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

KHOA ĐIỆN ĐIỆN TỬ

BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG



BÁO CÁO CUỐI KÌ

**NHẬN DIỆN KHUÔN MẶT**

GVHD: **PGS.TS Trương Ngọc Sơn**

Sinh viên: **Trương Đức Nhật Minh**

MSSV: 18142340

**Phạm Đình Khanh**

MSSV: 18142307

TP. HỒ CHÍ MINH – 6/2021

BẢN NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

|  |  |  |
| --- | --- | --- |
| Stt | Nội dung thực hiện | Nhận xét |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Nhận xét tổng quát:

**TÓM TẮT**

Deep Learning là thuật toán dựa trên một số ý tưởng từ não bộ tới việc tiếp thu nhiều tầng biểu đạt, cả cụ thể lẫn trừu tượng, qua đó làm rõ nghĩa của các loại dữ liệu. Deep Learning được ứng dụng trong nhận diện hình ảnh, nhận diện giọng nói, xử lý ngôn ngữ tự nhiên. Hiện nay rất nhiều các bài toán nhận dạng sử dụng Deep Learning, vì nó có thể giải quyết các bài toán với số lượng lớn các biến, tham số kích thước đầu vào lớn với hiệu năng cũng như độ chính xác vượt trội so với các phương pháp phân lớp truyền thống, xây dựng những hệ thống thông minh với độ chính xác cao. Trong báo cáo này, chúng tôi nghiên cứu về đề tài “Nhận diện khuôn mặt” theo sự hướng dẫn của PGS.TS Trương Ngọc Sơn.

**mục lục**

[**DANH MỤC HÌNH** iv](#_Toc74751132)

[**CÁC TỪ VIẾT TẮT** v](#_Toc74751133)

[Chương 1: Giới thiệu 1](#_Toc74751134)

[1.1. Giới thiệu 1](#_Toc74751135)

[1.2. mục tiêu đề tài 1](#_Toc74751136)

[1.3. GIỚI HẠN ĐỀ TÀI 1](#_Toc74751137)

[1.4. PHƯƠNG PHÁP NGHIÊN CỨU 2](#_Toc74751138)

[1.5. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU 2](#_Toc74751139)

[1.6. BỐ CỤC QUYỂN BÁO CÁO 2](#_Toc74751140)

[Chương 2: CƠ SỞ LÝ THUYẾT 3](#_Toc74751141)

[2.1. Giới thiệu mạng cnn 3](#_Toc74751142)

[2.2. hàm opencv 6](#_Toc74751143)

[2.2.1. OpenCV là gì? 6](#_Toc74751144)

[2.2.2. Cấu trúc OpenCV 7](#_Toc74751145)

[2.2.3. Phát hiện khuôn mặt với các hàm trong OpenCV 7](#_Toc74751146)

[2.3. Hàm OS 8](#_Toc74751147)

[2.3.1. Hàm os là gì? 8](#_Toc74751148)

[2.3.2. Tạo folder bằng os 8](#_Toc74751149)

[Chương 3: xây dựng chương trình 9](#_Toc74751150)

[3.1. xây dựng mô hình 9](#_Toc74751151)

[3.2. Các bước thực hiện 9](#_Toc74751152)

[3.2.1. Tạo tập dữ liệu 9](#_Toc74751153)

[3.2.2. Huấn luyện dữ liệu đưa vào 14](#_Toc74751154)

[3.2.3. Hiển thị dữ liệu đầu ra 18](#_Toc74751155)

[Chương 4: Kết quả thí nghiệm 20](#_Toc74751156)

[4.1. Kết quả sau khi huấn luyện 20](#_Toc74751157)

[4.2. Đưa hình ảnh một người cần nhận diện 21](#_Toc74751158)

[KẾT LUẬN 24](#_Toc74751159)

[PHỤ LỤC 25](#_Toc74751160)

[TÀI LIỆU THAM KHẢO 28](#_Toc74751161)

DANH MỤC HÌNH

[Hình 2.1 Kiến trúc cơ bản của mạng CNN 3](#_Toc74668452)

[Hình 2.2 Bộ lọc tích chập được sử dụng trên ma trận điểm ảnh 4](#_Toc74668453)

[Hình 2.3 Phương thức Avarage Pooling và Max Pooling 5](#_Toc74668454)

[Hình 2.4 Cấu trúc cơ bản của OpenCV 7](#_Toc74668455)

[Hình 2.5 Trong thư mục của OpenCV đã có các bộ huấn luyện có sẵn 8](#_Toc74668456)

[Hình 2.6 Trong thư mục của “os” đã có các bộ huấn luyện có sẵn 8](#_Toc74668457)

[Hình 3.1 Sơ đồ quy trình của mô hình nhận dạng khuôn mặt 9](#_Toc74668458)

[Hình 3.2 Chuyển về ảnh xám 10](#_Toc74668459)

[Hình 3.3 Tầm nhìn của máy tính khi các đặc trưng Haar chạy 11](#_Toc74668460)

[Hình 3.4 Tạo khung khuôn mặt 11](#_Toc74668461)

[Hình 3.5 Các đặc trưng Haar đánh dấu những nơi có thể là khuôn mặt 12](#_Toc74668462)

[Hình 3.6 Bức ảnh đã được xác định khuôn mặt 12](#_Toc74668463)

[Hình 3.7 Đưa bức ảnh cần nhận diện vào chương trình 13](#_Toc74668464)

[Hình 3.8 Sau khi phát hiện khuôn mặt 13](#_Toc74668465)

[Hình 3.9 Dữ liệu được lưu 13](#_Toc74668466)

[Hình 3.10 Kiến trúc dạng khối của mô hình CNN 14](#_Toc74668467)

[Hình 3.11 Một ảnh đầu vào kích thước 100x100x1 (đa cấp xám) 15](#_Toc74668468)

[Hình 3.12 Hình ảnh sau khi xử lý của khối B2 16](#_Toc74668469)

[Hình 3.13 Hình ảnh kết quả xử lý sau B4 và B6 17](#_Toc74668470)

[Hình 3.14 Dữ liệu phân lớp của mô hình CNN 18](#_Toc74668471)

[Hình 3.15 Dữ liệu danh tính 19](#_Toc74668472)

[Hình 4.1 Hơn 20 triệu trọng số được huấn luyện 20](#_Toc74668473)

[Hình 4.2 Độ chính và giá trị hàm lỗi theo từng epoch 21](#_Toc74668474)

[Hình 4.3 Hình ảnh trước khi nhận diện 21](#_Toc74668475)

[Hình 4.4 Hình ảnh sau khi nhận diện (Trường Giang) 22](#_Toc74668476)

[Hình 4.5 Hình ảnh trước khi nhận diện 5 người 22](#_Toc74668477)

[Hình 4.6 Hình ảnh sau khi nhận diện 5 người 23](#_Toc74668478)

CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| ReLu | Rectified Linear Unit |
| CNN | Convolutional Neural Network |
| OpenCV | Open Source Computer Vision |
| CV | Computer Vision |
| MLL | Machine Learning Library |
| CONV | Convolution |

# Giới thiệu

## Giới thiệu

CNN là một trong những mô hình mạng Học sâu phổ biến nhất hiện nay, có khả năng nhận dạng và phân loại hình ảnh với độ chính xác rất cao, thậm chí còn tốt hơn con người trong nhiều trường hợp. Mô hình này đã và đang được phát triển, ứng dụng vào các hệ thống xử lý ảnh lớn của Facebook, Google hay Amazon… cho các mục đích khác nhau, như các thuật toán gắn thẻ tự động, tìm kiếm ảnh hoặc gợi ý sản phẩm cho người tiêu dùng.

Mạng CNN với kiến trúc thay đổi, có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến node trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng nơ ron truyền thẳng.

Nhờ sự phát triển mạng CNN, chúng tôi có cơ sở thực hiện đề tài “Nhận diện khuôn mặt”.

## mục tiêu đề tài

Đề tài “Nhận diện khuôn mặt” có thể áp dụng trong các hệ thống:

- Hệ thống an ninh

- Phân tích dữ liệu

## GIỚI HẠN ĐỀ TÀI

Phạm vi giới hạn của đề tài:

Đề tài chủ yếu sử dụng các thư viện có sẵn trong lập trình python, đồng thời ứng dụng các mô hình toán học trong mạng nơ ron nhân tạo. Điều kiện đầu vào của mô hình là các tệp hình ảnh có giới hạn, đầu ra phân tích khuôn mặt được đưa vào có độ chính xác tương đối. Đối với chức năng thì có sự giới hạn do hiểu biết về lập trình và cách ứng dụng của mô hình vào các hệ thống lớn.

## PHƯƠNG PHÁP NGHIÊN CỨU

Trong phần này, chúng tôi thiết kế mô hình nhận dạng khuôn mặt dựa trên mạng CNN. Dựa trên các phương pháp nghiên cứu chính như phương pháp phân tích, tham khảo tài liệu, phương pháp tổng hợp tài liệu lý thuyết…

## ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Các đối tượng cần nghiên cứu và phạm vi nghiên cứu có thể giải quyết được đề tài:

- *Đối tượng nghiên cứu:* Trí tuệ nhân tạo, lập trình python.

- *Phạm vi nghiên cứu:* Mạng CNN, các module trong python: opencv, numpy, os, tensorflow, image.

## BỐ CỤC QUYỂN BÁO CÁO

Nội dung báo cáo:

CHƯƠNG 1: GIỚI THIỆU

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

CHƯƠNG 3: XÂY DỤNG CHƯƠNG TRÌNH

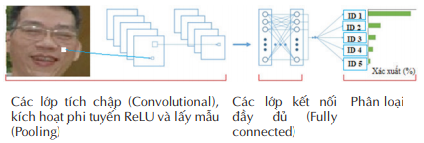
CHƯƠNG 4: KẾT QUẢ THÍ NGHIỆM

# CƠ SỞ LÝ THUYẾT

## Giới thiệu mạng cnn

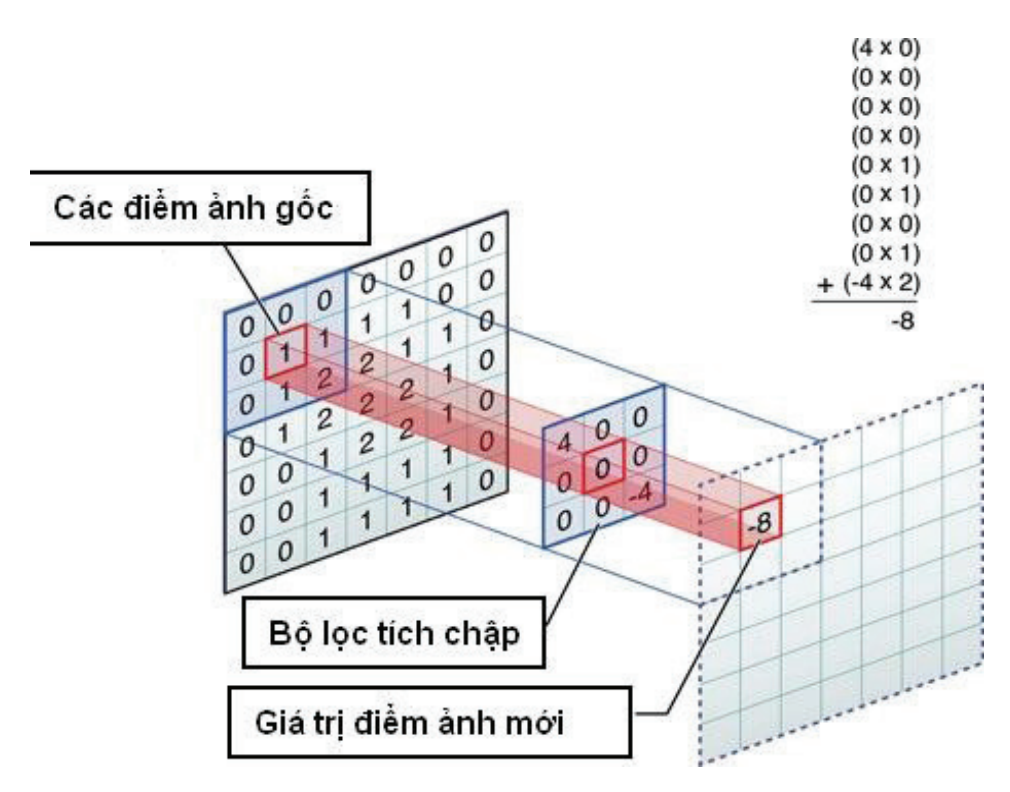
Hình 2.1 trình bày một kiến trúc mạng CNN, các lớp cơ bản trong một mạng CNN bao gồm: lớp tích chập (Convolutional); lớp kích hoạt phi tuyến ReLU (Rectified Linear Unit); lớp lấy mẫu (Pooling); lớp kết nối đầy đủ (Fully connected) được thay đổi về số lượng và cách sắp xếp để tạo ra các mô hình huấn luyện phù hợp cho từng bài toán khác nhau. Các lớp tích chập (Convolutional), kích hoạt phi tuyến ReLU và lấy mẫu (Pooling) Các lớp kết nối đầy đủ (Fully connected) Phân loại Hình 1. Kiến trúc cơ bản của một mạng CNN.

Lớp tích chập: đây là thành phần quan trọng nhất trong mạng CNN, thể hiện sự liên kết cục bộ thay vì kết nối toàn bộ các điểm ảnh. Các liên kết cục bộ được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc filters có kích thước nhỏ.



Hình 2.1 Kiến trúc cơ bản của mạng CNN

Lớp tích chập: đây là thành phần quan trọng nhất trong mạng CNN, thể hiện sự liên kết cục bộ thay vì kết nối toàn bộ các điểm ảnh. Các liên kết cục bộ được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc filters có kích thước nhỏ.



Hình 2.2 Bộ lọc tích chập được sử dụng trên ma trận điểm ảnh

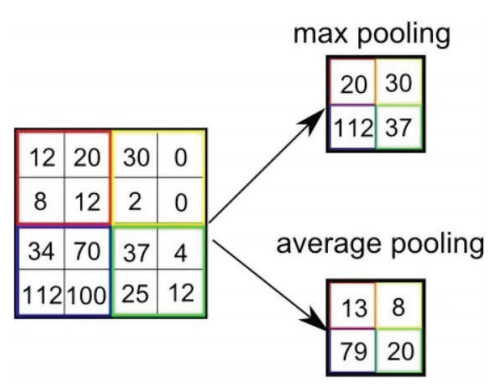
Bộ lọc tích chập được sử dụng trên ma trận điểm ảnh. Trong hình 2.2, bộ lọc được sử dụng là một ma trận có kích thước 3x3, bộ lọc này dịch chuyển lần lượt qua từng vùng ảnh đến khi hoàn thành quét toàn bộ bức ảnh, tạo ra một bức ảnh mới có kích thước nhỏ hơn hoặc bằng với kích thước ảnh đầu vào. Kích thước này được quyết định tùy theo kích thước các khoảng trắng được thêm ở viền bức ảnh gốc và được tính theo công thức sau:

Trong đó: O: kích thước ảnh đầu ra; i: kích thước ảnh đầu vào; p: kích thước khoảng trắng phía ngoài viền của ảnh gốc; k: kích thước bộ lọc; s: bước trượt của bộ lọc. Như vậy, sau khi đưa một bức ảnh đầu vào cho lớp tích chập nhận được kết quả đầu ra là một loạt ảnh tương ứng với các bộ lọc đã được sử dụng để thực hiện phép tích chập. Các trọng số của các bộ lọc này được khởi tạo ngẫu nhiên trong lần đầu tiên và sẽ được cập nhật trong quá trình huấn luyện. - Lớp kích hoạt phi tuyến ReLU: được xây dựng để đảm bảo tính phi tuyến của mô hình huấn luyện sau khi đã thực hiện một loạt các phép tính toán tuyến tính qua các lớp tích chập. Lớp kích hoạt phi tuyến sử dụng các hàm kích hoạt phi tuyến như ReLU hoặc sigmoid, tanh… để giới hạn phạm vi biên độ cho phép của giá trị đầu ra. Trong số các hàm kích hoạt này, hàm ReLU được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả. Phép tính toán của hàm ReLU chỉ đơn giản là chuyển tất cả các giá trị âm thành giá trị 0. Lớp ReLU được áp dụng ngay phía sau lớp tích chập, với đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh cũng hoàn toàn tương tự, trừ các giá trị âm đã bị loại bỏ.

𝑓(𝑥) = 𝑚𝑎𝑥(0, 𝑥) (2.2)

Lớp lấy mẫu: được đặt sau lớp tích chập và lớp ReLU để làm giảm kích thước ảnh đầu ra trong khi vẫn giữ được các thông tin quan trọng của ảnh đầu vào. Việc giảm kích thước dữ liệu có tác dụng làm giảm được số lượng tham số cũng như tăng hiệu quả tính toán. Lớp lấy mẫu cũng sử dụng một cửa sổ trượt để quét toàn bộ các vùng trong ảnh như lớp tích chập, và thực hiện phép lấy mẫu thay vì phép tích chập, sẽ chọn lưu lại một giá trị duy nhất đại diện cho toàn bộ thông tin của vùng ảnh đó.

Hình 3 thể hiện các phương thức lấy mẫu thường được sử dụng nhất hiện nay, đó là Max Pooling (lấy giá trị điểm ảnh lớn nhất) và Avarage Pooling (lấy giá trị trung bình của các điểm ảnh trong vùng ảnh cục bộ).



Hình 2.3 Phương thức Avarage Pooling và Max Pooling

Như vậy, với mỗi ảnh đầu vào được đưa qua lấy mẫu sẽ thu được một ảnh đầu ra tương ứng, có kích thước giảm xuống đáng kể nhưng vẫn giữ được các đặc trưng cần thiết cho quá trình tính toán và nhận dạng.

Lớp kết nối đầy đủ: được thiết kế tương tự như trong mạng nơ ron truyền thống, tất cả các điểm ảnh được kết nối đầy đủ với node trong lớp tiếp theo.

So với mạng nơ ron truyền thống, các ảnh đầu vào của lớp này đã có kích thước được giảm bớt rất nhiều, đồng thời vẫn đảm bảo các thông tin quan trọng của ảnh cho việc nhận dạng. Do vậy, việc tính toán nhận dạng sử dụng mô hình truyền thẳng đã không còn phức tạp và tốn nhiều thời gian như trong mạng nơ ron truyền thống.

## hàm opencv

### OpenCV là gì?

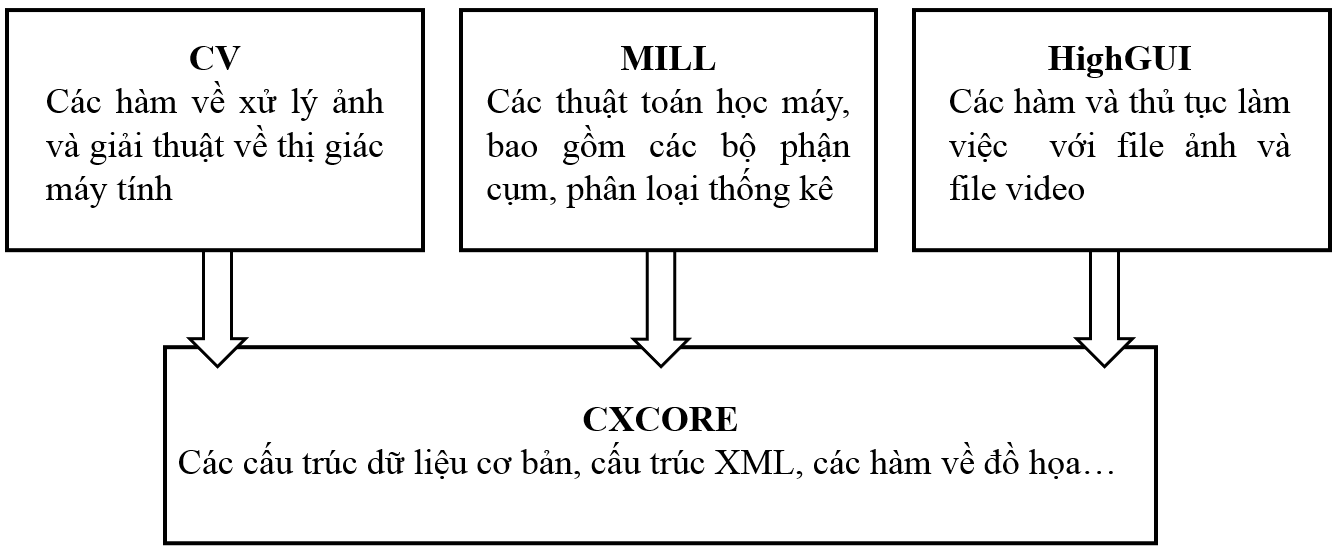
Opencv viết tắt của từ Open Source Computer Vision Library.

* OpenCV là một thư viện mã nguồn mở phục vụ cho việc nghiên cứu hay phát triển về thị giác máy tính.
* Tối ưu hóa và xử lý ứng dụng trong thời gian thực.
* Giúp cho việc xây dựng các ứng dụng xử lý ảnh, thị giác máy tính ... một cách nhanhhơn.
* OpenCV có hơn 500 hàm khác nhau, được chia làm nhiều phần phục vụ các công việc như: xử lý ảnh, an ninh, camera quan sát, nhận diện, robot...

Thư viện được viết bằng ngôn ngữ C và C++ có thể chạy trên các hệ điều hành Linux, Window và MacOsX. OpenCV được thiết kế để nâng cao hiệu suất tính toán và nhấn mạnh đến hệ thống thời gian thực. OpenCV đưa ra một hệ thống đơn giẩn, dễ sử dụng giúp mọi người nhanh chóng xây dựng các ứng dụng trong thị giác máy, kể cả các hệ thống kiểm tra trong nhà máy, bức ảnh trong lĩnh vực y học, bảo mật, robot học... Nó chứa các lập trình xử lý ảnh đơn giản, kể cả thực thi các hàm bậc cao như dò tìm khuôn mặt, theo dõi khuôn mặt, nhận dạng khuôn mặt...OpenCV được giới thiệu vào tháng 1/1999, OpenCV đã được sử dụng trong rất nhiều ứng dụng, các sản phẩm và các nghiên cứu như: trong lĩnh vực hàng không, sử dụng giảm nhiễu trong y học, phân tích đối tượng, an ninh, hệ thống dò tìm, theo dõi tự động và hệ thống bảo mật...., ngoài ra nó còn được sử dụng trong nhận dạng âm thanh. OpenCV còn là một chìa khóa quan trọng trong các robot sử dụng thị giác như Stanford, Asimo.

### Cấu trúc OpenCV

Cấu trúc của opencv được chia làm 5 phần chính, 4 trong số đó được chia ra như trong hình sau:



Hình 2.4 Cấu trúc cơ bản của OpenCV

CV (computer vision) là thành phần chữa những xử lý ảnh cơ sở và thuật toán thị giác máy tính mức cao.

MLL (machine learning library) là thư viện machine learning, cái này bao gồm rất nhiều lớp thống kê và gộp công cụ xử lý.

HighGUI chứa các thủ tục vào ra và các hàm dùng cho việc lưu trữ và tải ảnh video.

CXCore chứa cấu trúc và nội dung dữ liệu cơ sở.

### Phát hiện khuôn mặt với các hàm trong OpenCV

Từ các phương pháp được đưa ra ở phần trước, bài toán phát hiện khuôn mặt được thực hiện theo các bước:

* Huấn luyện bộ tập mẫu
* Xác định khuôn mặt

Ở bước huấn luyện bộ tập mẫu. Trong khi cài đặt OpenCV 4.4.0 đã có sẵn thư mục bộ huấn luyện có sẵn. Ta sẽ sử dụng luôn tài nguyên này. Ở bước xác định khuôn mặt sẽ chia làm 3 bước:

* Tiền xử lí
* Dò tìm khuôn mặt
* Hậu xử lí

Ta sử dụng luôn bộ huấn luyện có sẵn trong thư mục của OpenCV

Hình 2.5 Trong thư mục của OpenCV đã có các bộ huấn luyện có sẵn

## Hàm OS

### Hàm os là gì?

“os” có thể coi là một module bao gói các lệnh gọi POSIX hoặc mô phỏng chúng trên một số nền tảng mà không có POSIX. Bạn sẽ thấy các hàm gần như được đặt tên giống hệt như trong C và các ngôn ngữ khác cũng như cách thực hiện tương tự. Có thể dễ dàng nhận ra các hằng số như os.R\_OK, os.WNOHANG ... được đặt tên chính xác theo C/POSIX

### Tạo folder bằng os

os.path.exists() cho phép bạn có thể nối các đường dẫn với nhau theo để tạo nên 1 đường dẫn hoàn chỉnh và phù hợp nhất. Thay vì phải đau đầu sửa code để tạo đường dẫn theo từng hệ điều hành khác nhau, phương thức này sẽ giúp bạn tạo ra đường dẫn phù hợp nhất. Hàm này tạo thư mục theo đường dẫn được truyền vào.



Hình 2.6 Trong thư mục của “os” đã có các bộ huấn luyện có sẵn

# xây dựng chương trình

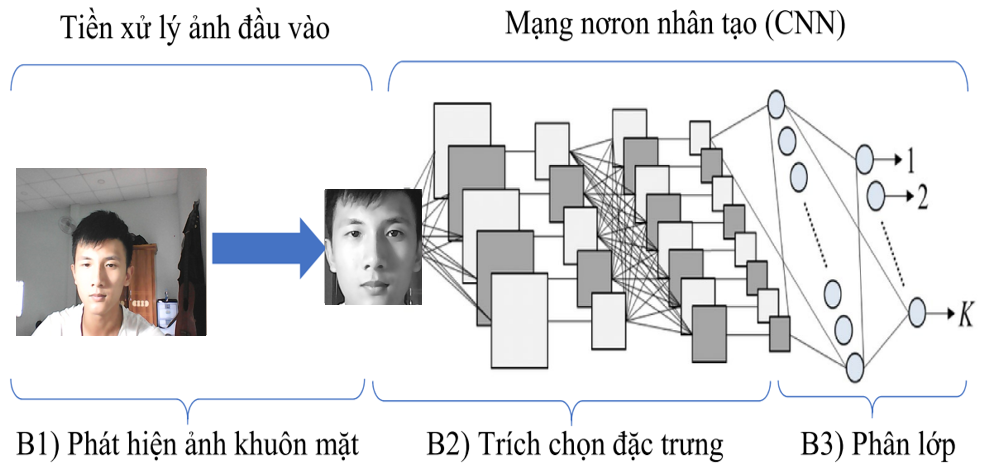
## xây dựng mô hình

Mô hình nhận dạng được chia thành 3 bước chính (Hình 3.1), bao gồm:

Bước 1: Tạo tập dữ liệu.

Bước 2: Huấn luyện dữ liệu đưa vào và trích chọn các đặc trưng.

Bước 3: Phân loại ảnh khuôn mặt dựa trên đặc trưng được trích chọn và đưa ra kết quả.



Hình 3.1 Sơ đồ quy trình của mô hình nhận dạng khuôn mặt

## Các bước thực hiện

### Tạo tập dữ liệu

Các dữ liệu dùng để huấn luyện có thể lấy từ ảnh có sẵn hoặc lấy từ webcam. Các bước phát hiện khuôn mặt và lấy dữ liệu là giống nhau.

Bước 1: Tiền xử lý

- Phương pháp thực hiện trên ảnh xám (gray image). Mỗi điểm ảnh (pixel) sẽ có giá trị mức xám từ 0 đến 255(không gian màu 8 bit). Như vậy phương pháp sẽ không khai thác những đặc điểm về màu sắc khuôn mặt để nhận dạng song vẫn rất hiệu quả. Ảnh màu sẽ được chuyển về ảnh xám để nhận dạng, việc chuyển đổi này khá đơn giản, thực hiện bằng một hàm chuyển đổi và sử dụng chỉ với một câu lệnh trong OpenCV nên báo cáo chưa đề cập tới. Sau khi chuyển thành ảnh xám, ảnh lại tiếp tục được chuyển thành “ảnh tích hợp” và trong bước đầu tiên của quá trình nhận dạng.



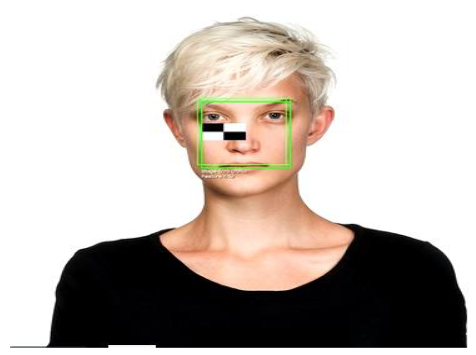
Hình 3.2 Chuyển về ảnh xám

Bước 2: Dò tìm khuôn mặt

Integal Image là mảng 2 chiều với kích thước ảnh cần tính đặc trưng Haar. Bắt đầu từ vị trí trên bên trái đến vị trí dưới, bên phải của ảnh. Sau khi đã tính được Integral Image, việc tính tổng điểm ảnh của một vùng bất kỳ nào đó trên ảnh thực hiện sẽ đơn giản hơn.

Để phát hiện khuôn mặt, hệ thống sẽ cho một cửa sổ con (sub-window) có kích thước cố định quét lên toàn bộ ảnh đầu vào. Như vậy sẽ có rất nhiều ảnh con ứng với từng cửa sổ con, các đặc trưng Haar-like sẽ được đặt lên các cửa sổ con này để từ đó tính ra giá trị của đặc trưng. Sau đó các giá trị này được bộ huấn luyện xác nhận xem khung hình đó có phải khuôn mặt hay không.

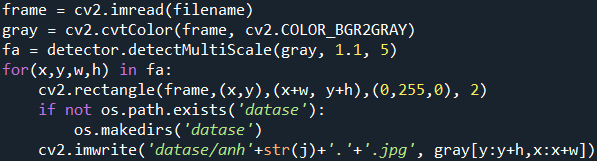
Hình dưới là một ví dụ: khung màu xanh là cửa sổ con, một đặc trưng Haar-like với kích thước và vị trí đặt như trong hình sau:



Hình 3.3 Tầm nhìn của máy tính khi các đặc trưng Haar chạy

Bước 3: Hậu xử lí

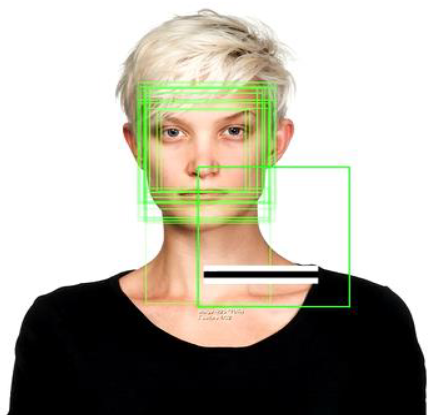
Sau khi quét các cửa sổ con khắp bức ảnh. Đặc trưng Haar sẽ so sánh với bộ huấn luyện mẫu và sẽ đánh dấu những nơi có thể là khuôn mặt trong ảnh.



Hình 3.4 Tạo khung khuôn mặt

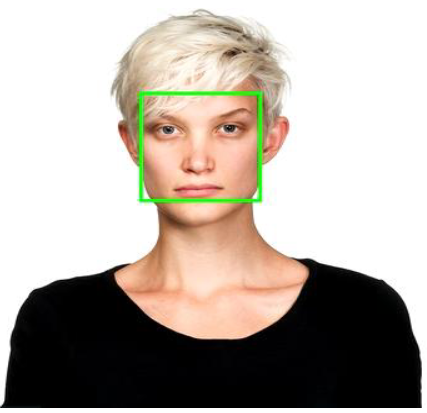
Hàm detectMultiScale là phần tìm kiếm khuôn mặt, hàm (hay phương thức) này thuộc lớp CascadeClassifier (lớp phục vụ tìm kiếm đối tượng của OpenCV)

Hàm detectMultiScale sau khi tìm kiếm xong sẽ trả về bộ giá trị gồm tọa độ gốc của khung chứa khuôn mặt x, y; chiều dài, rộng của khung w, h. Các giá trị này nằm trong mảng faces. Cấu trúc for...sẽ duyệt qua toàn bộ các bộ giá trị này,với mỗi bộ giá trị ta dùng hàm rectangle để vẽ một hình chữ nhật lên ảnh ban đầu với tọa độ 2 điểm trái trên và phải dưới: (x,y),(x+w,y+h). (0,0,255) là màu sẽ vẽ hình chữ nhật.



Hình 3.5 Các đặc trưng Haar đánh dấu những nơi có thể là khuôn mặt

Sau khi đã đánh dấu hết những nơi có thể là khuôn mặt thì sẽ gộp lại những hình chữ nhật chứa cùng một khuôn mặt để chỉ cho ra một hình chữ nhật cho 1 khuôn mặt.

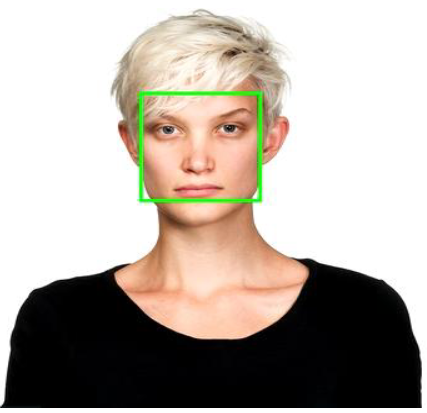


Hình 3.6 Bức ảnh đã được xác định khuôn mặt

- Kết quả của quá trình tạo tập dữ liệu:



Hình 3.7 Đưa bức ảnh cần nhận diện vào chương trình

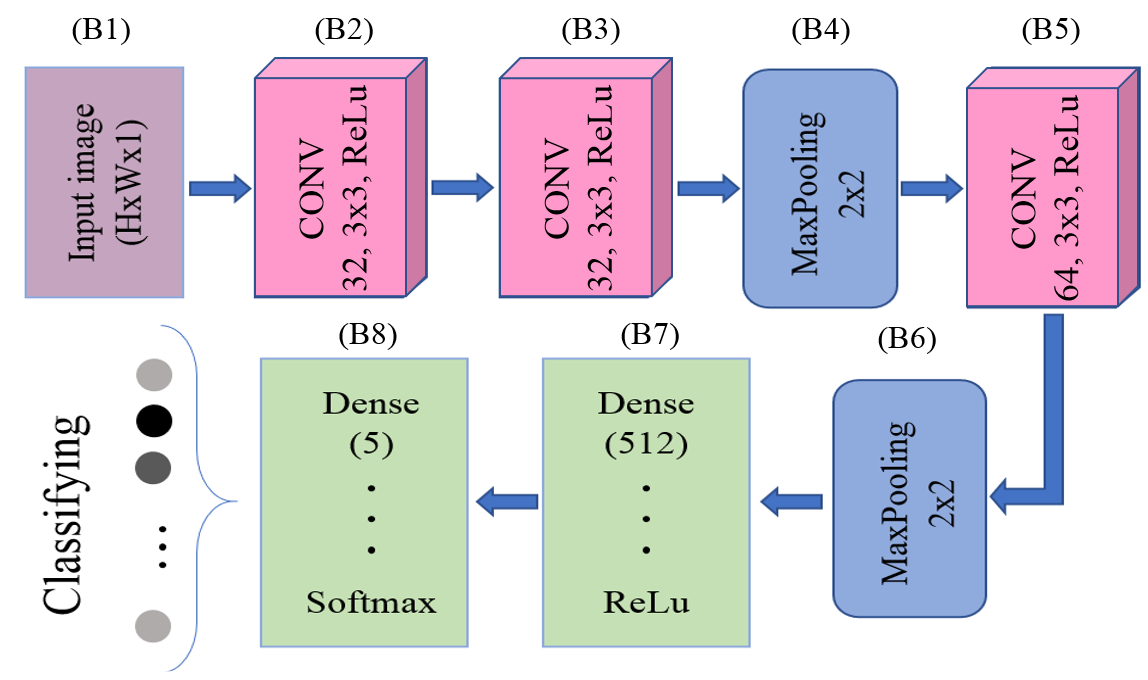


Hình 3.8 Sau khi phát hiện khuôn mặt



Hình 3.9 Dữ liệu được lưu

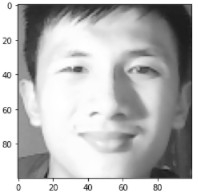
### Huấn luyện dữ liệu đưa vào

Mô hình CNN được thiết kế gồm hai phần chức năng là trích chọn đặc trưng của ảnh khuôn mặt và phân lớp đối tượng dựa trên đặc trưng đã chọn. Mô hình CNN bao gồm nhiều lớp, số lớp nơron và độ lớn (số nơron) của mỗi lớp ảnh hưởng đến chất lượng cũng như độ phức tạp trong tính toán của mạng nơron. Các nghiên cứu thường điều chỉnh hai yếu tố này tuỳ theo bài toán ứng dụng để đạt được chất lượng mong muốn và đồng thời đảm bảo sự phức tạp tính toán chấp nhận được. 

Hình 3.10 Kiến trúc dạng khối của mô hình CNN

Mỗi lớp nơron trong mô hình CNN lấy một mảng nhiều chiều gồm các số làm đầu vào và tạo ra một mảng số nhiều chiều khác ở đầu ra (sau đó trở thành đầu vào của lớp tiếp theo). Khi phân loại hình ảnh khuôn mặt, đầu vào của lớp nơron đầu tiên là kích thước hình ảnh đầu vào. Kích thước đầu ra của lớp cuối cùng là tập hợp các khả năng của các lớp khác nhau được phân loại cho mỗi ảnh đầu vào. Cả ba loại lớp nơron để xây dựng kiến trúc của CNN bao gồm: 3 lớp tích chập (CONV), 2 lớp nơron Maxpooling và 2 lớp nơron kết nối đầy đủ để phân loại (gọi là lớp Dense). Mỗi lớp CONV được kết nối theo sau nó bởi một lớp Maxpooling, áp dụng cơ chế kích hoạt ReLu (Rectified Linear Unit, mặc định là max(x,0)) sau mỗi lớp CONV để đảm bảo đầu vào không âm cho lớp nơron kế tiếp. Theo nguyên tắc xếp chồng các lớp nơron và giảm không gian mẫu (downsampling) tại các kết quả đầu ra của chúng, CNN thực hiện trích xuất các đặc trưng ngày càng trừu tượng và phức tạp hơn, đồng thời, là bất biến đối với các phép biến dạng và chuyển đổi. Mạng tích chập CNN trong mô hình này được chia thành 8 khối (Hình 3.10).

• Khối B1 là ảnh đầu vào có kích thước H×W×1 (cao × rộng × sâu). Để giảm bớt không gian và bộ nhớ của quá trình tính toán mạng nơron nên chúng tôi sử dụng ảnh đầu vào đa cấp xám (số chiều thứ 3 (độ sâu) trong kích thước ảnh bằng 1). Hình vẽ sau minh hoạ cho một ảnh đầu vào:



Hình 3.11 Một ảnh đầu vào kích thước 100x100x1 (đa cấp xám)

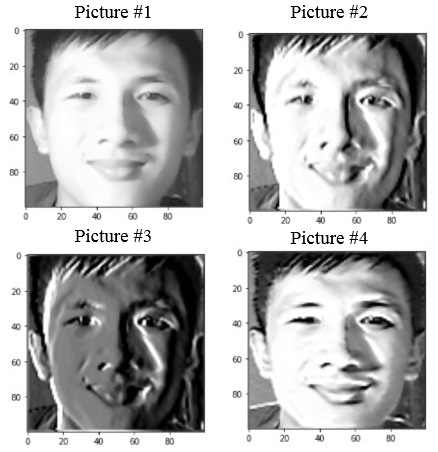
• Khối B2, B3 là lớp nơron tích chập có 2 bộ lọc đặc trưng với kích thước cửa sổ hàm nhân là 3×3. Hàm kích hoạt ReLu được sử dụng trong lớp nơron này. Hiệu ứng này nhằm cải thiện các đặc trưng thưa của toàn mạng và tránh sự phụ thuộc vào việc truyền tham số giữa các nơron.

• Khối B4 là lớp Maxpooling, cửa sổ xử lý có kích thước 2×2 được sử dụng. Lớp giảm không gian mẫu (downsampling) này sử dụng phương pháp MaxPooling với việc có thể giữ lại thông tin hữu ích và cắt giảm lượng dữ liệu cần xử lý ở bước tiếp theo.

Hình 3.11 minh hoạ kết quả xử lý của lớp nơron tích chập CONV ở khối B2 và lớp MaxPooling ở khối B3.

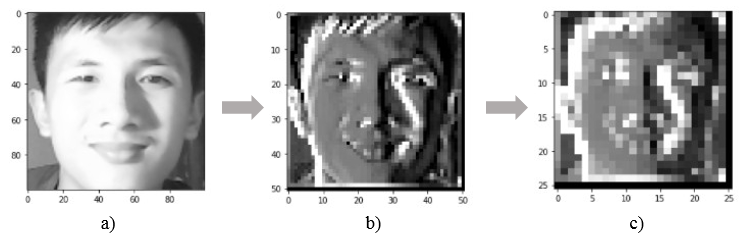
Khối B5 tương tự khối B3 nhưng số các bộ lọc (filter) tăng lên 4, sau đó khối này kết nối ngay theo sau khối tích chập CONV để thực hiện cắt giảm không gian mẫu.

Thông thường, chúng ta càng có nhiều bước sử dụng phép tích chập thì cơ hội trích chọn càng nhiều đặc trưng phức tạp hơn, qua đó kỳ vọng mô hình đề xuất có thể học để nhận biết đối tượng ở mức tốt hơn. Chẳng hạn, trong phân loại hình ảnh, mô hình CNN có thể học để phát hiện các đặc trưng cạnh từ các pixel thô trong lớp CONV đầu tiên, sau đó sử dụng các đặc trưng cạnh này để phát hiện các đặc trưng hình dạng đơn giản trong lớp CONV thứ hai, sau đó sử dụng các đặc trưng hình dạng này để phát hiện các đặc trưng ở mức cao hơn, chẳng hạn như hình dạng khuôn mặt ở các lớp cao hơn.



Hình 3.12 Hình ảnh sau khi xử lý của khối B2

Trong Hình 3.13/a/b/c dưới đây minh hoạ kết quả xử lý ở bộ lọc đầu tiên của các khối B3, B5 cùng với lớp nơron MaxPooling ngay sau chúng (tương ứng ở B4, B6), do đó kích thước của hình ảnh sau mỗi bước xử lý giảm dần với hệ số ½ (sau B4 là 50x50, sau B6 là 25x25). Kết quả minh hoạ trực quan cho thấy càng về sau hình ảnh càng “mờ”, thể hiện khả năng trừu tượng hoá và biểu diễn các đặc trưng chung nhất của khuôn mặt một cá nhân, dù được chụp dưới góc độ nào. Hay có thể nói rằng các đặc trưng này của cá nhân có độ bất biến cao nhất đối với bất kỳ hình ảnh khác nhau nào của họ dù dưới các dạng thức, độ sáng tối, màu sắc, kích thước khác nhau.



Hình 3.13 Hình ảnh kết quả xử lý sau B4 và B6

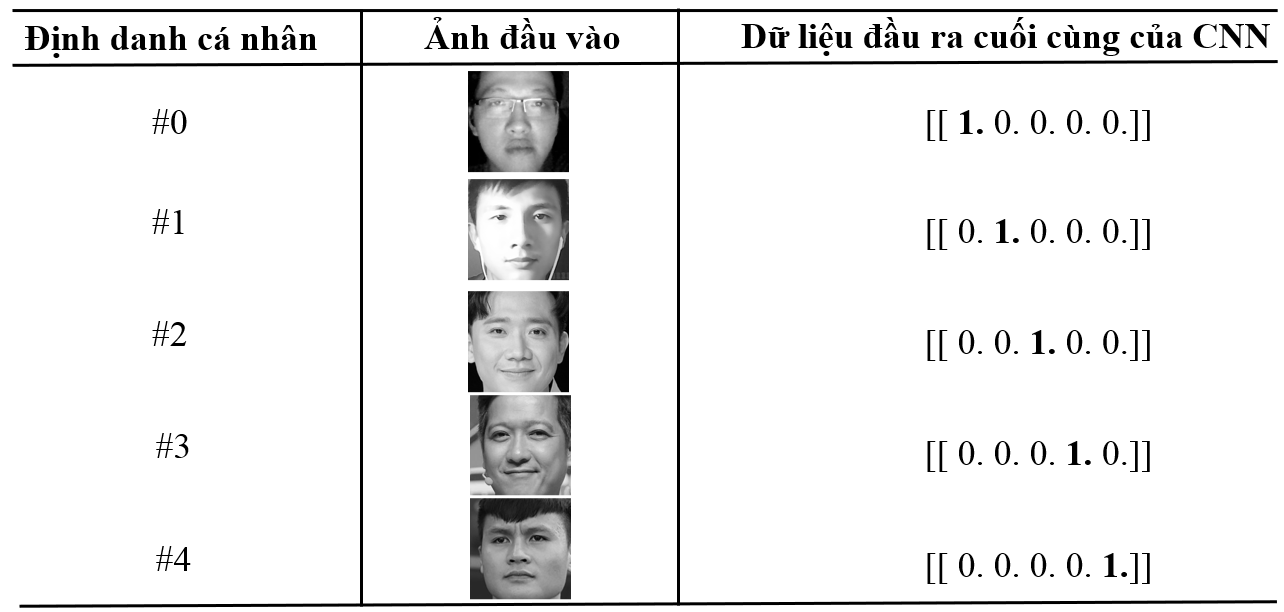
•Khối B7 là lớp nơron kết nối đầy đủ (fully connected layers). Lớp này cùng với lớp trong khối B8 nhằm mục tiêu phân lớp các đặc trưng được trích chọn ở các lớp trước, do đó chúng tôi thiết kế số nơron đủ lớn. Hàm kích hoạt ReLu cũng được áp dụng. Dữ liệu sau đây là đầu ra của 512 nơron đầu tiên trong khối B7, các giá trị ở đây được xem như biểu diễn dạng số của các đặc trưng khuôn mặt tương ứng với ảnh đầu vào.

• Khối B8 là lớp nơron đầu ra cuối cùng, đây chính là một phân bố cho phân loại của các lớp đối tượng khác nhau với tính năng của hàm kích hoạt Softmax. Áp dụng phương pháp hồi quy Softmax ở lớp đầu ra của mạng CNN nhằm thẩm định cho quá trình huấn luyện mạng. Cơ chế thẩm định này đảm bảo mô hình CNN không bị quá khớp (overfiting) dữ liệu học và có khả năng dự đoán tốt hơn. Cơ chế này được thực hiện thông qua việc chia dữ liệu học thành hai phần, một phần để tính toán cập nhật và điều chỉnh trọng số mạng, một phần để tính toán sai số và cũng đưa vào pha cập nhật thay đổi trọng số mạng. Đầu ra phân lớp cuối cùng của mô hình được xác định dựa trên giá trị cực đại hàm Softmax của nơron tương ứng, với mô hình CNN có lớp (tức là có nơron ở lớp ra) thì ta có công thức xác định như sau:

outputclassified = argmaxCk{Ok­­: k = 1, …, |Ok|} (3.1)

Trong đó, Ok là đầu ra của nơron thứ ở lớp nơron cuối cùng và tương ứng với nó là lớp đối tượng Ck.

Bảng 3.1 minh hoạ dữ liệu gồm 5 giá trị sau là kết quả đầu ra của lớp nơron cuối cùng (khối B8), tương ứng với kết quả phân lớp của ảnh đầu vào. Kết quả đầu ra của mô hình CNN tương ứng với mỗi ảnh đầu vào gồm 5 giá trị (ở đây chỉ minh hoạ với 5 lớp). Giá trị cao nhất ở vị trí nào (tính theo chỉ số từ 0 và từ trái sang phải) trong bộ 5 giá trị đầu ra chính là định danh cá nhân (vị trí tương ứng giá trị và số in đậm) của dữ liệu ảnh đầu vào.

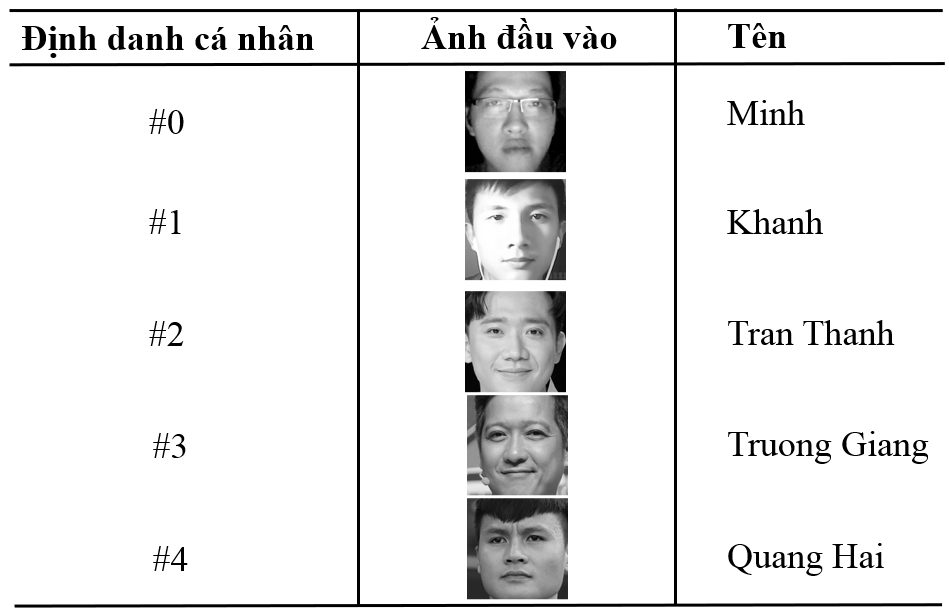


Hình 3.14 Dữ liệu phân lớp của mô hình CNN

### Hiển thị dữ liệu đầu ra

Sau khi mô hình được huấn luyên, lưu module vào file “khuonmat.h5”, sau đó dùng file này để nhận diện và dự đoán khuôn mặt của hình ảnh đưa vào.

Từ dữ liệu đầu ra của module, đặt tên cho khuôn mặt được nhận diện riêng biệt. Từ đó sử dụng hàm có sẵn trong OpenCV để hiển thị chữ phía dưới hình vuông nhận diện khuôn mặt. Cách đặt tên tương ứng với hình ảnh như hình 3.15.

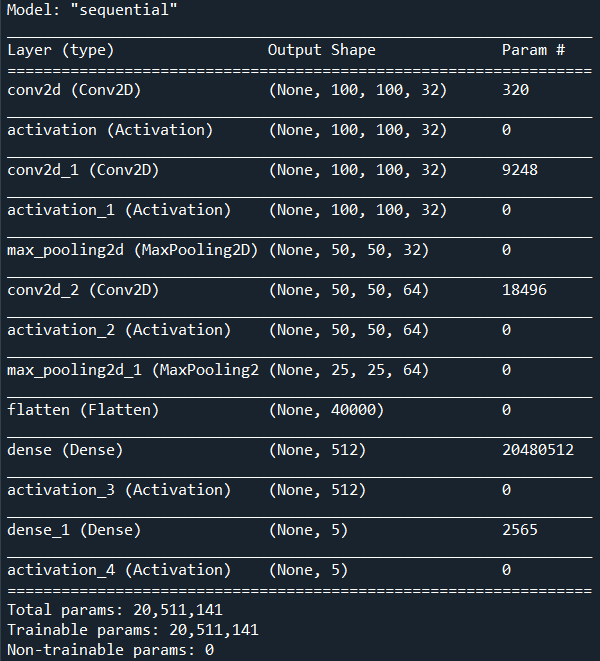


Hình 3.15 Dữ liệu danh tính

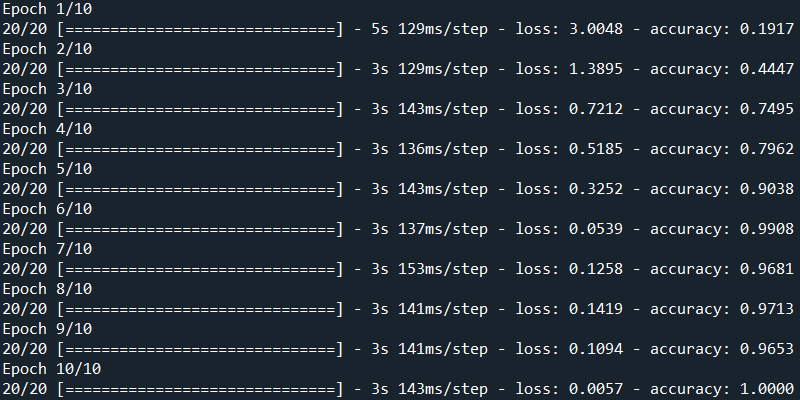
# Kết quả thí nghiệm

## Kết quả sau khi huấn luyện

Quá trình huấn luyện tập dữ liệu:



Hình 4.1 Hơn 20 triệu trọng số được huấn luyện



Hình 4.2 Độ chính và giá trị hàm lỗi theo từng epoch

\*Nhận xét:

- Giá trị của hàm lỗi giảm dần theo từng epoch. Giá trị của hàm lỗi ở epoch 1 là 3.0048, giá trị hàm lỗi ở epoch 10 giảm xuống còn 0.0057.

- Độ chính xác tăng dần qua từng epoch, tăng lên tối đa 100% ở epoch thứ 10.

## Đưa hình ảnh một người cần nhận diện

- So sánh kết quả trước và sau khi nhận diện 1 một khuôn mặt:

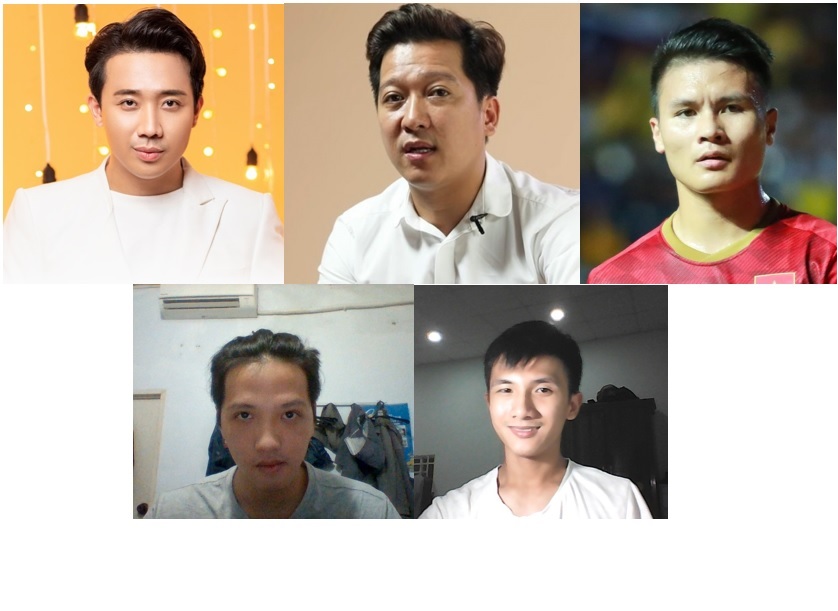


Hình 4.3 Hình ảnh trước khi nhận diện

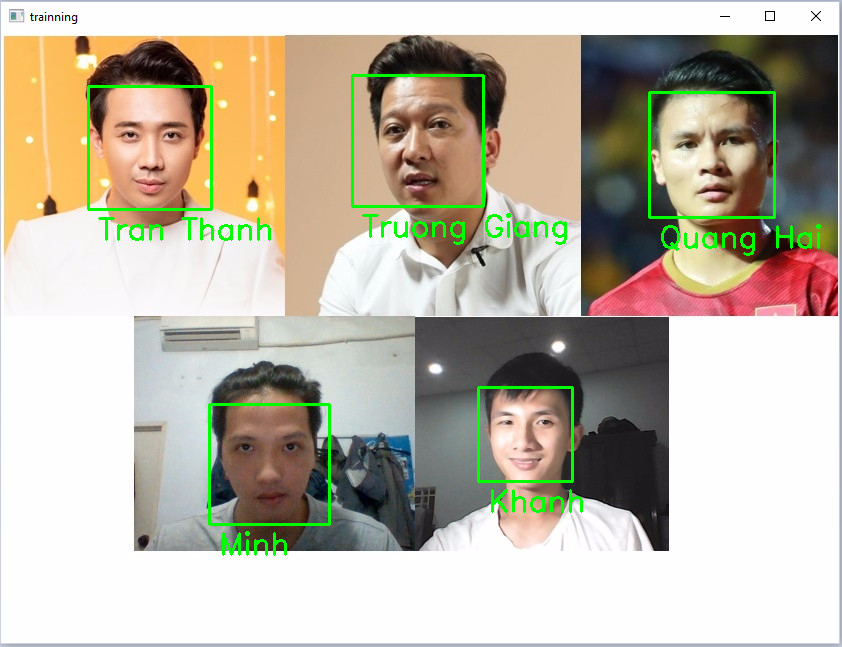


Hình 4.4 Hình ảnh sau khi nhận diện (Trường Giang)

- So sánh kết quả trước và sau khi nhận diện nhiều một khuôn mặt:



Hình 4.5 Hình ảnh trước khi nhận diện 5 người



Hình 4.6 Hình ảnh sau khi nhận diện 5 người

\* Nhận xét:

- Đối với nhận diện một khuôn mặt, kết quả cho ra là chính xác đồng thời phát hiện được khuôn mặt cần nhận diện

- Đối với nhận diện nhiều khuôn mặt, kết quả cho ra tương tự như nhận diện một khuôn mặt và có độ chính xác cao.

# KẾT LUẬN

Trong báo cáo này, chúng tôi đã đề xuất một mô hình dựa trên mạng nơron tích chập (CNN) để nhận dạng khuôn mặt con người. Mô hình này có 3 lớp nơron tích chập (Convolution) và 2 lớp nơron liên kết đầy đủ (Fully Connected), tổng số tham số là khoảng hơn 20 triệu.

Như vậy, có thể khẳng định mô hình của chúng tôi có độ phức tạp ở mức vừa phải, phù hợp với các hệ thống xử lý ở mức trung bình và đem lại tiềm năng khả thi trong ứng dụng thực tiễn.

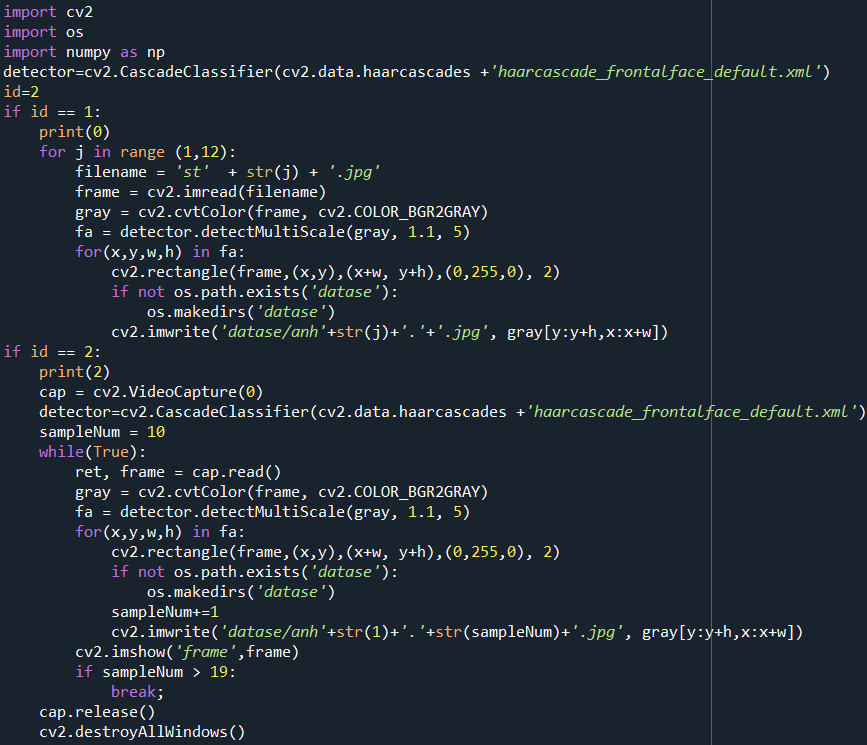
Mặc dù độ phức tạp của mô hình ở mức thấp so với các mô hình khác, nhưng kết quả thử nghiệm cho thấy tính hiệu quả của phân lớp khá cao. Hiện nay do điều kiện tính toán nên chỉ áp dụng số lần huấn luyện còn thấp, nếu được huấn luyện ở mức độ sâu hơn thì kỳ vọng sẽ đem lại kết quả cao hơn nữa.

Để phát triển thêm cho mô hình, chúng tôi sẽ tìm hiểu và thiết kế một hệ thống thu thập dữ liệu hình ảnh để tạo bộ dữ liệu huấn luyện đa dạng cho mô hình, từ đó xây dựng một ứng dụng cho bài toán thực tiễn như hệ thống điểm danh sinh viên có mặt ở lớp học, hệ thống giám sát cán bộ vào/ra cổng cơ quan, hệ thống theo dõi và định danh liên tục quá trình học tập của người học trực tuyến.

# PHỤ LỤC

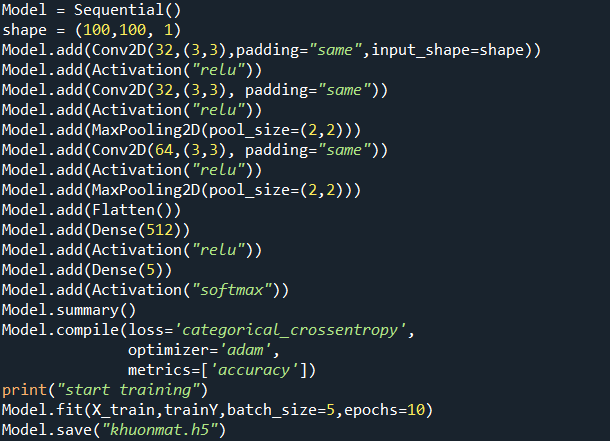
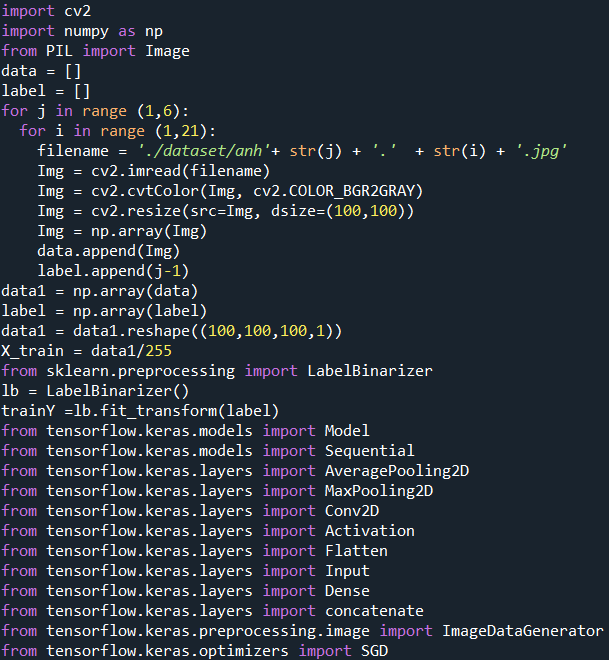
Chương trình được thực hiện trên môi trường Spyder (Anaconda) với 3 chương trình như sau:

+ Tạo tập dữ liệu:

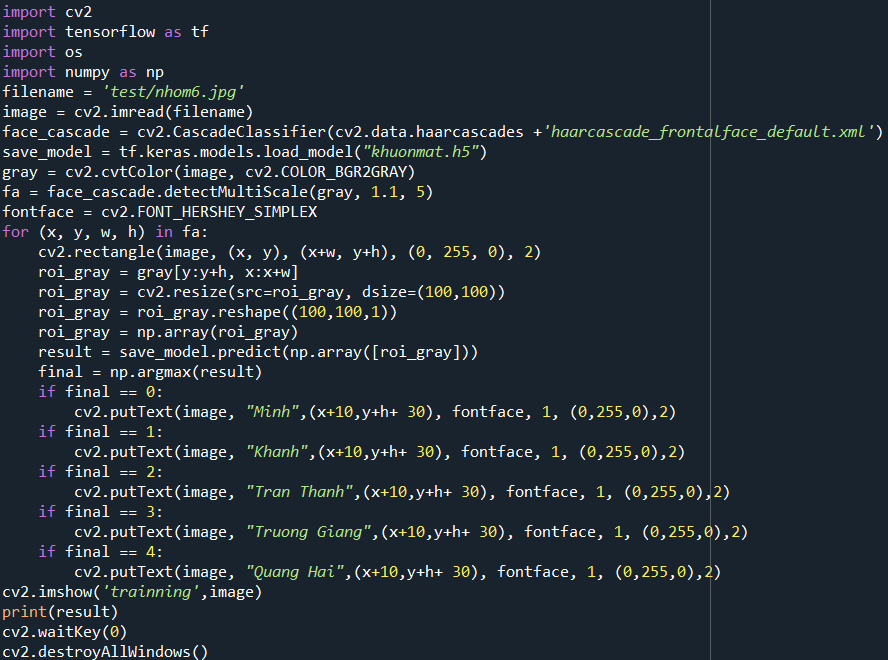


* Để tạo tập dữ liệu huấn luyên từ một ảnh có sẵn, ta nhập id = 1.
* Để tạo tập dữ liệu huấn luyên từ webcam, ta nhập id = 2.

+ Huấn luyện tập dữ liệu:



+ Nhận diện khuôn mặt:



+ Đường link tham khảo code:

[BÁO CÁO CUỐI KÌ - Google Drive](https://drive.google.com/drive/folders/16ttBp5Ecq8WEyxLKdhXvbOOYbW9M81dl?fbclid=IwAR0uo6sgdUVQEEWQLwZvKbW8jzjlh2b1VkyipYSDa1lWZgvIa4oXvFzoj8E)

TÀI LIỆU THAM KHẢO

[1] PGS. TS Trương Ngọc Sơn, *Trí tuệ nhân tạo cơ sở và ứng dụng,* Đại học quốc gia TPHCM, 2020.

[2] Tài liệu online.

[3] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, “Recent advances in convolutional neural network acceleration,” *Neurocomputing*, vol. 323, pp. 0–38, 2019, doi: 10.1016/j.neucom.2018.09.038.