

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | **học máy** | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | **BÁO CÁO BÀI TẬP LỚN**  **THUẬT TOÁN**  **NHẬN DẠNG KHUÔN MẶT** |  | | | |
|  | | | | Nhóm:  Lê Mạnh Đức  Vũ Nguyễn Đức Khôi  Nguyễn Trường Minh  —  Học Máy  —  Người hướng dẫn:  Nguyễn Văn Hậu  Hạ Long 2020 |  | | | |
|  | | |  | | |  | | |

# MỤC LỤC

[**MỤC LỤC** 1](#_Toc55341331)

[**LỜI CAM ĐOAN** 2](#_Toc55341332)

[**CHƯƠNG 1: Tổng quan về máy học và bài toán phát hiện khuôn mặt** 3](#_Toc55341333)

[1. Trình bày tổng quan máy học 3](#_Toc55341334)

[1.1. Tổng quan 3](#_Toc55341335)

[1.2. Các ứng dụng của học máy 4](#_Toc55341336)

[2. Bài toán nhận dạng khuôn mặt 4](#_Toc55341337)

[2.1. Tổng quan về các ứng dụng liên quan tới khuôn mặt 4](#_Toc55341338)

[2.2. Các hướng tiếp cận của bài toán 4](#_Toc55341339)

[3. Sử dụng phương pháp CNN trong phát hiện khuôn mặt 5](#_Toc55341340)

[**CHƯƠNG 2: Phương pháp CNN** 6](#_Toc55341341)

[1. Convolutional 6](#_Toc55341342)

[2. Cấu trúc mạng CNN 6](#_Toc55341343)

[2.1. Trường tiếp nhận cục bộ 7](#_Toc55341344)

[2.2. Trọng số chia sẻ 9](#_Toc55341345)

[2.3. Lớp tổng hợp 9](#_Toc55341346)

[2.4. Cách chọn tham số cho CNN 10](#_Toc55341347)

[**CHƯƠNG 3: Cài đặt Thuật Toán phát hiện khuôn mặt** 11](#_Toc55341348)

[1. Trình bày dữ liệu bài toán phát hiện khuôn mặt 11](#_Toc55341349)

[2. Hiển thị dữ liệu 11](#_Toc55341350)

[2.1. Cài đặt thư viện cv2 11](#_Toc55341351)

[2.2. Cài đặt thư viện face\_recognition 13](#_Toc55341352)

[3. Cài đặt mô hình CNN phát hiện khuôn mặt 14](#_Toc55341353)

[4. Cài đặt mô hình HOG phát hiện khuôn mặt 16](#_Toc55341354)

[5. Đánh giá độ chính xác của mô hình 17](#_Toc55341355)

[**KẾT LUẬN** 18](#_Toc55341356)

**LỜI CAM ĐOAN**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi, các số liệu và kết quả nghiên cứu nêu trong khóa luận là trung thực, chưa từng được công bố trong bất kỳ một công trình nào khác. Tôi xin chịu mọi trách nhiệm về những nội dung trình bày trong khóa luận này.

|  |  |
| --- | --- |
|  | **Nhóm tác giả báo cáo**  (kí tên thay mặt)  Lê Mạnh Đức  Vũ Nguyễn Đức Khôi  Nguyễn Trường Minh |

# **CHƯƠNG 1: Tổng quan về máy học và bài toán phát hiện khuôn mặt**

## Trình bày tổng quan máy học

### Tổng quan

Học máy (Machine Learning) là một ngành khoa học nghiên cứu các thuật toán cho phép máy tính có thể học được các khái niệm (concept).

Phân loại: Có hai loại phương pháp học máy chính

* Phương pháp quy nạp: Máy học/phân biệt các khái niệm dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.
* Phương pháp suy diễn: Máy học/phân biệt các khái niệm dựa vào các luật. Phương pháp này cho phép tận dụng được các kiến thức chuyên ngành để hỗ trợ máy tính.

Hiện nay, các thuật toán đều cố gắng tận dụng được ưu điểm của hai phương pháp này.

Các ngành khoa học liên quan:

* Lý thuyết thống kê: các kết quả trong xác suất thống kê là tiền đề cho rất nhiều phương pháp học máy. Đặc biệt, lý thuyết thống kê cho phép ước lượng sai số của các phương pháp học máy.
* Các phương pháp tính: các thuật toán học máy thường sử dụng các tính toán số thực/số nguyên trên dữ liệu rất lớn. Trong đó, các bài toán như: tối ưu có/không ràng buộc, giải phương trình tuyến tính v.v… được sử dụng rất phổ biến.
* Khoa học máy tính: là cơ sở để thiết kế các thuật toán, đồng thời đánh giá thời gian chạy, bộ nhớ của các thuật toán học máy.

Các nhóm giải thuật học máy:

* Học có giám sát: Máy tính được xem một số mẫu gồm đầu vào (input) và đầu ra (output) tương ứng trước. Sau khi học xong các mẫu này, máy tính quan sát một đầu vào mới và cho ra kết quả.
* Học không giám sát: Máy tính chỉ được xem các mẫu không có đầu ra, sau đó máy tính phải tự tìm cách phân loại các mẫu này và các mẫu mới.
* Học nửa giám sát: Một dạng lai giữa hai nhóm giải thuật trên.
* Học tăng cường: Máy tính đưa ra quyết định hành động (action) và nhận kết quả phản hồi (response/reward) từ môi trường (environment). Sau đó máy tính tìm cách chỉnh sửa cách ra quyết định hành động của mình.

### Các ứng dụng của học máy

Ứng dụng: Học máy có ứng dụng rộng khắp trong các ngành khoa học/sản xuất, đặc biệt những ngành cần phân tích khối lượng dữ liệu khổng lồ. Một số ứng dụng thường thấy

* Xử lý ngôn ngữ tự nhiên (Natural Language Processing): xử lý văn bản, giao tiếp người – máy, …
* Nhận dạng (Pattern Recognition): nhận dạng tiếng nói, chữ viết tay, vân tay, thị giác máy (Computer Vision) …
* Tìm kiếm (Search Engine) Chẩn đoán trong y tế: phân tích ảnh X-quang, các hệ chuyên gia chẩn đoán tự động.
* Tin sinh học: phân loại chuỗi gene, quá trình hình thành gene/protein
* Vật lý: phân tích ảnh thiên văn, tác động giữa các hạt …
* Phát hiện gian lận tài chính (financial fraud): gian lận thẻ tỉn dụng
* Phân tích thị trường chứng khoán (stock market analysis)
* Chơi trò chơi: tự động chơi cờ, hành động của các nhân vật ảo
* Robot: là tổng hợp của rất nhiều ngành khoa học, trong đó học máy tạo nên hệ thần kinh/bộ não của người máy.

## Bài toán nhận dạng khuôn mặt

Bài toán phát hiện khuôn mặt của nhóm sẽ tập trung vào việc phát hiện khuôn mặt, sau đó mới phát triển thêm nhận dạng khuôn mặt của ai và nhận diện cảm xúc, ...

### Tổng quan về các ứng dụng liên quan tới khuôn mặt

Từ những năm 2000 trở lại đây, chúng ta đã chứng kiến sự phát triển như vũ bão của các ngành công nghiệp, đặc biệc là ngành công nghiệp chế tạo điện tử. Các ứng dụng nhận diện khuôn mặt đã tràn ngập trên thế giới, từ những ứng dụng trong điện thoại, đến camera giám sát, robot phục vụ, … Hiện nay chi phí của các ứng dụng này đã giảm đáng kể, hỗ trợ cho nhiều ứng dụng khác xung quanh thế giới. Các hướng nghiên cứu về thị giác đang phát triển rất mạnh, bổ trợ cho các ứng dụng quan trọng khác như giáo tiếp giữa người – máy, trong đó hệ thống nhận dạng khuôn mặt đóng một vai trò không nhỏ. Dưới đây chúng tôi liệt kê một số ứng dụng khác của nhận diện khuôn mặt:

* Các ứng dụng chuyên biệt trong ngành hàng không
* Ứng dụng sử dụng trong nhà thông minh, …
* Ứng dụng trên các camera giám sát, …

### Các hướng tiếp cận của bài toán

Trong đề tài này chúng tôi sử thuật toán pháp phát hiện khuôn mặt dựa trên phương pháp là CNN – Convolutional Neural Network.

CNN - Convolutional Neural Network (Mạng nơron tích chập)

Ngoài ra nhóm còn cài đặt thêm thuật toán phát hiện khuôn mặt dựa trên mô hình default HOG-base.

HOG – Histogram Oriented Gradient (Biểu đồ định hướng Gradient)

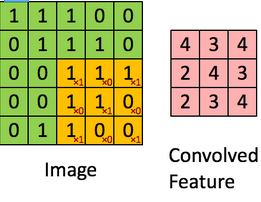
## Sử dụng phương pháp CNN trong phát hiện khuôn mặt

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. Sự ra đời của mạng CNN là dựa trên ý tưởng cải tiến cách thức các mạng nơ ron nhân tạo truyền thống học thông tin trong ảnh. Do sử dụng các liên kết đầy đủ giữa các điểm ảnh vào node, các mạng nơ ron nhân tạo truyền thẳng (Feedforward Neural Network) bị hạn chế rất nhiều bởi kích thước của ảnh, ảnh càng lớn thì số lượng liên kết càng tăng nhanh, kéo theo sự bùng nổ khối lượng tính toán. Ngoài ra, sự liên kết đầy đủ này cũng là sự dư thừa với mỗi bức ảnh, các thông tin chủ yếu thể hiện qua sự phụ thuộc giữa các điểm ảnh với những điểm xung quanh nó mà không quan tâm nhiều đến các điểm ảnh ở cách xa nhau. 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.

# **CHƯƠNG 2: Phương pháp CNN**

Convolutional Neural Network (CNN – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay như hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh.

## Convolutional

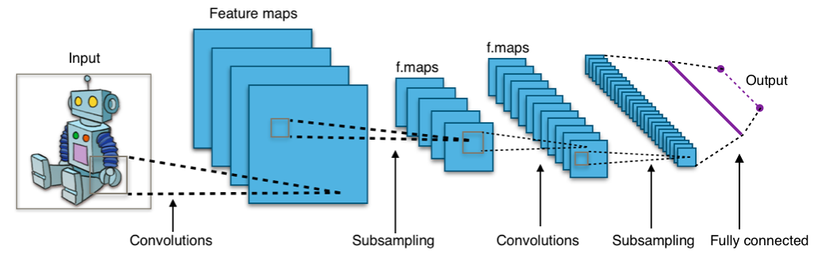
Là một cửa sổ trượt (Sliding Windows) trên một ma trận như mô tả hình dưới:

Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature. Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước 5x5 và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột. Convolution hay tích chập là nhân từng phần tử trong ma trận 3. Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có kích thước nhỏ như trong ví dụ trên là 3x3. Convolution hay tích chập là nhân từng phần tử bên trong ma trận 3x3 với ma trận bên trái. Kết quả được một ma trận gọi là Convoled feature được sinh ra từ việc nhận ma trận Filter với ma trận ảnh 5x5 bên trái.

## Cấu trúc mạng CNN

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo. Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.

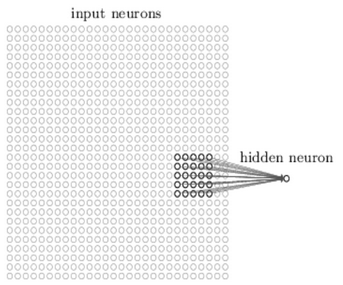


Trong mô hình CNN có 2 khía cạnh cần quan tâm là **tính bất biến** (Location Invariance) và **tính kết hợp** (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter. Đó là lý do tại sao CNN cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

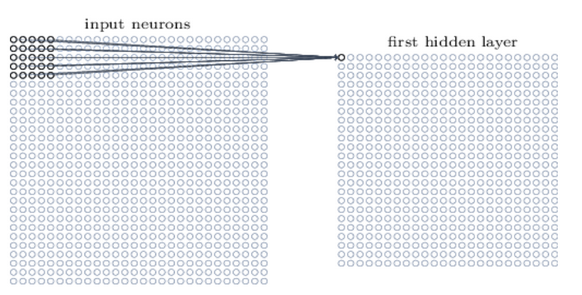
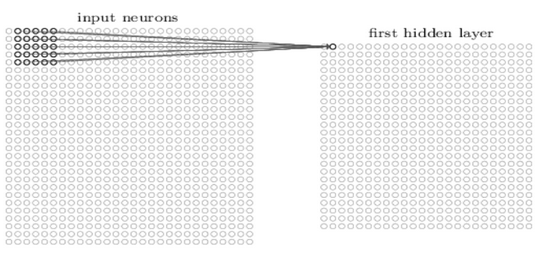
Mạng CNN sử dụng 3 ý tưởng cơ bản:

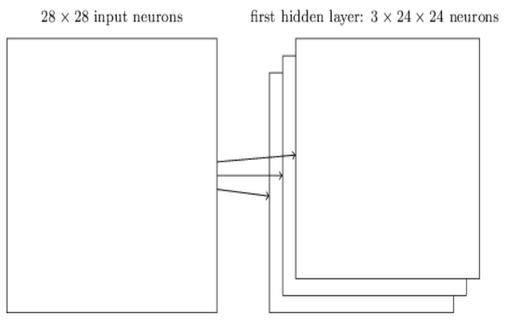
* các trường tiếp nhận cục bộ (local receptive field)
* trọng số chia sẻ (shared weights)
* tổng hợp (pooling).

### Trường tiếp nhận cục bộ

Đầu vào của mạng CNN là một ảnh. Ví dụ như ảnh có kích thước 28x28 thì tương ứng đầu vào là một ma trận có 28x28 và giá trị mỗi điểm ảnh là một ô trong ma trận. Trong mô hình mạng ANN truyền thống thì chúng ta sẽ kết nối các neuron đầu vào vào tầng ảnh. Tuy nhiên trong CNN chúng ta không làm như vậy mà chúng ta chỉ kết nối trong một vùng nhỏ của các neuron đầu vào như một filter có kích thước 5x5 tương ứng (28- 5 + 1) 24 điểm ảnh đầu vào. Mỗi một kết nối sẽ học một trọng số và mỗi neuron ẩn sẽ học một bias. Mỗi một vùng 5x5 đấy gọi là một trường tiếp nhận cục bộ.

Một cách tổng quan, ta có thể tóm tắt các bước tạo ra 1 hidden layer bằng các cách sau:

* + Tạo ra neuron ẩn đầu tiên trong lớp ẩn 1
  + Dịch filter qua bên phải một cột sẽ tạo được neuron ẩn thứ 2.

Đối với bài toán nhận dạng ảnh người ta thường gọi ma trận lớp đầu vào là feature map, trọng số xác định các đặc trương là shared weight và độ lệch xác định một feature map là shared bias. Như vậy đơn giản nhất là qua các bước trên chúng ta chỉ có 1 feature map. Tuy nhiên trong nhận dạng ảnh chúng ta cần nhiều hơn một feature map. Như vậy, local receptive field thích hợp cho việc phân tách dữ liệu ảnh, giúp chọn ra những vùng ảnh có giá trị nhất cho việc đánh giá phân lớp.

### Trọng số chia sẻ

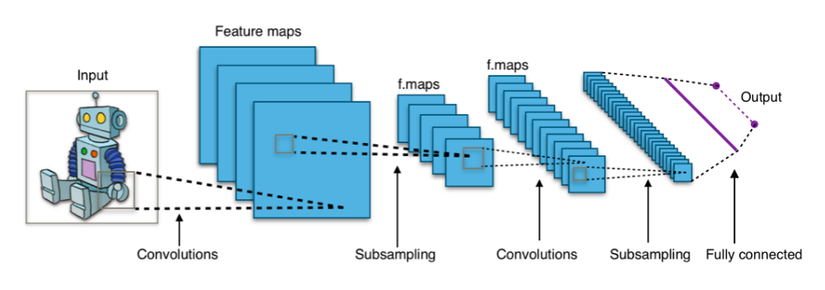
Đầu tiên, các trọng số cho mỗi filter (kernel) phải giống nhau. Tất cả các nơ-ron trong lớp ẩn đầu sẽ phát hiện chính xác feature tương tự chỉ ở các vị trí khác nhau trong hình ảnh đầu vào. Chúng ta gọi việc map từ input layer sang hidden layer là một feature map. Vậy mối quan hệ giữa số lượng Feature map với số lượng tham số là gì?

* Chúng ta thấy mỗi fearture map cần 25 = 5x5 shared weight và 1 shared bias. Như vậy mỗi feature map cần 5x5+1 = 26 tham số. Như vậy nếu có 10 feature map thì có 10x26 = 260 tham số. Chúng ta xét lại nếu layer đầu tiên có kết nối đầy đủ nghĩa là chúng ta có 28x28=784 neuron đầu vào như vậy ta chỉ có 30 neuron ẩn. Như vậy ta cần 28x28x30 shared weight và 30 shared bias. Tổng số tham số là 28x28x30+30 tham số lớn hơn nhiều so với CNN. Ví dụ vừa rồi chỉ mô tả để thấy được sự ước lượng số lượng tham số chứ chúng ta không so sánh được trực tiếp vì 2 mô hình khác nhau. Nhưng điều chắc chắn là nếu mô hình có số lượng tham số ít hơn thì nó sẽ chạy nhanh hơn.

Tóm lại, một convolutional layer bao gồm các feature map khác nhau. Mỗi một feature map giúp detect một vài feature trong bức ảnh. Lợi ích lớn nhất của trọng số chia sẻ là giảm tối đa số lượng tham số trong mạng CNN.

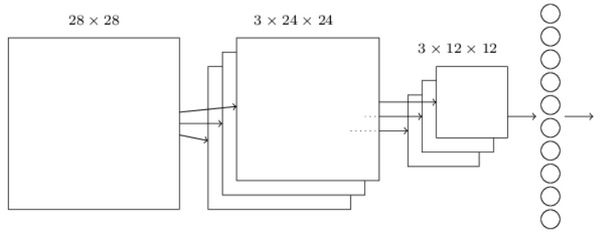
### Lớp tổng hợp

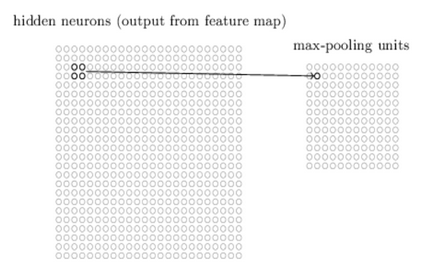
Lớp pooling thường được sử dụng ngay sau lớp convulational để đơn giản hóa thông tin đầu ra để giảm bớt số lượng neuron.



Thủ tục pooling phổ biến là max-pooling, thủ tục này chọn giá trị lớn nhất trong vùng đầu vào 2x2. Như vậy qua lớp Max Pooling thì số lượng neuron giảm đi phân nửa. Trong một mạng CNN có nhiều Feature Map nên mỗi Feature Map chúng ta sẽ cho mỗi Max Pooling khác nhau. Chúng ta có thể thấy rằng Max Pooling là cách hỏi xem trong các đặc trưng này thì đặc trưng nào là đặc trưng nhất. Ngoài Max Pooling còn có L2 Pooling.

Cuối cùng ta đặt tất cả các lớp lại với nhau thành một CNN với đầu ra gồm các neuron với số lượng tùy bài toán.



Hai lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối (fully connected layer). Lớp này nối mọi nơron từ lớp max pooled tới mọi nơron của tầng ra.

### Cách chọn tham số cho CNN

* Số các convolution layer: càng nhiều các convolution layer thì performance càng được cải thiện. Sau khoảng 3 hoặc 4 layer, các tác động được giảm một cách đáng kể
* Kích thước bộ lọc: bộ lọc theo kích thước 5x5 hoặc 3x3
* Pooling size: thường là 2x2 hoặc 4x4 cho ảnh đầu vào lớn
* Cách cuối cùng là thực hiện nhiều lần việc train test để chọn ra được param tốt nhất.

# **CHƯƠNG 3: Cài đặt Thuật Toán phát hiện khuôn mặt**

## Trình bày dữ liệu bài toán phát hiện khuôn mặt

Trong tự nhiên, mặt người thường có đặc điểm giống nhau như 2 mắt, 2 tai, 1 mũi, 1 miệng, … nên về việc phát hiện mặt người sẽ không cần dữ liệu cụ thể. Máy có thể dựa vào các đặc điểm giống nhau như trên để nhận diện đâu là mặt người và đâu không phải.

Khác với bài toán nhận diện khuôn mặt, khi mà sắc thái, biểu cảm, hình dáng của khuôn mặt mỗi người đều khác nhau. Do vậy, để cần nhận diện một khuôn mặt của một người, cần xác định được từng đặc điểm nhận dạng của người đó, nên cần một bộ dữ liệu đầy đủ. Dữ liệu đầy đủ để nhận diện được khuôn mặt sẽ là một bộ sưu tập từ 5 đến 10 bức ảnh của một người để có thể nhận diện khuôn mặt ở độ chính xác tầm 50-60%. Một điểm chính xác khá khiêm tốn.

Quay lại phát hiện khuôn mặt, bài toán của nhóm sẽ không cần dữ liệu của từng mặt vì mặt người cơ bản đều có đặc điểm như vậy. Có thể dựa vào đó để nhận dạng mà không cần tập học.

## Hiển thị dữ liệu

### Cài đặt thư viện cv2

Để hiển thị dữ liệu của bài toán, nhóm sẽ sử dụng thư viện cv2 để hiển thị được ảnh cần để phát hiện khuôn mặt. Cũng trong thư viện cv2 có phương thức rectangle có thể được sử dụng để vẽ hình vuông cho các khuôn mặt phát hiện được. Dưới đây là phần cài đặt hiển thị dữ liệu:

Cài đặt:

import cv2

from google.colab.patches import cv2\_imshow # cv2\_imshow phải import nếu thuật toán cài đặt trên google colab

imagePath = "/content/drive/My Drive/MachineLearningData/gocnghieng4.jpg"

image = cv2.imread(imagePath)

cv2.rectangle(image, (250, 250), (400, 400), (255, 255, 0), 2)

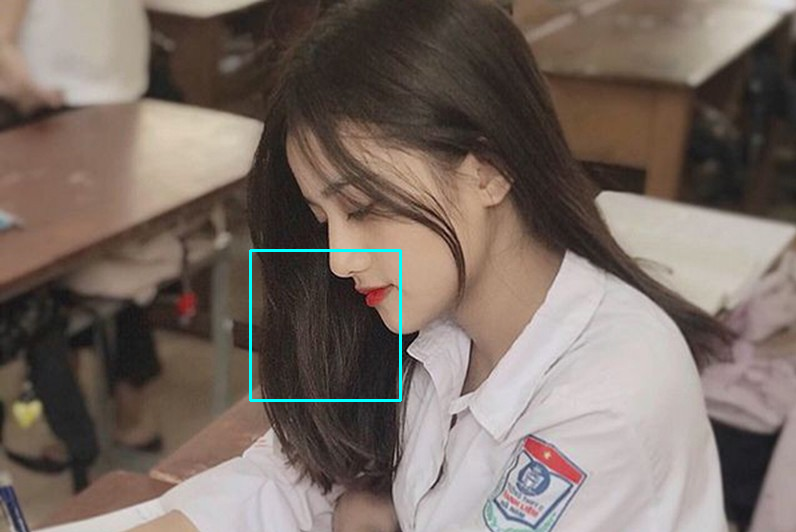
cv2\_imshow(image) # Nếu cài đặt trên môi trường máy tính thì sẽ là

cv2.imshow(img\_face\_draw)

Giải thích:

1. import thư viện cv2 vào code python.
2. Nếu cài đặt thuật toán trên google colab mới cần dòng này, còn nếu cài đặt trên môi trường python trên máy tính thì không cần.
3. Lấy đường dẫn của bức ảnh cần phát hiện khuôn mặt (input).
4. Dùng thư viện cv2 để đọc bức ảnh từ đường dẫn.
5. Phương thức rectangle của cv2 dùng để vẽ hình vuông trên bức ảnh. Lần lượt các thuộc tính là bức ảnh được vẽ lên, điểm bắt đầu của hình vuông (điểm phía trên bên trái), điểm kết thúc của hình vuông (điểm phía dưới bên phải), màu của hình vuông (RGB, CMYK, …), độ dày của hình vuông,…
6. Hiển thị bức ảnh đã được vẽ lên(output).

Kết quả:

****

### Cài đặt thư viện face\_recognition

Xác định được cốt lõi của bài toán là thuật toán phát hiện vị trí của khuôn mặt trên bức ảnh, nhóm sử dụng thư viện của cộng đồng python là face\_recognition. Đây là thư viện được phát triển bởi [Adam Geitgey](https://github.com/ageitgey) - là thư viện phát hiện và nhận diện khuôn mặt đơn giản nhất thế giới. Được xây dựng bằng cách sử dụng tính năng nhận dạng khuôn mặt hiện đại nhất (state-of-the-art) của [dlib](http://dlib.net/) xây dựng với học sâu (deep learning). Mô hình có độ chính xác 99.38% trên thang điểm chuẩn của [Labeled Faces in the Wild](http://vis-www.cs.umass.edu/lfw/).

Cài đặt thuật toán phát hiện khuôn mặt dựa trên thư viện face\_recognition:

import face\_recognition

import cv2

imagePath = "/content/drive/My Drive/MachineLearningData/gocnghieng4.jpg"

image = cv2.imread(imagePath)

face\_locations = face\_recognition.face\_locations(image, model="cnn")

print('Tìm đươc {} khuôn mặt trong bức ảnh.'.format(len(face\_locations)))

for face\_location in face\_locations:

    # In ra vị trí của từng mặt trong ảnh

    top, right, bottom, left = face\_location

    print("Mặt được xác định ở vị trí pixel như sau: Trên: {}, Trái: {}, Dưới: {}, Phải: {}".format(top, left, bottom, right))

Giải thích:

1. import thư viện face\_recognition.
2. import thư viện cv2.
3. Lấy đường dẫn của bức ảnh cần phát hiện khuôn mặt.
4. Đọc bức ảnh từ đường dẫn.
5. Lấy vị trí của các khuôn mặt nhờ phương thức face\_locations của thư viện face\_recognition. Phương thức này sẽ trả về một mảng vị trí của các khuôn mặt. Vị trí của từng khuôn mặt có định dạng [top, right, bottom, left]. Phương thức face\_locations nhận các tham số như ảnh truyền vào (input), mô hình được áp dụng (mô hình đang áp dụng là CNN, mặc định là HOG).
6. Hiển thị số khuôn mặt đã được phát hiện trong input.
7. Lặp qua vị trí của từng khuôn mặt trong mảng.
8. Gán tham số.
9. Hiển thị vị trí của từng khuôn mặt đã được phát hiện trong ảnh input.

Kết quả:

Tìm đươc 1 khuôn mặt trong bức ảnh.

Mặt được xác định ở vị trí pixel như sau Top:145, Left:284, Bottom:315, Right:454

## Cài đặt mô hình CNN phát hiện khuôn mặt

Kết hợp hai lần cài đặt trong phần hiển thị dữ liệu với nhau:

* Thêm phương thức rectangle vào trong vòng lặp để vẽ hình vuông vào các khuôn mặt đã phát hiện được

cv2.rectangle(image, (left, top), (right, bottom), (255,255,0), 2)

* Thêm phương thức hiển thị ảnh vào cuối để hiển thị kết quả.

cv2\_imshow(image)

# cv2.imshow(image) # nếu cài đặt trên môi trường máy tính

→ Ta đã hoàn thành phần cài đặt thuật toán phát hiện khuôn mặt dựa trên mô hình CNN.

Cài đặt thuật toán:

import cv2

from google.colab.patches import cv2\_imshow

import time

import face\_recognition

imagePath="/content/drive/My Drive/MachineLearningData/gocnghieng4.jpg"

image = cv2.imread(imagePath)

start\_time = time.time()

face\_locations = face\_recognition.face\_locations(image, model='cnn')

end\_time = time.time()

print("Tìm được {} mặt người trong bức ảnh.".format(len(face\_locations)))

for face\_location in face\_locations:

    top, right, bottom, left = face\_location

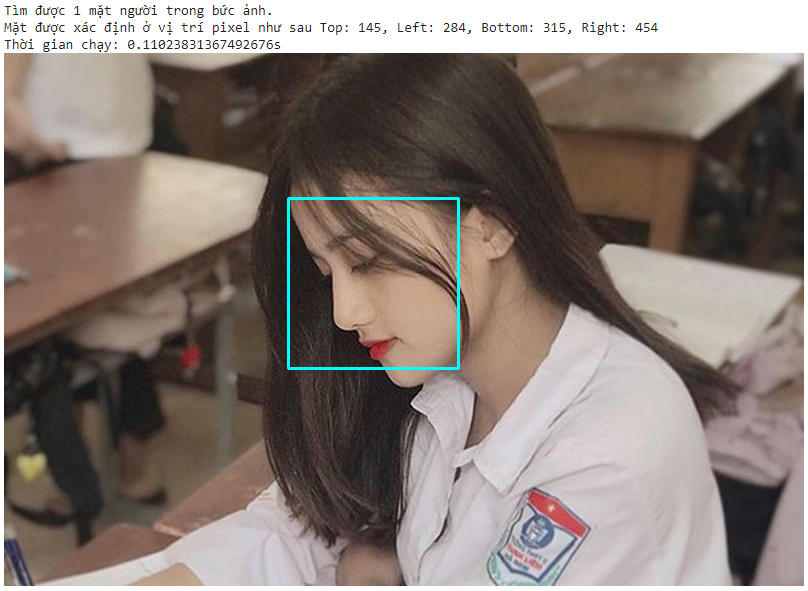
    print("Mặt được xác định ở vị trí pixel như sau Top:{}, Left:{}, Bottom:{}, Right:{}".format(top, left, bottom, right))

    cv2.rectangle(image, (left, top), (right, bottom), (255, 255, 0), 2)

print("Thời gian chạy: {}s".format(end\_time - start\_time))

cv2\_imshow(image)

Kết quả:



## Cài đặt mô hình HOG phát hiện khuôn mặt

Phương thức face\_location mặc định sẽ sử dụng mô hình HOG để tìm khuôn mặt. Nhóm sẽ cài đặt thêm mô hình này để có thể so sánh giữa hai mô hình.

Cài đặt thuật toán:

import cv2

from google.colab.patches import cv2\_imshow

import time

import face\_recognition

imagePath="/content/drive/My Drive/MachineLearningData/gocnghieng4.jpg"

image = cv2.imread(imagePath)

start\_time = time.time()

face\_locations = face\_recognition.face\_locations(image)

end\_time = time.time()

print("Tìm được {} mặt người trong ảnh.".format(len(face\_locations)))

for face\_location in face\_locations:

    top, right, bottom, left = face\_location

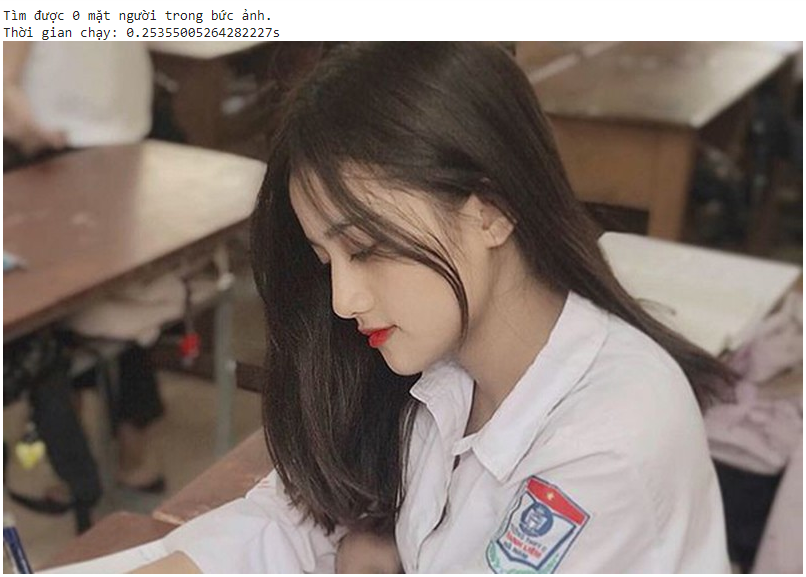
    print("Mặt được xác định ở vị trí pixel như sau Top: {}, Left: {}, Bottom: {}, Right: {}".format(top, left, bottom, right))

    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 255), 2)

print("Thời gian chạy: {}s".format(end\_time - start\_time))

cv2\_imshow(image)

Kết quả:



## Đánh giá độ chính xác của mô hình

Qua các phần cài đặt mô hình, nhóm nhận thấy độ chính xác của thuật toán phát hiện khuôn mặt tương đối chính xác (vào khoảng 80-85%). Khác với khẳng định của tác giả là 99.38%. Có lẽ tác giả đã thêm một số chỉnh sửa vào thuật toán để có thể tối ưu và phát huy được hết tiềm năng của thư viện.

So với mô hình mặc định của giải thuật là mô hình dựa trên HOG, mô hình CNN phát hiện được các khuôn mặt ở góc nghiêng khá tốt, cũng như các khuôn mặt bị biến dạng hoặc che mất một phần. Khác với HOG, chỉ có thể phát hiện tốt các khuôn mặt nhìn đối diện. Vì vậy, mô hình CNN thi thoảng cũng xử lý chậm hơn.

# **KẾT LUẬN**

Học máy (machine learning) là một trong những lĩnh vực nghiên cứu tạo nên cơn sốt cho cộng đồng công nghệ trên toàn thế giới, trở thành một trong những đề tài được khai thác nhiều nhất bởi các trang báo công nghệ uy tín. Trở thành một trong những lĩnh vực được nhiều công ty công nghệ quan tâm, từ các “ông lớn” như Google, Facebook hay Microsoft cho đến những công ty nhỏ, khởi nghiệp đều muốn đầu tư vào lĩnh vực này do tiềm năng phát triển và lợi ích nó mang lại là rất lớn. Machine learning được ứng trong nhiều lĩnh vực cuộc sống chứ không riêng gì trong giới công nghệ, một số ngành ít liên quan hơn hóa học, y học, chính trị cũng chịu ảnh hưởng bởi sự phát triển của nó. Qua học phần Machine Learning của mình thì nhóm tác giả đã thực hiện bài toán phát hiện khuôn mặt làm bài tập lớn để kết thúc học phần.

Thông qua việc tìm hiểu và nghiên cứu bài tập lớn, nhóm đã có cái nhìn toàn diện hơn trong việc sử dụng Machine Learning vào giải quyết các vấn đề thực tế. Do thời gian có hạn nên đề tài của nhóm không tránh khỏi những sai sót, mong thầy cô góp ý, đánh giá giúp nhóm hoàn thiện hơn đề tài.

Cuối cùng, nhóm cảm ơn thầy Nguyễn Văn Hậu đã hỗ trợ nhóm hoàn thiện báo cáo bài tập lớn này. Chúng em chân thành cảm ơn!