**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

A picture containing text, sign

Description automatically generated 🙢⬩🙠

**BÁO CÁO CUỐI KÌ**

**NHẬN DIỆN KHUÔN MẶT (FACE RECOGNITION)**

**HỌC PHẦN: CÔNG NGHỆ PHẦN MỀM NÂNG CAO - 2311COMP1069**

**Thành phố Hồ Chí Minh, tháng 10 năm 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

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

**KHOA CÔNG NGHỆ THÔNG TIN**

A picture containing text, sign

Description automatically generated 🙢⬩🙠

**BÁO CÁO GIỮA KỲ**

**NHẬN DIỆN KHUÔN MẶT (FACE RECOGNITION)**

**Học phần: Công nghệ phần mềm nâng cao - 2311COMP1069**

Giảng viên hướng dẫn: TS. Trần Sơn Hải

Sinh viên thực hiện:

1. Phan Bảo Ân - 46.01.104.004

2. Huỳnh Đăng Phong - 46.01.104.137

3. Nguyễn Minh Tâm - 46.01.104.157

4. Phạm Duy Triều - 46.01.104.197

**Thành phố Hồ Chí Minh, tháng 10 năm 2023**

**LỜI CẢM ƠN**

Nhóm AN3T xin gửi lời cảm ơn chân thành đến giảng viên dạy bộ môn Công nghệ phần mềm nâng cao – thầy Trần Sơn Hải từ khoa Công Nghệ Thông Tin đã quan tâm, hướng dẫn, truyền đạt những kiến thức, kinh nghiệm quý báu cho em trong suốt thời gian học tập vừa qua.

Trong thời gian tham gia lớp học Công nghệ phần mềm nâng cao, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang, là nền tảng để em có thể vững bước sau này.

Mặc dù, chúng em đã cố gắng hết sức nhưng chắc chắn khó có thể tránh khỏi những thiếu sót, kính mong thầy xem xét và góp ý để chúng em được hoàn thiện hơn.

Sau cùng, chúng em xin kính chúc quý Thầy/Cô trong khoa Công nghệ thông tin thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Chúng em xin chân thành cảm ơn!

*TP.Hồ Chí Minh, Tháng 10, Năm 2023*

*Nhóm AN3T*

**MỤC LỤC**

[CHƯƠNG 1. GIỚI THIỆU ĐỒ ÁN 3](#_Toc149929175)

[1.1 Tổng quan về đề tài 3](#_Toc149929176)

[1.2 Giới thiệu về đề tài 4](#_Toc149929177)

[CHƯƠNG 2. CƠ SỞ LÍ THUYẾT 7](#_Toc149929178)

[2.1 Ngôn ngữ python 7](#_Toc149929179)

[2.2 Haar Cascade là gì? 8](#_Toc149929180)

[2.3 Quy trình phát hiện khuôn mặt 9](#_Toc149929181)

[CHƯƠNG 3. ỨNG DỤNG VÀ KẾT QUẢ 19](#_Toc149929182)

[3.1Các bước thực hiện chương trình 19](#_Toc149929183)

[3.2.Kết quả thu được: 21](#_Toc149929184)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 23](#_Toc149929185)

[4.1 Kết luận 23](#_Toc149929186)

[4.2 Hướng phát triển 23](#_Toc149929187)

[TÀI LIỆU THAM KHẢO 24](#_Toc149929188)

# GIỚI THIỆU ĐỒ ÁN

## Tổng quan về đề tài

Với sự phát triển không ngừng của khoa học và công nghệ, đặc biệt là với những chiếc điện thoại thông minh (smartphone) ngày càng hiện đại và được sử dụng phổ biến trong đời sống | con người đã làm cho lượng thông tin thu được bằng hình ảnh ngày càng tăng. Theo đó, lĩnh vực xử lý ảnh cũng được chú trọng phát triển, ứng dụng rộng rãi trong đời sống xã hội hiện đại. Không chỉ dừng lại ở việc chỉnh sửa, tăng chất lượng hình ảnh mà với công nghệ xử lý ảnh hiện nay chúng ta có thể giải quyết các bài toán nhận dạng chữ viết, nhận dạng dấu vân tay, nhận dạng khuôn mặt...

Một trong những bài toán được nhiều người quan tâm nhất của lĩnh vực xử lý ảnh hiện nay đó là nhận dạng khuôn mặt (Face Recognition). Như chúng ta đã biết, 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 như từ khuôn mặt chúng ta có thể xác định giới tính, tuổi tác, chủng tộc, trạng thái cảm xúc, đặc biệt là xác định mối quan hệ với đối tượng (có quen biết hay không). Do đó, bài toán nhận dạng khuôn mặt đóng vai trò quan trọng trong nhiều lĩnh vực đời sống hằng ngày của con người như các hệ thống giám sát, quản lý vào ra, tìm kiếm thông tin một người nổi tiếng,...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.

Bài toán Nhận diện khuôn mặt (Face Recognition) bao gồm nhiều bài toán khác nhau như: phát hiện mặt người (face detection), đánh dấu (facial landmarking), trích chọn (rút) đặc trưng (feature extration), gán nhãn, phân lớp (classification). Trong thực tế, nhận dạng khuôn mặt người (Face Recognition) là một hướng nghiên cứu được nhiều nhà khoa học quan tâm, nghiên cứu để ứng dụng trong thực tiễn. Ở các trường đại học hàng đầu về Công Nghệ Thông Tin như Massachusetts Institute of Technology (MIT), Carnegie Mellon University (CMU), Standford, Berkeley và các công ty lớn như Microsoft, Apple, Google, Facebook đều có các trung tâm về sinh trắc học (Biometrics Center) và nghiên cứu về nhận dạng khuôn mặt người và nó đã trở thành một trong những lĩnh vực nghiên cứu chính cho đến nay. Gần đây, công ty Hitachi Kokusai Electric của Nhật mới cho ra đời một camera giám sát, có thể chụp ảnh và tìm ra 36 triệu khuôn mặt khác có nét tương tự trong cơ sở dữ liệu chỉ trong vòng một giây.

Có hai phương pháp nhận dạng phổ biến hiện nay là nhận dạng dựa trên đặc trưng của các phần tử trên khuôn mặt như biến đổi Gabor Wavelet và mạng Neural, SVM,...và nhận dạng dựa trên xét tổng thể toàn khuôn mặt như phương pháp PCA, LDA, LFA . Trong đó, PCA là phương pháp trích rút đặc trưng nhằm giảm số chiều của ảnh tuy đơn giản nhưng mang lại hiệu quả tốt. Hệ thống hoạt động ổn định và có tính thích nghi cao khi dữ liệu đầu vào thay đổi nhiều.

## Giới thiệu về đề tài

**Giới thiệu về các phương pháp nhận diện khuôn mặt con người:**

1. **Các phương pháp nhận dạng mặt người:**

Hiện nay có 2 phương pháp nhận dạng mặt con người:

* Nhận diện dựa trên các đặc trưng của phần tử trên khuôn mặt (Feature based face recognition)
* Nhận diện dựa trên xét duyệt tổng thể khuôn mặt (appearance based face recognition)

Ngoài ra còn có một số loại nhận diện sử dụng mô hình về khuôn mặt, một số phương pháp được dùng cho loại này:

* Nhận diện 2D: Elastic Bunch Graph, Active Appearance Model.
* Nhận diện 3D: 3D Morphable Model.

1. **Nhận dạng dựa trên các đặc trưng khuôn mặt:**

Đây là phương pháp nhận diện khuôn mặt dựa trên việc xác định các đặc trưng hình học của các chi tiết trên một khuôn mặt (như vị trí, diện tích, hình dạng của mắt, mũi, miệng,…), và mối quan hệ giữa chúng (như khoảng cách của hai mắt, khoảng cách của hai lông mày,…).

Ưu điểm của phương pháp này là giống với cách mà con người sử dụng để nhận biết khuôn mặt. Hơn nữa với việc xác định đặc tính và các mối quan hệ, phương pháp này cá thể cho kết quả tốt trong các trường hợp ảnh nhiều nhiễu như bị nghiêng, bị xoay hoặc ánh sáng thay đổi.

Nhược điểm của phương pháp này là cài đặt thuật toán phức tạp do việc xác định mối quan hệ giữa các đặc tính đòi hỏi các thuật toán phức tạp. Mặt khác, với các ảnh kích thước bé thì các đặc tính sẽ khó phân biệt do khó tính toán được các đặc trưng trên khuôn mặt.

1. **Nhận dạng dựa trên xét toàn diện khuôn mặt:**

Nhận diện đựa trên toàn diện khuôn mặt, có nghĩa mình sẽ không đi xét đến từng thành phần đặc trưng trên khuôn mặt nữa… Mình sẽ xem khuôn mặt là một không gian cụ thể và mình tìm những đặc trưng, những đặc điểm chính trên không gian đó.

Nội dung chính của hướng tiếp cận này là xem mỗi ảnh có kích thước RxC là một vector trong không gian RxC chiều. Ta sẽ xây dựng một không gian mới có chiều nhỏ hơn sao cho khi biểu diễn trong không gian đó các đặc điểm chính của một khuôn mặt không bị mất đi. Trong không gian đó, các ảnh của cùng một người sẽ được tập trung lại thành một nhóm gần nhau và cách xa các nhóm khác.

Hai phương pháp thường được sử dụng là trong hướng tiếp cận này:

* PCA (Principle Components Analysis)
* LDA (Linear Discriminant Analysis)

Với hướng tiếp cận này chúng ta có thể hình dung tương đối dễ chịu hơn.Chúng ta không còn phải phân tích nhiều về đặc điểm khuôn mặt nữa do đó phương pháp tiếp cận này tương đối dễ cài đặt hơn và cũng khá hiệu quả trong việc nhận diện.

Nhược điểm thì tất nhiên sẽ không khỏi tránh khỏi. Tùy theo phương pháp chúng ta sử dụng PCA hay LDA mà đánh giá nhược điểm trong hướng tiếp cận này.

# CƠ SỞ LÍ THUYẾT

## Ngôn ngữ python

Python là một ngôn ngữ lập trình thông dịch, phổ biến và đa năng. Dưới đây là một khái niệm tổng quan về ngôn ngữ Python:

Lịch sử: Python được tạo ra vào thập kỷ 1980 bởi Guido van Rossum tại Hà Lan. Nó là một ngôn ngữ lập trình mã nguồn mở và đã trải qua nhiều phiên bản cải tiến kể từ đó.

Đa năng và Dễ học: Python được biết đến với cú pháp đơn giản và dễ đọc, làm cho nó dễ học và sử dụng cho cả người mới học lập trình và lập trình viên kỳ cựu. Python có một cộng đồng lớn và nhiều tài liệu học tập, giúp bạn dễ dàng tiếp cận ngôn ngữ này.

Tích hợp với nhiều lĩnh vực: Python được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm phát triển phần mềm, web, khoa học dữ liệu, trí tuệ nhân tạo, máy học, thiết kế đồ họa, trò chơi, và nhiều ứng dụng khác.

Cú pháp và Làm việc với các đối tượng: Python sử dụng cú pháp dựa trên dấu lùi (indentation-based syntax), giúp đảm bảo sự đồng nhất trong mã nguồn và khuyến khích việc viết mã đọc dễ hiểu. Python cũng hỗ trợ các cấu trúc dữ liệu như danh sách, bộ, và từ điển, giúp làm việc với dữ liệu dễ dàng.

Mã nguồn mở và Cộng đồng phát triển: Python là một ngôn ngữ mã nguồn mở, có nghĩa là bạn có quyền tự do sử dụng, sửa đổi và phân phối mã nguồn Python. Cộng đồng phát triển Python rất lớn, và có rất nhiều thư viện và framework mở nguồn mà bạn có thể sử dụng để giảm thời gian và công sức trong phát triển ứng dụng.

Phong cách lập trình: Python khuyến khích việc tuân theo phong cách lập trình dựa trên PEP 8 (Python Enhancement Proposals), một tài liệu hướng dẫn về việc viết mã Python đẹp và đồng nhất.

Giao diện với C và C++: Python cung cấp khả năng giao tiếp với mã nguồn C và C++, cho phép bạn tích hợp các thư viện đã được viết bằng C/C++ vào mã Python.

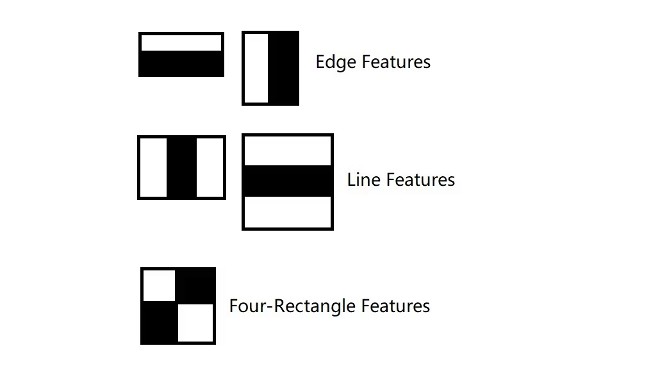
Python được sử dụng rộng rãi trên toàn thế giới và được coi là một trong những ngôn ngữ lập trình phổ biến nhất, đặc biệt trong lĩnh vực phát triển web, khoa học dữ liệu và trí tuệ nhân tạo. Python có sự đa dạng và cộng đồng mạnh mẽ, làm cho nó trở thành một ngôn ngữ hấp dẫn cho người mới học lập trình và lập trình viên chuyên nghiệp.

## Haar Cascade là gì?

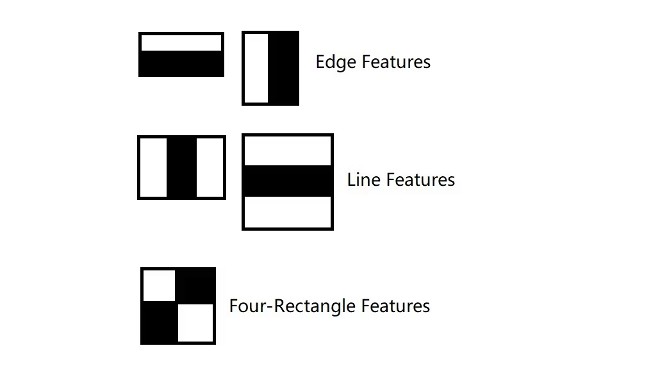
Haar Cascade là thuật toán phát hiện đối tượng được giới thiệu bởi Paul Viola và Michael Jones nhằm phát hiện khuôn mặt trong ảnh hoặc video. Một chức năng phân tầng được đào tạo bằng cách sử dụng nhiều bức ảnh âm và dương, sau này có thể được sử dụng để xác định bất kỳ đối tượng hoặc khuôn mặt nào. Các file được đào tạo sẵn này có sẵn trong repo OpenCV GitHub.

Sử dụng phương thức tiếp cận cửa sổ trượt, một cửa sổ có kích thước cố định sẽ lặp lại hình ảnh từ trái sang phải, từ trên xuống dưới. Ở mỗi giai đoạn, cửa sổ dừng lại và phân loại xem khu vực có chứa khuôn mặt hay không.

OpenCV, một công cụ thị giác máy, hoạt động với mô hình Haar Cascade được huấn luyện trước để phân loại các đối tượng. Mỗi giai đoạn kiểm tra 5 đối tượng, hai đối tượng cạnh, hai đối tượng đường thẳng và một đối tượng bốn ô đan xen.



Khi quan sát, vùng mắt có vẻ sẫm màu hơn vùng má trong khi vùng mũi sáng hơn vùng mắt. Bạn có thể hình dung các đối tượng trong hình bên dưới đây:



Mô hình lý thuyết của khuôn mặt

Sử dụng các đối tượng này và tính toán các pixel, thuật toán xác định hơn 100.000 điểm dữ liệu. Sau đó, bạn có thể sử dụng thêm thuật toán Adaboost để cải thiện độ chính xác và loại bỏ các đối tượng không liên quan. Qua nhiều lần lặp đi lặp lại, cách tiếp cận này giảm thiểu tỷ lệ lỗi và tiếp tục điều chỉnh đối tượng để đạt độ chính xác chấp nhận được.

Tuy nhiên, kỹ thuật cửa sổ trượt sẽ dừng lại nếu một trường hợp thử nghiệm cụ thể không thành công và nó đắt về mặt tính toán. Để giải quyết vấn đề này, bạn có thể áp dụng khái niệm Cascade of Classifier. Thay vì áp dụng tất cả các đối tượng trong một cửa sổ duy nhất, phương pháp này phân nhóm và áp dụng chúng theo từng giai đoạn.

Nếu cửa sổ thất bại ở giai đoạn đầu tiên, quá trình sẽ loại bỏ nó, nếu không nó sẽ tiếp tục. Điều này dẫn tới việc giảm thiểu đáng kể số lượng các thao tác phải thực hiện và giúp nó có thể được dùng cho các ứng dụng thời gian thực.

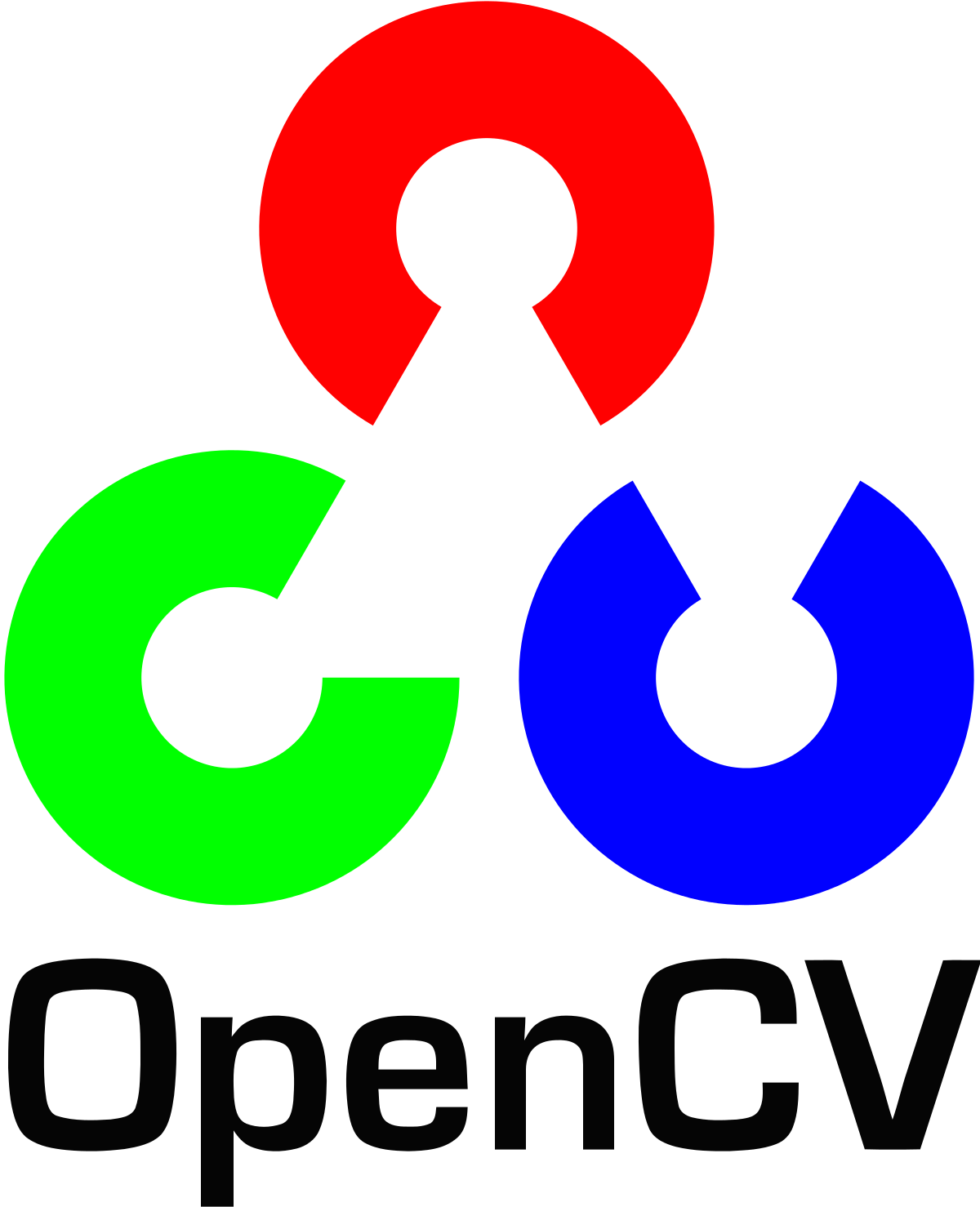
## 2.3 Quy trình phát hiện khuôn mặt

Đây là quy trình mà bạn cần thực hiện để xây dựng một công cụ nhận diện khuôn mặt:

1. Tải Thuật toán Haar Cascade Frontal Face
2. Khởi tạo camera
3. Đọc khung hình từ camera
4. Chuyển đổi hình ảnh sang thang độ xám
5. Lấy tọa độ khuôn mặt
6. Vẽ một hình chữ nhật và đặt thông điệp thích hợp
7. Hiển thị đầu ra

**2.4 Open CV**

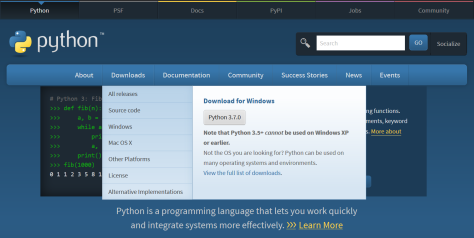
**a) Khái niệm về Open CV**



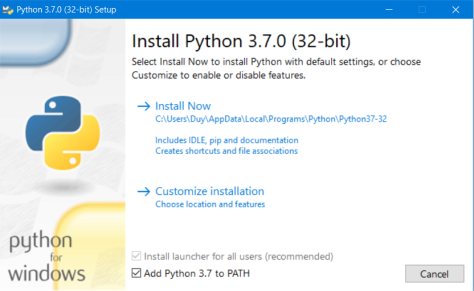
Project [OpenCV](https://opencv.org/) được bắt đầu từ Intel năm 1999 bởi Gary Bradsky. OpenCV viết tắt cho Open Source Computer Vision Library. OpenCV là thư viện nguồn mở hàng đầu cho Computer Vision và Machine Learning, và hiện có thêm tính năng tăng tốc GPU cho các hoạt động theo real-time.OpenCV được phát hành theo *giấy phép BSD***(\*)**, do đó nó miễn phí cho cả học tập và sử dụng với mục đích thương mại. Nó có trên các giao diện C++, C, Python và Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để hỗ trợ hiệu quả về tính toán và chuyên dùng cho các ứng dụng real-time (thời gian thực). Nếu được viết trên C/C++ tối ưu, thư viện này có thể tận dụng được bộ xử lý đa lõi (multi-core processing). OpenCV có một cộng đồng người dùng khá hùng hậu hoạt động trên khắp thế giới bởi nhu cầu cần đến nó ngày càng tăng theo xu hướng chạy đua về sử dụng computer vision của các công ty công nghệ. OpenCV hiện được ứng dụng rộng rãi toàn cầu, với cộng đồng hơn 47.000 người, với nhiều mục đích và tính năng khác nhau từ interactive art, đến khai thác mỏ, khai thác web map hoặc qua robotic cao cấp.

**b) Cách cài đặt Python và Open CV :**

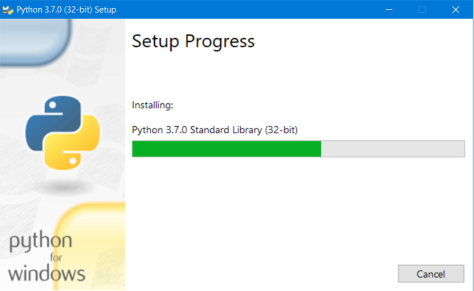
Download Python từ trang chủ: <https://www.python.org/>



Chạy file cài đặt sau khi download xong.



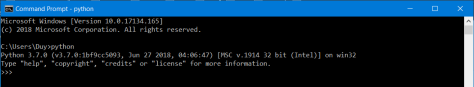
Check vào mục **Add Python 3.7 to PATH** để Python được tự động thêm vào biến môi trường, sau đó bấm vào Install Now để bắt đầu cài đặt.



Sau khi cài đặt hoàn tất, mở command line lên và chạy thử lệnh sau để kiểm tra.

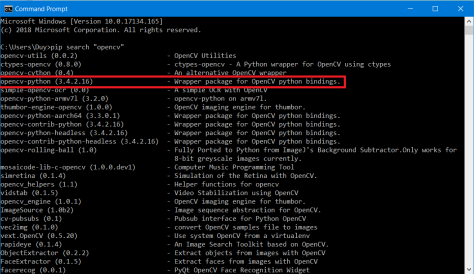
|  |  |
| --- | --- |
| 1 | python |

**. Cài đặt OpenCV**



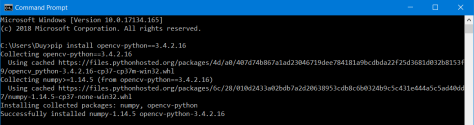
[Pip](http://pip.pypa.io/) là hệ thống quản lý các package của Python. Mình có thể tìm kiếm các package mà pip hỗ trợ thông qua lệnh **pip search**.

|  |  |
| --- | --- |
| 1 | pip search "opencv" |



Như hình trên mình thấy, pip hiện tại hỗ trợ package **opencv-python** (phiên bản 3.4.2.16), nên mình có thể cài đặt OpenCV cho Python thông qua pip. Sử dụng lệnh **pip install** để cài đặt OpenCV.

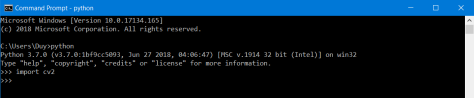
|  |  |
| --- | --- |
| 1 | pip install opencv-python==3.4.2.16 |



Kiểm tra lại, nếu có thể import được OpenCV mà không báo lỗi thì việc cài đặt thành công.

|  |  |
| --- | --- |
| 1 | import cv2 |

<



**2.5 Cấu trúc của Open CV :**

OpenCV do mình build có sự thay đổi đôi chút so với cấu trúc của package gốc. Bao gồm:  
– Thêm nhóm project **contrib**  
– Đưa file cvconfig.h và opencv\_modules.hpp ra ngoài nằm cùng với file solution \*.sln để dễ config  
– Thay đổi các đường dẫn tuyệt đối thành macro, thí dụ: D:\opencv\ thành $(SolutionDir)  
– Loại bỏ các project **perf**và **test**dùng để test hiệu suất OpenCV  
– Tạo các file \*.props chứa các config tối thiểu cho 1 dự án  
– Sửa các đường dẫn output lib, exe cho phù hợp với việc release sản phẩm

Do đó bài này viết trên solution OpenCV đã được build sẵn bên trên để dễ sử dụng.

+Các nhóm project

Ngoài folder include chỉ chứa các file header \*.h thì còn lại 3 folder. Mỗi folder chứa các nhóm project có cùng loại:

* **modules**: các modules chính của OpenCV
* **3rdparty**: các project bên thứ 3 viết ra được tổ chức OpenCV include vào
* **contrib**: các project do người dùng đóng góp

+Nhóm modules

Nhóm này chứa các project chính của OpenCV, chứa các thuật toán do tổ chức OpenCV viết ra. Có 4 project bắt buộc phải có là: core, highgui,imgcodecs và imgproc.

1. **calib3d** Camera Calibration and 3D Reconstruction. Công dụng là tìm tọa độ trong không gian 3D của vật thể. Ứng dụng như dùng để hiệu chuẩn camera khi ống kính bị méo góc (fisheye), chống rung cho camera (calibrate). Tìm tọa độ 3D của bàn cờ vua trong không gian 3 chiều,…
2. **core** Project chính của OpenCV vì cv::Mat được khai báo trong đây. **Bắt buộc phải có**
3. **dnn** Deep neural network hay còn gọi là Deep Learning. Bản thân nó chỉ dùng để kết nối tới các framework khác như Caffe, TensorFlow, Torch, Darknet,… chứ không implement code.
4. **features2d** Dùng để lấy các đặc trưng 2D của vật thể bằng các giải thuật như SIFT, FAST, KAZE, AKAZE,… Từ đó tìm ra các keypoint của vật thể để training nhận dạng.
5. **flann** Tìm kiếm vật thể bằng cách so khớp các keypoint 2D của vật thể
6. **highgui** Vẽ các GUI như cửa sổ cv::imshow(), trackbar. **Bắt buộc phải có**
7. **imgcodecs** Đọc ảnh \*.jpg, \*.png, \*.webp,… bằng cách gọi các project trong nhóm 3rdparty **Bắt buộc phải có**
8. **imgproc** Chứa các hàm xử lý, biến đổi,… hình ảnh như dilate, erode, blur,… **Bắt buộc phải có**
9. **js** Js chính là java script, dùng để nhúng vào website
10. **ml** các thuật toán của Machine Learning như SVM, KNN,…
11. **objdetect** Detect object dùng Cascade
12. **photo** Xử lý ảnh chụp tương tự lightroom: chỉnh tone màu, HDR,…
13. **shape** Tìm các hình cơ bản như tam giác, tứ giác, vuông,…
14. **stitching** Ghép nhiều ảnh thành ảnh panorama
15. **superres** Super Resolution: nâng cao resolution của ảnh bằng phương pháp nội suy
16. **video** Xử lý video
17. **videoio** Load file video
18. **videostab**Video stabilization: sửa video quay bị rung
19. **viz** Đồ họa 3D tương tự OpenGL

+ Nhóm 3rdparty

Các project trong nhóm 3rd chỉ cần include khi có nhu cầu sử dụng đọc loại ảnh khác nhau. Trong này có 2 project bắt buộc có là zlib và libjpeg

1. **ippicv** Intell IPP, dùng để tăng tốc độ xử lý ảnh
2. **ittnotify** thu thập data trong quá trình sử dụng, tìm hiểu thêm [ITT](https://software.intel.com/en-us/articles/intel-itt-api-open-source)
3. **libjasper** định dạng hiếm gặp, không quan trọng
4. **libjpeg-turbo** đọc ảnh jpeg, từ version 3.4 trở đi libjpeg thay bằng libjpeg-turbo **Bắt buộc phải có**
5. **libpng** đọc ảnh png
6. **libtiff** đọc ảnh tiff
7. **libwebp** Webp là định dạng ảnh mới, nhẹ hơn jpeg, hay dùng trên website
8. **openexr** định dạng ảnh EXR, ít khi sử dụng
9. **protobuf** là 1 giao thức văn bản mới (như XML, JSON,…) được Google tạo ra, chỉ include khi sử dụng DNN
10. **zlib** Nén và giải nén file **Bắt buộc phải có**

+ Nhóm contrib

Các project trong này do người dùng OpenCV đóng góp, nổi tiếng nhất là **face** để nhận diện khuôn mặt. Một số project lạ mình sẽ bổ sung sau

 **aruco** [AR – thực tế tăng cường](https://www.youtube.com/watch?v=A1ZnjySo9x4)

 **bgsegm** Background segment: tách nền sử dụng thuật toán Background subtraction

 **bioinspired** làm tăng độ tương phản & xác định chuyển động trong 1 chuỗi hình ảnh liên tiếp nhau.

 **ccalib** Custom calibration: có thêm option từ calib3d

 **cnn\_3dobj** Convolutional Neural Networks để phân lớp 3D object

 **cvv** debug hình ảnh trong các quá trình xử lý. Tuy nhiên cần có QT trong quá trình make

 **dnns\_easily\_fooled** Đây không phải là thư viện, đây chỉ là 1 chứng minh cho việc DNN (Deep Neural Network) dễ dàng bị đánh lừa bởi[ảnh trừu tượng](https://thigiacmaytinh.com/wp-content/uploads/2019/04/fahrelnissa-zeid-resolved-problems_web_1.jpg" \t "_blank)

 **dnn\_objdetect** Phát hiện vật thể bằng Deep Neural Network

 **dpm** Deformable Part-based Models dùng để phát hiện vật thể

 **face** nhận diện khuôn mặt

 **freetype** vẽ text lên ảnh với font bất kỳ tùy chọn

 **fuzzy** [phục hồi ảnh](https://docs.opencv.org/3.4.0/d3/db8/tutorial_fuzzy_inpainting.html" \t "_blank)

 **hdf** Dùng để đọc ghi [file HDF](https://www.hdfgroup.org/). File HDF có cấu trúc phức tạp, không giới hạn dung lượng, khả năng đọc ghi nhanh, thường dùng trong Big Data.

 **hfs** tách ảnh khỏi nền

 **img\_hash** Dùng để kiểm tra ảnh có giống nhau hay không bất kể định dạng và kích thước. Giống như mắt người sẽ nhận dạng ảnh giống nhau bởi nội dung chứ không quan tâm đến giá trị nhị phân lưu trữ.

 **line\_descriptor** tìm đường thẳng trong ảnh

 **matlab** sử dụng opencv với matlab

 **optflow** xác định [hướng chuyển động của vật thể](https://www.youtube.com/watch?v=7oukLheo3fg" \t "_blank)

 **ovis**

 **phase\_unwrapping**

 **plot** vẽ biểu đồ

 **reg** Image registration

 **rgbd** Color data with depth processing sử dụng dữ liệu từ máy Kinect. Hình ảnh được truyền về kèm theo kênh Depth (ngoài RGB) để tính toán độ sâu của vật thể.

 **saliency** tìm [vật thể nổi bật nhất](https://thigiacmaytinh.com/wp-content/uploads/2019/04/opencv_saliency_finegrained_boat.jpg" \t "_blank)trong ảnh

 **sfm** Tạo bản đồ 3D (hay 3D model) bằng ảnh chụp từ nhiều góc độ của vật thể

 **stereo** Xử lý ảnh stereo. [Ảnh stereo](http://extra.listverse.com/amazon/stereograms/ChipStephan.jpg" \t "_blank) là ảnh tạo ra sự khác biệt giữa mắt trái và mắt phải để cảm nhận được hình ảnh 3D bên trong. (Mình đã cố gắng thử nhưng không thể nào nhìn được 3D, không biết người khác thì sao)

 **structured\_light** Scan vật thể 3D bằng cách tạo ra các vân ánh sáng, từ đó xác định được hình khối của vật thể cần scan.

 **surface\_matching** Tìm kiếm vật thể 3D bằng dữ liệu từ các máy scan 3D trong môi trường thực tế. Giống như tìm vật thể 2D trong ảnh.

 **text** phát hiện vị trí text trong ảnh và đọc text

 **tracking** bám theo vật thể di chuyển

 **xfeatures2d** nâng cao của project feature2D

 **ximgproc** nâng cao của project imgproc

 **xobjdetect** nâng cao của project objdetect

 **xphoto** nâng cao của project photo

# ỨNG DỤNG VÀ KẾT QUẢ

## 3.1Các bước thực hiện chương trình

Bước 1:Chạy file datacollect.py để chương trình thu thập hình ảnh:

A screenshot of a computer program

Description automatically generated

Bước 2:Chương trình yêu cầu id để lưu hình ảnh thu thập được datasets:

A screenshot of a computer

Description automatically generated

Bước 3:chạy file training\_demo.py để chương trình duyệt và kiểm tra các ảnh vửa thu được dữ liệu trong dataset:

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Bước 4:Chạy file testmodel.py để chạy chương trình nhận diện khuôn mặt

A screen shot of a computer program

Description automatically generated

## 3.2.Kết quả thu được:

Chương trình nhận được khuôn mặt nằm trong dữ liệu thu thập được sẽ được

nhận diện ra dưới dạng “User” nếu đó là một cá thể nào khác không nằm trong datasets thì sẽ hiện ra dưới dạng “unknown”

A screenshot of a person's face

Description automatically generated

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Qua tìm hiểu bài toán phát hiện mặt người trong ảnh em đã thấy được sự quan trọng của xử lý ảnh cũng như sự phát triển và tầm quan trọng của bài toán phát hiện mặt người trong đời sống. Trong quá trình xây dựng chương trình em đã tìm hiểu về thư viện mã nguồn mở OpenCV. Qua đó biết cách sử dụng các hàm được xây dựng sẵn trong thư viện OpenCV. Về chương trình em mới chỉ thực hiện phát hiện mặ người trong ảnh, kết quả đạt được như mong muốn, thời gian phát hiện nhanh đối với mặt người chụp thẳng, chất lượng ảnh tốt, không có vật cản…., còn trong các ảnh như chụp hơi nghiêng hay ảnh có màu sắc tối thì chương trình vẫn chưa phát hiện ra khuôn mặt trong ảnh đó.

## Hướng phát triển

Chương trình phát hiện mặt người trong ảnh để nhận diện được ra khuôn mặt và

các chi tiết trên khuôn mặt. Nhằm đưa ra được những ứng dụng thực tế trong cuộc

sống hiện đại:

* Phân tích cảm xúc của khuôn mặt người.
* Tổ chức tìm kiếm liên quan đến con người thông qua khuôn mặt trên nhiều hệ cơ sở dữ liệu lớn.
* Điều khiển ra vào các cơ quan, văn phòng.
* Hệ thống quan sát, theo dõi và bảo vệ.
* Giải trí.
* Nhận diện tội phạm
* Hệ thống giao tiếp thông minh giữa người và máy.

# TÀI LIỆU THAM KHẢO

<https://acodary.wordpress.com/2018/07/21/opencv-cai-dat-opencv-python-tren-windows/>

<https://thigiacmaytinh.com/cau-truc-opencv/>

Link video demo: <https://by.tn/QSZl>

https://github.com/Ngnminhtam/Python-Face-Recognition.git