

**BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ỨNG DỤNG DEEP LEARNING NHẬN DIỆN KHUÔN MẶT  
CHO HỆ THỐNG ĐIỀM DANH**

**CBHD: Ts. Nguyễn Thị Mỹ Bình**

**Sinh viên: Phạm Hồng Thái**

**Mã số sinh viên: 2017605233**

Hà Nội – 2021

Phạm Hồng Thái

KHOA CÔNG NGHỆ THÔNG TIN

**BỘ CÔNG THƯƠNG  
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**



**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ỨNG DỤNG DEEP LEARNING NHẬN DIỆN KHUÔN MẶT  
CHO HỆ THỐNG ĐIỀM DANH**

**CBHD: Ts. Nguyễn Thị Mỹ Bình**

**Sinh viên: Phạm Hồng Thái**

**Mã số sinh viên: 2017605233**

Hà Nội – 2021

Phạm Hồng Thái

KHOA CÔNG NGHỆ THÔNG TIN

## Lời cảm ơn

Lời đầu tiên, em xin bày tỏ sự cảm ơn chân thành đối với Cô giáo, TS. Nguyễn Thị Mỹ Bình – giáo viên hướng dẫn trực tiếp em.

Em cũng xin gửi lời cảm ơn tới các thầy cô trong khoa Công nghệ thông tin, trường Đại học Công Nghiệp Hà Nội đã hướng dẫn, chỉ bảo và tạo điều kiện cho em học tập cũng như nghiên cứu trong thời gian qua.

Cảm ơn Câu lạc bộ HIT, Đội Olympic Tin học khoa Công nghệ thông tin đã đồng hành cùng em trong suốt quãng thời gian học tập, làm việc tại trường.

Mặc dù đã cố gắng hoàn thành báo cáo đồ án tốt nghiệp này nhưng chắc chắn sẽ không tránh khỏi những sai sót, em kính mong nhận được sự thông cảm và chỉ bảo của các thầy cô và các bạn.

## Mục lục

<b>MỞ ĐẦU .....</b>	4
0.1 Lý do chọn đề tài .....	4
0.2 Mục đích của đề tài .....	5
0.3 Đối tượng và phạm vi nghiên cứu của đề tài .....	5
0.4 Kết cấu của đồ án .....	6
0.5 Bảng các ký hiệu .....	7
<b>1 Nghiên cứu tổng quan .....</b>	8
1.1 Các phương pháp nghiên cứu .....	8
1.2 Ưu nhược điểm của các phương pháp .....	8
<b>2 Cơ sở lý thuyết .....</b>	10
2.1 Tổng quan về nhận diện khuôn mặt .....	10
2.2 Tìm hiểu về OpenCV và ngôn ngữ lập trình Python .....	11
2.3 Mô hình mạng neural tích chập (CNN - Convolutional neural network) ..	12
2.4 Máy dò khuôn mặt (Face detector) .....	19
2.5 Các kỹ thuật làm giàu dữ liệu (Data augmentation) .....	25

2.6	Các thuật toán học sâu sử dụng trong nhận diện khuôn mặt . . . . .	26
2.7	Mô hình học sâu được huấn luyện trước (Pre-train model) . . . . .	29
<b>3</b>	<b>Thiết kế và xây dựng hệ thống . . . . .</b>	<b>34</b>
3.1	Phân tích . . . . .	34
3.2	Xây dựng . . . . .	35
<b>4</b>	<b>Kết luận và hướng phát triển . . . . .</b>	<b>43</b>
4.1	Kết luận . . . . .	43
4.2	Hướng phát triển . . . . .	43
	<b>Tài liệu tham khảo . . . . .</b>	<b>45</b>

# MỞ ĐẦU

## 0.1. Lý do chọn đề tài

Với sự phát triển không ngừng của khoa học và công nghệ, đặc biệt là với những chiếc điện thoại thông minh (smartphone) ngày càng hiện đại và được sử dụng phổ biến trong đời sống con người đã làm cho lượng thông tin thu được bằng hình ảnh ngày càng tăng. Theo đó, lĩnh vực xử lý ảnh cũng được chú trọng phát triển, ứng dụng rộng rãi trong đời sống xã hội hiện đại. Không chỉ dừng lại ở việc chỉnh sửa, tăng chất lượng hình ảnh mà với công nghệ xử lý ảnh hiện nay chúng ta có thể giải quyết các bài toán nhận dạng chữ viết, nhận dạng dấu vân tay, nhận dạng khuôn mặt... Một trong những bài toán được nhiều người quan tâm nhất của lĩnh vực xử lý ảnh hiện nay đó là nhận dạng khuôn mặt (Face Recognition). Như chúng ta đã biết, khuôn mặt đóng vai trò quan trọng trong quá trình giao tiếp giữa người với người, nó mang một lượng thông tin giàu có, chẳng hạn như từ khuôn mặt chúng ta có thể xác định giới tính, tuổi tác, chủng tộc, trạng thái cảm xúc, đặc biệt là xác định mối quan hệ với đối tượng (có quen biết hay không).

Có rất nhiều phương pháp nhận dạng khuôn mặt để nâng cao hiệu suất tuy nhiên dù ít hay nhiều những phương pháp này đang vấp phải những thử thách về độ sáng, hướng nghiêng, kích thước ảnh, hay ảnh hưởng của tham số môi trường. Bài toán Nhận diện khuôn mặt (Face Recognition) bao gồm nhiều bài toán khác nhau như: phát hiện mặt người (face detection), đánh dấu (facial landmarking), trích chọn(rút) đặc trưng (feature extraction), gán nhãn, phân lớp (classification). Trong thực tế, nhận dạng khuôn mặt người (Face Recognition) là một hướng nghiên cứu được nhiều nhà khoa học quan tâm, nghiên cứu để ứng dụng trong thực tiễn. Vì thế có những cải tiến nghiên cứu về bài toán phát hiện khuôn mặt người trong những môi trường phức tạp

---

hơn, có nhiều khuôn mặt người trong ảnh hơn, và có nhiều tư thế thay đổi trong ảnh... Trong bài này tôi sẽ tìm hiểu về trích rút đặc trưng sử dụng học sâu (Deep learning) và áp dụng vào bài toán nhận diện khuôn mặt .

## 0.2. Mục đích của đề tài

- Xây dựng, tìm kiếm các mô hình học sâu để trích xuất thông tin khuôn mặt người dùng 1 cách chính xác, hiệu quả.
- Tìm hiểu sử dụng thành thạo các phương pháp làm giàu dữ liệu, ứng dụng cho bài toán nhận diện với dữ liệu là các ảnh khuôn mặt
- Xây dựng các phương pháp thu thập, làm giàu dữ liệu khuôn mặt
- Tìm hiểu ứng dụng các thuật toán phân loại trong học máy để áp dụng vào bài toán nhận diện khuôn mặt

## 0.3. Đối tượng và phạm vi nghiên cứu của đề tài

### 0.3.1. Đối tượng

- Các mô hình học sâu nổi tiếng về nhận dạng khuôn mặt được các nhà khoa học huấn luyện trước với những bộ dữ liệu cực lớn và chuẩn xác.
- Các mô hình, phương pháp phát hiện khuôn mặt trong ảnh với độ chính xác cao
- Các phương pháp làm giàu dữ liệu, đặc biệt là với dữ liệu khuôn mặt.
- Các phương pháp phân loại dữ liệu trong học máy và học sâu
- Các phương pháp đánh giá măt mát thường dùng để huấn luyện các mô hình học máy, học sâu

### 0.3.2. Phạm vi nghiên cứu

- Tập trung sử dụng bộ dữ liệu tự tạo của các sinh viên trong trường đại học Công nghiệp Hà Nội
- Huấn luyện mô hình theo phương pháp đánh giá bộ ba (triplet loss)
- Sử dụng các đánh giá cơ bản để đánh giá tính chính xác của mô hình

### 0.4. Kết cấu của đồ án

Gồm 4 chương:

Chương 1: Nghiên cứu tổng quan

Chương 2: Cơ sở lý thuyết

Chương 3: Thiết kế và xây dựng hệ thống

Chương 4: Kết luận và hướng phát triển

## 0.5. Bảng các ký hiệu

Các ký hiệu sử dụng trong sách được liệt kê trong Bảng 0.1.

Bảng 0.1: Các quy ước ký hiệu và tên gọi được sử dụng trong báo cáo

Ký hiệu	Ý nghĩa
$x, y, N, k$	in nghiêng, thường hoặc hoa, là các số vô hướng
$\mathbf{x}, \mathbf{y}$	in đậm, chữ thường, là các vector
$\mathbf{X}, \mathbf{Y}$	in đậm, chữ hoa, là các ma trận
$\mathbb{R}$	tập hợp các số thực
$\mathbb{N}$	tập hợp các số tự nhiên
$\mathbb{C}$	tập hợp các số phức
$\mathbb{R}^m$	tập hợp các vector thực có $m$ phần tử
$\mathbb{R}^{m \times n}$	tập hợp các ma trận thực có $m$ hàng, $n$ cột
$\mathbb{S}^n$	tập hợp các ma trận vuông đối xứng bậc $n$
$\mathbb{S}_+^n$	tập hợp các ma trận nửa xác định dương bậc $n$
$\mathbb{S}_{++}^n$	tập hợp các ma trận xác định dương bậc $n$
$\in$	phần tử thuộc tập hợp
$\exists$	tồn tại
$\forall$	mọi
$\triangleq$	ký hiệu là/bởi. Ví dụ $a \triangleq f(x)$ nghĩa là “ký hiệu $f(x)$ bởi $a$ ”.
$x_i$	phần tử thứ $i$ (tính từ 1) của vector $\mathbf{x}$
$\text{sgn}(x)$	hàm xác định dấu. Bằng 1 nếu $x \geq 0$ , bằng -1 nếu $x < 0$ .
$\exp(x)$	$e^x$
$\log(x)$	logarit <i>tự nhiên</i> của số thực dương $x$
$\underset{x}{\text{argmin}} f(x)$	giá trị của $x$ để hàm $f(x)$ đạt giá trị nhỏ nhất
$\underset{x}{\text{argmax}} f(x)$	giá trị của $x$ để hàm $f(x)$ đạt giá trị lớn nhất
$\text{o.w}$	<i>otherwise</i> – trong các trường hợp còn lại
$\frac{\partial f}{\partial x}$	đạo hàm của hàm số $f$ theo $x \in \mathbb{R}$
$\nabla_{\mathbf{x}} f$	gradient của hàm số $f$ theo $\mathbf{x}$ ( $\mathbf{x}$ là vector hoặc ma trận)
$\nabla_{\mathbf{x}}^2 f$	gradient bậc hai của hàm số $f$ theo $\mathbf{x}$ , còn được gọi là <i>Hesse</i>
$\odot$	Hadamard product (elementwise product). Phép nhân từng phần tử của hai vector hoặc ma trận cùng kích thước.
$\propto$	tỉ lệ với
$\rule[1.5ex]{0pt}{0pt}$	đường nét liền
$\rule[-0.5ex]{0pt}{0pt}$	đường nét đứt
$\dots\dots$	đường nét chấm (đường chấm chấm)
$\cdots\cdots$	đường chấm gạch
	nền chấm
	nền sọc chéo

---

**Chương 1**

---

# Nghiên cứu tổng quan

---

## 1.1. Các phương pháp nghiên cứu

- Hiện nay có 2 phương pháp nhận diện khuôn mặt được sử dụng rộng rãi nhất là:
  - Nhận dạng dựa trên các đặc trưng của các phần tử trên khuôn mặt (Feature base face recognition)
  - Nhận dạng dựa trên xét tổng thể khuôn mặt (Appearance based face recognition)
- Ngoài ra còn một số phương pháp về loại sử dụng mô hình về khuôn mặt:
  - Nhận dạng 2D: Elastics Bunch Graph, Active Appearance Model.
  - Nhận dạng 3D: 3D Morphale Model

## 1.2. Ưu nhược điểm của các phương pháp

### 1.2.1. Nhận dạng dựa trên các đặc trưng của các phần tử trên khuôn mặt:

Đây là phương pháp nhận dạng khuôn mặt dựa trên việc xác định các đặc trưng hình học của các chi tiết trên một khuôn mặt (vị trí, diện tích, hình dạng của mắt, mũi, miệng, ...) và mối quan hệ giữa chúng (khoảng cách của hai mắt, khoảng cách của hai lông mày, ...).

Ưu điểm của phương pháp này là nó gần với cách mà con người sử dụng để nhận biết khuôn mặt. Hơn nữa với việc xác định đặc tính cà mối quan hệ, phương pháp này

---

có thể cho kết quả tốt trong các trường hợp ảnh có nhiều nhiễu như bị nghiêng, bị xoay hoặc ánh sáng thay đổi.

Nhược điểm của phương pháp này là cài đặt thuật toán phức tạp do việc xác định mối quan hệ giữa các đặc tính sẽ khó phân biệt. Mặt khác, với các ảnh kích thước bé thì các đặc tính sẽ khó phân biệt.

### **1.2.2. Nhận dạng dựa trên xét tổng thể khuôn mặt:**

Đây là phương pháp xem mỗi ảnh có kích thước  $R \times C$  là một vector trong không gian  $R \times C$  chiều. Ta sẽ xây dựng một không gian mới có chiều nhỏ hơn sao cho khi biểu diễn trong không gian có các đặc điểm chính của một khuôn mặt không bị mất đi. Trong không gian đó, các ảnh cùng một người sẽ được tập trung lại một nhóm gần nhau và cách xa các nhóm khác.

Ưu điểm của phương pháp này là tìm được các đặc tính tiêu biểu của đối tượng cần nhận dạng mà không cần phải xác định các thành phần và mối quan hệ giữa các thành phần đó. Phương pháp sử dụng thuật toán có thể thực hiện tốt với các ảnh có độ phân giải cao, thu gọn ảnh thành một ảnh có kích thước nhỏ hơn. Có thể kết hợp các phương pháp khác như mạng Nơ-ron, Support Vector Machine.

Nhược điểm của phương pháp này phân loại theo chiều phân bố lớn nhất của vector. Tuy nhiên, chiều phân bố lớn nhất không phải lúc nào cũng mang lại hiệu quả tốt nhất cho bài toán nhận dạng và đặc biệt là phương pháp này rất nhạy với nhiễu.

### **1.2.3. Kết luận**

Vì kết quả nghiên cứu cuối cùng là ứng dụng với yêu cầu về độ chính xác cao, khả năng thích ứng linh hoạt, hoạt động ổn định trong môi trường thực tế và hoạt động với các camera với độ phân giải thấp. Tôi quyết định chọn phương pháp nhận dạng dựa trên xét tổng thể khuôn mặt (Appearance based face recognition).

---

***Chương 2***

---

---

# Cơ sở lý thuyết

---

## 2.1. Tổng quan về nhận diện khuôn mặt

Nhận diện khuôn mặt (Face recognition) đang được ứng dụng trong nhiều lĩnh vực. Hệ thống nhận dạng khuôn mặt là một ứng dụng cho phép máy tính tự động xác định hoặc nhận dạng một người nào đó từ một bức hình ảnh kỹ thuật số hoặc một khung hình.

Nhận diện khuôn mặt là một bài toán phức tạp, đòi hỏi cần phải xử lý một loạt các vấn đề.

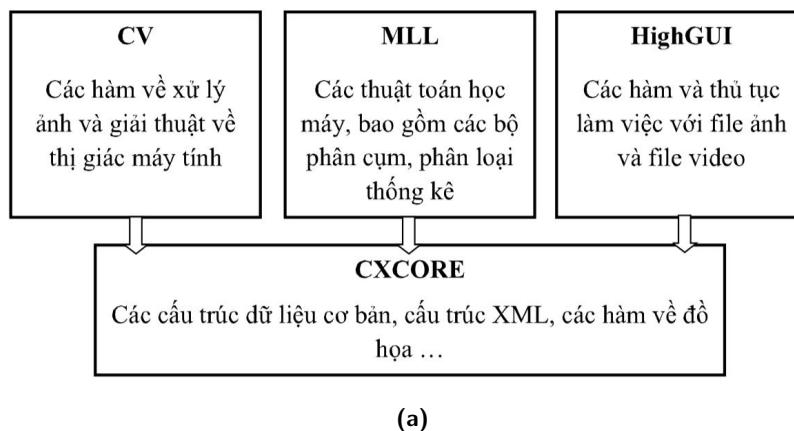
Mỗi khuôn mặt đều có những điểm mốc, những phần lồi lõm, hình dáng của các bộ phận trên khuôn mặt như mắt, mũi, miệng,... Các hệ thống nhận diện định nghĩa những điểm này là những điểm nút, và mỗi khuôn mặt có khoảng 80 nút như thế.

Ngày nay các mô hình học sâu đã phát triển một cách vượt trội, việc trích rút các đặc trưng của một khuôn mặt ngày một đơn giản và trở lên vô cùng chính xác. Mặc dù chúng ta không thể biết các đặc trưng đấy là những đặc trưng gì bởi cơ chế tự động học của các mô hình học sâu rất phức tạp. Nhưng chúng ta vẫn có thể kiểm tra độ chính xác bằng cách đưa những đặc trưng này vào các mô hình phân loại, xác nhận, ... sau đó tính toán độ chính xác trên những dữ liệu có được.

## 2.2. Tìm hiểu về OpenCV và ngôn ngữ lập trình Python

OpenCV (Open Source Computer Vision Library) là một thư viện các chức năng lập trình chủ yếu nhắm vào tầm nhìn máy tính thời gian thực. OpenCV hỗ trợ nhiều ngôn ngữ lập trình như C++, Python, Java,... và có sẵn trên các nền tảng khác nhau bao gồm Windows, Linux, Mac OS, Android và iOS. Các giao diện cho các hoạt động GPU tốc độ cao dựa trên CUDA và OpenCL cũng đang được phát triển tích cực.

Cấu trúc tổng quan của OpenCV bao gồm 5 phần chính. 4 trong 5 phần đó được chỉ ra trong hình vẽ dưới.



**Hình 2.1.** Cấu trúc các phần của OpenCV.

Phần CV bao gồm các thư viện cơ bản về xử lý ảnh và các giải thuật về xử lý ảnh. MLL là bộ thư viện về các thuật toán học máy, bao gồm rất nhiều bộ phân cụm và phân loại thống kê. HighGUI chứa đựng những thủ tục vào ra, các chức năng về lưu trữ cũng như đọc các file ảnh và video. Phần thứ 4, Cxcore chứa đựng các cấu trúc dữ liệu cơ bản (ví dụ như cấu trúc XML, các cây dữ liệu ...). Phần cuối cùng là CvAux, phần này bao gồm các thư viện cho việc phát hiện, theo dõi và nhận dạng đối tượng (khuôn mặt, mắt ...).

OpenCV - Python là một thư viện các ràng buộc Python được thiết kế để giải quyết các vấn đề về xử lý ảnh và thị giác máy tính.

Python là ngôn ngữ lập trình có mục đích chung được bắt đầu bởi Guido van Rossum, nó trở nên rất phổ biến rất nhanh trong thời gian gần đây, chủ yếu vì tính

đơn giản và khả năng đọc mã của nó. Nó cho phép lập trình viên thể hiện ý tưởng trong ít dòng mã hơn mà không làm giảm khả năng đọc.

So với các ngôn ngữ như C/C++, Python chậm hơn. Điều đó nói rằng, Python có thể dễ dàng được mở rộng với C/C++, cho phép chúng ta viết mã chuyên sâu tính toán trong C/C++ và tạo các trình bao bọc Python có thể được sử dụng làm mô-đun Python. Điều này mang lại cho chúng ta hai lợi thế: thứ nhất, mã nhanh như mã C/C++ gốc (vì đây là mã C++ thực tế hoạt động ở chế độ nền) và thứ hai, mã dễ dàng hơn trong Python so với C/C++. OpenCV - Python là một trình bao bọc Python để thực hiện OpenCV C++ ban đầu.

OpenCV - Python sử dụng Numpy, một thư viện được tối ưu hóa cao cho các hoạt động số với cú pháp kiểu MATLAB. Tất cả các cấu trúc mảng OpenCV được chuyển đổi sang và từ các mảng Numpy. Điều này cũng giúp tích hợp dễ dàng hơn với các thư viện khác sử dụng Numpy như SciPy và Matplotlib.

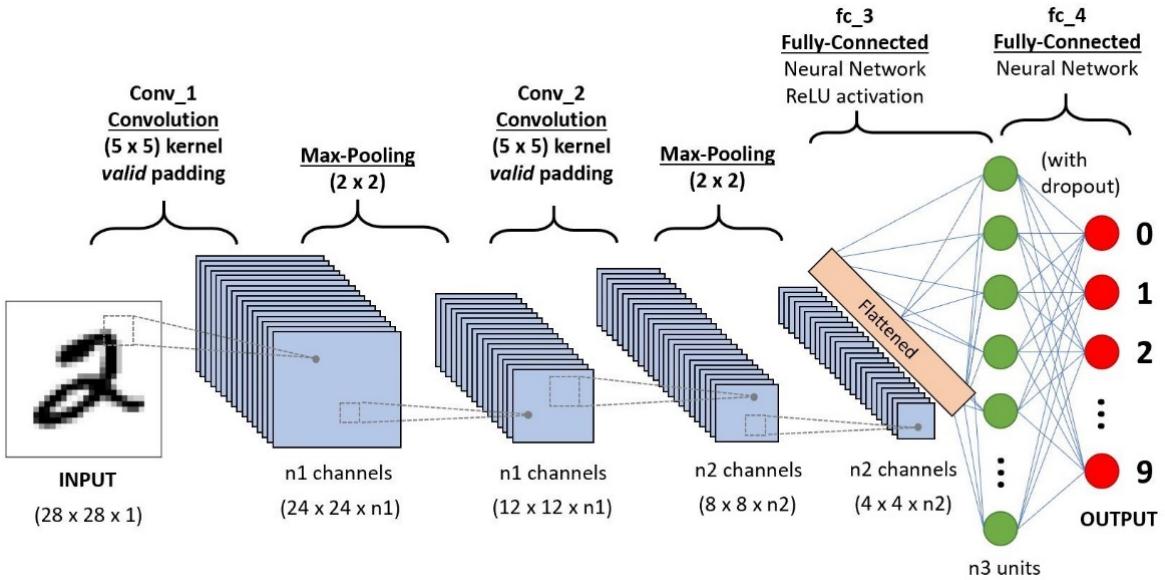
### **2.3. Mô hình mạng neural tích chập (CNN - Convolutional neural network)**

Mạng neural tích chập (CNN) là một trong những mô hình học sâu[2] tiên tiến phổ biến nhất và có ảnh hưởng nhất với công đồng thị giác máy tính (Computer vision). CNN thường được dùng trong các bài toán nhận dạng ảnh, phân tích ảnh, xử lý ngôn ngữ tự nhiên dưới dạng ảnh các bước sóng. Và hầu hết đều cho hiệu quả tốt đến rất tốt.

CNN là một kiến trúc mạng neural sinh ra để xử lý các dữ liệu phi cấu trúc dạng ảnh. Có 2 loại lớp chính trong CNN: lớp tích chập (Convolutional layer) và lớp gộp (Pooling layer)

#### **2.3.1. Lớp tích chập**

Lớp tích chập là lớp quan trọng nhất và thường cũng là lớp đầu tiên của của mô hình CNN. Lớp này có chức năng chính là phát hiện các đặc trưng có tính không gian hiệu quả. Trong tầng này có 4 đối tượng chính là: ma trận đầu vào, bộ lọc (filters) và trường thu cảm, bản đồ đặc trưng (feature map). Lớp tích chập nhận đầu vào là một ma trận 3 chiều và một bộ lọc cần phải học. Bộ lọc này sẽ trượt qua từng vị trí trên



**Hình 2.2.** CNN cho bài toán nhận diện chữ số.

bức ảnh để tính tích chập (convolution) giữa bộ lọc và phần tương ứng trên bức ảnh. Phần tương ứng này trên bức ảnh gọi là trường thực cảm (receptive field), tức là vùng mà một neuron có thể nhìn thấy để đưa ra quyết định, và mà trận cho ra bởi quá trình này được gọi là bản đồ đặc trưng (feature map).

Để hình dung, có thể tưởng tượng, bộ filters giống như các tháp canh trong nhà tù quét lần lượt qua không gian xung quanh để tìm kiếm tên tù nhân bỏ trốn. Khi phát hiện tên tù nhân bỏ trốn, thì chuông báo động sẽ reo lên, giống như các bộ lọc tìm kiếm được đặc trưng nhất định thì tích chập đó sẽ cho giá trị tương ứng.

#### a. Lớp tích chập được coi như xác định đặc trưng

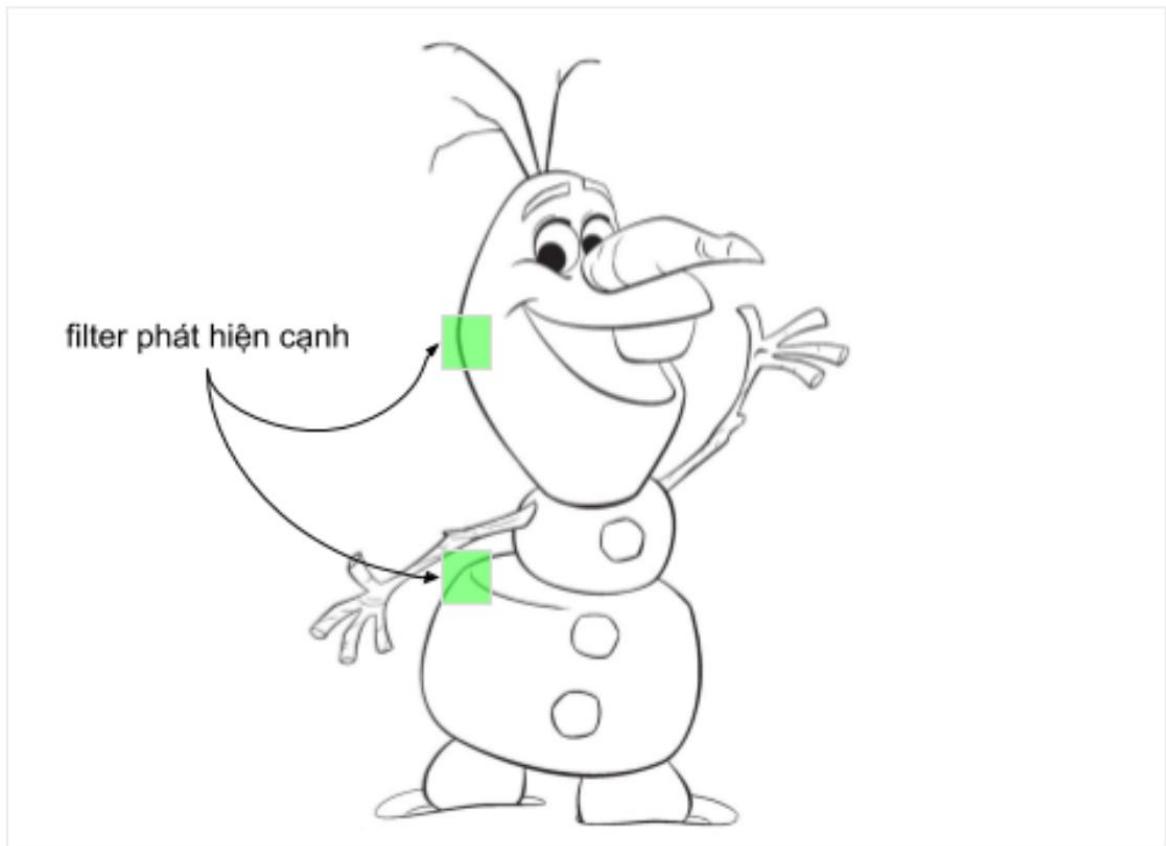
- Lớp tích chập có chức năng chính là phát hiện đặc trưng cụ thể của bức ảnh. Những đặc trưng này bao gồm đặc trưng cơ bản là góc, cạnh, màu sắc, hoặc đặc trưng phức tạp hơn như texture của ảnh. Vì bộ filter quét qua toàn bộ bức ảnh, nên những đặc trưng này có thể nằm ở vị trí bất kỳ trong bức ảnh, cho dù ảnh bị xoay trái/phải thì những đặc trưng này vẫn bị phát hiện.
- Ở minh họa dưới, có một filter 5x5 dùng để phát hiện góc/cạnh, với filter này chỉ có giá trị mờ tại các điểm tương ứng một góc cong.

0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	0	0	0
0	1	0	0	0

**Hình 2.3.** Bộ lọc phát hiện cạnh

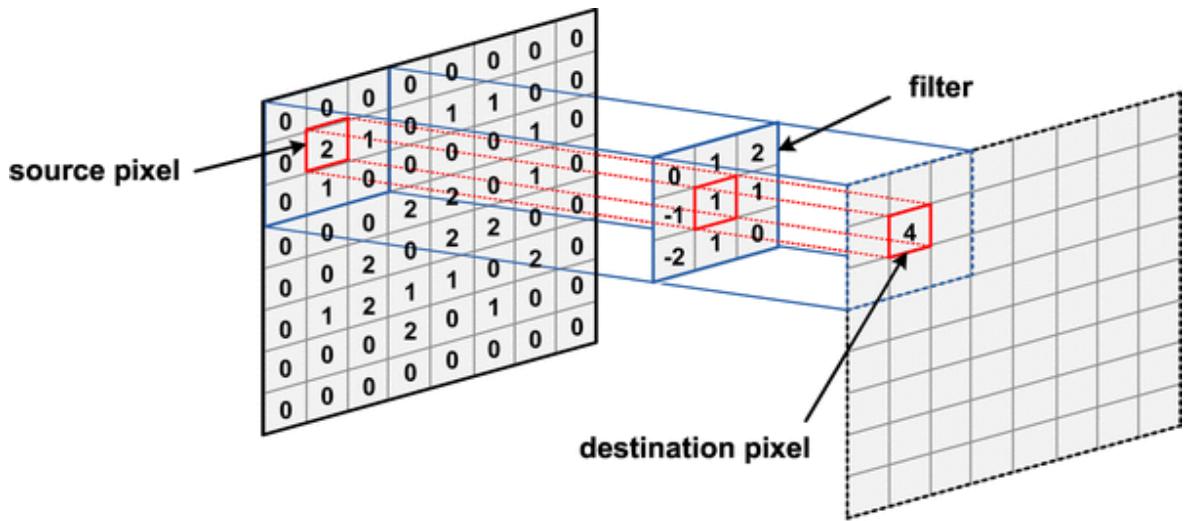
- Dùng bộ lọc ở trên trực quan ảnh của nhân vật Olaf trong bộ phim Frozen.

Chúng ta thấy rằng, chỉ ở những vị trí trên bức ảnh có dạng góc như đặc trưng ở bộ lọc mới có giá trị lớn trên bản đồ đặc trưng, những vị trí còn lại sẽ cho giá trị thấp hơn. Điều này có nghĩa là, bộ lọc đã phát hiện thành công một dạng góc/cạnh trên dữ liệu đầu vào. Tập hợp nhiều bộ lọc sẽ cho phép các bạn phát hiện được nhiều loại đặc trưng khác nhau, và giúp định danh được đối tượng.

**Hình 2.4.** Bộ lọc phát hiện cạnh

b. Các tham số: Kích thước bộ lọc, bước nhảy, lề

- Kích thước bộ lọc là một trong những tham số quan trọng nhất của lớp tích chập. Kích thước này tỉ lệ thuận với số tham số cần học tại mỗi lớp tích chập và là tham số quyết định trường thu cảm của tầng này. Kích thước phổ biến nhất của bộ lọc là 3x3.



**Hình 2.5.** Cách hoạt động của bộ lọc (filter)

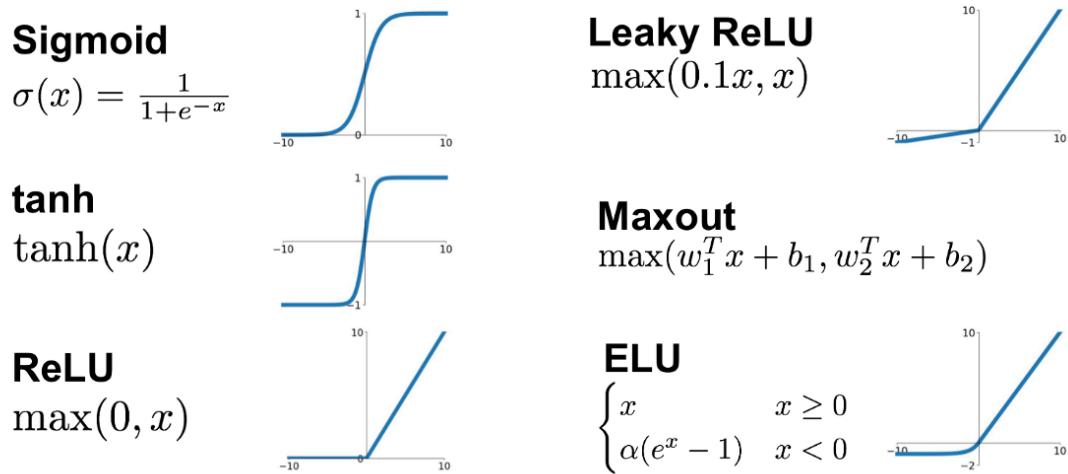
- Kích thước filter nhỏ được ưu tiên lựa chọn thay kích thước lớn vì những lý do sau đây:
  - Cho phép nhìn được các vùng nhỏ
  - Trích rút được những đặc trưng có tính cục bộ cao
  - Phát hiện đặc trưng nhỏ
  - Đặc trưng được trích rút sẽ nhiều, đa dạng
  - Giảm kích thước ảnh chập, cho phép mạng sâu hơn
  - Chia sẻ trọng số tốt hơn
- Kích thước của bộ lọc sẽ là những số lẻ để kết quả của phép tích chập sẽ nằm ở giữa ma trận

### 2.3.2. Lớp phi tuyến

Để các mô hình học sâu tìm được mối quan hệ phức tạp giữa các đặc trưng, cũng như tìm được những đặc trưng quan trọng (là sự kết hợp phi tuyến giữa các đặc trưng cơ bản khác) thì các mối quan hệ đó khó có thể được biểu diễn dưới các hàm tuyến tính mà cần sự kết hợp phi tuyến tính, chính vì vậy các hàm kích hoạt phi tuyến ra đời nhằm phá vỡ sự tuyến tính của giữa các đặc trưng từ đó tìm ra các đặc trưng mới quan trọng hơn.

Hiện nay hàm kích hoạt được sử dụng phổ biến nhất là hàm ReLU (Rectified Linear Units). Hàm ReLU được ưa chuộng vì tính đơn giản và cho kết quả tốt hơn ReLU cũng như những hàm kích hoạt khác, được đặt ngay sau tầng convolution, ReLU sẽ gán những giá trị âm bằng 0 và giữ nguyên giá trị của đầu vào khi lớn hơn 0.

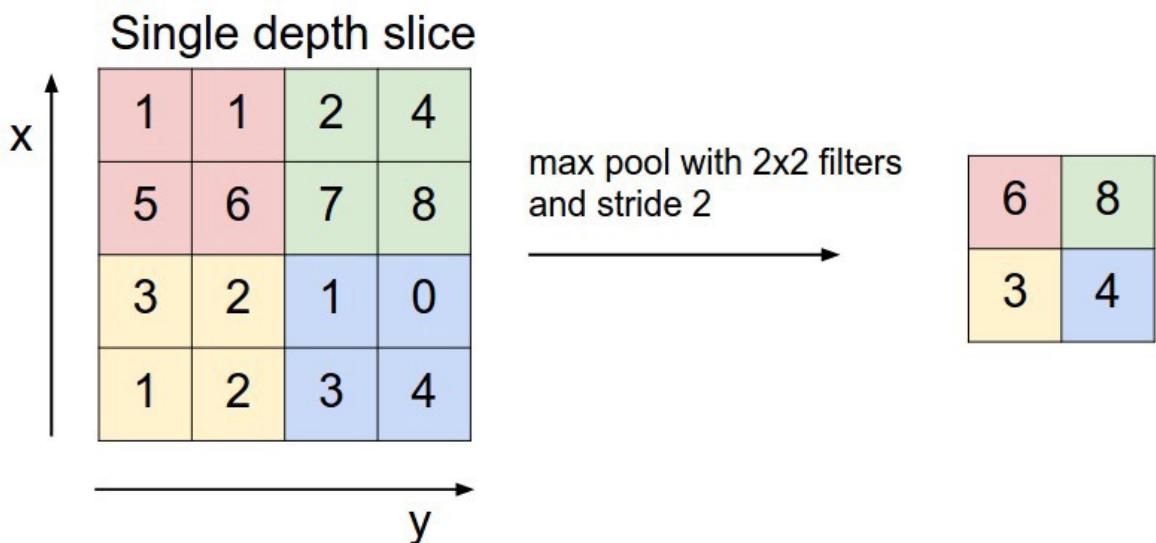
ReLU cũng có một số vấn đề tiềm ẩn như không có đạo hàm tại điểm 0, giá trị của hàm ReLU có thể lớn đến vô cùng và nếu chúng ta không khởi tạo trọng số cẩn thận, hoặc khởi tạo learning rate quá lớn thì những neuron ở tầng này sẽ rơi vào trạng thái chết, tức là luôn có giá trị  $< 0$ .



**Hình 2.6.** Một số hàm kích hoạt thường được sử dụng

### 2.3.3. Lớp gộp

Sau hàm kích hoạt, thông thường chúng ta sử dụng lớp gộp. Một số loại lớp gộp phổ biến như là max-pooling, average pooling, với chức năng chính là giảm chiều của tầng trước đó. Với một lớp gộp có kích thước  $2 \times 2$ , các bạn cần phải trượt bộ lọc  $2 \times 2$  này trên những vùng ảnh có kích thước tương tự rồi sau đó tính giá trị lớn nhất, hay trung bình cho vùng ảnh đó.



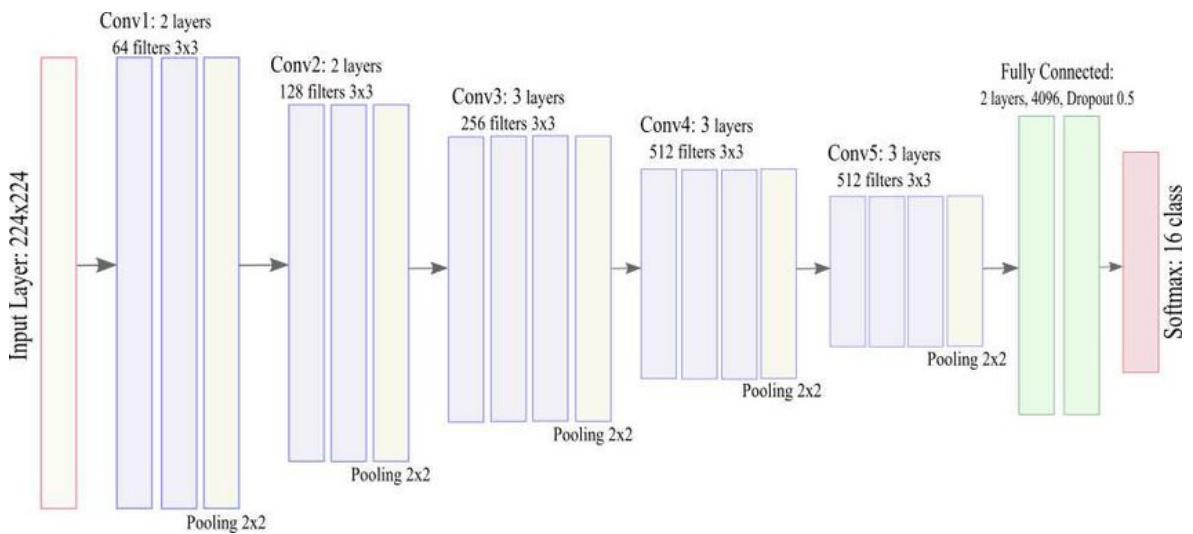
**Hình 2.7.** Ví dụ về max-pooling

Ý tưởng đằng sau lớp gộp là vị trí tuyết đối của những đặc trưng trong không gian ảnh không còn cần thiết, thay vào đó vị trí tương đối giữ các đặc trưng đã đủ để phân loại đối tượng. Hơn nữa, giảm tầng pooling có khả năng giảm chiều cực kì nhiều, làm hạn chế overfit, và giảm thời gian huấn luyện tốt.

### 2.3.4. Lớp kết nối đầy đủ

Lớp cuối cùng của mô hình CNN trong bài toán phân loại ảnh là lớp kết nối đầy đủ. Lớp này có chức năng chuyển ma trận đặc trưng ở tầng trước thành các vector chứa xác suất của các đối tượng cần được dự đoán.

Quá trình huấn luyện mô hình CNN cho bài toán phân loại ảnh cũng tương tự như huấn luyện các mô hình khác. Cần có hàm đánh giá măt mát để tính sai số giữa dự đoán của mô hình và nhãn chính xác, để sử dụng cơ chế của thuật toán lan truyền ngược (backprobagation) cho quá trình cập nhật trọng số.

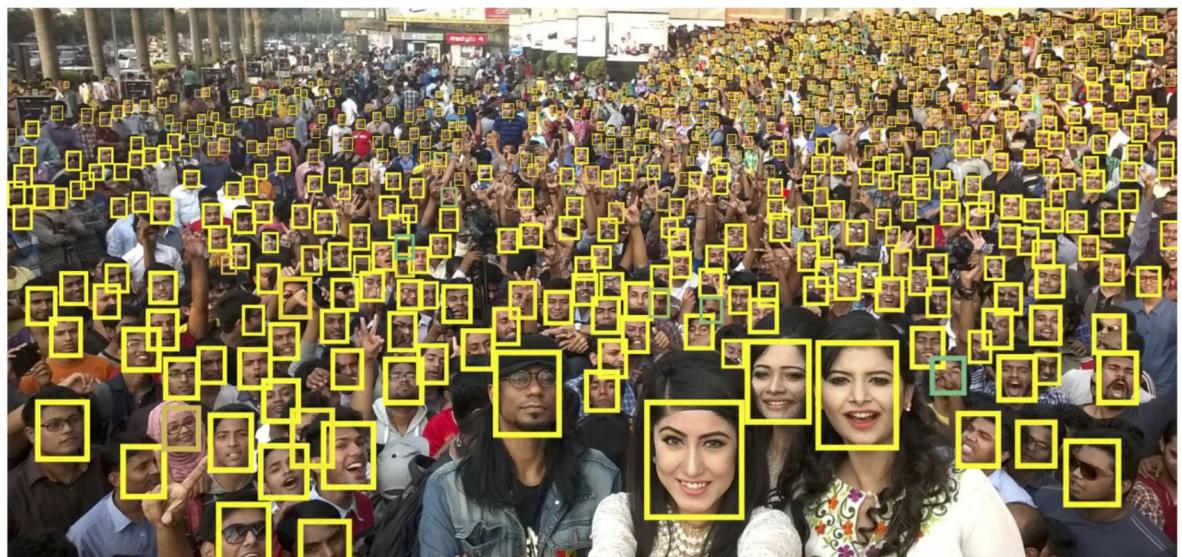


**Hình 2.8.** Một CNN đơn giản với đầy đủ các lớp

## 2.4. Máy dò khuôn mặt (Face detector)

Để xác định các khuôn mặt[4] trong các ảnh chứa nhiều yếu tố ngoại cảnh, và trích xuất các khuôn mặt đưa vào các mô hình học sâu tiến hành trích xuất các đặc trưng của các khuôn mặt này thì phải dùng các máy dò khuôn mặt.

Hiện nay có rất nhiều các máy dò khuôn mặt[5] được thiết kế bằng các mô hình học sâu khác nhau, ngay cả những máy dò được thiết kế bằng các thuật toán xử lý ảnh thông thường cũng đã được phát triển và đạt hiệu quả tốt. Ví dụ như các máy dò được tích hợp trong OpenCV (máy dò 5 điểm, 9 điểm, 68 điểm trên khuôn mặt). Nhưng để đạt được hiệu quả tốt nhất có thể thì tôi sử dụng 1 máy dò có tên MTCNN (Multi-task Cascaded Convolutional Networks) được sử dụng phổ biến trong các hệ thống đòi hỏi sự chính xác cao.

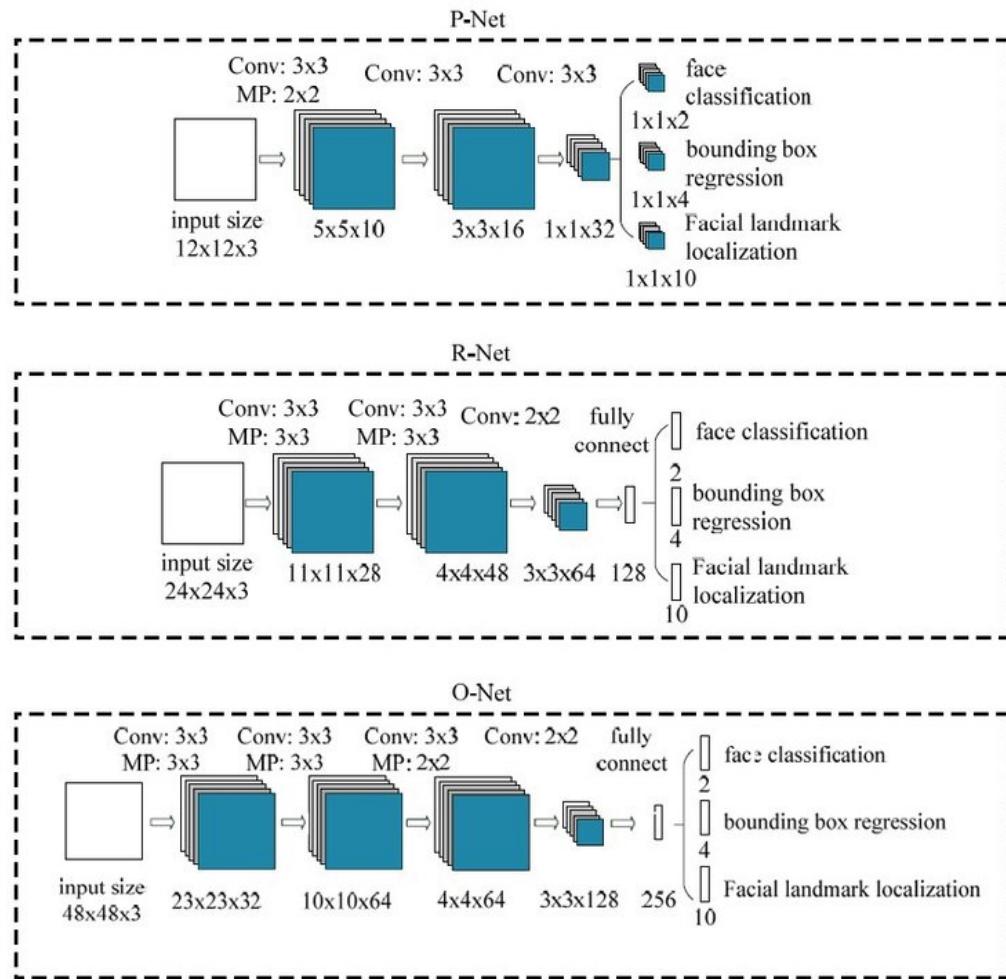


**Hình 2.9.** Ví dụ máy dò khuôn mặt

### 2.4.1. MTCNN

MTCNN[7] là viết tắt của Multi-task Cascaded Convolutional Networks (Mạng đa năng xếp tầng đa tác vụ). Nó bao gồm 3 mạng CNN xếp chồng và đồng thời hoạt động khi detect khuôn mặt. Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau trong task. Đầu ra của MTCNN là vị trí khuôn mặt và 5 điểm trên mặt: mắt, mũi, miệng...

MTCNN hoạt động theo 3 bước, mỗi bước có một mạng neural riêng lần lượt là: P-Net, R-Net và O-Net



**Hình 2.10.** Các mạng neural trong MTCNN

Với mỗi bức ảnh đầu vào, nó sẽ tạo ra nhiều bản sao của hình ảnh đó với các kích thước khác nhau.

Tại P-Net, thuật toán sử dụng 1 kernel 12x12 chạy qua mỗi bức hình để tìm kiếm khuôn mặt.



**Hình 2.11.** Mô hình P-Net

Sau lớp convolution thứ 3, mạng chia thành 2 lớp. Convolution 4-1 đưa ra xác suất của một khuôn mặt nằm trong mỗi bounding boxes, và Convolution 4-2 cung cấp tọa độ của các bounding boxes.

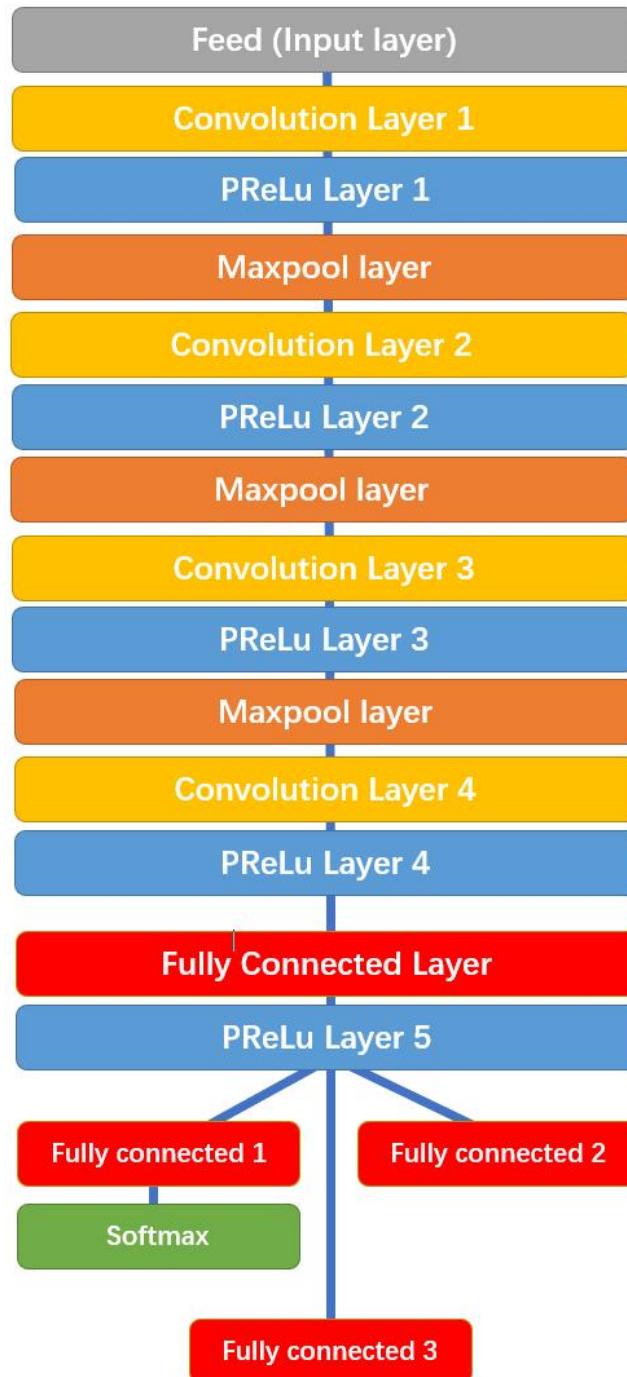
R-Net có cấu trúc tương tự với P-Net. Tuy nhiên sử dụng nhiều layer hơn. Tại đây, network sẽ sử dụng các bounding boxes được cung cấp từ P-Net và tinh chỉnh là tọa độ.



**Hình 2.12.** Mô hình R-Net

Tương tự R-Net chia ra làm 2 layers ở bước cuối, cung cấp 2 đầu ra đó là tọa độ mới của các bounding boxes, cùng độ tin tưởng của nó.

O-Net lấy các bounding boxes từ R-Net làm đầu vào và đánh dấu các tọa độ của các mốc trên khuôn mặt.



**Hình 2.13.** Mô hình O-Net

Ở bước này, thuật toán đưa ra 3 kết quả đầu ra khác nhau bao gồm: xác suất của khuôn mặt nằm trong bounding box, tọa độ của bounding box và tọa độ của các mốc trên khuôn mặt (vị trí mắt, mũi, miệng)



**Hình 2.14.** MTCNN xác định các khuôn mặt

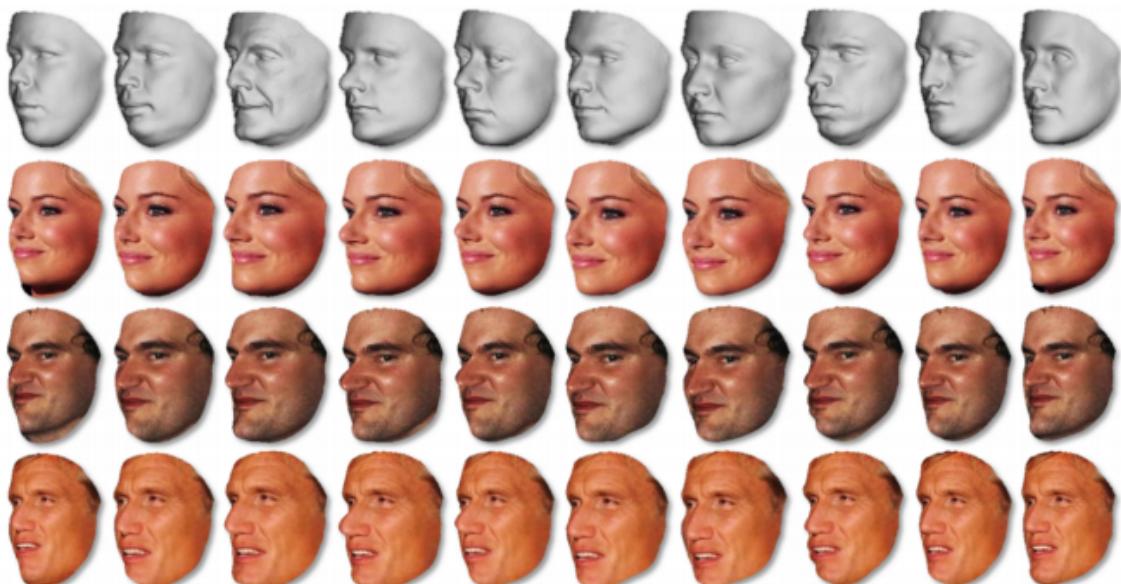
## 2.5. Các kĩ thuật làm giàu dữ liệu (Data agumentation)

Hiện nay các kĩ thuật làm giàu dữ liệu[6] đang ngày càng phát triển và phổ biến, khiến cho dữ liệu tăng lên mạnh mẽ mà vẫn dữ được tính đặc trưng

Các kĩ thuật thường được sử dụng phổ biến là: Lật, xoay, phóng to, thu nhỏ, gây nhiễu, cắt, ...

Ngoài ra còn có các kĩ thuật nâng cao khác như là: sử dụng phép nội suy, GAN, đối xứng, bọc, ...

Còn rất nhiều những kĩ thuật phức tạp khác nhau và học sâu lại được sử dụng để giải quyết vấn đề này



**Hình 2.15.** Làm giàu dữ liệu 3D sử dụng học sâu

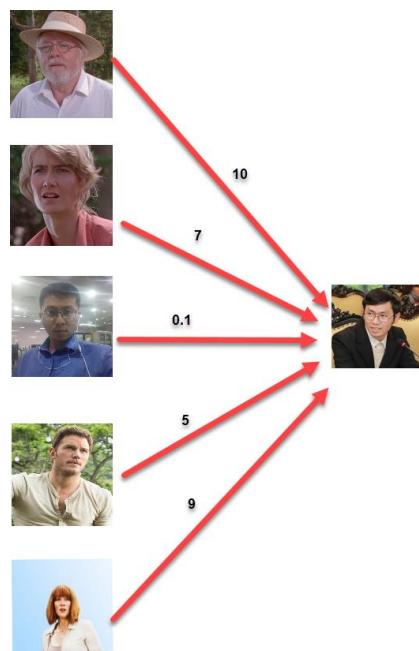
## 2.6. Các thuật toán học sâu sử dụng trong nhận diện khuôn mặt

### 2.6.1. One-shot learning

One-shot learning là thuật toán học có giám sát mà mỗi một người chỉ cần 1 vài, rất ít hoặc thậm chí chỉ 1 bức ảnh duy nhất (để khởi tạo ra nhiều biến thể). Từ đầu vào là bức ảnh của một người, chúng ta sử dụng một kiến trúc CNN đơn giản để dự báo người đó là ai. Tuy nhiên nhược điểm của phương pháp này là chúng ta phải huấn luyện lại thuật toán thường xuyên khi xuất hiện thêm một người mới vì số lượng của đầu ra thay đổi tăng lên 1. Rõ ràng là không tốt đối với các hệ thống nhận diện khuôn mặt của một công ty vì số lượng người luôn biến động theo thời gian.

### 2.6.2. Learning similarity

Phương pháp này dựa trên một phép đo khoảng cách giữa 2 bức ảnh, thông thường là các định mức chuẩn L1 hoặc L2 sao cho nếu 2 bức ảnh thuộc cùng một người thì khoảng cách là nhỏ nhất và nếu không thuộc thì khoảng cách sẽ lớn hơn.



**Hình 2.16.** Learning similarity

Learning similarity có thể trả ra nhiều hơn một ảnh là cùng loại với ảnh đầu vào tùy theo ngưỡng (threshold).

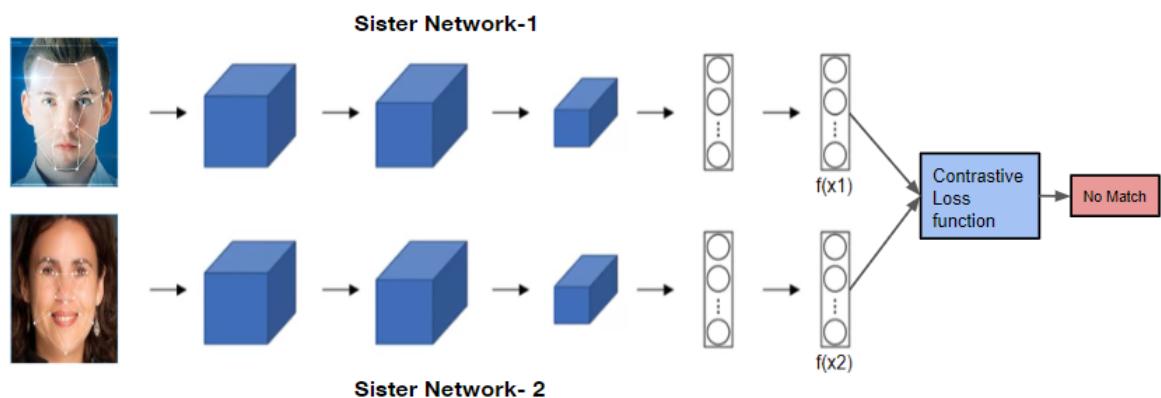
Ngoài ra phương pháp này không bị phụ thuộc vào số lượng lớp (classes). Do đó không cần phải huấn luyện lại khi xuất hiện class mới. Điểm mấu chốt là cần xây dựng được một mô hình mã hóa (model encoding) đủ tốt để chiếu các bức ảnh lên một không gian euclidean n chiều. Sau đó sử dụng khoảng cách để quyết định nhãn của chúng.

Như vậy learning similarity có ưu điểm hơn so với one-shot learning khi không phải huấn luyện lại model khi mà vẫn tìm ra được ảnh tương đồng. Vậy làm thế nào để học được biểu diễn của ảnh trong không gian euclidean n chiều? Kiến trúc siam network sẽ giúp chúng ta thực hiện điều này một cách dễ dàng.

### 2.6.3. Siam learning

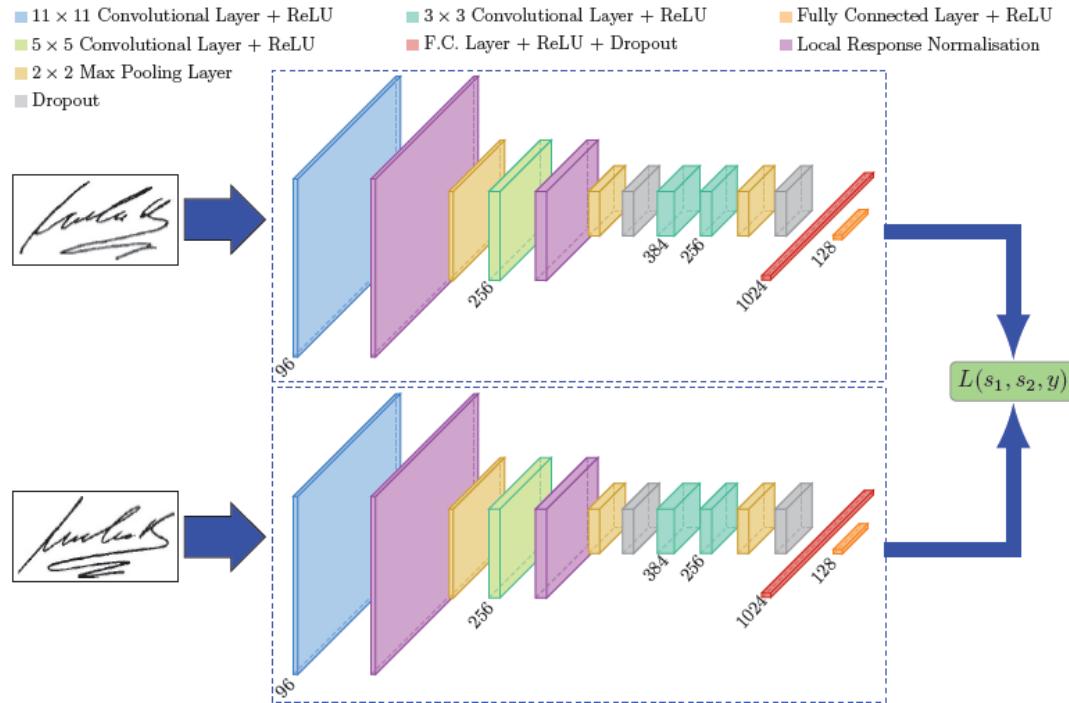
Những kiến trúc mạng mà khi bạn đưa vào 2 bức ảnh và mô hình sẽ trả lời chúng thuộc về cùng 1 người hay không được gọi chung là Siam network.

Kiến trúc của Siam network dựa trên mạng cơ sở (base network) là một mạng neural tích chập (Convolutional neural network) đã được loại bỏ các lớp đầu ra (output layer) có tác dụng mã hóa (encoding) ảnh thành vector nhúng (embedding). Đầu vào của mạng siam network là 2 bức ảnh bất kì được lựa chọn ngẫu nhiên từ dữ liệu ảnh. Output của Siam network là 2 vector tương ứng với biểu diễn của 2 ảnh đầu vào. Sau đó đưa 2 vector vào hàm đánh giá mây mát (loss function) để đo lường sự khác biệt giữa chúng. Thông thường hàm đánh giá mây mát là một hàm chuẩn bậc 2.



**Hình 2.17.** Siam learning

Từ mô hình CNN, mô hình trả ra 2 vector mã hóa (encoding) là  $x_1$  và  $x_2$  biểu diễn cho lần lượt ảnh 1 và 2.  $x_1$  và  $x_2$  có cùng số chiều. Hàm  $f(x)$  có tác dụng tương tự như một phép biến đổi qua lớp kết nối đầy đủ (layer fully connected) trong mạng neural để tạo tính phi tuyến và giảm chiều dữ liệu về các kích thước nhỏ. Thông thường là 128 đối với các mô hình đã được các nhà khoa học huấn luyện từ trước.



**Hình 2.18.** Siam network trong bài toán nhận dạng chữ ký

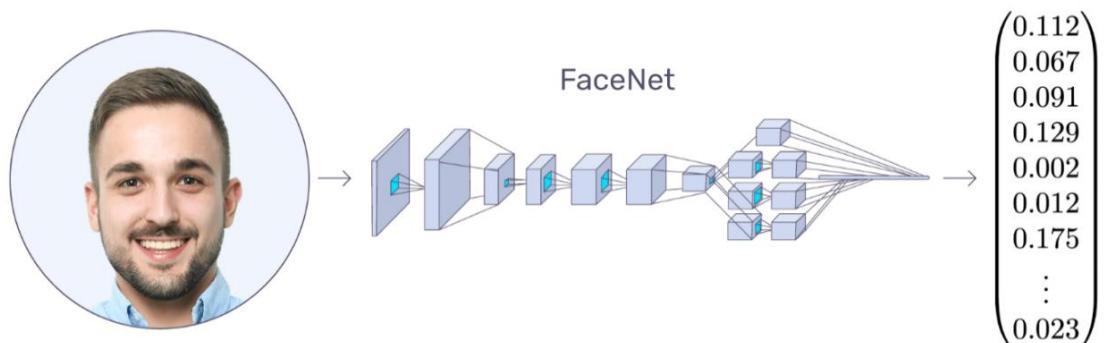
## 2.7. Mô hình học sâu được huấn luyện trước (Pre-train model)

### 2.7.1. Sử dụng mô hình được huấn luyện trước

Mô hình huấn luyện trước là một mô hình được đào tạo bởi một người khác để giải quyết một vấn đề tương tự. Thay vì xây dựng một mô hình từ đầu để giải quyết một vấn đề tương tự, ta sử dụng mô hình được đào tạo về vấn đề khác làm điểm khởi đầu. Thường thì những mô hình này là những mô hình rất lớn khó khăn trong việc huấn luyện một mô hình được đào tạo trước có thể không chính xác 100%, nhưng nó giúp tiết kiệm rất nhiều thời gian và công sức.

### 2.7.2. Giới thiệu Facenet

FaceNet[1] là một mạng lưới thần kinh sâu được sử dụng để trích xuất các tính năng từ hình ảnh của một khuôn mặt người. Nó được xuất bản vào năm 2015 bởi các nhà nghiên cứu của Google.



**Hình 2.19.** Facenet mã hóa hình ảnh khuôn mặt thành vector 128 chiều

FaceNet lấy hình ảnh của khuôn mặt người làm đầu vào và xuất ra một vector 128 chiều, đại diện cho các tính năng quan trọng nhất của khuôn mặt. Trong học máy, vector này được gọi là nhúng (embeddings). Tại sao phải nhúng? Bởi vì tất cả các thông tin quan trọng từ một hình ảnh được nhúng vào vector này. Về cơ bản, FaceNet lấy một khuôn mặt và nén nó thành một vector gồm 128 số. Khuôn mặt cần định danh cũng có nhúng tương tự.

Facenet chính là một dạng siam network có tác dụng biểu diễn các bức ảnh trong một không gian euclidean n chiều (thường là 128) sao cho khoảng cách giữa các vector nhúng(embedding) càng nhỏ, mức độ tương đồng giữa chúng càng lớn.

Hầu hết các thuật toán nhận diện khuôn mặt trước facenet đều tìm cách biểu diễn khuôn mặt bằng một vector nhúng (embedding) thông qua một layer bottle neck có tác dụng giảm chiều dữ liệu:

- Tuy nhiên hạn chế của các thuật toán này đó là số lượng chiều vector nhúng (embedding) tương đối lớn (thường  $\geq 1000$ ) và ảnh hưởng tới tốc độ của thuật toán. Thường chúng ta phải áp dụng thêm thuật toán PCA để giảm chiều dữ liệu để giảm tốc độ tính toán.
- Hàm đánh giá mất mát (loss function) chỉ đo lường khoảng cách giữa 2 bức ảnh. Như vậy trong một đầu vào huấn luyện chỉ học được một trong hai khả năng là sự giống nhau nếu chúng cùng 1 lớp hoặc sự khác nhau nếu chúng khác lớp mà không học được cùng lúc sự giống nhau và khác nhau trên cùng một lượt huấn luyện.

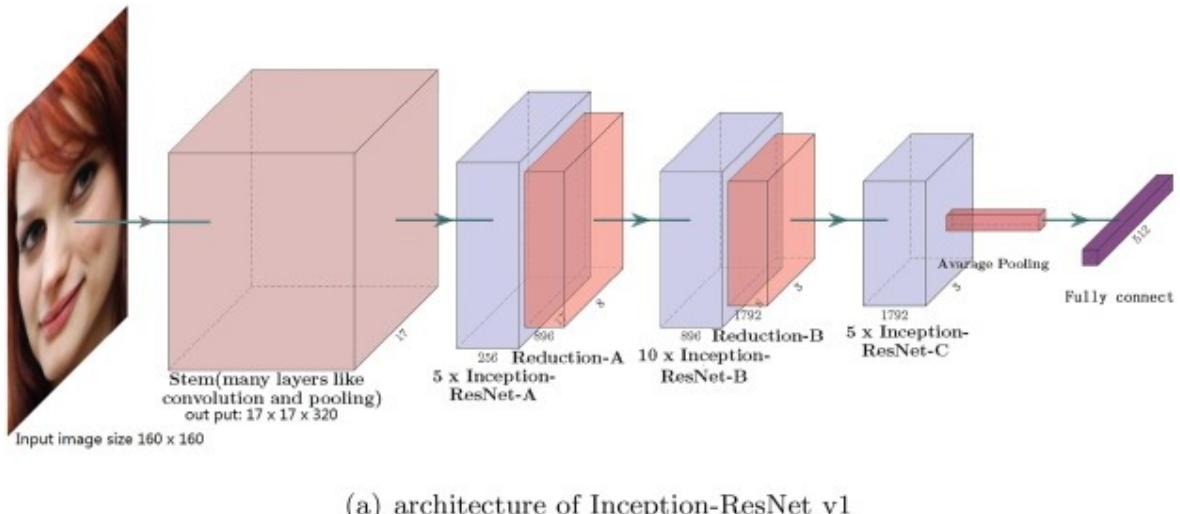
Facenet đã giải quyết cả 2 vấn đề trên bằng các hiệu chỉnh nhỏ nhưng mang lại hiệu quả lớn:

- Mạng cơ sở áp dụng một mạng neural tích chập và giảm chiều dữ liệu xuống chỉ còn 128 chiều. Do đó quá trình suy diễn và dự báo nhanh hơn và đồng thời độ chính xác vẫn được đảm bảo.
- Sử dụng hàm đánh giá mất mát là hàm đánh giá bộ ba (triplet loss) có khả năng học được đồng thời sự giống nhau giữa 2 bức ảnh cùng nhóm và phân biệt các bức ảnh không cùng nhóm. Do đó hiệu quả hơn rất nhiều so với các phương pháp trước đây.

### 2.7.3. Giới thiệu Mạng InceptionResnetV1

Mạng InceptionResnetV1[3] có là sự kết hợp giữa 2 mạng cơ sở là InceptionNet hay còn gọi là GoogLe Net và ResNet.

Nó được giới thiệu năm 2016 bởi các kĩ sư của Google, và cho thấy hiệu quả mạnh mẽ trên những tập dữ liệu ảnh lớn tiêu biểu là tập dữ liệu ImageNet.



**Hình 2.20.** Cấu trúc tổng quan của InceptionResnetV1

InceptionResnetV1 là một mạng CNN kết hợp giữa các cấu trúc lớn với khoảng 23 triệu tham số trong mạng đồng nghĩa với mỗi khi cho ảnh đi qua mạng này phải thực hiện 23 triệu phép tính, còn chưa kể trong quá trình huấn luyện mạng này phải thực hiện rất nhiều phép tính để thay đổi các tham số.

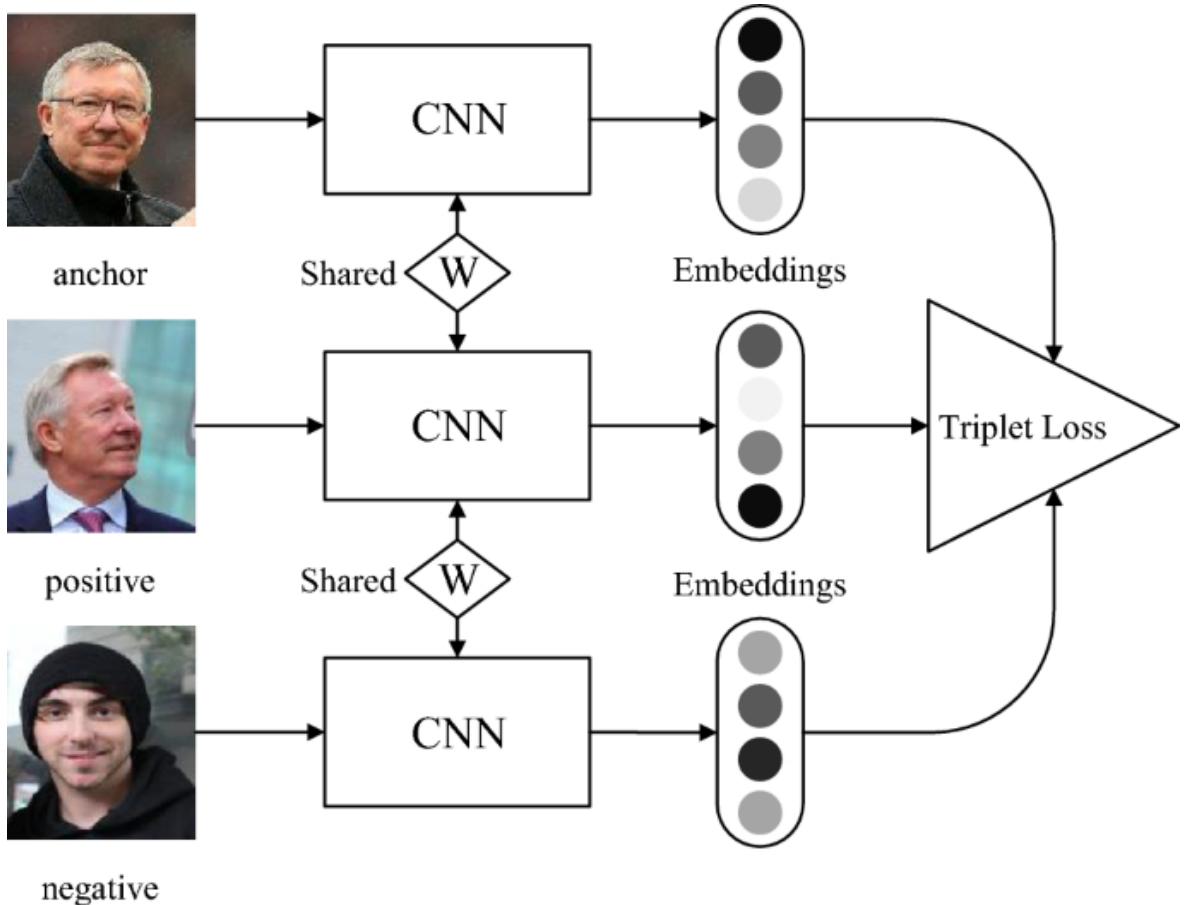
Chính vì sự khổng lồ của nó nên đã gây ra sự khó khăn trong quá trình huấn luyện, với những điều kiện như: lượng dữ liệu cho huấn luyện phải thật sự lớn (lên tới hàng chục triệu ảnh với hàng triệu khuôn mặt), phần cứng hỗ trợ tính toán tốn kém, tài nguyên lưu trữ lớn, ...

Rất may mắn rằng có các tổ chức với lợi thế về dữ liệu, tài nguyên phần cứng đã huấn luyện thành công các mạng này với những kết quả có độ chính xác cực cao.

Nên tôi đã sử dụng InceptionResnetV1 làm mạng cơ sở cho hệ thống này

#### 2.7.4. Kỹ thuật đánh giá bộ ba (Triplet loss)

Trong Facenet, quá trình mã hóa của CNN đã giúp ta mã hóa bức ảnh về 128 chiều. Sau đó những vector này sẽ làm đầu vào cho hàm đánh giá bộ ba để đánh giá khoảng cách giữa các vector.



**Hình 2.21.** Mô hình sử dụng hàm đánh giá bộ ba

Mục tiêu của triplet loss là đảm bảo rằng:

- Hai ví dụ có cùng nhãn có các phần nhúng của chúng gần nhau trong không gian nhúng.
- Hai ví dụ với các nhãn khác nhau có các nhúng của chúng ở xa nhau.

Dể áp dụng triple loss, chúng ta cần lấy ra 3 bức ảnh trong đó có một bức ảnh là anchor. Trong 3 ảnh thì ảnh anchor được cố định trước. Chúng ta sẽ lựa chọn 2 ảnh

còn lại sao cho một ảnh là negative (của một người khác với anchor) và một ảnh là positive (cùng một người với anchor).

Mục tiêu của hàm triplet loss là tối thiểu hóa khoảng cách giữa 2 ảnh khi chúng là negative và tối đa hóa khoảng cách khi chúng là positive. Như vậy chúng ta cần lựa chọn các bộ 3 ảnh sao cho:

- Ảnh Anchor và Positive khác nhau nhất: cần lựa chọn để khoảng cách  $d(A,P)$  lớn. Điều này cũng tương tự như bạn lựa chọn một ảnh của mình hồi nhỏ so với hiện tại để thuật toán học khó hơn. Nhưng nếu nhận biết được thì nó sẽ thông minh hơn.
- Ảnh Anchor và Negative giống nhau nhất: cần lựa chọn để khoảng cách  $d(A,N)$   $d(A,N)$  nhỏ. Điều này tương tự như việc thuật toán phân biệt được ảnh của một người anh em giống bạn.

$$Loss = \sum_{i=1}^N \left[ \|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha \right]_+$$

**Hình 2.22.** Công thức của hàm đánh giá bộ ba

---

***Chương 3***

---

# Thiết kế và xây dựng hệ thống

---

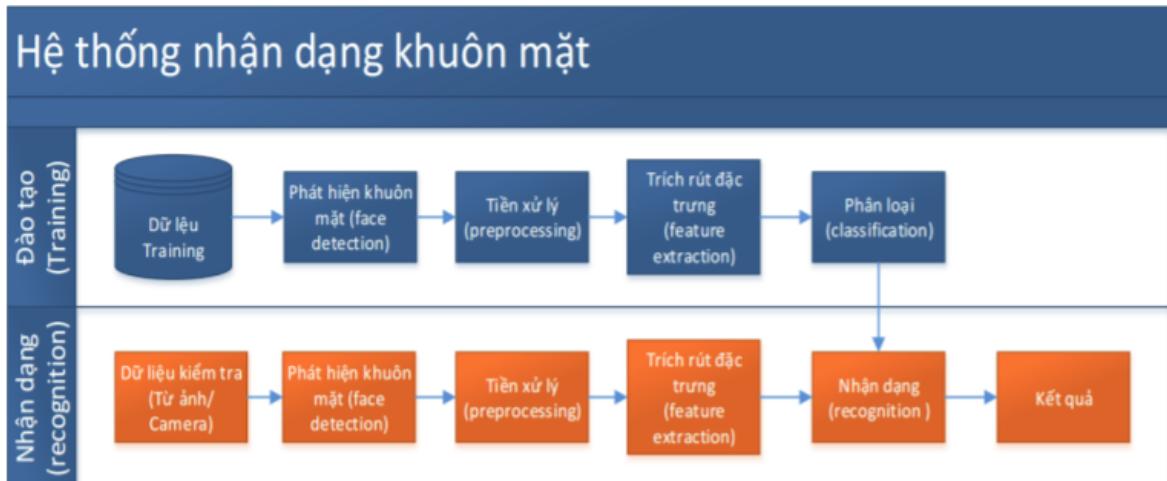
## 3.1. Phân tích

Vẽ cơ bản một hệ thống điểm danh bằng khuôn mặt gồm các bước sau:

- Thu thập dữ liệu khuôn mặt
- Phát hiện khuôn mặt dựa trên ảnh đầu vào và gán nhãn dữ liệu
- Làm giàu dữ liệu
- Trích xuất các đặc trưng (sử dụng học sâu)
- Dựa các đặc trưng đã được gán nhãn vào thuật toán phân loại
- Lưu trữ các thông tin và kết quả phân loại đã được học
- Nhận dạng khuôn mặt và tiến hành điểm danh

Hệ thống được chia thành 2 quy trình:

- Người dùng đăng ký khuôn mặt của mình
- Hệ thống tiến hành nhận dạng và điểm danh



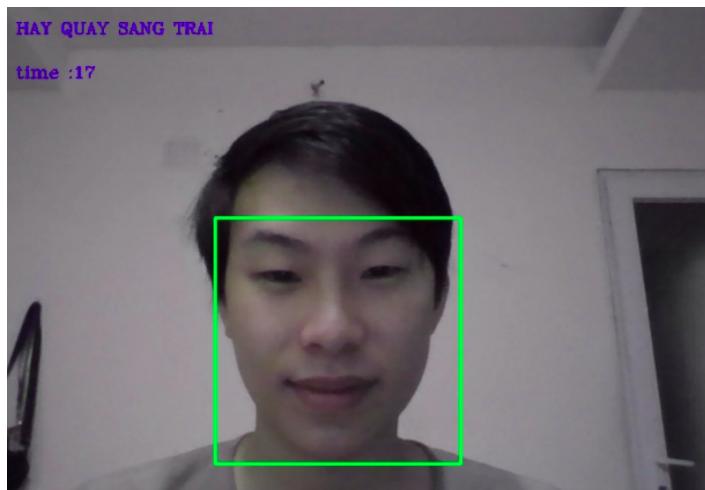
**Hình 3.1.** Tổng quan hệ thống

## 3.2. Xây dựng

### 3.2.1. Thu thập dữ liệu khuôn mặt

Hệ thống thu thập hình ảnh dữ liệu khuôn mặt bằng cách sử dụng chính webcam của máy tính, hoặc có thể là hình ảnh từ nhiều nguồn khác. Các ảnh được thu thập cần đảm bảo các yếu tố như điều kiện ánh sáng, các góc độc khác nhau của khuôn mặt, tuổi tác,... Và khuôn mặt không nên có các vật cản như kính.

Hệ thống có thể gợi ý người dùng quay mặt sang trái hay phải để có thể thu thập được nhiều góc mặt nhất có thể



**Hình 3.2.** Thu thập dữ liệu khuôn mặt người dùng

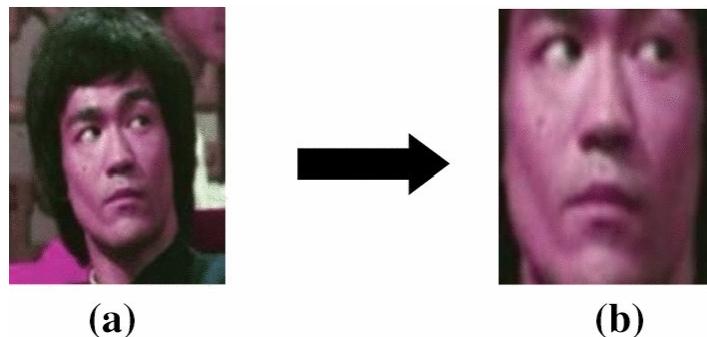
Ngoài ra, để đảm bảo độ chính xác cho hệ thống, đối với mỗi người dùng cần thu thập một số lượng ảnh nhất định không quá ít, và mỗi bức ảnh chỉ chứa duy nhất một khuôn mặt.

Bộ dữ liệu tôi sử dụng trong dự án này gồm 400 ảnh của 10 sinh viên. Với số lượng ảnh của mỗi sinh viên là khác nhau dao động từ 30 đến 40 ảnh cho mỗi sinh viên.

### 3.2.2. Phát hiện khuôn mặt và gán nhãn dữ liệu

Để trích chọn đặc trưng cho mỗi khuôn mặt, trước tiên ta cần tìm ra vị trí khuôn mặt trong bức hình. Vì bộ dữ liệu sẽ bao gồm nhiều ảnh có điều kiện ánh sáng cũng như các góc độ của khuôn mặt khác nhau, chính vì vậy việc lựa chọn máy dò khuôn mặt cũng rất quan trọng để đảm hiệu quả cao nhất cho hệ thống.

Tôi sử dụng MTCNN thực hiện công việc này và tiến hành gán nhãn dữ liệu, yêu cầu người dùng nhập tên.



**Hình 3.3.** MTCNN lấy khuôn mặt của người dùng

### 3.2.3. Làm giàu dữ liệu

Dữ liệu là nguồn sống của học máy và học sâu, nhận thấy sự quan trọng của dữ liệu trong các mô hình học sâu nên các phương pháp làm giàu dữ liệu đã ra đời và đóng góp rất nhiều vào sự thành công của các mô hình học sâu.

Ở trong dự án này tôi cũng sử dụng một số phương pháp làm giàu dữ liệu phổ biến sử dụng các thuật toán xử lý ảnh:

- Chuẩn hóa theo phân phối chuẩn các pixels của ảnh

- Tạo ảnh với các góc nghiêng 20 độ
- Dịch chuyển ảnh theo chiều dài và chiều rộng
- Lật ảnh theo chiều ngang



**Hình 3.4.** Dữ liệu đã được làm giàu

#### 3.2.4. Trích chọn các đặc trưng ảnh khuôn mặt

Trong hệ thống này tôi sử dụng 1 mô hình có sẵn với mạng cơ sở là InceptionResnetV1 được huấn luyện trong tập dữ liệu với hàng triệu ảnh khuôn mặt khác nhau trong đó có cả người Việt Nam.

$$f(\boxed{\text{[Image of Vladimir Putin]}}) = \begin{pmatrix} 0.112 \\ 0.067 \\ 0.091 \\ 0.129 \\ 0.002 \\ 0.012 \\ 0.175 \\ \vdots \\ 0.023 \end{pmatrix}$$

**Hình 3.5.** Mô tả nhúng ảnh khuôn mặt thành các vector đặc trưng

Bộ dữ liệu khuôn mặt sẽ được chia theo từng thư mục tương ứng với hình ảnh của từng đối tượng (sinh viên). Hệ thống sẽ tiến hành quét qua toàn bộ ảnh trong các thư

mục. máy dò khuôn mặt sẽ tìm kiếm khuôn mặt có trong ảnh (mặc định mỗi ảnh sẽ chỉ chứa một khuôn mặt), cắt lấy khuôn mặt và đưa kích thước về 160x160 pixel.

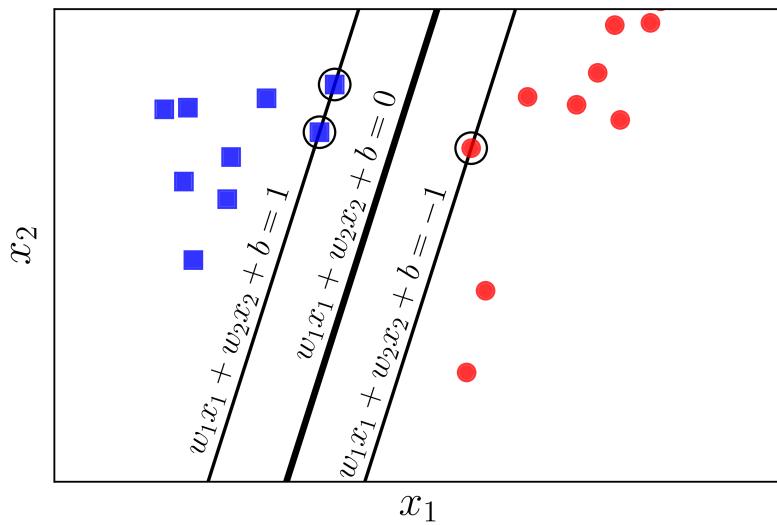
Sau đó FaceNet sẽ tiến hành trích rút đặc trưng của từng khuôn mặt, áp dụng mô hình học với thuật toán hàm đánh giá bộ ba và gắn nhãn cho từng khuôn mặt (nhãn sẽ được lấy theo tên thư mục chứa ảnh).

### 3.2.5. Đưa các vector đặc trưng vào mô hình phân loại

Sau khi đã có các vector đặc trưng của các khuôn mặt, tôi sẽ đưa các vector này vào một mô hình để thuật toán có thể học được cách phân loại các đối tượng đã đăng ký.

Mô hình thuật toán phân loại mà tôi sử dụng là thuật toán SVM (Support vector machine)

Sau khi thuật toán SVM đã được huấn luyện để phân loại xong thì tiến hành lưu kết quả huấn luyện cùng với các vector và nhãn đã đào tạo lại, nhằm mục đích huấn luyện lại mô hình khi có thêm khuôn mặt mới đăng ký.



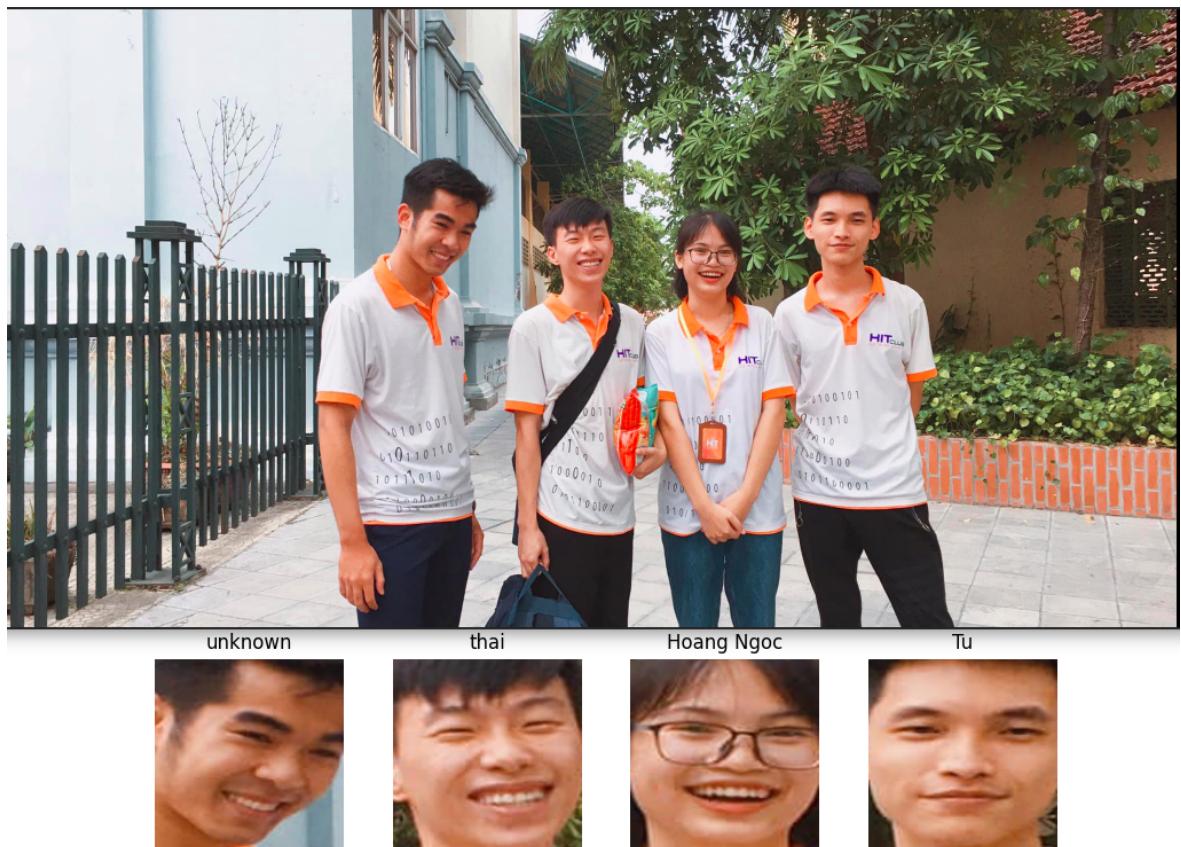
**Hình 3.6.** Mô tả cơ chế phân loại của SVM

### 3.2.6. Nhận diện khuôn mặt và tiến hành điểm danh

Khi hệ thống đã thực hiện huấn luyện xong các mô hình học sâu, tôi tiến hành thử nghiệm với một số ảnh có các khuôn mặt đã đăng ký và chưa đăng ký.

Hệ thống sẽ dò tìm các khuôn mặt trong ảnh, sau đó thực hiện việc mã hóa các khuôn mặt này thành các vector đặc trưng rồi đưa vào các mô hình phân loại đã được huấn luyện.

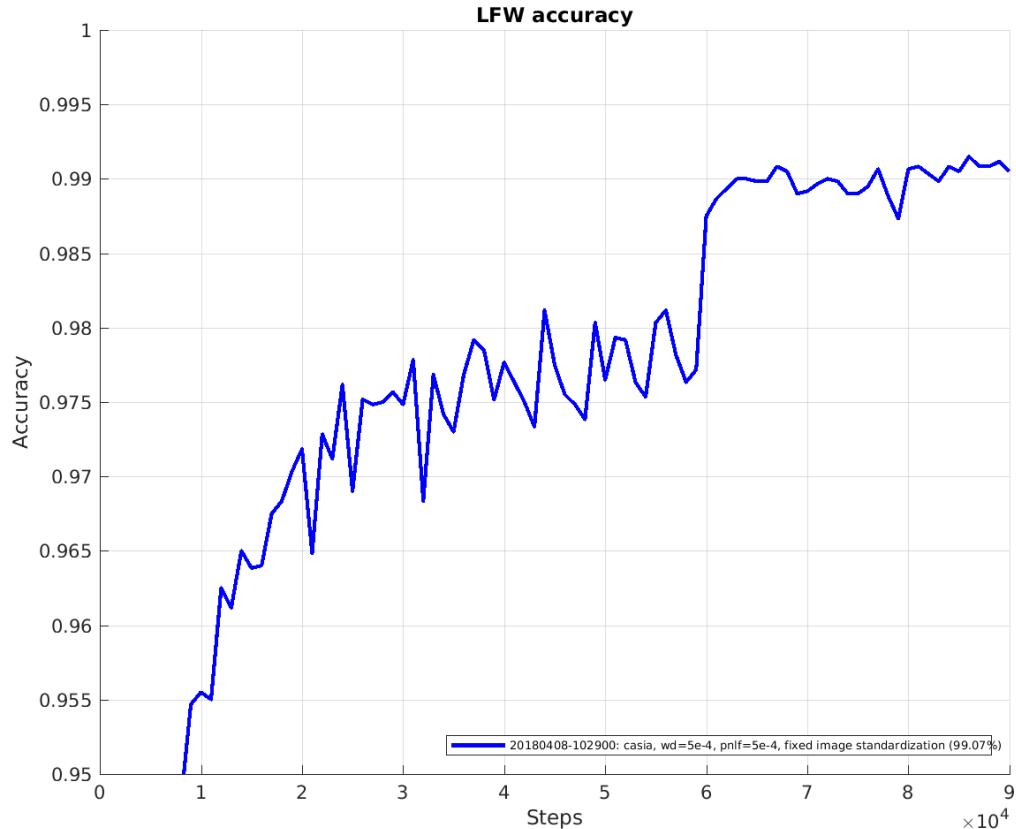
Kết quả cuối cùng hệ thống sẽ đưa ra hình ảnh các khuôn mặt và kèm theo các tên của khuôn mặt đó nếu khuôn mặt này đã được đăng ký, ngược lại hệ thống sẽ đưa ra "unknown face" nếu khuôn mặt này chưa xuất hiện trong tập dữ liệu đã đăng ký



**Hình 3.7.** Hệ thống tiến hành điểm danh

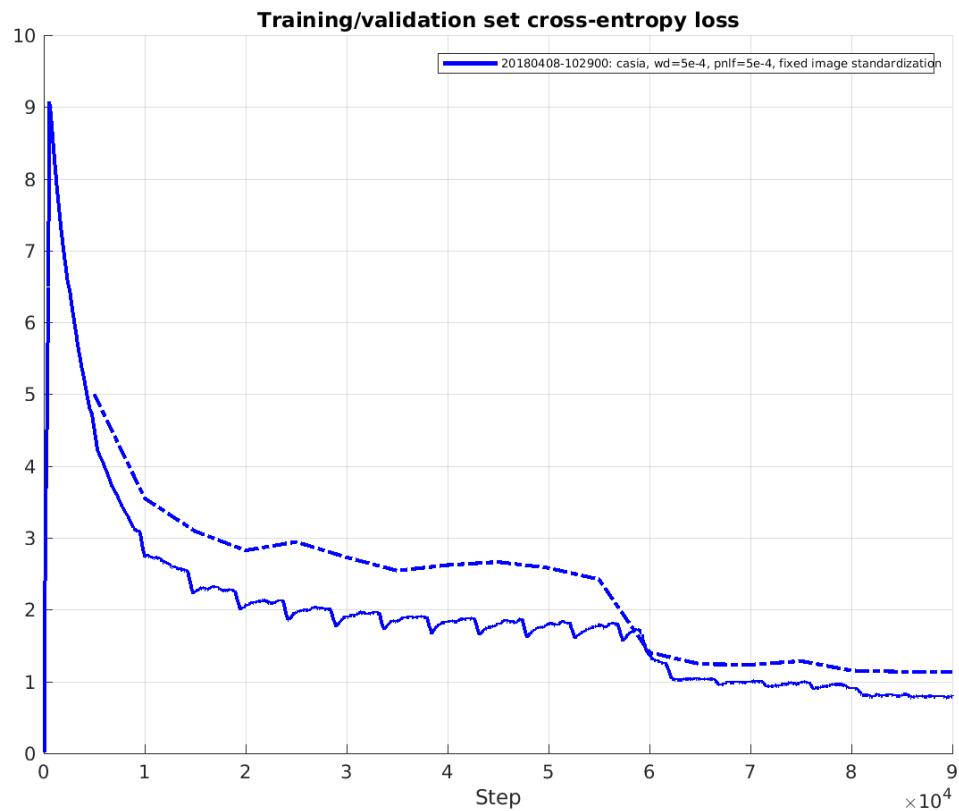
### 3.2.7. Kết quả thử nghiệm

#### *Độ chính xác của mô hình facenet*



**Hình 3.8.** Độ chính xác trên tập dữ liệu LFW

Ở đây có thể thấy rằng ngay cả khi độ chính xác trong lần đánh giá cuối cùng là 0,9965 thì độ chính xác trung bình cho 10 lần đánh giá cuối cùng thấp hơn một chút (0,9950), có lẽ gần với những gì người ta có thể mong đợi khi tái tạo kết quả. Mức trung bình có lẽ là một số liệu tốt hơn để sử dụng khi so sánh các cài đặt siêu tham số khác nhau.



**Hình 3.9.** Đánh giá mất mát bằng Entropy chéo

Hình này cho thấy sự mất mát entropy chéo trong quá trình huấn luyện (đường liền nét) và xác nhận (đường đứt nét). Bộ xác nhận bao gồm khoảng 20000 hình ảnh và đánh giá được thực hiện sau mỗi 5 ký nguyên. Entropy chéo trong quá trình đào tạo được ghi lại ở mỗi bước đào tạo nhưng đã được lọc bằng bộ lọc trung bình trượt trên 500 bước. Từ điều này, rõ ràng là mô hình có lợi hơn một chút vì vậy có thể tốt nếu thử nghiệm thêm một chút với xác suất phân rã trọng lượng L2 và xác suất bỏ học.

### ***Hiệu suất của hệ thống***

Khi sử dụng các phương pháp đo độ chính xác trên dữ liệu thực tế mà tôi tự thu thập từ những người bạn, sinh viên trong khoa, trường. Thì độ chính xác đo được là 96.4%



**Hình 3.10.** Dữ liệu tự thu thập

---

**Chương 4**

---

# Kết luận và hướng phát triển

---

## 4.1. Kết luận

Trên cơ sở tìm hiểu về bài toán nhận diện mặt người trong ảnh, sử dụng pre-trained model FaceNet được huấn luyện trước trên mạng cơ sở là InceptionResnetV1 do tiến sĩ khoa học máy tính David Sandberg cung cấp, tôi đã xây dựng thành công hệ thống điểm danh thông qua hình ảnh khuôn mặt.

Về khả năng phát hiện khuôn mặt, kết quả phát hiện khá tốt hầu hết các trường hợp, kể cả trong điều kiện thiếu sáng, góc nghiêng, hay có vật che khuất như kính mắt,...

Về khả năng nhận dạng, hệ thống đạt kết quả từ 96-98% đối với các khuôn mặt thẳng và điều kiện ánh sáng thích hợp, đạt 92-95% đối với các khuôn mặt nghiêng hoặc thiếu sáng.

Về khả năng loại trừ các khuôn mặt “unknown face”, kết quả đạt khoảng 85-90% khuôn mặt lạ được phát hiện trong quá trình thử nghiệm. Hệ thống điểm danh hoạt động ổn định và mượt mà nhờ máy chủ viết bằng Python. Giao diện được xây dựng trên nền Web là một lợi thế vì tính đơn giản và tiện lợi.

## 4.2. Hướng phát triển

Dựa trên những cơ sở sẵn có này hệ thống có thể được cải tiến trong tương lai bằng những phương pháp sau:

- Cải thiện thời gian chạy của hệ thống, nâng cấp lên có thể chạy trong thời gian thực
- Để cải thiện độ chính xác cho hệ thống, đầu tiên ta cần cải thiện bộ dữ liệu dựa trên các tiêu chí như tư thế chụp, góc chụp, hạn chế sự che khuất các bộ phận trên mặt, biểu cảm khuôn mặt, điều kiện ánh sáng, tuổi tác...
- Thử nghiệm với nhiều mô hình được huấn luyện trước và thuật toán huấn luyện khác nhau cho bộ dữ liệu của hệ thống.
- Thay thế phương pháp loại bỏ khuôn mặt lạ, thử nghiệm và chọn ra ngưỡng cho phép phù hợp hơn.

Không chỉ dừng lại ở việc điểm danh, của hệ thống nhận dạng khuôn mặt có thể được sử dụng trong các hệ thống mở khóa, thanh toán, hay truy tìm tội phạm,...

## Tài liệu tham khảo

- [1] D. K. Florian Schroff and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.
- [2] R. J.Hassan and A. M. Abdulazeez. Deep Learning Convolutional Neural Network for Face Recognition: A Review. *International Journal of Science and Business*, 5(2):114–127, 2021.
- [3] X. Qi and L. Zhang. Face recognition via centralized coordinate learning. *China Computing*, 01 2018.
- [4] R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localization of objects in images. In *IEE Conference Publication*, pages 26–29. Publ by IEE, 1993. 3rd International Conference on Artificial Neural Networks ; Conference date: 25-05-1993 Through 27-05-1993.
- [5] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.
- [6] X. Wang, K. Wang, and S. Lian. A survey on face data augmentation. *CoRR*, abs/1904.11685, 2019.
- [7] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.