**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**----🙣🕮🙡----**



**ĐỒ ÁN 3**

**ĐỀ TÀI:**

**NHẬN DIỆN MẶT NGƯỜI**

**GVHD : TS. Nguyễn Thiên Bảo**

**SVTH : Đoàn Văn Long 17110174**

**Võ Tấn Nguyên 17110194**

**Tp. Hồ Chí Minh, Tháng 1 năm 2021**

**LỜI CẢM ƠN**

Không một ai có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Nay nhóm tụi em xin được phép gửi lời cảm ơn chân thành này đến thầy Nguyễn Thiên Bảo, người đã trực tiếp hướng dẫn và hỗ trợ nhóm em trong suốt quá trình định hướng chọn đề tài, hướng dẫn, nhận xét và góp ý. Nếu không có những lời hướng dẫn, những kinh nghiệm thực tiễn của thầy thì em nghĩ rằng đồ án này sẽ khó có thể hoàn thiện và hoàn thành đúng thời hạn được. Nhóm tụi em xin cám ơn thầy một lần nữa.

Khoảng thời gian có hạn, cùng với kiến thức còn hạn chế và còn nhiều bỡ ngỡ khác do đó thiếu sót là điều không thể tránh khỏi nên em rất mong nhận được những ý kiến đóng góp quý báo từ các quý Thầy Cô để kiến thức của em được hoàn thiện hơn sau này. Em xin chân thành cảm ơn.

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

...........................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

TP.HCM, ngày…. tháng….. năm 20…..

Giáo viên hướng dẫn

*(Ký & ghi rõ họ tên)*

Mục lục

[**Lý do chọn đề tài**](#_tyjcwt) **5**

[**Mục tiêu đề tài**](#_3dy6vkm) **5**

[**Nội dung**](#_1t3h5sf) **5**

[**Phương pháp nghiên cứu**](#_4d34og8) **5**

[**Ý nghĩa khoa học thực tiễn**](#_17dp8vu) **5**

[**1. Python**](#_26in1rg) **6**

[1.1. Lịch sử](#_35nkun2) 6

[1.2. Giới thiệu:](#_z337ya) 6

[1.3. Tính năng của Python](#_qsh70q) 6

[1.4 Các lĩnh vực áp dụng](#_1ep9kbs34ouu) 8

[1.1.4.1 Phát triển web và internet (Web and internet development)](#_7gmeyfurizyo) 8

[1.1.4.2 Games và Đồ Họa 3D (Games and 3D Graphics)](#_guawdhgx3w1e) 8

[1.1.4.3 Phục vụ cho khoa học và tính toán (Scientific and numeric)](#_yn8ww2ofc4hz) 8

[1.1.4.4 Ứng dụng giao diện người dùng (Desktop GUIs)](#_aek3nnq1ok2r) 9

[1.1.4.5 Sử dụng trong lĩnh vực Trí tuệ nhân tạo (Artificial Intelligence)](#_9mcobr3qn1mw) 9

[**2. OpenCV**](#_xzyoefu8rcg4) **9**

[2.1 Giới thiệu](#_cq4tnjm545ny) 9

[2.2 Các ứng dụng](#_1790ak5slora) 10

[2.3 Chức năng](#_jg5e43ak9gmd) 10

[**3. Bài toán nhận diện mặt người và khó khăn**](#_a5z94ios1rb) **10**

[3.1 Bài toán nhận diện mặt người](#_8xd8wdkemq48) 10

[3.2 Những khó khăn của hệ thống nhận diện khuôn mặt](#_iy6l0ri7q3yv) 12

[**4. Haar Cascade**](#_e8ilq6jbk0r8) **13**

[4.1 Haar Cascade là gì](#_chvpbv3pwk9) 13

[4.2 Đặc trưng Haar](#_crhkk24wqvlf) 13

[4.3 Làm sao mà biết được bộ lọc Haar nào tốt? Nhìn cái nào cũng như nhau…](#_yx7opvxxzqfb) 15

[**5. LBPHFaceRecognizer**](#_5y0qssml6d1c) **16**

[**Lấy dữ liệu từ camera**](#_8wfkplxe83k5) **20**

[**Lấy dữ liệu từ database**](#_375fbgg) **21**

[**Train dữ liệu cho máy học**](#_c5agazl0srig) **23**

[**Nhận diện khuôn mặt**](#_dbzynsvcyzuj) **24**

[**1. Tổng kết**](#_1n1mu2y) **27**

[1.1. Về mặt lý thuyết](#_471acqr) 27

[Tìm hiểu được cách thức hoạt động của nhận diện khuôn mặt, các thư viện liên quan như OpenCV](#_2m6kmyk) 27

[1.2. Về mặt hiện thực](#_11bux6d) 27

[**2. Hạn chế**](#_3lbifu6) **27**

[**3. Hướng phát triển đề tài:**](#_20gsq1z) **27**

**CHƯƠNG I: MỞ ĐẦU**

## ***Lý do chọn đề tài***

Ngày nay, công nghệ thông tin không ngừng phát triển một cách mạnh mẽ và hiện đại. Các ứng dụng của công nghệ thông tin ngày càng đi sâu vào đời sống con người trở thành một bộ phận không thể thiếu của thế giới văn minh. Với xu thế 4.0 thì hiện nay AI hay lớn hơn là Machine Learning được áp dụng rộng rãi trong cuộc sống, ngay cả chiếc điện thoại smartphone bây giờ cũng có nhiều tính năng AI trong đó ( nhận diện khuôn mặc, giọng nói, camera AI, ...) nó giúp con người làm mọi việc dễ dàng hơn, tiết kiệm thời gian và cuộc sống ngày càng hiện đại. Vì thế mà nhóm tụi em chọn đề tài nhận diện khuôn mặt.

## ***Mục tiêu đề tài***

* Xây dựng chức năng nhận diện khuôn mặt.
* Tìm hiểu thư viện nhận diện khuôn mặt. (Opencv, ....)

## ***Nội dung***

Tìm hiểu lý thuyết về các công nghệ, thuật toán áp dụng để thực hiện đề tài

* Python
* Thư viện OpenCV

## ***Phương pháp nghiên cứu***

Phương pháp tổng hợp lý thuyết:

* Nghiên cứu và tìm hiểu các tài liệu, các bài hướng dẫn về nhận diện khuôn mặt.

Phương pháp chuyên gia:

* Tham khảo ý kiến của Thầy hướng dẫn cùng các anh, chị và bạn bè có kinh nghiệm liên quan đến các vấn đề của khóa luận để tạo tính chính xác.

Phương pháp mô hình hóa:

* Mô phỏng chức năng nhận diện khuôn mặt

## ***Ý nghĩa khoa học thực tiễn***

Website có giao diện trực quan, dễ sử dụng, dễ dàng thao tác cho người dùng.

**CHƯƠNG II: CƠ SỞ LÝ THUYẾT**

## **1. *Python***

### **1.1. Lịch sử**

Python là ngôn ngữ lập trình cấp cao, đa mục đích được sử dụng rộng rãi. Ban đầu được thiết kế bởi Guido van Rossum vào năm 1991 và được duy trì, phát triển bởi tổ chức Python Software Foundation.

Guido Van Rossum bắt đầu nghiên cứu các sản phẩm dựa trên ứng dụng của mình vào tháng 12 năm 1989 ở Centrum Wiskunde & Informatica (CWI). Tại đây Guido van Rossum có cơ hội được tiếp xúc với ngôn ngữ lập trình ABC, trong quá trình làm việc ông đã nhận thấy ABC có rất nhiều vấn đề , tuy nhiên ông vẫn ưa chuộng những tính năng mà ABC mang lại.

Bằng sự sáng tạo và sự trợ giúp từ các cộng sự, ông đã kết hợp cú pháp của ngôn ngữ ABC và một số tính năng của nó để tạo ra ngôn ngữ lập trình mới. Cái tên được lấy cảm hứng từ một chương trình truyền hình BBC – Monty Python, Flying Circus Circus mà ông rất hâm mộ. Thêm vào đó, với ý muốn đặt một cái tên ngắn, độc đáo và hơi bí ẩn cho phát minh của mình, Guido đặt tên cho ngôn ngữ này là Python!

### **1.2. Giới thiệu:**

Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level) ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.

### **1.3. Tính năng của Python**

* Tính phổ Biến

Python có sự phát triển rất nhanh trong những năm gần đây. Theo một khảo sát vào năm 2018 trên trang web Stack Overflow, Python được xếp hạng thứ 7 trong những ngôn ngữ lập trình phổ biến nhất và xếp hạng thứ nhất trong những công nghệ được ưa thích nhất năm. Không những vậy, những công ty phần mềm hàng đầu thế giới sử dụng Python trong các dự án của họ mỗi ngày.

* Dễ học

Khi so sánh giữ Python và các ngôn ngữ lập trình cấp cao khác như C++, C#, Javascript, Java,.. thì việc học ngôn ngữ Python dễ hơn. Hơn thế, Python còn được dạy cho trẻ nhỏ nhờ vào cú pháp đơn giản dễ hiểu và gần với ngôn ngữ giao tiếp thường ngày. Chính vì vậy Python đã được nhiều trường học trên thế giới đem vào giảng dạy ở bậc Tiểu học đến Trung học, nhờ thế học sinh có thể dễ dàng tiếp thu và khơi dậy niềm đam mê lập trình trong các em.

* Không tính phí và mã nguồn mở

Trình thông dịch của Python được phát triển dưới giấy phép mã nguồn mở ( OSI-approved open-source license ), chính vì vậy ta có thể tải và cài đặt nó hoàn toàn miễn phí, kể cả dùng cho các mục đích thương mại khác.

* Ngôn ngữ hướng đối tượng

Hướng đối tượng là đặc trưng quan trọng của ngôn ngữ Python và các ngôn ngữ bậc cao hiện nay. Phương pháp lập trình hướng đối tượng mô tả bài toán dưới sự tác động giữa các đối tượng và hiện thực vào chương trình máy tính. Các đặc trưng quan trọng của một ngôn ngữ hướng đối tượng là: tính trừu tượng hóa (Abstraction), tính đóng gói (Encapsulation), tính kế thừa (Inheritance), tính đa hình (Polymorphism).

* Hỗ trợ lập trình giao diện người dùng

Ngôn ngữ Python có các modules hỗ trợ tạo giao diện trong lập trình như PyQt5, PyQt4, Tkinter giúp cho chương trình thu hút người dùng và trực quan hơn.

* Ngôn ngữ thông dịch

Hầu hết các ngôn ngữ lập trình đều chạy bằng trình biên dịch, đều này có nghĩa là source code mà bạn viết ra cần phải được dịch qua mã máy (machine code) trước khi chạy được chương trình. Tuy nhiên, những chương trình được viết bằng ngôn ngữ thông dịch sẽ bỏ qua bước đó mà chạy thẳng qua trình thông dịch

Điều này giúp ích rất nhiều trong quy trình phát triển chương trình của bạn, bời vì bạn chỉ cần gõ code và chạy thôi, chứ không cần phải qua một quá trình biên dịch nào cả.

* Dễ dàng mở rộng

Tính năng mở rộng của Python đó là Python có để được viết trong một ngôn ngữ khác và có thể dùng trình biên dịch/thông dịch để thực thi chương trình. Ví dụ, chúng ta có thể viết mã nguồn Python và biên dịch nó bên trong ngôn ngữ C/C++.

* Ngôn ngữ không cần định nghĩa kiểu dữ liệu

Đây là một đặc trưng khác biệt hoàn toàn so với các ngôn ngữ lập trình C, C++, Java,… Các biến trong chương trình Python khi khai báo không cần chỉ rõ kiểu dữ liệu cho nó và một biến có thể nắm giữ các đối tượng có kiểu dữ liệu khác nhau.

### **1.4 Các lĩnh vực áp dụng**

##### **1.1.4.1 Phát triển web và internet (Web and internet development)**

Python cung cấp cho người dùng chúng ta rất nhiều công cụ phục vụ cho việc phát triển web, điển hình có thể kể đến như:

Frameworks nổi bật như Django và Pyramid.

Các thư viện như Flask và Bottle.

Các hệ thống quản trị nội dung như Plone và django CMS.

##### **1.1.4.2 Games và Đồ Họa 3D (Games and 3D Graphics)**

Python cũng được sử dụng trong việc phát triển các trò chơi . Có các thư viện ví dụ PySoy là engine trò chơi 3D hỗ trợ cho Python 3, PyGame cung cấp chức năng và thư viện để phát triển trò chơi. Các trò chơi như Civilization-IV, Disney’s Toontown Online, Vega Strike, v.v. đã được xây dựng bằng Python.

##### **1.1.4.3 Phục vụ cho khoa học và tính toán (Scientific and numeric)**

Việc triển khai các thuật toán học máy (Machine Learning) yêu cầu tính toán toán học vô cùng phức tạp. Rất may Python cung cấp cho chúng ta rất nhiều thư viện cho khoa học và số liệu như Numpy, Pandas, Scipy, Scikit-learning, v.v. Dưới đây là một vài thư viện phổ biến trong lĩnh vực này.

* SciPy
* Scikit-learn
* NumPy
* Pandas
* Matplotlib

##### **1.1.4.4 Ứng dụng giao diện người dùng (Desktop GUIs)**

Chúng ta có thể sử dụng Python để lập trình các ứng dụng giao diện. Và điển hình nhất chính là thư viện Tkinter. Ngoài ra một số bộ thư viện hữu ích khác như wxWidgets, Kivy, PYQT, v.v

##### **1.1.4.5 Sử dụng trong lĩnh vực Trí tuệ nhân tạo (Artificial Intelligence)**

Triển khai các dự án về AI liên quan đến hàng tấn thuật toán. Để tiết kiệm thời gian và công sức,chúng ta không cần phải code lại các thuật toán cũng như để cho mọi thứ dễ dàng hơn, Python cung cấp cho chúng ta hơn 100 thư viện được xây dựng sẵn để thực hiện các thuật toán khác nhau.

Vì vậy, mỗi khi bạn muốn chạy một thuật toán trên một tập dữ liệu, tất cả những gì bạn phải làm là cài đặt và tải các thư viện cần thiết bằng một lệnh duy nhất. Ví dụ về các thư viện nổi bật trong lĩnh vực này như NumPy, Keras, Tensorflow, Pytorch, v.v.

## **2. OpenCV**

### **2.1 Giới thiệu**

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot.

### **2.2 Các ứng dụng**

OpenCV đang được sử dụng rộng rãi trong các ứng dụng bao gồm:

Hình ảnh street view Kiểm tra và giám sát tự động Robot và xe hơi tự lái Phân tích hình ảnh y tế Tìm kiếm và phục hồi hình ảnh/video Phim - cấu trúc 3D từ chuyển động Nghệ thuật sắp đặt tương tác

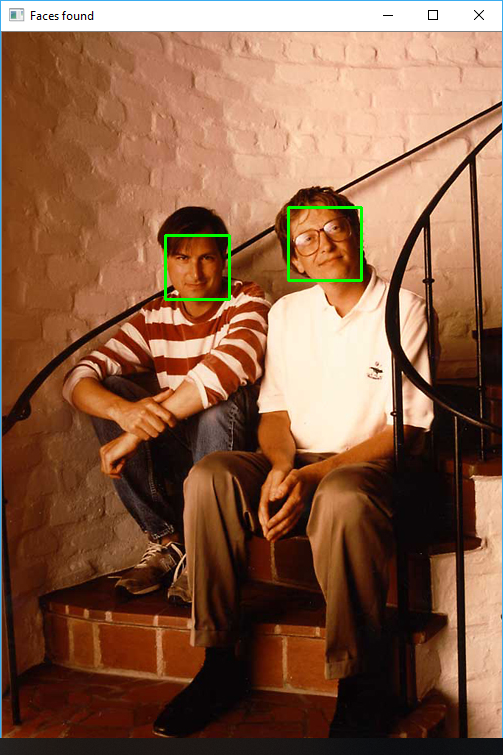
### **2.3 Chức năng**

Image/video I/O, xử lý, hiển thị (core, imgproc, highgui) Phát hiện các vật thể (objdetect, features2d, nonfree) Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab) Computational photography (photo, video, superres) Machine learning & clustering (ml, flann) CUDA acceleration (gpu)

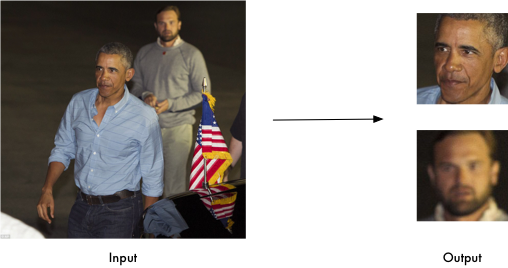
## **3. Bài toán nhận diện mặt người và khó khăn**

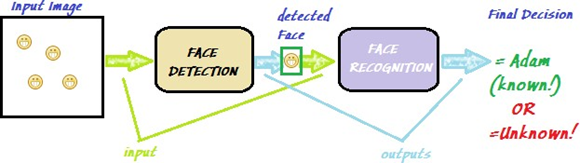
### **3.1 Bài toán nhận diện mặt người**

Hệ thống nhận dạng mặt người là một hệ thống nhận vào là một ảnh hoặc một đoạn video (một dòng các hình ảnh liên tục). Qua xử lý, tính toán hệ thống xác định được vị trí mặt người (nếu có) trong ảnh và xác định là người nào trong số những người mà hệ thống đã được biết (qua quá trình học) hoặc là người lạ.



* Máy tính nhận dạng được Bill Gates và Steve Jobs





Hệ thống nhận dạng mặt người

### **3.2 Những khó khăn của hệ thống nhận diện khuôn mặt**

Bài toán nhận dạng mặt người là bài toán đã được nghiên cứu từ những năm 70. Tuy nhiên, đây là một bài toán khó nên những nghiên cứu hiện tại vẫn chưa đạt được những kết quả mong muốn. Chính vì thế, vấn đề này vẫn đang được nhiều nhóm trên thế giới quan tâm nghiên cứu. Khó khăn của bài toán nhận dạng mặt người có thể kể đến như sau:

a. Tư thế chụp, 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 45o hay chụp chéo bên phải 45o, chụp từ trên xuống, chụp từ dưới lên, v.v… 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.

b. Sự xuất hiện hoặc thiếu một số thành phần của khuôn mặt: Các đặc trưng như: râu mép, râu hàm, mắt kính, v.v… có thể xuất hiện hoặc không. Vấn đề này làm cho bài toán càng trở nên khó hơn rất nhiều.

c. Sự biểu cảm của khuôn mặt: Biểu cảm của khuôn mặt con người 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, v.v…

d. Sự che khuất: Khuôn mặt có thể bị che khuất bởi các đối tượng khác hoặc các khuôn mặt khác.

e. Hướng của ảnh (pose variations): Các ảnh 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.

f. Điều kiện của ảnh: Ảnh được chụp trong các điều kiện khác nhau về: chiếu sáng, về tính chất camera (máy kỹ thuật số, máy hồng ngoại, v.v…), ảnh có chất lượng thấp ảnh hưởng rất nhiều đến chất lượng ảnh khuôn mặt.

g. Aging condition: Việc nhận dạng ảnh mặt thay đổi theo thời gian còn là một vấn đề khó khăn, ngay cả đối với khả năng nhận dạng của con người.

h. Các hệ thống cực lớn (very large scale systems): Các CSDL ảnh mặt được test bởi các nhà nghiên cứu còn khá nhỏ (vài trăm tới vài chục nghìn ảnh mặt), tuy nhiên trên thực tế các CSDL có thể rất lớn, ví dụ CSDL ảnh mặt của cảnh sát của một đất nước có thể chứa từ hàng triệu tới hơn 1 tỉ ảnh…

Tổng quan kiến trúc của một hệ thống nhận dạng mặt người.

Một hệ thống nhận dạng mặt người thông thường bao gồm bốn bước xử lý sau:

1. Phát hiện khuôn mặt (Face Detection).

2. Phân đoạn khuôn mặt (Face Alignment hay Segmentation).

3. Trích chọn đặc trưng (Feature Extraction).

4. Nhận dạng (Recognition) hay Phân lớp khuôn mặt (Face Classification).

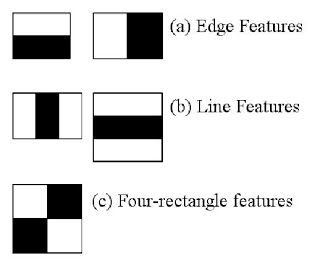
## **4. Haar Cascade**

### **4.1 Haar Cascade là gì**

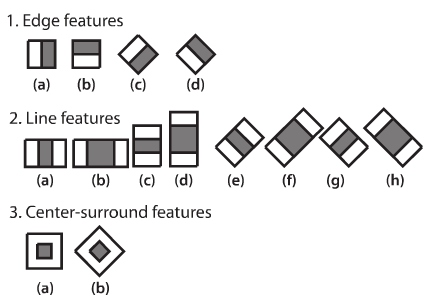
Về cơ bản là sử dụng các đặc trưng loại *Haar* và sau đó sử dụng thật nhiều đặc trưng đó qua nhiều lượt (*cascade*) để tạo thành một cỗ máy nhận diện hoàn chỉnh. Vẫn khó hiểu phải không? Vậy chúng ta nhảy vào từng khái niệm một nhé.

### **4.2 Đặc trưng Haar**

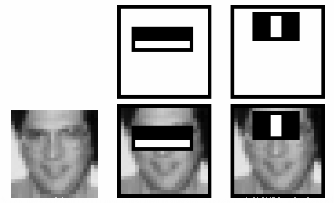
Nếu bạn đã làm việc với xử lý ảnh hoặc Convolutional Neural Networks rồi thì chắc bạn cũng không còn lạ gì với các bộ lọc trong xử lý ảnh nữa cả — nếu không, bạn có thể đọc phần này trong một bài của mình về CNN để hiểu thêm. Các ví dụ bộ lọc được liệt kê ở dưới, trong đó a) là các bộ lọc bắt các cạnh trong ảnh, và b) bắt các đường thẳng trong ảnh, tương tự như các bộ lọc đã được mình nhắc tới trong bài trên. Ngoài ra, còn có các bộ lọc Haar khác, như ví dụ c) về đặc trưng 4-hình vuông dưới đây,



hoặc đặc trưng nằm gọn trong trung tâm một vùng như ví dụ *3.* trong ảnh dưới đây:



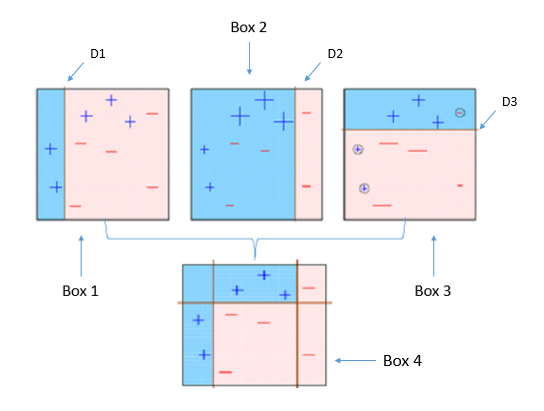
Tuy nhiên, cách áp dụng các bộ lọc này khác một chút so với các cửa sổ bộ lọc bên CNN. Ở CNN, bộ lọc chiếm toàn bộ cửa sổ trượt, trong khi ở đặc trưng Haar, bộ lọc chỉ chiếm một phần trong cửa sổ trượt thôi. Điều đó được minh hoạ trên ảnh sau:



Trong hình trên, cửa sổ trượt được đặt ngay ngắn vừa gọn để nhìn được toàn bộ ảnh. Các bạn có thể nhận ra rằng bộ lọc đầu trong đó đang tìm một "cạnh" phân cách giữa mắt/lông mày với mũi, vì ở đoạn đọc có chênh lệch về màu đáng kể; và ở bộ lọc sau, mô hình đang tìm đường sống mũi, vì ở đó sẽ có màu sáng hơn so với 2 bên (vì nó cao hơn dễ bắt sáng). Và như đã nói trên, bộ lọc Haar chỉ nhìn cụ thể vào một vùng trong cửa sổ để tìm thôi: trong khuôn mặt thì mũi lúc nào cũng ở chính giữa chứ không ở các góc, nên không cần nhìn các góc để làm gì cả.

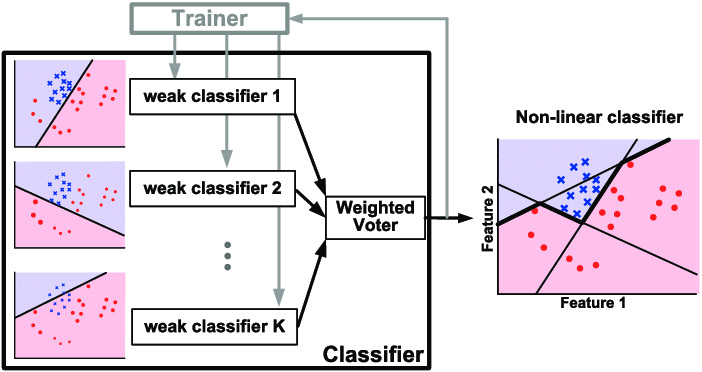
### **4.3 Làm sao mà biết được bộ lọc Haar nào tốt? Nhìn cái nào cũng như nhau…**

Chúng ta có tầm 160k+ các bộ lọc như vậy cơ! Tuy nhiên, chúng ta có thể sử dụng *Adaboost (adaptive boosting)* để kết hợp các bộ lọc trên như sau:



Ảnh trên là minh hoạ của quá trình boosting: khi bạn có các classifier yếu khác nhau, kết hợp chúng để tạo thành một classifier mạnh. Việc kết hợp như các bạn thấy khá đơn giản, chỉ là các khối logic AND/OR phụ thuộc vào kết quả được đưa ra. Câu trả lời là một lớp perceptron đơn giản! Một tổ hợp tuyến tính đơn giản đã có thể tính được kết hợp các đầu ra quyết định trên, bạn có thể tìm hiểu thêm ở mục ngoài lề này trong bài về linear classifier mình đã viết.

Vậy đó là boosting rồi, còn adaptive ở đây là gì? Ở boosting thường thì các classifier yếu trên có tiếng nói ngang nhau, bình đẳng và dân chủ; nhưng sau khi qua Adaboost, nhưng classifier khôn hơn thì tiếng nói sẽ có trọng lượng cao hơn. Sau đây là một hình ảnh minh hoạ khác để giúp bạn dễ tưởng tượng hơn chút:



Kết quả là 2 bộ lọc ở mục trên là những bộ lọc tốt nhất theo như Adaboost

## 5. LBPHFaceRecognizer

**L**ocal **B**inary **P**atterns (hay còn viết tắt là LBP) là một phương pháp rút trích đặc trưng trong xử lý ảnh. Đặc trưng được rút trích sẽ tiếp tục được tiến hành chọn lọc (feature selection) thu gọn thành vector đặc trưng. Vector đặc trưng này sau đó có thể dùng để đưa vào mô hình học máy để học / phân loại.

Các bước tiến hành trích xuất đặc trưng theo phương pháp LBP:

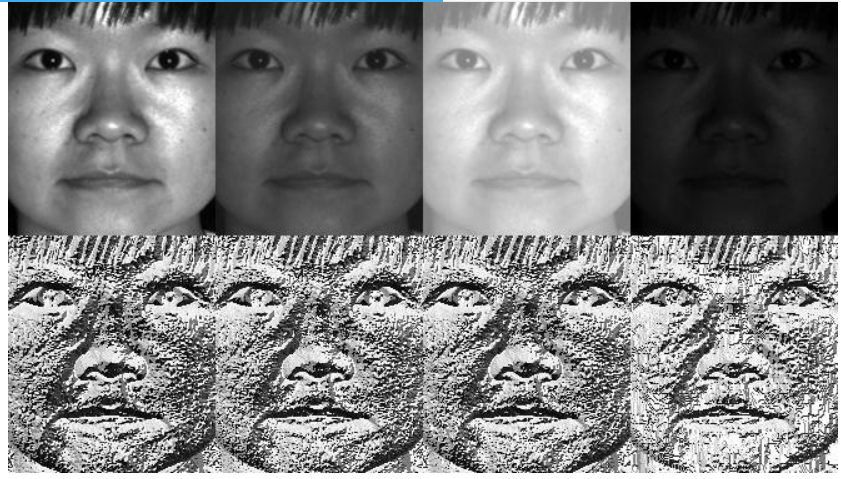
1. Duyệt lần lượt từng pixel trên ảnh (theo cột -> theo hàng), với pixel đang xét, ta áp dụng bước 2->4.
2. Xét lần lượt 8 pixel lân cận (hàng xóm - neighbor) của pixel đang duyệt (trung tâm - center). Mỗi pixel hàng xóm sẽ ứng với một bit trong một chuỗi 8-bit. Chuỗi 8-bit này ban đầu sẽ bằng: 00000000. Chuỗi 8-bit này sẽ được cập nhật theo mô tả ở bước 3.
3. Nếu mức sáng tại pixel hàng xóm >= mức sáng tại pixel trung tâm: vote bit ở vị trí tương ứng lên 1 trong chuỗi 8-bit đề cập ở bước 2.
4. Sau khi hoàn tất bước 2 và 3, ta sẽ có mội chuỗi 8-bit (vd: 00101100) -> đổi giá trị nhị phân này sang thập phân để lưu trữ (vd: 00101100 nhị phân = 44 thập phân)
5. Lặp hết toàn ảnh (bước 1->4), ta sẽ có kết quả đầu ra bằng kích thước với ảnh đầu vào. Mỗi giá trị trên ảnh đầu ra là đặc trưng LBP.

LBP ta có thể tạm hiểu (để nhớ như sau):

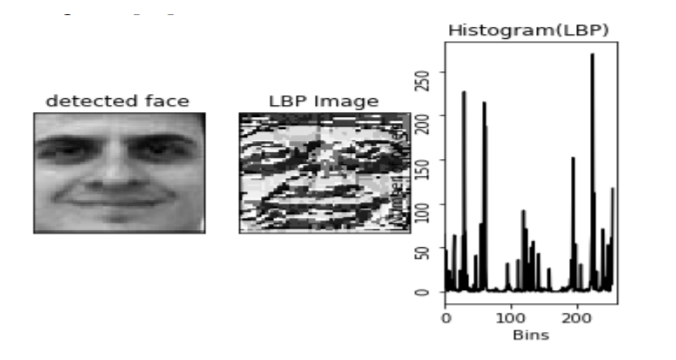
* Local: thể hiện tính chất cục bộ địa phương, đó là khi ở bước 2 ta xét pixel lân cận -> mỗi đặc trưng trong output sẽ mang đặc trưng đại diện cục bộ.
* Binary Patterns: các mẫu hình nhị phân -> cách nhị phân hóa mô tả ở bước 3, 4

Tổng quát hóa phương pháp tiếp cận LBP trên, ta sẽ có các tham số sau:

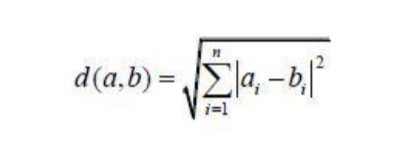
* P: Số pixel lận cận pixel trung tâm (vd: P=8).
* R: Bán kính của pixel lân cận mà ta sẽ xét - cách pixel trung tâm bao nhiêu pixel (vd: R=1 nghĩa là liền kề).
* Thứ tự các pixel lân cận mã hóa vào chuỗi 8-bit sẽ theo chiều kim đồng hồ hay ngược chiều kim đồng hồ.
* Interpolation: do lấy pixel lân cận theo hình tròn, do đó tọa độ của các pixel lân cận khi tính toán ra sẽ là số thực -> quyết định lấy ra giá trị mức sáng của pixel đó theo cách nào: pixel gần nhất (nearest) hay có trọng số (bilinear).
* Toán tử LBP hỗ trợ mạnh mẽ các phép biến đổi thang màu xám đơn điệu



* Sau khi tạo biểu đồ giá trị LBP của vùng được tạo bằng cách đếm số lượng giá trị LBP tương tự trong vùng.
* Sau khi tạo biểu đồ cho mỗi vùng, tất cả các biểu đồ được hợp nhất để tạo thành một biểu đồ duy nhất và đây được gọi là vectơ đặc trưng của hình ảnh.



* Bây giờ so sánh biểu đồ của hình ảnh thử nghiệm và các hình ảnh trong cơ sở dữ liệu và sau đó chúng tôi trả về hình ảnh với biểu đồ gần nhất.  
  (Điều này có thể được thực hiện bằng nhiều kỹ thuật như khoảng cách euclid, chi-square, giá trị tuyệt đối, v.v.)
* Khoảng cách Euclid được tính bằng cách so sánh các đặc điểm của hình ảnh thử nghiệm với các đặc điểm được lưu trữ trong bộ dữ liệu. Khoảng cách tối thiểu giữa hình ảnh thử nghiệm và hình ảnh gốc cho tỷ lệ khớp.



* Như một đầu ra, chúng tôi nhận được một ID của hình ảnh từ cơ sở dữ liệu nếu hình ảnh thử nghiệm được nhận dạng.

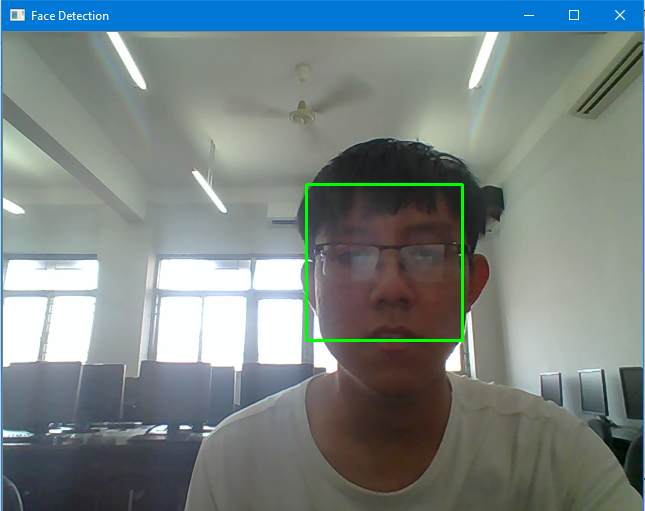


**CHƯƠNG III: XÂY DỰNG HỆ THỐNG NHẬN DIỆN KHUÔN MẶT**

## Lấy dữ liệu từ camera

|  |
| --- |
| import cv2 # Import thư viện OpenCV  # Tạo bộ nhận diện khuôn mặt  faceCascade = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")  cap = cv2.VideoCapture(0)  while True:  # Chụp lại từng khung hình  ret, frame = cap.read()  # Quá trình nhận diện sẽ được thực hiện trên ảnh xám (Đen/Trắng)  gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) # Chuyển ảnh màu sang ảnh xám  # Thực thi Face Detection  faces = faceCascade.detectMultiScale(  gray,  scaleFactor=1.1,  minNeighbors=5,  minSize=(40, 40)  )  print("Tìm thấy {0} khuôn mặt!".format(len(faces)))    # Vẽ một hình tứ giác xung quanh các khuôn mặt phát hiện được. Vẽ trên ảnh màu.  for (x, y, w, h) in faces:  cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)  cv2.imshow('Face Detection', frame) # Hiển thị kết quả ra màn hình  if cv2.waitKey(1) & 0xFF == ord('q'): # Nhấn phím q để dừng  break;  # Trước khi kết thúc chương trình, ta phải giải phóng tài nguyên camera  cap.release()  cv2.destroyAllWindows() |

* Chạy thử code test nhận diện được khuôn mặt của người không.



## Lấy dữ liệu từ database

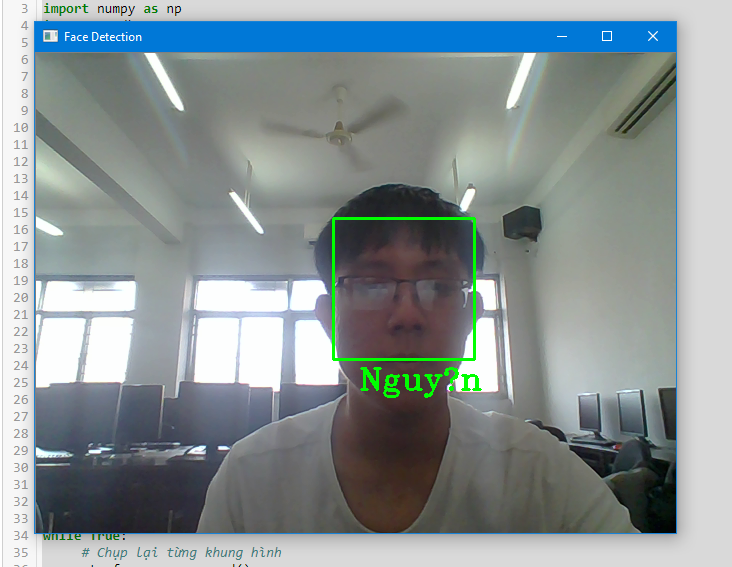
|  |
| --- |
| #Import library  import cv2  import numpy as np  import pyodbc  import os  def insertOrUpdate(id, name):  #Kết nối csdl  conn = pyodbc.connect('Driver={SQL Server};'  'Server=DESKTOP-5NP05L4\VANLONGDOAN;'  'Database=data;'  'Trusted\_Connection=yes;')  #Tìm id của ảnh trong cơ sở dữ liệu  query = "SELECT \*FROM people WHERE ID=" +str(id)  cursor = conn.execute(query)    #Biến kiểm tra có tồn tại hay chưa (nếu tồn tại thì =1)  isRecordExist = 0    for row in cursor:  isRecordExist = 1  #Kiểm tra nếu có tồn tại id ảnh trong cơ sở dữ liệu  if(isRecordExist == 0):  #Chưa có thì thêm id và tên vào cơ sở dữ liệu  query = "INSERT INTO people(ID, NAME) VALUES("+str(id)+ ", ' "+str(name)+" ')"  else:  #Đã tồn tại id thì update tên theo id  query = "UPDATE people SET NAME=' "+str(name)+" ' WHERE ID="+str(id)  conn.execute(query)  conn.commit()  conn.close()  # Load library  faceCascade = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")  cap = cv2.VideoCapture(0)  #Nhập id và tên  id = input("Enter your ID: ")  name = input("Enter Your Name: ")  insertOrUpdate(id, name)  sampleNum =0  while(True):  # Chụp lại từng khung hình  ret, frame = cap.read()  assert ret  # Quá trình nhận diện sẽ được thực hiện trên ảnh xám (Đen/Trắng)  gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) # Chuyển ảnh màu sang ảnh xám  # Thực thi Face Detection  faces = faceCascade.detectMultiScale(gray, 1.3, 5)  for (x, y, w, h) in faces:  cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)  if not os.path.exists('dataset'):  os.makedirs('dataset')  sampleNum +=1  #Cắt ảnh theo hình vuông rồi lưu lại để train  cv2.imwrite('dataset/User.' + str(id) + '.' + str(sampleNum)+ '.jpg', gray[y: y+h, x: x+w])  cv2.imshow('frame', frame)  cv2.waitKey(1)    #Nếu lưu được 150 tấm thì dừng  if(sampleNum >149):  break  cap.release()  cv2.destroyAllWindows() |

## Train dữ liệu cho máy học

|  |
| --- |
| #Import library  import cv2  import numpy as np  import os  from PIL import Image  #Tạo bộ train dữ liệu  recognizer = cv2.face.LBPHFaceRecognizer\_create()  #Đường dẫn  path = 'dataset'  #Lấy ảnh và id của ảnh  def getImageWithId(path):  print(path)  #Lấy danh sách các ảnh trong cơ sở dữ liệu  imagePaths = [os.path.join(path, f) for f in os.listdir(path)]  print(imagePaths)    #Tạo mảng chứa dữ liệu của ảnh và id được mã hóa  faces =[]  IDs = []    #Lưu ảnh và id vô từng mảng tương ứng  for imagePath in imagePaths:  faceImg = Image.open(imagePath).convert('L')  faceNp = np.array(faceImg, 'uint8')  #print(faceNp)    Id = int(imagePath.split('\\')[1].split('.')[1])  faces.append(faceNp)  IDs.append(Id)    cv2.imshow('training', faceNp)  cv2.waitKey(10)    return faces, IDs    faces, Ids = getImageWithId(path)  #Train dữ liệu của ảnh và id  recognizer.train(faces, np.array(Ids))  if not os.path.exists('recognizer'):  os.makedirs('recognizer')  #Lưu dữ liệu train thành file  recognizer.save('recognizer/trainingData.yml')  cv2.destroyAllWindows() |

## Nhận diện khuôn mặt

|  |
| --- |
| #Import library  import cv2  import numpy as np  import pyodbc  import os  from PIL import Image  import time  #Training hình ảnh và thư viện nhận diện  faceCascade = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")  recognizer = cv2.face.LBPHFaceRecognizer\_create()  #Đọc dữ liệu file đã train  recognizer.read('recognizer/trainingData.yml')  #Hàm lấy tên theo id  def getProfile(id):  conn = pyodbc.connect('Driver={SQL Server};'  'Server=DESKTOP-03G7ADL\TANNGUYEN;'  'Database=data;'  'Trusted\_Connection=yes;')  query = "SELECT \*FROM people WHERE ID=" +str(id)  cursor = conn.execute(query)    profile = None    for row in cursor:  profile = row    conn.close()  return profile  #Lấy dữ liệu từ camera  cap = cv2.VideoCapture(0)  fontface = cv2.FONT\_HERSHEY\_COMPLEX  while True:  # Chụp lại từng khung hình  ret, frame = cap.read()  assert ret  # Quá trình nhận diện sẽ được thực hiện trên ảnh xám (Đen/Trắng)  gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) # Chuyển ảnh màu sang ảnh xám  # Thực thi Face Detection  faces = faceCascade.detectMultiScale(gray)    for (x, y, w, h) in faces:  cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)    #Cắt ảnh  roi\_gray = gray[y: y+h, x: x+w]  #Nhận diện (dự đoán theo id) - confidence: sự chính xác  id, confidence = recognizer.predict(roi\_gray)  if(confidence <50):  print(confidence)  profile = getProfile(id)  print(profile)  if(profile != None):  #Profile có 2 giá trị (0: id, 1: name)  cv2.putText(frame, ""+str(profile[1]), (x+ 10, y+h+30), fontface, 1, (0,255,0), 2)  else:  cv2.putText(frame, ""+str(confidence), (x+ 10, y+h+30), fontface, 1, (0,0,255), 2)  cv2.imshow('Face Detection', frame) # Hiển thị kết quả ra màn hình  if cv2.waitKey(1) & 0xFF == ord('q'): # Nhấn phím q để dừng  break  # Trước khi kết thúc chương trình, ta phải giải phóng tài nguyên camera  cap.release()  cv2.destroyAllWindows() |



**CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **1. Tổng kết**

Trong thời gian bắt đầu nhận đề tài đến nay sau khi trải qua việc phân tích và tìm kiếm hướng giải quyết để thực hiện đề tài thì nhìn chung nhóm đã hoàn thành được các mục tiêu cũng như là những mong muốn mà ban đầu đặt ra. Qua việc tự học và tìm kiếm thông tin qua mạng để thực hiện đề tài, nhóm em đã biết và hiểu hơn về quy trình xây dựng một ứng dụng về cách sử dụng những công cụ hỗ trợ cho việc lập trình, về cách phân tích cơ sở dữ liệu, về việc giải quyết vấn đề… Qua đó giúp nhóm em cải thiện được nhiều kỹ năng hơn và có thêm nhiều kinh nghiệm hơn.

### ***1.1. Về mặt lý thuyết***

### Tìm hiểu được cách thức hoạt động của nhận diện khuôn mặt, các thư viện liên quan như OpenCV

### ***1.2. Về mặt hiện thực***

* Thực hiện được hệ thống nhận dạng khuôn mặt đơn giản, chạy bằng code trên Anaconda

## **2. Hạn chế**

* Chỉ có thể chạy trên nền code, chưa có giao diện rõ ràng.

## **3. Hướng phát triển đề tài:**

* Xây dựng ứng dụng đăng nhập bằng khuôn mặt

**Chương V: TÀI LIỆU THAM KHẢO**

* <https://www.stdio.vn/computer-vision/opencv-voi-python-trong-ung-dung-phat-hien-khuon-mat-trong-buc-anh-dYG31n1>
* <https://topdev.vn/blog/opencv-la-gi-hoc-computer-vision-khong-kho/>
* <https://blog.vietnamlab.vn/nhan-dien-khuon-mat-voi-python-va-opencv/>
* <https://viblo.asia/p/tim-hieu-mtcnn-va-ap-dung-de-xac-dinh-vi-tri-cac-khuon-mat-3Q75wkO75Wb?fbclid=IwAR04BRQa7asqSowVZ_U9PUPEfj1Ac2Ns7uoRA9rHQCaTD9cT_1gpX0dwgIM>
* <https://thigiacmaytinh.com/lay-hinh-anh-tu-webcam/?fbclid=IwAR1fox6GmT1WiEh-5-Z1nPd2WK3Zr_oEDB_Qd_8Fh8_K-ayAbaGLMS7yx2U>