**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

KHOA CÔNG NGHỆ THÔNG TIN



**Báo cáo**

**Trí tuệ nhân tạo**

**Đề tài:**

**Ứng dụng nhận dạng khuôn mặt trong xây dựng phần mềm điểm danh.**

**Giáo viên hướng dẫn: Thầy TRần cao Trưởng**

**Thành viên: Nguyễn Danh Hiếu – KHMT 15 (C)**

**Nguyễn chiến thắng – KHMT 15**

**Ngô Hoàng Việt – KHMT 15**

Mục Lục

[A. Thuật toán Linear SVM sử dụng trên bài tập lớn 4](#_Toc10790124)

[B. Chi tiết trên bài tập lớn 4](#_Toc10790125)

[I. Extract embeddings : 4](#_Toc10790126)

[ Bước 1 : Truyền tham số đầu vào. 4](#_Toc10790127)

[ Bước 2: Nhập các gói của chúng tôi và phân tích các đối số dòng lệnh, cho phép tải trình phát hiện khuôn mặt và trình nhúng từ đĩa. 5](#_Toc10790128)

[ Bước 3 : Lấy đường dẫn hình ảnh của chúng tôi và thực hiện khởi tạo. 6](#_Toc10790129)

[ Bước 4 : Hãy bắt đầu lặp qua các đường dẫn hình ảnh - vòng lặp này sẽ chịu trách nhiệm trích xuất các nhúng từ các khuôn mặt được tìm thấy trong mỗi hình ảnh. 7](#_Toc10790130)

[ Bước 5 : Phát hiện và khoanh vùng khuôn mặt. 7](#_Toc10790131)

[ Bước 6 : Xử lý các Detections : 8](#_Toc10790132)

[ Bước 7 : Sử dụng embedder  , CNN trích xuất các embeddings mặt 9](#_Toc10790133)

[ Bước 8 : Khi vòng lặp kết thúc, kết xuất dữ liệu vào đĩa. 9](#_Toc10790134)

[II. Train face recognition model. 10](#_Toc10790135)

[ Bước 1 : Truyền tham số đầu vào import các thư viện 10](#_Toc10790136)

[ Bước 2 : Load nhúng và mã hóa label. 11](#_Toc10790137)

[ Bước 3 : Đào tạo mô hình SVM để nhận diện khuôn mặt. 12](#_Toc10790138)

[ Bước 4 : Sau khi đào tạo mô hình, xuất bộ mã hóa mô hình và nhãn vào đĩa dưới dạng tệp. 12](#_Toc10790139)

[III. Recognize faces with OpenCV 13](#_Toc10790140)

[ Bước 1 : Truyền tham số đầu vào import các thư viện 13](#_Toc10790141)

[ Bước 2 : Tải ba mô hình từ đĩa vào bộ nhớ : 14](#_Toc10790142)

[ Bước 3 : Tải hình ảnh và phát hiện khuôn mặt. 15](#_Toc10790143)

[ Bước 4 : Đưa ra các phát hiện mới, phát hiện khuôn mặt trong hình ảnh, lọc các phát hiện yếu và trích xuất ROI mặt. 15](#_Toc10790144)

[ Bước 5 : Nhận dạng tên của ROI khuôn mặt. 16](#_Toc10790145)

[ Bước 6 :  Hiển thị kết quả nhận dạng khuôn mặt OpenCV: 17](#_Toc10790146)

[IV. Recognize faces in video streams 18](#_Toc10790147)

[ Bước 1 : Truyền tham số đầu vào import các thư viện 18](#_Toc10790148)

[ Bước 2 : Load 3 mô hình và bộ mã hóa nhãn. 19](#_Toc10790149)

[ Bước 3 : khởi tạo luồng video và bắt đầu xử lý các khung: 20](#_Toc10790150)

[ Bước 3 : Xử lý các phát hiện khuôn mặt. 21](#_Toc10790151)

[ Bước 4 : Thực hiện nhận dạng khuôn mặt với OpenCV. 22](#_Toc10790152)

[ Bước 5 : Hiển thị kết quả và dọn dẹp: 23](#_Toc10790153)

# Thuật toán Linear SVM sử dụng trên bài tập lớn

#### Với bài toán binary classification mà 2 classes là *linearly separable*, có vô số các siêu mặt phẳng giúp phân biệt hai classes, tức mặt phân cách. Với mỗi mặt phân cách, ta có một *classifier*. Khoảng cách gần nhất từ 1 điểm dữ liệu tới mặt phân cách ấy được gọi là *margin* của classifier đó.

#### Support Vector Machine là bài toán đi tìm mặt phân cách sao cho *margin* tìm được là lớn nhất, đồng nghĩa với việc các điểm dữ liệu *an toàn nhất* so với mặt phân cách.

#### Bài toán tối ưu trong SVM là một bài toán lồi với hàm mục tiêu là *stricly convex*, nghiệm của bài toán này là duy nhất. Hơn nữa, bài toán tối ưu đó là một Quadratic Programming (QP).

#### Mặc dù có thể trực tiếp giải SVM qua bài toán tối ưu gốc này, thông thường người ta thường giải bài toán đối ngẫu. Bài toán đối ngẫu cũng là một QP nhưng nghiệm là *sparse* nên có những phương pháp giải hiệu quả hơn.

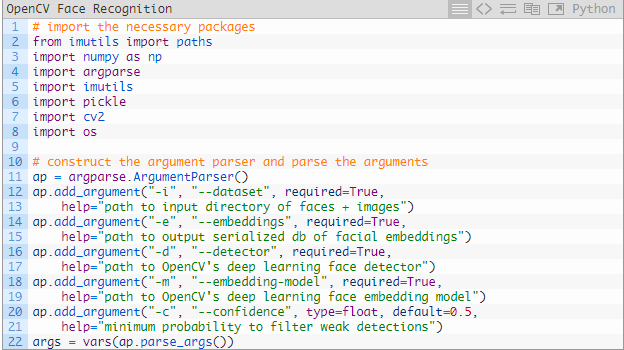
#### Với các bài toán mà dữ liệu *gần linearly separable* hoặc *nonlinear separable*, có những cải tiền khác của SVM để thích nghi với dữ liệu đó. Mời bạn đón đọc bài tiếp theo.

# Chi tiết trên bài tập lớn

## Extract embeddings :

(Trích xuất nhúng các bộ từ dữ liệu khuôn mặt )

### Bước 1 : Truyền tham số đầu vào.



#### -- dataset : Đường dẫn đến tập dữ liệu đầu vào về hình ảnh khuôn mặt.

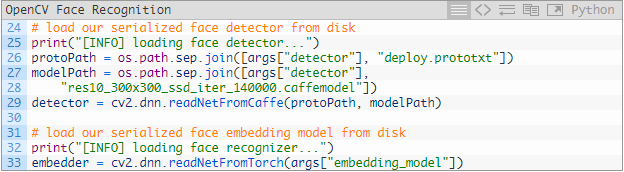
#### --embeddings : Đường dẫn đến tập tin nhúng đầu ra. Kịch bản của chúng tôi sẽ tính toán các phần nhúng mặt mà chúng tôi sẽ tuần tự hóa vào đĩa.

#### --detector : Đường dẫn đến trình phát hiện khuôn mặt học tập sâu dựa trên Caffe của OpenCV được sử dụng để thực sự định vị các khuôn mặt trong ảnh.

#### --embedding-model : Đường dẫn đến mô hình nhúng Torch học tập sâu OpenCV. Mô hình này sẽ cho phép chúng tôi trích xuất một vectơ nhúng mặt 128-D.

Từ ảnh data truyền vào -> phát hiện mặt (face detection) -> Nhúng thành véc tơ 128 –chiều.

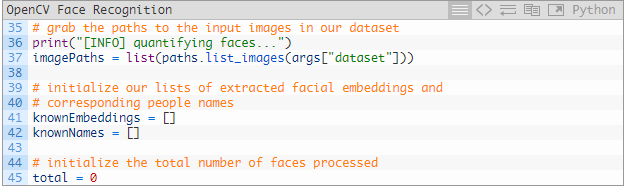
### Bước 2: Nhập các gói của chúng tôi và phân tích các đối số dòng lệnh, cho phép tải trình phát hiện khuôn mặt và trình nhúng từ đĩa.



#### **Dòng 26-29** : thực hiện Face Detection sử dụng DL dựa trên Caffe để bản địa hóa khuôn mặt trong ảnh (sử dụng file : res10\_300x300\_ssd\_iter\_140000.caffemodel )

#### **Dòng 33** : Mô hình này dựa trên Torch và chịu trách nhiệm trích xuất các nhúng mặt thông qua trích xuất tính năng học sâu.

### Bước 3 : Lấy đường dẫn hình ảnh của chúng tôi và thực hiện khởi tạo.



#### **Dòng 37** :  ImagePaths : Danh sách có chứa đường dẫn đến mỗi hình ảnh trong tập dữ liệu. Tôi đã thực hiện điều này dễ dàng thông qua  chức năng, thư viện Imutils, dường dẫn List\_image.

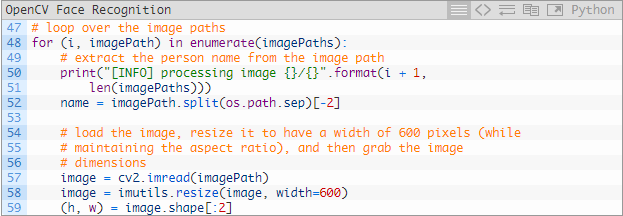
#### Các tên nhúng và tên tương ứng của chúng tôi sẽ được giữ trong hai danh sách :

#### **knownEmbeddings** : tên nhúng

#### **knownNames** : tên tương ứng của nhúng

#### Biến **total** : theo dõi xem có bao nhiêu khuôn mặt chúng tôi đã xử lý

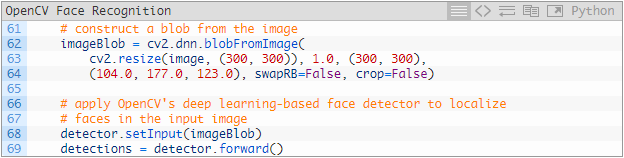
### Bước 4 : Hãy bắt đầu lặp qua các đường dẫn hình ảnh - vòng lặp này sẽ chịu trách nhiệm trích xuất các nhúng từ các khuôn mặt được tìm thấy trong mỗi hình ảnh.



#### Chúng tôi bắt đầu lặp qua **imagePaths (Dòng 48)** . Đầu tiên, chúng tôi trích xuất tên   của người đó từ đường dẫn **( Dòng 52 )**

#### **Dòng 57 – 59** : chuẩn hóa ảnh về cùng 1 size có chiều rộng 600

### Bước 5 : Phát hiện và khoanh vùng khuôn mặt.



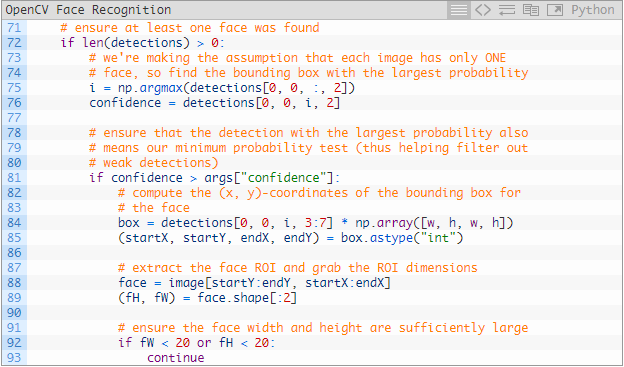
#### **Dòng 62 - 64**, chúng tôi xây dựng một đốm màu ( Tiền xử lí hình ảnh truyền vào ).

#### Từ đó, chúng tôi phát hiện các khuôn mặt trong ảnh bằng cách chuyển imageBlog qua detector network **(68,69)**

**Detector** sử dụng 2 file hỗ trợ face detection : (“**deploy.prototxt** ” và

“**res10\_300x300\_ssd\_iter\_140000.caffemodel** ”)

### Bước 6 : Xử lý các Detections :

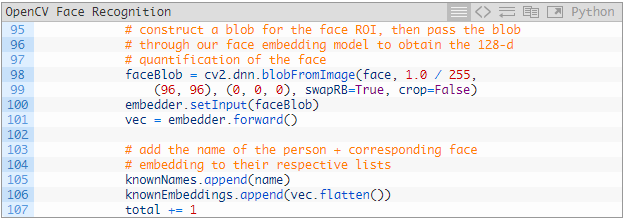


#### Detection : danh sách chứa xác suất và phối để bản địa hoá khuôn mặt trong ảnh.

#### Giả sử có ít nhất 1 khuôn mặt được tìm thấy (72) Chúng tôi đưa ra giả định rằng chỉ có một khuôn mặt trong ảnh, vì vậy chúng tôi trích xuất phát hiện với **confident** (độ tin cậy) lớn nhất và kiểm tra để đảm bảo rằng độ tin cậy đáp ứng ngưỡng xác suất tối thiểu được sử dụng để lọc các phát hiện yếu .

#### Giả sử đáp ứng ngưỡng đó (81) **confident = 0.5**,  chúng tôi trích xuất ROI khuôn mặt và lấy / kiểm tra kích thước để đảm bảo ROI mặt đủ lớn . (ROI : vùng khuôn mặt …)

### Bước 7 : Sử dụng embedder  , CNN trích xuất các embeddings mặt



#### Chúng tôi xây dựng một đốm màu khác, lần này là từ ROI mặt (không phải toàn bộ hình ảnh như chúng tôi đã làm trước đây).(98 - 99)

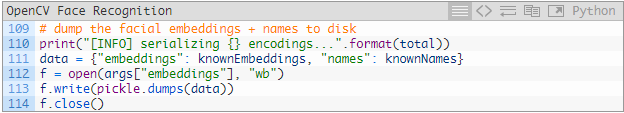
#### Sau đó, chúng tôi chuyển faceBlob   thông qua CNN nhúng ( 100-101 ). Điều này tạo ra một vectơ 128-D ( vec  ) mô tả khuôn mặt. Chúng tôi sẽ tận dụng dữ liệu này để nhận diện khuôn mặt mới thông qua học máy. ( embedder : sử dụng file openface\_nn4.small2.v1.t7 để thực hiện nhúng ).

#### Và sau đó chúng ta chỉ cần thêm tên label và tên của các nhúng tương ứng vào **Knownames** và **KnowEmbeddings** tương ứng để lưu .

#### Sau đó tăng Total (tổng số khuôn mặt đã nhúng) thêm 1.

#### Tiếp tục vòng lặp.

### Bước 8 : Khi vòng lặp kết thúc, kết xuất dữ liệu vào đĩa.

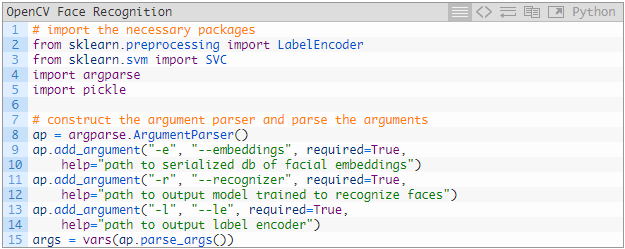


#### Chúng tôi thêm tên và nhúng dữ liệu vào một từ điển và sau đó tuần tự hóa dữ liệu trong file pickle. (embeddings.pickle)

## Train face recognition model.

( Mô hình nhận dạng , đào tạo khuôn mặt )

### Bước 1 : Truyền tham số đầu vào import các thư viện



#### Chúng tôi nhập các gói và mô-đun của chúng tôi trên  **Dòng 2-5** . Chúng tôi sẽ sử dụng triển khai Support Vector Machines (SVM) của scikit-learn, một mô hình học máy phổ biến.

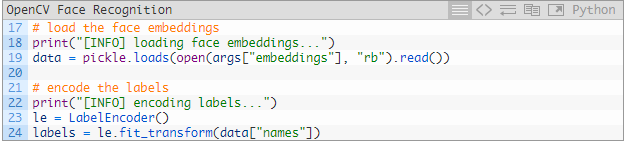
Từ đó, chúng tôi phân tích các đối số dòng lệnh của chúng tôi:

#### embeddings  : Đường dẫn đến các nhúng được tuần tự hóa (chúng tôi đã xuất nó bằng cách chạy  tập lệnh extract\_embeddings .py trước đó ).

#### recognizer  : Đây sẽ là mô hình đầu ra của chúng tôi *nhận* diện khuôn mặt. Nó dựa trên SVM. Chúng tôi sẽ lưu nó để chúng tôi có thể sử dụng nó trong hai tập lệnh nhận dạng tiếp theo.

#### le  : Đường dẫn tệp đầu ra bộ mã hóa nhãn của chúng tôi. Chúng tôi sẽ tuần tự hóa bộ mã hóa nhãn của chúng tôi vào đĩa để chúng tôi có thể sử dụng nó và mô hình nhận dạng trong các tập lệnh nhận dạng khuôn mặt hình ảnh / video của chúng tôi.

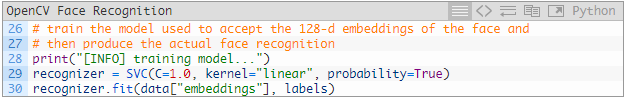
### Bước 2 : Load nhúng và mã hóa label.



#### Ở đây chúng tôi tải các phần nhúng của chúng tôi từ **Phần 1** (Extract embeddings) **(Dòng 19).** Chúng tôi sẽ không tạo ra bất kỳ nhúng nào trong tập lệnh đào tạo mô hình này - chúng tôi sẽ sử dụng các nhúng được tạo trước đó và tuần tự hóa.

#### Sau đó, chúng tôi khởi tạo scikit-learn  LabelEncoder   và mã hóa Label name.

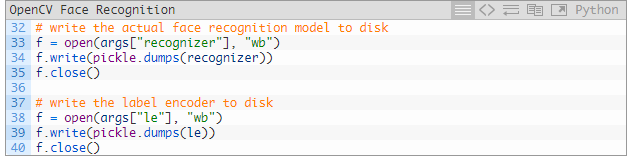
### Bước 3 : Đào tạo mô hình SVM để nhận diện khuôn mặt.



#### **Dòng 29**,  Khởi tạo mô hình SVM .

#### **Dòng 30,** Fit mô hình (training model).

### Bước 4 : Sau khi đào tạo mô hình, xuất bộ mã hóa mô hình và nhãn vào đĩa dưới dạng tệp.

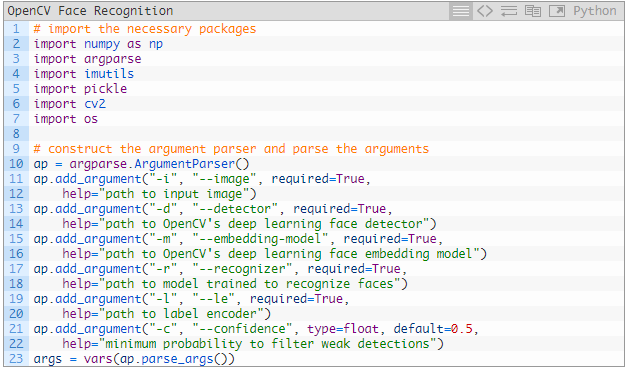


#### le.pickle : lưu nhãn.

#### recognizer.pickle : lưu mô hình nhận dạng khuôn mặt thực tế.

## Recognize faces with OpenCV

### Bước 1 : Truyền tham số đầu vào import các thư viện



#### - image  : Đường dẫn đến hình ảnh đầu vào. Chúng tôi sẽ cố gắng nhận ra khuôn mặt trong hình ảnh này.

#### - detector :  Đường dẫn đến OpenCV’s deep learning phát hiện khuôn mặt. Chúng tôi sẽ sử dụng mô hình này để phát hiện ROI ở đâu trong hình ảnh.

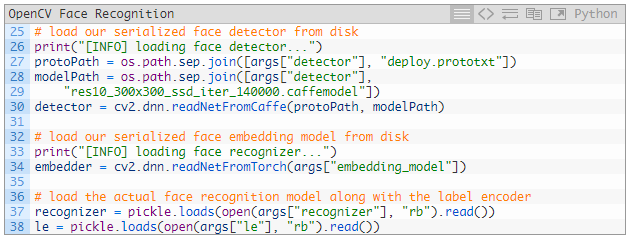
#### - embedding-model : Đường dẫn đến mô hình nhúng khuôn mặt học tập sâu của OpenCV. Chúng tôi sẽ sử dụng mô hình này để*trích xuất*khuôn mặt 128-D từ ROI mặt - chúng tôi sẽ đưa dữ liệu vào bộ nhận dạng.

#### - Recognizer  : Đường dẫn đến mô hình nhận dạng của chúng tôi. Chúng tôi đã đào tạo trình nhận dạng SVM của mình trong **phần II** .Đây là những gì thực sự sẽ xác định một khuôn mặt là ai.

#### - le  : Đường dẫn đến bộ mã hóa nhãn của chúng tôi. Điều này có chứa nhãn khuôn mặt của chúng tôi như 'adrian'   hoặc 'trisha'  .

#### - confidence : Ngưỡng tùy chọn để lọc và phát hiện khuôn mặt yếu.

### Bước 2 : Tải ba mô hình từ đĩa vào bộ nhớ :

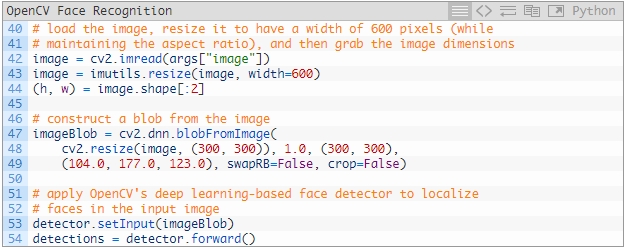


#### detector : Mộtmô hình Caffe DL được đào tạo trước để phát hiện vị trí của face ở đâu trong ảnh **(dòng 27-30).**

#### embedder : Một mô hình Torch DL được đào tạo trước để tính toán các nhúng mặt 128-D. **(dòng 34).**

#### recognizer :Mô hình nhận dạng khuôn mặt tuyến tính SVM của chúng tôi **(** **dòng 37**).

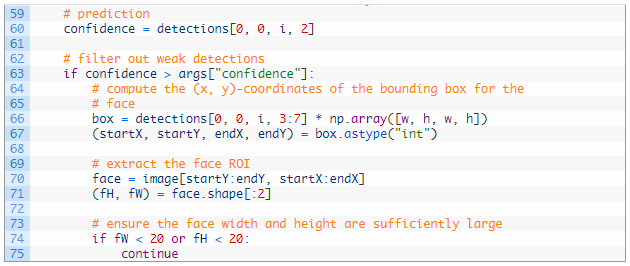
### Bước 3 : Tải hình ảnh và phát hiện khuôn mặt.



#### Tải hình ảnh vào bộ nhớ và tiền xử lí ảnh **( Dòng 42-49 ).**

#### Khoanh vùng các khuôn mặt trong hình ảnh thông qua **detector** .

### Bước 4 : Đưa ra các phát hiện mới, phát hiện khuôn mặt trong hình ảnh, lọc các phát hiện yếu và trích xuất ROI mặt.

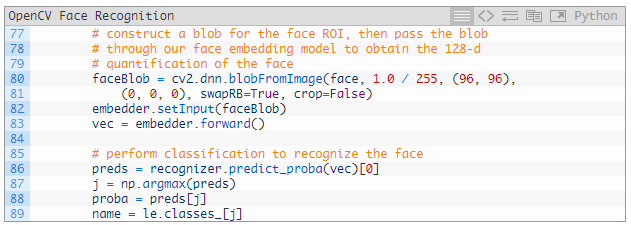


#### Lặp lại các phát hiện trên ( **Dòng 57)** và rút ra sự độ tin cậy của từng phát hiện trên.

#### Sau đó, chúng tôi so sánh độ tin cậy với ngưỡng phát hiện xác suất tối thiểu đã khai báo từ trước, đảm bảo rằng xác suất tính toán lớn hơn xác suất tối thiểu **( Dòng 63 ).**

#### Từ đó, chúng tôi trích xuất vùng khuôn mặt **( Dòng 66-70 )** cũng như đảm bảo kích thước không gian của nó đủ lớn.

### Bước 5 : Nhận dạng tên của ROI khuôn mặt.

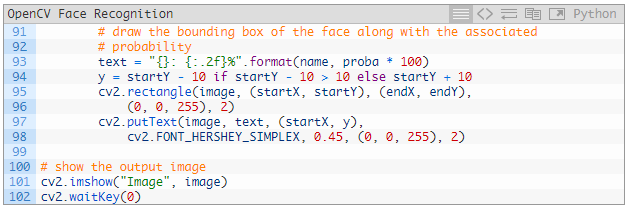


#### Đầu tiên, Xây dựng một FaceBlob (từ   ROI mặt ) và chuyển nó qua bộ embedder để tạo ra một vectơ 128-D mô tả khuôn mặt **( Dòng 80-83 )**

#### Sau đó, chúng tôi chuyển **vec** qua mô hình nhận dạng SVM của chúng tôi ( **Dòng 86** ), kết quả của nó là dự đoán của chúng tôi cho ai đang trong ROI face.

#### Chúng tôi lấy chỉ số xác suất cao nhất **( Dòng 87 )** và truy vấn bộ mã hóa nhãn của chúng tôi để tìm **Names** **( Dòng 89 ).**Ở giữa, tôi trích xuất xác suất trên  **(Dòng 88) .**

### Bước 6 :  Hiển thị kết quả nhận dạng khuôn mặt OpenCV:



Đối với mọi khuôn mặt, chúng tôi nhận ra trong vòng lặp (bao gồm cả những người chưa biết đến) :

#### Chúng tôi xây dựng một **text** chứa tên và xác suất trên (**Dòng 93)** .

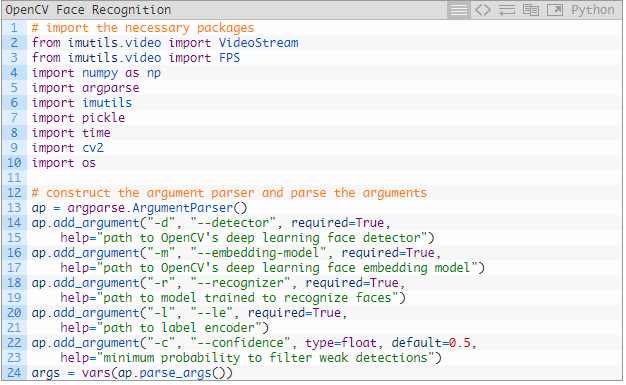
#### Và sau đó chúng ta vẽ một hình chữ nhật xung quanh mặt và đặt văn bản phía trên hộp ( **Dòng 94-98** ).

#### Và cuối cùng, chúng tôi hình dung kết quả trên màn hình cho đến khi nhấn phím ( **Dòng 101 và 102** ).

## Recognize faces in video streams

(Nhận diện khuôn mặt trên video streams)

### Bước 1 : Truyền tham số đầu vào import các thư viện



#### Dòng 2 – 3 , import thư viện imutils sử dụng mô đun VideoStream để chụp khung hình từ máy ảnh và FPS của chúng tôi để tính toán số liệu thống kê khung hình mỗi giây.

#### - detector :  Đường dẫn đến OpenCV’s deep learning phát hiện khuôn mặt. Chúng tôi sẽ sử dụng mô hình này để phát hiện ROI ở đâu trong hình ảnh.

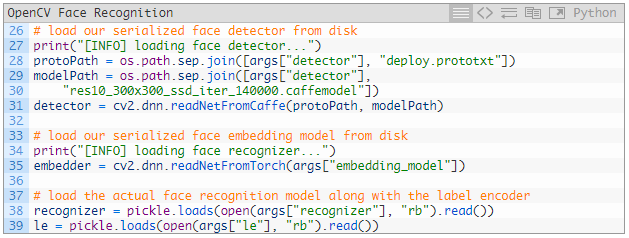
#### - embedding-model : Đường dẫn đến mô hình nhúng khuôn mặt học tập sâu của OpenCV. Chúng tôi sẽ sử dụng mô hình này để*trích xuất*khuôn mặt 128-D từ ROI mặt - chúng tôi sẽ đưa dữ liệu vào bộ nhận dạng.

#### - Recognizer  : Đường dẫn đến mô hình nhận dạng của chúng tôi. Chúng tôi đã đào tạo trình nhận dạng SVM của mình trong **phần II** .Đây là những gì thực sự sẽ xác định một khuôn mặt là ai.

#### - le  : Đường dẫn đến bộ mã hóa nhãn của chúng tôi. Điều này có chứa nhãn khuôn mặt của chúng tôi như 'adrian'   hoặc 'trisha'  .

#### - confidence : Ngưỡng tùy chọn để lọc và phát hiện khuôn mặt yếu.

### Bước 2 : Load 3 mô hình và bộ mã hóa nhãn.



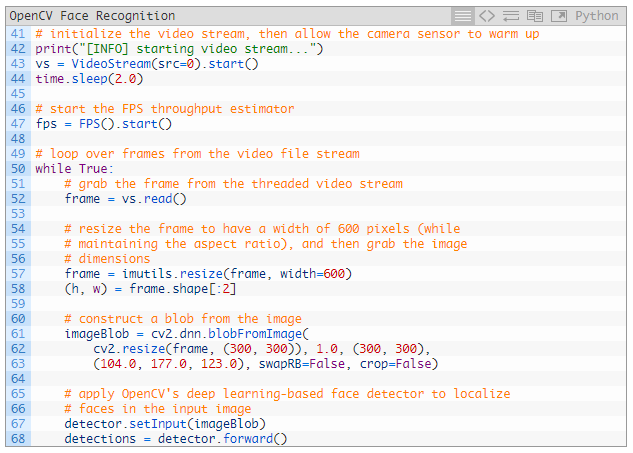
#### Ở đây chúng tôi tải bộ phát hiện khuôn mặt (**face detector)**.

#### Mô hình nhúng khuôn mặt (**embedder**).

#### Mô hình nhận dạng khuôn mặt (Tuyến tính SVM) (**recognizer**).

#### Bộ mã hóa nhãn (**le**).

### Bước 3 : khởi tạo luồng video và bắt đầu xử lý các khung:

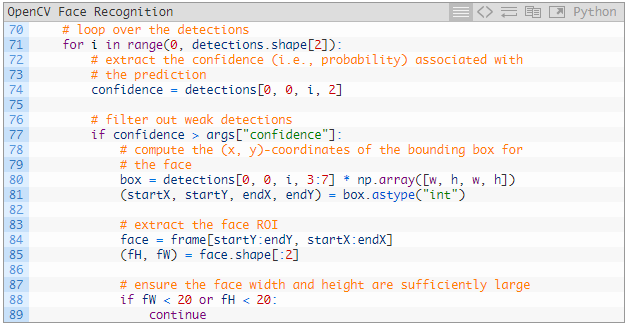


#### Đối tượng VideoStream được khởi tạo và bắt đầu trên  **(Dòng 43)**, chờ cảm biến camera khởi động **(Dòng 44)** .

#### Chúng tôi cũng khởi tạo các khung của chúng tôi trên bộ đếm thứ hai **( Dòng 47 )** và bắt đầu lặp qua các khung trên  **(Dòng 50)**. Lấy 1 khung hình từ webcam trên  **(Dòng 52)**.

#### Từ đây mọi thứ giống như **Phần 3**. Chúng tôi thay đổi kích thước khung hình **(Dòng 57)**và sau đó chúng tôi xây dựng một đốm màu từ khung hình + phát hiện các khuôn mặt ở đâu **(Dòng 61-68) .**

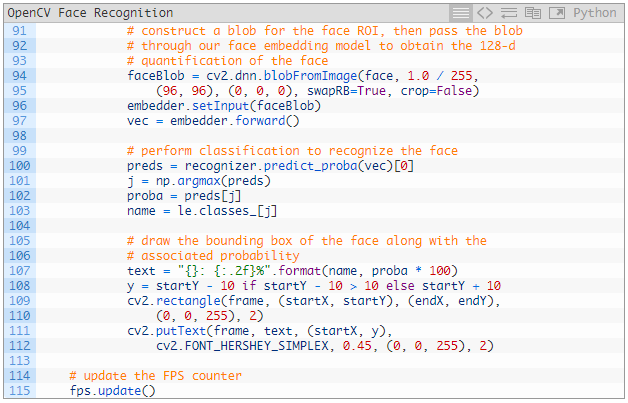
### Bước 3 : Xử lý các phát hiện khuôn mặt.



#### Cũng giống như trong phần trước, chúng tôi bắt đầu lặp qua các detections và lọc ra những phát hiện yếu **( Dòng 71-77 )**.

#### Sau đó, chúng tôi trích xuất ROI mặt cũng như đảm bảo kích thước không gian đủ lớn cho các bước tiếp theo **( Dòng 84-89 )**.

### Bước 4 : Thực hiện nhận dạng khuôn mặt với OpenCV.



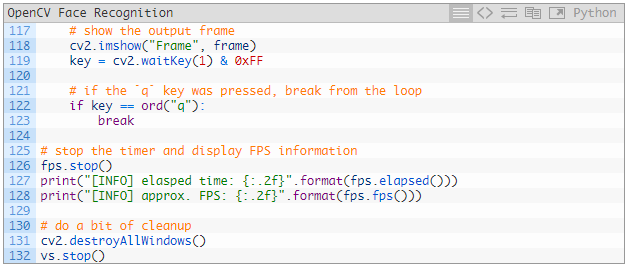
#### Xây dựng **faceBlob** ( **Dòng 94 và 95** ) và tính toán các nhúng mặt thông qua deep learning ( **Dòng 96 và 97** ).

#### Nhận biết Names có khả năng nhất của khuôn mặt trong khi tính xác suất ( **Dòng 100-103**).

#### Vẽ một hộp giới hạn xung quanh khuôn mặt và tên người được nhận diện + xác suất ( **Dòng 107 -112** ).

#### Bộ đếm fps của chúng tôi được cập nhật trên (**Dòng 115).**

### Bước 5 : Hiển thị kết quả và dọn dẹp:



#### Hiển thị khung chú thích   ( **Dòng 118** ) và đợi phím phím q q được nhấn tại điểm chúng tôi thoát ra khỏi vòng lặp ( **Dòng 119-123** ).

#### Dừng bộ   đếm fps của chúng tôi và in số liệu thống kê trong terminal ( **Dòng 126-128** ).

#### Dọn dẹp bằng cách đóng các cửa sổ và giải phóng con trỏ ( **Dòng 131 và 132** ).