**TRƯỜNG ĐẠI HỌC BÁCH KHOA ĐHQG TP.HCM**

**KHOA ĐIỆN – ĐIỆN TỬ**

**---o0o---**



**NHẬN DIỆN KHUÔN MẶT**

Môn: Thị Giác Máy

GVHD: TS. Nguyễn Đức Thành

**Danh sách nhóm 11:**

Trương Thiện Nhân 1512281

Bùi Thanh Tính 1513505

Trịnh Vũ Đăng Nguyên 1512232

Lê Quang Đức 1510787

*TP.HCM 11/2018*

Mục lục

[1 Giới thiệu đề tài 3](#_Toc530324303)

[1.1 Tổng quan về nhận dạng khuôn mặt 3](#_Toc530324304)

[1.1.1 Bài toán nhận diện khuôn mặt 3](#_Toc530324305)

[1.1.2 Những khó khăn và thách thức trong bài toán nhận diện khuôn mặt 3](#_Toc530324306)

[1.2 Các bước nhận dạng khuôn mặt 4](#_Toc530324307)

[1.3 Các ứng dụng của bài toán nhận diện khuôn mặt 5](#_Toc530324308)

[2 Giải thuật nhận dạng 8](#_Toc530324309)

[2.1 Haarcascade 8](#_Toc530324310)

[2.2 Local Binary Patterns Histograms (LBPH) 9](#_Toc530324311)

[2.3 Sơ đồ khối 12](#_Toc530324312)

[2.3.1 Đăng ký 12](#_Toc530324313)

[2.3.2 Kiểm tra 16](#_Toc530324314)

[3 Kết quả: 19](#_Toc530324315)

[4 Kết luận và hướng phát triển 21](#_Toc530324316)

[4.1 Kết luận 21](#_Toc530324317)

[4.2 Hướng phát triển 21](#_Toc530324318)

[5 Tài liệu tham khảo 22](#_Toc530324319)

# 1 Giới thiệu đề tài

**1.1 Tổng quan về nhận dạng khuôn mặt**

**1.1.1 Bài toán nhận diện khuôn mặt**

Ngày nay cùng với sự bùng nổ thông tin, sự phát triển công nghệ cao, sự giao tiếp giữa con người và máy tính đang thay đổi rất nhanh, giờ đây giao tiếp này không còn đơn thuần dùng những thiết bị cơ học như chuột, bàn phím... mà có thể thông qua các biểu hiện của khuôn mặt. Bên cạnh đó, công nghệ càng phát triển thì giá cả ngày càng giảm, thêm vào đó tốc độ xử lý của máy tính ngày càng cao, do đó hệ thống xử lí khuôn mặt đang được phát triển rất nhiều.

Trong số đó, có thể nói đến hệ thống phát hiện khuôn mặt, hệ thống này có thể giúp máy tính và con người giao tiếp với nhau tốt hơn. Những nghiên cứu trong hệ thống này chủ yếu dựa trên những thông tin trong ảnh để phát hiện vị trí khuôn mặt, làm bước đệm cho các ứng dụng tiếp theo. Rất nhiều nghiên cứu và ứng dụng được phát triển dựa trên hệ thống này. Đó là nhiệm vụ đầu tiên của bất kì hệ thống xử lý khuôn mặt nào. Tuy nhiên, đây cũng là một thử thách rất lớn bởi phát hiện được khuôn mặt còn dựa vào nhiều yếu tố như ti lệ, vị trí, hướng nhìn (từ trên xuống, quay,...), kiểu chụp (chụp đối diện, chụp ngang,...). Ngoài ra, những cảm xúc của khuôn mặt, một số phần bị che, hoặc hướng ánh sáng cũng ảnh hưởng đến bài toán phát hiện khuôn mặt.

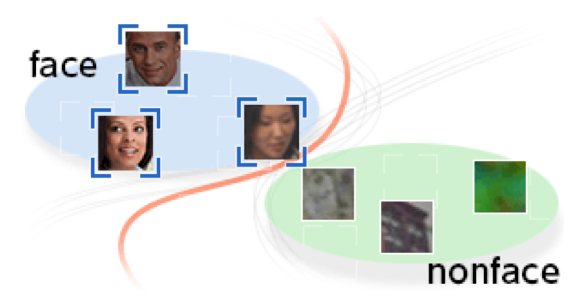
**1.1.2 Những khó khăn và thách thức trong bài toán nhận diện khuôn mặt**

Bài toán nhận dạng mặt người đã được nghiên cứu từ những năm 70, người đầu tiên là Kanade. Tuy nhiên, đây là một bài toán khó nên những nghiên cứu hiện tại vẫn chưa đạt được kết quả mong muốn. Có thể kể đến những khó khăn của bài toán nhận dạng mặt người như dưới đây:

* *Tư thế, góc chụp*: Ảnh chụp khuôn mặt có thể thay đổi rất nhiều bởi vì góc chụp giữa camera và khuôn mặt. Chẳng hạn như: chụp thẳng, chụp chéo bên trái 45° hay chéo bên phải 45°, chụp từ trên xuống, chụp từ dưới lên ...). Với các tư thế khác nhau, các thành phần trên khuôn mặt như mắt, mũi, miệng có thể bị khuất một phần hoặc thậm chí khuất hết.
* *Sự xuất hiện hoặc thiếu một số thành phần của khuôn mặt: Các đặc trưng như: râu* mép, râu hàm, mắt kính ... có thể xuất hiện hoặc không. Vấn đề này làm cho bài toán càng trở nên khó hơn rất nhiều. 3 Cảm xúc biểu hiện trên khuôn mặt: Cảm xúc có thể làm ảnh hưởng đáng kể lên các thông số của khuôn mặt. Chẳng hạn, cùng một khuôn mặt một người, nhưng có thể sẽ rất khác khi họ cười hoặc sợ hãi...
* *Sự che khuất*: Khuôn mặt có thể bị che khuất bởi các đối tượng khác hoặc các khuôn mặt khác.
* *Hướng của ảnh*: Các ảnh của khuôn mặt có thể biến đổi rất nhiều với các góc quay khác nhau của trục camera. Chẳng hạn chụp với trục máy ảnh nghiêng làm cho khuôn mặt bị nghiêng so với trục của ảnh.
* *Điều kiện của ảnh*: Anh được chụp trong các điều kiện khác nhau về: chiếu sáng, về tính chất camera (máy kỹ thuật số, máy hồng ngoại ...) ảnh hưởng rất nhiều đến chất lượng ảnh khuôn mặt.

**1.2 Các bước nhận dạng khuôn mặt**

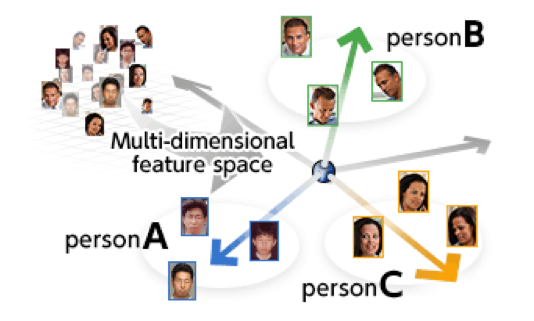
**Bước 1: phát hiện vị trí khuôn mặt**



**Bước 2: Tìm những vị trí đặc trưng trên khuôn mặt**



**Bước 3: Tìm kiếm và nhận diện khuôn mặt dựa trên các đặc trưng**



**1.3 Các ứng dụng của bài toán nhận diện khuôn mặt**

Ứng dụng của bài toán nhận diện khuôn mặt có rất nhiều và đã được triển khai tương đối hiệu quả trong thực tế. Có thể kể đến một số ứng dụng điển hình sau đây:

**Xác minh tội phạm**

Dựa vào ảnh của một người, nhận dạng xem người đấy có phải là tội phạm hay không bằng cách so sánh với các ảnh tội phạm đang được lưu trữ. Hoặc có thể sử dụng | camera để phát hiện tội phạm trong đám đông. Ứng dụng này giúp cơ quan an ninh quản lý con người tốt hơn.

**Camera chống trộm**

Các hệ thống camera sẽ xác định đầu là con người và theo dõi xem con người đó có làm gì phạm pháp không, ví dụ như lấy trộm đồ, xâm nhập bất hợp pháp vào một khu vực nào đó.

**Bảo mật**

Các ứng dụng về bảo mật rất đa dạng, một trong số đó là công nghệ nhận dạng mặt người của laptop, công nghệ này cho phép chủ nhân của máy tính chỉ cần ngồi trước máy là có thể đăng nhập được. Để sử dụng công nghệ này, người dùng phải sử dụng một webcam để chụp ảnh khuôn mặt của mình và cho máy “học” thuộc các đặc điểm của khuôn mặt giúp cho quá trình đăng nhập sau này.

**Lưu trữ khuôn mặt**

Xác định mặt người có thể được ứng dụng trong các trạm rút tiền tự động (ATM) để lưu trữ khuôn mặt của người rút tiền. Hiện nay có những người bị người khác lấy trộm thẻ ATM và mã PIN, và bị rút tiền trộm, hoặc có những chủ tài khoản đi rút tiền nhưng lại báo với ngân hàng là bị mất thẻ và bị rút tiền trộm. Nếu lưu trữ được khuôn mặt của người rút tiền, ngân hàng có thể đối chứng và xử lý dễ dàng hơn.

**Các ứng dụng khác**

Điều khiển vào ra: văn phòng, công ty, trụ sở, máy tính, Palm, .... Kết hợp thêm vân tay và mống mắt. Cho phép nhân viên được ra vào nơi cần thiết. go An ninh sân bay, xuất nhập cảnh (hiện nay cơ quan xuất nhập cảnh Mỹ đã áp dụng). Dùng camera quan sát để xác thực người nhập cảnh và kiểm tra xem người đây có phải là tội phạm hay phân tử khủng bố không.

Tìm kiếm và tổ chức dữ liệu liên quan đến con người thông qua khuôn mặt người trên nhiều hệ cơ sở dữ liệu lưu trữ thật lớn, như internet, các hãng truyền hình, ... Ví dụ: tìm các đoạn video có tổng thống Bush phát biểu, tìm các phim có diễn viên Lý Liên Kiệt đóng, tìm các trận đá bóng có Ronaldo đá, ...

Kiểm tra trạng thái người lái xe có ngủ gật, mất tập trung hay không, và hỗ trợ thông báo khi cần thiết.

Tương lai sẽ phát triển các loại thẻ thông minh có tích hợp sẵn đặc trưng của người dùng trên đó, khi bất cứ người dùng khác dùng để truy cập hay xử lý tại các hệ thống sẽ được yêu cầu kiểm tra các đặc trưng khuôn mặt so với thẻ để biết nay có phải là chủ thể hay không.

Hãng máy chụp hình Canon đã ứng dụng bài toán xác định khuôn mặt người vào

máy chụp hình thế hệ mới để cho kết quả hình ảnh đẹp hơn, nhất là khuôn mặt người.

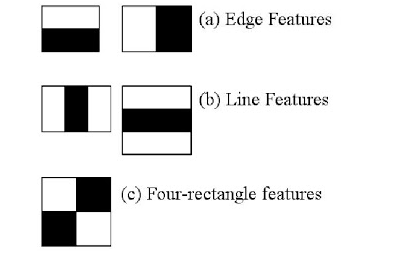
# 2 Giải thuật nhận dạng

## 2.1 Haarcascade

-Phát hiện đối tượng bằng cách sử dụng Haar dựa trên tính năng phân loại cascade

-Thuật toán cần nhiều hình ảnh tích cực (hình ảnh khuôn mặt) và hình ảnh tiêu cực (hình ảnh không có khuôn mặt) để đào tạo trình phân loại

-Các tính năng Haar được hiển thị trong hình ảnh dưới đây được sử dụng và thu được bằng cách trừ tổng số điểm ảnh trong hình chữ nhật màu trắng từ tổng số pixel dưới hình chữ nhật màu đen.



  Ví dụ: Hàng trên cùng hiển thị hai tính năng tốt. Tính năng đầu tiên được chọn có vẻ tập trung vào tài sản mà vùng mắt thường tối hơn vùng mũi và má. Tính năng thứ hai được lựa chọn dựa trên tài sản mà đôi mắt tối hơn cầu của mũi.

Áp dụng từng tính năng trên tất cả các hình ảnh đào tạo. Đối với mỗi đối tượng , tìm ra ngưỡng tốt nhất sẽ phân loại các mặt thành dương và âm. Rõ ràng, sẽ có lỗi hoặc phân loại sai.  Chọn các tính năng có tỷ lệ lỗi tối thiểu, có nghĩa là chúng là các tính năng phân loại chính xác nhất hình ảnh khuôn mặt và hình ảnh không phải khuôn mặt. (Quá trình này không đơn giản như vậy. Mỗi hình ảnh được đưa ra một trọng lượng bằng nhau trong phần đầu. Sau mỗi lần phân loại, trọng số của các hình ảnh sai phân loại được tăng lên. quá trình được tiếp tục cho đến khi đạt được độ chính xác hoặc tỷ lệ lỗi cần thiết hoặc số lượng đối tượng được yêu cầu).

Trình phân loại cuối cùng là tổng trọng số của các bộ phân loại yếu này. Nó được gọi là yếu bởi vì nó một mình không thể phân loại hình ảnh, nhưng cùng với những người khác tạo thành một phân loại mạnh mẽ.

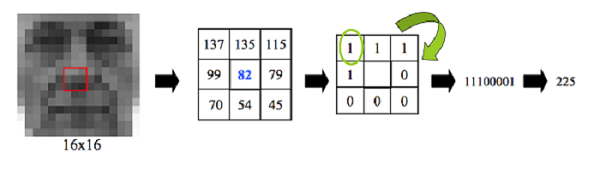
Trong một hình ảnh, hầu hết các hình ảnh là khu vực không phải đối mặt. Vì vậy, tốt hơn là nên có một phương pháp đơn giản để kiểm tra xem một cửa sổ không phải là vùng mặt. Nếu không, loại bỏ nó trong một shot duy nhất, và không xử lý nó một lần nữa. Thay vào đó, hãy tập trung vào các khu vực có thể có khuôn mặt. Bằng cách này, chúng tôi dành nhiều thời gian hơn để kiểm tra các vùng mặt có thể xảy ra.

Đối với điều này, họ đã giới thiệu khái niệm **Cascade of Classifiers** . Thay vì áp dụng tất cả 6000 tính năng trên một cửa sổ, các tính năng được nhóm lại thành các giai đoạn khác nhau của các trình phân loại và được áp dụng từng cái một. (Thông thường, một vài giai đoạn đầu sẽ chứa rất ít tính năng hơn). Nếu một cửa sổ thất bại trong giai đoạn đầu tiên, hãy loại bỏ nó. Chúng tôi không xem xét các tính năng còn lại trên đó. Nếu nó trôi qua, hãy áp dụng giai đoạn thứ hai của các tính năng và tiếp tục quá trình. Cửa sổ đi qua tất cả các giai đoạn là vùng mặt. Kế hoạch đó thế nào!

## 2.2 Local Binary Patterns Histograms (LBPH)

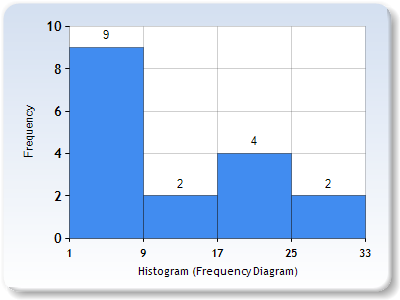
Lấy cửa sổ 3x3 và trượt nó trên một bức ảnh. Tại mỗi lượt di chuyển (mỗi phần local của bức ảnh), so sánh pixel tại trung tâm với những pixels xung quanh. Nếu pixels xung quanh có giá trị mức xám lớn hơn hoặc bằng pixel trung tâm thì ký hiệu là 1 và ngược lại thì bằng 0.

Sau đó, nó đọc giá trị mới của pixel theo chiều kim đồng hồ dưới dạng một số binary. Tiếp theo, nó chuyển số binary này sang thập phân, và số thập phân đó là giá trị mới của pixel trung tâm. Thuật toán LBPH thực hiện quá trình này cho mỗi pixel trong khối.



Hình LBP chuyển ảnh sang binary. Nguồn: López & Ruiz; Local Binary Patterns applied to Face Detection and Recognition.

Bây giờ, sau khi chúng ta được một danh sách các LBP, nó thành lập một đò thị histogram của những giá trị thập phân. Một đò thị histogram mẫu như sau:



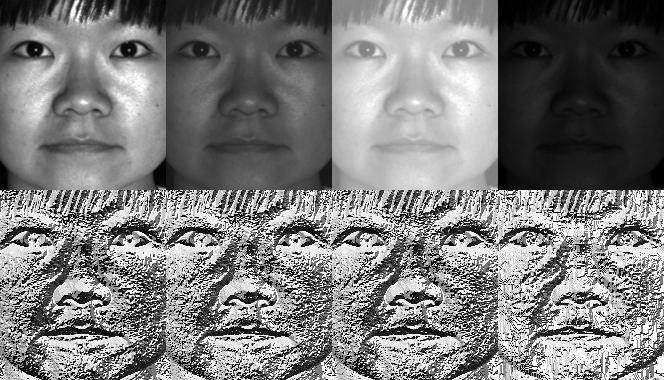
Hình một đồ thị histogram mẫu.

Sau đó, chúng ta sẽ có một histogram cho mỗi ảnh khuôn mặt trong training dataset. Điều đó có nghĩa là nếu có 100 ảnh trong dataset thì LBPH sẽ tạo ra 100 histograms sau khi training và lưu chúng cho bước nhận dạng.

Trong bước nhận dạng, quá trình diễn ra như sau:

* Cung cấp một ảnh mới đến recognizer để nhận dạng.
* Recognizer tạo ra một histogram cho ảnh mới.
* Sau đó, nó so sánh histogram này với những histogram đã có sẵn.
* Cuối cùng, nó tìm khuôn mặt phù hợp nhất, rồi trả lại nhãn của khuôn mặt phù hợp và khoảng cách giữa ảnh mới và ảnh này.

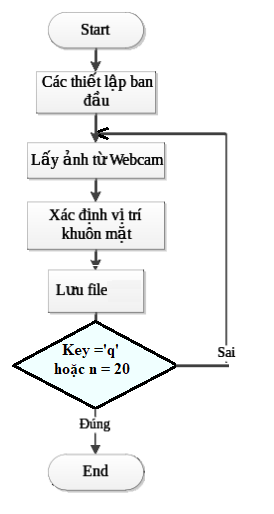
Do đặc trưng được chọn lọc dựa theo sự chênh lệch mức xám nên LBPH có tính chất tương đối ổn định đối với việc ảnh bị thay đổi cường độ sáng. Ảnh sau thể hiện cụ thể:



Hình LBPH Face Recognizer Principal Components. Source: [**docs.opencv.org**](http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html)

## 2.3 Sơ đồ khối

### 2.3.1 Đăng ký

****

**Các thư viện cần sử dụng:**

|  |
| --- |
| import cv2 as cv  import numpy as np  import sqlite3 |

**Tạo hàm chỉnh sửa hoặc update dữ liệu**

|  |
| --- |
| def insertOrUpdate(Id,Name):  conn=sqlite3.connect("facedata.db")  cmd="SELECT \* FROM facedata WHERE ID="+str(Id)  cursor=conn.execute(cmd)  isRecordExist=0  # Neu gia tri ID da co thi update nguoc lai insert  for row in cursor:  isRecordExist=1  if(isRecordExist==1):  cmd="UPDATE facedata SET Name='"+str(Name)+"'WHERE ID="+str(Id)  else:  cmd="INSERT INTO facedata(ID,Name) Values("+str(Id)+",'"+str(Name)+"')"  conn.execute(cmd)  conn.commit()  conn.close() |

**Các thiết lập ban đầu:**

Lựa chọn camera, nhập tên và ID người cần đăng ký.

|  |
| --- |
| faces\_cascade = cv.CascadeClassifier('haarcascade\_frontalface\_default.xml')  cam = cv.VideoCapture(0)  ident = input('Enter your ID: ')  name=input('Enter your name: ')  insertOrUpdate(ident,name)  number\_sample = 0 |

**Xác định vị trí khuôn mặt:**

Ở bước này, đầu vào sẽ là bức ảnh có chứa khuôn mặt người cần đăng ký. Yêu cầu ngõ ra phải xác định được vị trí khuôn mặt trên bức ảnh để ta thực hiện lưu vào dataset.

|  |
| --- |
| while True:  ret, img = cam.read()  img\_gray = cv.cvtColor(img, cv.COLOR\_BGR2GRAY)  # phat hien khuon mat  faces = faces\_cascade.detectMultiScale(img\_gray, 1.3, 5)  for (x, y, w, h) in faces:  number\_sample = number\_sample + 1  # luu khuon mat dung dia chi file dataset  cv.imwrite('dataset/person' + '\_' + str(ident) + '\_' + str(number\_sample) + '.jpg', img\_gray[y:y+h, x:x+w])  cv.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)  cv.waitKey(200)  cv.imshow('image', img)  # Lay 20 anh khuon mat sau moi 0.2 giay  if (number\_sample > 19):  break  if (cv.waitKey(100) & 0xff == ord('q')):  break  cam.release()  cv.destroyAllWindows() |

**Code training:**

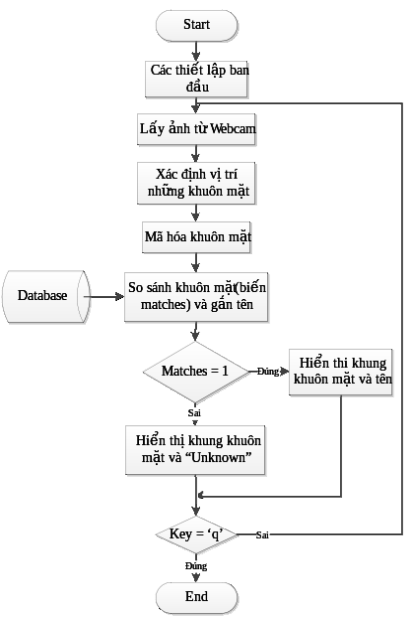
**Tạo hàm lấy dữ liệu ảnh để training:**

|  |
| --- |
| def FacesAndID(path):  # Dia chi cua nhung khuon mat  face\_paths = [os.path.join(path, file) for file in os.listdir(path)]  faces = []  IDs = []  for face\_path in face\_paths:  imagepil = Image.open(face\_path).convert('L')  imagenp = np.array(imagepil, 'uint8')  Id=int(os.path.split(face\_path)[1].split("\_")[1])  faces.append(imagenp)  IDs.append(Id)  return faces, np.array(IDs) |

**Lưu mã khuôn mặt sau khi training:**

|  |
| --- |
| faces,IDs = FacesAndID(path)  # Thuc hien training  training.train(faces, IDs)  # Luu ket qua vao file .yml  training.save('trained\_file/trained\_file.yml') |

### 2.3.2 Kiểm tra

****

**Các thư viện cần sử dụng:**

|  |
| --- |
| import cv2 as cv  import numpy as np  import sqlite3 |

**Các khởi tạo ban đầu**

|  |
| --- |
| faces\_detect = cv.CascadeClassifier('haarcascade\_frontalface\_default.xml')  cam = cv.VideoCapture(0)  recognize = cv.face.LBPHFaceRecognizer\_create()  recognize.read('trained\_file/trained\_file.yml') |

**Tạo hàm trả về tên tương ứng với ID:**

|  |
| --- |
| # Ham tra ve ten tuong ung voi ID  def getProfile(id):  conn=sqlite3.connect("facedata.db")  cmd="SELECT \* FROM facedata WHERE ID="+str(id)  cursor=conn.execute(cmd)  profile=None  for row in cursor:  profile=row[1]  conn.close()  return profile |

**Tiến hành matching:**

|  |
| --- |
| while True:  ret, image = cam.read()  image\_gray = cv.cvtColor(image, cv.COLOR\_BGR2GRAY)  # phat hien khuon mat  faces = faces\_detect.detectMultiScale(image\_gray, 1.3, 5)  for (x, y, w, h) in faces:  cv.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)  # Matching khuon mat moi voi nhung khuon mat trong dataset  ID, conf = recognize.predict(image\_gray[y:y+h, x:x+w])  #print(str(conf))  profile = getProfile(ID)  # Neu khoang cach > 60 thi 2 khuon mat khong cung mot nguoi  # Nguoc lai, hien ten nguoi tuong ung  if(conf < 60):  if(profile!=None):  cv.putText(image, str(profile), (x, y+h), cv.FONT\_HERSHEY\_SIMPLEX, 1, (255, 0, 0), 2)  else:  cv.putText(image, str('Unknown'), (x, y+h), cv.FONT\_HERSHEY\_SIMPLEX, 1, (255, 0, 0), 2)  cv.imshow('Face', image)  if (cv.waitKey(1) & 0xff == ord('q')):  break  cam.release()  cv.destroyAllWindows() |

# 3 Kết quả:

**-Hình ảnh và video ảnh huấn luyện: (mỗi người 50 hình)**



**Moose NA RuBy Smolder**

**Ảnh đầu ra (nhận dạng)**



Ảnh nhận dạng đúng tất cả.

Ảnh Nhận dạng đúng 2 người

Kiểm tra: độ chính xác không cao (85%) nguyên nhân do ảnh hưởng của ánh sáng và cảnh quan xung quanh làm cho nhận diện sai, mặt nghiêng khó nhận diện

-Độ chính xác cao cần phải huấn luyện nhiều hình

**Video nhận dạng:tệp đính kèm**

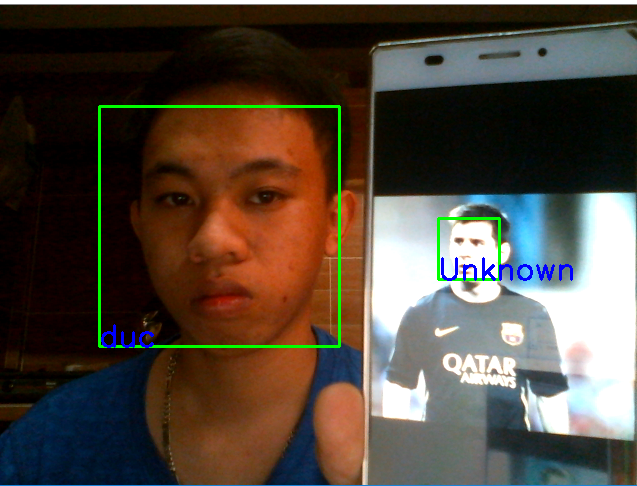
**-Camera:**

**Ảnh huấn luyện: (Mỗi người 50 ảnh huấn luyện)**



Tinh duc

Kết quả đầu ra camera



# 4 Kết luận và hướng phát triển

## 4.1 Kết luận

Nhận dạng khuôn mặt là một trong những đề tài lớn được phát triển từ rất sớm trong lĩnh vực thị giác máy. Tuy nhiên, nó vẫn được rất nhiều nhà nghiên cứu trong nước và trên thế giới qua tâm trong suốt 20 năm qua, do các ứng dụng sâu rộng trong đời sống.

Trong đề tài nhận dạng khuôn mặt, nhóm 11 đã phát triển chương trình Python dùng để phát giác, training và nhận dạng khuôn mặt từ dữ liệu training. Nhóm đã nắm được cấu trúc chương trình nhận dạng và sử dụng một thuật toán có tên là Haar classifier, một trong những thuật toán đơn giản và độ chính xác cao, để phát giác khuôn mặt. Sau đó, trong quá trình nhận dạng, thuật toán LBPH được sử dụng để trích đặc trưng (training) và so sánh các khuôn mặt với nhau. Các thuật toán nêu trên đều có sẵn model trong OPENCV và được sử dụng rộng rãi.

## 4.2 Hướng phát triển

Tuy hệ thống được phát triển có độ chính xác tương đối cao, nhưng trong các ứng dụng hiện nay đang yêu cầu cao hơn thế nữa. Các đặc trưng ổn định hơn có thể được sử dụng như: SIFT, SURF,… Trong thời đại Internet phát triển với nguồn cung cấp và trao đổi dữ liệu khổng lồ, các model được training sử dụng CNN đang chím ưu thế.

# 5 Tài liệu tham khảo

[1] Face Recognition using OpenCV and Python: A Beginner’s Guide, https://www.superdatascience.com/opencv-face-recognition/ , (access: November 18th, 2018).

[2] Face Detection using OpenCV and Python: A Beginner’s Guide, <https://www.superdatascience.com/opencv-face-detection/>, (access: November 18th, 2018).

[3] Marcelo Rovai, Real-Time Face Recognition: An End-To-End Project, <https://medium.com/@rovai/real-time-face-recognition-an-end-to-end-project-b738bb0f7348> , (access: November 18th, 2018).