

# SEMINAR

Hệ thống điểm danh bằng nhận diện khuôn mặt

Nhóm 1

## Tổng quan đồ án

Bài toán được nhiều người quan tâm nhất của lĩnh vực xử lý ảnh hiện nay đó là nhận dạng khuôn mặt (Face Recognition).

Khuôn mặt đóng vai trò quan trọng trong quá trình giao tiếp giữa người với người, nó mang một lượng thông tin giàu có, chẳng hạn chúng ta có thể xác định giới tính, tuổi tác, chủng tộc, trạng thái cảm xúc,...

Bài toán nhận dạng khuôn mặt đóng vai trò quan trọng trong nhiều lĩnh vực quan trọng như các hệ thống giám sát, quản lý vào ra,...đặc biệt là an ninh, bảo mật.

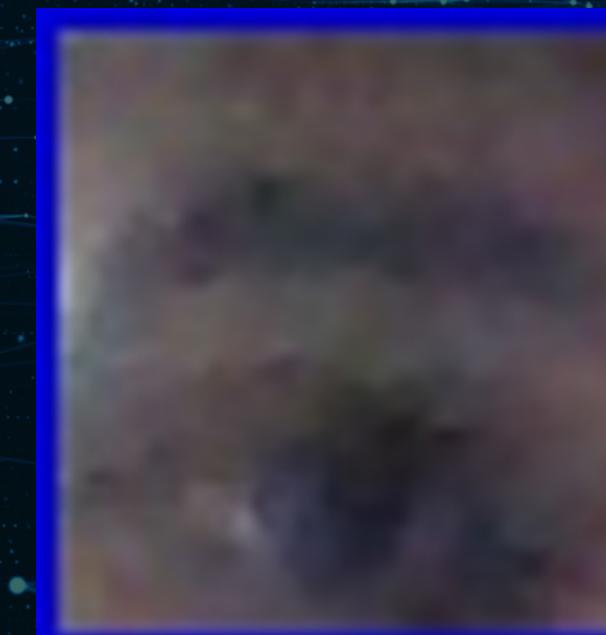
Có rất nhiều phương pháp nhận dạng khuôn mặt để nâng cao hiệu suất tuy nhiên dù ít hay nhiều những phương pháp này đang vấp phải những thử thách về độ sáng, hướng nghiêng, kích thước ảnh, hay ảnh hưởng của tham số môi trường.

Chính nhờ những yếu tố trên, nhóm đã đưa ra ý tưởng tạo ra phần mềm **nhận diện khuôn mặt và in ra thông tin sinh viên**.

# Công nghệ

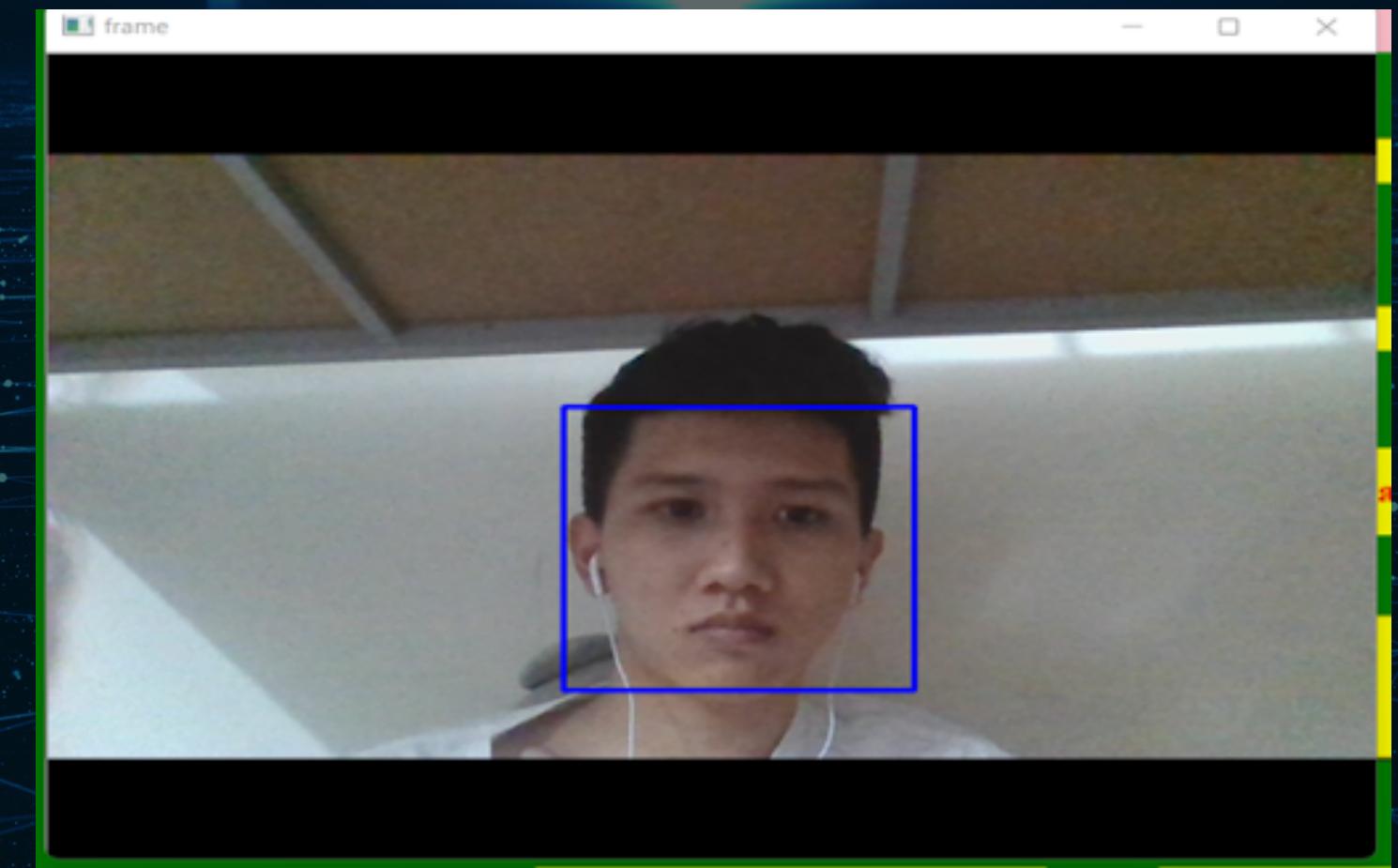
## Công nghệ MTCNN

- Về mặt cấu trúc MTCNN bao gồm 3 mạng CNN (Convolutional Neural Networks) xếp chồng và đồng thời hoạt động khi phát hiện và xác định khuôn mặt. Mỗi mạng CNN trong MTCNN có cấu trúc và vai trò khác nhau trong việc phát hiện khuôn mặt.
- Kết quả dữ liệu đầu ra MTCNN là véc-tơ đặc trưng biểu diễn cho vị trí khuôn mặt được xác định trong bức ảnh (mắt, mũi, miệng,...) MTCNN hoạt động theo 3 bước với 3 mạng nơ-ron riêng cho mỗi bước (P-Net, R-Net và O-Net). Khi sử dụng, MTCNN sẽ cho phép tạo ra nhiều bản sao của hình ảnh đầu vào, với các kích thước khác nhau để làm dữ liệu đầu vào.



## Công nghệ

**Tầng 1:** Sử dụng mạng CNN, gọi là Mạng đề xuất (P-Net), để thu được các cửa sổ chứa khuôn mặt và các vectơ hồi quy trong các cửa sổ đó. Tiếp theo, các cửa sổ chứa khuôn mặt được hiệu chuẩn dựa trên các vector hồi quy. Cuối cùng, những cửa sổ xếp chồng nhau tại một vùng được hợp nhất thành một cửa sổ. Kết quả đầu ra là các cửa sổ có thể chứa khuôn mặt.

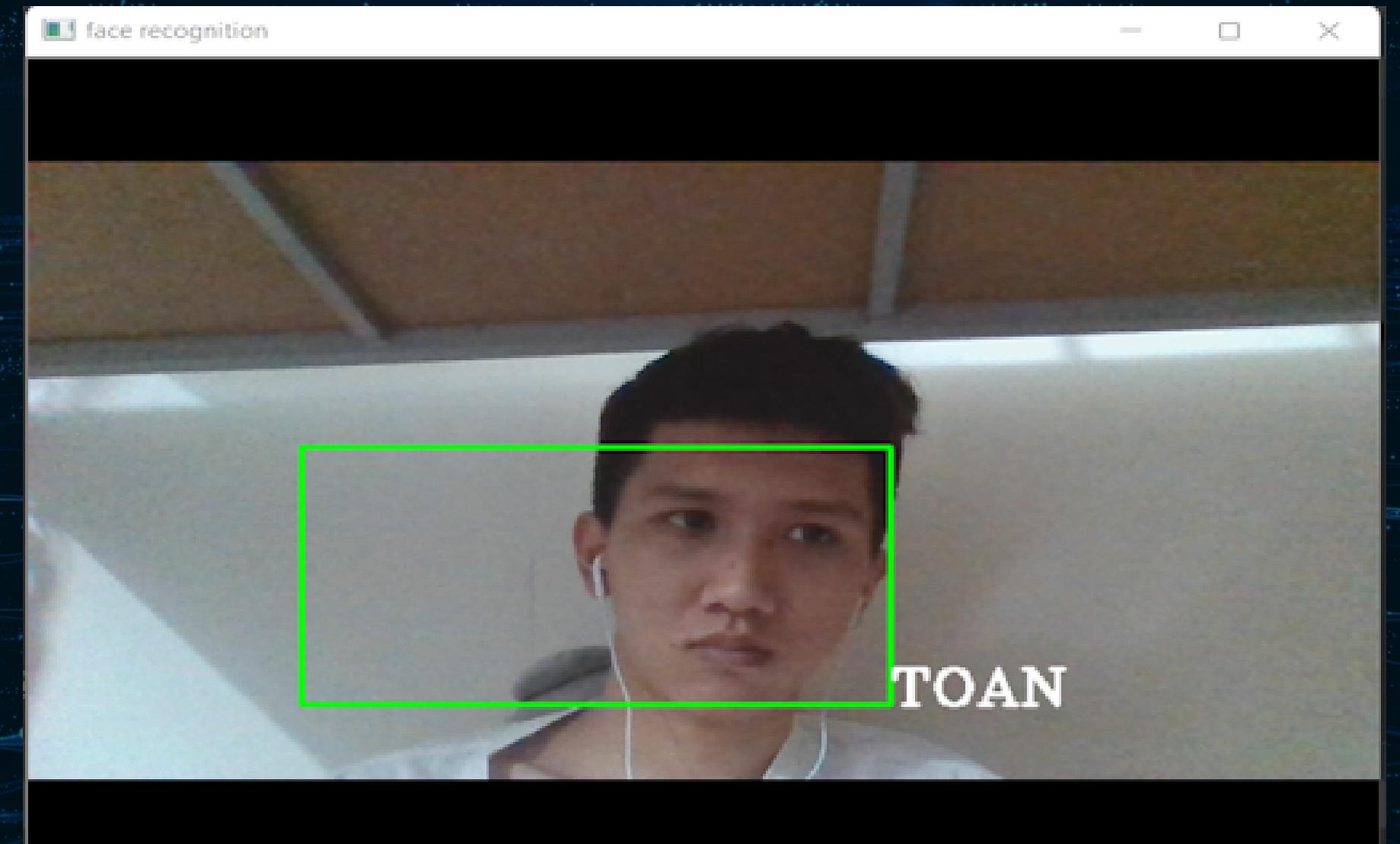


## Công nghệ

**Tầng 2:** Tất cả các cửa sổ chứa khuôn mặt từ tầng 1 sẽ được sàng lọc bằng cách đưa vào một CNN khác gọi là Màng lọc (R-Net) để tiếp tục loại bỏ một số lượng lớn các cửa sổ không chứa khuôn mặt. Sau đó, thực hiện hiệu chuẩn với véc-tơ hồi quy và thực hiện hợp nhất các cửa sổ xếp chồng nhau tại một vùng. Trong bước R-Net sử dụng kiến trúc CNN gồm: 3 lớp tích chập, 2 lớp co và 1 lớp kết nối đầy đủ. Đầu vào cửa sổ trượt với kích thước  $24 \times 24 \times 3$  (3 tương ứng với 3 màu: Đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường).

# Công nghệ

**Tầng 3:** Tầng này tương tự như tầng 2, sử dụng CNN chi tiết nhất được gọi là Mạng đầu ra (O-Net) để lọc kết quả một lần nữa và đánh dấu vị trí năm điểm chính trên khuôn mặt. Kết quả trả ra khuôn mặt đã được nhận diện



## Thư viện

- Cài đặt Python 3.9 và OpenCV => Làm việc với hình ảnh
- Cài đặt PIL => Xử lý hình ảnh
- Cài đặt numpy => numpy array: nơi lưu trữ những hình ảnh từ PIL image chuyển sang.
- Cài đặt Tkinter => Thiết kế giao diện làm việc với người dùng
- Cài đặt pandas => Tạo các DataFrame
- Cài đặt os => Làm việc với hệ thống kho ảnh trên máy tính
- Cài đặt Face\_recognition => thư viện nhận diện khuôn mặt bằng ngôn ngữ python
- Cài đặt Datetime => lấy được ngày tháng tại thời điểm hiện tại

# Function

## 4.1 Lấy dữ liệu từ bàn phím và webcam

- Lấy dữ liệu:

```
def TakeImages():  
    mssv = txt_mssv.get()  
    ho_ten = txt_ten.get()  
    nam_sinh = txt_nam_sinh.get()  
    khoa_hoc = txt_khoa_hoc.get()  
    lop_hoc = txt_lop.get()  
    ngay_cap_nhat = txt_ngay_cap_nhat.get()
```

MSSV	1
Họ và tên	Truong Cong Vinh
Năm sinh	2002
Khóa học	K20
Lớp sinh viên	K20416C
Ngày cập nhật	04/05/2022

# Function

## 4.1 Lấy dữ liệu từ bàn phím và webcam

- Mã nguồn dùng camera để chụp hình dữ liệu:

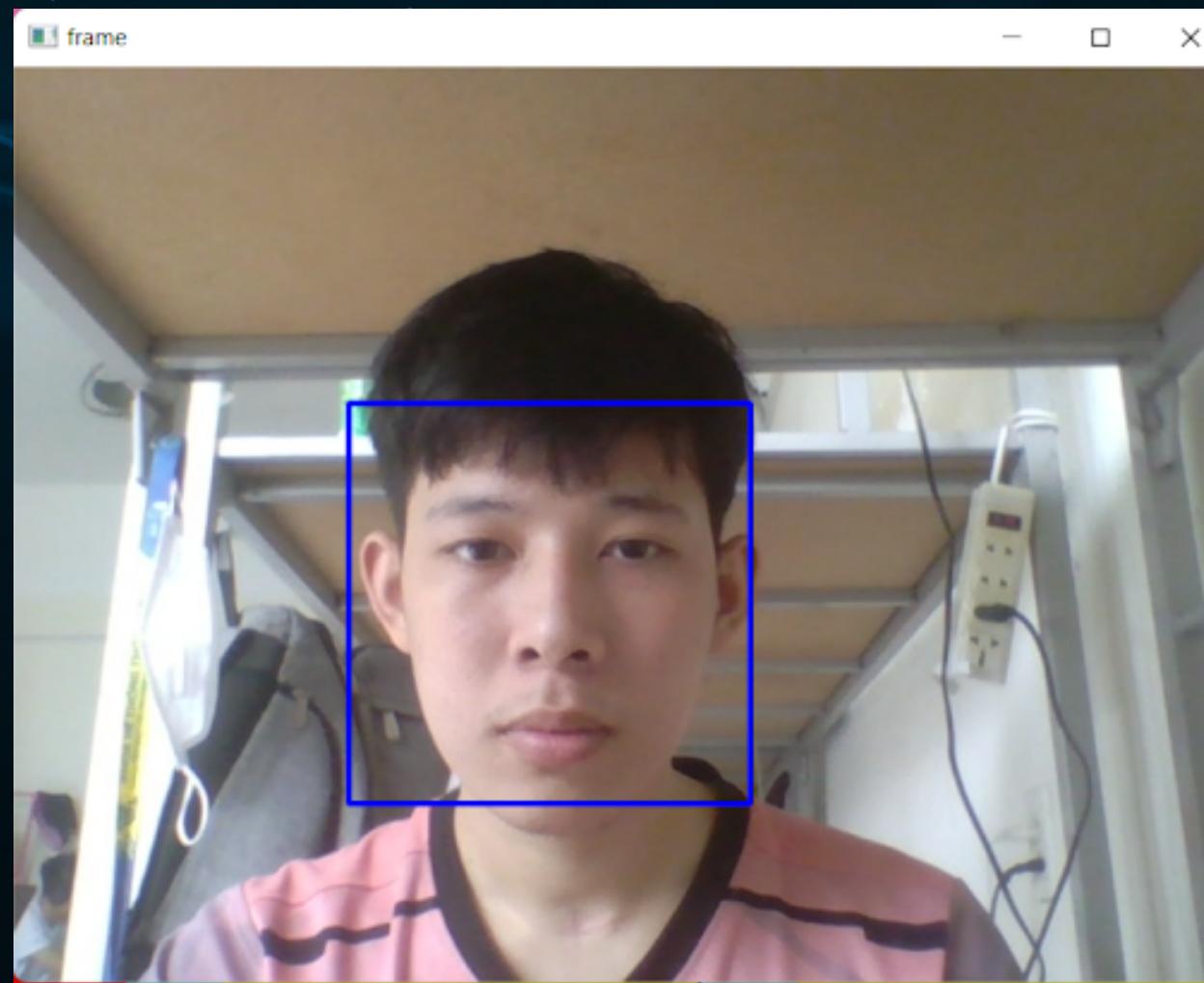
```
if is_number(mssv):
    cam = cv2.VideoCapture(0)
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector=cv2.CascadeClassifier(harcascadePath)
    sampleNum=0
    while(True):
        ret, img = cam.read()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = detector.detectMultiScale(gray, 1.3, 5)
        for (x, y, w, h) in faces:
            #Vẽ hình vuông nhận diện khuôn mặt
            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
            # Tăng sampleNum (ảnh thứ ..)
            sampleNum = sampleNum + 1
            # Lưu ảnh vào folder TrainingImage
            cv2.imwrite("TrainingImage\_" + ho_ten + "\_" + mssv + '\.' + str(sampleNum) + ".jpg",
                        gray[y:y + h, x:x + w])
            # Hiển thị frame
            cv2.imshow('frame', img)
```

```
cv2.imshow('frame', img)
# wait for 100 milliseconds
# Nhấn q để dừng quá trình thêm ảnh
if cv2.waitKey(100) & 0xFF == ord('q'):
    break
# break if the sample number is morethan 60
elif sampleNum > 60:
    break
cam.release()
cv2.destroyAllWindows()
res = "Thông tin đã được lưu trữ, MSSV: " + mssv + ", Tên: " + ho_ten
row = [mssv, ho_ten, nam_sinh, khoa_hoc, lop_hoc, ngay_cap_nhat]
with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
    writer = csv.writer(csvFile)
    writer.writerow(row)
csvFile.close()
messagebox.showinfo("THÔNG BÁO", res)
else:
    res = "Enter Numeric Id"
    messagebox.showinfo("THÔNG BÁO", res)
```

# Function

## 4.1 Lấy dữ liệu từ bàn phím và webcam

Hình ảnh được lưu trữ giới hạn là 60 hình



A	B	C	D	E	F
1	1 Truong Cong Vinh	2002	K20	K20416C	4/5/2022
2					

# Function

## 4.2 Training hình ảnh sinh viên và lưu trữ

Hình ảnh được lưu trữ giới hạn là 60 hình

```
def getImagesAndLabels(path):
    # Lấy đường dẫn tất cả các file trong folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]

    # Tạo một facelist
    faces=[]
    # Tạo một list MSSV
    Ids=[]
    # Run các path để tải mssv và image
    for imagePath in imagePaths:
        # tải hình ảnh và chuyển nó thành grayscale
        pilImage=PIL.Image.open(imagePath).convert('L')
        #Chuyển tất cả PIL image sang dạng numpy array
        imageNp=np.array(pilImage,'uint8')
        # Lấy mssv từ image
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        # Trích xuất các face có trong Image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces, Ids
```

```
def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, Id = getImagesAndLabels("TrainingImage")
    recognizer.train(faces, np.array(Id))
    recognizer.save("TrainingImageLabel\Trainer.yml")
    res = "Hình ảnh đã được học"
    messagebox.showinfo("THÔNG BÁO", res)
```

Tiến hành training và lưu vào file.

Tiến hành tách Id(mssv) từ tên file ảnh. Rồi Convert ảnh và add vào mảng faces cùng với ID. Đối với bài toán Classification ở đây thì Feature là ảnh còn Label chính là id người dùng.

# Function

## 4.3 Lớp detector nhận diện khuôn mặt đã training.

```
cam = cv2.VideoCapture(0)
while True:
    ret, im =cam.read()
    gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
    faces=faceCascade.detectMultiScale(gray, 1.2,5)
```

Sử dụng camera để nhận diện khuôn mặt người dùng

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

Thoát màn hình nhận diện

```
for(x,y,w,h) in faces:
    cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
    mssv, conf = recognizer.predict(gray[y:y+h,x:x+w])
    if(conf < 50):
        ts = time.time()
        date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
        timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

        s=df.loc[df['MSSV'] == mssv]['Họ và tên'].values
        tt=str(mssv)+"-"+s
        attendance.loc[len(attendance)] = [mssv, s, date, timeStamp]
    else:
        mssv='Unknown'
        tt=str(mssv)
    if(conf > 75):
        noOfFile=len(os.listdir("ImagesUnknown"))+1
        cv2.imwrite("ImagesUnknown\Image"+str(noOfFile) + ".jpg", im[y:y+h,x:x+w])
        cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)
attendance = attendance.drop_duplicates(subset=['MSSV'], keep='first')
cv2.imshow('Face-Recognition',im)
```

Hiển thị thông tin sinh viên lên màn hình nhận diện

## Đánh giá kết quả

Đạt được:

- Từ kết quả cho thấy, độ chính xác của chương trình chỉ đạt ở 70% nhận dạng đúng.
- Các quá trình quét, phân tích dữ liệu và trả kết quả được xử lý tương đối nhanh chóng.
- Khi nhận diện được khuôn mặt, ngoài chức năng điểm danh còn có một số chức năng khác như: xem thông tin sinh viên, danh sách sinh viên.

## Hệ thống vẫn còn một số hạn chế sau:

- Hệ thống không hoạt động, nếu ánh sáng không được cung cấp đủ.
- Hình ảnh nhận diện cần được chụp trong thời gian từ 2 tháng trở xuống.
- Hoạt động chưa đạt hiệu quả với người sử dụng khẩu trang và kính mát.
- Hệ thống điểm danh vẫn chưa tối ưu khi vẫn điểm danh cả những người không xác định và vẫn chưa kiểm soát được tình trạng nhờ bạn điểm danh giùm thông qua hình ảnh.
- Do thời gian hạn chế nên chương trình còn nhiều khuyết điểm và nhiều ý tưởng chưa được thực hiện.