**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated

**BÁO CÁO ĐỀ TÀI CUỐI KỲ**

**Tìm hiểu và ứng dụng về xác thực bằng sinh trắc học – khuôn mặt**

Giảng viên hướng dẫn: ThS. Nguyễn Thị Thanh Vân

Sinh viên thực hiện: Nguyễn Phúc An 19110321

Bành Đăng Khoa 19110378

TP. Hồ Chí Minh, tháng 12 năm 2021

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến cô Nguyễn Thị Thanh Vân đã giảng dạy và hướng dẫn chúng em trong suốt quá trình học môn An Toàn Thông Tin, cô luôn giải đáp thắc mắc mà chúng em có cũng như luôn tạo điều kiện học tập thuận lợi nhất cho chúng em trong quá trình học tập. Cô cũng đã hướng dẫn, góp ý cho chúng em trong suốt quá trình làm đề tài để cho chúng em không bị lạc hướng hay sai đề.

Vì khả năng còn hạn chế nên trong quá trình thực hiện báo cáo không tránh khỏi sơ sài, sai sót, nên mong cô thông cảm cũng như có những nhận xét, đánh giá để em có thể hoàn thành bài báo cáo này tốt hơn

# MỤC LỤC

LỜI CẢM ƠN 2

MỤC LỤC 1

CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI 2

1.1 Tổng quan về đề tài 2

1.2 Hướng đi của đề tài 2

1.3 Bố cục của bài báo cáo 2

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THUẬT TOÁN SỬ DỤNG 3

2.1 Lý thuyết 3

2.1.1 Sinh trắc học là gì ? 3

2.1.2 Bài toán phát hiện và nhận diện khuôn mặt và các vấn đề gặp phải 4

2.1.3 Các phương thức nhận diện khuôn mặt 5

2.1.4 Phương pháp mô tả đặc trưng Histogram of Oriented Gradients 6

CHƯƠNG 3: ỨNG DỤNG THỰC TẾ 12

3.1 Ứng dụng máy tính sử dụng Python 12

3.2 Ứng dụng điện thoại sử dụng Python 14

3.3 Ứng dụng máy tính sử dụng C# 16

TÀI LIỆU THAM KHẢO 17

# CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI

## Tổng quan về đề tài

- Mục tiêu của đề tài là tìm hiểu về lý thuyết của sinh trắc học khuôn mặt, cách hoạt động, các thuật toán được sử dụng và cách áp dụng sinh trắc học khuôn mặt vào một ứng dụng thực tế.

## Hướng đi của đề tài

- Tìm hiểu về lý thuyết cơ bản của sinh trắc học.

- Tìm hiểu lý thuyết cơ bản của nhận diện khuôn mặt.

- Tìm hiểu về các thuật toán được sử dụng để phát hiện và nhận diện khuôn mặt.

- Xây dựng ứng dụng sử dụng bằng ngôn ngữ lập trình Python sử dụng sinh trắc học khuôn mặt để đăng nhập vào ứng dụng.

- Giải thích cách thuật toán, code hoạt động.

## Bố cục của bài báo cáo

CHƯƠNG1: GIỚI THIỆU VỀ ĐỀ TÀI

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THUẬT TOÁN SỬ DỤNG

CHƯƠNG 3: ỨNG DỤNG THỰC TẾ

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THUẬT TOÁN SỬ DỤNG

## Lý thuyết

### Sinh trắc học là gì ?

**-** Sinh trắc học là môn khoa học ứng dụng phân tích toán học thống kê xác suất để nghiên cứu các hiện tượng sinh học hoặc các chỉ tiêu sinh học có thể đo lường được.Khái niệm này có nguồn gốc từ tiếng Anh "biometry" hoặc tiếng Pháp "biométrie".

- Mỗi người có một đặc điểm sinh học duy nhất. Dữ liệu sinh trắc học của từng cá nhân với đặc điểm khuôn mặt, ảnh chụp võng mạc, giọng nói sẽ được kết hợp với nhau bằng phần mềm để tạo ra mật khẩu dành cho những giao dịch điện tử, phương thức đó là "công nghệ sinh trắc đa nhân tố"Sự phát triển của công nghệ đã thay đổi từ việc lăn tay trên mực và lưu trữ trên giấy sang quét trên máy và lưu trữ kỹ thuật số.

- Xác thực sinh trắc học là một hình thức bảo mật đo lường và đối sánh các tính năng sinh trắc học của người dùng để xác minh rằng một người đang cố gắng truy cập vào một thiết bị cụ thể được phép làm như vậy. Đặc điểm sinh trắc học là các đặc điểm vật lý và sinh học dành riêng cho một cá nhân và có thể dễ dàng so sánh với các đặc điểm được phép lưu trong cơ sở dữ liệu. Nếu các tính năng sinh trắc học của một người dùng đang cố gắng truy cập vào một thiết bị khớp với các tính năng của một người dùng được phê duyệt thì quyền truy cập vào thiết bị sẽ được cấp. Xác thực sinh trắc học cũng có thể được cài đặt trong môi trường vật lý, kiểm soát các điểm truy cập như cửa ra vào và cổng.

- Công nghệ nhận dạng khuôn mặt dựa trên việc so khớp hàng chục phép đo khác nhau từ khuôn mặt đã được phê duyệt với khuôn mặt của người dùng đang cố gắng truy cập, tạo ra cái được gọi là khuôn mặt. Tương tự như máy quét vân tay, nếu đủ số lượng phép đo từ người dùng khớp với khuôn mặt đã được phê duyệt, quyền truy cập sẽ được cấp. Tính năng nhận dạng khuôn mặt đã được thêm vào một số điện thoại thông minh và các thiết bị phổ biến khác, mặc dù nó có thể không nhất quán khi so sánh các khuôn mặt khi nhìn từ các góc độ khác nhau hoặc khi cố gắng phân biệt giữa những người trông giống nhau, chẳng hạn như họ hàng gần.

### Bài toán phát hiện và nhận diện khuôn mặt và các vấn đề gặp phải

- Phát hiện khuôn mặt (Face detection) là một chủ đề nghiên cứu được quan tâm nhiều đến trong lĩnh vực thị giác máy tính (Computer vision) cũng như trong Xử lý ảnh (Image processing) vào những năm 90 và là tiền đề cho nhận diện khuôn mặt (Face recognition). Cho tới hiện nay chủ đề này không dừng ở việc nhận diện không mà còn được nâng cao hơn giúp dự đoán hành vi, dự đoán độ tuổi, dự đoán cảm xúc, dự đoán giới tính và được áp dụng làm cơ chế bảo mật sinh trắc học (Biometrics). Hiện nay mặc dù đã có các loại sinh trắc học khác như sinh trắc học vân tay, sinh trắc học mẫu mắt thì sinh trắc học khuôn mặt vẫn được rất quan tấm đến, điển hình là Apple là người tiên phong đầu tiên áp dụng hệ thống sinh trắc học khuôn mặt lên điện thoại thông minh.

- Trước khi có được những ứng dụng tuyệt vời như thế, những nhà nghiên cứu đi đầu đã phải giải quyết các vấn đề như:

+ Kích cỡ hình ảnh thay đổi liên tục, không có một mẫu cụ thể, kèm vào đó là kích cỡ của khuôn mặt con người trong mỗi bức hình cũng sẽ thay đổi liên tục.

+ Các vấn đề về điều kiện ánh sáng của bức ảnh: Mỗi bức ảnh có thể có độ sáng khác nhau, bức ảnh có thể bị chói sáng hoặc không đủ ánh sáng để khuôn mặt con người được hiển thị rõ ràng.

+ Các vấn đề về hướng của khuôn mặt trong bức ảnh: Các bức ảnh khác nhau có thể có các khuôn mặt đang xoay ở các hướng khác nhau, điều này cũng là một vấn đề cần giải quyết khi cố gắng cho máy phát hiện được đâu là khuôn mặt con người hay nhận diện đâu là khuôn mặt con người và đâu không phải là khuôn mặt con người.

+ Các vấn đề về biểu cảm của khuôn mặt trong bức ảnh: Khuôn mặt con người có thể có các biểu cảm khác nhau dẫn đến thay đổi nhỏ về các đặc trưng của khuôn mặt ấy.

+ Các vấn đề về khuôn mặt bị che khuất trong bức ảnh: chân dung của một người trong bức ảnh có thể có các trường hợp bị che khuẩn nửa trên, nửa dưới,…

### Các phương thức nhận diện khuôn mặt

- Để có thể nhận diện và phát hiện khuôn mặt, chúng ta cần sử dụng đến các hệ thống học máy (Machine learning) để khuôn mặt con người trong một bức ảnh, thông thường trong bức ảnh đấy sẽ có các đối tượng khác không phải là khuôn mặt con người tựa như toà nhà, phong cảnh… Các thuật toán nhận diện khuôn mặt thường tìm kiếm con mắt của người đầu tiên do con mắt là đặc trưng dễ nhất để nhận diện khuôn mặt, sau đấy thuật toán có thể thử tìm kiếm lông mày, miệng, mũi,… Một khi thuật toán kết luận rằng nó đã tìm thấy một khu vực hình ảnh có thể chứa khuôn mặt con người thì nó tiếp tục thử qua các điều kiện, yêu cầu để đưa ra kết luận cuối cùng.

- Để đạt được độ chính xác cao, thuật toán cần được rèn luyện (trainning) qua một tệp dữ liệu lớn bao gồm ngàn các bức ảnh khác nhau, các bức ảnh không chứa khuôn mặt con người (negative images) và các bức ảnh có chứa khuôn mặt con người (positive images).

- Các phương pháp được sử dụng để phát hiện khuôn mặt:

+ Phương pháp dựa vào đặc trưng (Feature-based): Dựa vào đặc trưng của một khuôn mặt như mắt, mũi, miệng… Có thể bị ảnh hưởng bới các yếu tố như ánh sáng, độ mờ của bức ảnh.

+ Phương pháp dựa vào biểu hiện (Appearance-based): Dựa vào thống kê phân tích và học máy để tìm kiếm những điểm tương đồng đặc trưng giữa các khuôn mặt trong các bức ảnh. Phương thức này còn được sử dụng trong các thuật toán trích xuất đặc trưng để nhận diện khuôn mặt.

+ Phương pháp dựa vào tri thức (Knowledge-based): Dựa vào các đặc điểm, yêu cầu bắt buộc mà một khuôn mặt phải có để phát hiện khuôn mặt.

+ Phương pháp dựa vào so sánh khớp (Templatematching): Dựa vào việc so sánh hình ảnh với một kiểu mẫu khuôn mặt tiêu chuẩn hoặc các đặc trưng đã được lưu trữ trước và so sánh song song giữa cả hai để phát hiện khuôn mặt.

- Một vài cách được sử dụng để việc phát hiện khuôn mặt dễ dàng hơn:

+ Thủ thuật xoá ảnh nền, giả sử một bức ảnh có một phong nền đơn sắc, thì việc xoá ảnh nền có thể giúp phát hiện viền khuôn mặt ở đâu trên bức ảnh.

+ Thủ thuật dựa vào màu da, màu da đôi khi được sử dụng để tìm kiếm khuôn mặt trong bức hình.

### Phương pháp mô tả đặc trưng Histogram of Oriented Gradients

- Phương pháp mô tả đặc trưng Histogram of Oriented Gradients (gọi tắc là HOG) được công bố lần đầu tiên vào năm 1986 và trở nên phổ biên hơn bởi khi được công bố những bổ sung về phương pháp HOG vào năm 2005 Navneet Dalal và Bill Triggs ở Hội nghị thị giác máy tính và nhận diện khuôn mẫu (Conference on Computer Vision and Pattern Recognition). Phương pháp HOG được sử dụng để trừu tượng hoá đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích.

- Phương pháp HOG có 5 bước chính:

**Bước 1**: Chuẩn hoá hình ảnh.

+ Hình ảnh có thể chuyển đổi sang dạng trắng đen và được resize lại nhỏ hơn để tiết kiệm thời gian chạy của phương pháp cũng như chuẩn hoá kết quả đầu ra.

**Bước 2**: Tính toán hướng dốc của vector cường độ màu sắc trong ảnh (Gradient).

+ Chúng ta sẽ lấy từng pixel một trên toàn bộ hình ảnh, với mỗi pixel chúng ta sẽ quan tâm đến các pixel xung quanh nó và dựa vào các pixel ấy chúng ta có thẻ vẽ một vector có hướng chỉ chiều của màu sắc đang tối dần, các vector này gọi là gradient.

Graphical user interface

Description automatically generated with medium confidence

NGUỒN: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>

+ Lập lại bước này liên tục cho từng pixel trên bức ảnh ta sẽ thay thế các pixel thành các gradient.

+ Việc tính toán các gradient trên toàn pixel sẽ tốn rất nhiều thời gian cũng như là không cần thiết cho bài toán của chúng ta vì thế chúng ta sẽ chia hình ảnh thành các khối 16x16 pixel. Với mỗi khối ảnh như thế ta sẽ đếm theo chiều của các vector đang chỉ theo từng pixel (hướng lên-phải, hướng lên-trái, hướng phải, hướng trái,…,), hướng vector nào có số lượng nhiều nhất sẽ là hướng vector chủ đạo của khối đó.

+ Kết quả của việc tính toán, chuyển đổi này là ta sẽ có được một bức ảnh đơn giản hoá của bức ảnh ban đầu mà ở đó các cấu trúc của khuôn mặt được biểu diễn một cách đơn giản.

A picture containing text

Description automatically generated

NGUỒN: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>

+ Để tìm kiếm khuôn mặt trong hình ảnh đã được chuyển đổi thành dạng HOG, chúng ta sẽ so sánh giữa bức ảnh ở dạng HOG với một cấu trúc HOG được tạo ra từ việc training học máy với cả ngàn bức ảnh chứa khuôn mặt khác nhau để có được một cấu trúc HOG đại diện cho khuôn mặt của con người. Khi so sánh, ta sẽ tìm khu vực nào trong bức ảnh HOG trùng với cấu trúc HOG được nói ở trên nhiều nhất, khu vực ấy sẽ là khu vực của khuôn mặt con người trong ảnh.

**Bước 3:** Chỉnh hướng và hình chiếu của khuôn mặt.

+ Sau khi đã nhận diện được khu vực của khuôn mặt trong hình ảnh thì chúng ta phải giải quyết các vấn đề liên quan đến hướng của khuôn mặt trong bức ảnh, có khả năng khuôn mặt trong bức ảnh có hướng khác với khuôn mặt mà chúng ta lấy làm mẫu để so sánh cùng và nhận diện.

+ Để phương pháp HOG và việc nhận diện có thể được thực hiện với độ chính xác cao ta sẽ điều chỉnh hướng của khuôn mặt trong ảnh sao cho phần môi và hai mắt sẽ ở cùng một vị trí so với khuôn mặt mẫu.

+ Giải quyết vấn đề về hướng của khuôn mặt, chúng ta sẽ sử dụng thuật toán ước lượng các điểm đặc trưng của khuôn mặt (face landmark estimation). Ý tưởng của thuật toán này là chúng ta sẽ tìm kiếm 68 điểm đặc trưng (landmarks) tồn tại trên mọi khuôn mặt ( đỉnh của cằm, các điểm viền của mắt,…), chúng ta sẽ sử dụng học máy để rèn luyện cho thuật toán có thể tìm được 68 điểm ấy trên mọi khuôn mặt.

Chart, scatter chart

Description automatically generated

TÁC GIẢ: Brandon Amos, <https://bamos.github.io/>

+ Sau khi xác định được các điểm đặc trưng của khuôn mặt, chúng ta sẽ cố gắng xoay, phóng to hoặc thu nhỏ bức ảnh cho tới khi hai mắt và miệng ở giữa khung ảnh nhất có thể.

A collage of a person's face

Description automatically generated with medium confidence

NGUỒN: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>

**Bước 4:** Số hoá đặc trưng của khuôn mặt.

+ Qua 3 bước trên ta đã có khu vực của khuôn mặt con người trong hình ảnh đã được điều chỉnh hướng, với dữ liệu này việc dễ nhất để nhận diện khuôn mặt trong hình ảnh là so sánh với các khuôn mặt mẫu đã được đánh tên trước đó, tuy nhiên có rất nhiều vấn đề có thể xảy ra khi sử dụng phương pháp này khi áp dụng với cơ sở dữ liệu lớn.

+ Để có thể xử lý vấn đề này một cách tốt hơn ta có thể trích xuất một vài đặc trưng cơ bản của từng khuôn mặt, sau đó chúng ta có thể so sánh các trích xuất đặc trưng này với một trích xuất đặc trưng đã được chuẩn bị trước và được đánh tên, trích xuất với tên nào cho ra kết quả gần giống nhất thì sẽ là tên của người trong ảnh. Các đặc trưng này có thể là kích thước của tai, khoản cách giữa hai mắt, độ dài của mũi…

+ Các đặc trưng tốt nhất mà được trích xuất ra sẽ được quyết định bởi chính các thuật toán học máy.

+ Để có thể trích xuất được các đặc trưng của khuôn mặt chúng ta phải có một thuật toán học sâu (Deep Convolutional Neural Network), thuật toán sẽ được rèn luyện để tạo ra 128 đặc trưng quan trọng theo từng khuôn mặt.

+ Việc rèn luyện được thực thi bằng cách cho thuật toán làm việc với 3 hình ảnh khuôn mặt:

1. Hình ảnh của đối tượng A.

2. Hình ảnh khác cũng là đối tượng A (Nội dung của hình ảnh 1. Sẽ khác với hình ảnh 2. nhưng đối tượng trong ảnh sẽ cùng là đối tượng A).

3. Hình ảnh của đối tượng B.

+ Thuật toán sẽ nhìn vào các đặc trưng mà nó đã tạo ra cho cả 3 hình ảnh và tiếp tục điều chỉnh. Thuật toán sẽ điều chỉnh hệ thống mạng neu-ron sao cho các đặc trưng ở hình ảnh 1. sẽ gần với các đặc trưng ở hình ảnh 2. và đảm bảo rằng đặc trưng ở hình ảnh 2. sẽ khác với hình ảnh 3. nhiều hơn.

+ Lập lại việc rèn luyện này nhiều lần, thuật toán sẽ có thể tạo ra 128 đặc trưng của khuôn mặt con người một cách tốt nhất. 10 bức ảnh khác nhau của cùng 1 đối tượng sẽ cho 128 đặc trưng gần giống nhau.

Table

Description automatically generated with medium confidence

NGUỒN: <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>

**Bước 5:** Nhận diện khuôn mặt.

+ Để nhận diện chúng ta cần phải so sánh 128 đặc trưng của hình ảnh nhận vào với kho chứa các đặc trưng khác.

+ Để so sánh chúng ta sẽ sử dụng phân loại SVM (SVM classifier), có thẻ sử dụng các thuật toán phân loại khác.

+ Chúng ta cần huấn luyện một thuật toán phân loại có thể nhận vào 128 đặc trưng của hình ảnh input và đưa ra tên của người trong ảnh nếu có trong kho dữ liệu và 2 nhóm 128 đặc trưng ấy phải giống nhau nhất trong các nhóm đặc trung khác trong kho dữ liệu.

# CHƯƠNG 3: ỨNG DỤNG THỰC TẾ

## Ứng dụng máy tính sử dụng Python

- Cần có:

+ Python 3.9 trở lên.

+ OpenCV, dùng để nạp hình ảnh vào phần mềm, hoặc sử dụng webcam để làm sản phẩm thử nghiệm.

+ tkinter để xây dựng giao diện.

+ face\_detection, dùng để nhận diện xem khuôn mặt là ai (Cần có Cmake và dlib để có thể cài thư viện face\_detection).

1. from tkinter import \*
2. from tkinter import messagebox
3. from tkinter import ttk
4. from gtts import gTTS
5. import os
6. import playsound
7. import cv2
8. import face\_recognition
9. import os
10. import numpy as np
11. import threading
12. import tkinter as tk
13. from datetime import date, datetime
14. from tkinter import filedialog
15. from functools import partial
16. root = Tk()
17. frm = ttk.Frame(root,padding=100)
18. frm.grid()
20. UserLst = {"Brad":"123","Emma":"321","Robert":"robert"}
22. def validateLogin(username, password):
24. try:
25. if UserLst[username.get()] == password.get():
26. messagebox.showinfo("","Welcome " + username.get())
27. # pass
28. #
29. except:
30. messagebox.showinfo("","Please try again")
31. finally:
32. pass
34. def FacialRecognitionWithWebcam():
35. ## duyệt từng hình ảnh có trong thư mục và encode các đặc tính của nó sau đó đưa vào một list để lưu trữ
36. path = 'C:\\Git\\DoAnATTT\\rieng tu\\HinhAnh'
37. listofImg = []
38. listofName = []

41. listofImgInDir = os.listdir(path)
43. for img in listofImgInDir: #duyệt từng hình ảnh trong thư mục và lưu lại vào list hình ảnh và tên
44. currentImg =  cv2.imread(f'{path}/{img}')
45. listofImg.append(currentImg)
46. listofName.append(os.path.splitext(img)[0])
48. def Encoding(listofImg): #duyệt từng hình ảnh trong list và encode nó
49. listEncode = []
50. for img in listofImg:
51. imgRGB = cv2.cvtColor(img,cv2.COLOR\_BGR2RGB)
52. encode = face\_recognition.face\_encodings(imgRGB)[0]
53. listEncode.append(encode)
54. return listEncode
56. listKnowEncode = Encoding(listofImg)
57. ##
59. #lấy hình ảnh từ webcam
60. webcam = cv2.VideoCapture(0)
61. while True:
62. #đọc khung ảnh hiện tại từ webcam
63. place\_holder,frame = webcam.read()
65. #tạo một bản sao nhỏ của khung hình đang đọc
66. frameS = cv2.resize(frame,(0,0),None,0.25,0.25)
67. rgb\_img = cv2.cvtColor(frameS,cv2.COLOR\_BGR2RGB)
68. #chuyển tấm hình sang dạng trắng đen
69. #grayscaled\_img = cv2.cvtColor(frame,cv2.COLOR\_BGR2GRAY)
70. #chuyển bản sao nhỏ sang dạng rgb
72. #rgb\_img1 = cv2.cvtColor(frame,cv2.COLOR\_BGR2RGB)
74. #encode các đặc tính ở khuôn mặt được đưa vào
75. frameSCurrentFrame = face\_recognition.face\_locations(rgb\_img)
76. encodingCurrentFrame = face\_recognition.face\_encodings(rgb\_img,frameSCurrentFrame)
78. #so sánh khuôn mặt trong webcam với các khuôn mặt có sẵn
79. for encodingCurrent, faceCorCurrent in zip(encodingCurrentFrame,frameSCurrentFrame):
80. cv2.rectangle(frame,(faceCorCurrent[3]\*4,faceCorCurrent[0]\*4),(faceCorCurrent[1]\*4,faceCorCurrent[2]\*4),(255,0,255),2)
81. matches = face\_recognition.compare\_faces(listKnowEncode,encodingCurrent) #so sánh
82. faceDis = face\_recognition.face\_distance(listKnowEncode,encodingCurrent) #tính toán sự khác biệt giữa bức ảnh đc đưa vào và ảnh trong thư mục
83. matchIndex = np.argmin(faceDis)  #phải lấy khuôn mặt có độ lệch nhỏ nhất (gần giống với khuôn mặt trong thư mục nhất)
84. if matches[matchIndex]:
85. #cv2.destroyAllWindows()
86. Name = listofName[matchIndex]
87. messagebox.showinfo("","Welcome " + Name)
88. # x,y,w,h =  faceCorCurrent
89. # x,y,w,h = x\*4,y\*4,w\*4,h\*4
91. # cv2.putText(frame,Name,(h+6,w-6),cv2.FONT\_ITALIC,1,(0,255,0),2)
93. #xem hình ảnh
94. cv2.imshow('Face Detection',frame)
95. key = cv2.waitKey(1) #tạm dừng chương trình
96. if key == 27: #nhấn ESC để dừng chương trình
97. break
98. usernameLabel = Label(root, text="User Name").grid(row=0, column=0)
99. username = StringVar()
100. usernameEntry = Entry(root, textvariable=username).grid(row=0, column=1)
102. #password label and password entry box
103. passwordLabel = Label(root,text="Password").grid(row=1, column=0)
104. password = StringVar()
105. passwordEntry = Entry(root, textvariable=password, show='\*').grid(row=1, column=1)
107. # ttk.Button(frm,text="Đăng nhập khác",command= FacialRecognitionWithWebcam).grid(column=0,row =0)
108. # ttk.Button(frm,text="quit",command=root.destroy).grid(column=1,row =0)
109. validateLogin = partial(validateLogin, username, password)
111. #login button
112. loginButton = Button(root, text="Login", command=validateLogin).grid(row=4, column=0)
113. loginButtonFace = Button(root,text="FaceId",command=FacialRecognitionWithWebcam).grid(row=4,column=1)
114. '''Label(root,text = "Đăng nhập").pack()
116. def myClick():
117. myLabel = Label(root, text = "Hello " + e.get())
118. myLabel.pack()
119. myButton = Button(root, command = myClick)
120. myButton.pack()
121. '''
123. root.mainloop()
124. # %%

## Ứng dụng điện thoại sử dụng Python

- Cần có:

+ Python 3.9 trở lên.

+ OpenCV, dùng để nạp hình ảnh vào phần mềm, hoặc sử dụng webcam để làm sản phẩm thử nghiệm.

+ kivy xây dựng giao diện cho android, hỗ trợ xây dựng file .apk.

+ face\_detection, dùng để nhận diện xem khuôn mặt là ai (Cần có Cmake và dlib để có thể cài thư viện face\_detection).

1. import face\_recognition
3. from kivymd.app import MDApp
4. from kivy.clock import Clock
5. from kivymd.uix.boxlayout import MDBoxLayout
6. # Import kivy UX components
7. from kivy.uix.image import AsyncImage, Image
8. from kivymd.uix.button import MDRaisedButton
9. #from kivy.uix.image import Image
10. # Import other kivy stuff
11. from kivy.graphics.texture import Texture
12. # Import other dependencies
13. import cv2
14. import os
15. import numpy as np
17. def FacialRecognitionWithWebcam(frame):
18. ## duyệt từng hình ảnh có trong thư mục và encode các đặc tính của nó sau đó đưa vào một list để lưu trữ
19. path = 'assets'
20. listofImg = []
21. listofName = []
23. listofImgInDir = os.listdir(path)
25. for img in listofImgInDir: #duyệt từng hình ảnh trong thư mục và lưu lại vào list hình ảnh và tên
26. currentImg =  cv2.imread(f'{path}/{img}')
27. listofImg.append(currentImg)
28. listofName.append(os.path.splitext(img)[0])
30. def Encoding(listofImg): #duyệt từng hình ảnh trong list và encode nó
31. listEncode = []
32. for img in listofImg:
33. imgRGB = cv2.cvtColor(img,cv2.COLOR\_BGR2RGB)
34. encode = face\_recognition.face\_encodings(imgRGB)[0]
35. listEncode.append(encode)
36. return listEncode
38. listKnowEncode = Encoding(listofImg)
39. ##
41. #lấy hình ảnh từ webcam


45. frameS = cv2.resize(frame,(0,0),None,0.25,0.25)
46. rgb\_img = cv2.cvtColor(frameS,cv2.COLOR\_BGR2RGB)
48. frameSCurrentFrame = face\_recognition.face\_locations(rgb\_img)
49. encodingCurrentFrame = face\_recognition.face\_encodings(rgb\_img,frameSCurrentFrame)
51. #so sánh khuôn mặt trong webcam với các khuôn mặt có sẵn
52. for encodingCurrent, faceCorCurrent in zip(encodingCurrentFrame,frameSCurrentFrame):
53. cv2.rectangle(frame,(faceCorCurrent[3]\*4,faceCorCurrent[0]\*4),(faceCorCurrent[1]\*4,faceCorCurrent[2]\*4),(255,0,255),2)
54. matches = face\_recognition.compare\_faces(listKnowEncode,encodingCurrent) #so sánh
55. faceDis = face\_recognition.face\_distance(listKnowEncode,encodingCurrent) #tính toán sự khác biệt giữa bức ảnh đc đưa vào và ảnh trong thư mục
56. matchIndex = np.argmin(faceDis)  #phải lấy khuôn mặt có độ lệch nhỏ nhất (gần giống với khuôn mặt trong thư mục nhất)
57. if matches[matchIndex]:
58. x,y,w,h =  faceCorCurrent
59. x,y,w,h = x\*4,y\*4,w\*4,h\*4
60. Name = listofName[matchIndex].upper()
61. cv2.putText(frame,Name,(h+6,w-6),cv2.FONT\_ITALIC,1,(0,255,0),2)
63. return frame
64. #xem hình ảnh
65. # cv2.imshow('Face Detection',frame)

68. class TheLabApp(MDApp):
70. def build(self):
71. layout = MDBoxLayout(orientation='vertical')
72. self.image = Image()
73. layout.add\_widget(self.image)
74. self.captureBtn = MDRaisedButton(
75. text="Capture",
76. pos\_hint={'center\_x': .5, 'center\_y': .5},
77. size\_hint=(None, None))
78. self.captureBtn.bind(on\_press=self.take\_picture)
79. layout.add\_widget(self.captureBtn)
80. self.displayImage = Image()
81. layout.add\_widget(self.displayImage)
82. self.capture = cv2.VideoCapture(1)
83. Clock.schedule\_interval(self.load\_video, 1.0/30.0)
84. return layout


88. def load\_video(self, \*args):
89. ret, frame = self.capture.read()
90. # Frame initialize
91. self.image\_frame = frame
92. buffer = cv2.flip(frame, 0).tostring()
93. texture = Texture.create(size=(frame.shape[1], frame.shape[0]), colorfmt='bgr')
94. texture.blit\_buffer(buffer, colorfmt='bgr', bufferfmt='ubyte')
95. self.image.texture = texture
97. def take\_picture(self,\*args):
98. frame = FacialRecognitionWithWebcam(self.image\_frame)
99. buffer = cv2.flip(frame, 0).tostring()
100. texture = Texture.create(size=(frame.shape[1], frame.shape[0]), colorfmt='bgr')
101. texture.blit\_buffer(buffer, colorfmt='bgr', bufferfmt='ubyte')
102. self.displayImage.texture = texture
103. if \_\_name\_\_ == '\_\_main\_\_':
104. TheLabApp().run()

## Ứng dụng máy tính sử dụng C#

- Cần thử viện:

+ EmguCV để xử lý việc mở webcam và nhận diện khuôn mặt.

# TÀI LIỆU THAM KHẢO

1. Nguyễn Đình Thắng (12/2018), *THIẾT KẾ HỆ THỐNG NHẬN DẠNG KHUÔN MẶT TRÊN KIT RASPBERRY PY* : <https://123docz.net/document/5222186-nhan-dang-khuon-mat.htm>
2. Corinne Bernstein (2/2020), *Face detection* : <https://searchenterpriseai.techtarget.com/definition/face-detection>
3. Adam Geitgey (24/7/2016), *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning* : <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>
4. Hải Hà, *Tìm hiểu về phương pháp mô tả đặc trưng HOG (Histogram of Oriented Gradients)* : [*https://viblo.asia/p/tim-hieu-ve-phuong-phap-mo-ta-dac-trung-hog-histogram-of-oriented-gradients-V3m5WAwxZO7*](https://viblo.asia/p/tim-hieu-ve-phuong-phap-mo-ta-dac-trung-hog-histogram-of-oriented-gradients-V3m5WAwxZO7)
5. *Sinh trắc học* : <https://vi.wikipedia.org/wiki/Sinh_tr%E1%BA%AFc_h%E1%BB%8Dc>
6. *Xác thực sinh trắc học là gì? Các loại sinh trắc học phổ biến* : <https://cyberlotus.com/xac-thuc-sinh-trac-hoc-la-gi-cac-loai-sinh-trac-hoc-pho-bien.html>