**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO MÔN HỌC**

***Đề tài:***

**XÂY DỰNG MẠNG NEURAL TÍCH CHẬP**

**TRONG BÀI TOÁN NHẬN DIỆN KHUÔN MẶT**

**Môn học: PHÂN TÍCH DỮ LIỆU LỚN**

**Giảng viên hướng dẫn: TS. TRẦN QUANG QUÝ**

**Nhóm : 14**

**Sinh viên thực hiện : Lương Văn Hoàn**

**Bùi Thị Ánh Mai**

**Nguyễn Văn Dương**

**Lớp : CNTTK19M**

**Thái Nguyên - 2024**

# 

# MỤC LỤC

[MỤC LỤC 2](#_Toc10209)

[BẢNG PHÂN CÔNG 3](#_Toc29029)

[GIỚI THIỆU CHUNG 4](#_Toc2755)

[CHƯƠNG I. CƠ SỞ LÝ THUYẾT 5](#_Toc23324)

[1. Giới thiệu về mạng Neural. 5](#_Toc23081)

[1.1. Khái niệm mạng Neural nhân tạo. 5](#_Toc31848)

[1.2. Các thành phần cơ bản của một neural nhân tạo. 5](#_Toc10340)

[1.3. Đặc trưng của mô hình mạng Neural 6](#_Toc22901)

[1.4. Phân loại mạng neural. 6](#_Toc8103)

[1.4.1. Mạng Neural có một đầu vào 6](#_Toc6032)

[1.4.2. Mạng Neural có nhiều đầu vào 8](#_Toc3146)

[Phân loại theo số lớp 8](#_Toc9452)

[Phân loại theo liên kết neural 8](#_Toc4612)

[2. Giới thiệu mạng Neural tích chập (CNN - Convolutional Neural Network) 9](#_Toc4973)

[2.1. Khái niệm mạng Neural tích chập 9](#_Toc11280)

[2.2. Cấu trúc của mạng Neural tích chập 9](#_Toc26304)

[3. Hàm OpenCV 12](#_Toc22359)

[3.1. Hàm OpenCV là gì? 12](#_Toc22608)

[3.2. Cấu trúc của OpenCV 13](#_Toc32536)

[3.3. Phát hiện khuôn mặt với các hàm trong OnpenCV 13](#_Toc26589)

[CHƯƠNG II. XÂY DỰNG MÔ HÌNH NHẬN DIỆN 14](#_Toc18340)

[1. Xây dựng mô hình nhận diện. 14](#_Toc1466)

[2. Các bước thực hiện 15](#_Toc28714)

[2.3. Hiển thị dữ liệu đầu ra 22](#_Toc19448)

[CHƯƠNG III. DEMO MÔ HÌNH 23](#_Toc2978)

[I. Mục đích và chuẩn bị 23](#_Toc9747)

[1. Mục đích 23](#_Toc4701)

[2. Chuẩn bị 23](#_Toc730)

[II. Khởi tạo chương trình 24](#_Toc6338)

[1. Xử lý dữ liệu đầu vào 24](#_Toc19103)

[2. Huấn luyện mô hình 24](#_Toc7483)

[Độ chính và giá trị hàm lỗi theo từng epoch 26](#_Toc25596)

[3. Hiển thị kết quả 27](#_Toc11884)

[4. Đánh giá kết quả 27](#_Toc9135)

[KẾT LUẬN 28](#_Toc29728)

[TÀI LIỆU THAM KHẢO 29](#_Toc10325)

# BẢNG PHÂN CÔNG

| **Stt** | **Mã sinh viên** | **Họ và tên sinh viên** | **Ngày sinh** | **Tên tiểu luận** | **Tên nhiệm vụ** |
| --- | --- | --- | --- | --- | --- |
|  | DTC2054802010030 | Bùi Thị Ánh Mai | 12/08/2002 | Xây dựng mạng Neural tích chập trong bài toán nhận diện khuôn mặt | - Tìm hiểu về mạng Neural nói chung và mạng Neural tích chập (CNN)  - Cách xây dựng mô hình nhận diện khuôn mặt bằng mạng Neural tích chập (CNN) |
|  | DTC20H5202120005 | Lương Văn Hoàn | 24/08/2002 | Xây dựng mạng Neural tích chập trong bài toán nhận diện khuôn mặt | - Tìm hiểu và xây dựng mô hình nhận dạng  - Tìm dữ liệu và gán nhãn  - Demo mô hình |
|  | DTC2054802010005 | Nguyễn Văn Dương | 16/03/2002 | Xây dựng mạng Neural tích chập trong bài toán nhận diện khuôn mặt | - Tổng hợp, chỉnh sửa bản báo cáo: word, power point |

# GIỚI THIỆU CHUNG

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, có rất nhiều 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 bài toán này, chúng em sử dụng mạng neural tích chập (CNN - Convolutional Neural Network) cho bài toán nhận diện khuôn mặt, dưới sự hướng dẫn của TS. Trần Quang Quý.

# CHƯƠNG I. CƠ SỞ LÝ THUYẾT

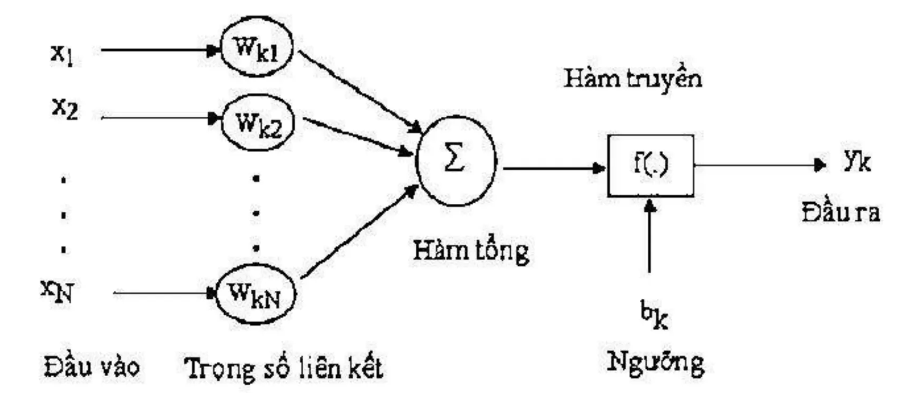
## Giới thiệu về mạng Neural.

## **Khái niệm mạng Neural nhân tạo.**

Mạng neural nhân tạo bao gồm các neural (nút, đơn vị xử lý) được nối với nhau bởi các liên kết. Mỗi liên kết kèm theo một trọng số nào đó, đặc trưng cho đặc tính kích hoạt hay ức chế giữa các neural. Có thể xem các trọng số là phương tiện để lưu trữ thông tin dài hạn trong mạng neural và nhiệm vụ của quá trình huấn luyện mạng là cập nhật các trọng số khi có thêm các thông tin về các mẫu đọc, hay nói một cách khác, các trọng số được điều chỉnh sao cho hình dáng vào của nó mô phỏng hoàn toàn phù hợp với môi trường đang xem xét.

Mạng neural có thể huấn luyện từ dữ liệu mẫu và tổng quát hóa dựa trên các dữ liệu mẫu học. Trong mạng neural, các neural đón nhận tín hiệu vào gọi là neural vào, các neural đưa thông tin ra gọi là neural ra.

* 1. **Các thành phần cơ bản của một neural nhân tạo.**



Các thành phần cơ bản của một mạng neural nhân tạo bao gồm:

+ Tập các đầu vào: là các tín hiệu vào (input signals) của neural, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.

+ Tập các liên kết: Mỗi liên kết được thể hiện bởi một trọng số liên kết (Synaptic Weight). Trọng số liên kết giữa tín hiệu vào thứ j với neural k thường được ký hiệu là Wkj. Thông thường, các trọng số này được tạo một cách ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.

+ Bộ tổng (Summing Function): thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.

+ Ngưỡng (hay còn gọi là độ lệch): thường được đưa vào như một thành phần của hàm truyền.

+ Hàm truyền (Transfer function): hàm này được dùng để giới hạn phạm vi đầu ra của mỗi neural. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng.

+ Đầu ra: Là tín hiệu đầu ra của một neural, với mỗi neural sẽ có tối đa là một đầu ra.

* 1. **Đặc trưng của mô hình mạng Neural**

Những mô hình mạng neural đã trình bày có tiềm năng tạo nên một cuộc

cách mạng trong công nghệ máy tính và các quá trình xử lý thông tin. Những mong muốn và hy vọng đó chủ yếu bắt nguồn từ các đặc trưng chính sau:

- Khả năng của các *quá trình xử lý song song và phân tán*: Có thể đưa vào

mạng một lượng lớn các neural liên kết với nhau theo những lược đồ với các kiến trúc khác nhau.

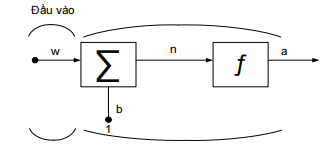
- Khả năng *thích nghi và tự tổ chức*: Về đặc trưng này người ta đề cập đến khả năng xử lý thích nghi và điều chỉnh bền vững dựa vào các thuật toán học thích nghi và các quy tắc tự tổ chức.

- Khả năng *dung thứ lỗi*: Cố gắng bắt chước khả năng dung thứ lỗi của não theo nghĩa hệ thống có thể tiếp tục làm việc và điều chỉnh khi nhận tín hiệu vào một phần thông tin bị sai lệch hoặc bị thiếu.

- *Xử lý các quá trình phi tuyến*: Đặc trưng này rất quan trọng, ví dụ trong

xấp xỉ mạng, miễn nhiễu ( chấp nhận nhiễu) và có khả năng phân lớp.

* 1. **Phân loại mạng neural.**
     1. **Mạng Neural có một đầu vào**

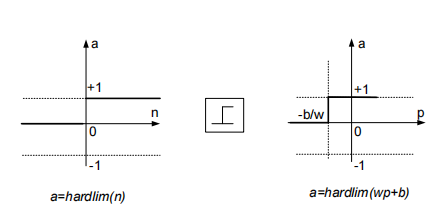


Đầu vào p được nhân với một trọng số w được wp rồi gửi đến bộ tổng, đầu vào khác, 1, qua ngưỡng b và sau đó đưa đến bộ tổng. Tổng các đầu vào n được truyền qua một hàm f nào đó, như vậy đầu ra được xác định:

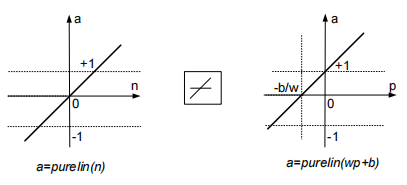
*a =f(wp + b)*

Hàm kích hoạt của từng neural trong mạng đóng vai trò quan trọng trong sự liên kết giữa các neural. Có 3 loại hàm kích hoặt thường được sử dụng:

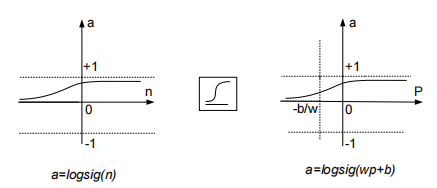
*+ Hàm Hardlimit:*



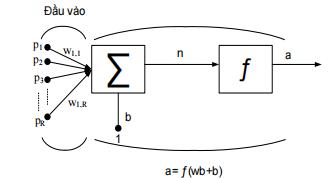
*+ Hàm Linear:*



*+ Hàm Sigmoid:*



* + 1. **Mạng Neural có nhiều đầu vào**



Các đầu vào p1, p2, p3, …, pR, tương ứng với các trọng số w1.1, w1.2,…, w1.R của ma trận trọng số W. Tổng tín hiệu đầu vào n được tính như sau:

n = w1.1p1 + w1.2p2 + … + w1.RpR + b

Công thức trên có thể viết dưới dạng ma trận như sau:

n = Wp + b

Trong đó: W = (w1.1, w1.2, …, w1.R), p = (p1,p2, …, pR)

Cuối cùng đầu ra của mạng sẽ là: a = ƒ(wp + b)

* **Phân loại theo số lớp**

+ Mạng neural một lớp.

+ Mạng neural nhiều lớp.

* **Phân loại theo liên kết neural**

**+** Mạng neural truyền thẳng

+ Mạng neural hồi quy

## Giới thiệu mạng Neural tích chập (CNN - Convolutional Neural Network)

* 1. **Khái niệm mạng Neural tích chập**

Mạng Neural tích chập (CNN - Convolutional Neural Network) 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. 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 sau, như các thuật toán gắn thẻ tự động, tìm kiếm ảnh hoặc gợi ý sản phẩm cho người dùng.

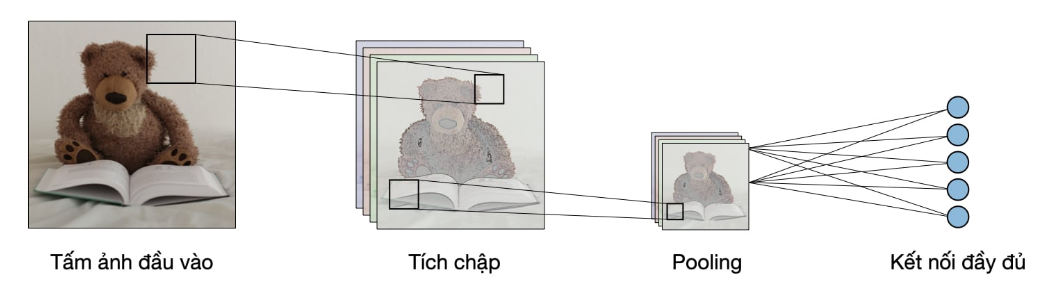
* 1. **Cấu trúc của mạng Neural tích chập**

Mạng Neural tích chập bao gồm:

+ Lớp chập (Convolutional)

+ Lớp tổng hợp (Pooling)

+ Lớp kết nối đầy đủ (Fully - Connected)

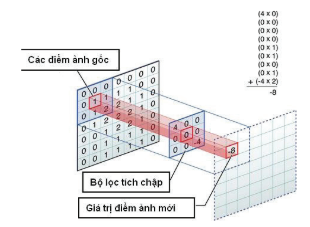


* + 1. ***Lớp tích chập (Convolutional):*** đây là thành phần quan trọng nhất trong mạng CNN, thể hiện sự liên kết cục bộ thay vì kết nối toàn bộ các điểm ảnh. Các liên kết cục bộ được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc filters có kích thước nhỏ.

+ *Convolution Filter* hay còn gọi là Filter/ Kernel là bộ lọc, bộ lọc này sẽ quét lần lượt qua từng vùng của bức ảnh đầu vào, sau đó đưa qua hàm kích hoạt thường là hàm ReLu để thu lại được một ma trận mới gọi là feature map.

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.

*+ Convolution Layer* là một lớp ẩn chứa các feature map và mỗi feature map này sẽ là các đặc trưng của đầu vào.



Hình trên đây thể hiện bộ lọc tích chập sử dụng ma trận điểm ảnh, 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 nhở hơn hoặc bằng 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:

+1

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 (Padding)

- k: Kích thước bộ lọc

- s: bước trượt (bước nhảy) của bộ lọc (Stride)

Lớp tích chập kết hợp với các hàm kích hoạt phi tuyến để tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

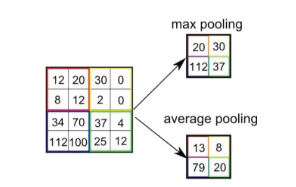
+ Lớp kích hoạt phi tuyến ReLu: Các trọng số của các bộ lọc này được khởi tạo ngẫu nhiên trong lần đầu tiên và sẽ được cập nhật trong quá trình huấn luyện loạt các phép tính toán tuyến tính qua các lớp tích chập. Lớp kích hoạt phi tuyến sử dụng các hàm kích hoạt phi tuyến như ReLU hoặc sigmoid, tanh… 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.

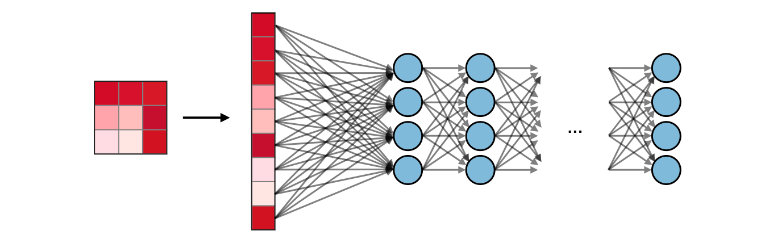
* + 1. ***Lớp tổng hợp (Pooling):*** được đặt sau lớp tích chập và lớp ReLU để làm giảm kích thước ảnh đầu ra trong khi vẫn giữ được các thông tin quan trọng của ảnh đầu vào. Việc giảm kích thước dữ liệu có tác dụng làm giảm được số lượng tham số cũng như tăng hiệu quả tính toán. Lớp lấy mẫu cũng sử 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.

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



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.

* + 1. ***Lớp kết nối đầy đủ (Fully - Connected):*** được thiết kế tương tự như trong mạng Neural truyền thống, tất cả các điểm ảnh được kết nối đầy đủ với node trong lớp tiếp theo.



Tầng kết nối đầy đủ nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNN, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.

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.

## Hàm OpenCV

* 1. **Hàm OpenCV là gì?**

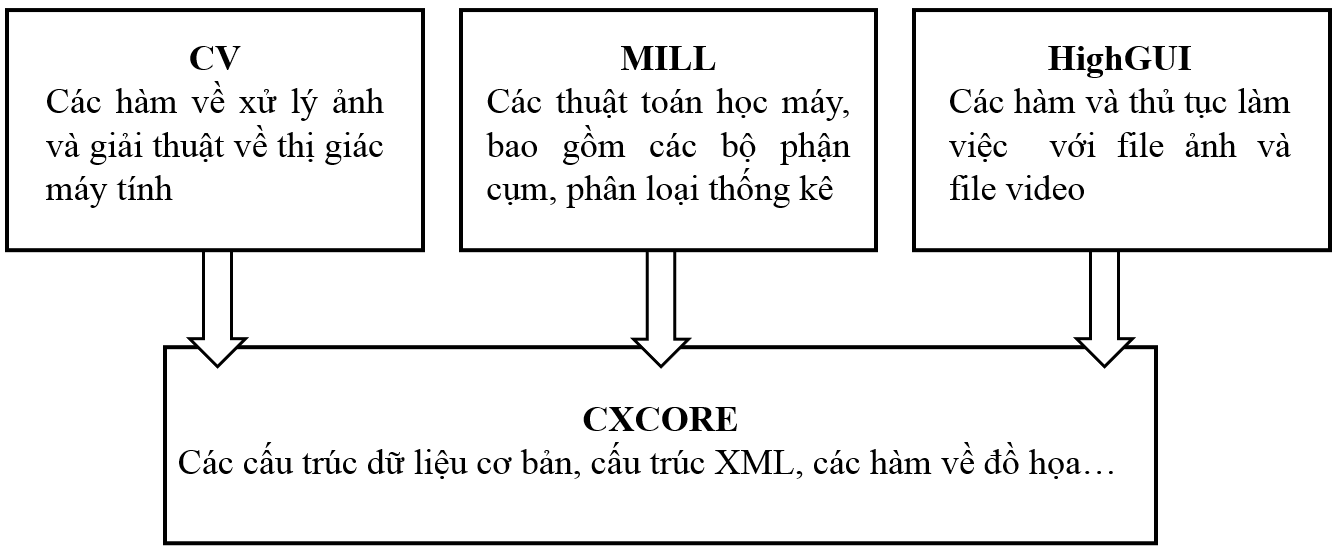
OpenCV viết tắt của từ Open Source Computer Vision Library.

* 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 nhanh hơn.
* 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.

* 1. **Cấu trúc của OpenCV**

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



+ 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.

* 1. **Phát hiện khuôn mặt với các hàm trong OnpenCV**

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

# CHƯƠNG II. XÂY DỰNG MÔ HÌNH NHẬN DIỆN

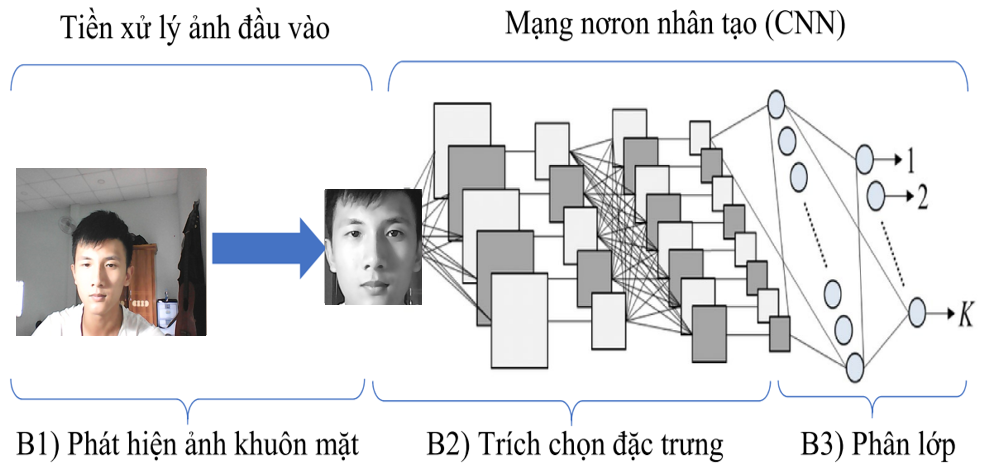
1. **Xây dựng mô hình nhận diện.**

Mô hình nhận diện được chia thành 3 bước chính bao gồm:

*Bước 1:* Nhận diện khuôn mặt và 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.



1. **Các bước thực hiện**
   1. **Nhận diện khuôn mặt và 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.

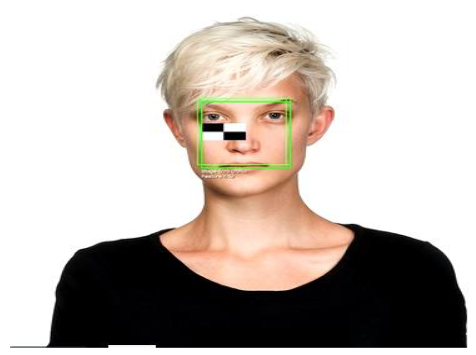


Hình ảnh sau khi được chuyển từ ảnh màu thành ảnh xám

Bước 2: Dò tìm khuôn mặt

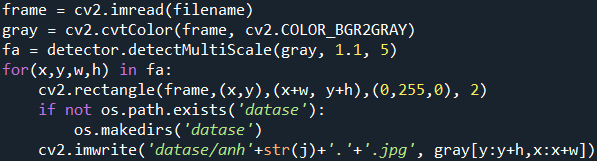
Integal Image là mảng 2 chiều với kích thước ảnh cần tính đặc trưng Haar. Bắt đầu từ vị trí trên bên trái đến vị trí dưới, bên phải của ảnh. Sau khi đã tính được Integral Image, việc tính tổng điểm ảnh của một vùng bất kỳ nào đó trên ảnh thực hiện sẽ đơn giản hơn.

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.



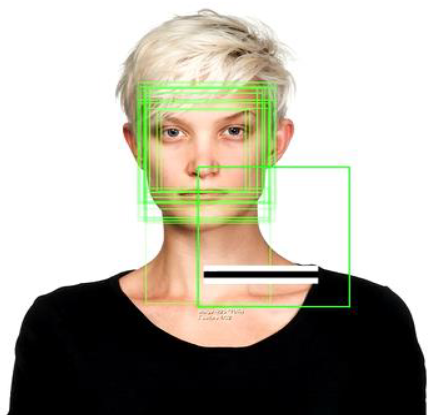
Bước 3: Hậu xử lí

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.

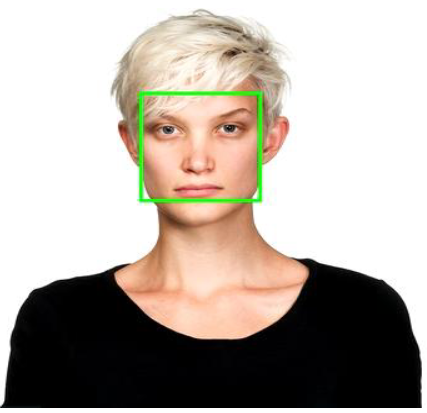


Hàm detectMultiScale là phần tìm kiếm khuôn mặt, hàm (hay phương thức) này thuộc lớp CascadeClassifier (lớp phục vụ tìm kiếm đối tượng của OpenCV)

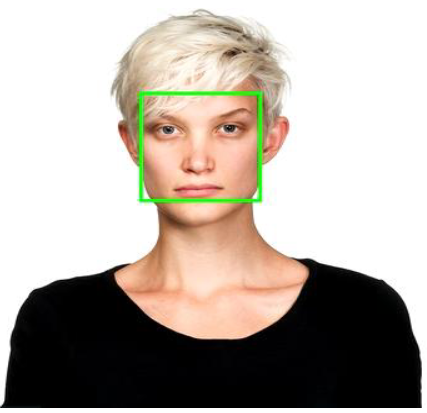
Hàm detectMultiScale sau khi tìm kiếm xong sẽ trả về bộ giá trị gồm tọa độ gốc của khung chứa khuôn mặt x, y; chiều dài, rộng của khung w, h. Các giá trị này nằm trong mảng faces. Cấu trúc for...sẽ duyệt qua toàn bộ các bộ giá trị này,với mỗi bộ giá trị ta dùng hàm rectangle để vẽ một hình chữ nhật lên 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.



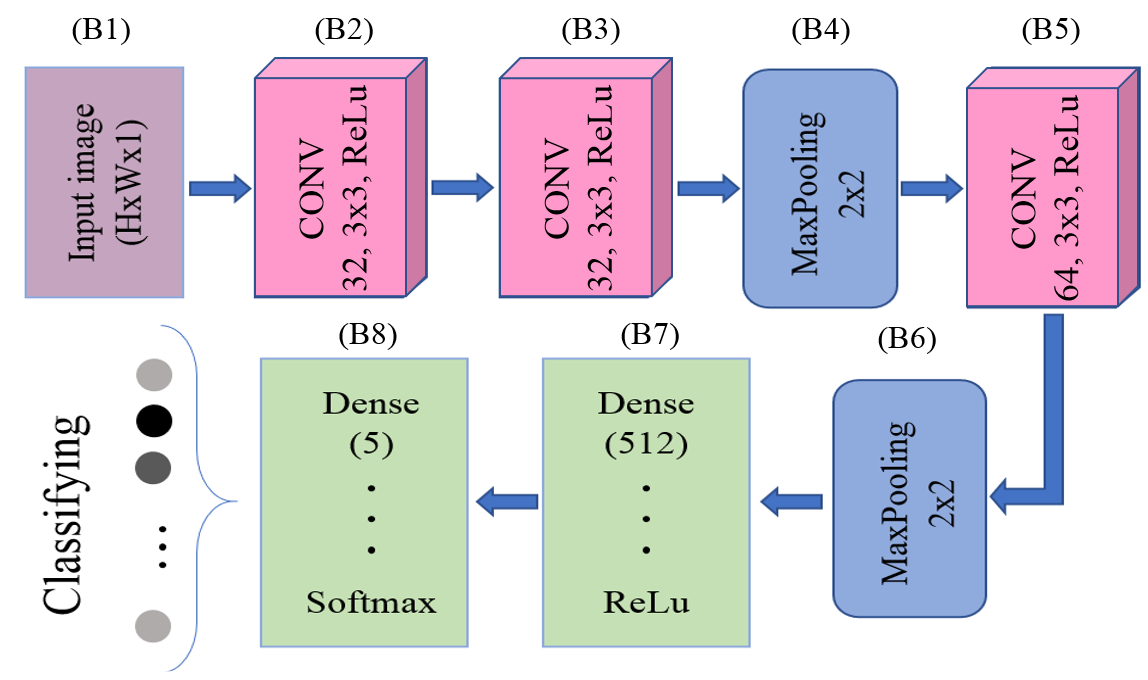
Sau khi đã đánh dấu hết những nơi có thể là khuôn mặt thì sẽ gộp lại những hình chữ nhật chứa cùng một khuôn mặt để chỉ cho ra một hình chữ nhật cho 1 khuôn mặt.



+ Kết quả quá trình tạo lập dữ liệu:

* 1. **Huấn luyện dữ liệu đầu vào và trích chọn đặc trưng**

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 neural và độ lớn (số neural) của mỗi lớp ảnh hưởng đến chất lượng cũng như độ phức tạp trong tính toán của mạng neural. 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. 

Mỗi lớp neural 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 neural 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 neural để xây dựng kiến trúc của CNN bao gồm:

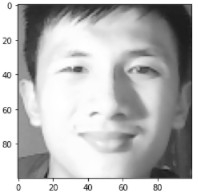
+ 3 lớp tích chập (CONV)

+ 2 lớp neural Max Pooling

+ 2 lớp neural kết nối đầy đủ để phân loại (gọi là lớp Dense).

Mỗi lớp CONV được kết nối theo sau nó bởi một lớp Max Pooling, 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 neural kế tiếp. Theo nguyên tắc xếp chồng các lớp neural 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 neural 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:

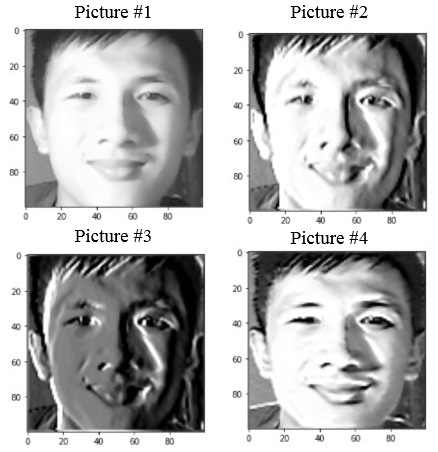


• Khối B2, B3 là lớp neural tích chập có 2 bộ lọc đặc trưng với kích thước cửa sổ hàm nhân là 3×3. Hàm kích hoạt ReLu được sử dụng trong lớp neural này. Hiệu ứng này nhằm cải thiện các đặc trưng thưa của toàn mạng và tránh sự phụ thuộc vào việc truyền tham số giữa các neural.

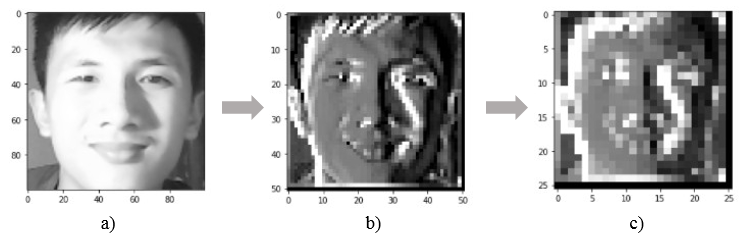
• Khối B4 là lớp Max Pooling, cửa sổ xử lý có kích thước 2×2 được sử dụng. Lớp giảm không gian mẫu (downsampling) này sử dụng phương pháp Max Pooling với việc có thể giữ lại thông tin hữu ích và cắt giảm lượng dữ liệu cần xử lý ở bước tiếp theo.

•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.



Trong hình a/b/c dưới đây minh hoạ kết quả xử lý ở bộ lọc đầu tiên của các khối B3, B5 cùng với lớp neural Max Pooling 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.



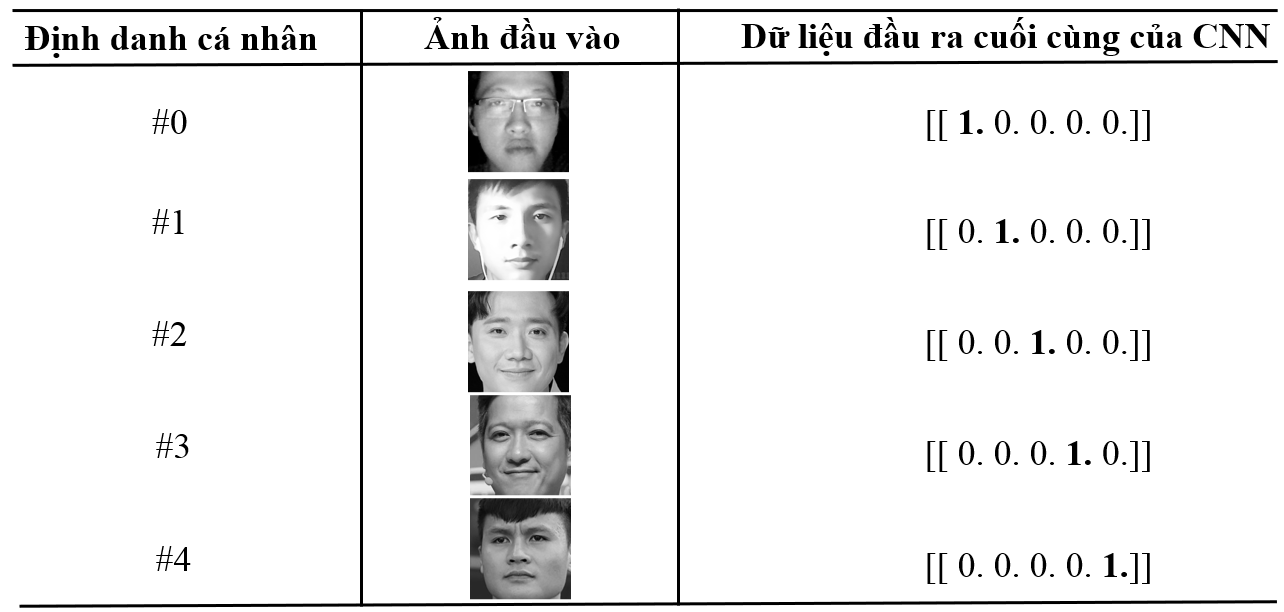
•Khối B7 là lớp neural kết nối đầy đủ (fully connected layers). Lớp này cùng với lớp trong khối B8 nhằm mục tiêu phân lớp các đặc trưng được trích chọn ở các lớp trước, do đó chúng tôi thiết kế số neural đủ lớn. Hàm kích hoạt ReLu cũng được áp dụng. Dữ liệu sau đây là đầu ra của 512 neural đầu tiên trong 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 neural 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 neural tương ứng, với mô hình CNN có lớp (tức là có neural 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 neural thứ ở lớp neural cuối cùng và tương ứng với nó là lớp đối tượng Ck.

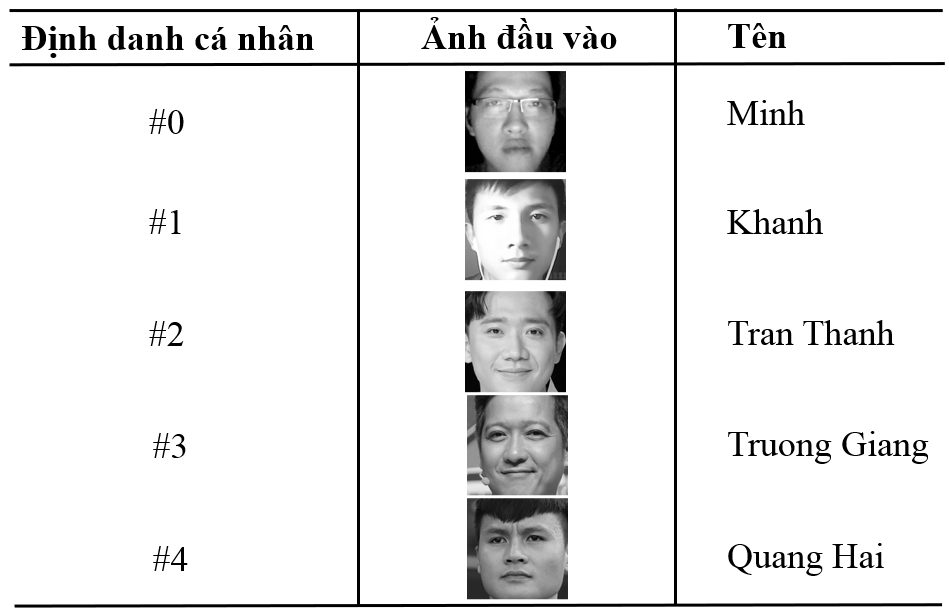
Bảng dưới minh hoạ dữ liệu gồm 5 giá trị sau là kết quả đầu ra của lớp neural cuối cùng (khối B8), tương ứng với kết quả phân lớp của ảnh đầu vào. Kết quả đầu ra của mô hình CNN tương ứng với mỗi ảnh đầu vào gồm 5 giá trị (ở đây chỉ minh hoạ với 5 lớp). Giá trị cao nhất ở vị trí nào (tính theo chỉ số 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.



### Hiển thị dữ liệu đầu ra

Sau khi mô hình được huấn luyện, 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 bên dưới.



# CHƯƠNG III. DEMO MÔ HÌNH

## Mục đích và chuẩn bị

1. **Mục đích**

- Tìm hiểu để hiểu biết hơn về các ứng dụng của Big Data nói chung và việc nhận diện khuôn mặt nói chung.

- Tạo và phát triển một chương trình nhận diện khuôn mặt có thể ứng dụng trong thực tế, ứng dụng trong một số lĩnh vực: Giáo dục (điểm danh, thi cử,…), an ninh (giám sát, định danh,…).

1. **Chuẩn bị**

- Visual studio code để chạy chương trình

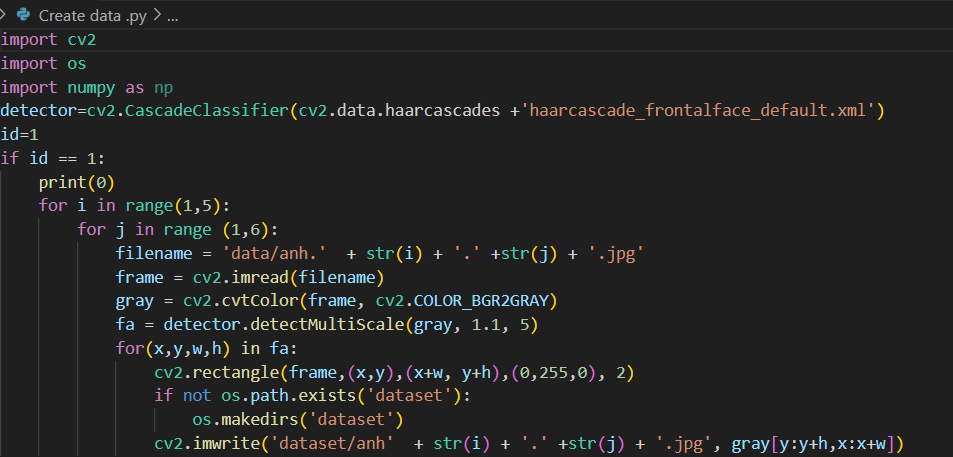
- Cài đặt các thư viện cần thiết như: OpenCV, cv2 , numpy ,os, Tensflow, haarcascade\_frontalface\_default.xmt, PIL.

- Tập dữ liệu có tên là data, các dữ liệu ảnh được sưu tầm từ google

## Khởi tạo chương trình

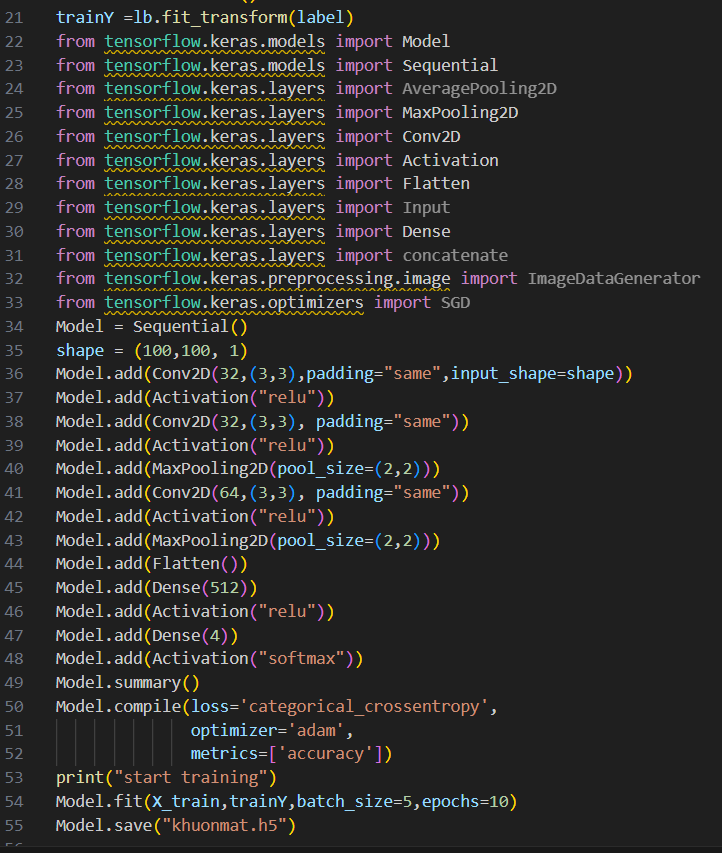
1. **Xử lý dữ liệu đầu vào**

Mở visual studio code khởi chạy chương trình có tên ***creat\_data.py*** lấy các dữ liệu từ file data chuẩn bị sẵn, để đưa từ ảnh màu về ảnh xám, giúp dễ dàng nhận diện các đặc trưng mà không làm thay đổi về kết quả đầu ra. Sau đó tạo một ô vuông cắt khuôn mặt và lưu vào file ***dataset***, nếu chưa có thì sử dụng lệnh ***os.makedirs(‘dataset’)*** để ghi ảnh sau khi tiền xử lý dữ liệu.

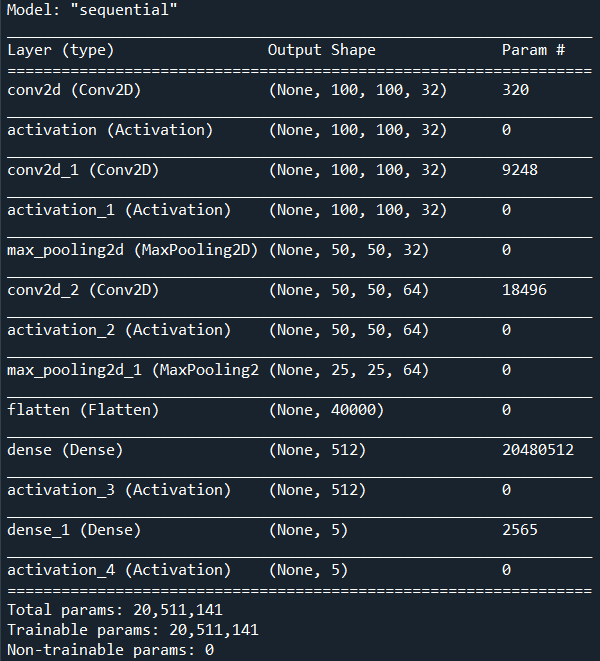


1. **Huấn luyện mô hình**

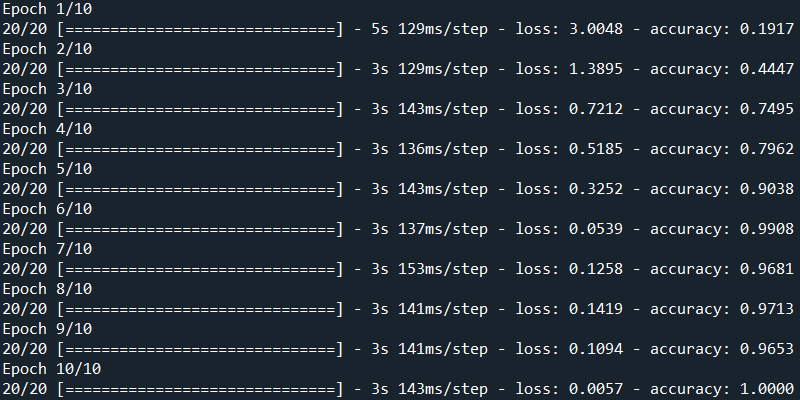
Sau khi xử lý dữ liệu, chúng ta cần huấn luyện và gán nhãn cho từng đối tượng bằng cách chạy chương trình có tên ***train.py*** bằng mô hình CNN và lưu module vào file *khuonmat.h5*



Quá trình huấn luyện tập dữ liệu:



Hơn 20 triệu trọng số được huấn luyện



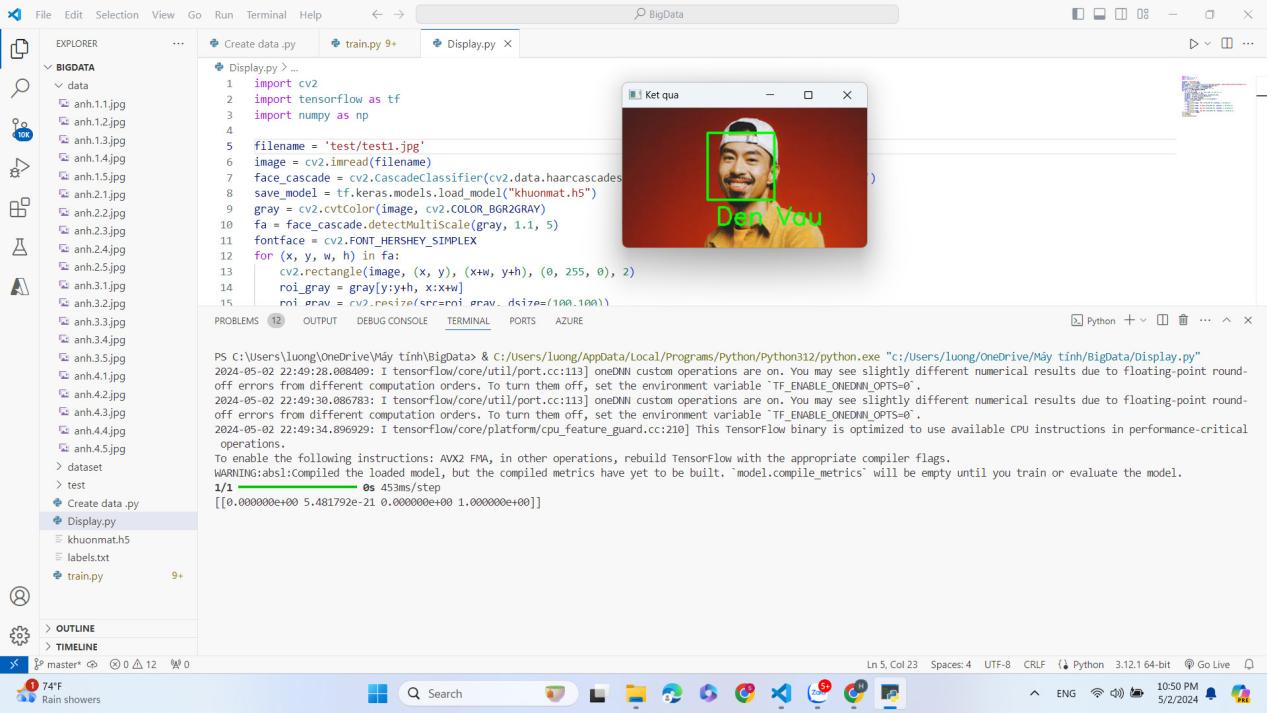
Độ chính và giá trị hàm lỗi theo từng epoch

\*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.

1. **Hiển thị kết quả**

****

1. **Đánh giá kết quả**

Kết quả cho ra là chính xác đồng thời phát hiện được khuôn mặt cần nhận diện

# 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.

# TÀI LIỆU THAM KHẢO

1. Giáo trình môn Mạng Neural và ứng dụng
2. <https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-convolutional-neural-networks>
3. <https://tailieu.vn/doc/nhan-dang-khuon-mat-trong-video-bang-mang-no-ron-tich-chap-2224637.html>
4. Tham khảo code: <https://youtu.be/Zz5O5AIN9qs?si=S7svu4Pg2q84GvYE>