|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BỘ CÔNG NGHIỆP**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **KHOA CÔNG NGHỆ THÔNG TIN**  **======\*\*\*======**  **Logo%20HaUI%20ban%20chuan**  **BÁO CÁO NGHIÊN CỨU KHOA HỌC**  **NHẬN DIỆN VÀ XÁC ĐỊNH GIỚI TÍNH**   |  |  | | --- | --- | | **Giáo viên hướng dẫn:** | **TS. Trần Chí Kiên** | | **Thành viên:** | **Đỗ Mạnh Quang (Chủ nhiệm đề tài)** | |  | **Bùi Chiến Thắng** | |  |  | |  |  | |  |  | |  |  |   *Hà nội, Năm 2018* |

# LỜI CAM ĐOAN

Nhóm em bao gồm Đỗ Mạnh Quang và Bùi Chiến Thắng, sinh viên lớp Hệ Thống Thông Tin 1 K11 và lớp Kỹ Thuật Phần Mềm 3 K11 xin cam đoan báo cáo luận văn này được viết bởi nhóm em dưới sự hướng dẫn của Thầy giáo, TS. Trần Chí Kiên. Tất cả các kết quả đạt được trong luận văn này là quá trình tìm hiểu, nghiên cứu của riêng nhóm em. Trong toàn bộ nội dung của luận văn, những điều được trình bày là kết quả của nhóm em hoặc là được tổng hợp từ nhiều nguồn tài liệu khác. Các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Nhóm em xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà Nội, ngày ….. tháng …… năm 2018

Người cam đoan Người cam đoan

Quang Thắng

Đỗ Mạnh Quang Bùi Chiến Thắng

# LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin bày tỏ sự cảm ơn chân thành đối với Thầy giáo TS. Trần Chí Kiên – giáo viên hướng dẫn trực tiếp của tôi. Thầy Kiên đã cho tôi những gợi ý và chỉ dẫn quý báu trong quá trình nghiên cứu và hoàn thiện đề tài nghiên cứu khoa học.

Nhóm em cũng xin gửi lời cảm ơn tới các thầy cô trong khoa Công nghệ thông tin, trường Đại học Công Nghiệp Hà Nội đã hướng dẫn, chỉ bảo và tạo điều kiện cho chúng em học tập và nghiên cứu tại trường trong suốt thời gian qua.

Nhóm em cũng xin được cảm ơn gia đình, những người thân và bạn bè đã quan tâm, động viên, giới thiệu các tài liệu hữu ích trong thời gian học tập và nghiên cứu khoa học.

Mặc dù đã cố gắng hoàn thành đề tài nghiên cứu khoa học nhưng chắc chắn sẽ không tránh khỏi những sai sót, nhóm em kính mong nhận được sự thông cảm và chỉ bảo của các thầy cô và các bạn.

Nhóm em xin chân thành cảm ơn!

# MỤC LỤC

Contents

[LỜI CAM ĐOAN 2](#_Toc7426931)

[LỜI CẢM ƠN 3](#_Toc7426932)

[MỤC LỤC 4](#_Toc7426933)

[CHƯƠNG 1 : GIỚI THIỆU NỘI DUNG NGHIÊN CỨU 7](#_Toc7426934)

[1.1. Mục đích của đề tài nghiên cứu khoa học. 7](#_Toc7426935)

[1.2. Các mục tiêu cần đạt được sau khi tham gia nghiên cứu. 7](#_Toc7426936)

[1.3. Khái quát bài toán về xử lý ảnh trong học máy. 7](#_Toc7426937)

[1.4. Mô tả bài toán cần nghiên cứu và xử lý. 9](#_Toc7426938)

[1.5. Các hướng tiếp cận bài toán. 9](#_Toc7426939)

[1.6. Các hướng ứng dụng và phát triển. 9](#_Toc7426940)

[CHƯƠNG 2 : NGHIÊN CỨU NHẬN DIỆN VÀ XÁC ĐỊNH GIỚI TÍNH 10](#_Toc7426941)

[**2.1.** Nghiên cứu ngôn ngữ Python. 10](#_Toc7426942)

[**2.1.1.** Lý do sử dụng ngôn ngữ Python. 10](#_Toc7426943)

[**2.1.2.** Cài đặt ngôn ngữ Python và công cụ IDE. 10](#_Toc7426944)

[**2.2.** Nghiên cứu sử dụng công cụ IDE Pycharm. 12](#_Toc7426945)

[**2.2.1.** Hỗ trợ mã hóa thông minh 12](#_Toc7426946)

[**2.2.2.** Công cụ dành cho nhà phát triển tích hợp 12](#_Toc7426947)

[**2.2.3.** Dụng cụ khoa học 12](#_Toc7426948)

[**2.2.4.** IDE có thể tùy chỉnh và đa nền tảng 13](#_Toc7426949)

[**2.2.5.** Tạo một project python trong Pycharm 13](#_Toc7426950)

[**2.3.** Nghiên cứu các thư viện xử lý ảnh trong Python. 17](#_Toc7426951)

[**2.3.1.** Thư viện OpenCV. 17](#_Toc7426952)

[**2.3.2.** Thư viện Numpy. 18](#_Toc7426953)

[**2.3.3.** Thư viện Mahotas. 18](#_Toc7426954)

[**2.3.4.** Thư viện Keras. 18](#_Toc7426955)

[**2.3.5.** Thư viện Tensorflow. 18](#_Toc7426956)

[**2.3.6.** Thư viện Scikit-Image. 19](#_Toc7426957)

[**2.4.** Nghiên cứu thuật toán xử lý nhân dạng giới tính. 20](#_Toc7426958)

[**2.4.1.** Các thuật toán nghiên cứu. 20](#_Toc7426959)

[**2.5.** các hướng phát triển bàn toán trong thực tế. 48](#_Toc7426960)

[**2.5.1.** Giải quyết vấn đề giai lận thi hộ trong kỳ thi. 48](#_Toc7426961)

[**2.5.2.** Giải quyết vấn đề xác định đối tượng truy nã. 49](#_Toc7426962)

[**2.5.3.** Giải quyết vấn đề điểm danh trong lớp học. 49](#_Toc7426963)

[**2.5.4.** Khai phá dữ liệu phân tích thông kê và tính toán tỷ số giới tính trong khu vực. 50](#_Toc7426964)

[CHƯƠNG 3 : QUY TRÌNH THỰC HIỆN 51](#_Toc7426965)

[3.1. Thu thập dữ liệu cho học máy ( tranning ). 51](#_Toc7426966)

[3.1.1. Dữ liệu ảnh cho học máy. 51](#_Toc7426967)

[3.1.2. Quy trình thực hiện công việc học máy. 53](#_Toc7426968)

[3.2. Xử lý dữ liệu ảnh hoặc phim đầu vào. 56](#_Toc7426969)

[3.2.1. Xử lý dữ liệu Webcam. 56](#_Toc7426970)

[3.2.2. Xử lý dữ liệu CameraIP. 57](#_Toc7426971)

[3.2.3. Xử lý dữ liệu Camera Smart Phone. 58](#_Toc7426972)

[3.3. Xử lý dữ liệu nhận diện ( Nhận dạng ) 59](#_Toc7426973)

[3.3.1. Mô tả Sơ đồ quá trình. 59](#_Toc7426974)

[3.3.2. Chương trình xử lý. 60](#_Toc7426975)

[3.4. Xử lý dữ liệu phân loại hoặc xác định giới tính. 62](#_Toc7426976)

[3.4.1. Mô tả sơ đồ quá trình. 62](#_Toc7426977)

[3.4.2. Chương trình xử lý. 62](#_Toc7426978)

[3.5. Kết quả thu được sau khi xử lý. 64](#_Toc7426979)

[3.5.1. Phân loại giới tính. 64](#_Toc7426980)

[3.5.2. Nhận diện ( Nhận dạng) Gương mặt 64](#_Toc7426981)

[3.5.3. Kết hợp xử lý. 65](#_Toc7426982)

[3.5.4. Nhận xét. 65](#_Toc7426983)

[CHƯƠNG 4 : ĐÁNH GIÁ QUÁ TRÌNH NGHIÊN CỨU 66](#_Toc7426984)

[4.1. Kết quả đạt được sau quá trình nghiên cứu. 66](#_Toc7426985)

[4.2. Đánh giá kết quả sản phẩm demo. 67](#_Toc7426986)

[CHƯƠNG 5 : LIÊN HỆ VÀ TÀI LIỆU THAM KHẢO 67](#_Toc7426987)

[5.1. Liên hệ nhóm nghiên cứu. 67](#_Toc7426988)

[5.2. Các nguồn tài liệu tham khảo. 68](#_Toc7426989)

# CHƯƠNG 1 : GIỚI THIỆU NỘI DUNG NGHIÊN CỨU

## Mục đích của đề tài nghiên cứu khoa học.

Vận dụng các thuật toán, thư viện học máy để xác định giới tính, nhận diện người trong khung ảnh , video , video stream.

## Các mục tiêu cần đạt được sau khi tham gia nghiên cứu.

Hiểu rõ được ngôn ngữ và các thư viện của python.

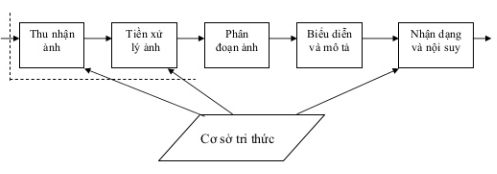
Sử dụng thành thạo IDE Pycharm.

Nghiên cứu và vận dụng các thuật toán, thư viện xử lý ảnh (Eigen Faces, Fischer Faces, LPBH, PCA, k-NN,…) .

Áp dụng thư viện và đề xuất phương án để giải quyết vấn đề trong thực tế.

## Khái quát bài toán về xử lý ảnh trong học máy.

Bài toán xử lý ảnh gồm các giai đoạn tổng quát sau:



Hình 1.1 Các bước cơ bản trong xử lý ảnh

Sơ đồ này gồm các thành phần sau:

Thu nhận ảnh: Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình XLA. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, sensor, máy scanner,v.v… và sau đó các tín hiệu này sẽ được số hóa. Việc lựa chọn các thiết bị thu nhận ảnh sẽ phụ thuộc vào đặc tính của các đối tƣợng cần xử lý. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.

Tiền xử lý: Ở bước này, ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch,v.v… với mục đích làm cho chất lượng ảnh trở lên tốt hơn nữa, chuẩn bị cho các bước xử lý phức tạp hơn về sau trong quá trình XLA. Quá trình này thường được thực hiện bởi các bộ lọc.

Phân đoạn ảnh: phân đoạn ảnh là bước then chốt trong XLA. Giai đoạn này phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng màu, cùng mức xám v.v… Mục đích của phân đoạn ảnh là để có một miêu tả tổng hợp về nhiều phần tử khác nhau cấu tạo lên ảnh thô. Vì lượng thông tin chứa trong ảnh rất lớn, trong khi đa số các ứng dụng chúng ta chỉ cần trích một vài đặc trưng nào đó, do vậy cần có một quá trình để giảm lượng thông tin khổng lồ đó. Quá trình này bao gồm phân vùng ảnh và trích chọn đặc tính chủ yếu.

Tách các đặc tính: Kết quả của bước phân đoạn ảnh thường được cho dưới dạng dữ liệu điểm ảnh thô, trong đó hàm chứa biên của một vùng ảnh, hoặc tập hợp tất cả các điểm ảnh thuộc về chính vùng ảnh đó. Trong cả hai trường hợp, sự chuyển đổi dữ liệu thô này thành một dạng thích hợp hơn cho việc xử lý trong máy tính là rất cần thiết. Để chuyển đổi chúng, câu hỏi đầu tiên cần phải trả lời là nên biểu diễn một vùng ảnh dưới dạng biên hay dưới dạng một vùng hoàn chỉnh gồm tất cả những điểm ảnh thuộc về nó. Biểu diễn dạng biên cho một vùng phù hợp với những ứng dụng chỉ quan tâm chủ yếu đến các đặc trưng hình dạng bên ngoài của đối tượng, ví dụ như các góc cạnh và điểm uốn trên biên chẳng hạn. Biểu diễn dạng vùng lại thích hợp cho những ứng dụng khai thác các tính chất bên trong của đối tượng, ví dụ như vân ảnh hoặc cấu trúc xương của nó. Chúng ta còn phải đưa ra một phương pháp mô tả dữ liệu đã được chuyển đổi đó sao cho những tính chất cần quan tâm đến sẽ được làm nổi bật lên, thuận tiện cho việc xử lý chúng.

Nhận dạng và giải thích: Đây là bước cuối cùng trong quá trình XLA. Nhận dạng ảnh có thể được nhìn nhận một cách đơn giản là việc gán nhãn cho các đối tượng trong ảnh. Ví dụ đối với nhận dạng chữ viết, các đối tượng trong ảnh cần nhận dạng là các mẫu chữ, ta cần tách riêng các mẫu chữ đó ra và tìm cách gán đúng các ký tự của bảng chữ cái tương ứng cho các mẫu chữ thu được trong ảnh. Giải thích là công đoạn gán nghĩa cho một tập các đối tượng đã được nhận biết. Chúng ta cũng có thể thấy rằng, không phải bất kỳ một ứng dụng XLA nào cũng bắt buộc phải tuân theo tất cả các bước xử lý đã nêu ở trên, ví dụ như các ứng dụng chỉnh sửa ảnh nghệ thuật chỉ dừng lại ở bước tiền xử lý. Một cách tổng quát thì những chức năng xử lý bao gồm cả nhận dạng và giải thích thường chỉ có mặt trong hệ thống phân tích ảnh tự động hoặc bán tự động, được dùng để rút trích ra những thông tin quan trọng từ ảnh, ví dụ như các ứng dụng nhận dạng ký tự quang học, nhận dạng chữ viết tay v.v…

## Mô tả bài toán cần nghiên cứu và xử lý.

Nhận diện (nhận dạng) khuôn mặt và giới tính bằng phương phát nghiên cứu và sử dụng thư viện thuật toán về Machine learning, Deep learning, Neural network, …trong Python.

Chúng ta cần tìm hiểu một không gian đặc trưng chiều cao có khả năng phân biệt khuôn mặt của những người khác nhau. Khi chúng ta có một mô hình đào tạo, chúng ta có thể sử dụng nó để tạo ra các đặc trưng duy nhất (đặc điểm duy nhất) cho mỗi khuôn mặt. Cuối cùng, chúng ta có thể so sánh các đặc điểm của khuôn mặt mới với khuôn mặt đã xác nhận (biết) để nhận dạng người đó.

Chúng ta cần tìm kiểu cách phân loại vertor đặc trưng của giới tính sau đó tính khoảng cách với vertor đặc trưng mới trong không gian n chiều để đưa ra kết quả là nam hoặc nữ.

## Các hướng tiếp cận bài toán.

Hướng tiếp cận nghiên cứu thuật toán : Tìm kiếm các bài báo , tài liệu liên quan đến các thuật toán xử lý ảnh, các vấn đề cần xử lý trong nhận diện(nhận dạng) phân loại giới tính trên trang uy tín như learopencv, geekforgeek , wikipedia ,…để nắm vứng các vấn đề cơ bản cần xử lý.

Hướng tiếp cận sử dụng thư viện có sẵn : Trong python có nhiều thư viện mạnh về xử lý ảnh chúng ta có thể sử dụng thư viện và cải tiến thêm để đạt được kết quả tốt nhất.

## Các hướng ứng dụng và phát triển.

Phát triển hệ thống điểm danh trong trường học.

Phát triển hệ thống nhận diện ( nhận dạng ) chống gian lận thi hộ trong các kỳ thi.

Phát triển hệ thống nhận diện (nhận dạng) đối tượng.

Phân tích và khai phá dữ liệu trong bán hàng cho từng giới tính.

# CHƯƠNG 2 : NGHIÊN CỨU NHẬN DIỆN VÀ XÁC ĐỊNH GIỚI TÍNH

## Nghiên cứu ngôn ngữ Python.

### Lý do sử dụng ngôn ngữ Python.

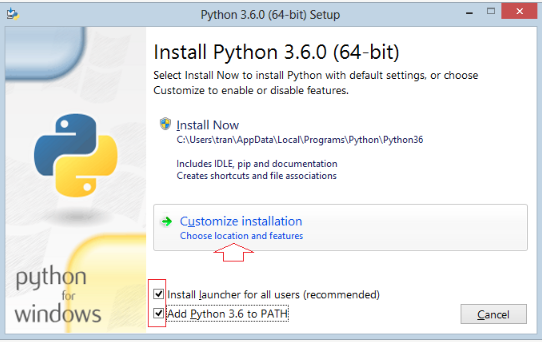
**Python** là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao cho các mục đích lập trình đa năng, ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, được thiết kế để trở thành một ngôn ngữ dễ học, mã nguồn dễ đọc, bố cục trực quan, dễ hiểu thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Đặc biệt Python có nhiều thư viện để áp dụng vào xử lý ảnh.

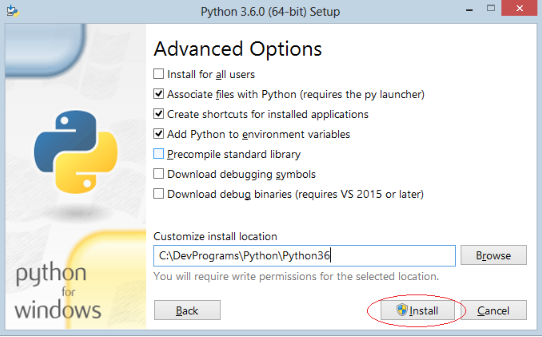
### Cài đặt ngôn ngữ Python và công cụ IDE.

Để download **Python**, bạn truy cập địa chỉ:

<https://www.python.org/downloads/>

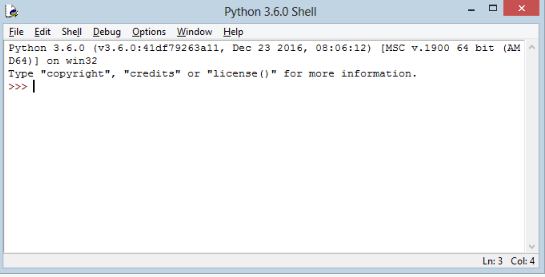
Chạy file bạn download được ở bước trên để bắt đầu cài đặt. Chọn **"Customize Installation"** để bạn có thể tùy chọn ví trí **Python** sẽ được cài đặt ra.





Trên "Start Menu" của **Windows** bạn chạy **IDLE (Python 3.6 64bit)**:

Chương trình **"Python Shell"** đã được chạy, nó là một chương trình giúp bạn viết mã **Python.**



Một số **IDE** giúp bạn lập trình **Python**:

1. PyCharm
2. Eclipse (Cài đặt PyDev plugin cho Eclipse)

## Nghiên cứu sử dụng công cụ IDE Pycharm.

### Hỗ trợ mã hóa thông minh

PyCharm cung cấp hoàn thành mã thông minh, kiểm tra mã, đánh dấu lỗi nhanh và sửa lỗi nhanh, cùng với tái cấu trúc mã tự động và khả năng điều hướng phong phú. Trình định dạng mã với cấu hình kiểu mã và các tính năng khác giúp bạn viết mã gọn gàng, dễ hỗ trợ. PyCharm chứa PEP-8 tích hợp cho Python và định dạng mã tuân thủ tiêu chuẩn khác cho các ngôn ngữ được hỗ trợ.

### Công cụ dành cho nhà phát triển tích hợp

Bộ sưu tập lớn các công cụ có sẵn của PyCharm có sẵn bao gồm trình gỡ lỗi tích hợp và trình chạy thử nghiệm; một hồ sơ Python; một thiết bị đầu cuối tích hợp; tích hợp với VCS chính và các công cụ cơ sở dữ liệu tích hợp; khả năng phát triển từ xa với thông dịch viên từ xa; một thiết bị đầu cuối ssh tích hợp; và tích hợp với Docker và Vagrant.

### Dụng cụ khoa học

PyCharm Professional Edition giúp bạn phân tích dữ liệu của mình với Python. Chỉ cần tạo một dự án khoa học, thêm dữ liệu của bạn và bắt đầu phân tích.

Bắt đầu phân tích của bạn bằng cách chạy các lệnh Python ad-hoc trong bảng điều khiển Python. PyCharm giúp bạn bằng cách hiển thị cho bạn tất cả các biến bạn đã tạo. Bạn cũng có thể sử dụng PyCharm từ xa SciView để tìm hiểu sâu hơn về DataFrames và NumPy Series của bạn.

Trực quan hóa dữ liệu là một bước thiết yếu trong bất kỳ phân tích dữ liệu nào, PyCharm giúp bạn bằng cách hiển thị cho bạn các lô của bạn bên trong IDE. PyCharm cũng theo dõi các ô cuối cùng mà bạn đã tạo, giúp dễ dàng phát hiện các thay đổi giữa hai ô.

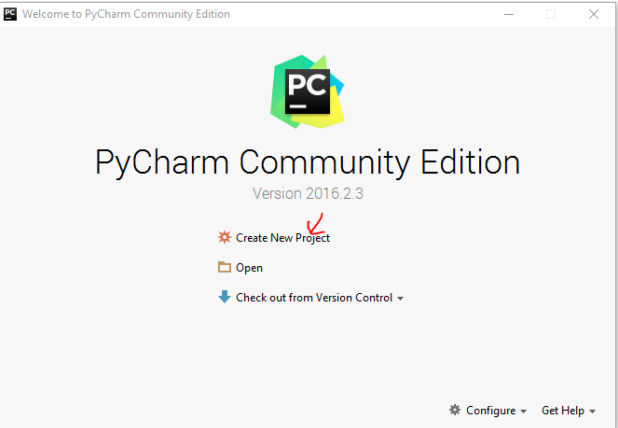
Sau khi bạn tinh chỉnh các lệnh của mình, bạn có thể sao chép và dán chúng vào tệp .py, PyCharm sẽ xử lý định dạng cho bạn. Bạn có thể dễ dàng chia các tệp Python của mình thành các phần logic bằng cách xác định các ô mã. Chỉ cần tạo một nhận xét bắt đầu bằng # %% để bắt đầu một ô.

### IDE có thể tùy chỉnh và đa nền tảng

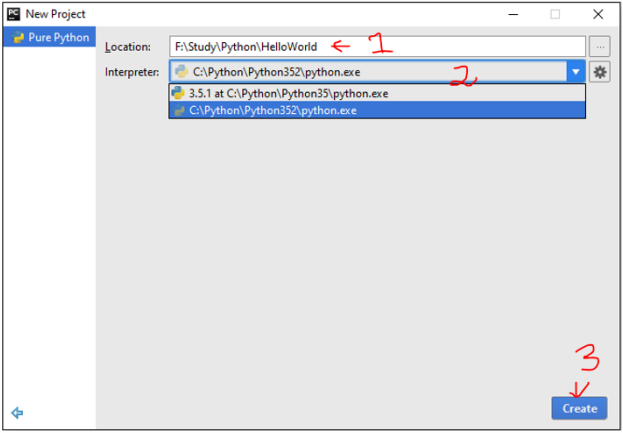
Sử dụng PyCharm trên Windows, Mac OS và Linux với một khóa cấp phép duy nhất. Tận hưởng một không gian làm việc được tinh chỉnh với các bảng màu tùy chỉnh và các ràng buộc phím, với mô phỏng VIM có sẵn.

### Tạo một project python trong Pycharm

Khi khởi động JetBrains PyCharm Community Edition, ta sẽ có giao diện sau:



Ở màn hình chon Create New Project:

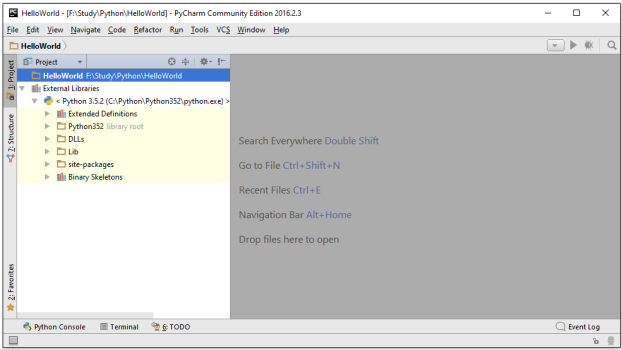


Chọn location lưu trữ.

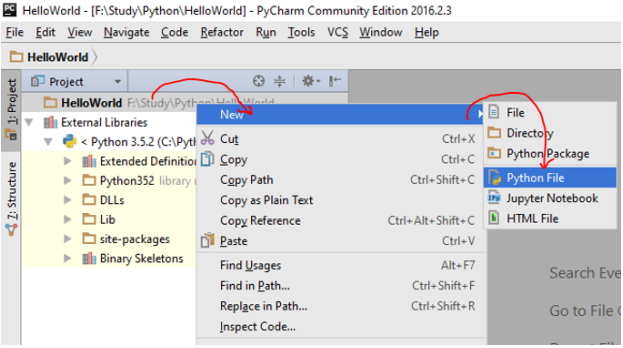
Chọn trình thông dịch Python.

Nhấn Create để tạo Project.

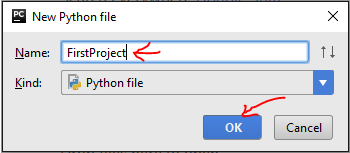
Sauk hi nhấn Create , Pycharm sẽ hiển thị cấu trúc mặc đinh như sau:



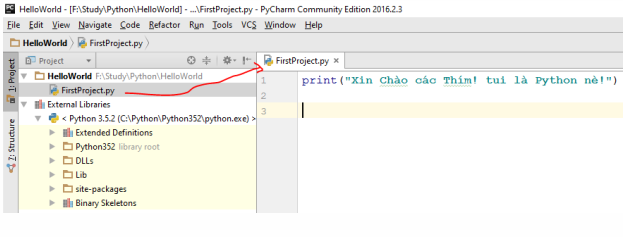
Để tạo tập tin Python: Ta bấm chuột phải vào Project HelloWorld/ chọn New/ Python File:



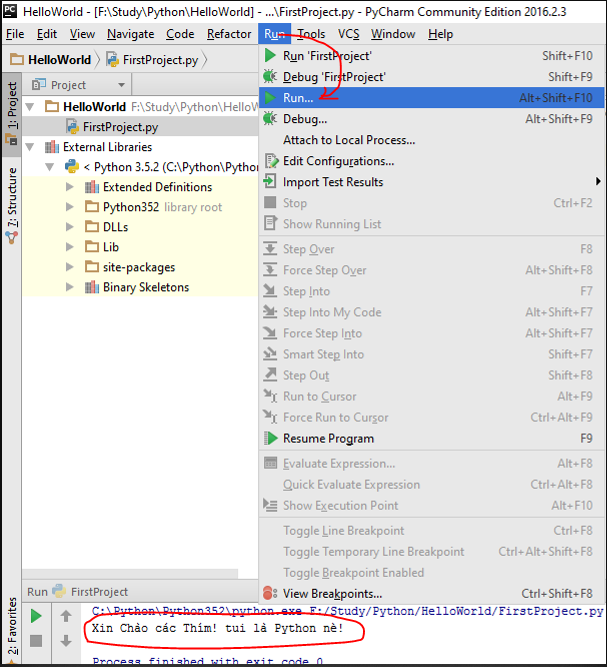
Sau khi nhấn Python File, màn hình New Python file hiển thị ra như dưới đây:



Ta tiến hành đặt tên File, loại file là Python file rồi nhấn nút OK, tập tin **FirstProject.py** sẽ được hiển thị như dưới đây (py là mở rộng của các tập tin mã nguồn Python):



Để chạy mã tập tin FirstProject.py, các bạn vào Run/chọn Run.



## Nghiên cứu các thư viện xử lý ảnh trong Python.

### Thư viện OpenCV.

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.

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

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).

### Thư viện Numpy.

Numpy(viết tắt của Nummerical Python) là một thư viện của Python hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm số đã được tối ưu áp dụng lên các mảng nhiều chiều đó. Numpy đặc biệt hữu ích khi thực hiện các hàm số liên quan tới Đại Số Tuyến Tính.Bên cạnh những ứng dụng khoa học rõ ràng, NumPy cũng có thể được sử dụng như một kho chứa dữ liệu chung đa chiều hiệu quả. Kiểu dữ liệu tùy ý có thể được xác định. Điều này cho phép NumPy tích hợp liền mạch và nhanh chóng với nhiều loại cơ sở dữ liệu.

### Thư viện Mahotas.

Mahotas là một thư viện xử lý ảnh và thị giác máy tính cho python. Bao gồm nhiều thuật toán đước triển khai trong c++ cho tốc độ trong khi hoạt động trong các mảng khó hiểu và với giao diện python rất sạch.

Mahotas hiện có hơn 100 chức năng xử lý hình ảnh và thị giác máy tính và nó tiếp tục phát triển. Một số ví dụ về chức năng mahota:

### Thư viện Keras.

Keras là một library được phát triển vào năm 2015 bởi françois chollet, là một kỹ sư nghiên cứu deep learning tại google. Nó là một open source cho neural network được viết bởi ngôn ngữ python. Keras là một api bậc cao có thể sử dụng chung với các thư viện deep learning nổi tiếng như tensorflow(được phát triển bởi gg), cntk(được phát triển bởi microsoft),theano(người phát triển chính yoshua bengio).

### Thư viện Tensorflow.

Tensorflow là một thư viện phần mềm nguồn mở để tính toán số bằng các biểu đồ luồng dữ liệu. Các nút trong biểu đồ đại diện cho các hoạt động toán học, trong khi các cạnh của biểu đồ đại diện cho các mảng dữ liệu đa chiều (tenor) được truyền thông giữa chúng. Kiến trúc linh hoạt cho phép bạn triển khai tính toán cho một hoặc nhiều CPU hoặc GPU trong máy tính để bàn, máy chủ hoặc thiết bị di động với một API duy nhất. Tensorflow ban đầu được phát triển bởi các nhà nghiên cứu và kỹ sư làm việc trong Nhóm Google Brain trong tổ chức nghiên cứu Máy thông minh của Google nhằm mục đích tiến hành học máy và nghiên cứu mạng nơ-ron sâu, nhưng hệ thống này đủ để áp dụng trong nhiều lĩnh vực khác như tốt.

Tensorflow mô tả lại dòng chảy của dữ liệu thông qua graph nên mỗi một điểm giao cắt trong graph thì được gọi là Node. Tại sao điều này quan trọng thì là vì các Node chính là điểm đại diện cho việc thay đổi của dữ liệu nên việc lưu trữ lại tham chiếu của các Node này là rất quan trọng.

Tensorflow cung cấp một loại dữ liệu mới được gọi là Tensor. Trong thế giới của Tensorflow, mọi kiểu dữ liệu đều được quy về một mối được gọi là Tensor hay trong Tensorflow, tất cả các loại dữ liệu đều là Tensor. Vậy nên có thể hiểu được phần nào cái tên Tensorflow là một thư viện mô tả, điều chỉnh dòng chảy của các Tensor.

### Thư viện Scikit-Image.

Scikit-learn (viết tắt là sklearn) là một thư viện mã nguồn mở dành cho học máy một ngành trong trí tuệ nhân tạo, rất mạnh mẽ và thông dụng với cộng đồng Python, được thiết kế trên nền numpy và scipy. Scikit-learn chứa hầu hết các thuật toán machine learning hiện đại nhất, đi kèm với documentations, luôn được cập nhật.

Sklearn được xây dựng trên numpy và scipy nên để sử dụng sklearn, chúng ta bắt buộc phải có 2 packages này. Tất nhiên ta nên cài đặt matplotlib, một package không thể thiếu trong scientific plotting. Cuối cùng, chúng ta cần cài thêm package pandas, phục vụ cho data wrangling và analysis.

## Nghiên cứu thuật toán xử lý nhân dạng giới tính.

### Các thuật toán nghiên cứu.

#### Thuật toán PCA.

##### 2.4.1.1.1. Khái niệm về PCA trong bài toán bầu cử.

Chúng ta sẽ tìm hiểu về Phân tích thành phần chính (PCA) - một kỹ thuật giảm kích thước phổ biến trong Machine Learning. Mục tiêu của chúng tôi là hình thành một sự hiểu biết trực quan về PCA mà không đi sâu vào tất cả các chi tiết toán học.

Tại thời điểm viết bài này, dân số của Hoa Kỳ là khoảng 325 triệu. Bạn có thể nghĩ rằng hàng triệu người sẽ có hàng triệu ý tưởng, ý kiến ​​và suy nghĩ khác nhau, sau tất cả, chỉ có một người duy nhất đúng?

Sai rồi!

Con người giống như con cừu. Chúng tôi theo một bầy. Thật đáng buồn nhưng là sự thật.

Giả sử bạn chọn 20 câu hỏi chính trị hàng đầu ở Hoa Kỳ và yêu cầu hàng triệu người trả lời những câu hỏi này bằng cách sử dụng có hoặc không. Đây là vài ví dụ:

1. Bạn có hỗ trợ kiểm soát súng?   
2. Bạn có ủng hộ quyền phá thai của phụ nữ không?

Vân vân và vân vân. Về mặt kỹ thuật, bạn có thể nhận được 2^{20}các bộ câu trả lời khác nhau vì bạn có 20 câu hỏi và mỗi câu hỏi phải được trả lời bằng cách sử dụng có hoặc không.

Trong thực tế, bạn sẽ nhận thấy bộ câu trả lời nhỏ hơn nhiều. Trên thực tế, bạn thay thế 20 câu hỏi hàng đầu bằng một câu hỏi duy nhất

Bạn ủng hộ Đảng Dân Chủ hay Đảng Cộng Hòa ?

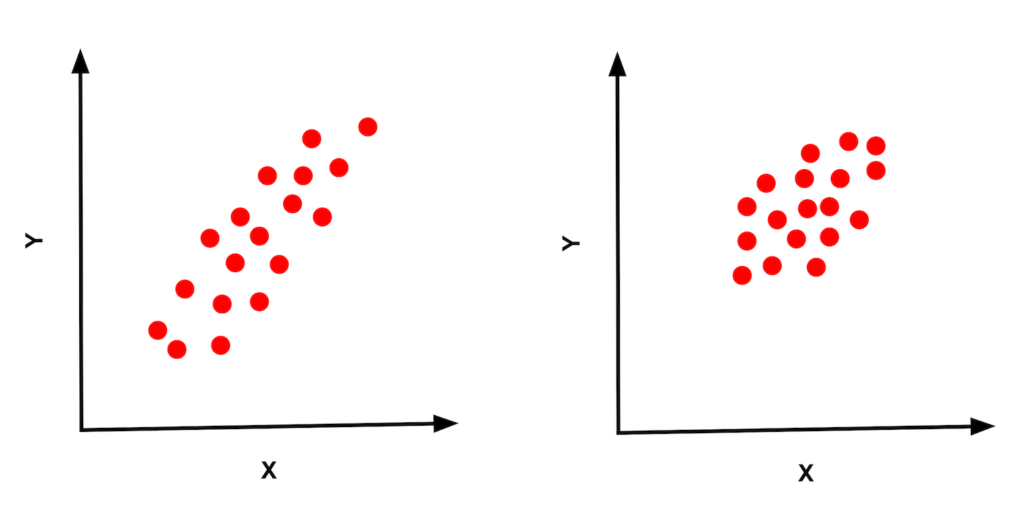
và dự đoán chính xác câu trả lời cho các câu hỏi còn lại với độ chính xác cao. Vì vậy, dữ liệu 20 chiều này được nén thành một chiều và không có nhiều thông tin bị mất!

Đây chính xác là những gì PCA cho phép chúng tôi làm. Trong dữ liệu đa chiều, nó sẽ giúp chúng tôi tìm thấy các kích thước hữu ích nhất và chứa nhiều thông tin nhất. Nó sẽ giúp chúng tôi trích xuất thông tin cần thiết từ dữ liệu bằng cách giảm kích thước.

Chúng ta sẽ cần một số công cụ toán học để hiểu PCA và hãy bắt đầu với một khái niệm quan trọng trong thống kê được gọi là **phương sai** .

##### 2.4.1.1.2. Phương sai là gì ?

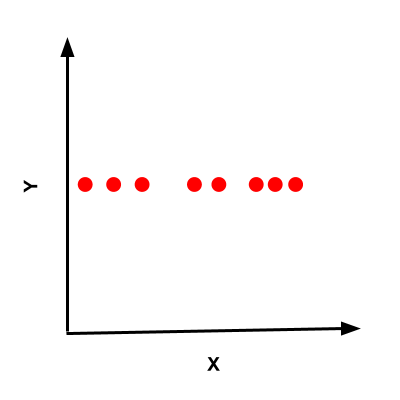
Phương sai đo lường sự lan truyền của dữ liệu. Trong Hình 1 (a), các điểm có phương sai cao vì chúng được trải ra, nhưng trong Hình 1 (b), các điểm có phương sai thấp vì chúng gần nhau.



**Hình 1. (a) Trái: Dữ liệu phương sai cao (b) Phải: Dữ liệu phương sai thấp**

Ngoài ra, lưu ý rằng trong Hình 1 (a) phương sai không giống nhau theo mọi hướng. Hướng của phương sai tối đa là đặc biệt quan trọng. Hãy xem tại sao.

**Tại sao chúng ta quan tâm đến hướng của phương sai tối đa ?**

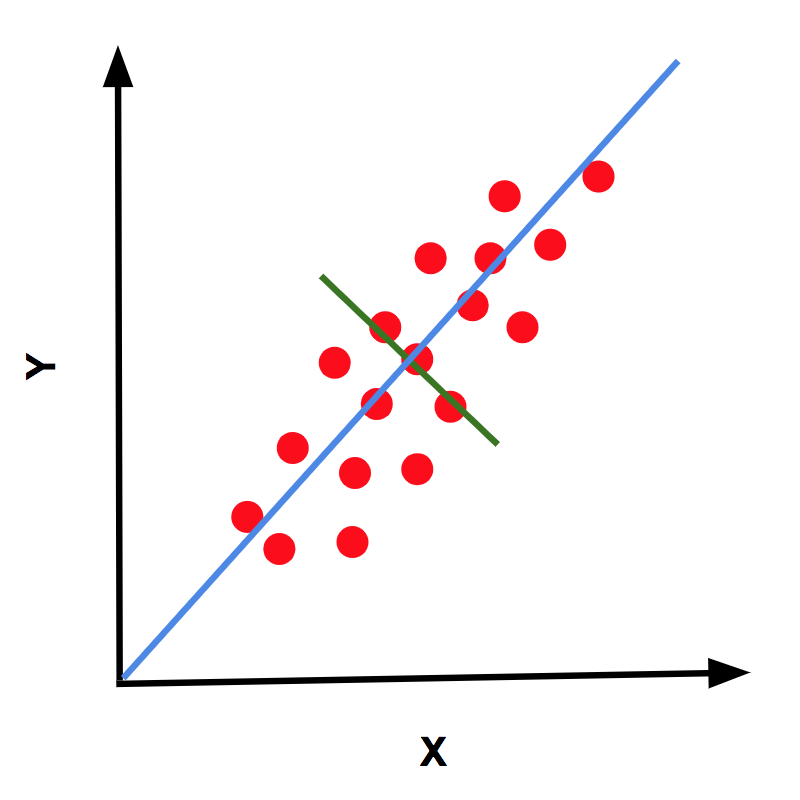


**Hình 2: Phương sai bằng không dọc theo trục y**

Phương sai mã hóa thông tin có trong dữ liệu. Ví dụ: nếu bạn có dữ liệu 2D được biểu thị bằng các điểm có (x,y)tọa độ. Đối với n điểm như vậy, bạn cần 2n số để biểu thị dữ liệu này. Hãy xem xét một trường hợp đặc biệt trong đó với mỗi điểm dữ liệu, giá trị dọc theo trục y là 0 (hoặc không đổi). Điều này được thể hiện trong Hình 2

Thật công bằng khi nói rằng không có (hoặc rất ít) thông tin dọc theo trục y. Bạn có thể biểu diễn ngắn gọn dữ liệu này bằng cách sử dụng n số để biểu thị giá trị của nó dọc theo trục x và chỉ có 1 số chung để biểu thị hằng số dọc theo trục y. Do có nhiều phương sai dọc theo trục x, nên có nhiều thông tin hơn và do đó chúng ta phải sử dụng nhiều số hơn để biểu thị dữ liệu này. Mặt khác, do không có phương sai dọc theo trục y, nên một số duy nhất có thể được sử dụng để thể hiện tất cả thông tin có trong n điểm dọc theo trục này.

##### 2.4.1.1.3. PCA là gì ?

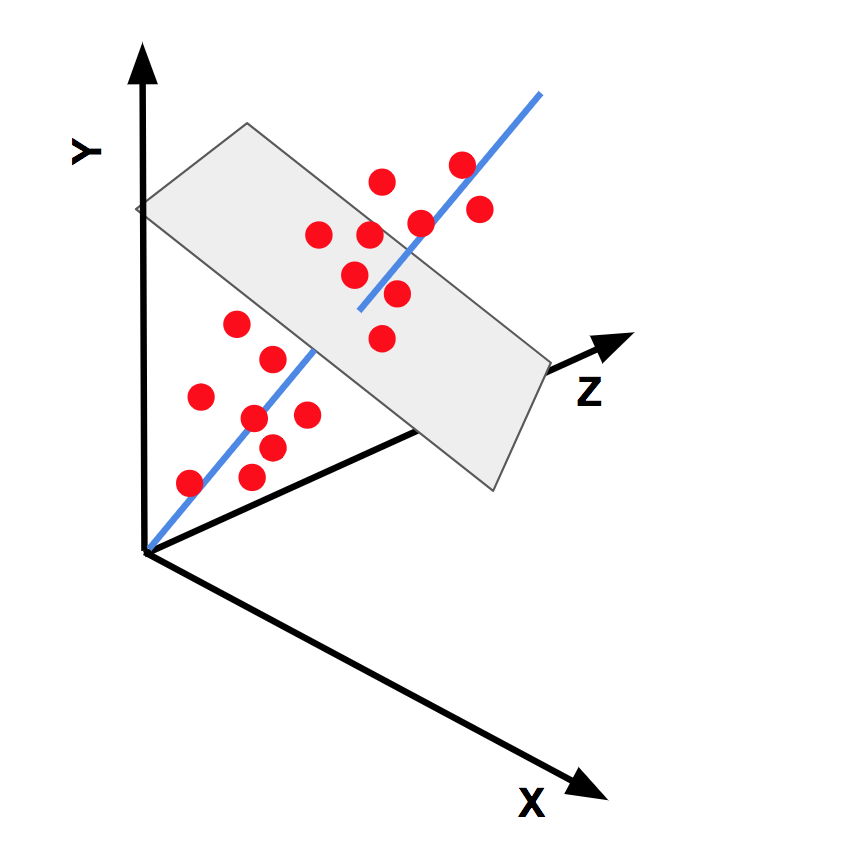


**Hình 3: Phân tích thành phần chính trong 2D**

Bây giờ hãy xem xét một bộ dữ liệu phức tạp hơn một chút được hiển thị trong Hình 3 bằng cách sử dụng các dấu chấm màu đỏ. Dữ liệu được lan truyền trong một hình dạng gần giống như hình elip. Trục chính của hình elip là hướng của phương sai tối đa và như chúng ta biết bây giờ, đó là hướng của thông tin tối đa. Hướng này, được biểu thị bằng đường màu xanh trong Hình 3, được gọi là thành phần chính đầu tiên của dữ liệu.

Thành phần chính thứ hai là hướng của phương sai tối đa vuông góc với hướng của thành phần chính thứ nhất. Trong 2D, chỉ có một hướng vuông góc với thành phần chính thứ nhất và đó là thành phần chính thứ hai. Điều này được hiển thị trong Hình 3 bằng cách sử dụng một đường màu xanh lá cây.

Bây giờ hãy xem xét việc truyền dữ liệu 3D giống như một ellipsoid (được hiển thị trong Hình 4). Thành phần chính đầu tiên được đại diện bởi dòng màu xanh. Có toàn bộ mặt phẳng vuông góc với thành phần chính đầu tiên. Do đó, có các hướng vô hạn để lựa chọn và thành phần chính thứ hai được chọn là hướng của phương sai tối đa trong mặt phẳng này. Như bạn có thể đoán, thành phần chính thứ ba chỉ đơn giản là hướng vuông góc với cả hai thành phần chính thứ nhất và thứ hai.



**Hình 4: Trong 3D, có một mặt phẳng vuông góc với thành phần chính đầu tiên. Thành phần chính thứ hai là hướng của phương sai cực đại trong mặt phẳng này**

##### 2.4.1.1.4. PCA và giảm kích thước ảnh.

Trong phần đầu của bài đăng này, chúng tôi đã đề cập rằng động lực lớn nhất cho PCA là giảm kích thước. Nói cách khác, chúng tôi muốn nắm bắt thông tin có trong dữ liệu bằng cách sử dụng ít kích thước hơn.

Hãy xem xét dữ liệu 3D được hiển thị trong Hình 4. Mỗi điểm dữ liệu có 3 tọa độ - x, y và z đại diện cho các giá trị của chúng dọc theo trục X, Y và Z. Lưu ý ba thành phần chính không là gì ngoài một bộ trục mới vì chúng vuông góc với nhau. Chúng ta có thể gọi các trục này được hình thành bởi các thành phần chính X ', Y' và Z '.

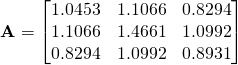
Trên thực tế, bạn có thể xoay trục X, Y, Z cùng với tất cả các điểm dữ liệu trong 3D sao cho trục X thẳng hàng với thành phần chính thứ nhất, trục Y thẳng hàng với thành phần chính thứ hai và trục Z thẳng hàng với các thành phần chính thứ ba. Bằng cách áp dụng phép quay này, chúng ta có thể biến đổi bất kỳ điểm (x, y, z) nào trong hệ tọa độ XYZ thành một điểm (x ', y', z ') trong hệ tọa độ X'Y'Z' mới. Đó là cùng một thông tin được trình bày trong một hệ tọa độ khác, nhưng cái hay của hệ tọa độ mới này X'Y'Z 'là thông tin chứa trong X' là tối đa, tiếp theo là Y 'và sau đó là Z'. Nếu bạn thả tọa độ z 'cho mọi điểm (x', y ', z'), chúng tôi vẫn giữ lại hầu hết thông tin, nhưng bây giờ chúng tôi chỉ cần hai chiều để thể hiện dữ liệu này.

Điều này có thể trông giống như một khoản tiết kiệm nhỏ, nhưng nếu bạn có dữ liệu 1000 chiều, bạn có thể giảm đáng kể kích thước xuống chỉ còn 20 chiều. Ngoài việc giảm kích thước, PCA cũng sẽ loại bỏ nhiễu trong dữ liệu.

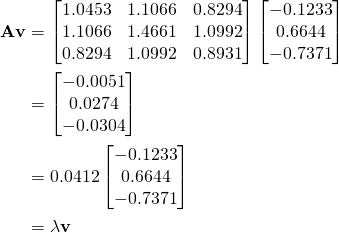
##### 2.4.1.1.5. Vertor Eigen và giá trị Eigen của ma trận là gì ?

Trong phần tiếp theo, chúng tôi sẽ giải thích từng bước cách tính PCA, nhưng trước khi chúng tôi làm điều đó, chúng tôi cần phải hiểu các vectơ Eigen và giá trị Eigen là gì.

Hãy xem xét ma trận 3 × 3 sau đây :



Chúng ta hãy nhân ma trận \ mathbf {A}với vectơ \ mathbf {v}và xem tại sao tôi gọi vectơ này là đặc biệt.



nơi \ lambda = 0,0412. Lưu ý, khi chúng ta nhân ma trận \ mathbf {A}với vectơ \ mathbf {v}, nó chỉ thay đổi độ lớn của vectơ \ mathbf {v}bằng \ lambdanhưng không thay đổi hướng của nó. Chỉ có 3 hướng (bao gồm cả \ mathbf {v}trong ví dụ trên) mà ma trận \ mathbf {A}sẽ hoạt động như một số nhân vô hướng. Ba hướng này là các vectơ Eigen của ma trận và các bội số vô hướng là các giá trị Eigen.

Vì vậy, một vectơ Eigen \ mathbf {v}của ma trận \ mathbf {A}là một vectơ có hướng không thay đổi khi ma trận được nhân với nó. Nói cách khác,

bắt đầu {align *} hhh {{}} = lambda

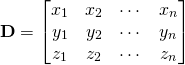
trong đó \ lambdamột vô hướng (chỉ là một số) và được gọi là giá trị Eigen tương ứng với vectơ Eigen\ mathbf {v}

##### 2.4.1.1.6. Làm thế nào để tính toán PCA.

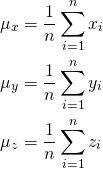
Thông thường, bạn có thể dễ dàng tìm thấy các thành phần chính của dữ liệu đã cho bằng cách sử dụng gói đại số tuyến tính mà bạn chọn. Trong bài tiếp theo, chúng ta sẽ tìm hiểu cách sử dụng lớp PCA trong OpenCV. Ở đây, chúng tôi giải thích ngắn gọn các bước để tính PCA để bạn hiểu được cách nó được thực hiện trong các gói toán khác nhau.

Dưới đây là các bước để tính PCA. Chúng tôi đã giải thích các bước sử dụng dữ liệu 3D để đơn giản, nhưng ý tưởng tương tự áp dụng cho bất kỳ số lượng kích thước nào.

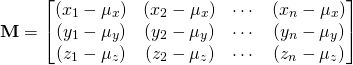
1. **Lắp ráp một ma trận dữ liệu** : Bước đầu tiên là tập hợp tất cả các điểm dữ liệu thành một ma trận trong đó mỗi cột là một điểm dữ liệu. Một ma trận dữ liệu \ mathbf {D}, các nđiểm 3D sẽ giống như thế này



1. **Tính giá trị trung bình** : Bước tiếp theo là tính giá trị trung bình (trung bình) của tất cả các điểm dữ liệu. Lưu ý, nếu dữ liệu là 3D, giá trị trung bình cũng là điểm 3D với tọa độ x, y và z. Tương tự, nếu dữ liệu là m chiều, giá trị trung bình cũng sẽ là m chiều. Giá trị trung bình (\mu_x, \mu_y, \mu_z) là :



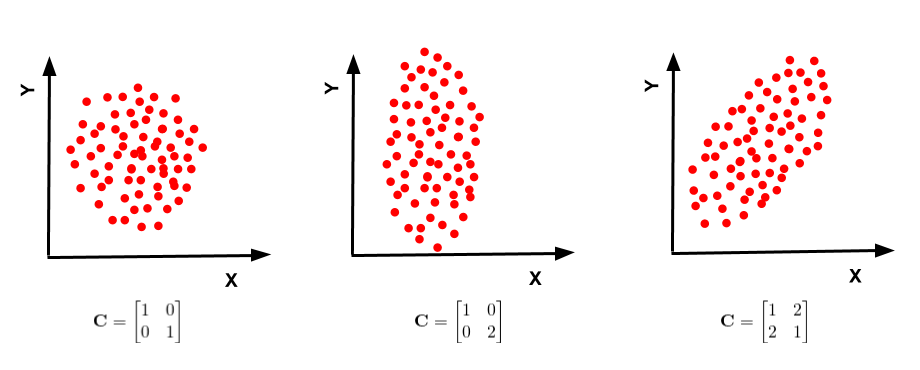
1. **Trừ trung bình từ ma trận dữ liệu** : Tiếp theo chúng ta tạo một ma trận khác \ mathbf {M} bằng cách trừ giá trị trung bình từ mọi điểm dữ liệu của \ mathbf {D}



1. **Tính ma trận hiệp phương sai** : Hãy nhớ rằng chúng ta muốn tìm hướng của phương sai tối đa. Ma trận hiệp phương sai nắm bắt thông tin về sự lan truyền của dữ liệu. Các phần tử đường chéo của ma trận hiệp phương sai là các phương sai dọc theo trục X, Y và Z. Các phần tử nằm ngoài đường chéo biểu thị hiệp phương sai giữa hai chiều (X và Y, Y và Z, Z và X). Ma trận hiệp phương sai, \ Mathbf {C}được tính bằng cách sử dụng sản phẩm sau.

\ started {align *} \ mathbf {C} = \ mathbf {M} \ mathbf {M} ^ T \ end {align *}

trong đó, Tđại diện cho hoạt động chuyển vị. Ma trận \ Mathbf {C}có kích thước m \ lần mtrong đó mlà số lượng kích thước (là 3 trong ví dụ của chúng tôi). Hình 5 cho thấy ma trận hiệp phương sai thay đổi như thế nào tùy thuộc vào sự lan truyền dữ liệu theo các hướng khác nhau.



**Hình 5: Trái: Khi dữ liệu được trải đều theo mọi hướng, ma trận hiệp phương sai có các phần tử đường chéo bằng nhau và các phần tử ngoài đường chéo bằng không. Trung tâm: Khi sự lan truyền dữ liệu được kéo dài dọc theo một trong các trục, các phần tử đường chéo là không bằng nhau, nhưng các phần tử đường chéo là bằng không. Phải: Nói chung, ma trận hiệp phương sai có cả hai phần tử đường chéo và đường chéo.**

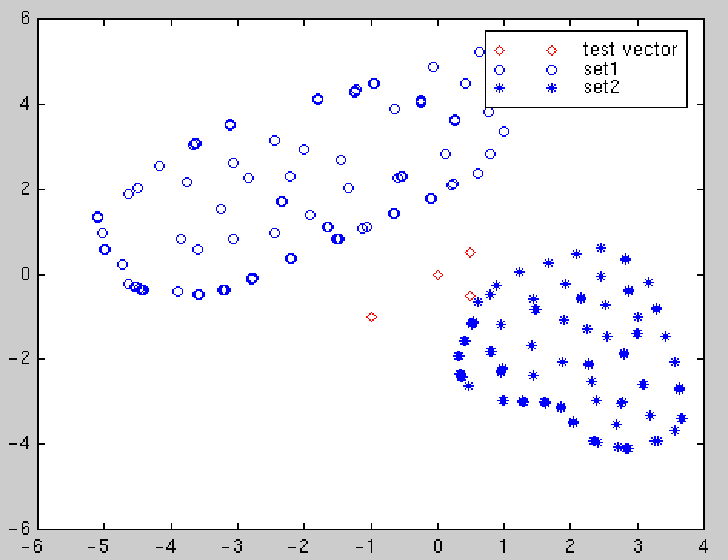
1. **Tính toán các vectơ Eigen và giá trị Eigen của ma trận hiệp phương sai** : Các thành phần chính là các vectơ Eigen của ma trận hiệp phương sai. Thành phần chính đầu tiên là vectơ Eigen tương ứng với giá trị Eigen lớn nhất, thành phần chính thứ hai là vectơ Eigen tương ứng với giá trị Eigen lớn thứ hai, v.v.

#### Thuật toán LDA

##### 2.4.1.2.1. Khái Niệm LDA.

Có rất nhiều phương pháp khả thi để phân lớp dữ liệu mà phân tích khác biệt tuyến tính (Linear Discriminant Analysis) là phương pháp điển hình để giảm số chiều đặc trưng và phân lớp dữ liệu. LDA được sử dụng đối với tập mẫu có phân bố độc lập ngẫu nhiên. Cách làm này cực đại hóa tỉ lệ của between-class variance và within-class variance trong bất kỳ bộ dữ liệu đặc thù nào, theo đó, đảm bảo hiệu quả tối ưu. Phương pháp LDA được áp dụng lần đầu tiên cho bài toán nhận dạng tiếng nói. Chúng tôi quyết định đề xuất một thuật toán LDA cho bài toán nhận dạng mặt người với hy vọng cung cấp một phương pháp phân lớp tốt hơn so với PCA. Điều khác biệt trước tiên của LDA và PCA là PCA nặng về phân lớp đặc trưng và LDA nặng về phân lớp dữ liệu.

Trong PCA, trạng thái và vị trí của dữ liệu gốc bị thay đổi khi được chiếu sang một không gian khác, trái lại LDA không thay đổi vị trí mà chỉ cố gắng cung cấp thêm một phân lớp và vẽ ra vùng quyết định giữa 2 lớp cho trước. Cách làm này giúp cho to hiểu rõ hơn về sự phân bố của đặc trưng dữ liệu. Hình 0-2 là ví dụ để giải thích và minh họa cho lý thuyết của LDA.



Hình 0‑2 Tập dữ liệu học và vector test ở trạng thái gốc

##### 2.4.1.2.2. Hướng tiếp cận LDA.

**Biến đổi phụ thuộc lớp :**

Hướng tiếp cận này đòi hỏi phải cực đại hóa tỉ lệ giữa between class variance và within class variance. Mục đích chính là cực đại hóa tỉ lệ này để tìm ra sự phân lớp chính xác. Hướng tiếp cận này đòi hỏi sử dụng n tiêu chuẩn khách quan để biến đổi tập dữ liệu gồm n phân lớp về các tập độc lập.

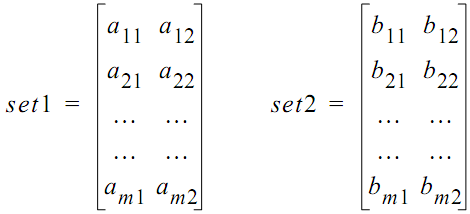
**Biến đổi không phụ thuộc lớp :**

Hướng tiếp cận này đòi hỏi phải cực đại hóa tỉ lệ giữa overall variance và within class variance. Cách này chỉ sử dụng 1 tiêu chuẩn khách quan để biến đổi dữ liệu và kể từ đây mọi điểm dữ liệu bất chấp lớp của nó sẽ được biến đổi bằng phép biến đổi này. Trong phương pháp này, mỗi lớp được xem như là một lớp riêng lẻ so với các lớp còn lại.

##### 2.4.1.2.3. Làm thế nào để tính toán LDA.

Trong phần này, những phép toán được sử dụng trong LDA sẽ được phân tích theo tập mẫu trong hình C - 1. Để cho dễ hiểu, thuật toán sẽ được áp dụng cho bài toán phân lớp 2 lớp. Mỗi tập dữ liệu có 100 điểm 2D. Chú ý rằng công thức toán học của chiến lược phân lớp này sẽ được trình bày bằng Matlab.

Công thức hóa toàn bộ tập dữ liệu và tập test, những thứ mà sẽ được phân lớp trong không gian gốc. Tập dữ liệu và test được công thức hóa, đồ thị của chúng được minh họa trong hình F - 1. Để tiện việc theo dõi chúng tôi xin mô tả tập dữ liệu dưới dạng ma trận các đặc trưng như sau:



Tính trung bình của mỗi tập dữ liệu và trung bình của tất cả các bộ dữ liệu. Gọi µ1 và µ2 lần lượt là vector trung bình của tập 1 và tập 2. Trung bình của tất cả các bộ dữ liệu được tính bằng cách kết hợp tập 1 và 2 theo công thức (C.2\_1).

 (F.2\_1)  
với p1 và p2 là xác suất tiên nghiệm của các lớp. Được tính bằng công thức:  
p = (Số mẫu của lớp) / (Tổng số mẫu các lớp) (C.2\_2)  
Trong trường hợp bài toán phân 2 lớp đơn giản này, hệ số ấy là 0.5.

Trong LDA, within-class và between-class scatter được sử dụng để xác định tiêu chuẩn phân lớp. Within-class scatter là kỳ vọng hiệp phương sai của mỗi lớp, được minh họa trong công thức (3) và (4).

(C.2\_3)

Do đó, trong bài toán phân 2 lớp,

(C.2\_4)

Tất cả ma trận hiệp phương sai đều đối xứng. Gọi cov1 và cov2 lần lượt là hiệp phương sai của tập 1 và tập 2. Ma trận hiệp phương sai được tính bởi công thức sau.

(C.2\_5)

Chú ý rằng Sb có thể được hiểu như là môt hiệp phương sai của tập dữ liệu mà mỗi phần tử là 1 vector trung bình của 1 phân lớp. Như đã nói ở trên, tiêu chuẩn khách quan trong LDA là tỉ lệ giữa between-class scatter và within-class scatter. Lời giải được hình thành bằng cách cực đại hóa tỉ lệ này để xác định không gian biến đổi. Tuy nhiên trong phương pháp biến đổi phụ thuộc lớp, tiêu chuẩn khách quan được tính bằng công thức (C.2\_6). Cần phải lưu ý rằng nếu LDA theo hướng phụ thuộc lớp, để phân L lớp, cần 1 tiêu chuẩn khách quan cho mỗi lớp. Những nhân tố khách quan trong trường hợp phụ thuộc lớp được tính bởi

 (C.2\_6)

Trong phép biến đổi không phụ thuộc lớp, tiêu chuẩn khách quan được tính bởi  (C.2\_7)

Theo như định nghĩa, một vector riêng của 1 phép biến đổi tượng trưng cho 1 không gian con 1 chiều của không gian vector mà phép biến đổi được áp dụng. Một tập các vector riêng này ứng với các giá trị riêng khác 0 là độc lập tuyến tính và bất biến trong phép biến đổi. Vì vậy bất kỳ không gian vector nào cũng có thể được biểu diễn bằng sự kết hợp tuyến tính của các vector riêng. Một sự phụ thuộc tuyến tính giữa các đặc trưng được chỉ ra bởi 1 trị riêng quyết định. Để tìm được 1 tập tường minh các đặc trưng, chỉ những vector riêng ứng với mỗi giá trị riêng khác 0 được xem xét và số còn lại bị bỏ qua. Trong bối cảnh của LDA, những phép biến đổi được thực hiện trên ma trận vector riêng của các tiêu chuẩn khác nhau định nghĩa trong công thức (C.2\_8) và (C.2\_9).

Để giải quyết bài toán phân L lớp, chúng tôi luôn có L-1 giá trị riêng khác 0. Điều này tượng trưng cho những sự ràng buộc trên các vector trung bình của các lớp trong công thức (C.2\_2). Các vector riêng tương ứng với các trị riêng khác 0 có ý nghĩa trong phép biến đổi.

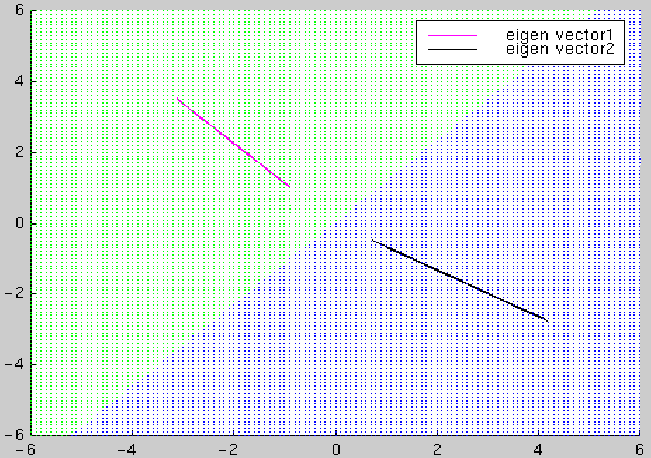
Trong ví dụ bài toán phân 2 lớp của chúng ta, hình 2 và 3 thể hiện phương hướng của tín hiệu vector riêng với sự khác biệt rõ nhất. Để xây dựng những ma trận biến đổi, chúng tôi biến đổi tập dữ liệu bằng phép biến đổi LDA đơn lẻ hoặc phép biến đổi lớp xác định mà mỗi trường hợp có thể có. Theo hình vẽ có thể quan sát được điều đó, biến đổi tất cả tập dữ liệu về 1 trục ranh giới phân lớp dữ liệu. Vùng quyết định trong không gian biến đổi là 1 đường thẳng phân chia tập dữ liệu đã được biến đổi .

Đối với LDA phụ thuộc lớp :  
 (C.2\_8)

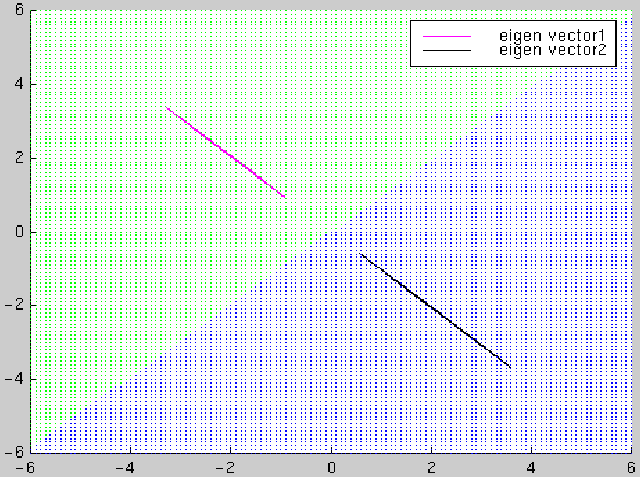
Đối với LDA không phụ thuộc lớp :  
 (C.2\_9)

Khi đã thực hiện xong phép biển đổi LDA, khoảng cách Euclidean hoặc khoảng cách RMS được dùng để phân lớp các điểm dữ liệu. Khoảng cách Euclidean được tính với µntrans là vector trung bình cả tập đã biến đổi, n là chỉ số của lớp và x là vector test. Vì thế đối với n lớp, n khoảng cách Euclidean được xác định cho mỗi điểm.

Sau khi đã chiếu tất cả vào không gian LDA, tiến hành đo khoảng cách từ mẫu test đến các mẫu huấn luyện để quyết định xem nó thuộc lớp nào.



*Hình 0‑3 Hướng của Vector riêng trong mẫu phụ thuộc lớp*



*Hình 0‑4 Hướng của Vector riêng trong mẫu không phụ thuộc lớp*

 (C.2\_10)

**Tóm lại các bước cần tiến hành để áp dụng phương pháp LDA cho ảnh:**

* 1. Biểu diễn các ảnh dưới dạng các vector cột.
  2. Tính các vector trung bình của mỗi lớp (μi):  
      với M là số ảnh của lớp i
  3. Tính vector trung bình của tất cả các ảnh trong tất cả các lớp (μ):  
      với C là số lớp, M là số ảnh của lớp i, N là tổng số ảnh của tất cả các lớp.
  4. Tính “ma trận quan hệ bên trong” (within-class scatter matrix) (Sw):  
     
  5. Tính “ma trận quan hệ bên cạnh” (between-class scatter matrix) (Sb):  
     
  6. Tính ma trận chiếu (WLDA):  
     W = eig(Sw-1.Sb)

Sau khi đã tính ra được ma trận chiếu, ta tiến hành chiếu toàn bộ ảnh huấn luyện cũng như ảnh test vào vùng không gian của LDA theo công thức: Y = WT.X với X là ảnh ban đầu, WT là ma trận chuyển vị của WLDA và Y là ảnh sau khi đã được chiếu vào không gian LDA. Cuối cùng ta thực hiện tính khoảng cách từ ảnh test tới ảnh huấn luyện trong không gian LDA để quyết định xem ảnh test thuộc lớp nào. Hoặc cũng có thể sử dụng vector ảnh sau khi đã được chiếu vào không gian LDA để làm đầu vào cho một phương pháp phân lớp khác như mạng nơron nhân tạo, SVM ….

#### Thuật toán Eigen Faces

##### 2.4.1.2.1. Engen Faces là gì ?

Eigenfaces là hình ảnh có thể được thêm vào khuôn mặt trung bình (trung bình) để tạo hình ảnh khuôn mặt mới. Chúng ta có thể viết điều này một cách toán học :

Trong đó :

là một gương mặt mới.   
F_m là trung bình hoặc khuôn mặt trung bình,   
F_i là EigenFace,   
\ alpha_i là bội số vô hướng mà chúng ta có thể chọn để tạo khuôn mặt mới. Họ có thể tích cực hoặc tiêu cực.



Eigenfaces được tính bằng cách ước tính các thành phần chính của bộ dữ liệu hình ảnh khuôn mặt. Chúng được sử dụng cho các ứng dụng như Nhận diện khuôn mặt và Nhận diện khuôn mặt.

##### 2.4.1.2.2. Vertor hình ảnh.

Làm thế nào để chúng ta biểu diễn một hình ảnh như một điểm trong một không gian chiều cao hơn? Hãy xem xét một ví dụ.

Một hình ảnh màu 100 x 100 không là gì ngoài một mảng gồm 100 x 100 x 3 (một cho mỗi kênh màu R, G, B). Thông thường, chúng tôi muốn nghĩ mảng 100 x 100 x 3 là mảng 3D, nhưng bạn có thể nghĩ nó là mảng 1D dài bao gồm 30.000 phần tử.

Bạn có thể nghĩ mảng 30k phần tử này là một điểm trong không gian 30 chiều giống như bạn có thể tưởng tượng một mảng gồm 3 số (x, y, z) là một điểm trong không gian 3D!

Làm thế nào để bạn hình dung một không gian 30k chiều? Bạn không thể. Hầu hết thời gian bạn có thể xây dựng đối số của mình như thể chỉ có ba chiều và thông thường (nhưng không phải luôn luôn), chúng cũng đúng với không gian chiều cao hơn.

##### 2.4.1.2.3. Cách tính toán Eigen Faces.

Để tính toán EigenFaces, chúng ta cần sử dụng các bước sau.

1. **Lấy bộ dữ liệu hình ảnh khuôn mặt** : Chúng tôi cần một bộ sưu tập các hình ảnh khuôn mặt có chứa các loại khuôn mặt khác nhau. Trong bài đăng này, chúng tôi đã sử dụng khoảng 200 hình ảnh từ [CelebA](http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html) .
2. **Căn chỉnh và thay đổi kích thước hình ảnh** : Tiếp theo chúng ta cần căn chỉnh và thay đổi kích thước hình ảnh để trung tâm của mắt được căn chỉnh trong tất cả các hình ảnh. Điều này có thể được thực hiện bằng cách đầu tiên tìm thấy các [mốc trên khuôn mặt](https://www.learnopencv.com/facial-landmark-detection) . Trong bài viết này, chúng tôi đã sử dụng hình ảnh phù hợp được cung cấp trong [CelebA](http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html) . Tại thời điểm này, tất cả các hình ảnh trong bộ dữ liệu nên có cùng kích thước.
3. **Tạo ma trận dữ liệu** : Tạo ma trận dữ liệu chứa tất cả các hình ảnh dưới dạng vectơ hàng. Nếu tất cả các hình ảnh trong bộ dữ liệu có kích thước 100 x 100 và có 1000 hình ảnh, chúng ta sẽ có một ma trận dữ liệu có kích thước 30k x 1000.
4. **Tính toán vectơ trung bình [Tùy chọn]** : Trước khi thực hiện PCA trên dữ liệu, chúng ta cần trừ đi vectơ trung bình. Trong trường hợp của chúng tôi, vectơ trung bình sẽ là một vectơ 30k x 1 hàng được tính bằng cách lấy trung bình tất cả các hàng của ma trận dữ liệu. Lý do tính toán vectơ trung bình này là không cần thiết để sử dụng lớp PCA của OpenCV là vì OpenCV tính toán thuận tiện giá trị trung bình cho chúng tôi nếu vectơ không được cung cấp. Đây có thể không phải là trường hợp trong các gói đại số tuyến tính khác.
5. **Tính toán các thành phần chính** : Các thành phần chính của ma trận dữ liệu này được tính bằng cách tìm Eigenvector của ma trận hiệp phương sai. May mắn thay, lớp PCA trong OpenCV xử lý tính toán này cho chúng tôi. Chúng ta chỉ cần cung cấp datamatrix và xuất hiện một ma trận chứa Eigenvector.
6. **Định hình lại Eigenvertor để lấy EigenFaces** : Eigenvector thu được sẽ có độ dài 30.000 nếu tập dữ liệu của chúng tôi chứa hình ảnh kích thước 100 x 100 x 3. Chúng tôi có thể định hình lại các Eigenvertor này thành hình ảnh 100 x 100 x 3 để có được EigenFace.

#### Thuật toán LPBH.

##### 2.4.1.4.1. LPBH là gì ?

**Mô hình nhị phân cục bộ** (LBP) là một toán tử kết cấu đơn giản nhưng rất hiệu quả, gắn nhãn các pixel của hình ảnh bằng cách chia vùng lân cận của mỗi pixel và coi kết quả là số nhị phân.

LPBH được mô tả lần đầu tiên vào năm 1994 (LBP) và từ đó được phát hiện là một tính năng mạnh mẽ để phân loại kết cấu. Người ta đã xác định thêm rằng khi LBP được kết hợp với mô tả biểu đồ định hướng độ dốc (HOG), nó sẽ cải thiện đáng kể hiệu suất phát hiện trên một số bộ dữ liệu.

Sử dụng LBP kết hợp với biểu đồ, chúng ta có thể biểu diễn hình ảnh khuôn mặt bằng một vectơ dữ liệu đơn giản.

Vì LBP là một mô tả trực quan, nó cũng có thể được sử dụng cho các tác vụ nhận dạng khuôn mặt, như có thể thấy trong phần giải thích từng bước sau đây.

##### 2.4.1.4.2. LPBH hoạt động như thế nào ?

Bây giờ chúng ta đã biết thêm một chút về nhận dạng khuôn mặt và LBPH hãy xem các bước của thuật toán:

**Tham số** : LBPH sử dụng 4 tham số:

**Bán kính** : bán kính được sử dụng để xây dựng mô hình nhị phân cục bộ tròn và biểu thị bán kính xung quanh pixel trung tâm. Nó thường được đặt thành 1.

**Lân cận:** số lượng điểm mẫu để xây dựng mô hình nhị phân cục bộ tròn. Hãy ghi nhớ: có càng nhiều điểm mẫu, giá trị tính toán *(mọi khi hay dịch là chi phí tính toán)* càng cao. Nó thường được đặt thành 8.

**Lưới X** : số lượng ô theo hướng ngang. Càng nhiều ô, lưới càng mịn, thì chiều của vectơ đặc trưng có kết quả càng cao. Nó thường được đặt thành 8.

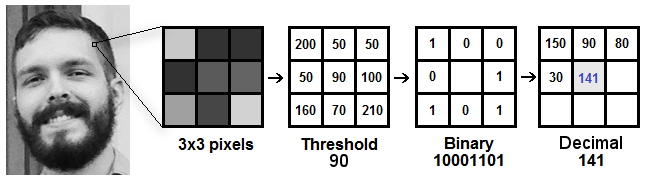
**Lưới Y** : số lượng ô theo hướng dọc. Càng nhiều ô, lưới càng mịn, thì chiều của vectơ đặc trưng có kết quả càng cao. Nó thường được đặt thành 8.

Đừng lo lắng về các thông số ngay, bạn sẽ hiểu chúng sau khi đọc các bước tiếp theo.

**Đào tạo thuật toán :** Đầu tiên, chúng ta cần củng cố (training) thuật toán. Để làm như vậy, chúng ta cần sử dụng một bộ dữ liệu với hình ảnh khuôn mặt của những người mà chúng ta muốn nhận ra. Chúng ta cũng cần đặt ID (có thể là số hoặc tên của người đó) cho mỗi hình ảnh, vì vậy thuật toán sẽ sử dụng thông tin này để nhận ra hình ảnh đầu vào và cung cấp cho bạn đầu ra. Hình ảnh của cùng một người phải có cùng ID. Với bộ củng cố *(bộ huấn luyện, dào tạo)* đã được xây dựng, hãy xem các bước tính toán LBPH.

**Áp dụng thao tác LBP** : Bước tính toán đầu tiên của LBPH là tạo một hình ảnh trung gian mô tả hình ảnh gốc theo cách tốt hơn, bằng cách làm nổi bật các đặc điểm khuôn mặt. Để làm như vậy, thuật toán sử dụng khái niệm cửa sổ trượt, dựa trên **bán kính** tham số và **các lân cận** .

Hình ảnh dưới đây cho thấy phương pháp này:



Dựa trên hình ảnh trên, hãy chia nó thành nhiều bước nhỏ để chúng ta có thể hiểu nó dễ dàng:

Giả sử chúng ta có một hình ảnh khuôn mặt trong thang độ xám.

Chúng ta có thể lấy một phần của hình ảnh này dưới dạng cửa sổ 3x3 pixel.

Nó cũng có thể được biểu diễn dưới dạng ma trận 3x3 chứa cường độ của từng pixel (0 ~ 255).

Sau đó, chúng ta cần lấy giá trị trung tâm của ma trận được sử dụng làm ngưỡng.

Giá trị này sẽ được sử dụng để xác định các giá trị mới từ 8 lân cận.

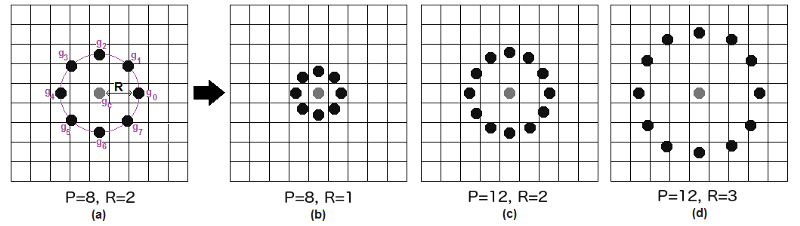
Đối với mỗi lân cận của giá trị trung tâm (ngưỡng), chúng ta đặt một giá trị nhị phân mới. Chúng ta đặt 1 cho các giá trị bằng hoặc cao hơn ngưỡng và 0 cho các giá trị thấp hơn ngưỡng.

Bây giờ, ma trận sẽ chỉ chứa các giá trị nhị phân (bỏ qua giá trị trung tâm). Chúng ta cần nối từng giá trị nhị phân từ mỗi vị trí từ dòng ma trận theo từng dòng thành một giá trị nhị phân mới (ví dụ 10001101). Lưu ý: một số tác giả sử dụng các cách tiếp cận khác để nối các giá trị nhị phân (ví dụ: theo chiều kim đồng hồ), nhưng kết quả cuối cùng sẽ giống nhau.

Sau đó, chúng ta chuyển đổi giá trị nhị phân này thành giá trị thập phân và đặt nó thành giá trị trung tâm của ma trận, đây thực sự là một pixel từ ảnh gốc.

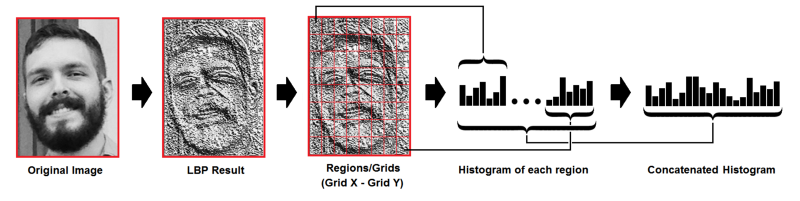
Vào cuối quy trình này (thủ tục LBP), chúng tao có một hình ảnh mới thể hiện rõ hơn các đặc điểm của hình ảnh gốc.

**Lưu ý** : Thủ tục LBP được mở rộng để sử dụng một số bán kính và lân cận khác nhau, nó được gọi là vòng tròn LBP.



Nó có thể được thực hiện bằng cách sử dụng **phép nội suy song tuyến tính** . Nếu một số điểm dữ liệu nằm giữa các pixel, nó sử dụng các giá trị từ 4 pixel gần nhất (2x2) để ước tính giá trị của điểm dữ liệu mới.

**Trích xuất biểu đồ** : Bây giờ, bằng cách sử dụng hình ảnh được tạo ở bước cuối cùng, chúng ta có thể sử dụng tham số **Grid X** và **Grid Y** để chia hình ảnh thành nhiều lưới, như có thể thấy trong hình ảnh sau:



Dựa vào hình ảnh trên, chúng ta có thể trích xuất biểu đồ của từng vùng như sau:

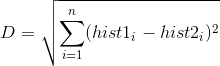
Khi chúng ta có một hình ảnh ở thang độ xám, mỗi biểu đồ (từ mỗi lưới) sẽ chỉ chứa 256 vị trí (0 ~ 255) đại diện cho các lần xuất hiện của từng cường độ pixel.

Sau đó, chúng ta cần nối từng biểu đồ để tạo ra biểu đồ mới và lớn hơn. Giả sử chúng ta có các lưới 8x8, chúng ta sẽ có các vị trí 8x8x256 = 16.384 trong biểu đồ cuối cùng. Biểu đồ cuối cùng đại diện cho các đặc điểm của hình ảnh gốc hình ảnh. Thuật toán LBPH là khá nhiều :

**Thực hiện nhận dạng khuôn mặt** : Trong bước này, thuật toán đã được củng cố (đào tạo) *không biết dùng từ gì cho hợp nghĩa*. Mỗi biểu đồ được tạo ra được sử dụng để thể hiện mỗi hình ảnh từ tập dữ liệu đào tạo. Vì vậy, với một hình ảnh đầu vào, chúng tôi thực hiện lại các bước cho hình ảnh mới này và tạo ra một biểu đồ đại diện cho hình ảnh.

Vì vậy, để tìm hình ảnh phù hợp với hình ảnh đầu vào, chúng ta chỉ cần so sánh hai biểu đồ và trả lại hình ảnh với biểu đồ gần nhất.

Chúng ta có thể sử dụng các cách tiếp cận khác nhau để so sánh biểu đồ (tính khoảng cách giữa hai biểu đồ), ví dụ: **khoảng cách euclidean** , **kiểm định bình phương** *chi square*, **giá trị tuyệt đối** , v.v. Trong ví dụ này, chúng ta có thể sử dụng khoảng cách Euclidean (khá nổi tiếng) theo công thức sau:



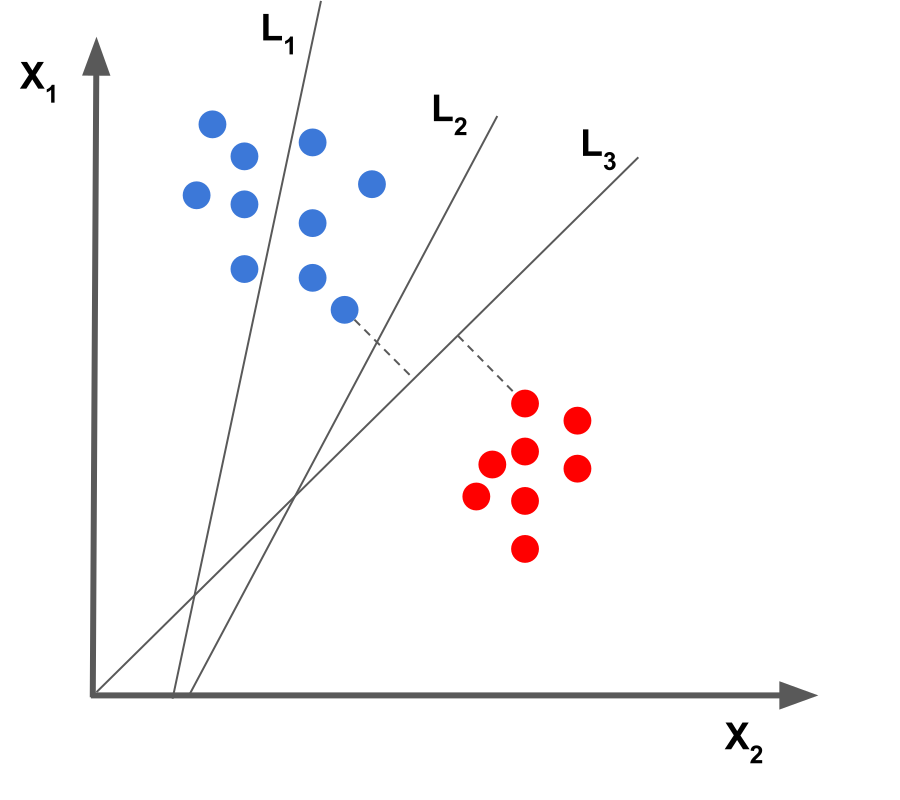
Vì vậy, đầu ra thuật toán là ID từ hình ảnh với biểu đồ gần nhất. Thuật toán cũng sẽ trả về khoảng cách tính toán, có thể được sử dụng làm phép đo ' **độ tin cậy** '. **Lưu ý** : đừng bị lừa về tên 'độ tin cậy', vì độ tin cậy thấp hơn sẽ tốt hơn bởi vì điều đó có nghĩa là khoảng cách giữa hai biểu đồ gần hơn.

Sau đó, chúng ta có thể sử dụng ngưỡng và 'độ tin cậy' để tự động ước tính nếu thuật toán đã nhận dạng chính xác hình ảnh. Chúng ta có thể giả định rằng thuật toán đã được công nhận thành công nếu độ tin cậy thấp hơn ngưỡng được xác định.

#### Thuật toán SVM.

##### 2.4.1.5.1. SVM là gì ?

Cách dễ nhất để hiểu SVM là sử dụng bài toán phân loại nhị phân.



**Hình 1: Hai lớp được hiển thị bằng hai màu khác nhau. SVM tìm thấy dòng tốt nhất để phân tách 2 lớp.**

Trong Hình 1, chúng ta thấy dữ liệu được biểu diễn dưới dạng các chấm trên mặt phẳng 2D. Dữ liệu thuộc về hai lớp khác nhau được biểu thị bằng màu của các chấm. Một cách để học cách phân biệt giữa hai lớp là vẽ một đường phân vùng không gian 2D thành hai phần. **Đào tạo** hệ thống đơn giản có nghĩa là tìm dòng. Khi bạn đã huấn luyện hệ thống (tức là tìm thấy dòng), bạn có thể nói nếu một điểm dữ liệu mới thuộc về lớp màu xanh hoặc màu đỏ bằng cách kiểm tra xem nó nằm ở phía nào của dòng.

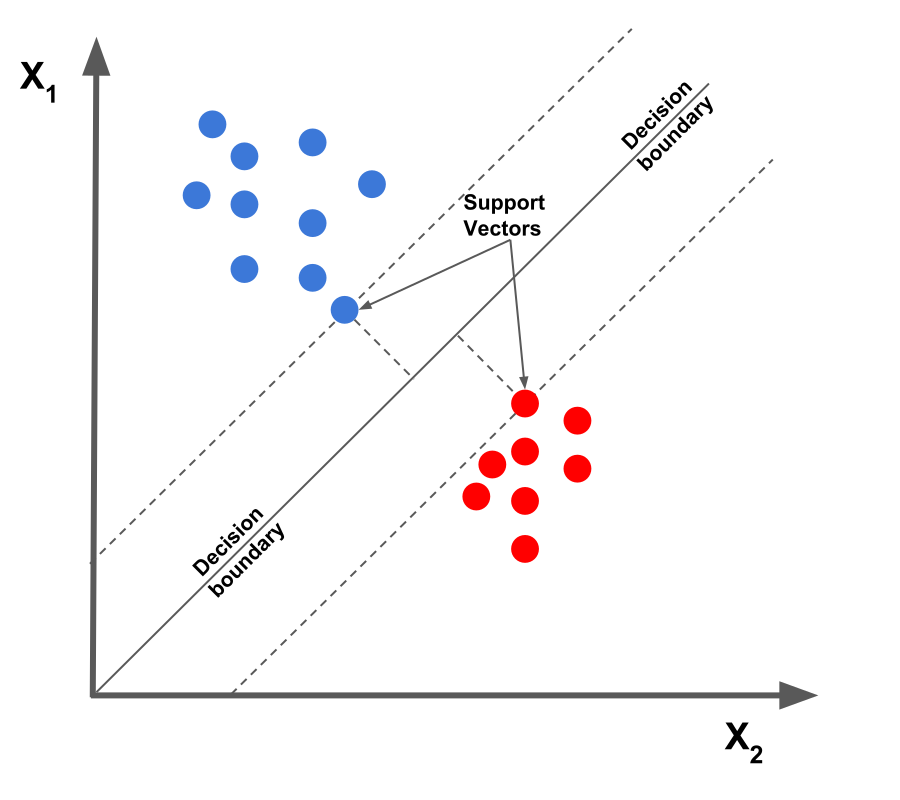
Trong Hình 1, rõ ràng dòng L1 không phải là một lựa chọn tốt vì nó không tách rời hai lớp. L2 và L3 đều tách biệt hai lớp, nhưng theo trực giác chúng ta biết L3 là lựa chọn tốt hơn L2 vì nó tách biệt rõ ràng hơn hai lớp.

##### 2.4.1.5.2. SVM làm việc như thế nào ?

Support Vector Machine (SVM) về cơ bản tìm thấy dòng tốt nhất phân tách dữ liệu trong 2D. Dòng này được gọi là **Ranh giới quyết định** .

Nếu chúng tôi có dữ liệu 1D, chúng tôi sẽ phân tách dữ liệu bằng một giá trị ngưỡng duy nhất. Nếu chúng ta có dữ liệu 3D, đầu ra của SVM là một mặt phẳng phân tách hai lớp. Cuối cùng, nếu dữ liệu nhiều hơn ba chiều, ranh giới quyết định là một **siêu phẳng** , không có gì ngoài một mặt phẳng ở các chiều cao hơn. Không, bạn không thể hình dung nó, nhưng bạn có ý tưởng!

Bây giờ, hãy xem dòng L3 được chọn bởi SVM như thế nào.



**Hình 2: Các điểm gần nhất với ranh giới quyết định được gọi là vectơ hỗ trợ. SVM tìm thấy ranh giới quyết định bằng cách tối đa hóa khoảng cách của nó từ các vectơ hỗ trợ.**

Các điểm gần nhất với các siêu phẳng tách biệt được gọi là các **vectơ hỗ trợ**. SVM giải quyết vấn đề tối ưu hóa sao cho

1. Các vectơ hỗ trợ có khoảng cách lớn nhất có thể từ ranh giới quyết định (tức là tách siêu phẳng).
2. Hai lớp nằm trên các mặt khác nhau của siêu phẳng.

Vấn đề tối ưu hóa này tương đương với tối đa hóa **Margin hình học ( \ gamma)** được hiển thị trong phương trình dưới đây.

\ started {align *} \ gamma = \ min ^ N_ {i = 1} \ hspace {0.25in} y_i \ frac {\ bf {w} ^ T \ bf {x_i} + b} {|| \ bf {w } ||} \ end {căn chỉnh *} (1)

trong đó \ bf {x} _ilà một ví dụ đào tạo, y_ilấy hai giá trị (1 và -1) cho phân loại nhị phân và siêu phẳng tách biệt được tham số hóa bởi \ bf {w}và {b}. Trong ví dụ 2D của chúng tôi, \ bf {x} _ichỉ đơn giản là tọa độ của các điểm 2D, y_ilà 1 cho màu xanh lam và -1 cho các chấm đỏ, và các tham số \ bf {w}và {b}có liên quan đến độ dốc và giao thoa của đường phân cách.

##### 2.4.1.5.3. Tham số c SVM.

Bây giờ, bạn có thể nghĩ rằng ví dụ đồ chơi tôi chọn là quá dễ dàng và dữ liệu thực sự ồn ào và hầu như không bao giờ có thể phân tách gọn gàng bằng cách sử dụng một siêu phẳng.

Trong những trường hợp như vậy, SVM vẫn tìm thấy siêu phẳng tốt nhất bằng cách giải quyết vấn đề tối ưu hóa cố gắng tăng khoảng cách của siêu phẳng từ hai lớp trong khi cố gắng đảm bảo nhiều ví dụ đào tạo được phân loại đúng.

Sự cân bằng này được kiểm soát bởi một tham số có tên C. Khi giá trị của C nhỏ, một siêu phẳng có biên độ lớn được chọn với chi phí cho một số lượng lớn hơn các phân loại sai. Ngược lại, khi C lớn, một siêu phẳng lề nhỏ hơn được chọn để cố gắng phân loại chính xác hơn nhiều ví dụ. Hình 3, mô tả bằng đồ họa sự đánh đổi này.

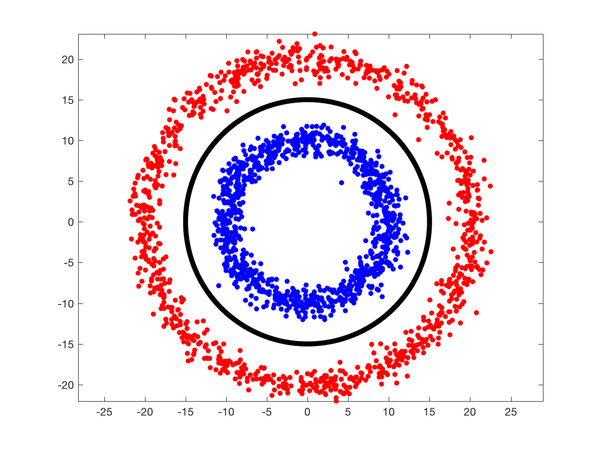


**Hình 3. Tham số SVM C**

Lưu ý : Dòng tương ứng với C = 100 không nhất thiết là một lựa chọn tốt. Điều này là do điểm màu xanh đơn độc có thể là một ngoại lệ.

**Tham số Gamma SVM ( \ gamma)**

Điều gì xảy ra nếu dữ liệu không thể phân tách bằng một siêu phẳng? Ví dụ, trong Hình 4, hai lớp được biểu thị bằng các chấm màu đỏ và màu xanh không thể phân tách tuyến tính. Ranh giới quyết định hiển thị trong màu đen thực sự là hình tròn.



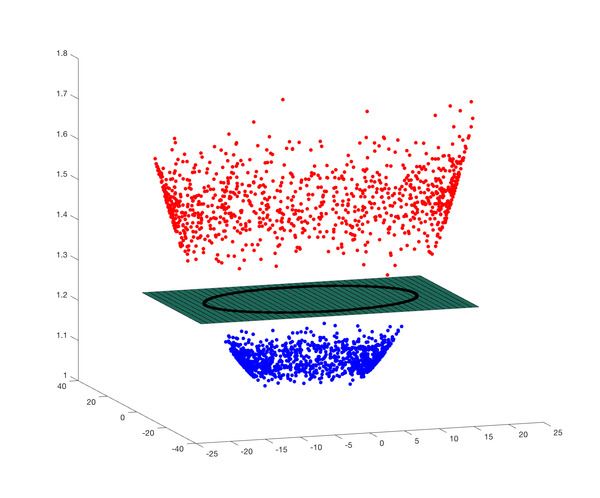
**Hình 4: Dữ liệu không phân tách tuyến tính**

Trong trường hợp như vậy, chúng tôi sử dụng **Thủ thuật hạt nhân** trong đó chúng tôi thêm một thứ nguyên mới cho dữ liệu hiện có và nếu chúng tôi may mắn, trong không gian mới, dữ liệu có thể phân tách tuyến tính. Hãy xem xét Kernel Trick bằng một ví dụ.

Trong Hình 5, chúng tôi đã thêm một chiều thứ ba (z) vào dữ liệu trong đó,

\[z = e^{-\gamma(x^2+y^2)}\]

Biểu thức trên được gọi là **Hàm cơ sở bán kính Gaussian** hoặc **Hàm cơ sở bán kính với hạt nhân Gaussian** . Chúng ta có thể thấy dữ liệu 3D mới có thể tách rời bằng mặt phẳng chứa vòng tròn màu đen! Tham số \ gammađiều khiển lượng kéo dài theo hướng z.



**Hình 5: Sử dụng Kernel Trick để tạo dữ liệu có thể phân tách tuyến tính.**

#### Thuật toán k-NN

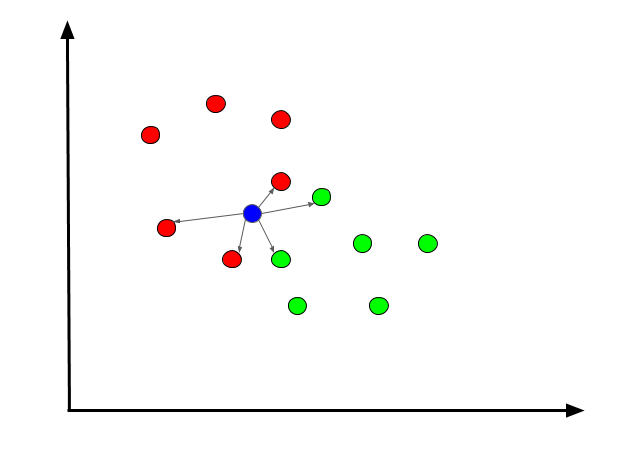
##### 2.4.1.6.1. Khái niệm k-NN.

Mặc dù cách tiếp cận SVM nhanh hơn, nhưng nó có một nhược điểm. SVM là một phương pháp Machine Learning tham số, có nghĩa là nếu một người mới được thêm vào cơ sở dữ liệu, các tham số cũ có thể không hoạt động và do đó, chúng ta sẽ cần phải đào tạo lại mô hình SVM. Điều gì xảy ra nếu có một phương pháp không tham số? - Vâng! Thuật toán k-NN.

K-nearest neighbor (KNN) là một trong những thuật toán supervised learning đơn giản. Khi huấn luyện, thuật toán này không học một điều gì từ dữ liệu huấn luyện mà nhớ lại một cách máy móc toàn bộ dữ liệu đó. Đây cũng là lý do thuật toán này được xếp vào loại lazy learning, mọi tính toán được thực hiện khi nó cần dự đoán đầu ra của dữ liệu mới. KNN có thể áp dụng được vào cả classification và regression. KNN còn được gọi là một thuật toán instance-based hay memory-based learning. Với KNN, trong bài toán classification, nhãn của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong tập huấn luyện. Nhãn đó có thể được quyết định bằng bầu chọn theo đa số (major voting) trong số K điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra kết quả. Chi tiết sẽ được nêu trong phần tiếp theo. Trong bài toán regresssion, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K = 1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên các điểm gần nhất đó và khoảng cách tới chúng.

Nói một cách dễ hiểu k-nearest neighbor (k-NN) là một trong những thuật toán Machine Learning đơn giản nhất. Nó không thực hiện một **tính toán kiểu tấn công.**của khoảng cách tại thời điểm suy luận. Đối với mỗi điểm mới, nó chỉ so sánh k- láng giềng gần nhất và sử dụng kê hoạch bỏ phiếu đa số để đưa ra quyết định.

Ví dụ, hãy xem xét các nhúng như hình dưới đây. Nếu chúng tôi sử dụng k = 5, điểm màu xanh (nhúng mới) được so sánh với các hàng xóm và chúng tôi bỏ phiếu đa số từ 5 người trong số họ. 3 trong số những người hàng xóm bỏ phiếu cho lớp đỏ và 2 phiếu cho lớp xanh. Như vậy, dự đoán cuối cùng sẽ là màu đỏ!

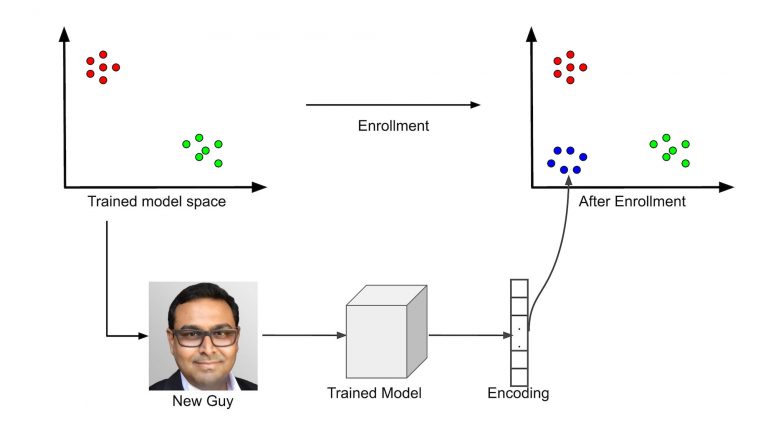


**Bỏ phiếu đa số bằng k-NN (k = 5) dự đoán điểm mới là lớp màu đỏ**

##### 2.4.1.6.2. Ghi danh một gương mặt mới.

Cho đến bây giờ chúng ta đã thấy không gian nhúng được tạo bằng cơ sở dữ liệu khuôn mặt. Nhưng điều gì xảy ra nếu chúng ta muốn thêm một người mới vào cơ sở dữ liệu? Chúng ta có cần phải đào tạo lại mô hình không? - KHÔNG. Nó khá đơn giản.

Chúng ta chỉ có thể sử dụng cùng một mô hình để tạo các nhúng cho người mới. Vì mô hình đã được đào tạo và đây là một người khác, nên các nhúng được tạo ra sẽ khác xa so với những người khác trong cơ sở dữ liệu. Chúng tôi chỉ cập nhật cơ sở dữ liệu của khuôn mặt được biết đến cho phù hợp.



Vì cơ sở dữ liệu hiện đã được cập nhật, chúng tôi có thể suy luận theo cách tương tự như được mô tả ở trên.

## các hướng phát triển bàn toán trong thực tế.

### Giải quyết vấn đề giai lận thi hộ trong kỳ thi.

#### Mô tả

Ngày nay, vấn đề học tập của học sinh, sinh viên rất quan trọng và cần được quan tâm bởi nó chính là nền tảng của một đất nước phát triển. Nhưng thực trạng học tập của nước ta hiện nay là chất lượng dạy và học của học sinh, sinh viên có chiều hướng giảm sút rất nhiều, một trong những nguyên nhân đó là thái độ thiếu trung thực trong thi cử, gian lận, quay cóp ,thi hộ.

Vấn đề thi hộ là ngày càng xảy ra phổ biến và tinh vi hơn và khó kiểm soát hơn. Một học sinh, sinh viên có học lực kém nhưng lại muốn có kết quả cao thì bắt buộc phải quay cóp hoặc nhờ người khác thi hộ, nhứng ngnguowiddi thi hộ sẽ sử dụng những chiều trò như sử dụng giấy tờ giả để qua mắt các giám thị coi thi.Việc học sinh, sinh viên học không trung thực là vấn đề rất nguy hiểm, nó gây ra những tác hại khôn lường. Học sinh, sinh viên sẽ không có kiến thức khi bước vào đời, ứng dụng vào công việc. Hơn nữa việc học đối phó sẽ ảnh hưởng tới sự trung thực của con người, học sinh sẽ dần đánh mất những nhân cách tốtnếu tình trạng này vẫn tiếp tục, về lâu dài sẽ làm suy thoái nền giáo dục nước nhà.

#### Hướng giải quyết

Để giải quyết vấn đề tránh gian lận thi hộ trong các kỳ thi chúng ta cần ứng dụng máy học nhận diện khuôn mặt để nhận diện khuôn mặt của từng thí sinh. Khi bắt đầu vào phòng thi chúng ta một camera để thu nhập dữ liệu hình ảnh của thí sinh qua đó xử lý bằng học máy và đưa ra kết quả người đi thi có phải là thật hay không.

### Giải quyết vấn đề xác định đối tượng truy nã.

#### Mô tả

Ngày nay xã hội càng phát triển tuy nhiên kèm theo đó những tệ nạn xã hội, tội phạm vẫn còn tồn tại rất nhiều. Có rất nhiều vụ án vô cùng thương tâm xảy ra hay những đường dây buôn bán hang cấm. Và những tên tội phạm thì luôn tìm cách trốn khỏi vòng vây của pháp luật. Việc này ngay trở ngại rất lớn trong quá trình phá án.

#### Hướng giải quyết

Để giải quyết vấn đề truy nã những tên tội phạm chúng ta có thể ứng dụng học máy nhận diện khuôn mặt để xác định đối tượng đang bị truy nã. Để thực hiện được chúng ta cần kết nối một máy tính chủ với những camera ở xung quanh khu vực truy nã . Khi nhận diện được đối tượng truy nã lập tức mày tính sẽ đưa ra thông báo.

### Giải quyết vấn đề điểm danh trong lớp học.

#### Mô tả

Ngày nay hình thức học hộ điểm danh hộ đã không còn xa lạ đối với mỗi sinh viên. Hình thức học hộ điểm danh hộ ngày càng phổ biến , những sinh viên không muốn đi học nhưng lại lo nghỉ quá số buổi quy định và bị cấm thi nên đã nhờ bạn hoặc thuê một người khác để điểm danh hộ. Và việc này rất khó để các giáo viên có thể kiểm soát triệt để được.

#### Hướng giải quyết

Để giải quyết vấn đề điểm danh hộ và học hộ chúng ta cần ứng dụng học máy nhận diện khuôn mặt để xác định các sinh viên trong buổi học. Để thực hiện chúng ta cần lắp đặt mỗi phòng học từ một đến hai camera đề lấy dữ liệu hình ảnh các sinh viên trong giờ học đừa về máy tính của thầy cô để xử lý và đưa ra thống kê số sinh viên đi học và những sinh viên học hộ.

### Khai phá dữ liệu phân tích thông kê và tính toán tỷ số giới tính trong khu vực.

#### Thông kê số lượng nam nữ mua hàng trong phạm vi nhất định.

##### Để giải quyết vấn đề thống kê và tính toán số lượng giới tính trong khu vực nhanh hơn, chính xác hơn . Chúng ta có thể ứng dụng học máy nhận diện, phân loại giới tính. Để thực hiện khu vực cửa hàng cần được lắp camera để lấy dữ liệu hình ảnh của những người mua hàng ở đây , dữ liệu sẽ được đưa vào máy tính xử lý và thông qua học máy nhận diện giới tính máy tính sẽ xác định số lượng giới tính nam, nữ. Qua đó chúng ta có thể biết nên nhập mặt hàng về nam nhiều hơn và ngược lại.

#### Tính toán tỷ số giới tính trong phạm vi nhất định.

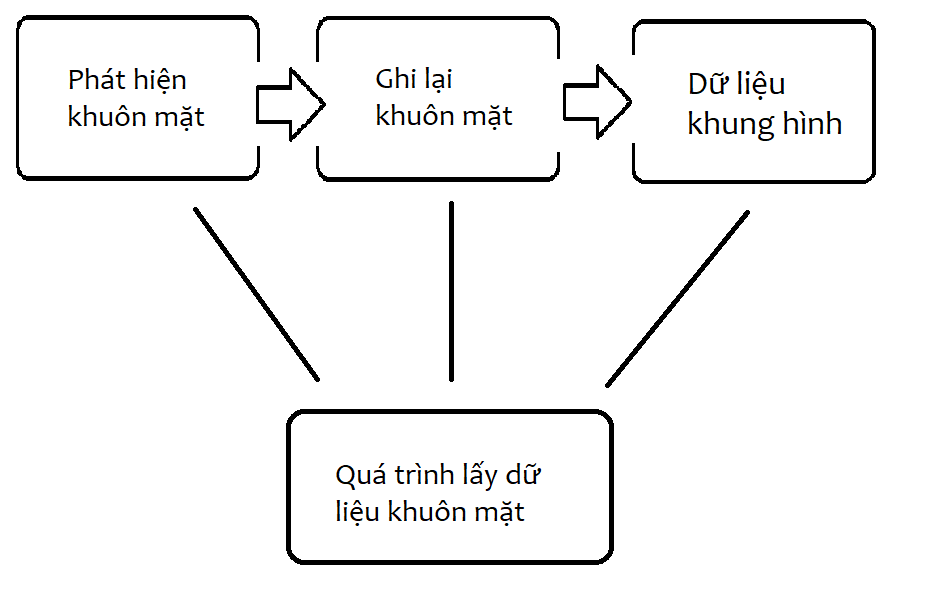
##### Để giải quyết vấn đề thống kê và tính toán tỷ số giới tính trong phạm vi nhất định, như trong địa phương , huyện , xã. Chúng ta có thể ứng dụng học máy nhận diện, phân loại giới tính. Để thực hiện chúng ta cần dữ liệu ảnh của tất cả những người trong khu vực đó, dữ liệu sẽ được đưa vào máy tính xử lý và thông qua học máy nhận diện giới tính máy tính sẽ xác định số lượng giới tính nam, nữ và đưa ra thống kê ,tỷ số giới tính. Qua đó tiết kiện thời gian ,tiền bạc, công sức trong việc thống kê số liệu của từng xã.

# CHƯƠNG 3 : QUY TRÌNH THỰC HIỆN

## 3.1. Thu thập dữ liệu cho học máy ( tranning ).

### 3.1.1. Dữ liệu ảnh cho học máy.

#### 3.1.1.1. Biểu đồ quy trình học máy.



#### 3.1.1.2. Chương trình thu thập dữ liệu ảnh.

**PYTHON**

import cv2

import os

import cvlib as cv

import urllib.request

import numpy as

def StreamVideo(\_url):

frameRequest = urllib.request.urlopen(\_url) # get data from http

framearr = np.array(bytearray(frameRequest.read()), dtype=np.uint8) # convert data to arrays 1D

frame = cv2.imdecode(framearr, -1) # convert data to arrays 2D

return frame

pass

#webcam = cv2.VideoCapture(0)

count = 0

name = input("My name : ")

url = input("URL : ")

\_url = str(url)+'/shot.jpg'

while True:

#ret, frame = webcam.read()

frame = StreamVideo(\_url)

face, confidence = cv.detect\_face(frame)

if not face :

print("The face cannot be identified")

else:

path = "Dataset/"+str(name)+"/"+str(name)+"\_"+str(count)+".jpg"

cv2.imwrite(path, frame)

print('DONE')

count +=1

cv2.imshow('frame',frame)

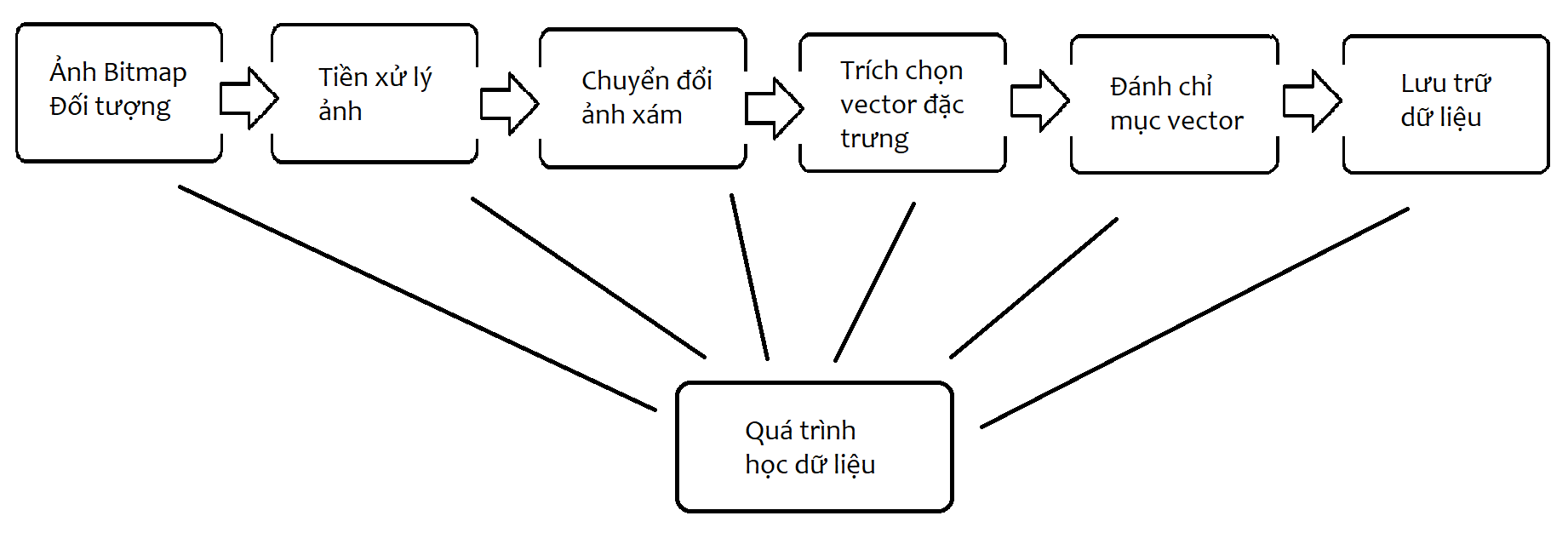
if cv2.waitKey(1) & 0xFF == ord('q') or count == 1000:

break

cv2.destroyAllWindows()

### 3.1.2. Quy trình thực hiện công việc học máy.

#### 3.1.2.1. Biểu đồ quy trình học máy.



#### 3.1.2.2. Chương trình học máy.

**PYTHON**

import cv2

import os

import numpy as np

from pil import image

import pickle

import argparse

# handle command-line arguments

ap = argparse.argumentparser()

ap.add\_argument('-f', '--folder', help='path to folder dataset')

args = ap.parse\_args()

name\_folder = args.folder

base\_dir = os.path.dirname(os.path.abspath(\_\_file\_\_))

image\_dir = os.path.join(base\_dir, name\_folder)

path\_Haar = 'haarcascades/haarcascade\_frontalface\_alt\_tree.xml'

face\_cascade = cv2.cascadeclassifier(path\_Haar)

recognizer = cv2.face.lbphfacerecognizer\_create()

current\_id = 0

label\_ids = {}

y\_labels = []

x\_train = []

for root, dirs, files in os.walk(image\_dir):

for file in files:

if file.endswith("png") or file.endswith("jpg"):

path = os.path.join(root, file)

label = os.path.basename(root).replace(" ", "-").lower()

if not label in label\_ids:

label\_ids[label] = current\_id

current\_id += 1

id\_ = label\_ids[label]

pil\_image = Image.open(path).convert("L")

image\_array = np.array(pil\_image, "uint8")

faces = face\_cascade.detectMultiScale(image\_array, scaleFactor=1.5, minNeighbors=5)

for (x,y,w,h) in faces:

roi = image\_array[y:y+h, x:x+w]

x\_train.append(roi)

y\_labels.append(id\_)

with open("pickles/face-labels.pickle", 'wb') as f:

pickle.dump(label\_ids, f)

recognizer.train(x\_train, np.array(y\_labels))

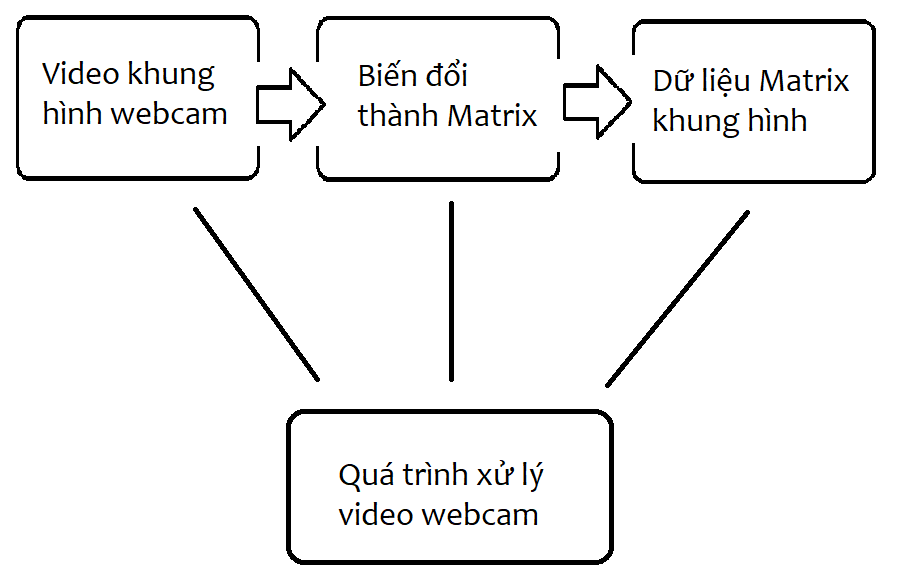
recognizer.write("trainner-face/trainner\_alt\_tree.yml")

print("\_\_Finish\_\_")

## 3.2. Xử lý dữ liệu ảnh hoặc phim đầu vào.

### 3.2.1. Xử lý dữ liệu Webcam.

#### 3.2.1.1. Biểu đồ quy trình xử lý.



#### 3.2.1.2. Chương tring xử lý.

**PYTHON**

import cv2

video = cv2.VideoCapture(0)

def WebcamVideo():

status , frame = video.read()

if not status:

print('Connect webcam is false')

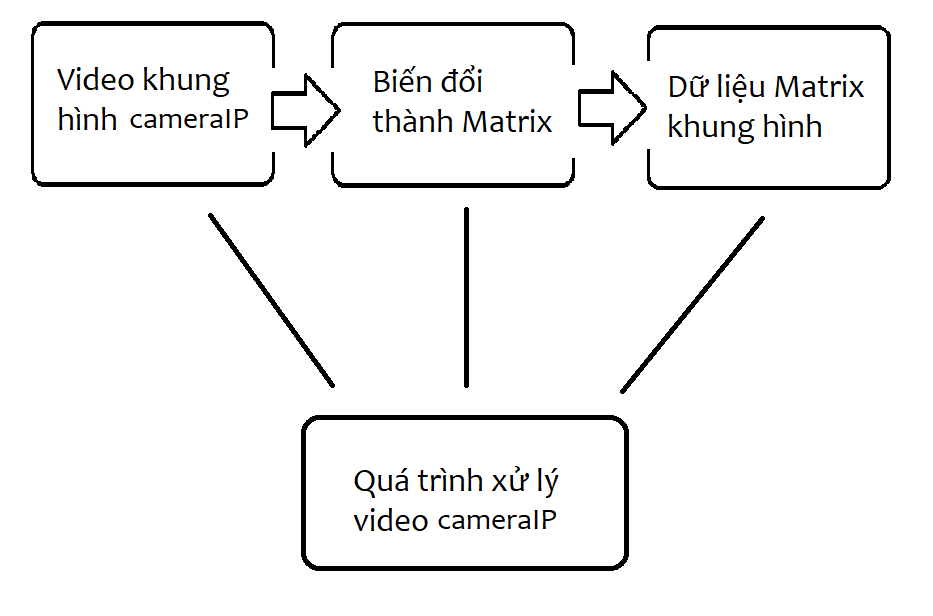
exit(0)

return frame

pass

### 3.2.2. Xử lý dữ liệu CameraIP.

#### 3.2.2.1. Biểu đồ quy trình xử lý.



#### 3.2.2.2. Chương trình xử lý.

truy cập Camera IP trong Python OpenCV.

Đưa ra một camera IP:

* Tìm IP địa chỉ máy ảnh của bạn
* Tìm port địa chỉ IP được truy cập
* Tìm protocol (HTTP / RTSP, v.v.) được chỉ định bởi nhà cung cấp máy ảnh

Sau đó, nếu máy ảnh của bạn được bảo vệ hãy tiếp tục và tìm hiểu:

* của bạn username
* của bạn password

Sau đó sử dụng dữ liệu của bạn để chạy tập lệnh sau:

**PYTHON**

import cv2

stream = cv2.VideoCapture('protocol://IP:port/1')

# Use the next line if your camera has a username and password

# stream = cv2.VideoCapture('protocol://username:password@IP:port/1')

while True:

r, f = stream.read()

cv2.imshow('IP Camera stream',f)

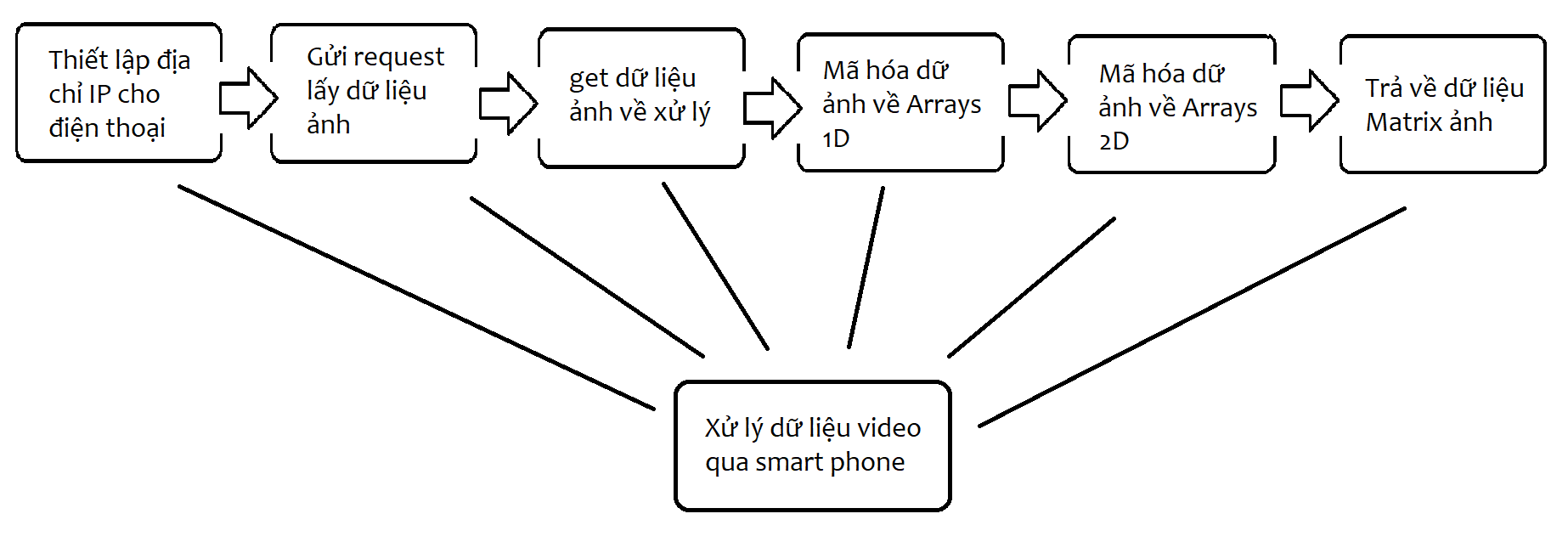
if cv2.waitKey(1) & 0xFF == ord('q'):

break

cv2.destroyAllWindows()

### 3.2.3. Xử lý dữ liệu Camera Smart Phone.

#### 3.2.3.1. Biểu đồ quy trình xử lý.



#### 3.2.3.2. Chương trình xử lý.

**PYTHON**

#Stream camera ip

def StreamVideo(\_url):

frameRequest = urllib.request.urlopen(\_url) # get data from https

framearr = np.array(bytearray(frameRequest.read()), dtype=np.uint8)

# convert data to arrays 1D

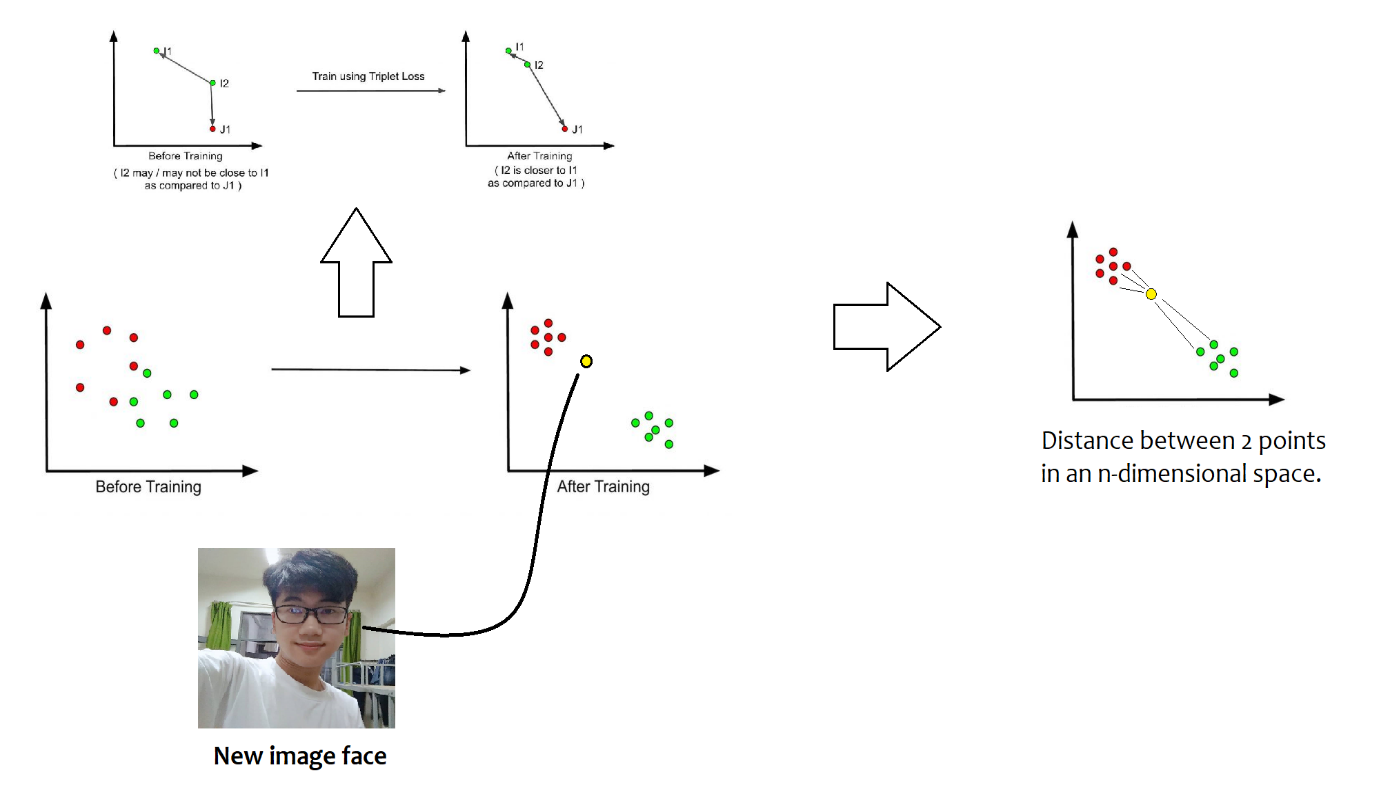
frame = cv2.imdecode(framearr, -1) # convert data to arrays 2D

return frame

pass

## 3.3. Xử lý dữ liệu nhận diện ( Nhận dạng )

### 3.3.1. Mô tả Sơ đồ quá trình.



### 3.3.2. Chương trình xử lý.

**PYTHON**

import numpy as np

import cv2

import cvlib as cv

import pickle

# read face

path\_haar = 'haarcascades/haarcascade\_frontalface\_alt\_tree.xml'

face\_cascade = cv2.CascadeClassifier(path\_haar)

recognizer = cv2.face.LBPHFaceRecognizer\_create()

recognizer.read("trainner-face/trainner\_alt\_tree.yml")

print("Read face : OK ")

# gets name person

labels = {"person\_name": 0}

with open("pickles/face-labels.pickle", 'rb') as f:

og\_labels = pickle.load(f)

labels = {v:k for k,v in og\_labels.items()}

print("Gets Object : OK ")

def Face\_Recognizer(frame):

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) #convert frame to frame gray

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)

for (x, y, w, h) in faces:

roi\_gray = gray[y:y+h, x:x+w]

id\_, conf = recognizer.predict(roi\_gray)

if conf >3 and conf < 86:

font = cv2.FONT\_HERSHEY\_SIMPLEX

name = labels[id\_] # read name from labels[]

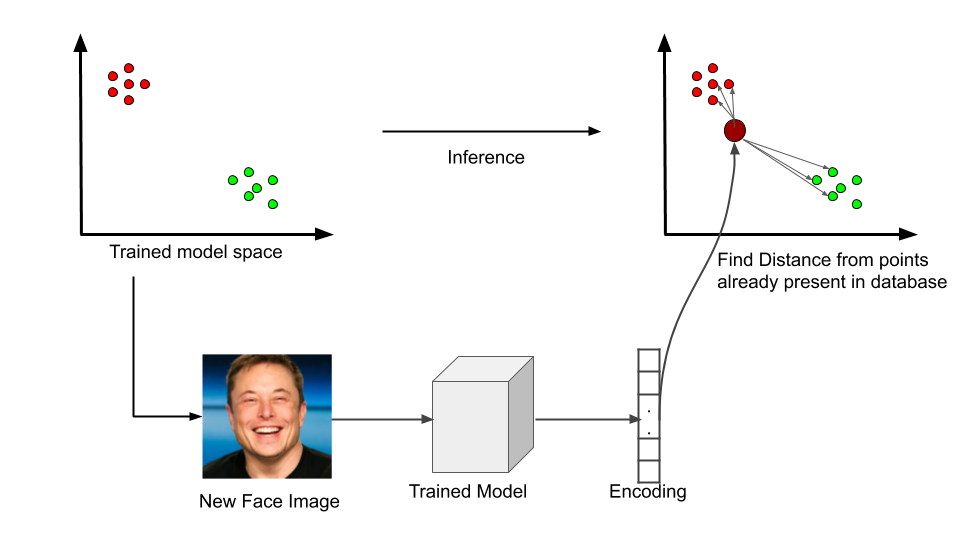
cv2.putText(frame, name, (x,y), font, 1, (10,20,255), 2,cv2.LINE\_AA)

cv2.rectangle(frame, (x, y), (x+w, y+h), (255,0,0), 2)

pass

## 3.4. Xử lý dữ liệu phân loại hoặc xác định giới tính.

### 3.4.1. Mô tả sơ đồ quá trình.



### 3.4.2. Chương trình xử lý.

**PYTHON**

import cv2 as cv

import math

import time

import argparse

faceProto = "opencv\_face\_detector.pbtxt"

faceModel = "opencv\_face\_detector\_uint8.pb"

genderProto = "gender\_deploy.prototxt"

genderModel = "gender\_net.caffemodel"

MODEL\_MEAN\_VALUES = (78.4263377603, 87.7689143744, 114.895847746)

genderList = ['Nam', 'Nu']

def getFaceBox(net, frame, conf\_threshold=0.7):

frameOpencvDnn = frame.copy()

frameHeight = frameOpencvDnn.shape[0]

frameWidth = frameOpencvDnn.shape[1]

blob = cv.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

net.setInput(blob)

detections = net.forward()

bboxes = []

for i in range(detections.shape[2]):

confidence = detections[0, 0, i, 2]

if confidence > conf\_threshold:

x1 = int(detections[0, 0, i, 3] \* frameWidth)

y1 = int(detections[0, 0, i, 4] \* frameHeight)

x2 = int(detections[0, 0, i, 5] \* frameWidth)

y2 = int(detections[0, 0, i, 6] \* frameHeight)

bboxes.append([x1, y1, x2, y2])

cv.rectangle(frameOpencvDnn, (x1, y1), (x2, y2), (0, 255, 0), int(round(frameHeight/150)), 8)

return frameOpencvDnn, bboxes

# Load network

genderNet = cv.dnn.readNet(genderModel, genderProto)

faceNet = cv.dnn.readNet(faceModel, faceProto)

def GenderClassification(bboxes):

padding = 20

for bbox in bboxes:

#print(bbox)

face = frame[max(0,bbox[1]-padding):min(bbox[3]+padding,frame.shape[0]-1),

max(0,bbox[0]-padding):min(bbox[2]+padding, frame.shape[1]-1)]

blob = cv.dnn.blobFromImage(face, 1.0, (227, 227), MODEL\_MEAN\_VALUES, swapRB=False)

genderNet.setInput(blob)

genderPreds = genderNet.forward()

gender = genderList[genderPreds[0].argmax()]

label = "{}".format(gender)

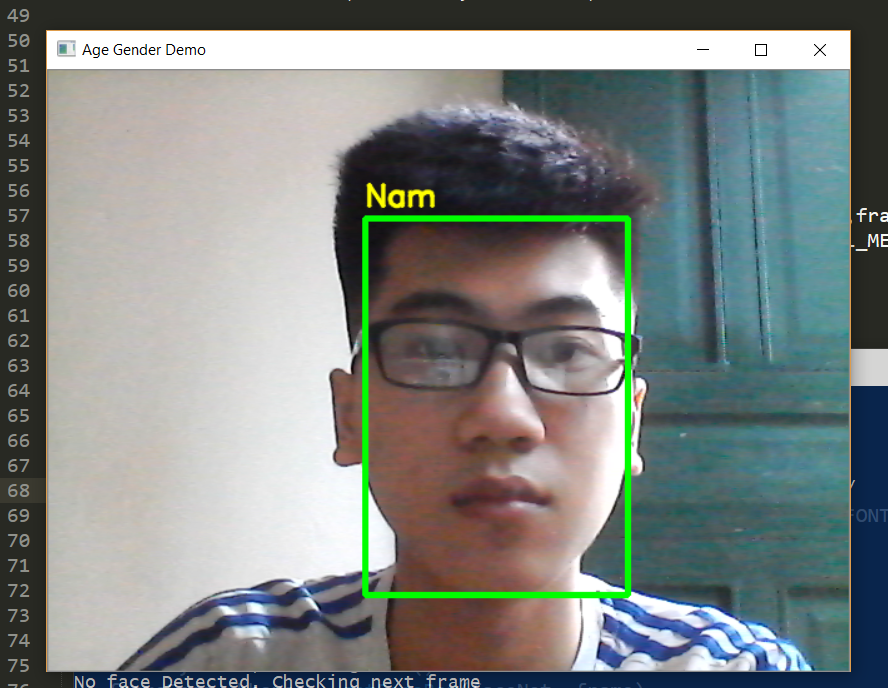
cv.putText(frameFace, label, (bbox[0], bbox[1]-10),

cv.FONT\_HERSHEY\_SIMPLEX, 0.8, (0, 255, 255), 2, cv.LINE\_AA)

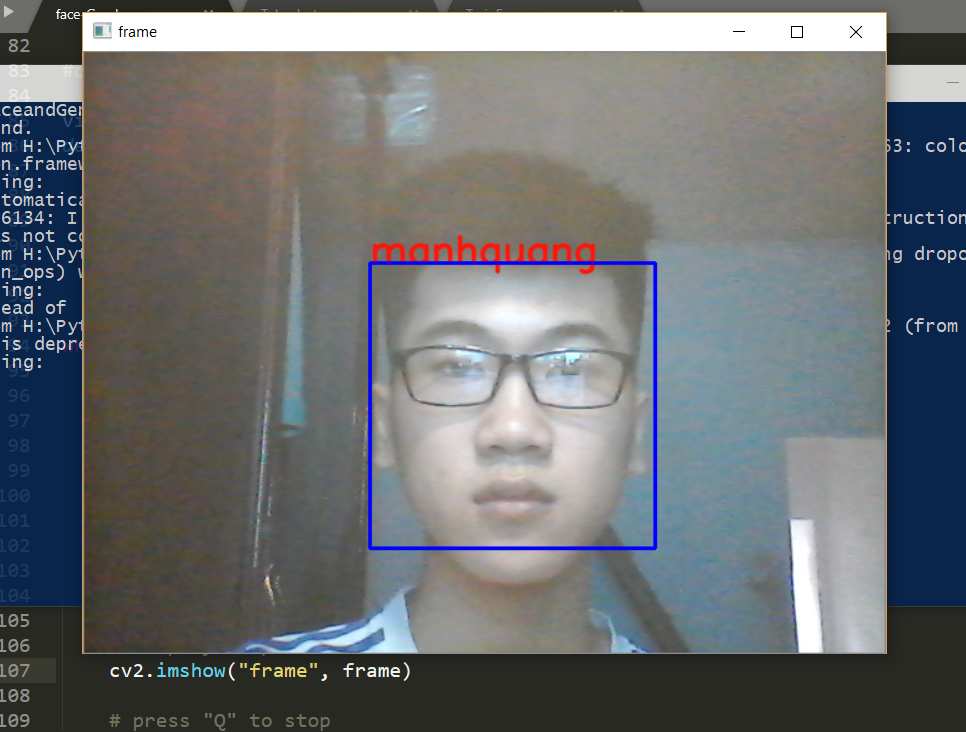
cv.imshow("Age Gender Demo", frameFace)

## 3.5. Kết quả thu được sau khi xử lý.

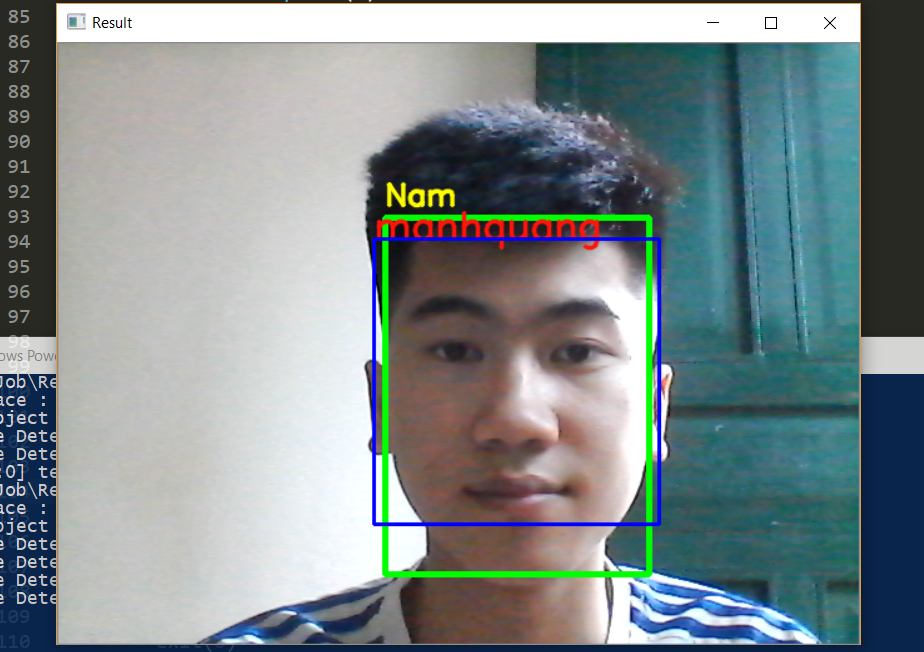
### 3.5.1. Phân loại giới tính.



### 3.5.2. Nhận diện ( Nhận dạng) Gương mặt



### 3.5.3. Kết hợp xử lý.



### 3.5.4. Nhận xét.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | ĐÁNH GIÁ | MỨC ĐỘ ĐÁNH GIÁ | | | | KẾT QUẢ | GHI CHÚ |
| TỐT | VỪA | TRUNG BÌNH | YẾU |
| 1 | Nhận diện  (nhận dạng). |  |  | OK |  | 30/100 | Khả năng nhận diện (nhận dạng) còn rất chậm và tỷ lệ chính xác ở mức trung bình,tỷ lệ tăng lên khi bức ảnh thu được rõ đủ ánh sáng. |
| 2 | Phân loại giới tính. |  | OK |  |  | 70/100 | Khả năng phân loại giới tính tốc độ ôn định, tỷ lệ chính xác cao kể cả khi trong môi trường thiếu ánh sang. Tỷ lệ chính xác giảm khi người có khuôn mặt gần giống nam hoặc gần giống nữ. |
| 3 | Kết hợp |  |  | OK |  | 50/100 |  |

# CHƯƠNG 4 : ĐÁNH GIÁ QUÁ TRÌNH NGHIÊN CỨU

## 4.1. Kết quả đạt được sau quá trình nghiên cứu.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | MỤC TIÊU | KẾT QUẢ | GHI CHÚ |
| 1 | Thành thạo ngôn ngữ Python. | Đã hoàn thành |  |
| 2 | Thành thạo Pycharm IDE. | Đã hoàn thành |  |
| 3 | Nghiên cứu các thuật toán xử lý ảnh . | Đã hoàn thành | Nắm vững cơ bản về phương pháp và tư tưởng thuật toán. |
| 4 | Nghiên cứu các thư viện xử lý ảnh trong OpenCV Python. | Đã hoàn thành | Nắm vững cơ bản về cách xử dụng thư viện xử lý ảnh. |
| 5 | Vận dụng và đưa ra giải pháp giải quyết vấn đề xử cần xử lý. | Đã hoàn thành | Hoàn thành các bản demo sản phẩm. |
| 6 | Hoàn thiện sản phẩm đưa vào thực tế đời sống . | Chưa hoàn thành | Tiếp tục nghiên cứu và cài thiện sản phẩm demo. |

## 4.2. Đánh giá kết quả sản phẩm demo.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | NGHIÊN CỨU | KẾT QUẢ | GHI CHÚ |
| 1 | Xác định giới tính. | Đã xác định được giới tính. | Độ chính xác mức khá cần cải thiệu thêm. |
| 2 | Nhận diện ( nhận dạng ) gương mặt. | Đã nhận diện được gương mặt. | Độ chính xác mức  Trung bình cần cải thiện thêm. |

# CHƯƠNG 5 : LIÊN HỆ VÀ TÀI LIỆU THAM KHẢO

## 5.1. Liên hệ nhóm nghiên cứu.

THÔNG TIN NHÓM THỰC HIỆN NGHIÊN CỨU.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | HỌ VÀ TÊN | CHỨC VỤ | LIÊN HỆ |
| 1 | Ts. Trần Chí Kiên | Giáo viên hướng dẫn. | Chikien.tran@haui.edu.vn |
| 2 | Đỗ Mạnh Quang | Chủ nhiệm đề tài. | domanhquang.rnd@gmail.com |
| 3 | Bùi Chiến Thắng | Thành viên | buichienthang.haui@gmail.com |

## 5.2. Các nguồn tài liệu tham khảo.

<https://www.learnopencv.com/face-recognition-an-introduction-for-beginners/>

<https://machinelearningcoban.com/>

<https://www.learnopencv.com/>

<https://www.researchgate.net/publication/236953573_Face_Recognition_Based_on_Facial_Features>

<https://pythonmachinelearning.pro/face-recognition-with-eigenfaces/>

<https://www.learnopencv.com/eigenface-using-opencv-c-python/>

<https://docs.opencv.org/2.4.13.7/modules/contrib/doc/facerec/facerec_tutorial.html>

<https://www.researchgate.net/publication/327980768_LBPH_Based_Improved_Face_Recognition_At_Low_Resolution>

<https://en.wikipedia.org/wiki/Local_binary_patterns>

<https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b>

<http://www.scholarpedia.org/article/Eigenfaces>

<https://en.wikipedia.org/wiki/Local_binary_patterns>

<https://en.wikipedia.org/wiki/Speeded_up_robust_features>