label(window, text="Face Recognition Based Attendance

System" ,fg="white",bg="#262523" ,width=55 ,height=1,font=('times', 29, ' bold '))

message3.place(x=10, y=10)


frame3 = tk.Frame(window, bg="#c4c6ce")

frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)


frame4 = tk.Frame(window, bg="#c4c6ce")

frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)


datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+"  |  ",

fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))

datef.pack(fill='both',expand=1)
```

```
clock = tk.Label(frame3,fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold
'))

clock.pack(fill='both',expand=1)

tick()


head2 = tk.Label(frame2, text="                    For New Registrations                    ",
fg="black",bg="#3ece48" ,font=('times', 17, ' bold ') )

head2.grid(row=0,column=0)


head1 = tk.Label(frame1, text="                    For Already Registered                    ",
fg="black",bg="#3ece48" ,font=('times', 17, ' bold ') )

head1.place(x=0,y=0)


lbl = tk.Label(frame2, text="Enter

ID",width=20 ,height=1 ,fg="black" ,bg="#00aeff" ,font=('times', 17, ' bold ') )

lbl.place(x=80, y=55)


txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))

txt.place(x=30, y=88)


lbl2 = tk.Label(frame2, text="Enter Name",width=20 ,fg="black" ,bg="#00aeff" ,font=('times',
17, ' bold '))

lbl2.place(x=80, y=140)
```

```python
txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ')  )

txt2.place(x=30, y=173)


message1 = tk.Label(frame2, text="1)Take Images  >>>  2)Save

Profile" ,bg="#00aeff" ,fg="black"  ,width=39 ,height=1, activebackground =

"yellow" ,font=('times', 15, ' bold '))

message1.place(x=7, y=230)


message = tk.Label(frame2, text="" ,bg="#00aeff" ,fg="black"  ,width=39,height=1,

activebackground = "yellow" ,font=('times', 16, ' bold '))

message.place(x=7, y=450)


lbl3 = tk.Label(frame1,

text="Attendance",width=20  ,fg="black"  ,bg="#00aeff"  ,height=1 ,font=('times', 17, ' bold '))

lbl3.place(x=100, y=115)


res=0

exists = os.path.isfile("StudentDetails\StudentDetails.csv")

if exists:

    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:

        reader1 = csv.reader(csvFile1)

        for l in reader1:
```

```python
        res = res + 1

    res = (res // 2) - 1

    csvFile1.close()

else:

    res = 0

message.configure(text='Total Registrations till now  : '+str(res))




######################### MENUBAR ##################################


menubar = tk.Menu(window,relief='ridge')

filemenu = tk.Menu(menubar,tearoff=0)

filemenu.add_command(label='Change Password', command = change_pass)

filemenu.add_command(label='Contact Us', command = contact)

filemenu.add_command(label='Exit',command = window.destroy)

menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)




################### TREEVIEW ATTENDANCE TABLE ####################


tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))

tv.column('#0',width=82)

tv.column('name',width=130)

tv.column('date',width=133)

tv.column('time',width=133)
```

```
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)

tv.heading('#0',text ='ID')

tv.heading('name',text ='NAME')

tv.heading('date',text ='DATE')

tv.heading('time',text ='TIME')
```

```
###################### SCROLLBAR ##############################
```

```
scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)

scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')

tv.configure(yscrollcommand=scroll.set)
```

```
###################### BUTTONS ##############################
```

```
clearButton = tk.Button(frame2, text="Clear",

command=clear ,fg="black" ,bg="#ea2a2a" ,width=11 ,activebackground =

"white" ,font=('times', 11, ' bold '))

clearButton.place(x=335, y=86)

clearButton2 = tk.Button(frame2, text="Clear",

command=clear2 ,fg="black" ,bg="#ea2a2a" ,width=11 , activebackground =

"white" ,font=('times', 11, ' bold '))

clearButton2.place(x=335, y=172)
```

```python
takeImg = tk.Button(frame2, text="Take Images",

command=TakeImages ,fg="white" ,bg="blue" ,width=34 ,height=1, activebackground =

"white" ,font=('times', 15, ' bold '))

takeImg.place(x=30, y=300)

trainImg = tk.Button(frame2, text="Save Profile",

command=psw ,fg="white" ,bg="blue" ,width=34 ,height=1, activebackground =

"white" ,font=('times', 15, ' bold '))

trainImg.place(x=30, y=380)

trackImg = tk.Button(frame1, text="Take Attendance",

command=TrackImages ,fg="black" ,bg="yellow" ,width=35 ,height=1, activebackground =

"white" ,font=('times', 15, ' bold '))

trackImg.place(x=30,y=50)

quitWindow = tk.Button(frame1, text="Quit",

command=window.destroy ,fg="black" ,bg="red" ,width=35 ,height=1, activebackground =

"white" ,font=('times', 15, ' bold '))

quitWindow.place(x=30, y=450)


###################### END #######################################

window.configure(menu=menubar)

window.mainloop()
```

##
##############

<u>CONCLUSION</u>

1.) The first conclusion we conclude is related to cost-Generally we used face recognition less system in very expensive top secure systems.

2.) Some factors like environmental changes and mild changes in appearance due to age or an incidence happen impact the technology to a greater degree than many expect.

3.) This one of the fast and convenient method of biometric identification and has greater use in society.

4.) It will find the faces with exhaustively wasting time in searching of images.

5.) This system has widespread application in the smart environment.

REFERENCES

[1]   First reference we take from wikipedia their are some articles on wikipedia which we take into   consideration for making our research paper.

List of those article are following:-

Article published by Cambridge University-The Quest For Artificial Intelligence.

A hand book name as The history of information Security.

[1]   We Use Some book for making our research paper .

These books are :-

i. Face Detection and Recognition theory and practice -writer Asit kumar Datta, Madhura Datta, Pradipta Kumar Banerjee.

ii. Face detection and Recognition on Mobile Devices-writer Haowei Liu.

iii.Our Biometric Future by Gates kelly.

[2]   We use some previously published IEEE papers into consideration for making our research paper.

[3]   https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9

[4]   An international journal of Bio-Science and Technology.

[5]   DeepFace a face recognition system develop by facebook.

THANK YOU!!!