

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Khoa CNTT1



BÀI BÁO CÁO BÀI TẬP LỚN PYTHON

NHÓM BT 10 – NHÓM LỚP 7

Đề tài: Face_Recognition

Thành viên :

1. Đinh Hữu Nam – B20DCCN026
2. Phạm Quốc Cường – B20DCCN107
3. Vũ Duy Hưng – B20DCCN345

Giảng viên hướng dẫn:

Trần Quý Nam

Nhóm môn học: Nhóm 7

Nhóm bài tập lớn: Nhóm 8

Hà Nội , Ngày 01/11/2022

Mục Lục :

I. Danh sách thành viên nhóm và nhiệm vụ

I. Giới thiệu chủ đề

II. Lý do chọn chủ đề và ý nghĩa thực tế

1. Đặt vấn đề, nêu lý do
2. Ý nghĩa thực tế

II. Nội dung chủ đề

1. Phân tích chung , cụ thể
2. Mã nguồn , hình ảnh chạy thử

IV. Kết luận

1. Kết quả của chủ đề
2. Khuyến nghị và cải tiến trong tương lai

V. Tài Liệu tham khảo

I. Danh sách thành viên trong nhóm và nhiệm vụ:

1. Đinh Hữu Nam: (70%)

- Xây dựng ý tưởng, chọn ý tưởng.
- Tìm hiểu và trực tiếp code, chỉnh sửa code.
- Chỉnh sửa báo cáo.
- Thuyết trình.

2. Phạm Quốc Cường: (20%)

- Góp ý xây dựng ý tưởng.
- Viết báo cáo.

3. Vũ Duy Hưng: (10%)

- Góp ý xây dựng ý tưởng.
- Làm powpoint.

II. GIỚI THIỆU ĐỀ TÀI

+ Python là một ngôn ngữ được sử dụng rộng rãi, từ làm game , data science hay làm desktop applications ... và một số yêu cầu cơ bản của người code như làm web, tính toán và làm những dự án nhỏ.

Với sự mạnh mẽ của các library và package trong python mọi thứ dường như trở lên dễ dàng hơn

+ Trong môn học lập trình với python , sinh viên đã được tiếp cận gần hơn với python , vận dụng khả năng sử dụng ngôn ngữ python vào trong bài toán thực tế , thông qua việc làm bài tập lớn nhóm 10 chúng em đã có những thử nghiệm khai thác các thư viện , framework (OpenCV, Numpy, Pandas, ...) . Nó sẽ là bài học quý báu cho sinh viên chúng em được tiếp cận sớm với môi trường làm việc.

+ Chủ đề làm bài tập nhóm : Nhận dạng khuôn mặt (face_recognition) ứng dụng vào việc giám sát việc ra vào của sinh viên Học Viện Công Nghệ Bưu Chính Viễn Thông

III. LÝ DO CHỌN VÀ Ý NGHĨA THỰC TẾ

1. Đặt vấn đề , lý do chọn chủ đề

+ Hiện nay , tại Học Viện chúng ta hàng ngày đang có sự hoạt động của một câu lạc bộ có tên : “ Đội Cờ Đỏ” .

- Nhiệm vụ của đội :

- Giám sát sinh viên học viện (kiểm tra thẻ sinh viên) , tác phong của sinh viên học viện.
- Ngăn chặn những đối tượng không phải là người trong trường đến trường với mục đích khác.

- Vai trò của đội :

- Đảm bảo an toàn trong học viện .
- Giữ tác phong tốt của sinh viên học viện (phải đeo thẻ sinh viên khi đến trường).

- Hạn chế :

- Cần quá nhiều nhân lực và tốn thời gian công sức của các bạn sinh viên trong đội cờ đỏ.
- Không kiểm tra kĩ khuôn mặt (đôi khi chỉ cần nhìn sinh viên có cầm thẻ sinh viên) là cho vào.
- Có lúc sinh viên không mang thẻ, lúc này sẽ không phân biệt được bạn ấy có đang là sinh viên học viện hay không.

+ Chủ đề của nhóm đang xây dựng cũng sẽ theo hướng dần thay thế đội cờ đỏ , phần mềm này sẽ giúp việc giám sát trong học viện trở lên dễ dàng hơn , đó sẽ là lý do chính để nhóm em thử nghiệm phần mềm.

2. Ý nghĩa thực tế :

Như đã nói ở phần đặt vấn đề bài tập lớn của bọn em sẽ giải quyết những khó khăn của đội cờ đỏ , và tiếp tục giữ vững mục tiêu và nhiệm vụ của đội cờ đỏ . Việc ứng dụng bài tập lớn này mang nhiều ý nghĩa:

+ Sinh viên không cần phải mang thẻ sinh viên mà vẫn xác định được đó là sinh viên của học viện

+ Đội cờ đỏ sẽ không phát mất quá nhiều nhân lực (gây ảnh hưởng việc học tập của các bạn trong đội cờ đỏ)

+ Việc kiểm tra sẽ diễn ra 1 cách tự động và sẽ đảm bảo an toàn cho học viện

+ Hỗ trợ trong công cuộc chuyển đổi số tại Học viện

III. NỘI DUNG CHỦ ĐỀ

1.Phân tích chung và cụ thể

a. Phân tích chung: Bài tập lớn gồm 3 phần chính và các thư viện cần cài:

- Tổng quát công việc cần thực hiện :

Phần 1: Lấy dữ liệu hình ảnh từ webcam máy tính và lưu vào folder có tên “dataset” trong máy tính. Lưu dữ liệu người dùng gồm: ID,Tên,Tuổi,Mã SV,Lớp vào cơ sở dữ liệu database.

Phần 2: Train các ảnh đã được lấy từ webcam máy tính và được lưu trong folder “dataset” trong máy tính.

Phần 3: So sánh đối chiếu ảnh khuôn mặt trên webcam hiện tại với kho ảnh vừa được train để nhận diện khuôn mặt.

Phần 4: Thêm giao diện cho ứng dụng nhận diện khuôn mặt

-Các thư viện sử dụng chính và các module sử dụng trong bài tập lớn

```
import tkinter as tk
from tkinter import ttk
import cv2
import numpy as np
import sqlite3
import os
from PIL import Image
from gtts import gTTS
from playsound import playsound
import pyttsx3
```

+ Thư viện OpenCV2. Đây là một thư viện mã nguồn mở dùng để xử lý ảnh và thị giác máy tính. Chức năng cơ bản của thư viện này liên quan đến các hoạt động xử lý ảnh, nhận dạng khuôn mặt, học máy và rất nhiều các chức năng khác...

+ Thư viện Numpy là một thư viện lõi phục vụ cho khoa học máy tính của Python, hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó.

+ Module os trong Python cung cấp các chức năng được sử dụng để tương tác với hệ điều hành và cũng có được thông tin liên quan về nó. OS đi theo các Module tiện ích tiêu chuẩn của Python. Module này cung cấp một cách linh

động sử dụng chức năng phụ thuộc vào hệ điều hành. Module os trong python cho phép chúng ta làm việc với các tập tin và thư mục.

+ Module Sqlite3 để giúp tạo kết nối đến database SQLite, thao tác với các database.

+Thư viện Playsound trong python để xuất ra file âm thanh và xử lý file âm thanh.

+ Thư viện pyttsx3 dùng để hỗ trợ trợ lý ảo tiếng việt.

+ from PIL import Image dùng để hỗ trợ xử lý ảnh

+Thư viện Tkinter. Tkinter là thư viện GUI tiêu chuẩn cho Python. Khi kết hợp với Tkinter, Python sẽ được cung cấp các công cụ một cách nhanh chóng và dễ dàng để tạo các ứng dụng GUI. Tkinter cung cấp giao diện hướng đối tượng mạnh mẽ đến các bộ công cụ Tk GUI.

b. Phân tích cụ thể:

-Phần 1: Lấy dữ liệu hình ảnh từ webcam máy tính và lưu vào folder có tên “dataset” trong máy tính. Lưu dữ liệu người dùng gồm: ID,Tên,Tuổi,Mã SV,Lớp vào cơ sở dữ liệu database.

+ Bước 1:

Cho phép người dùng nhập dữ liệu vào từ bàn phím. Người dùng sẽ nhập ID, tên, tuổi, lớp. Mỗi một lần nhập sẽ có giọng trợ lý ảo hướng dẫn để người dùng dễ sử dụng hơn.

Sử dụng câu lệnh nhập input() cơ bản trong python.

```

54 # Người dùng nhập dữ liệu từ bàn phím
55
56 engine.say("Mời bạn nhập ID")
57 engine.runAndWait()
58 id = input("Enter your ID: ")
59 engine.say("Mời bạn nhập tên:")
60 engine.runAndWait()
61 name = input("Enter your Name: ")
62 engine.say("Mời bạn nhập tuổi:")
63 engine.runAndWait()
64 age = input("Enter your Age: ")
65 engine.say("Mời bạn nhập lớp:")
66 engine.runAndWait()
67 lop = input("Enter your lop: ")

```

+ Bước 2 : Khởi tạo camera để nhận diện khuôn mặt .Để camera chụp được hình khuôn mặt để train cần vẽ 1 hình vuông lên trên khuôn mặt trong webcam.

```

77 # Load thư viện mặc định nhận diện khuôn mặt của opencv
78 face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
79 # hàm trong opencv để truy cập vào webcam máy tính
80 cap = cv2.VideoCapture(0)
81 # nhận diện khuôn mặt từ webcam và lưu vào database
82 sampleNum=0
83 while (True):
84     # biến ret nếu truy cập thành công sẽ trả về true
85     # biến frame là data dữ liệu lấy được từ webcam
86     ret, frame = cap.read()
87
88     # chuyển frame từ webcam về ảnh sáng: chuyển từ màu RGB sang màu gray
89     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
90
91     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
92
93     for (x, y, w, h) in faces:
94         # hàm rectangle để vẽ hình vuông lên frame
95         # ta có (x,y) là toạ độ điểm
96         # x+w,y+h là toạ độ điểm tính tiến
97         # (0,225,0) là chỉ số màu xanh.
98         # 2 là độ dày hình vuông
99         cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 225, 0),2)

```

-Khởi tạo camera sử dụng hàm cv2.VideoCapture(0) trong thư viện open cv2.

```

# hàm trong opencv để truy cập vào webcam máy tính
cap = cv2.VideoCapture(0)

```

-Bước tiếp theo ta chuyển ảnh từ BGR sang GRAY.Vì màu GRAY là màu chuẩn để máy có thể train được sau này. Để chuyển được màu sang gray ta sử dụng câu lệnh:

```

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

```

-Để vẽ được hình vuông lên frame từ webcam sử dụng hàm cv2.rectangle() được hỗ trợ từ thư viện OpenCV2.

```
for (x, y, w, h) in faces:
    # hàm rectangle để vẽ hình vuông lên frame
    # ta có (x,y) là toạ độ điểm
    # x+w,y+h là toạ độ điểm tịnh tiến
    # (0,225,0) là chỉ số màu xanh.
    # 2 là độ dày hình vuông
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 225,
0),2)
```

+ Bước 3 : Lưu ảnh vừa chụp (mỗi người sẽ có 300 ảnh hoặc có thể nhiều tấm hơn tùy thuộc vào mong muốn người dùng) vào dataSet. Biến sampleNum để kiểm soát số ảnh chụp từ webcam. Mỗi ảnh sẽ có định dạng ví dụ như sau: User.1.1, User.1.2, User.1.3,...

```
100
101     # tạo 1 folders ảnh để Lưu Lại ảnh đã cắt được từ hình vuông
102     # kiểm tra xem đã tồn tại đường dẫn dataSet chưa
103     # nếu chưa tạo
104     if not os.path.exists('dataSet'):
105         os.makedirs('dataSet')
106
107     # số ảnh lấy tăng dần
108     sampleNum +=1
109     # Lưu ảnh đã chụp khuôn mặt trong file dataSet dữ liệu dưới dạng User...
110     # ảnh cắt được từ hình vuông là ảnh xám gray có toạ độ [y+h,x:w]
111     cv2.imwrite('dataSet/User.'+str(id)+'.'+str(sampleNum)+'.jpg', gray[y: y+h,x: x+w])
112
```

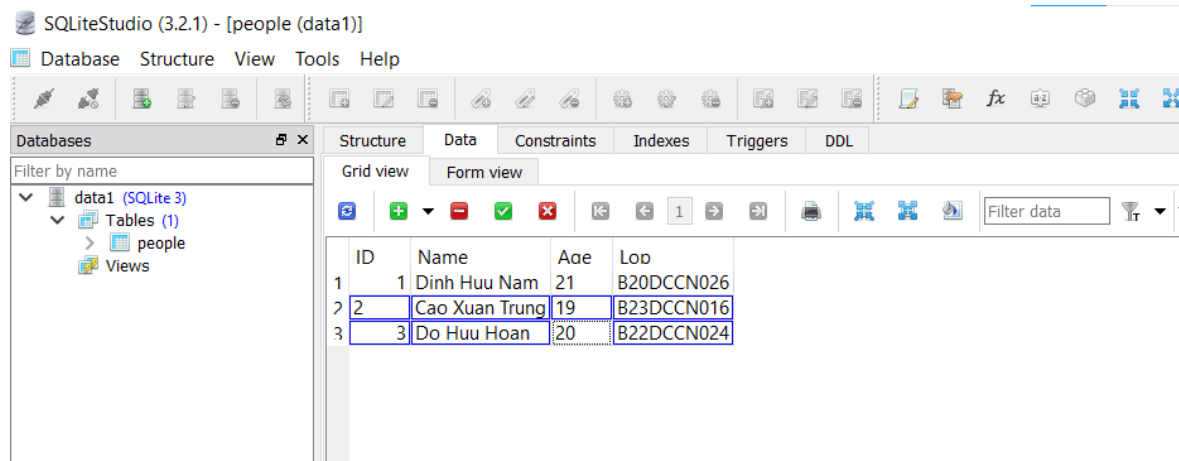
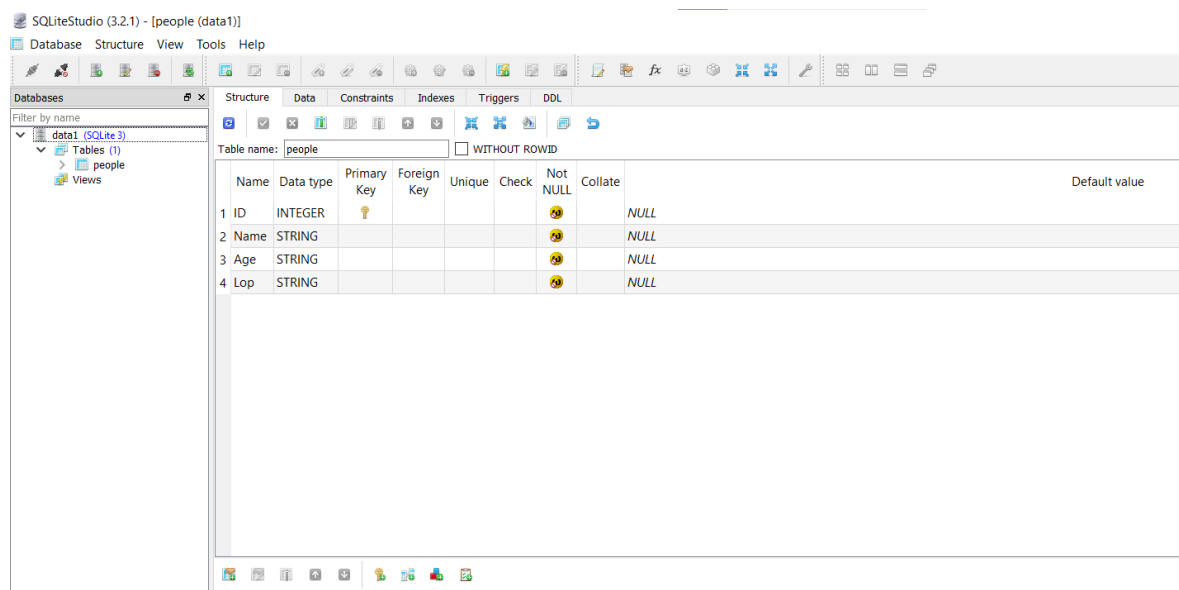
+ Bước 4: Lưu dữ liệu người dùng vào database sử dụng SQLITE3. Database có tên đường dẫn “data1.db”. Database của mỗi người gồm (id,name,age,lop).


```

18 # viết 1 hàm để truy cập đến database
19 # gồm có 4 trường dữ liệu: id,name,age,lop
20 def insertOrUpdate(id,name,age,lop):
21     # truy cập đến đường dẫn trong sqlite3
22     conn = sqlite3.connect('C:\\Users\\Admin\\Project\\data1.db')
23     # viết 1 câu lệnh kiểm tra xem id đã tồn tại chưa
24     query = "Select * from people where ID=" + str(id)
25     # truy cập vào các query
26     cursor = conn.execute(query)
27     # biến isRecordExist để kiểm tra xem id đã tồn tại chưa
28     # chưa tồn tại sẽ gán bằng 0
29     # tồn tại rồi gán bằng 1
30     isRecordExist = 0
31     # duyệt từng hàng trên bản ghi
32     for row in cursor:
33         isRecordExist = 1
34         if (isRecordExist == 0):
35             query = "Insert into People(id,name,age,lop) values('"+str(id)+"','"+str(name)+"','"+str(age)+"','"+str(lop)+"'"
36         else:
37             query = "Update People SET Name = '"+str(name)+"', Age= '"+str(age)+"',Lop= '"+str(lop)+"' where ID= " + str(id)
38     conn.execute(query)
39     conn.commit()
40     conn.close()

```

+Dữ liệu được lưu trong database “data1.db” trong SQLITE3:



-Phần 2: Train các dữ liệu ảnh trong folder DataSet.

+ Bước 1: Khởi tạo camera để bắt đầu train

```
cap = cv2.VideoCapture(0)
```

+ Bước 2: Lấy ID và mảng dữ liệu ảnh của người dùng. Lưu ID và mảng dữ liệu ảnh vào 2 mảng IDs và faces.

```
path = 'dataSet'
# tiếp theo sẽ có 1 hàm để lấy ra ID và list các ảnh
# lấy dữ liệu từ SQL theo ID. Tên hàm: getImageWithId
def getImageWithId(path):

    # tạo ra 1 biến imagePaths để truy cập vào tất cả các đường
    # dẫn của các ảnh
    imagePaths = [os.path.join(path, f) for f in
os.listdir(path)]
    # tạo ra 2 mảng.
    faces=[]
    IDs=[]
    # Lấy tất cả các đường dẫn từ imagePaths
    for imagePath in imagePaths:
        # sử dụng thư viện Image để mở đường dẫn imagePath và
        # chuyển ảnh về đúng định dạng gray
        faceImg = Image.open(imagePath).convert('L')
        faceNp = np.array(faceImg, 'uint8')
        print(faceNp)
        # cần phải lấy được ID để các ảnh thuộc ID nào để
train
        # Cần phải cắt số 1 ra
        Id = int(imagePath.split('.')[1])
        # thêm các dữ liệu ảnh vào mảng faces
        faces.append(faceNp)
        # Lưu Id vào mảng IDs
        IDs.append(Id)
        # để hiển thị các hình ảnh train ra màn hình ta dùng
        # hàm imshow
        cv2.imshow('training',faceNp)
        cv2.waitKey(10)
```

Ảnh minh họa:

```
TrainData.py > ...
20 # tạo biến recognizer
21 recognizer = cv2.face_LBPHFaceRecognizer.create()
22 # tạo 1 biến path để lưu đường dẫn dataSet
23 path = 'dataSet'
24 # tiếp theo sẽ có 1 hàm để lấy ra ID và List các ảnh
25 # lấy dữ liệu từ SQL theo ID
26 def getImageWithId(path):
27     # tạo ra 1 biến imagePath để truy cập vào tất cả các đường dẫn của các ảnh
28     imagePath = [os.path.join(path, f) for f in os.listdir(path)]
29     # tạo ra 2 mảng.
30     faces=[]
31     IDs=[]
32     # Lấy tất cả các đường dẫn từ imagePath
33     for imagePath in imagePath:
34         # sử dụng thư viện Image để mở đường dẫn imagePath
35         faceImg = Image.open(imagePath).convert('L')
36         faceNp = np.array(faceImg, 'uint8')
37         print(faceNp)
38         # cần phải lấy được ID để các ảnh thuộc ID nào để train
39         # Cần phải cắt số 1 ra
40         Id = int(imagePath.split('.')[1])
41         # thêm các dữ liệu ảnh vào mảng faces
42         faces.append(faceNp)
43         # Lưu Id vào mảng IDs
44         IDs.append(Id)
45         # để hiển thị các hình ảnh train ra màn hình ta dùng hàm imshow
46         cv2.imshow('training',faceNp)
47         cv2.waitKey(10)
48     return faces,IDs
49
```

+ Bước 3 :Bắt đầu train ảnh. Sử dụng hàm train trong thư viện open cv2 và truyền vào 2 biến faces và Ids (2 biến được lấy ra từ function getImageWithId ở bước 2).

```
TrainData.py > ...
50 |
51 recognizer = cv2.face_LBPHFaceRecognizer.create()
52 faces ,Ids=getImageWithId(path)
53 # Ta sẽ có 1 hàm train. Ta truyền vào hai tham số faces và Ids
54 recognizer.train(faces ,np.array(Ids))
55
56 if not os.path.exists('recognizer'):
57     os.makedirs('recognizer')
58 # Sau khi train xong sẽ tạo ra 1 file đuôi yml để lưu dữ liệu sau khi đã train
59 recognizer.save('recognizer/trainingData.yml')
60
61 cv2.destroyAllWindows()
```

Sau khi train xong sẽ trả ra 1 file. Tạo ra 1 folder recognizer và lưu lại file vừa được train. File này có tên là “trainingData.yml”.

- Phần 3 : Hoàn thành nhận diện. So sánh khuôn mặt người dùng đưa vào webcam và dữ liệu khuôn mặt được chụp trong dataset.

+ Bước 1: Lấy Profile mỗi người theo ID từ database

```

24 # get profile by id from database
25 def getProfile(id):
26
27     conn = sqlite3.connect("data1.db")
28
29     query = "SELECT * FROM people WHERE ID=" + str(id)
30     cursor = conn.execute(query)
31     profile = None
32     for row in cursor:
33         profile = row
34
35     conn.close()
36     return profile
37

```

+ Bước 2 : Vẽ hình vuông trên khuôn mặt của webcam (trình tự giống phần 2)

```

face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades
'haarcascade_frontalface_default.xml')
recognizer = cv2.face_LBPHFaceRecognizer.create()

# Khởi tạo camera của máy tính
cap = cv2.VideoCapture(0)
fontFace = cv2.FONT_HERSHEY_SIMPLEX
while (True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # Chuyển từ ảnh RGB sang màu xám
    faces = face_cascade.detectMultiScale(gray)
    # Cần vẽ được hình vuông bao quanh khuôn mặt
    for (x, y, w, h) in faces:
        # hàm rectangle để vẽ hình vuông
        # Điểm (x,y) là tọa độ điểm ban đầu
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0,
225,0), 2)

```

Ảnh minh họa

```

37
38 # training hình ảnh nhanh diện và thu viện nhận diện khuôn mặt
39 face_cascade = cv2.CascadeClassifier(
40     cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
41 recognizer = cv2.face_LBPHFaceRecognizer.create()
42 recognizer.read('/Users/Admin/Project/recognizer/trainingData.yml')
43 cap = cv2.VideoCapture(0)
44 fontFace = cv2.FONT_HERSHEY_SIMPLEX
45 while (True):
46     ret, frame = cap.read()
47     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
48     faces = face_cascade.detectMultiScale(gray)
49     for (x, y, w, h) in faces:
50         cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
51

```

+Bước 3: So sánh khuôn mặt trong webcam và ảnh khuôn mặt trong file đã được train.

Để so sánh được cần 2 biến: biến id là để trả ra id và biến confidence là biến trả về độ tin cậy. Ảnh được so sánh dựa vào hàm predict hỗ trợ trong thư viện openCV2.

Nếu độ tự tin càng lớn thì 2 hình ảnh so sánh càng khác nhau và ngược lại. Nghe thì có vẻ hơi ngược nhưng trên docs của OpenCV nó quy ước như vậy. Trong trường hợp conf < 40 có nghĩa nếu mức độ giống nhau từ 60 ảnh trở lên trong 100 ảnh thì hiển thị kết quả nhận dạng đúng người và ngược lại thì cho kết quả 2 người khác nhau..

```

49     for (x, y, w, h) in faces:
50         cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)
51
52         roi_gray = gray[y:y+h, x:x+w]
53         # id trong database, confidence trả về độ chính xác của ảnh
54         id, confidence = recognizer.predict(roi_gray)
55
56         if confidence < 40:
57             profile = getProfile(id)
58
59             if (profile != None):
60                 cv2.putText(frame, "Name: "+str(profile[1]), (x+10, y+h+30), fontFace, 1, (0, 255, 0), 2)
61                 cv2.putText(frame, "Age : "+str(profile[2]), (x+10, y+h+60), fontFace, 1, (0, 255, 0), 2)
62                 cv2.putText(frame, "Class : "+str(profile[3]), (x+10, y+h+90), fontFace, 1, (0, 255, 0), 2)
63             else:
64                 cv2.putText(frame, "Unknown", (x+10, y+h+30), fontFace, 1, (0, 0, 255), 2)
65

```

+Bước 4: Hoàn thành nhận dạng :

Khi đã nhận dạng được khuôn mặt thì ảnh sẽ trả ra được tên tuổi và lớp của người đưa khuôn mặt vào webcam. Nếu không nhận dạng được sẽ hiển thị “Unknown”.

Đồng thời trong thư viện pyttsx3 sẽ hỗ trợ giọng nói (giọng nói trợ lý ảo), sẽ đọc ra các thông báo cho người dùng. Nếu nhận diện được ảnh sẽ thông báo “Bạn là sinh viên học viện bưu chính viễn thông”. Nếu không nhận diện được ảnh sẽ thông báo “Chưa nhận dạng được khuôn mặt!!! Bạn chưa được vào lớp!!!”.

```

60 id, confidence = recognizer.predict(roi_gray)
61
62 if confidence < 40:
63     profile = getProfile(id)
64
65     if (profile != None):
66         cv2.putText(frame, "Name: "+str(profile[1]), (x+10, y+h+30), fontFace, 1, (0, 255, 0), 2)
67         cv2.putText(frame, "Age : "+str(profile[2]), (x+10, y+h+60), fontFace, 1, (0, 255, 0), 2)
68         cv2.putText(frame, "Class : "+str(profile[3]), (x+10, y+h+90), fontFace, 1, (0, 255, 0), 2)
69
70         while(i<1):
71             # engine.say("Thông tin sinh viên: ")
72             engine.say("Bạn là sinh viên học viện bưu chính viễn thông ")
73             engine.say("Lớp: "+ profile[3])
74             engine.say("Thông tin sinh viên hợp lệ!!! Mời bạn vào lớp!!!")
75             engine.runAndWait()
76             i=i+1
77             break
78     else:
79         cv2.putText(frame, "Unknown", (x+10, y+h+30), fontFace, 1, (0, 255, 0), 2)
80         while(j<1):
81             engine.say("Chưa nhận dạng được khuôn mặt!!! Bạn chưa được vào lớp!!!")
82             engine.runAndWait()
83             j=j+1
84             break
85

```

-Phần 4: Thêm giao diện cho ứng dụng nhận diện khuôn mặt

Bước 1: Đầu tiên có tác dụng thêm tất cả các chức năng và các module từ thư viện Tkinter.

```

win = tk.Tk() #khởi tạo cửa sổ chính
win.title("He thong nhan dien khuon mat") #hiển thị dòng text
win.geometry('500x300')#quy định kích thước của cửa sổ
win.configure(bg='#263D42') #điều chỉnh các thuộc tính cửa sổ
chính

```

Dòng lệnh win = Tk() là dòng lệnh khởi tạo cửa sổ chính của App. Dòng lệnh thứ hai để hiển thị văn bản dòng text có tên (“Hệ thống nhận diện khuôn mặt”). Dòng lệnh thứ ba có tác dụng quy định kích thước của cửa sổ. Dòng lệnh thứ tư có tác dụng điều chỉnh các thuộc tính của cửa sổ chính, ví dụ như: màu nền (bg), chiều cao (height),...

Bước 2: Quản lý vị trí các thành phần trong giao diện

Hàm Label dùng để hiển thị văn bản text trong giao diện

```

label = ttk.Label(win, text="Hệ Thống Nhận Diện Khuôn
Mặt", background="grey", foreground="white", font=20)
label.grid(column =1, row =0)
label.place(x=100)

label1 =
ttk.Label(win, text="Id:", background="#263D42", foreground="whi
te")

```

```

label1.grid(column =0, row =2)
label1.place(y=80)

label2 =
ttk.Label(win,text="Name:",background="#263D42",foreground="white")
label2.grid(column =0, row =3)
label2.place(y=120)

label3 =
ttk.Label(win,text="Age:",background="#263D42",foreground="white")
label3.grid(column =0, row =4)
label3.place(y=160)

label4 =
ttk.Label(win,text="Lop:",background="#263D42",foreground="white")
label4.grid(column =0, row =5)
label4.place(y=200)

int1 =tk.IntVar()
edit_id=ttk.Entry(win,textvariable=int1, width=50)
edit_id.grid(column =1, row =2)
edit_id.focus()
edit_id.place(x=90,y=80)

str1 =tk.StringVar()
edit_name=ttk.Entry(win,textvariable=str1,width=50)
edit_name.grid(column =1, row =3)
edit_name.focus()
edit_name.place(x=90,y=120)

str2 =tk.StringVar()
edit_age=ttk.Entry(win,textvariable=str2,width=50)
edit_age.grid(column =1, row =4)
edit_age.focus()
edit_age.place(x=90,y=160)

str3 =tk.StringVar()
edit_lop=ttk.Entry(win,textvariable=str3,width=50)

```

```
edit_lop.grid(column =1, row =5)
edit_lop.focus()
edit_lop.place(x=90,y=200)
```

Bước 3: Thêm các Button cho giao diện. Nút bấm có thể hiển thị văn bản hoặc hình ảnh, chúng ta có thể thiết lập sự kiện khi chúng ta click vào nút bấm như tự động gọi hàm chẳng hạn.

```
btlaydulieu= ttk.Button(win, text ="Lấy Dữ Liệu",
command=laydulieu)
btlaydulieu.grid(column =0, row =8)
#btlaydulieu.place()
bttrain= ttk.Button(win, text ="Training", command=train)
bttrain.grid(column =1, row =8)
btnhandien= ttk.Button(win, text ="Nhận Diện",
command=nhandien)
btnhandien.grid(column =2, row =8)
bttrain.place(x=200,y=250)
btnhandien.place(x=350,y=250)
btlaydulieu.place(x=50,y=250)
```

Bước 4: Để chạy được giao diện cần có câu lệnh sau

```
win.mainloop()
```

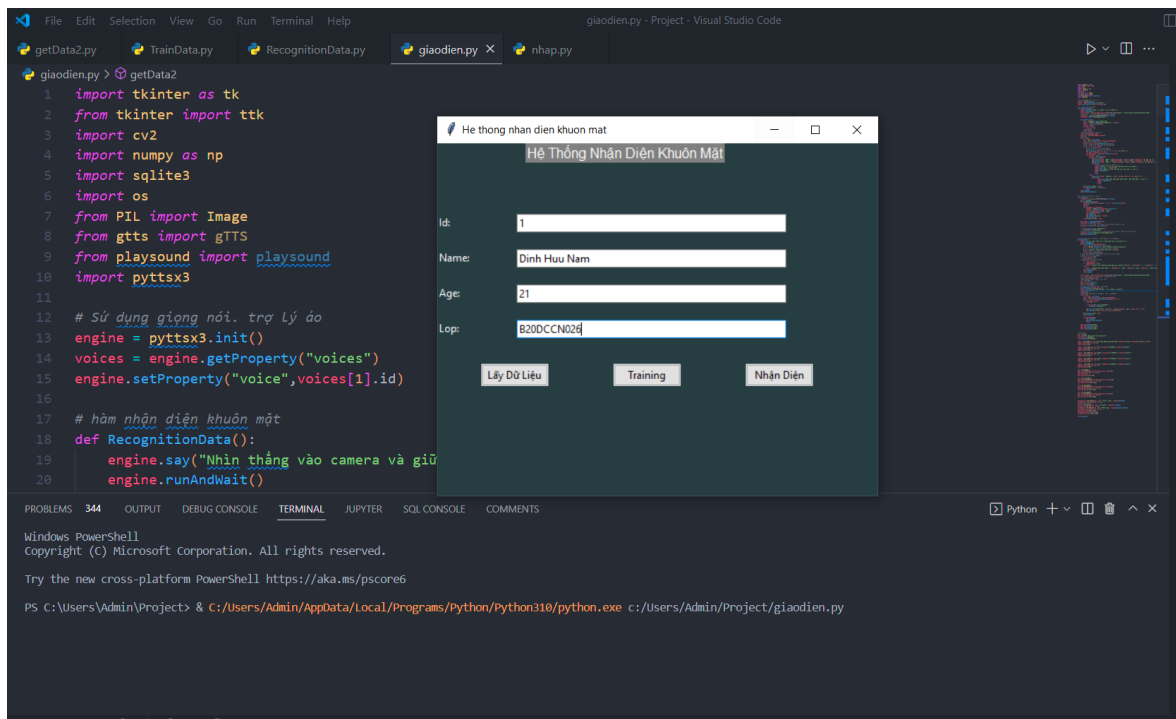
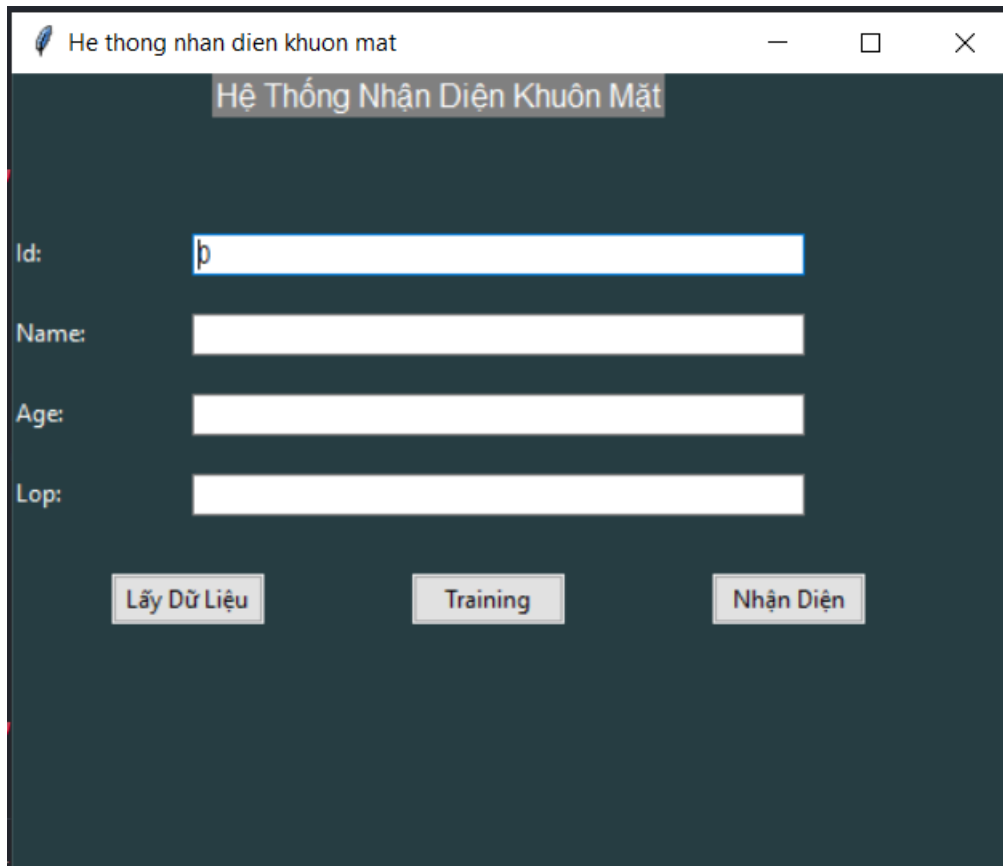
2 . Mã nguồn và hình ảnh kết quả chạy

a. Mã nguồn:

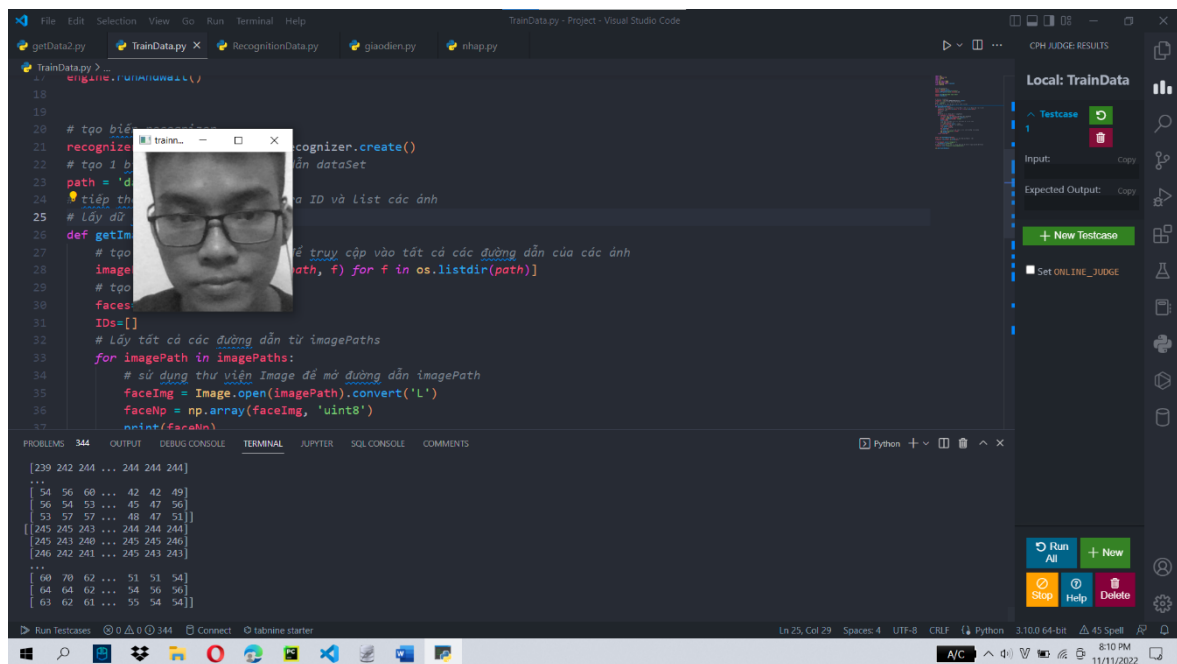
<https://drive.google.com/drive/u/0/folders/1Zr1fdwWvzRvaYNSdA9GLi5TCnskdOIda>

b. Kết quả chạy từng phần:

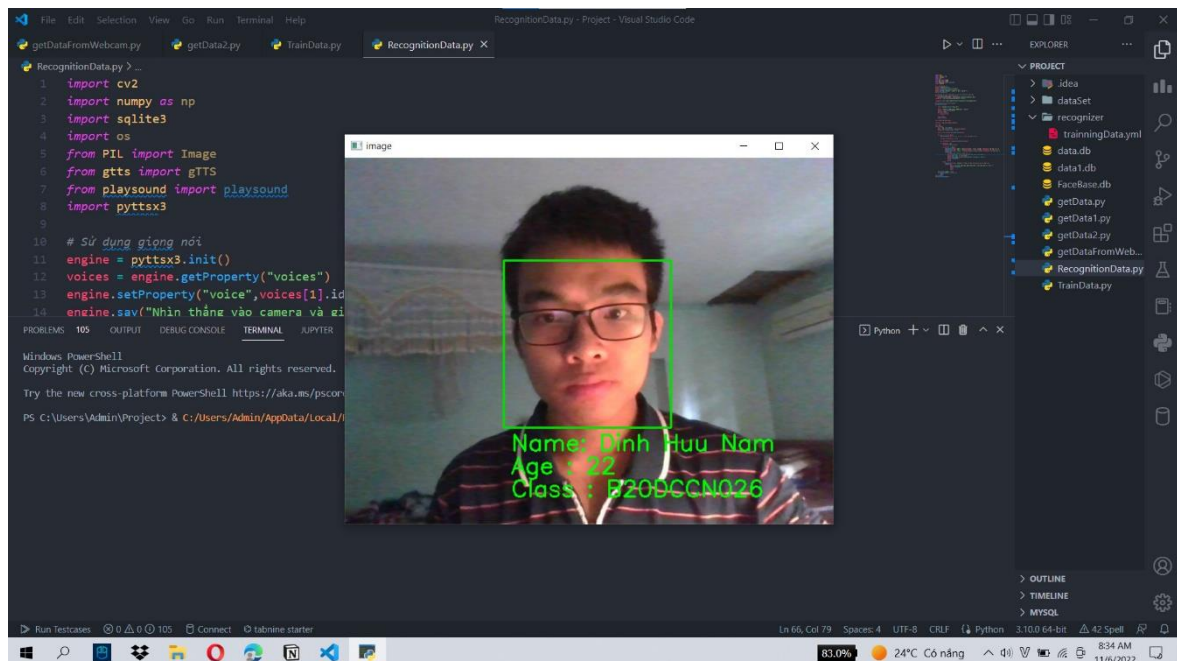
-Giao diện khi khởi chạy chương trình:



-Bắt đầu train dữ liệu ảnh:



-Sau khi chạy chương trình hiển thị kết quả:



Màn hình chạy khi nhận diện khuôn mặt thành công hiển thị được tên, tuổi và lớp của sinh viên. Đồng thời có trợ lý ảo thông báo “nhận diện ra sinh viên”, nếu không nhận diện ra ảnh sẽ thông báo “không nhận dạng được ảnh”.

IV. KẾT LUẬN

1, Kết quả của chủ đề :

Chương trình đã chạy được đúng với ý tưởng mà nhóm em đã tìm hiểu .

Đáp ứng được nhu cầu cần thiết đã đề ra của nhóm :

- + Đã nhận diện được và đối chiếu sinh viên đã có ở trong cơ sở dữ liệu hay không
- + Có bảng database lưu trữ các thông tin của sinh viên
- + Đã hiện lên tên , tuổi , lớp của sinh viên khi nhận dạng sinh viên trong khung hình camera
- + Có hướng dẫn cụ thể trong từng bước bằng việc tích hợp giọng nói .

2, Khuyến nghị và cải tiến trong tương lai

- + Dù đã nhận diện những sinh viên có trong kho dữ liệu nhưng cần có kết quả lưu trữ của những người vào (thời gian vào) , đặc biệt hơn là những người lạ. Việc này rất quan trọng , nó giúp cho quá trình tìm kiếm và nhận dạng khi những người lạ này gây ra sự việc không đáng có
- + Phần mềm không nhận diện được người che mặt (như đeo khẩu trang,...)
- + Với kho dữ liệu lớn cần thời gian để lưu trữ, đôi khi việc lưu trữ tốn dung lượng khi có quá nhiều dữ liệu vào ,và độ chính xác , thời gian của việc nhận diện cũng giảm đi khi phải tìm dữ liệu trong kho dữ liệu lớn.
- + Nhận diện khuôn mặt này có tính 2 mặt : Khi người khác dùng ảnh chụp khuôn mặt một trong những người có trong cơ sở dữ liệu thì có thể ra vào mà không cần sử dụng khuôn mặt của mình

Còn rất nhiều mặt hạn chế về chủ đề này và nhóm em sẽ cố gắng tìm hiểu và đưa ra những giải pháp tốt nhất . Đặc biệt hơn qua việc làm bài tập lớn này , nó sẽ giúp chúng em có cơ hội tìm hiểu nhiều ứng dụng khác của ngôn ngữ lập trình python.

V. TÀI LIỆU THAM KHẢO

1. https://github.com/ageitgey/face_recognition
2. [\(131\) Bài 8.2: Nhận dạng khuôn mặt Part 2 - Nhận dạng qua webcam, video - face detection python - YouTube](#)
3. [\(131\) NHẬN DIỆN KHUÔN MẶT TRÊN PYTHON VỚI OPENCV - YouTube](#)
3. <https://opencv.org/books/>

4. https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Object_Detection_Face_Detection_Haar_Cascade_Classifiers.php

5. [ageitgey/face_recognition: The world's simplest facial recognition api for Python and the command line \(github.com\)](#)

6. Lập trình giao diện với Python: [Hướng dẫn lập trình giao diện GUI bằng Tkinter trong ngôn ngữ Python - VniTeach - Giáo viên 4.0](#)

7. [Lập trình App với Tkinter #1: App đổi số từ hệ thập phân sang nhị phân – Lanus Coder](#)