**BỘ QUỐC PHÒNG**

**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

-----🙞🙜🕮🙞🙜-----



**BÀI TẬP LỚN KHAI PHÁ DỮ LIỆU**

Sinh viên: Nguyễn Ngọc Khánh - 16151507

Đề tài: Nhận dạng khuôn mặt và áp dụng vào việc chấm công dựa vào camera

**HÀ NỘI, 12/2020**

# Tổng quan về nhận dạng khuôn mặt

## Nhận dạng khuôn mặt là gì?

Hệ thống nhận dạng khuôn mặt là một ứng dụng máy tính tự động xác định hoặc nhận dạng một người nào đó từ một bức hình ảnh hoặc một khung hình video. Một trong những cách để thực hiện điều này là so sánh các đặc điểm khuôn mặt từ hình ảnh với một cơ sở dữ liệu về khuôn mặt có từ trước.

Hệ thống này thường được sử dụng trong các hệ thống an ninh, các hệ thống chấm công tự động ...

Một số ngân hàng ở Autralia bắt đầu ứng dụng xác thực khuôn mặt trong các giao dịch ATM.

Tại Trung Quốc, hệ thống nhận diện khuôn mặt được triển khai trên toàn quốc giúp chấm điểm công dân và đồng thời xác minh nhiều tội phạm lẩn trốn.

Tại công ty em, hệ thống nhận diện khuôn mặt được áp dụng để chấm công cho nhân viên.

Nhận diện khuôn mặt có tính ứng dụng cao. Rất nhiều công ty, doanh nghiệp và quốc gia đang cần.

## Nhận dạng khuôn mặt trong chấm công tự động

Điểm đặc biệt về cấu tạo của giải pháp chấm công khuôn mặt là tích hợp hệ thống camera để chụp lại khuôn mặt của người chấm công. Cơ chế hoạt động của máy là đối chiếu ảnh chụp khuôn mặt thu được từ mỗi lần chấm công với hình ảnh khuôn mặt có sẵn trong cơ sở dữ liệu để xác định có đúng nhân viên đó đến làm việc hay không và nhân viên đó đến khi nào, về khi nào.

Các thông tin về nhân viên, thời gian chấm công cũng được lưu vào bộ nhớ máy, sau đó bộ phận nhân sự có thể chuyển những dữ liệu này về máy tính để tổng hợp, xử lý bằng phần mềm, từ đó có căn cứ để tính lương, thưởng phạt đối với người lao động.

## Các bước triển khai

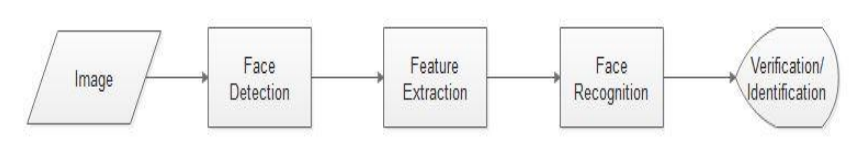
Yêu cầu:

Trước cửa ra vào công ty sẽ lắp một camera, nhân viên khi đến có thể đứng trước camera vài giây để điểm danh, hệ thống sẽ mất một chút thời gian để:

- Tìm ra vị trí khuôn mặt trong bức ảnh (face detection)

- Trích xuất đặc trưng của khuôn mặt (feature extraction)

- Sau đó nhận ra đó là nhân viên nào (recognition), nếu đó là nhân viên công ty sẽ chấm công và mở cửa cho người đó vào

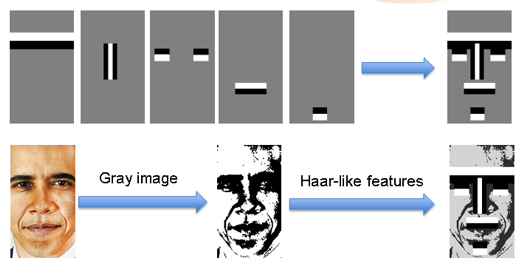


## Một số kỹ thuật nhận dạng khuôn mặt

**OpenCV Cascade nhận dạng như thế nào?**

OpenCV Cascade sẽ sử dụng một model haarcascade\_frontalface để lưu các đặc trưng khuôn mặt như mắt mũi miệng, được train từ trước

Các đặc trưng haar-like (haar-like features) là các hình chữ nhật được phân thành các vùng khác nhau.



Đầu tiên đưa ảnh về xám hóa (gray), sau đó cho các đặc trưng haar-like trượt khắp bức ảnh, những khu vực so sánh giống với nhiều đặc trưng haar-like nhất được đánh dấu lại. Vùng nào có đủ các đặc trưng bên trên sẽ được coi là khuôn mặt. Từ bước trên, ta có được bounding box của các khuôn mặt có trong bức ảnh.

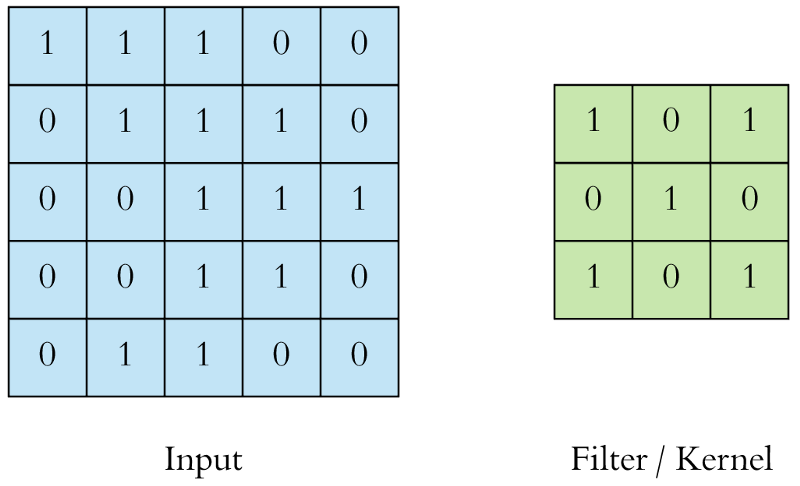
**CNN là gì?**

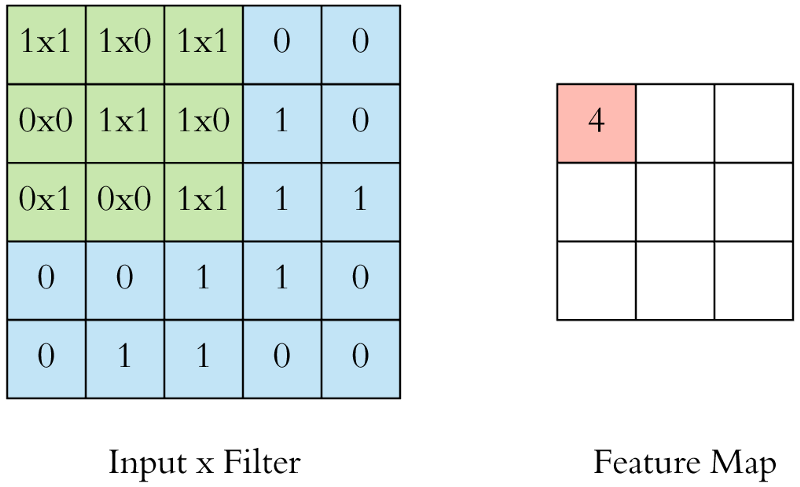
Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

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:



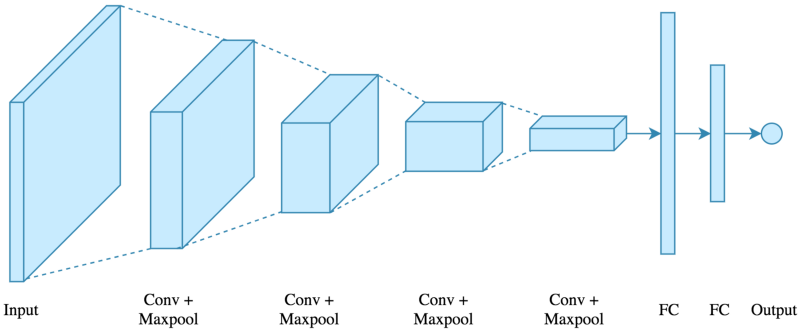


Quét **filter** qua từng phần tử của input và nhân tương ứng các ô cùng vị trí, rồi cộng tổng kết quả ta thu được một giá trị tại feature map. Lần lượt thực hiện cho đến hết từ trái qua phải, từ trên xuống dưới ta sẽ thu được một ma trận output mới.

**Pooling**

Mục đích của pooling rất đơn giản, nó làm giảm số tham số mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh overfitting. Loại pooling ta thường gặp nhất là max pooling, lấy giá trị lớn nhất trong một cửa sổ trượt. Pooling hoạt động gần giống với convolution, nó cũng có 1 cửa sổ trượt gọi là pooling window, cửa sổ này trượt qua từng giá trị của ma trận dữ liệu đầu vào (thường là các feature map trong convolutional layer), chọn ra một giá trị từ các gía trị nằm trong cửa sổ trượt (với max pooling ta sẽ lấy giá trị lớn nhất). Cuối cùng cũng thu được một ma trận output tương tự Convolutional.

**Cấu trúc một mạng CNN:**



Ta có ảnh input đầu vào. Qua hàng loạt các **Convolutional Layer** cùng **Max Pooling** (thường pooling sẽ theo ngay sau 1 convolutional layer), cuối cùng là 2 fully connected neural network, ta sẽ đưa ra được output cuối của bức hình.

**Với bước Face Detection:**

Sử dụng OpenCV CascadeClassifier có thể detect khuôn mặt rất nhanh, tuy nhiên còn thiếu chính xác và chỉ tìm được khuôn mặt chứ không xác định được vị trí các bộ phận trên khuôn mặt

Sử dụng MTCNN có thể detect khá chính xác, hơn nữa có thể trả về nhiều thông tin từ khuôn mặt đến vị trí mắt, mũi, miệng

Ngoài ra còn có Dlib trong OpenCV

**Với bước Trích xuất đặc trưng:**

Mô hình facenet (xây dựng dựa trên mạng CNN, Tensorflow), thông qua nhiều bước convolutional, pooling, cuối cùng là các lớp fully connected sẽ trích xuất đặc trưng trên một khuôn mặt thành một vector đặc trưng có số chiều thấp (ở mô hình này số chiều là 160)

**Với bước Face Recognition:**

Sau khi tìm ra vector đặc trưng, có thể sử dụng các hàm tính khoảng cách để tìm ra khuôn mặt giống với các khuôn mặt đã train từ trước nhất, tuy nhiên với một ngưỡng cho trước, chẳng hạn với ngưỡng bằng 1.0, những vector đặc trưng có khoảng cách với vector đang xét < 1.0 mới được chấp nhận



# Xây dựng chương trình

**Lưu trữ dữ liệu**

Phần mềm sẽ sử dụng OpenCV Cascade và Facenet để xác định khuôn mặt, các khuôn mặt sau khi train sẽ được lưu dưới dạng json

Mỗi khuôn mặt sẽ có một nhãn kèm theo danh sách các vector đặc trưng

Format file json:

{

"data": [

{"name": "Nhan Vien 1", "values": [vector11, vector12, ...]},

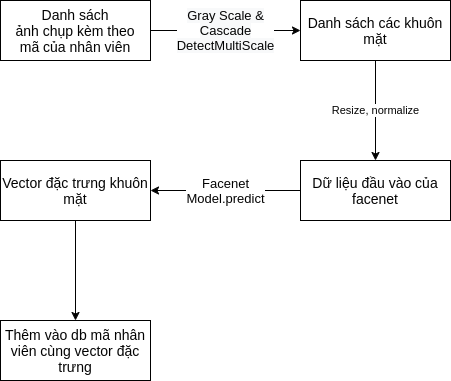
{"name": "Nhan Vien 2", "values": [vector21, vector22, ...]},

... ]

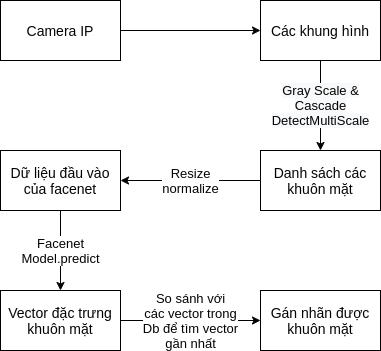
}

**Các bước triển khai**

* **Bước train**



* **Bước dự đoán**

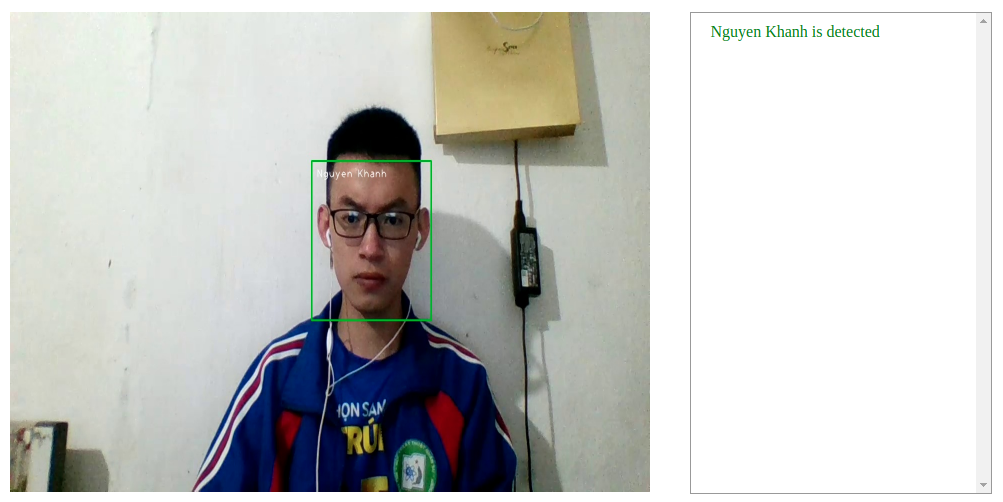


# Demo

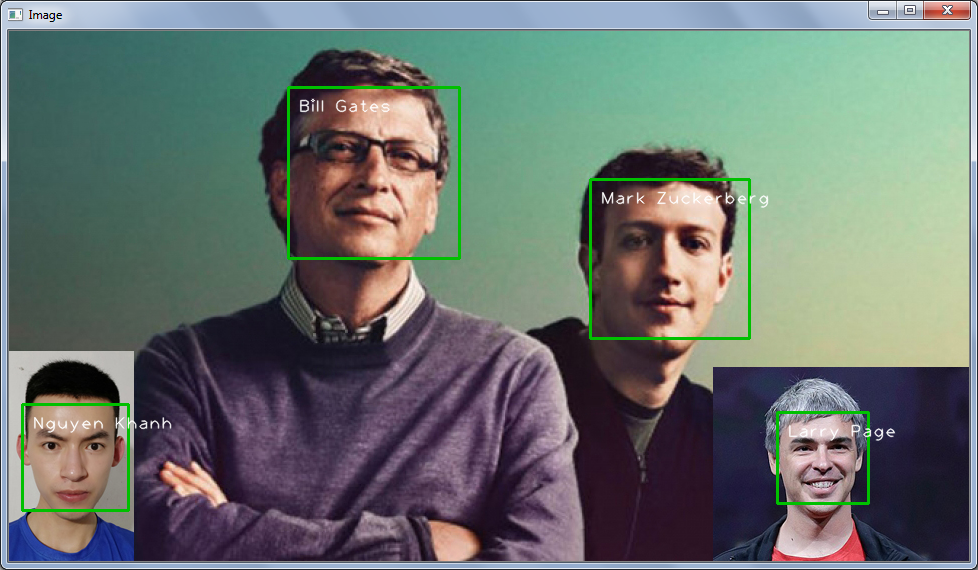
Trong main.py, sau khi khởi tạo đối tượng timekeeper

* Gọi hàm timekeeper.train() để train ra danh sách đặc trưng khuôn mặt, chương trình sẽ tự động quét toàn bộ images/train, mỗi thư mục sẽ là một nhân viên, sau khi train, toàn bộ đặc trưng sẽ được lưu tại models/data.json
* Gọi hàm timekeeper.run() để bắt đầu quá trình nhận diện khuôn mặt, trong hàm \_\_init\_\_() có thể sử dụng cv2.VideoCapture() để đọc các frame ảnh từ webcam hoặc file video, sau khi ảnh được dự đoán xong sẽ được stream sang gui.html

**Giao diện GUI**



Chương trình cho phép nhận diện nhiều khuôn mặt một lúc



# Kết luận

Chương trình đã cài đặt thành công các thuật toán phát hiện khuôn mặt, nhận dạng khuôn mặt, thực thi khá nhanh

Tuy nhiên, chương trình còn phụ thuộc vào nhiều yếu tố môi trường, hướng nghiêng khuôn mặt, để tăng chính xác cho hệ thống cần bổ sung thêm các biện pháp tiền xử lý, bổ sung thêm nhiều bộ dữ liệu

# Tham khảo

* <https://viblo.asia/p/opencv-nhan-dang-face-recognition-ZjlearmzkqJ>
* <https://phamdinhkhanh.github.io/2020/03/12/faceNetAlgorithm.html>