

**ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC**



**MẠNG NỐ RƠM TÍCH CHẬP VÀ ỨNG DỤNG  
TRONG BÀI TOÁN SO KHỐP KHUÔN MẶT**

**ĐỒ ÁN I**

Chuyên ngành: TOÁN TIN

Chuyên sâu: Thị giác máy tính

Giảng viên hướng dẫn: TS. NGUYỄN TRUNG ĐŨNG

Sinh viên thực hiện: NGUYỄN HOÀNG SƠN

Lớp: CTTN Toán Tin - K65

HÀ NỘI – 2023

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

### 1. Mục tiêu và nội dung của đồ án

(a) Mục tiêu: .....

.....  
.....  
.....

(b) Nội dung: .....

.....  
.....  
.....

### 2. Kết quả đạt được: .....

.....  
.....  
.....  
.....  
.....

### 3. Ý thức làm việc của sinh viên: .....

.....  
.....  
.....  
.....  
.....

Hà Nội, ngày tháng năm 2023

Giảng viên hướng dẫn

TS. Nguyễn Trung Dũng

## Lời cảm ơn

Em xin phép được bày tỏ lòng kính trọng và lời cảm ơn chân thành đến thầy giáo, TS. Nguyễn Trung Dũng – Viện Toán ứng dụng và Tin học – Đại học Bách khoa Hà Nội, đã định hướng đề tài, tận tình chỉ bảo cho em trong suốt quá trình thực hiện đồ án I - bài toán so khớp khuôn mặt.

Bên cạnh đó, em xin chân thành cảm ơn TS. Lê Chí Ngọc, người đã truyền đạt cho em những kiến thức và kinh nghiệm quý báu trong lĩnh vực xử lý ảnh và máy học. Nhờ sự chỉ dẫn tận tình và những lời góp ý hữu ích của TS. Lê Chí Ngọc, em đã có cơ hội học hỏi và áp dụng những phương pháp mới nhất vào bài toán so khớp khuôn mặt và có thể cải thiện hiệu suất cũng như độ chính xác của mô hình.

Đồng thời em cũng xin gửi lời cảm ơn và tri ân sâu sắc đến các tác giả đã cung cấp dữ liệu quý báu giúp em có thể thực hiện đề tài này. Sự đóng góp của mọi người là rất quan trọng, giúp em hoàn thành báo cáo một cách chính xác và kết quả tích cực.

Lời cuối cùng, em xin bày tỏ lòng biết ơn chân thành đến Viện Toán ứng dụng và Tin học, đã cho em một môi trường học tập và nghiên cứu tốt.

Em xin trân trọng cảm ơn!

Hà Nội, ngày tháng năm 2023

Tác giả đồ án

Nguyễn Hoàng Sơn

# Mục lục

<b>Bảng ký hiệu và chữ viết tắt</b>	<b>1</b>
<b>Danh sách bảng</b>	<b>2</b>
<b>Danh sách hình vẽ</b>	<b>3</b>
<b>Mở đầu</b>	<b>5</b>
<b>Chương 1. Cơ sở lý thuyết</b>	<b>6</b>
1.1 Mạng nơ ron tích chập . . . . .	6
1.2 Thuật toán Stochastic Gradient Descent (SGD) . . . . .	12
<b>Chương 2. Bài toán so khớp khuôn mặt</b>	<b>15</b>
2.1 Tổng quan bài toán . . . . .	15
2.2 Các hướng tiếp cận bài toán . . . . .	16
2.2.1 Phương pháp so khớp toàn diện (Holistic Matching) . . . . .	16
2.2.2 Phương pháp so khớp dựa trên đặc trưng (Feature-based)	17
2.2.3 Phương pháp dựa trên mô hình mạng nơ ron tích chập . . . . .	17
2.3 Một số mô hình mạng nơ ron tích chập phổ biến . . . . .	18
2.3.1 Mô hình FaceNet . . . . .	18
2.3.2 Mô hình Xception . . . . .	19
<b>Chương 3. Dữ liệu, mô hình và kết quả</b>	<b>23</b>
3.1 Dữ liệu và các chỉ số đánh giá . . . . .	23
3.1.1 Dữ liệu sử dụng . . . . .	23

3.1.2	Chỉ số đánh giá hiệu quả của mô hình . . . . .	24
3.2	Triển khai mô hình và kết quả thực nghiệm . . . . .	26
3.2.1	Triển khai mô hình . . . . .	26
3.2.2	Kết quả thực nghiệm . . . . .	28
<b>Kết luận</b>		<b>32</b>
<b>Chỉ mục</b>		<b>34</b>
<b>Tài liệu tham khảo</b>		<b>35</b>

# Bảng ký hiệu và chữ viết tắt

CNN	mạng nơ ron tích chập
$x \in D$	$x$ thuộc tập $D$
$\mathbb{R}^n$	không gian Euclide $n$
$\forall x$	với mọi $x$
$\ x\ _2$	chuẩn Euclide của $x$
$\nabla_{\theta}(J)$	đạo hàm của hàm $J$ theo biến $\theta$
$\text{argmax}(f)$	tập nghiệm tối ưu của bài toán giá trị lớn nhất hàm $f$
$\text{argmin}(f)$	tập nghiệm tối ưu của bài toán giá trị nhỏ nhất hàm $f$

# Danh sách bảng

3.1	Bảng so sánh chỉ số đánh giá mô hình . . . . .	29
-----	--	----

# Danh sách hình vẽ

1.1	Kiến trúc mạng nơ ron [1]	7
1.2	Minh họa tích chập [2]	8
1.3	Minh họa stride [2]	8
1.4	Ví dụ về kernel [3]	9
1.5	Minh họa Activation map [2]	9
1.6	Phương pháp zero-padding [2]	10
1.7	Max pooling và Average pooling với stride là 2 [2]	11
1.8	Minh họa Dropout [4]	12
1.9	Mô hình CNN [5]	12
2.1	Quy trình nhận diện khuôn mặt [6]	16
2.2	Mô tả quá trình học của mô hình [7]	19
2.3	Mô tả tích chập thông thường [8]	20
2.4	Mô tả Depthwise Convolution [8]	20
2.5	Mô tả Pointwise Convolution [8]	21
2.6	Mô hình Xception [9]	22
3.1	Ảnh Trung tướng Phạm Tuân trong tập dữ liệu VN-Celeb [10]	23
3.2	Ví dụ về bộ ba (Anchor, Positive, Negative)	24
3.3	Mô hình trích xuất vector đặc trưng	27
3.4	Mô hình chung	28
3.5	Biểu đồ hàm mất mát và độ chính xác trên tập kiểm tra	28

3.6 Biểu đồ tương quan phân bố khoảng cách (Anchor, Positive) và (Anchor, Negative) . . . . .	29
3.7 Ma trận nhầm lẫn giữa Xception (trái) và Xception đã huấn luyện trên tập dữ liệu (phải) . . . . .	29
3.8 Ví dụ phân loại sai . . . . .	30

# Mở đầu

Khi xã hội phát triển, nhu cầu sử dụng các ứng dụng liên quan đến công nghệ thông tin cũng ngày càng tăng cao. Trong đó, bài toán so khớp khuôn mặt là một trong những lĩnh vực nghiên cứu được đặc biệt quan tâm bởi tính ứng dụng rộng rãi của nó. Bài toán so khớp khuôn mặt được ứng dụng trong nhiều lĩnh vực khác nhau như an ninh, giám sát, xác thực người dùng, hệ thống tìm kiếm thông tin dựa trên ảnh, phân loại ảnh và nhiều ứng dụng khác.

Bằng cách sử dụng các phương pháp xử lý ảnh và máy học, bài toán so khớp khuôn mặt đã đạt được những thành tựu đáng kể trong thời gian gần đây. Tuy nhiên, việc tiếp tục nghiên cứu và phát triển các phương pháp mới để cải thiện độ chính xác và tốc độ xử lý là rất cần thiết.

Trong báo cáo này, em đã thực hiện xây dựng một mô hình so khớp mặt dựa trên mô hình được huấn luyện trước là mô hình “Xception [9]”, được huấn luyện trên bộ dữ liệu người nổi tiếng Việt Nam [10].

Nội dung đồ án được chia thành 3 chương như sau:

- Chương 1 - Cơ sở lý thuyết:** Chương này sẽ giới thiệu sơ lược về mạng nơ-ron tích chập và thuật toán tối ưu Stochastic Gradient Descent được sử dụng phổ biến trong huấn luyện mạng.
- Chương 2 - Bài toán so khớp khuôn mặt:** Chương này sẽ đi sâu vào bài toán so khớp khuôn mặt, giới thiệu các hướng tiếp cận để giải quyết bài toán, quy trình của mô hình so khớp khuôn mặt. Cuối cùng là đề xuất một số mô hình mạng nơ ron tích chập để giải quyết bài toán.
- Chương 3 - Ứng dụng xây dựng mô hình:** Chương này sẽ mô tả và cách xử lý dữ liệu, cách cài đặt chương trình và kết quả đạt được.

# Chương 1

## Cơ sở lý thuyết

### 1.1 Mạng nơ ron tích chập

#### Lịch sử phát triển

Mạng nơ ron tích chập (Convolution neural Network - CNN) là một thành tựu quan trọng trong lĩnh vực trí tuệ nhân tạo nói chung và xử lý ảnh nói riêng.

CNN hiện đại đầu tiên được giới thiệu vào những năm 1990, dựa vào các nghiên cứu sinh học trước đó về sự nhận diện hình ảnh của con người. Một trong số đó có thể kể đến là mô hình **LeNet-5** [11], được áp dụng trong việc nhận diện chữ số viết tay và đạt độ chính xác vượt trội.

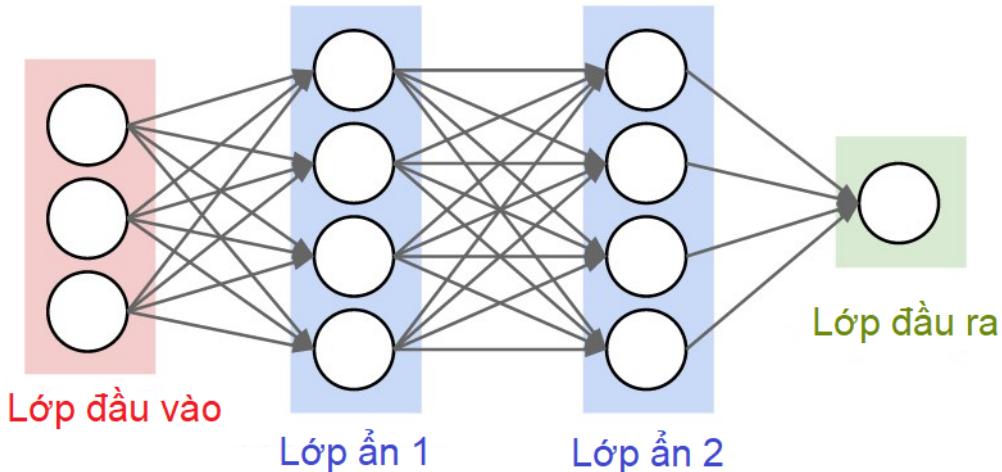
Việc nghiên cứu và phát triển CNN vẫn luôn được duy trì. Cùng với sự phát triển mạnh mẽ của phần cứng thì độ chính xác của CNN cũng ngày càng được cải thiện.

Vào năm 2012, mô hình **AlexNet** [12] đã được giới thiệu trong cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) và giành chiến thắng với cách biệt không tưởng. **AlexNet** chứng minh được sự ưu việt của CNN trong việc nhận diện ảnh và đánh dấu bước đầu của sự phát triển mạnh mẽ của CNN.

Từ đó cho đến nay, rất nhiều mô hình CNN đã được giới thiệu cải thiện cả về hiệu năng và độ chính xác. Một số mô hình có thể kể đến như: **GoogLenet** [13], **VGGNet** [14], ...

## Tổng quan kiến trúc

Trong bài toán phân loại ảnh, khi thực hiện bằng mạng nơ ron thông thường thì số lượng tham số sẽ vô cùng lớn cùng với việc không giữ được những tính không gian trong ảnh.



Hình 1.1: Kiến trúc mạng nơ ron [1]

Giả sử ta muốn xây dựng một mô hình so khớp khuôn mặt được huấn luyện trên tập dữ liệu VN-Celeb [10] thì mỗi ảnh sẽ có kích thước là  $128 \times 128 \times 3$ . Một nơ ron trong lớp ẩn đầu tiên sẽ mang lên đến  $128 \times 128 \times 3 = 49152$  tham số. Tất nhiên, đây dường như là một con số nhỏ nhưng chúng ta cần rất nhiều nơ ron như vậy trong mạng, cùng với việc kích thước ảnh tăng thì số lượng tham số cần thiết trong mạng là vô cùng lớn !

Kiến trúc CNN mang đến cho chúng ta một cách tiếp cận bài toán tự nhiên hơn.

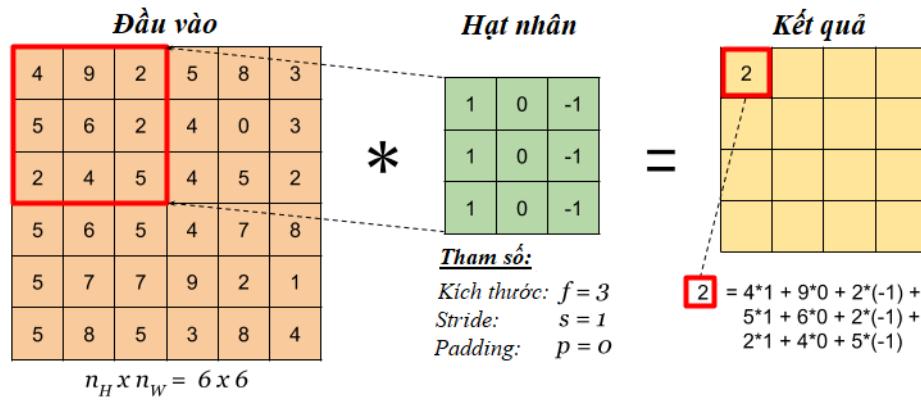
Với đầu vào là ảnh có kích thước  $128 \times 128 \times 3$ , chúng ta sẽ áp dụng từng bộ lọc vào ảnh với các tham số được học trong quá trình huấn luyện để đạt được hiệu quả cao nhất. Số lượng tham số sẽ giảm đáng kể so với các tiếp cận thông thường trên. Chi tiết sẽ được trình bày trong các mục nhỏ dưới đây.

## Lớp tích chập

Lớp tích chập là lớp quan trọng nhất trong một mô hình CNN. Lớp này đóng vai trò giúp trích xuất những đặc trưng trong một bức ảnh. Với đầu vào là ma trận ảnh I, ta sẽ tiến hành nhân tích chập với một ma trận hạt nhân (kernel - filter) K.

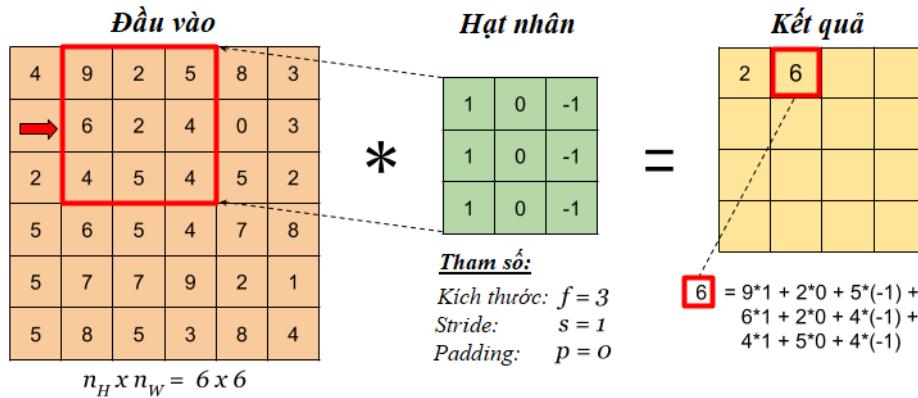
$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

Phép tích chập có thể được minh họa bằng hình sau.



Hình 1.2: Minh họa tích chập [2]

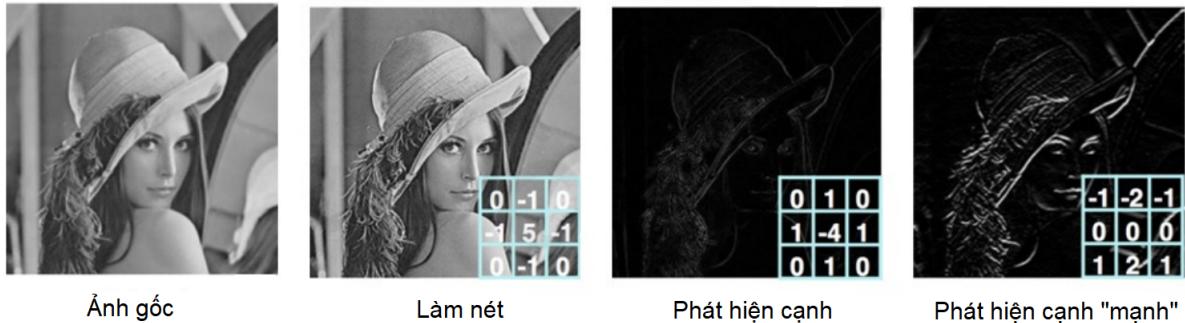
Stride là sai khác giữa hai vùng liên tiếp áp dụng phép nhân vô hướng (từ trái sang phải, từ trên xuống dưới). Với stride là 1 thì phần tử tiếp theo của ma trận S được tính như sau.



Hình 1.3: Minh họa stride [2]

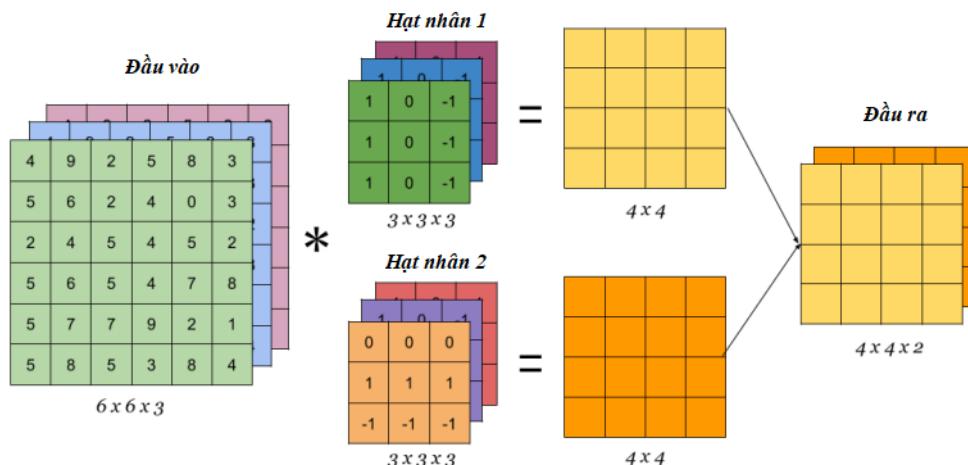
Đầu ra thu được từ mỗi phép tích chập với hạt nhân K được gọi là feature

map. Mỗi hạt nhân K sẽ có từng vai trò riêng, giúp làm nổi bật những đặc trưng của ảnh như: cạnh, góc, kết cấu,...



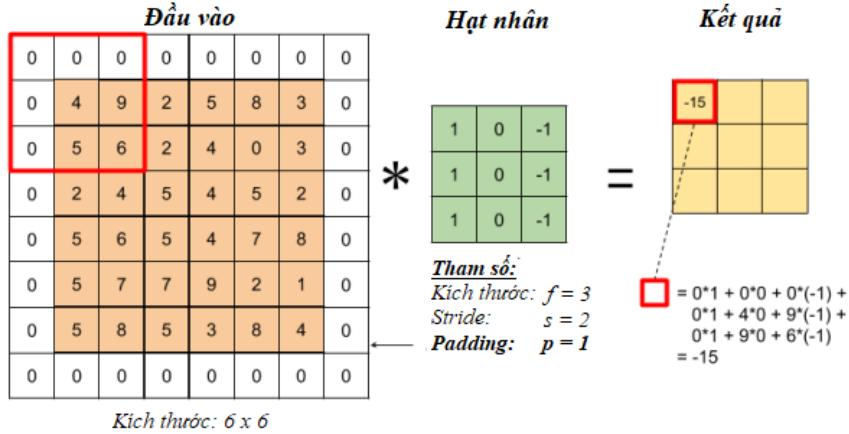
Hình 1.4: Ví dụ về kernel [3]

Khi ta áp dụng nhiều hạt nhân trong cùng một lớp thì các feature map sẽ được xếp chồng lên nhau tạo thành activation map là output của mỗi lớp tích chập.



Hình 1.5: Minh họa Activation map [2]

Khi áp dụng phép tích chập liên tục như vậy sẽ khiến kích thước của ảnh nhỏ dần, gây khó khăn khi áp dụng tiếp các phép tích chập ở lớp sau. Do đó, tùy thuộc vào stride mà ta sẽ thêm một lượng hàng và cột vào lề của ma trận để nó không thay đổi kích thước sau mỗi phép tích chập, gọi là padding. Phổ biến nhất là zero-padding.



Hình 1.6: Phương pháp zero-padding [2]

Sau mỗi lớp, ta sẽ áp dụng một hàm kích hoạt để chuẩn hoá dữ liệu. Thường dùng là hàm ReLU.

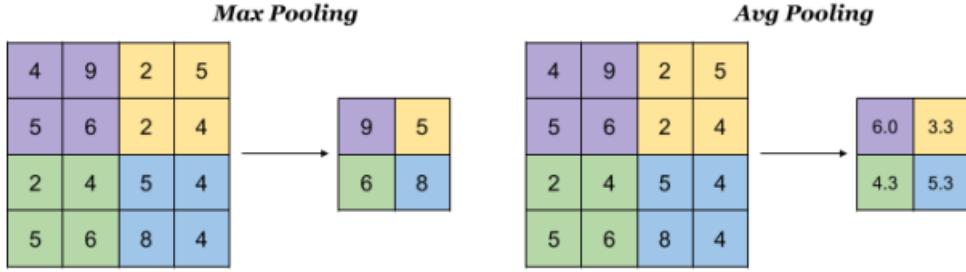
$$ReLU(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

## Lớp pooling

Lớp Pooling (Pooling layer) thường được xếp và giữa các lớp convolution trong mô hình CNN. Chức năng của lớp này dùng để giảm dần kích thước của ma trận, giữ lại các thông tin quan trọng, từ đó giảm số lượng tham số, thời gian tính toán cũng như tránh được việc overfitting. Tất nhiên, sẽ có sự mất mát thông tin nhưng ở mức độ chấp nhận được.

Có 2 kiểu pooling phổ biến là: Max pooling và Average pooling. Max pooling thường được nằm ở giữa các lớp convolution để giảm dần kích thước tham số. Trong khi đó, average pooling thường được sử dụng làm lớp cuối cùng của mô hình.

Với bức ảnh cỡ  $n \times n$ , Pooling với hạt nhân kích thước  $2 \times 2$  với stride là 2 sẽ thu được ảnh có kích thước  $n/2 \times n/2$ .



Hình 1.7: Max pooling và Average pooling với stride là 2 [2]

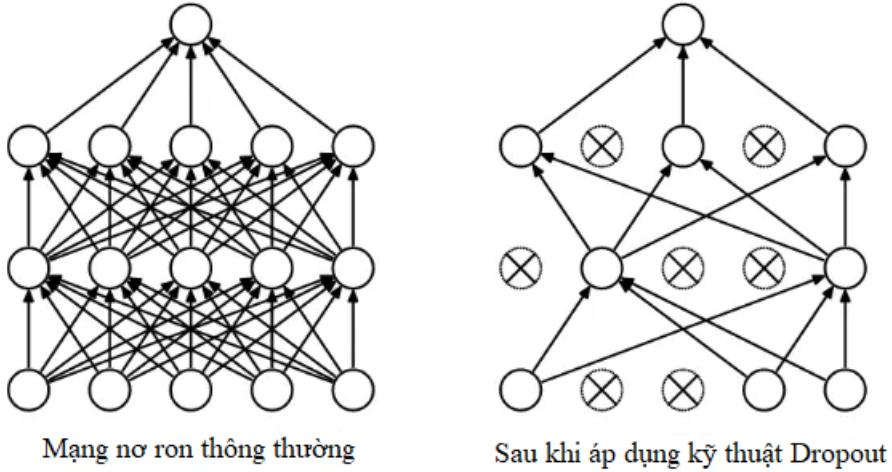
## Lớp fully-connected

Lớp này có chức năng nhận các feature map ở tầng trước và đưa ra một vector chứa các xác suất của các đối tượng cần dự đoán. Để thực hiện điều này, ta sẽ trải dài ma trận đầu vào thành các vector và đưa vào mạng nơ ron thông thường. Số chiều của vector đầu ra tương ứng với số lớp mà ta cần phân lớp. Output của lớp fully-connected trong bài toán phân loại thường có thêm lớp softmax để đảm bảo kết quả của các output nơ rons có tổng bằng 1.

$$\sigma_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

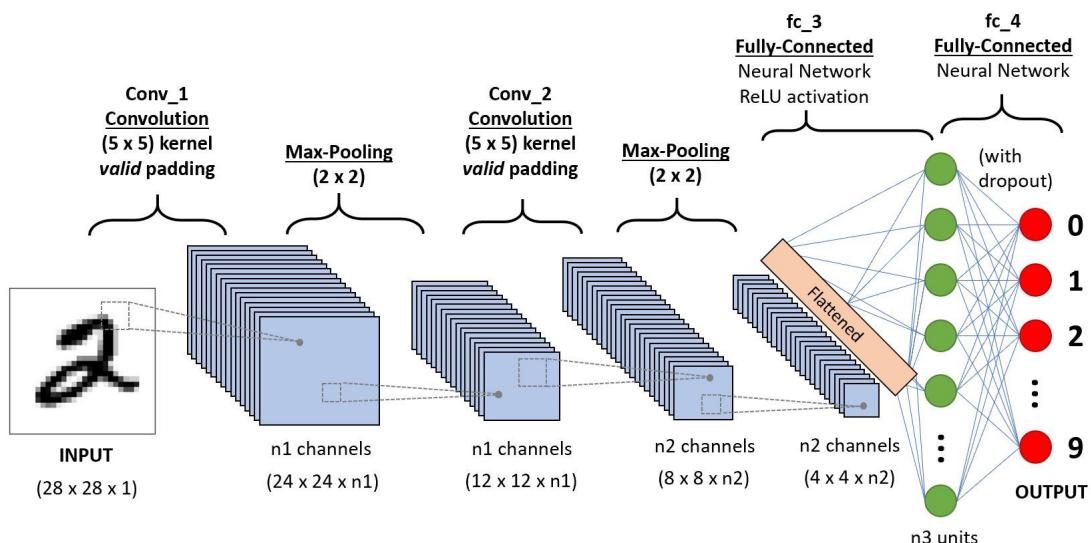
## Kỹ thuật Dropout

Dropout là một dạng kỹ thuật giúp ngăn chặn hiện tượng overfitting. Overfitting là hiện tượng chúng ta huấn luyện mô hình rất tốt trong tập huấn luyện nhưng quá trình dự đoán lại đạt kết quả kém với những dữ liệu mới. Có rất nhiều nguyên nhân dẫn đến hiện tượng này, tiêu biểu như: mô hình quá phức tạp, dữ liệu huấn luyện chưa đủ,... Dropout ngăn chặn hiện tượng bằng cách: ở mỗi lần huấn luyện, với xác suất là  $p$ , sẽ ngẫu nhiên bỏ đi các kết nối giữa các nơ ron từ lớp trước đến lớp tiếp theo trong kiến trúc mạng. Trong khi huấn luyện, ta cần thử nghiệm để có thể chọn được tham số  $p$  phù hợp với mạng.



Hình 1.8: Minh họa Dropout [4]

Sau đây là minh họa cho một mô hình CNN đơn giản.



Hình 1.9: Mô hình CNN [5]

## 1.2 Thuật toán Stochastic Gradient Descent (SGD)

Đầu tiên, ta cùng nói qua về thuật toán Gradient Descent thuần tuý. Để tìm nghiệm tối ưu của bài toán với hàm mất mát  $J(\theta)$  với  $\theta$  là bộ trọng số của mô hình, ta tiến hành các bước sau:

1. Khởi tạo  $\theta = \theta_0$ .

2. Cập nhật  $\theta$  đến khi đạt kết quả chấp nhận được:

$$\theta = \theta - \eta \nabla_{\theta} J(\theta, x, y) \quad (1.1)$$

trong đó

- $\nabla_{\theta} J(\theta, x, y)$ : đạo hàm hàm mất mát theo  $\theta$  tại các điểm  $(x, y)$ .
- $\eta$ : tốc độ học.

Tuy nhiên, trong nhiều trường hợp với thuật toán Gradient Descent có thể bị mắc kẹt tại các điểm tối ưu cục bộ mà không thể tiến tới các điểm tối ưu toàn cục. Do đó, chúng ta cần cung cấp một bước "đà" để có thể "vượt qua" điểm tối ưu cục bộ. Gradient Descent với momentum được phát triển để giải quyết vấn đề trên.

Trong thuật toán Gradient Descent, chúng ta cần tính lượng thay đổi ở thời điểm  $t$  để cập nhật  $\theta$ . Nếu chúng ta coi đại lượng này như vận tốc  $v_t$  trong một đơn vị thời gian thì sẽ có công thức cập nhật  $\theta$  là  $\theta_{t+1} = \theta_t - v_t$ . Ý tưởng của Gradient Descent với momentum sẽ là thêm vào đại lượng  $v_t$  một "đà" dựa vào  $v_{t-1}$  để có thể "vượt qua" điểm tối ưu cục bộ, tức là ta sẽ có:

$$v_t = \gamma v_{t-1} + \eta \nabla_{\theta} J(\theta, x, y) \quad (1.2)$$

Trong đó,  $\gamma$  được gọi là momentum. Ta sẽ có công thức cập nhật  $\theta$  như sau:

$$\theta = \theta - v_t. \quad (1.3)$$

Trong thuật toán trên, mỗi lần cập nhật  $\theta$  ta cần phải tính toán giá trị đạo hàm tại tất cả các điểm  $(x_i, y_i)$  trong tập dữ liệu. Điều này sẽ trở nên cồng kềnh và kém hiệu quả nếu tập dữ liệu của ta lớn. Khi này, thuật toán Stochastic Gradient Descent (SGD) tỏ ra đơn giản và rất hiệu quả.

Trong SGD, tại một thời điểm, ta chỉ tính đạo hàm của hàm mất mát  $J(\theta, x, y)$  trên một lô nhỏ dữ liệu (batch)  $(x_i, y_i)$  rồi cập nhật  $\theta$ .

Ta có các bước của thuật toán SGD như sau:

1. Khởi tạo  $\theta = \theta_0$

2. Chia dữ liệu thành từng lô.
3. Trong mỗi lô, ta cập nhật  $\theta$  dựa vào công thức Gradient Descent 1.3.
4. Kết thúc một vòng lặp (epoch) khi tất cả các lô được duyệt.

Trong các chương tiếp theo sẽ trình bày chi tiết cách áp dụng mô hình mạng và thuật toán trên để giải quyết bài toán so khớp khuôn mặt.

## Chương 2

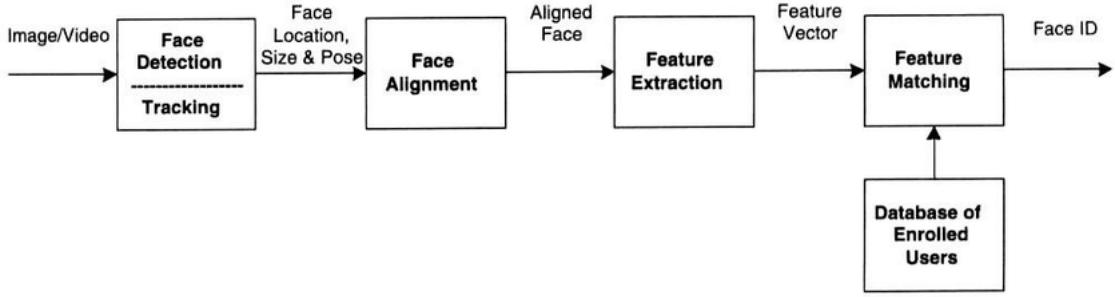
# Bài toán so khớp khuôn mặt

### 2.1 Tổng quan bài toán

So khớp khuôn mặt, tức là so sánh sự giống khác nhau giữa một khuôn mặt với một khuôn mặt khác, là một bài toán mà mỗi chúng ta đối diện hàng ngày. Đó có thể là khi ta muốn xác minh danh tính của một người xem họ là ai, quê quán ở đâu; người này có phải là người quen của ta,... Tất nhiên, con người chúng ta có thể dễ dàng giải quyết những vấn đề trên bằng bộ não của mình kể cả khi họ có một vài thay đổi nhỏ so với trước đây. Nhưng với số lượng mẫu cần so khớp lớn thì việc ta dùng sức người tỏ ra không hiệu quả. Chính vì vậy, việc làm sao để "dạy" cho máy so khớp tự động khuôn mặt người đã được các nhà nghiên cứu quan tâm.

Những nghiên cứu về so khớp khuôn mặt đã được thực hiện từ những năm 1960. Những năm gần đây đã chứng kiến sự tiến bộ đáng kể trong lĩnh vực này. Những mô hình so khớp với tốc độ cùng độ chính xác cao đã được phát minh và triển khai.

Về cơ bản, bài toán nhận diện khuôn mặt nói chung có thể được giải quyết qua các quy trình sau:



Hình 2.1: Quy trình nhận diện khuôn mặt [6]

Trong đó:

- Face Detection: phát hiện khuôn mặt người nằm ở vị trí nào trong bức ảnh và trích xuất ra một ảnh riêng.
- Face Alignment: Từ ảnh trên, ta sẽ áp dụng một số phép chuẩn hoá về màu sắc, ảnh sáng, độ tương phản, góc chụp theo một tỉ lệ nhất định giúp tăng tốc độ và độ chính xác cho toàn bộ mô hình.
- Face Extraction: Sau đó, ta sẽ vector hoá bức ảnh trên bằng một số phương pháp để thu được vector đặc trưng biểu diễn cho mỗi ảnh.
- Face Matching: Từ các vector đặc trưng trên, ta sẽ tiến hành phân loại trên tập dữ liệu để thu được kết quả của bài toán.

Trong đồ án này sẽ tập trung giải quyết hai quy trình là Face Extraction và Face Matching.

## 2.2 Các hướng tiếp cận bài toán

### 2.2.1 Phương pháp so khớp toàn diện (Holistic Matching)

Mục tiêu của phương pháp so khớp toàn diện là so sánh toàn bộ khuôn mặt thay vì chỉ tập trung vào các đặc trưng của khuôn mặt. Ví dụ tiêu biểu là EigenFace [15].

Ý tưởng của phương pháp EigenFace là từ một tập dữ liệu ảnh ban đầu và ảnh cần so khớp, ta sẽ tiến hành giảm chiều dữ liệu của ảnh và sử dụng vector

biểu diễn bức ảnh trong không gian con này như một vector đặc trưng. Sau đó, ta có thể áp dụng các thuật toán phân loại vào tập các vector đặc trưng để giải bài toán.

Với việc xem xét toàn bộ khuôn mặt, phương pháp này có thể nhận ra các thay đổi về hình dạng, tỷ lệ, biểu cảm và ánh sáng. Điều này giúp nâng cao độ chính xác và độ tin cậy của quá trình so khớp khuôn mặt. Tuy nhiên, việc tính toán trên toàn bộ khuôn mặt cũng có thể tốn kém về mặt tính toán so với các phương pháp chỉ tập trung vào các đặc trưng cụ thể.

### **2.2.2 Phương pháp so khớp dựa trên đặc trưng (Feature-based)**

Phương pháp so khớp dựa trên đặc trưng dựa trên việc phát hiện những đặc trưng được coi là quan trọng khi cần nhận diện một người (mắt, mũi, miệng, lông mày, hình dạng khuôn mặt) và tính toán các giá trị tương quan giữa các đặc trưng trên (góc, diện tích, khoảng cách). Các giá trị tương quan sẽ được phân cấp và được biểu diễn dưới dạng vector và áp dụng các thuật toán phân loại để giải quyết bài toán. Ta dễ dàng nhận ra được rằng lớp phương pháp này có điểm yếu chí mạng là phụ thuộc rất lớn vào các điều kiện ngoại cảnh. Chỉ cần thay đổi một vài yếu tố ngoại cảnh sẽ dẫn đến một dự đoán sai lầm !

### **2.2.3 Phương pháp dựa trên mô hình mạng nơ ron tích chập**

Phương pháp sử dụng mô hình CNN được coi là một bước đột phá trong các phương pháp giải quyết bài toán so khớp khuôn mặt. Với khả năng mạnh mẽ trong việc học và nhận dạng, CNN đã đạt được nhiều thành tựu trong các cuộc thi về so khớp khuôn mặt. CNN có khả năng tự động học và trích xuất các đặc trưng quan trọng từ hình ảnh, bao gồm cạnh, góc, đường cong và các chi tiết khác trong khuôn mặt.

Quá trình giải quyết bài toán so khớp khuôn mặt sử dụng CNN thường bao gồm việc xây dựng một kiến trúc CNN phù hợp và huấn luyện nó trên một lượng lớn dữ liệu khuôn mặt được gắn nhãn. Các lớp tích chập của CNN sẽ tự động

học các bộ lọc để trích xuất các đặc trưng quan trọng từ ảnh khuôn mặt, trong khi các lớp fully connected sẽ thực hiện quá trình phân loại và so khớp.

Ưu điểm của phương pháp sử dụng CNN trong bài toán so khớp khuôn mặt là khả năng học và nhận dạng các đặc trưng phức tạp của khuôn mặt, cũng như khả năng tổng quát hóa trên các dữ liệu mới. Ngoài ra, với sự phát triển của công nghệ GPU, việc huấn luyện và triển khai mạng CNN cũng trở nên nhanh chóng và hiệu quả hơn.

## 2.3 Một số mô hình mạng nơ ron tích chập phổ biến

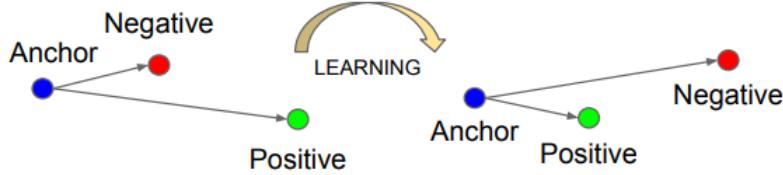
### 2.3.1 Mô hình FaceNet

Với những mô hình trước đây, hàm mất mát của chúng chỉ đo lường khoảng cách giữa hai bức ảnh. Do vậy, với mỗi đầu vào thì mạng chỉ học được một trong hai khả năng là sự giống nhau hoặc sự khác nhau giữa hai đầu vào.

FaceNet [7] là một mô hình CNN được giới thiệu năm 2015 đã giải quyết được vấn đề trên. Mô hình sử dụng một hàm mất mát mang tên Triplet Loss học được đồng thời sự giống nhau và khác nhau giữa các bức ảnh trong mỗi đầu vào. Mục tiêu của mô hình là tạo ra các vector đặc trưng đủ tốt để khiến khoảng cách giữa hai ảnh cùng một lớp lớn hơn khoảng cách giữa ảnh đó với ảnh trong lớp khác.

Gọi  $f(x) \in \mathbb{R}^d$  là đầu ra của mô hình với  $x$  là ảnh đầu vào,  $d$  là số chiều của vector đặc trưng của ảnh.

Chọn một ảnh làm mốc (anchor), ký hiệu  $x_i^a$ . Từ mốc ta sẽ xác định được các ảnh cùng lớp với mốc (positive)  $x_i^p$  và các ảnh khác lớp với mốc (negative)  $x_i^n$ . Mô hình kỳ vọng trong quá trình học ta sẽ giảm  $\|f(x_i^a) - f(x_i^p)\|_2$  và tăng  $\|f(x_i^a) - f(x_i^n)\|_2$  giúp mô hình phân loại tốt hơn. Mục tiêu chung là giảm thiểu các trường hợp mô hình nhận diện sai ảnh Negative thành Postive nhất có thể.



Hình 2.2: Mô tả quá trình học của mô hình [7]

Hay nói cách khác, chúng ta cần

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

$$\forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in T$$

Trong đó  $\alpha$  là một lề (margin) ta thêm vào giúp hai lớp được tách biệt nhau hơn,  $T$  là tập các (Anchor, Positive, Negative) gồm  $N$  phần tử.

Do đó, ta thu được hàm mất mát của mô hình có dạng

$$L = \sum_i^N \max(\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha, 0) \quad (2.1)$$

Đây được gọi là hàm mất mát TripletLoss.

Trong thực tế, để tăng tốc độ hội tụ và kết quả dự đoán của mô hình thì ta thường chọn các bộ ba (Anchor, Positive, Negative) sao cho

$$x_*^p = \operatorname{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$$

$$x_*^n = \operatorname{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$$

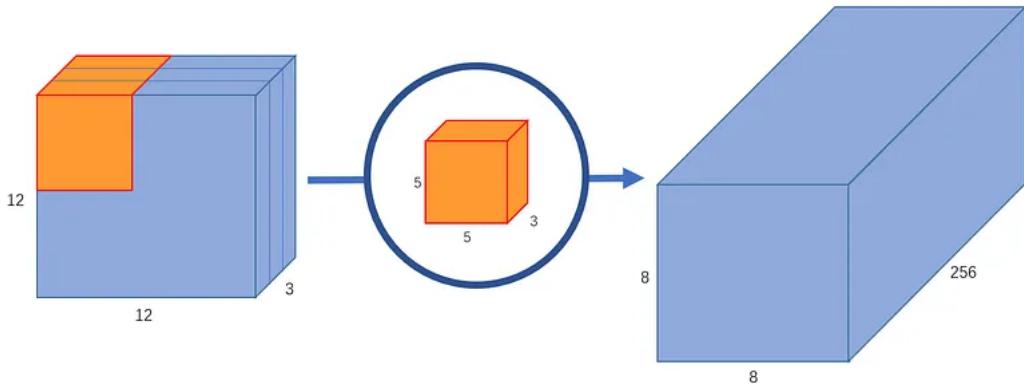
Điều này thường không khả thi khi ta tính trên toàn bộ tập dữ liệu. Do vậy, ta có thể tính argmax và argmin của bộ ba trên từng lô dữ liệu trong mỗi lần huấn luyện.

### 2.3.2 Mô hình Xception

Mô hình Xception [9] là một mô hình CNN được giới thiệu năm 2017 đã đạt được hiệu suất ấn tượng và trở thành một lựa chọn phổ biến trong cộng đồng xử lý ảnh.

Thay vì việc sử dụng các phép tích chập truyền thống, Xception sử dụng phép tích chập phân rời (Separable convolutions) giúp giảm đáng số lượng tham số và lượng tính toán trong quá trình huấn luyện mà vẫn giữ được hiệu quả cao.

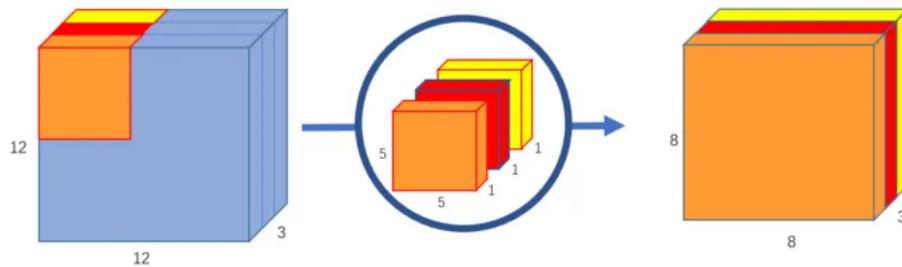
Trong một bài toán với bức ảnh đầu vào có độ phân giải  $12 \times 12 \times 3$ . Thông thường, khi ta áp dụng ta muốn thu được feature map với kích thước  $8 \times 8 \times 256$  thì ta cần phải áp dụng 256 bộ hạt nhân  $5 \times 5 \times 3$  với bước nhảy 1.



Hình 2.3: Mô tả tích chập thông thường [8]

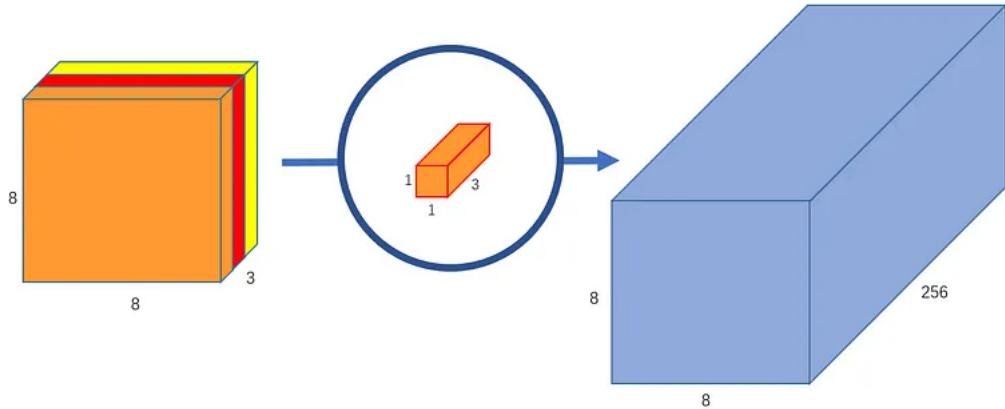
Còn với tích chập phân rời thì quá trình sẽ được chia làm hai giai đoạn:

1. Depthwise Convolution: Chúng ta sẽ thực hiện tích chập trên ảnh mà không làm thay đổi số lượng kênh của ảnh, bằng cách sử dụng 3 hạt nhân  $5 \times 5 \times 1$ .



Hình 2.4: Mô tả Depthwise Convolution [8]

2. Pointwise Convolution: Kết quả từ Depthwise Convolution sẽ được áp dụng 258 bộ hạt nhân  $1 \times 1 \times 3$  để thu được feature map theo yêu cầu.



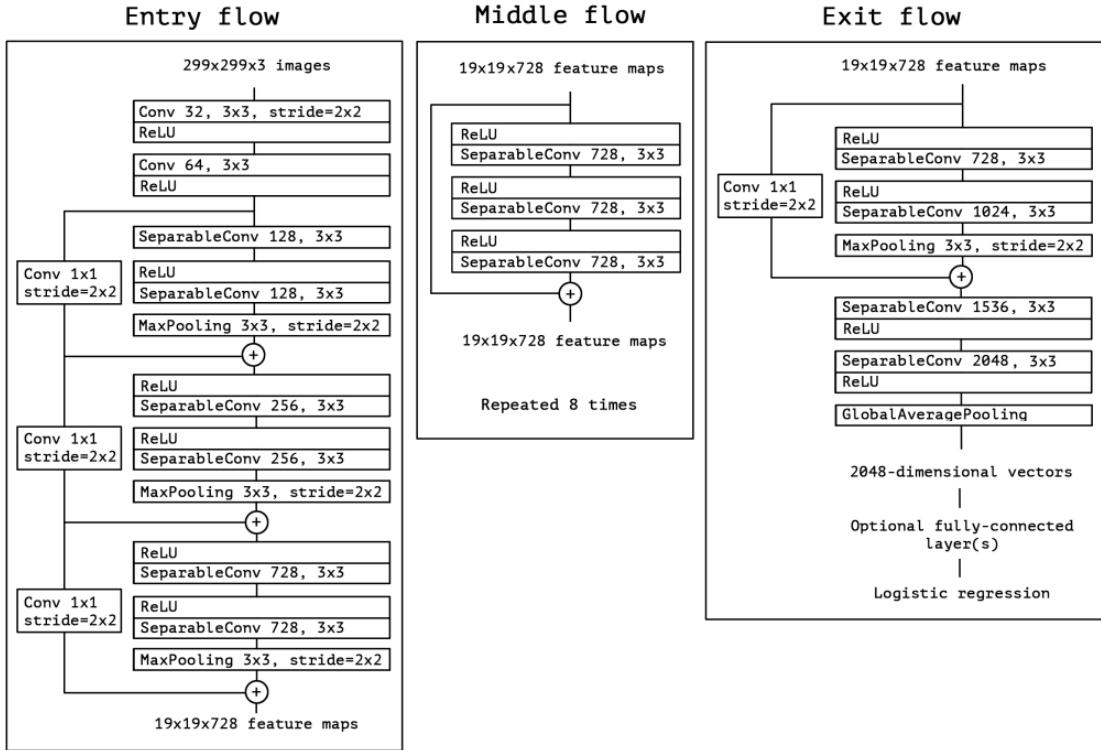
Hình 2.5: Mô tả Pointwise Convolution [8]

Trong ví dụ trên, với tích chập thông thường, ta đã thực hiện 256 bộ hạt nhân  $5 \times 5 \times 3$  dịch chuyển  $8 \times 8$  lần trên ảnh đầu vào, nghĩa là  $256 \times 3 \times 5 \times 5 \times 8 \times 8 = 1,228,800$  phép nhân.

Còn với tích chập phân rời, trong phần Depthwise convolution, chúng ta có 3 hạt nhân  $5 \times 5 \times 1$  dịch chuyển  $8 \times 8$  lần ảnh đầu vào, nghĩa là  $3 \times 5 \times 5 \times 1 \times 8 \times 8 = 4,800$  phép nhân. Còn pointwise convolution, chúng ta có 256 hạt nhân  $1 \times 1 \times 3$  dịch chuyển  $8 \times 8$  lần thì sẽ tạo  $256 \times 1 \times 1 \times 3 \times 8 \times 8 = 49,152$  phép nhân. Tổng hai quá trình lại chúng ta mất 53,952 phép nhân tất cả.

Như vậy, ta có thể thấy tích chập phân rời đã giảm đáng kể lượng tính toán cho chúng ta.

Mô hình Xception là sự kết hợp của các lớp tích chập thông thường, lớp tích chập phân rời và các kết nối tắt. Kết nối tắt giúp ta bỏ qua các lớp tích chập trung gian trong, góp phần xây dựng được mạng rất sâu mà vẫn duy trì được độ hiệu quả.



Hình 2.6: Mô hình Xception [9]

Chương tiếp theo sẽ trình bày cách kết hợp giữa mô hình Xception và mô hình FaceNet tạo thành một mô hình hoàn chỉnh giúp giải quyết bài toán so khớp khuôn mặt.

## Chương 3

# Dữ liệu, mô hình và kết quả

### 3.1 Dữ liệu và các chỉ số đánh giá

#### 3.1.1 Dữ liệu sử dụng

Trong phạm vi đồ án, bộ dữ liệu được sử dụng cho mô hình là bộ dữ liệu VN-Celeb [10]. Đây là bộ dữ liệu gồm 23105 ảnh của 1020 người có mặt trên Wikipedia Việt Nam. Trung bình một người có gần 20 ảnh, cá biệt có vài trường hợp chỉ có 2 ảnh.

Ảnh được thu thập của cùng một người có thể trong các hoàn cảnh khác nhau, có ảnh lúc trẻ có ảnh lúc già, có ảnh đen trắng có ảnh màu.



Hình 3.1: Ảnh Trung tướng Phạm Tuân trong tập dữ liệu VN-Celeb [10]

Dữ liệu đã được khử nhiễu và trích xuất ra khuôn mặt của từng người với kích thước cố định  $128 \times 128$  pixel và định dạng ".PNG".

Sau khi chuẩn hoá dữ liệu từng ảnh, ta tiến hành ghép dữ liệu thành từng cặp bộ ba ngẫu nhiên (Anchor, Positive, Negative) trong đó:

- Anchor: Một khuôn mặt của một người
- Positive: Một khuôn mặt khác của cùng một người so với Anchor
- Negative: Một khuôn mặt của người khác so với Anchor



Hình 3.2: Ví dụ về bộ ba (Anchor, Positive, Negative)

Do giới hạn về phần cứng nên khi huấn luyện ta chỉ lấy tập dữ liệu huấn luyện gồm 3500 bộ ba và tập kiểm tra gồm 800 bộ.

### 3.1.2 Chỉ số đánh giá hiệu quả của mô hình

Trong đồ án này, các chỉ số đánh giá hiệu quả của mô hình được sử dụng bao gồm:

- **Ma trận nhầm lẫn (Confusion Matrix):** Với đầu vào là hai bức ảnh cần so khớp  $x_a$  và  $x_b$ , sau khi qua mạng CNN sẽ thu được hai vector  $f(x_a)$  và  $f(x_b)$ . Với một giá trị ngưỡng  $\varepsilon$ , nhãn của bài toán sẽ được quy ước thực tế như sau:

- Nhãn 0: Hai khuôn mặt của cùng một người nếu  $\|f(x_a) - f(x_b)\|_2 \leq \varepsilon$ .
- Nhãn 1: Hai khuôn mặt không cùng một người nếu  $\|f(x_a) - f(x_b)\|_2 > \varepsilon$ .

Một ma trận nhầm lẫn bao gồm các giá trị TP, FP, FN và TN. Trong đó:

- True Positive (TP): Số trường hợp hai khuôn mặt cùng một người và được phát hiện chính xác là cùng một người.
- False Positive (FP): Số trường hợp hai khuôn mặt không cùng một người nhưng được phát hiện là cùng một người.
- True Negative (TN): Số trường hợp hai khuôn mặt không cùng là một người và được phát hiện là hai người khác nhau.
- False Negative (FN): Số trường hợp hai khuôn mặt là cùng một người nhưng lại được phát hiện là hai người khác nhau.

- **Độ chính xác - Accuracy:** Tỉ lệ số lượng cặp ảnh mà mô hình dự đoán đúng trên tổng số lượng cặp ảnh, được tính bằng công thức:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

- **Precision:** Tỉ lệ số lượng cặp ảnh mà mô hình dự đoán là có cùng khuôn mặt đúng trên tổng số lượng cặp ảnh mà mô hình dự đoán là có cùng khuôn mặt, được tính bằng công thức:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision càng cao thì mô hình phân loại càng chính xác trong việc phân loại các mẫu vào lớp positive.

- **Recall:** Tỉ lệ số lượng cặp ảnh mà mô hình dự đoán là có cùng khuôn mặt đúng trên tổng số lượng cặp ảnh thực tế là có cùng khuôn mặt, được tính bằng công thức:

$$Recall = \frac{TP}{TP + FN}$$

Recall càng cao thì mô hình phân loại càng chính xác trong việc phân loại các mẫu vào lớp positive.

Khi Precision = 1, mọi điểm tìm được đều thực sự là positive, tức không có điểm negative nào lẩn vào kết quả. Tuy nhiên, Precision = 1 không đảm bảo mô hình là tốt, vì câu hỏi đặt ra là liệu mô hình đã tìm được tất cả các điểm positive hay chưa. Nếu một mô hình chỉ tìm được đúng một điểm positive mà nó chắc chắn nhất thì ta không thể gọi nó là một mô hình tốt.

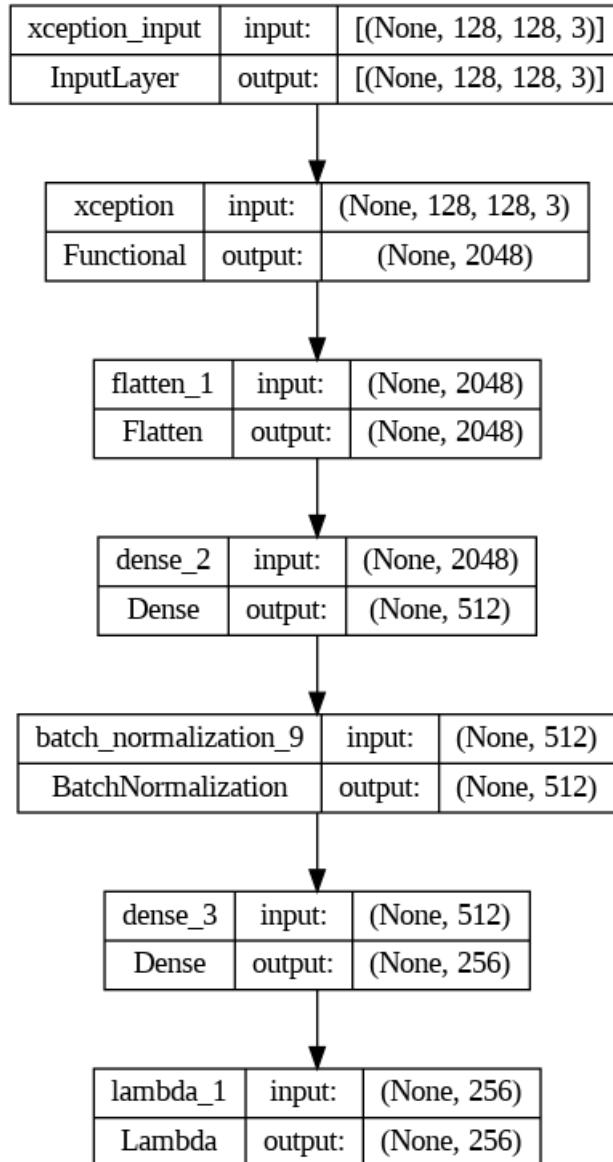
Khi Recall = 1, mọi điểm positive đều được tìm thấy. Tuy nhiên, đại lượng này lại không đo liệu có bao nhiêu điểm negative bị lẩn trong đó. Nếu mô hình phân loại mọi điểm là positive thì chắc chắn Recall = 1, tuy nhiên dễ nhận ra đây là một mô hình cực tồi.

Một mô hình phân lớp tốt là mô hình có cả Precision và Recall đều cao, tức càng gần một càng tốt.

## 3.2 Triển khai mô hình và kết quả thực nghiệm

### 3.2.1 Triển khai mô hình

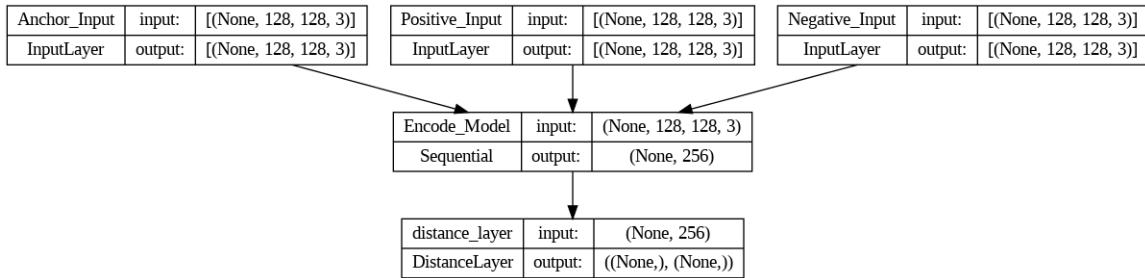
Mô hình được lựa chọn triển khai là sự kết hợp của mô hình FaceNet và Xception đã được trình bày ở chương trước. Để có thể đạt kết quả tốt trên tập dữ liệu của mình và giảm bớt được chí phí cũng như thời gian huấn luyện của mô hình thì giải pháp được chọn là huấn luyện tiếp mô hình pretrained Xception trích xuất vector đặc trưng.



Hình 3.3: Mô hình trích xuất vector đặc trưng

Cấu trúc mô hình trích xuất vector đặc trưng được biểu diễn qua hình trên. Trong đó, lớp BatchNormalization đóng vai trò chuẩn hóa dữ liệu theo lô giúp ổn định phân bố của các ảnh đầu vào; lớp Lambda đóng vai trò chuẩn hóa độ dài của từng vector đặc trưng.

Mô hình chung được xây dựng với ba đầu vào là các bộ ba (Anchor, Positive, Negative) và hai đầu ra là khoảng cách giữa các vector đặc trưng của các cặp (Anchor, Positive) và (Anchor, Negative). Hàm mất mát được lựa chọn là TripletLoss.



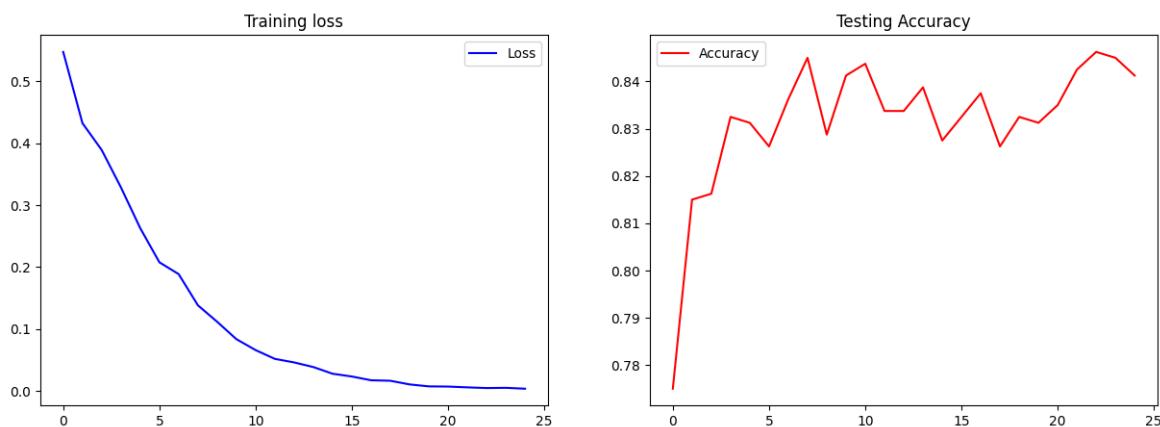
Hình 3.4: Mô hình chung

### 3.2.2 Kết quả thực nghiệm

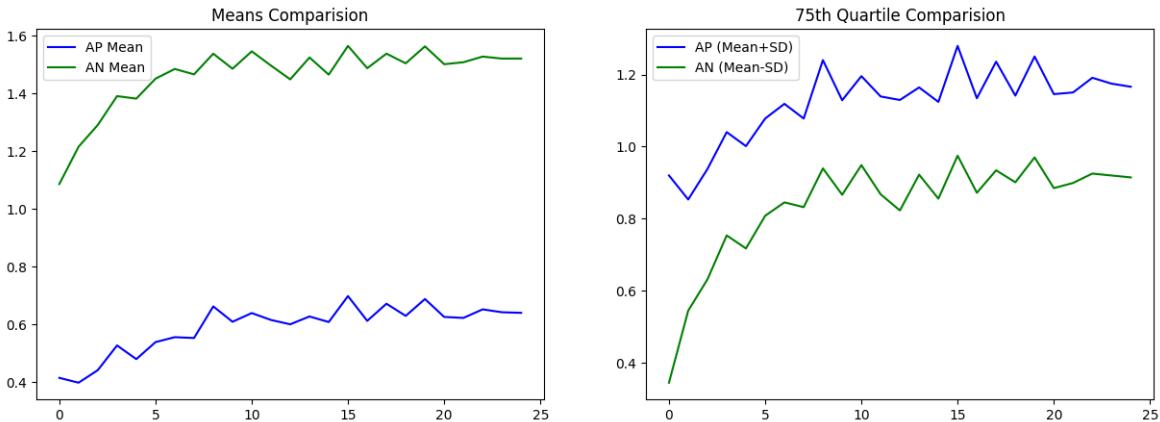
Thông số trong quá trình huấn luyện được cài đặt:

- Mô hình được xây dựng dựa trên thư viện TensorFlow và huấn luyện trên phần cứng của Google Colab: RAM 12.7Gb, GPU T4 15Gb, Disk 78.2Gb.
- Hàm mất mát: TripletLoss với margin = 1.
- Hàm tối ưu: SGD với learning\_rate =  $10^{-3}$ , momentum = 0.9.
- Batch\_size: 64.
- Epochs: 25.

Qua quá trình huấn luyện, thu được một số kết quả như sau.



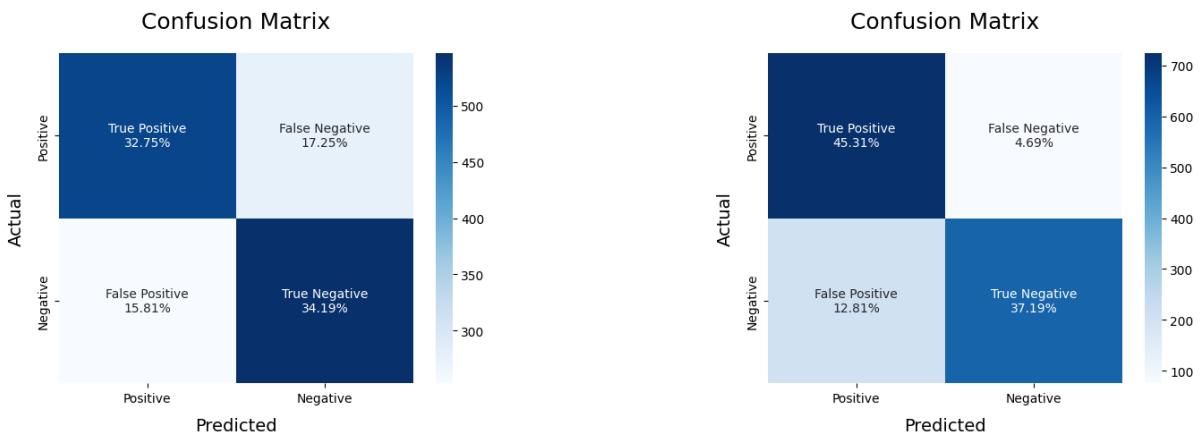
Hình 3.5: Biểu đồ hàm mất mát và độ chính xác trên tập kiểm tra



Hình 3.6: Biểu đồ tương quan phân bố khoảng cách (Anchor, Positive) và (Anchor, Negative)

Ta thấy được hàm măt măt giảm có xu hướng giảm khá nhanh và đã dần đạt đến giá trị hội tụ. Nhìn chung khoảng cách (Anchor, Positive) (AP) đã tách biệt khá rõ ràng so với khoảng cách (Anchor, Negative) (AN). Điều đó chứng tỏ mô hình đã đạt được hiệu quả khá tốt qua quá trình huấn luyện.

Để có thể đánh giá một cách rõ ràng hơn, ta tiến hành so sánh mô hình đã được huấn luyện trên tập dữ liệu mới với mô hình gốc qua các chỉ số đánh giá.



Hình 3.7: Ma trận nhầm lẫn giữa Xception (trái) và Xception đã huấn luyện trên tập dữ liệu (phải)

	Accuracy	Precision	Recall
Xception	0.6694	0.6646	0.6837
Xception đã huấn luyện trên tập dữ liệu	0.8250	0.8881	0.7438

Bảng 3.1: Bảng so sánh chỉ số đánh giá mô hình

Từ các kết quả trên, có thể thấy mô hình đạt độ chính xác khá cao ( $\approx 82.5\%$ ),

tất cả các chỉ số đánh giá đều được cải thiện chỉ qua một số lượng ít epoch. Trong ma trận nhầm lẫn thì chỉ số TP là chỉ số được cải thiện rõ rệt nhất, tức là mô hình nhận diện rất hiệu quả khi xác thực khuôn mặt của một người. Ta có được những sự cải thiện này là do mô hình Xception thực ra được huấn luyện trên tập dữ liệu không chứa nhiều khuôn mặt của người châu Á nên hiệu quả của mô hình chưa được huấn luyện là thấp hơn.

Ta cùng xem qua một vài ví dụ phân loại sai, tức là có khoảng cách (Anchor, Positive) > khoảng cách (Anchor, Negative) trong bộ ba.



Hình 3.8: Ví dụ phân loại sai

Từ ví dụ trên, ta có thể thấy phân loại sai do một số nguyên nhân sau:

- Khác biệt giữa màu sắc của ảnh Anchor và ảnh Positive.
- Khác biệt về ngoại cảnh (có đeo kính - không đeo kính, gầy - béo, già - trẻ, ...).
- Khác biệt về góc độ chụp.

- Bản thân có nét giống với người khác.

Qua đó, ta có thể xem xét thêm một số phương pháp chỉnh góc độ của ảnh, chỉnh màu, huấn luyện trên tập dữ liệu có cùng các điều kiện ngoại cảnh,... để có thể cải thiện độ chính xác của mô hình.

# Kết luận

## Kết quả của đồ án

Đồ án đã nghiên cứu và tìm hiểu về cách huấn luyện lại mô hình Xception trên dữ liệu người Việt và đã đạt được độ chính xác 82.5%. Tuy là kết quả cải thiện còn chưa phải là tốt nhất do thiếu thốn về mặt tài nguyên phần cứng cũng như thời gian.

## Kỹ năng đạt được

1. Biết đầu biết tìm kiếm, đọc, dịch tài liệu chuyên ngành liên quan đến nội dung đồ án.
2. Biết tổng hợp các kiến thức đã học và kiến thức trong tài liệu tham khảo để viết báo cáo đồ án.
3. Chế bản đồ án bằng L<sup>A</sup>T<sub>E</sub>X, viết mô hình sử dụng ngôn ngữ PYTHON.
4. Biết tóm tắt nội dung đồ án và biết trình bày một báo cáo khoa học.
5. Nắm được quy trình nghiên cứu, cải tiến và đề xuất kết quả mới.

## Hướng phát triển của đồ án trong tương lai

1. Áp dụng thêm các mô hình, phương pháp phù hợp để tiền xử lý dữ liệu sao cho dữ liệu sẽ có phân phối tương đồng về cả độ sắc nét, ánh sáng, độ phân giải,...

2. Áp dụng thêm các mô hình, phương pháp mới để giải quyết vấn đề khuôn mặt có thêm đối tượng như khẩu trang, kính măt, ...
3. Xây dựng thêm mô hình nhận diện phát hiện khuôn mặt để ứng dụng kết hợp thiết kế phần mềm điểm danh cho bài toán nhận diện khuôn mặt.

# Chỉ mục

- bộ dữ liệu, 23
- hướng tiếp cận bài toán, 16
  - dựa trên đặc trưng, 17
  - mô hình mạng nơ ron tích chập, 17
  - so khớp toàn diện, 16
  - mạng nơ ron tích chập, 6
    - AlexNet, 6
    - EigenFace, 16
    - FaceNet, 18
    - TripletLoss, 19
  - GoogLenet, 6
  - Lenet, 6
- VGGNet, 6
- Xception, 19
- tích chập phân rời, 20
- nhận diện khuôn mặt, 15
  - Face Alignment, 16
  - Face Detection, 16
  - Face Extraction, 16
  - Face Matching, 16
- thuật toán SGD, 12
- độ chính xác, 24
  - Ma trận nhầm lẫn, 25
  - Precision, 25
  - Recall, 25

# Tài liệu tham khảo

- [1] F.-F. Li, J. Johnson, and S. Yeung, “Cs231n: Convolutional neural networks for visual recognition,” Stanford University, 2019.
- [2] A. Ng, “Deep learning specialization,” Coursera, 2017.
- [3] A. Amini, “Deep computer vision,” MIT, 1 2023. [Online]. Available: [https://introtodeeplearning.com/slides/6S191\\_MIT\\_DeepLearning\\_L3.pdf](https://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L3.pdf)
- [4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [5] Y. Kundathil, “Cnn for mnist handwritten dataset,” Medium, 5 2020. [Online]. Available: <https://medium.com/@yash.4198/cnn-for-mnist-handwritten-dataset-cc94d61f6c94>
- [6] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [7] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.
- [8] C.-F. Wang, “A basic introduction to separable convolutions,” Towards Data Science, 8 2019. [Online]. Available: <https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728>
- [9] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” 2017.

- [10] P. H. Quang, “Vn-celeb: Dữ liệu khuôn mặt người nổi tiếng việt nam và bài toán face recognition,” Viblo, 5 2019. [Online]. Available: <https://viblo.asia/p/vn-celeb-du-lieu-khuon-mat-nguo-i-noi-tieng-viet-nam-va-bai-toan-face-recognition-Az45>
- [11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, ser. NIPS’12. Curran Associates Inc., 2012.
- [13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd International Conference on Learning Representations (ICLR 2015)*, pp. 1–14, 2015.
- [15] M. Turk and A. Pentland, “Face recognition using eigenfaces,” in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.
- [16] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, “Siamese neural networks for one-shot image recognition,” in *ICML deep learning workshop*, vol. 2, no. 1. Lille, 2015.
- [17] N. H. Sơn, “Mã nguồn mô hình,” 7 2023. [Online]. Available: <https://drive.google.com/drive/folders/1OrT1zJLgkpV9S9BV3r5X6wIiRx6E40iq?usp=sharing>