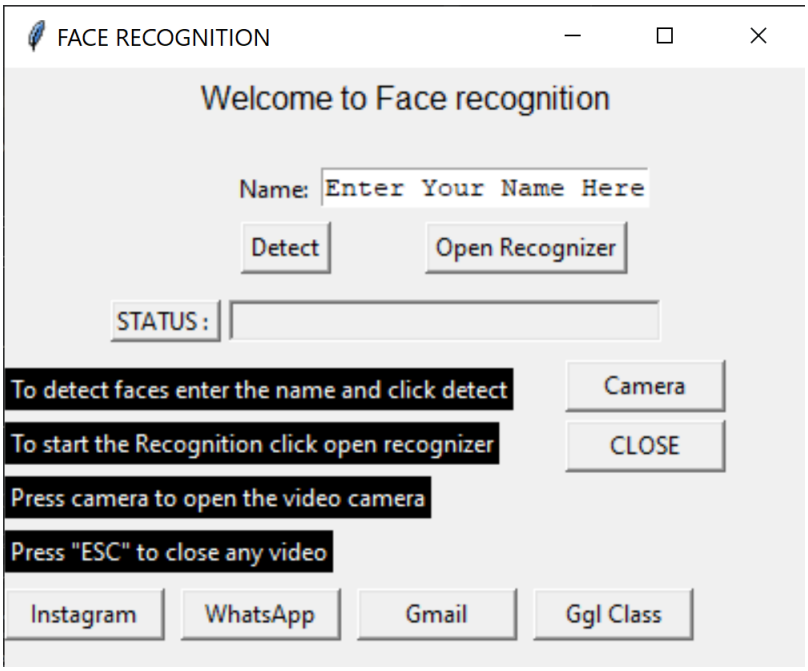
Face recognition and Application

This project is about face detection and recognition using python and the applications of it. This project uses face detection for the opening of every apps or software using a common framework i.e the software will be opened if and only if the face of a person is present in-front of the camera.



Features available

- Detection

  If we have to add a new face into the data then we have to just enter the name of the person in the given text box and click ‘detect’ button then the person has to just look at the camera, a camera window will be opened and it will take the image samples. Then the persons face will be added in the data.

- Recognition

  We can check whether our faces are available in the data by just clicking the ‘open recognizer’ button , a camera window will be opened and it will display the name of the face recognized if the data is available.

- Camera

  This option is for just opening a camera and it will detect the faces available in the video by drawing a yellow rectangle around the faces.

- Status

  This displays the process and what the program is actually doing.

- Instagram

  When this button is clicked it will recognize whether the face is present in-front of the camera. If present it will open the Instagram page if not it will show ‘Face not Found’ message in the status box

- Whatsapp

  When this button is clicked it will recognize whether the face is present in-front of the camera. If present it will open the whatsapp page if not it will show ‘Face not Found’ message in the status box

- Gmail

  When this button is clicked it will recognize whether the face is present in-front of the camera. If present it will open the G-mail page if not it will show ‘Face not Found’ message in the status box

- Ggl class

  When this button is clicked it will recognize whether the face is present in-front of the camera. If present it will open the Google Classroom page if not it will show ‘Face not Found’ message in the status box

Modules Used

- ✓ Tkinter

  This library is used to design the graphical user interface which acts a platform of communication between the machine and the man.

- ✓ Open cv

  This is the library where the detection training and recognition of the images are done with the help of cascade classifiers.

- ✓ PIL

  This library is used for importing the sample image saved in the system and processing it.

- ✓ Numpy

  All the images are nothing but a three dimensional array. That is the image will saved as a three 2 dimensional array in z dimension with colour codes like RGB,BGR. So in the processing and editing of images numpy is used

- ✓ Os

  To define the path of the images present in the system the os library is used.

- ✓ Webbrowser

  In this program I used the face detection to open the url in the browser so I used the library webbrowser to open the respective URL's.

Program:

Main Module

```
from tkinter import *
import reco
import train
import detect
import othe

x = []

def gclass():
    s = 'opening Google Classroom'
    statu(s)
    t = []
    f = open('name.txt', 'r')
    for i in f:
        t.append(i)
    statu(othe.gc(t))


def gmai():
    s = 'opening Gmail'
    statu(s)
    t = []
    f = open('name.txt', 'r')
    for i in f:
        t.append(i)
    statu(othe.gmail(t))


def insta():
    s = 'opening instagram'
    statu(s)
    t = []
    f = open('name.txt', 'r')
    for i in f:
        t.append(i)
    statu(othe.instagram(t))


def whats():
    s = 'opening WhatsApp'
    statu(s)
    t = []
    f = open('name.txt', 'r')
    for i in f:
        t.append(i)
    statu(othe.whatsapp(t))


def close():
    window.destroy()


def came():
    s = 'Camera Opened'
    statu(s)
    othe.camera()
```

```
def statu(s):
    stat = Label(window, text=str(s), width=30, relief=SUNKEN)
    stat.place(relx=0.55, rely=0.42, anchor=CENTER)


def det():
    id_name = nam.get('1.0', 'end')
    f = open('name.txt', 'r')
    for i in f:
        x.append(i)
    f.close()
    if (id_name in x or id_name == 'Enter Your Name Here'):
        s = 'Name already Recorded'
        statu(s)
        pass
    else:
        x.append(id_name)
        f = open('name.txt', 'a')
        f.write(id_name + '\n')
        f.close()
        s = 'Detecting face Please look at the camera'
        statu(s)
        reco.recognise(x.index(id_name))
        s = 'Training on process Please wait......'
        statu(s)
        train.training()
        nam.delete('1.0', 'end')
        s = 'Training Done Face Recognition ready'
        statu(s)


def recog():
    t = []
    f = open('name.txt', 'r')
    for i in f:
        t.append(i)
    s = 'Regognizing.....'
    statu(s)
    detect.detec(t)


window = Tk()
window.title("FACE RECOGNITION")
window.geometry("400x300")

i = Label(window, text='Welcome to Face recognition', font="arial")
i.place(relx=0.5, rely=0.05, anchor=CENTER)

det = Button(window, text='Detect', command=det)
det.place(relx=0.35, rely=0.3, anchor=CENTER)

rec = Button(window, text='Open Recognizer', command=recog)
rec.place(relx=0.65, rely=0.3, anchor=CENTER)

nam = Text(window, height='1', width='20')
nam.place(relx=0.6, rely=0.2, anchor=CENTER)
nam.insert('1.0', 'Enter Your Name Here')

na = Label(window, text='Name: ')
```

```
na.place(relx=0.34, rely=0.2, anchor=CENTER)

st = Label(window, text='STATUS : ', relief=RAISED)
st.place(relx=0.2, rely=0.42, anchor=CENTER)

came = Button(window, text='Camera', width=10, command=came)
came.place(relx=0.8, rely=0.53, anchor=CENTER)

clos = Button(window, text='CLOSE', width=10, command=close)
clos.place(relx=0.8, rely=0.63, anchor=CENTER)

inst = Button(window, text='Instagram', width=10, command=insta)
inst.place(relx=0.10, rely=0.91, anchor=CENTER)

wat = Button(window, text='WhatsApp', width=10, command=whats)
wat.place(relx=0.32, rely=0.91, anchor=CENTER)

gma = Button(window, text='Gmail', width=10, command=gmai)
gma.place(relx=0.54, rely=0.91, anchor=CENTER)

gclas = Button(window, text='Ggl Class', width=10, command=gclass)
gclas.place(relx=0.76, rely=0.91, anchor=CENTER)

y = Label(window, text='To detect faces enter the name and click detect',
fg='white', bg='black')
y.place(relx=0, rely=0.5, anchor=NW)

y = Label(window, text='To start the Recognition click open recognizer',
fg='white', bg='black')
y.place(relx=0, rely=0.59, anchor=NW)

c = Label(window, text='Press camera to open the video camera', fg='white',
bg='black')
c.place(relx=0, rely=0.68, anchor=NW)

z = Label(window, text='Press "ESC" to close any video', fg='white',
bg='black')
z.place(relx=0, rely=0.77, anchor=NW)

s = ''
statu(s)
window.mainloop()
```

SUB-MODULES:

reco:

```
import cv2

def recognise(Id):
    cam = cv2.VideoCapture(0)
    face_detector =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    face_id = Id
    print("\nInitializing face capture. Look at the camera and wait ...")

    count = 0
    while True:
        ret, img = cam.read()
        img = cv2.flip(img, 1)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_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)
            count += 1

            cv2.imwrite("data/User." + str(face_id) + '.' +
                        str(count) + ".jpg", gray[y:y + h, x:x + w])
            cv2.imshow('Detecting', img)
        print(count,'image saved')
        k = cv2.waitKey(50)
        if k == 27:
            break
        elif count >= 100:
            break

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

train:

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


def training():
    path = 'data'
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

    def getImagesAndLabels(path):
        imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
        print('training')
        faceSamples = []
        ids = []
        for imagePath in imagePaths:
            PIL_img = Image.open(imagePath).convert('L')  # grayscale
            img_numpy = np.array(PIL_img, 'uint8')
            id = int(os.path.split(imagePath)[-1].split(".")[1])
```

```
            faces = detector.detectMultiScale(img_numpy)
            for (x, y, w, h) in faces:
                faceSamples.append(img_numpy[y:y + h, x:x + w])
                ids.append(id)
        return faceSamples, ids

    print("Training faces. It will take some time. Wait ...")
    faces, ids = getImagesAndLabels(path)
    recognizer.train(faces, np.array(ids))

    recognizer.write('trainer/trainer.yml')

    print("{0} faces trained.".format(len(np.unique(ids))))
```

detect:

```
import cv2

def detec(x):
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('trainer/trainer.yml')
    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath)
    font = cv2.FONT_HERSHEY_SIMPLEX

    id = 0

    names = x

    cam = cv2.VideoCapture(0)

    minW = 0.1 * cam.get(3)
    minH = 0.1 * cam.get(4)
    while True:
        ret, img = cam.read()
        img = cv2.flip(img, 1)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor=1.3,
            minNeighbors=10,
            minSize=(int(minW), int(minH)),
        )
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 100, 150), 2)
            id, confidence = recognizer.predict(gray[y:y + h, x:x + w])

            if (confidence > 20):
                id = names[id]
                confidence = "  {0}%".format(round(100 - confidence))
            else:
                id = "unknown"
                confidence = "  {0}%".format(round(100 - confidence))

            cv2.putText(
                img,
                str(id),
                (x + 5, y - 5),
```

```
                font,
                1,
                (255, 255, 255),
                2
            )
            cv2.putText(
                img,
                str(confidence),
                (x + 5, y + h - 5),
                font,
                1,
                (255, 255, 0),
                1
            )

        cv2.imshow('Recognizing', img)
        k = cv2.waitKey(10)
        if k == 27:
            break

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

othe:

```
import cv2
import webbrowser

def camera():
    face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    cap = cv2.VideoCapture(0)
    while True:
        ret, img = cap.read()
        img = cv2.flip(img, 1)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.5, 3)
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 255), 2)
        cv2.imshow('Face capture', img)
        k = cv2.waitKey(5)
        if k == 27:
            break
    cap.release()
    cv2.destroyAllWindows()


def instagram(x):
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('trainer/trainer.yml')
    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath)
    font = cv2.FONT_HERSHEY_SIMPLEX
    key = 0
    names = x
    count = 0
    cam = cv2.VideoCapture(0)

    minW = 0.1 * cam.get(3)
```

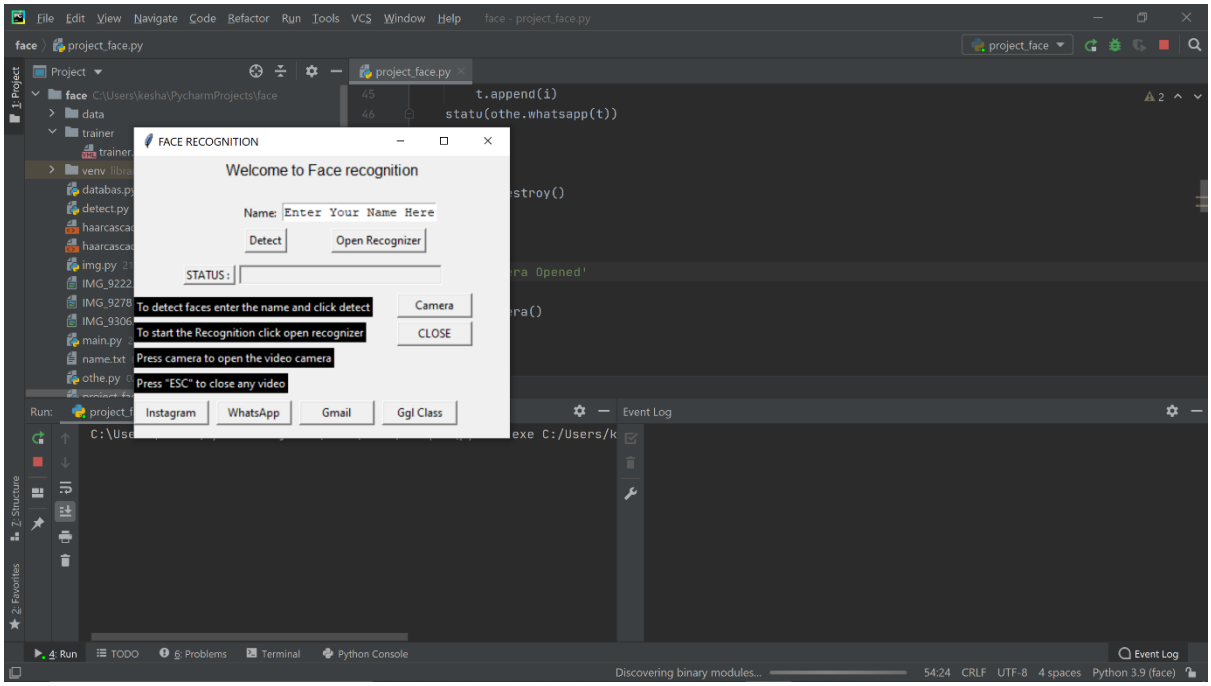
```
    minH = 0.1 * cam.get(4)
    while True:
        ret, img = cam.read()
        img = cv2.flip(img, 1)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor=1.3,
            minNeighbors=10,
            minSize=(int(minW), int(minH)),
        )
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 100, 150), 2)
            id, confidence = recognizer.predict(gray[y:y + h, x:x + w])

            if id == 0:
                key = 1
            if (confidence > 20):
                id = names[id]

            else:
                id = "unknown"

            cv2.putText(
                img,
                str(id),
                (x + 5, y - 5),
                font,
                1,
                (255, 255, 255),
                2
            )

        count += 1
        cv2.imshow('Recognizing', img)
        if count == 10:
            break

    cam.release()
    cv2.destroyAllWindows()
    if key == 1:
        url = 'https://instagram.com/'
        chrome_path = 'C:/Program Files
(x86)/Google/Chrome/Application/chrome.exe %s'
        webbrowser.get(chrome_path).open(url)
        return 'Instagram opened'
    else:
        return 'Face not found'


def whatsapp(x):
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('trainer/trainer.yml')
    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath)
    font = cv2.FONT_HERSHEY_SIMPLEX
    key = 0
    names = x
    count = 0
    cam = cv2.VideoCapture(0)
```

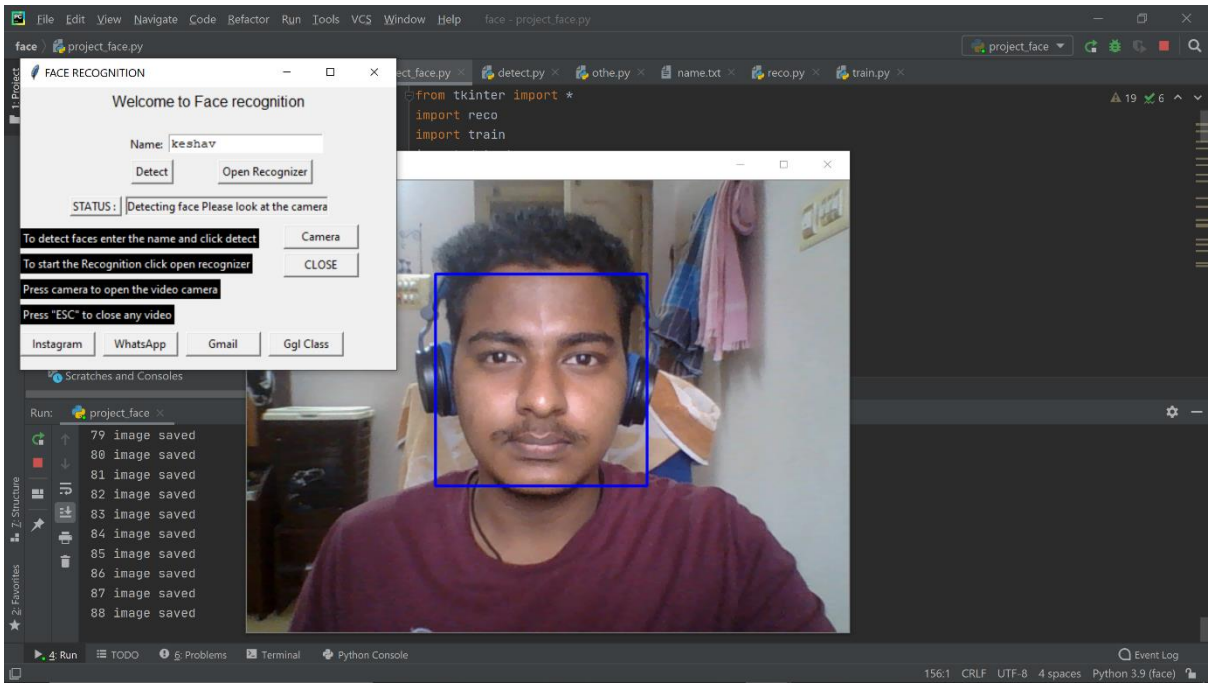
```
    minW = 0.1 * cam.get(3)
    minH = 0.1 * cam.get(4)
    while True:
        ret, img = cam.read()
        img = cv2.flip(img, 1)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor=1.3,
            minNeighbors=10,
            minSize=(int(minW), int(minH)),
        )
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 100, 150), 2)
            id, confidence = recognizer.predict(gray[y:y + h, x:x + w])

            if id == 0:
                key = 1
            if (confidence > 20):
                id = names[id]

            else:
                id = "unknown"

            cv2.putText(
                img,
                str(id),
                (x + 5, y - 5),
                font,
                1,
                (255, 255, 255),
                2
            )

        count += 1
        cv2.imshow('Recognizing', img)
        if count == 10:
            break

    cam.release()
    cv2.destroyAllWindows()
    if key == 1:
        url = 'https://web.whatsapp.com/'
        chrome_path = 'C:/Program Files
(x86)/Google/Chrome/Application/chrome.exe %s'
        webbrowser.get(chrome_path).open(url)
        return 'WhatsApp opened'
    else:
        return 'Face not found'

def gmail(x):
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('trainer/trainer.yml')
    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath)
    font = cv2.FONT_HERSHEY_SIMPLEX
    key = 0
    names = x
    count = 0
```

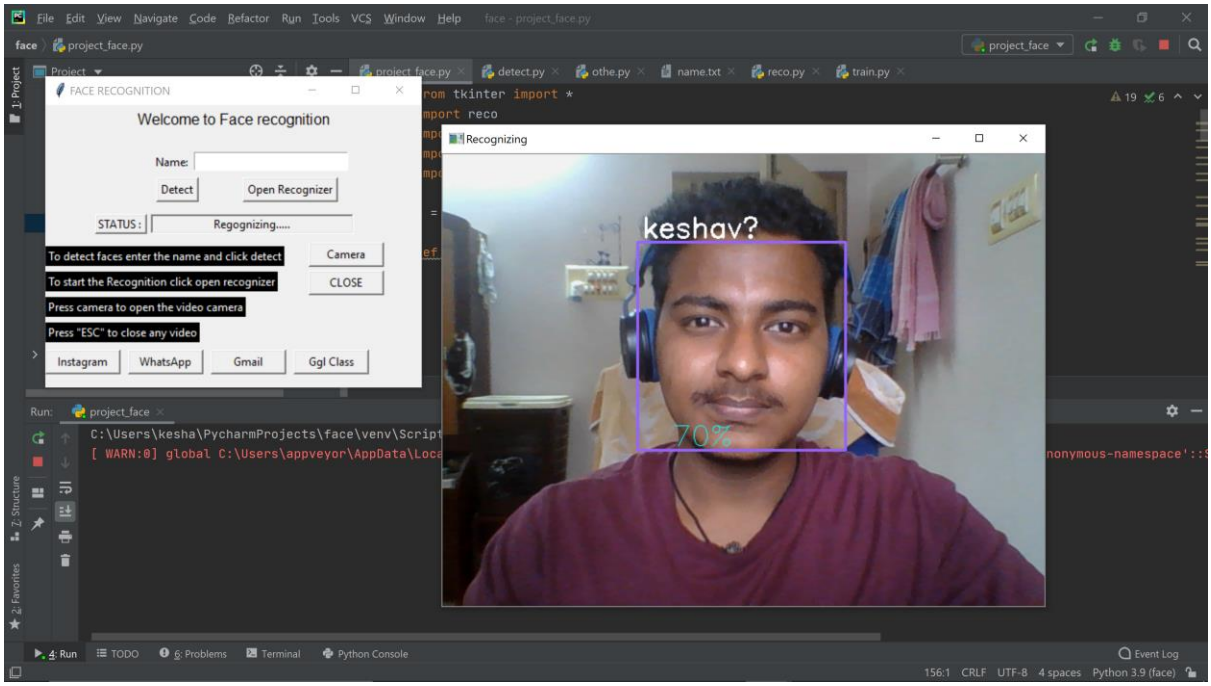
```
    cam = cv2.VideoCapture(0)

    minW = 0.1 * cam.get(3)
    minH = 0.1 * cam.get(4)
    while True:
        ret, img = cam.read()
        img = cv2.flip(img, 1)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor=1.3,
            minNeighbors=10,
            minSize=(int(minW), int(minH)),
        )
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 100, 150), 2)
            id, confidence = recognizer.predict(gray[y:y + h, x:x + w])

            if id == 0:
                key = 1
            if (confidence > 20):
                id = names[id]

            else:
                id = "unknown"

            cv2.putText(
                img,
                str(id),
                (x + 5, y - 5),
                font,
                1,
                (255, 255, 255),
                2
            )

        count += 1
        cv2.imshow('Recognizing', img)
        if count == 10:
            break

    cam.release()
    cv2.destroyAllWindows()
    if key == 1:
        url = 'https://mail.google.com/mail/u/0/#inbox'
        chrome_path = 'C:/Program Files
(x86)/Google/Chrome/Application/chrome.exe %s'
        webbrowser.get(chrome_path).open(url)
        return 'Gmail opened'
    else:
        return 'Face not found'

def gc(x):
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read('trainer/trainer.yml')
    cascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(cascadePath)
    font = cv2.FONT_HERSHEY_SIMPLEX
    key = 0
    names = x
```

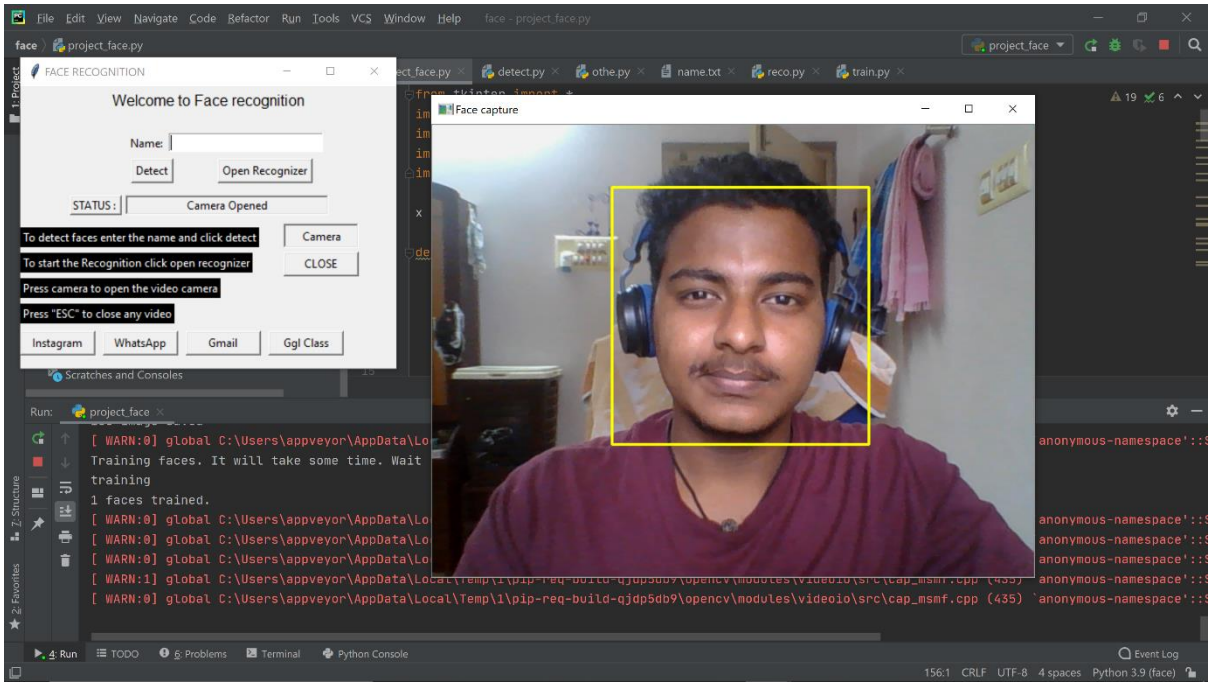
```
    count = 0
    cam = cv2.VideoCapture(0)

    minW = 0.1 * cam.get(3)
    minH = 0.1 * cam.get(4)
    while True:
        ret, img = cam.read()
        img = cv2.flip(img, 1)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        faces = faceCascade.detectMultiScale(
            gray,
            scaleFactor=1.3,
            minNeighbors=10,
            minSize=(int(minW), int(minH)),
        )
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 100, 150), 2)
            id, confidence = recognizer.predict(gray[y:y + h, x:x + w])

            if id == 0:
                key = 1
            if (confidence > 20):
                id = names[id]

            else:
                id = "unknown"

            cv2.putText(
                img,
                str(id),
                (x + 5, y - 5),
                font,
                1,
                (255, 255, 255),
                2
            )

        count += 1
        cv2.imshow('Recognizing', img)
        if count == 10:
            break

    cam.release()
    cv2.destroyAllWindows()
    if key == 1:
        url = 'https://classroom.google.com/u/2/h'
        chrome_path = 'C:/Program Files
(x86)/Google/Chrome/Application/chrome.exe %s'
        webbrowser.get(chrome_path).open(url)
        return 'Google Classroom opened'
    else:
        return 'Face not found'
```

Output:


FACE RECOGNITION
Welcome to Face recognition
Name:
Enter Your Name Here
Detect
Open Recognizer
STATUS :
To detect faces enter the name and click detect
Camera
To start the Recognition click open recognizer
CLOSE
Press camera to open the video camera
Press "ESC" to close any video
Instagram
WhatsApp
Gmail
Ggl Class


FACE RECOGNITION
Welcome to Face recognition
Name:
keshav
Detect
Open Recognizer
STATUS :
Detecting face Please look at the camera
To detect faces enter the name and click detect
Camera
To start the Recognition click open recognizer
CLOSE
Press camera to open the video camera
Press "ESC" to close any video
Instagram
WhatsApp
Gmail
Ggl Class
79 image saved
80 image saved
81 image saved
82 image saved
83 image saved
84 image saved
85 image saved
86 image saved
87 image saved
88 image saved


FACE RECOGNITION
Welcome to Face recognition
Name:
Detect
Open Recognizer
STATUS :
Regognizing.....
To detect faces enter the name and click detect
Camera
To start the Recognition click open recognizer
CLOSE
Press camera to open the video camera
Press "ESC" to close any video
Instagram
WhatsApp
Gmail
Ggl Class
Recognizing
keshav?
70%


FACE RECOGNITION
Welcome to Face recognition
Name:
Detect
Open Recognizer
STATUS :
Camera Opened
To detect faces enter the name and click detect
Camera
To start the Recognition click open recognizer
CLOSE
Press camera to open the video camera
Press "ESC" to close any video
Instagram
WhatsApp
Gmail
Ggl Class
Face capture
Training faces. It will take some time. Wait
training
1 faces trained.