



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH

---



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

**Tìm kiếm đối tượng trong tập dữ liệu ảnh dựa  
trên đặc trưng của đối tượng**

HỘI ĐỒNG: Khoa học máy tính

GVHD: PGS. TS. Nguyễn Thanh Bình

GVPB: TS. Phạm Văn Hai

—o0o—  
SVTH 1: Dương Thanh Nam (1512058)

SVTH 2: Nguyễn Ngọc Tâm (1620058)

SVTH 3: Nguyễn Mậu Vĩnh (1627058)

TP. HỒ CHÍ MINH, 6/2019

## **Lời cam đoan**

Tôi xin cam đoan...

**Lời cảm ơn / Lời ngỏ**

Tôi xin chân thành cảm ơn ...

## **Tóm tắt nội dung**

Tóm tắt luận văn ...

# Mục lục

<b>Mục lục</b>	<b>i</b>
<b>Danh sách bảng</b>	<b>iii</b>
<b>Danh sách hình vẽ</b>	<b>iv</b>
<b>1 Giới thiệu</b>	<b>1</b>
1.1 Giới thiệu đề tài . . . . .	1
1.2 Mục tiêu, nội dung đề tài . . . . .	1
1.2.1 Mục tiêu . . . . .	1
1.2.2 Nội dung đề tài . . . . .	2
1.3 Giới hạn đề tài . . . . .	2
1.4 Cấu trúc luận văn . . . . .	2
<b>2 Cơ sở lí thuyết và các nghiên cứu liên quan</b>	<b>4</b>
2.1 Cơ sở lí thuyết . . . . .	4
2.1.1 Ảnh số . . . . .	4
2.1.2 Các định dạng ảnh số . . . . .	4
2.1.3 Trích xuất dữ hình ảnh theo đặc trưng . . . . .	6
2.1.4 Các đặc trưng của hình ảnh. . . . .	7
2.1.5 Deep learning . . . . .	8
2.1.6 Các mô hình mạng neural . . . . .	17
2.1.7 Giải thuật phân cụm K-means . . . . .	21
2.2 Các nghiên cứu liên quan . . . . .	26
2.2.1 Bài báo: "Deep learning for content-based image retrieval: A comprehensive study." [16] . . . . .	26
2.2.2 Bài báo: "A simple texture feature for retrieval of medical images." [17] . . . . .	26
2.2.3 Bài báo: "A new method of content based medical image retrieval and its applications to CT imaging sign retrieval" [18] . . . . .	26
2.2.4 Bài báo: "Deep convolutional learning for Content Based Image Retrieval" [19] . . . . .	27
2.2.5 Bài báo: "Learning Fine-grained Image Similarity with Deep Ranking" [20] . . . . .	27
<b>3 Phương pháp đề xuất</b>	<b>28</b>
3.1 Yêu cầu bài toán . . . . .	28
3.2 Phương pháp đề xuất . . . . .	28
3.3 Các phương pháp đánh giá . . . . .	30
3.3.1 Accuracy . . . . .	30

3.3.2	Precision-Recall[21]	30
3.3.3	Mean Average Precision (mAP)[21]	30
<b>4</b>	<b>Hiện thực và đánh giá kết quả</b>	<b>31</b>
4.1	Tài nguyên cần thiết	31
4.1.1	Ngôn ngữ Python	31
4.1.2	Các Framework sử dụng trong Luận văn	31
4.1.3	Google Colaboratory	32
4.2	Hiện thực các module	32
4.2.1	Load Dataset và xử lý Dataset	32
4.2.2	Huấn luyện mạng CNN	33
4.2.3	Tinh chỉnh với Triplet loss	33
4.2.4	Trích xuất hình ảnh	33
4.3	Kết quả	33
<b>5</b>	<b>Kết luận</b>	<b>34</b>
5.1	Kết quả đạt được	34
5.2	Ưu và nhược điểm của phương pháp đề xuất	34
5.2.1	Ưu điểm	34
5.2.2	Nhược điểm	34
5.3	Hướng mở rộng trong tương lai	34
	<b>Tài liệu tham khảo</b>	<b>35</b>

# Danh sách bảng

4.1	Các Dataset trong quá trình huấn luyện mô hình . . . . .	33
-----	--	----



# Danh sách hình vẽ

2.1	Kiến trúc tiêu biểu của một hệ thống truy xuất hình ảnh dựa trên nội dung. . . .	6
2.2	Minh họa chuẩn màu RGB [6] . . . . .	8
2.3	Mô hình mạng neural đơn giản. [9] . . . . .	9
2.4	Minh họa biểu diễn nhãn của một hình ảnh bằng one-hot encoding [10] . . . .	10
2.5	Minh họa so sánh giá trị dự đoán và nhãn dựa vào one-hot encoding [10] . . . .	10
2.6	Minh họa đồ thị hàm mất mát với cross entropy [10] . . . . .	11
2.7	Minh họa đồ thị hàm mất mát với Gradient descent. [11] . . . . .	12
2.8	Minh họa cập nhật tham số của ba loại giải thuật Gradient descent. [12] . . . .	13
2.9	Forward pass. . . . .	14
2.10	Ví dụ kiến trúc mạng nơ-ron tích chập. . . . .	15
2.11	Đồ thị hàm ReLU . . . . .	16
2.12	Max pooling với filter size = 2 x 2, stride = 1 . . . . .	17
2.13	Kiến trúc mạng dựa trên Triplet Loss . . . . .	19
2.14	Kiến trúc VGG . . . . .	19
2.15	Cấu trúc mạng VGG . . . . .	20
2.16	Chọn số cụm k = 2 . . . . .	22
2.17	Chọn ngẫu nhiên 2 điểm làm center . . . . .	22
2.18	Lần lượt gán các điểm vào center gần nhất . . . . .	23
2.19	sau khi phân cụm với center cũ, tiến hành chọn center mới . . . . .	23
2.20	Phân cụm với center mới . . . . .	24
2.21	Sau các bước lặp mà không có sự thay đổi, ta nhận được 2 cụm kết quả cuối cùng	24
3.1	Sơ đồ xử lý của giải thuật sử dụng mạng nơ-ron tích chập. . . . .	29

# Chương 1

## Giới thiệu

### 1.1 Giới thiệu đề tài

Với sự phát triển của Internet, cùng với những tiến bộ trong công nghệ lưu trữ dữ liệu và thu thập hình ảnh, nhất là xung quanh chúng ta luôn có sẵn các thiết bị để chụp lại các bức ảnh như máy ảnh kỹ thuật số, điện thoại di động, tablet,... khiến cho kích thước tập dữ liệu ảnh kỹ thuật số đang tăng lên một cách nhanh chóng. Với một tập hình ảnh đang ngày một lớn như thế thì câu hỏi đặt ra là làm thế nào để tìm kiếm đối tượng trong tập dữ liệu hình ảnh theo nhu cầu sử dụng một cách nhanh chóng và chính xác. Câu hỏi này cũng chính là một nội dung được các nhà khoa học máy tính quan tâm nghiên cứu trong vài thập kỷ qua.

Các phương pháp tìm kiếm hình ảnh hiện nay có hai hướng tiếp cận chính, thứ nhất đó là tìm kiếm theo metadata của dữ liệu ảnh, thứ hai là tìm kiếm theo các đặc trưng được thể hiện trong chính bức ảnh đó. Tuy nhiên, với số lượng hình ảnh quá lớn, việc tiếp cận theo hướng tìm kiếm theo metadata tỏ ra không hiệu quả, bởi vì việc tạo metadata thủ công tốn rất nhiều thời gian và công sức, hơn nữa, chúng ta rất khó kiểm soát việc gắn metadata như vậy có chính xác hay không, vì nó phụ thuộc nhiều vào con người, ngôn ngữ và bối cảnh của hình ảnh. Để khắc phục những hạn chế này, tiếp cận bằng cách tìm kiếm đối tượng theo đặc trưng của ảnh được các nhà nghiên cứu quan tâm, phát triển và cũng đã đạt được nhiều kết quả đáng kể.

Các phương pháp được thực hiện theo hướng này giúp cho việc lưu trữ đặc trưng cũng như tìm kiếm đối tượng diễn ra một cách có hệ thống. Đồng thời hạn chế sự tham gia của con người trong việc tạo dữ liệu đặc trưng, qua đó tăng độ chính xác cũng như tốc độ khởi tạo dữ liệu trong cơ sở dữ liệu hình ảnh.

Với kích thước dữ liệu ngày càng lớn lên nhanh chóng như hiện nay và việc tìm kiếm tri thức trên dữ liệu đó ngày càng quan trọng và có giá trị. Các ứng dụng trên dữ liệu đó ngày càng được chú trọng, đồng thời ngày càng có nhiều phương pháp tìm kiếm hình ảnh đã được đề xuất và đang được nghiên cứu.

Trong Luận văn này, nhóm sẽ tìm hiểu một số phương pháp tìm kiếm đối tượng theo đặc trưng đã được nghiên cứu và phát triển, đồng thời hiện thực một phương pháp mà nhóm đề xuất.

### 1.2 Mục tiêu, nội dung đề tài

#### 1.2.1 Mục tiêu

Mục tiêu của đề tài là hiểu được ảnh số và những đặc trưng của nó, làm sao để tìm kiếm được những đối tượng trong tập dữ liệu ảnh. Sau khi tìm hiểu các công trình nghiên cứu liên quan từ trước tới nay, nhóm tiến hành đề xuất một phương pháp tìm kiếm đối tượng và hiện

---

thực chúng. Sau khi hiện thực sẽ triển khai hệ thống và đánh giá kết quả đạt được. Trình bày ưu nhược điểm và xác định được hướng đi kế tiếp sau luận văn.

### 1.2.2 Nội dung đề tài

Nội dung chính của đề tài bao gồm các những phần sau đây:

1. Tìm hiểu các định dạng ảnh số.
2. Tìm hiểu các một số đặc trưng của hình ảnh.
3. Tìm hiểu về Deep Learning và các khái niệm, ứng dụng liên quan.
4. Tìm hiểu các công trình nghiên cứu liên quan đến đề tài.
5. Đề xuất phương pháp tìm kiếm đối tượng dựa trên đặc trưng.
6. Hiện thực phương pháp đã đề xuất và triển khai.
7. Đánh giá kết quả đạt được.

## 1.3 Giới hạn đề tài

Trong bài luận văn này, nhóm sẽ chỉ nghiên cứu về phương pháp tìm kiếm cho các ảnh chỉ chứa một đối tượng, và đối tượng này sẽ chiếm phần lớn nội dung của bức ảnh, đồng thời bức ảnh cũng thể hiện đầy đủ đối tượng. Ví dụ bức ảnh sẽ chỉ chứa một chiếc xe hơi được chụp ở khoảng cách gần, bức ảnh sẽ thể hiện đầy đủ chiếc xe hơi, đồng thời nội dung bức ảnh sẽ chủ yếu thể hiện chiếc xe hơi, không phải một đoạn đường có nhiều xe hơn hoặc một chiếc xe hơi rất nhỏ nằm trong một khung cảnh lớn.

## 1.4 Cấu trúc luận văn

Nội dung bài luận văn này được chia thành 5 chương như sau:

**Chương 1:** Giới thiệu về nội dung đề tài, tính cần thiết của đề tài, mục tiêu và nội dung của đề tài và phạm vi nghiên cứu của nhóm.

**Chương 2:** Trình bày cơ sở lý thuyết về hình ảnh, các phương pháp trích xuất đặc trưng ảnh và các phương pháp đo lường độ tương đồng của hình ảnh dựa trên đặc trưng trích xuất được. Trình bày các cơ sở lý thuyết liên quan tới phương pháp đề xuất của nhóm. Ngoài ra chương này còn trình bày tóm lược về các nguyên cứu liên quan và những kết quả mà các nghiên cứu này đã đạt được.

**Chương 3:** Trình bày câu câu của bài toán, phương pháp đề xuất và các chỉ số đánh giá phương pháp đó.

---

**Chương 4:** Trình bày những tài nguyên cần thiết khi hiện thực phương pháp đề xuất. Hiện thực các module trong phương pháp và kết quả đạt sau khi hiện thực và triển khai.

**Chương 5:** Chương này sẽ nêu ra các kết quả đạt được, ưu nhược điểm và định hướng mở rộng trong tương lai.

# Chương 2

## Cơ sở lý thuyết và các nghiên cứu liên quan

### 2.1 Cơ sở lý thuyết

#### 2.1.1 Ảnh số

**Ảnh số** là đại diện của hình ảnh thực, là một tập hợp những con số được lưu trữ, xử lý bởi máy tính. Để chuyển hình ảnh thành số, hình ảnh được chia thành các khu vực nhỏ gọi là pixel. Đối với mỗi pixel, thiết bị hình ảnh ghi lại một số hoặc một bộ số nhỏ mô tả một số thuộc tính của pixel này, chẳng hạn như độ sáng của nó hoặc màu của nó. Các số được sắp xếp trong một mảng các hàng và cột tương ứng với vị trí dọc và ngang của các pixel trong ảnh.[1] Mỗi số có giá trị trong khoảng 0 - 255. Với ảnh màu ta sẽ có ba ma trận hai chiều. Mỗi ma trận là một kênh màu của ảnh. Ba ma trận tương ứng với ba kênh màu Red, Green và Blue đối với ảnh RGB. Giá trị tại một pixel trên ảnh là tổng hợp ba giá trị tại vị trí tương ứng ở ba ma trận trên.

Hình ảnh kỹ thuật số có một số đặc điểm cơ bản:

- Loại hình ảnh. Ví dụ, hình ảnh đen trắng chỉ ghi lại cường độ ánh sáng trên các pixel. Một hình ảnh màu có thể có ba màu, thường là RGB (Đỏ, Xanh lục, Xanh lam) hoặc bốn màu, CMYK (Cyan, Magenta, Yellow, black). (ppi).[1]
- Độ phân giải. Độ phân giải được thể hiện bằng số pixel trên mỗi inch (ppi). Độ phân giải cao hơn cho hình ảnh chi tiết hơn. Một màn hình máy tính thường có độ phân giải 100 ppi, trong khi máy in có độ phân giải từ 300 ppi đến hơn 1440 ppi. Đây là lý do tại sao một hình ảnh được in trông tốt hơn nhiều so với trên màn hình.[1]
- Độ sâu màu (của hình ảnh màu) hoặc "bit trên pixel" là số lượng bit mô tả độ sáng hoặc màu. Nhiều bit hơn có thể ghi lại nhiều sắc thái của màu xám hoặc nhiều màu hơn.[1]
- Định dạng của hình ảnh cung cấp thêm chi tiết về cách các số được sắp xếp trong tệp hình ảnh, bao gồm loại nén nào được sử dụng (nếu có). Các định dạng phổ biến nhất là JPG, PNG, GIF, TIFF, và BMP.[1]

#### 2.1.2 Các định dạng ảnh số

##### a) Định dạng ảnh JPG

File JPG hay còn gọi là các tập tin JPEG, là một định dạng ảnh kỹ thuật số phổ biến. Định dạng JPG được lưu với thuật toán nén lossy, điều này đồng nghĩa với việc chất lượng hình ảnh sẽ bị giảm đi và kích thước tập tin cũng được giảm đáng kể.[2]

Ưu điểm của JPG/JPEG:

- Độ sâu màu từ 24 bit đến 16 triệu màu.

- 
- JPEG là chuẩn hình ảnh thông dụng nhất cho hầu hết các máy ảnh số hiện nay.
  - JPEG tương thích với mọi trình duyệt web hiện nay.
  - Sử dụng tốt nhất, hiệu quả nhất cho ảnh trắng đen, ảnh với màu sắc phức tạp, ảnh tĩnh vật, ảnh đời thường, chân dung.

#### **b) Định dạng ảnh PNG**

PNG (Portable Network Graphics) là một định dạng tập tin đồ họa raster, hỗ trợ nén dữ liệu không bị suy giảm chất lượng, được sử dụng rất nhiều trên internet.[2]

Ưu điểm của PNG:

- Nén theo chuẩn LossLess, có nghĩa là hình ảnh sau khi bị nén vẫn giữ nguyên được chất lượng.
- Định dạng PNG sử dụng tốt trên web/blog, những mảng màu phẳng, thiết kế Logo, hình ảnh có nền trong suốt hoặc bán trong suốt.
- Thích hợp với hình ảnh đơn giản như văn bản.

#### **c) Định dạng ảnh GIF**

GIF thường được dùng cho hình ảnh trên web, GIF sử dụng thuật nén Lossless mà không làm giảm chất lượng hình ảnh sau khi nén. GIF lưu dữ liệu bằng cách sử dụng màu indexed, có nghĩa là mỗi hình ảnh có thể bao gồm 256 màu.[2]

Ưu điểm của GIF:

- GIF hỗ trợ ít màu nên các tập tin thường có dung lượng nhỏ hơn JPEG rất nhiều.
- Hình ảnh được nén theo chuẩn Lossless nên không bị mất dữ liệu khi nén.
- GIF sống động với hình ảnh động.

#### **d) Định dạng ảnh TIFF**

TIFF (Tagged Image Format File) là một định dạng file ảnh chất lượng cao và được sử dụng nhiều cho việc Scan.[2]

Ưu điểm của TIFF:

- Cho dù bị nén hay không nén thì file TIFF cũng không bị mất bất kỳ dữ liệu.
- Do chất lượng hình ảnh của định dạng này rất tốt nên thường được sử dụng để lưu những hình ảnh có màu sắc phức tạp và thường được sử dụng để Scan.
- File TIFF có thể xem được, chỉnh sửa được.

#### **e) Định dạng ảnh BMP**

Định dạng BMP là một dạng file ảnh đồ họa dạng lưới (raster) được sử dụng để lưu trữ hình ảnh kỹ thuật số bitmap.[2]

Đặc điểm của BMP:

- File BMP không hỗ trợ tốt cho việc nén hình ảnh.
- Dễ dàng được tạo ra từ những dữ liệu pixel được lưu trong bộ nhớ máy tính.
- File Bitmap dễ dàng được dịch ra thành định dạng điểm cho các thiết bị như màn hình CRT và máy in.

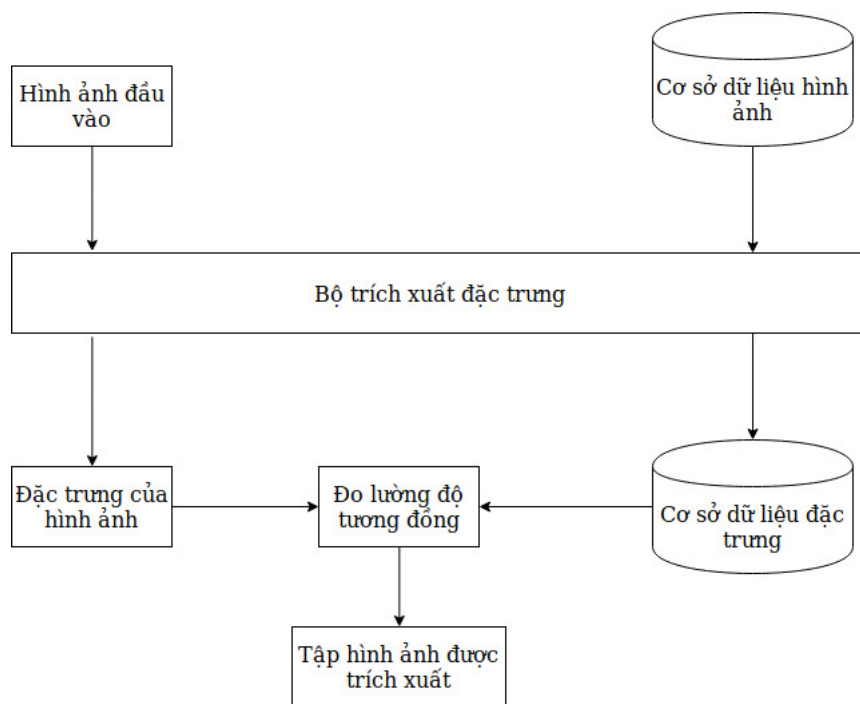
### 2.1.3 Trích xuất dữ hình ảnh theo đặc trưng

Content-Based Image Retrieval (CBIR) là một ứng dụng của thị giác máy tính nhằm trích xuất các hình ảnh tương tự với hình ảnh mẫu làm đầu vào cho hệ thống. Các kỹ thuật CBIR sử dụng nội dung hiển thị của hình ảnh được mô tả dưới dạng các đặc trưng để biểu diễn và tìm kiếm hình ảnh trong tập dữ liệu. CBIR là kỹ thuật đáng tin cậy vì hầu hết các công cụ tìm kiếm hình ảnh dựa trên web dựa hoàn toàn vào metadata và điều này rất tốn thời gian, tạo ra rất nhiều kết quả rác.

Người dùng có thể query hệ thống CBIR theo nhiều cách khác nhau, có thể là query theo một hình ảnh đầu vào, theo caption, theo một đoạn văn mô tả, theo một thuộc tính nào đó của hình ảnh. Sau đó hệ thống sẽ tìm kiếm và output những hình ảnh tương tự.

Những hình ảnh được coi là tương tự nhau nếu chúng tương tự nhau ở các đặc điểm ở trong hình ảnh (tương tự về màu, texture..), hoặc những hình ảnh gần như giống nhau hoàn toàn, chỉ khác nhau về màu, về phóng to thu nhỏ, xoay.., hoặc chứa những vật thể giống nhau, background giống nhau... Có hai mức độ tương tự nhau là tương tự nội dung trực quan và tương tự nội dung ngữ nghĩa.

Hình 2.1 minh họa một kiến trúc cơ bản của một hệ thống trích xuất ảnh theo đặc trưng với query là một hình ảnh đầu vào và tìm những hình ảnh tương tự dựa trên nội dung trực quan trong tập dữ liệu ảnh.



Hình 2.1: Kiến trúc tiêu biểu của một hệ thống truy xuất hình ảnh dựa trên nội dung.

Kỹ thuật truy xuất hình ảnh dựa trên nội dung sử dụng nội dung trực quan (visual content) của hình ảnh được mô tả dưới dạng các đặc trưng cấp thấp (low-level feature) như màu sắc (color), kết cấu (texture), hình dạng (shape) và vị trí không gian (spatial locations) để biểu diễn và tìm kiếm hình ảnh trong cơ sở dữ liệu.[3]

Trong CBIR, mỗi hình ảnh được lưu trữ trong cơ sở dữ liệu đều được trích xuất và so sánh với các đặc trưng với hình ảnh đầu vào. Nó bao gồm hai giai đoạn sau: [4]

- **Trích xuất đặc trưng (Feature extraction):** giai đoạn đầu tiên trong quá trình là trích xuất đặc trưng của hình ảnh đến mức độ khác biệt.

- **Kết nối (Mathching):** Giai thứ hai liên quan đến việc kết nối, so sánh các đặc trưng giữa những bức ảnh này để tìm ra kết quả tương tự như việc quan sát bằng mắt con người.

Trong [5], Eakins đã đề cập đến 3 mức truy vấn hình ảnh trong CBIR.

Level 1: Truy xuất bằng các đặc trưng gốc như màu sắc, kết cấu, hình dạng hoặc vị trí không gian của các phần tử hình ảnh. Truy vấn điển hình là truy vấn bằng ví dụ "tìm hình ảnh như thế này".

Level 2: Truy xuất các đối tượng thuộc loại đã cho được xác định bằng các tính năng dẫn xuất, với một mức suy luận logic nào đó. Ví dụ: "tìm hình ảnh của một bông hoa".

Level 3: Truy xuất bằng thuộc tính trừu tượng, liên quan đến một số lượng đáng kể lý luận cấp cao về mục đích của các đối tượng hoặc những cảnh được mô tả. Điều này bao gồm truy xuất về các sự kiện được đặt tên, ảnh có cảm xúc hoặc những điều đáng chú ý, v.v. Ví dụ truy vấn "tìm hình ảnh của một đám đông vui vẻ".

Một vài ví dụ về ứng dụng của CBIR: [4]

- **Phòng chống tội phạm:** Hệ thống nhận diện khuôn mặt tự động, được sử dụng bởi lực lượng cảnh sát.
- **Kiểm tra bảo mật:** Quét vân tay hoặc quét võng mạc để có các đặc quyền truy cập.
- **Chẩn đoán y tế:** Sử dụng CBIR trong cơ sở dữ liệu y tế về hình ảnh y tế để hỗ trợ chẩn đoán xác định các trường hợp tương tự trong quá khứ.
- **Sở hữu Trí tuệ:** Đăng ký hình ảnh nhãn hiệu, nơi so sánh nhãn hiệu ứng viên mới với các nhãn hiệu hiện có để đảm bảo không có nguy cơ gây nhầm lẫn quyền sở hữu tài sản

## 2.1.4 Các đặc trưng của hình ảnh.

### Color feature

Một trong những tính năng quan trọng nhất giúp con người nhận ra hình ảnh là màu sắc. Màu sắc là một đặc tính phụ thuộc vào sự phản xạ ánh sáng đến mắt và xử lý thông tin đó trong não. Chúng tôi sử dụng màu sắc hàng ngày để nói lên sự khác biệt giữa các vật thể, địa điểm và thời gian trong ngày.

Một số định dạng màu sắc thường gặp như RGB (Red, Green và Blue), HSV (Hue, Saturation, và Value), HSB (Hue, Saturation và Brightness)

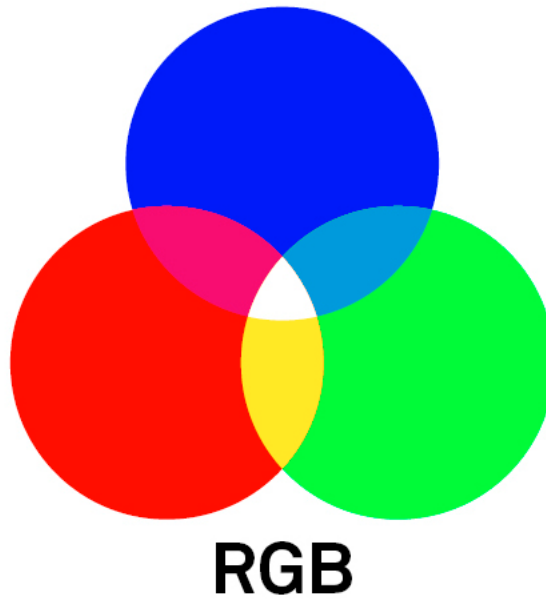
Hầu hết các định dạng hình ảnh như JPEG, BMP, GIF, sử dụng RGB để lưu trữ thông tin màu sắc của bức hình. RGB là viết tắt của ba màu cơ bản Đỏ (Red), Xanh lục (Green) và Xanh dương (Blue). Bằng cách trộn ba màu này theo các tỷ lệ khác nhau, người ta có thể đạt được tổng cộng khoảng 16,8 triệu màu như hình 2.2 [6]

### Texture feature

**Texture** là một thuộc tính quan trọng của hình ảnh. Kết cấu là thuộc tính nguyên bản của tất cả các bề mặt cái mà mô tả các mẫu trực quan, mỗi bề mặt có các tính chất đồng nhất. Nó chứa thông tin quan trọng về sự sắp xếp cấu trúc của bề mặt, chẳng hạn như; mây, lá, gạch, vải, v.v ... Nó cũng mô tả mối quan hệ của bề mặt với môi trường xung quanh. [4]

Về cơ bản, các phương pháp biểu diễn kết cấu có thể được phân thành hai loại: cấu trúc (structural) và thống kê (statistical). Các phương pháp cấu trúc, bao gồm toán tử hình thái và đồ thị kề, mô tả kết cấu bằng cách xác định nguyên thủy cấu trúc và quy tắc vị trí của chúng. Chúng có xu hướng hiệu quả nhất khi được áp dụng cho các kết cấu rất thường xuyên. Các phương pháp thống kê, bao gồm Fourier power spectra, cooccurrence matrices, phân tích thành phần chính bất biến (SPCA), Tamura feature, Wold decomposition, trường ngẫu nhiên Markov,





Hình 2.2: Minh họa chuẩn màu RGB [6]

mô hình fractal,.. đặc trưng cho kết cấu được xác định bằng cách thông kê phân bố cường độ hình ảnh. [7]

### Shape feature

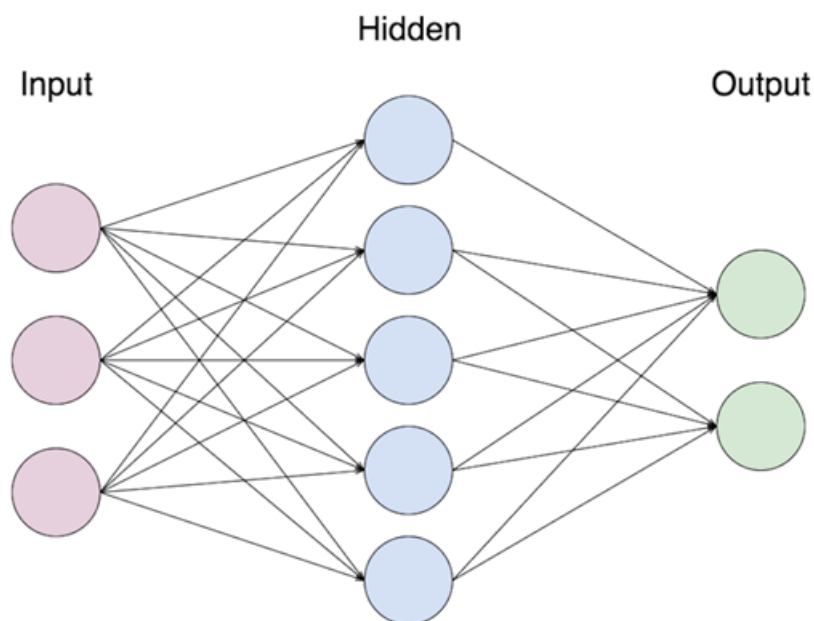
Shape có thể được định nghĩa là cấu hình bề mặt đặc trưng của một đối tượng; một hình thể hoặc đường viền. Nó cho phép một đối tượng được phân biệt với môi trường xung quanh bằng phác thảo của nó. Shape representations có thể được thường được chia thành hai loại là Boundary-based (dựa trên ranh giới) và Region-based (dựa trên khu vực). Một đặc trưng biểu diễn hình dạng tốt cho một đối tượng nên không bị thay đổi đối với việc dịch, xoay và chia tỷ lệ hình ảnh.[4]

## 2.1.5 Deep learning

### Mạng nơ-ron

Được lấy cảm hứng từ bộ não người, mạng nơ-ron sử dụng lớp nơ-ron được kết nối với nhau tạo thành một mô hình dùng để huấn luyện và dự báo dựa trên dữ liệu. Một mạng nơ-ron bao gồm một lớp đầu vào (input layer) chứa các đặc trưng của dữ liệu, một lớp đầu ra (output layer) chứa kết quả dự báo mà mạng tính toán được. Mạng nơ-ron cũng chứa một hoặc nhiều các lớp ẩn (hidden layer) nằm giữa hai lớp trên để thực hiện các bước tính toán trung gian nhằm dự đoán đầu ra dựa trên đầu vào của mạng. Trong hình 2.3 là mô hình một mạng nơ-ron đơn giản với lớp đầu vào gồm 3 nơ-ron, lớp ẩn gồm 5 nơ-ron và lớp đầu ra gồm 2 nơ-ron [9].

Một mạng nơ-ron trước khi được đưa vào sử dụng phải trải qua quá trình huấn luyện (train) và kiểm thử (test). Để thực hiện điều này, cần phải cung cấp các giá trị đầu vào và đầu ra chuẩn tương ứng. Sau mỗi vòng lặp trong quá trình huấn luyện, mạng nơ-ron được kỳ vọng sẽ cho ra kết quả dự đoán gần với kết quả thực tế, nói cách khác, sai số giữa đầu ra dự đoán và đầu ra chuẩn là nhỏ. Để đo lường sai số này, người ta sử dụng một hàm số gọi là hàm mất mát (loss function), giá trị của hàm này được gọi là giá trị mất mát (loss). Quá trình huấn luyện sẽ giảm



Hình 2.3: Mô hình mạng neural đơn giản. [9]

giá trị mất mát bằng cách áp dụng giải thuật lan truyền ngược (Backpropagation) kết hợp với các giải thuật tối ưu hóa như Gradient Descent, các giải thuật này sẽ được giải thích kỹ hơn ở các phần sau.

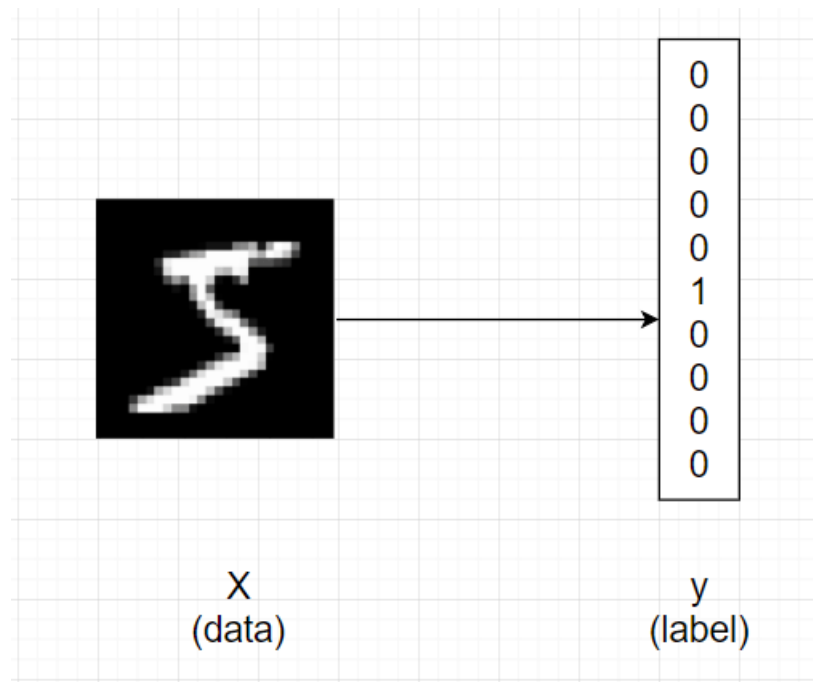
### Deep neural network

Mạng nơ-ron sâu (deep neural network) là một dạng mạng nơ-ron chứa nhiều lớp ẩn (hidden layer). Mạng nơ-ron sâu có thể học được các bài toán phức tạp hơn so với mạng nơ-ron nông.

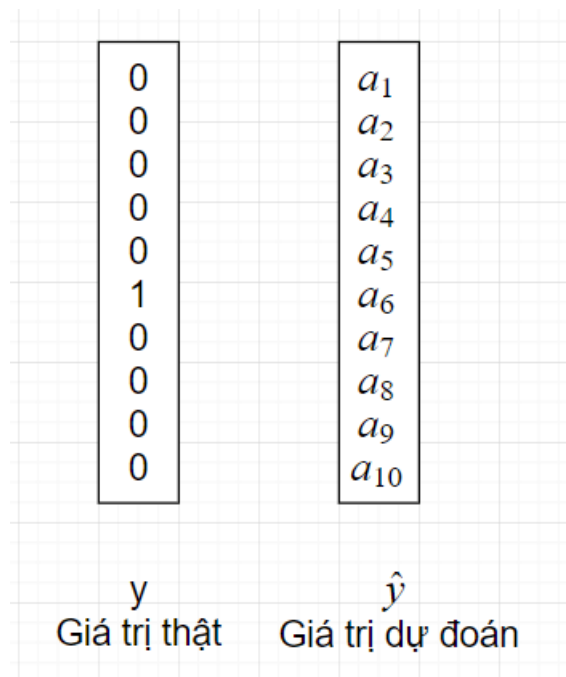
### Hàm mất mát

Để đo lường sai số giữa đầu dự đoán của mạng nơ-ron so với đầu ra chuẩn, người ta sử dụng một hàm số gọi là hàm mất mát. Giá trị của hàm số này càng nhỏ, thì mạng nơ-ron dự đoán càng gần với đầu ra chuẩn. Cho nên, nhìn chung, mục tiêu của quá trình huấn luyện là để tối thiểu hóa giá trị này. Giải thuật thường được áp dụng để giúp đạt được mục tiêu này là Gradient Descent.

Để định nghĩa toán học được hàm mất mát, trước tiên ta phải có đại lượng để tính được sai số giữa đầu ra dự đoán của mạng nơ-ron và đầu ra chuẩn. Đầu ra chuẩn này thường được gọi là nhãn (label). Vì đầu ra của mạng nơ-ron là một vector nên nhãn cũng phải được biểu diễn là một vector có cùng kích thước. Người ta thường dùng kỹ thuật gọi là one-hot encoding để chuyển đổi label của ảnh từ giá trị số sang vector.



Hình 2.4: Minh họa biểu diễn nhãn của một hình ảnh bằng one-hot encoding [10]



Hình 2.5: Minh họa so sánh giá trị dự đoán và nhãn dựa vào one-hot encoding [10]

Hình 2.4 minh họa ví dụ bài toán phân loại một hình ảnh thuộc class nào trong 10 class, vì có tất cả 10 class nên output của mạng nơ-ron sẽ là một vector có chiều dài là 10. Nên nhãn của hình ảnh cũng sẽ là một vector có chiều dài là 10. Những giá trị trong vector của nhãn có giá trị là 0 hoặc 1. Giá trị 1 ở vị trí nào thì tương ứng hình ảnh đó thuộc class tương ứng.

Đối với giá trị thật của một hình ảnh sau khi được dự đoán bởi mạng nơ-ron sẽ được biểu diễn bởi vector  $\hat{y}$  như trong hình 2.5. Giá trị của từng phần tử trong vector  $\hat{y}$  sẽ nằm trong khoảng từ 0 đến 1, giá trị càng cao thì độ chắc chắn hình ảnh nằm trong class tương ứng càng cao.

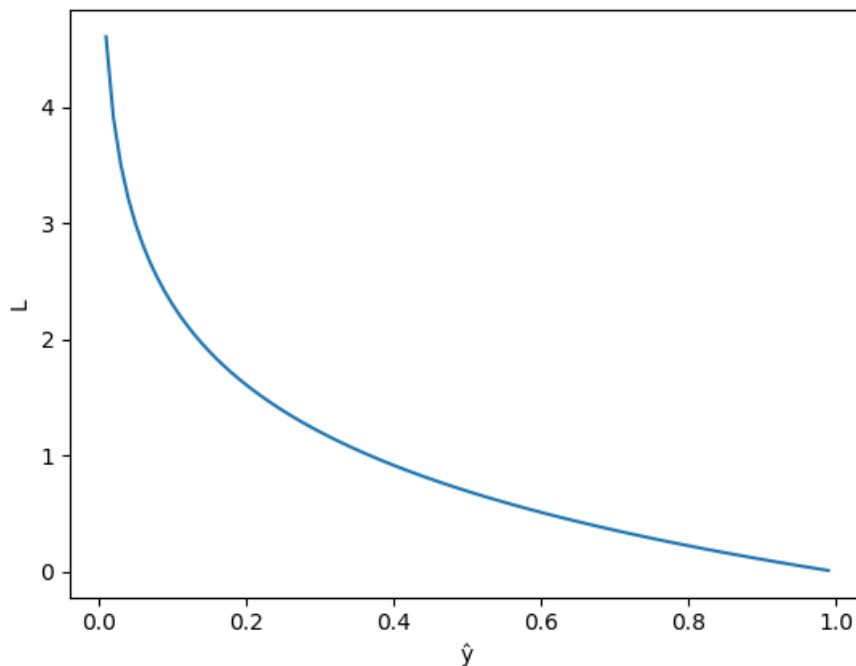
Ta định nghĩa hàm mất mát để đánh giá được độ tốt của mạng nơ-ron, đánh giá được sự khác biệt giữa  $y$  và  $\hat{y}$ . Tính hàm mất mát để điều chỉnh trọng số để  $\hat{y}$  tiến càng gần  $y$ . Như trong hình 2.5, mong muốn là giá trị dự đoán  $a_6$  sẽ càng gần với giá trị 1 và các giá trị khác trong vector càng gần giá trị 0 càng tốt.

Có nhiều cách để đánh giá sự khác biệt này, và một trong những cách phổ biến nhất đó là categorical cross entropy. Và categorical cross entropy rất phù hợp với các bài toán phân loại.

Tiếp tục với ví dụ về bài toán phân loại ảnh trong 10 class, hàm mất mát theo categorical cross entropy được định nghĩa như sau:

$$L = - \sum_{i=1}^{10} y_i * \log(\hat{y})$$

Giả sử ảnh đang xét là ảnh số 1, thì đồ thị hàm mất mát sẽ có dạng:



Hình 2.6: Minh họa đồ thị hàm mất mát với cross entropy [10]

Từ hình 2.6 ta thấy:

- Nếu kết quả dự đoán của mô hình càng gần 1 thì giá trị hàm mất mát càng nhỏ, nếu kết quả dự đoán của mô hình càng gần về 0 thì giá trị hàm mất mát càng lớn.
- Giá trị hàm mất mát giảm dần khi giá trị dự đoán từ 0 đến 1.
- Khi kết quả dự đoán càng gần 1, nghĩa là giá trị dự đoán càng gần với giá trị thật của hình nên hàm mất mát có giá trị nhỏ, và ngược lại.
- Khi mô hình dự đoán càng đúng thì hàm mất mát giá trị càng nhỏ và ngược lại. Và bài toán phân loại ảnh sẽ thành bài toán tìm giá trị nhỏ nhất của hàm mất mát.

Trong bài toán phân loại ảnh thì hàm mất mát categorical cross entropy có nhiều ưu điểm hơn so với hàm mất mát mean square.

---

## Gradient Descent

Là một giải thuật nhằm tìm kiếm điểm cực tiểu của hàm số. Giải thuật này thường được sử dụng trong lĩnh vực học máy nhằm giúp tối ưu hóa hàm mất mát. Giải thuật này được thực hiện bằng cách tính đạo hàm riêng của giá trị mất mát so với từng tham số của mô hình, sau đó điều chỉnh các tham số này theo giá trị đạo hàm tính được. Công thức để điều chỉnh như sau:

$$w = w - \alpha * \Delta w$$

Trong đó:

$w$ : tham số của mô hình

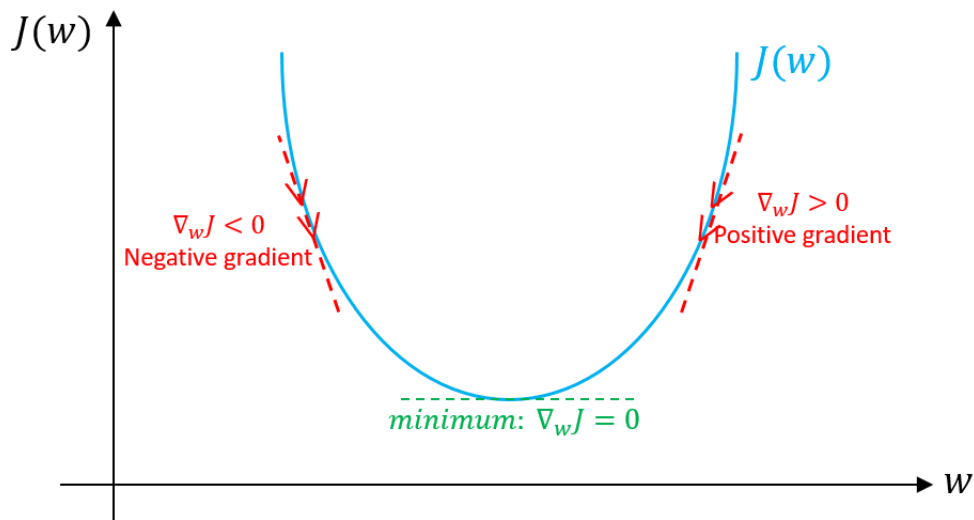
$\alpha$ : hệ số học (learning rate)

$\Delta w$ : đạo hàm của hàm mất mát theo  $w$

Để minh họa cho công thức trên, ta xem một ví dụ cho đồ thị hàm mất mát  $J(w)$  với  $w$  là một tham số của mạng nơ-ron (hình 2.7) . [11]

Ta biết rằng, tại điểm cực tiểu của hàm  $J$ , giá trị đạo hàm bằng 0, với các điểm bên trái điểm cực tiểu, giá trị đạo hàm  $< 0$ , với các điểm bên phải, giá trị đạo hàm  $> 0$ . Như vậy, với các điểm  $w$  nằm bên trái điểm cực tiểu, ta cần tăng  $w$  lên, và vì  $\Delta w$  lúc này  $< 0$ , nên công thức trên đã giúp tăng giá trị  $w$  lên đúng như mong đợi. Tương tự với các điểm bên phải, công thức trên cũng giúp giảm  $w$  về sát với điểm cực tiểu.

Một lưu ý là với **learning rate** lớn, các bước cập nhật lớn hơn giúp mạng có thể học nhanh hơn, tuy nhiên cũng tạo ra nguy cơ là sau khi cập nhật, giá trị hàm mất mát sẽ vượt qua khỏi điểm cực tiểu, và dao động qua lại 2 bên điểm cực tiểu nhưng không thể tới được đó. Ngược lại, với **learning rate** nhỏ, mạng nơ-ron sẽ học chậm hơn, nhưng có thể tiến sát tới điểm cực tiểu.

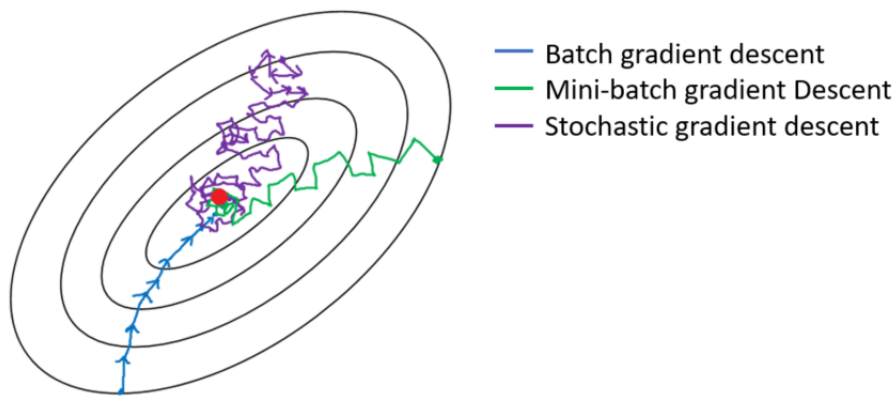


Hình 2.7: Minh họa đồ thị hàm mất mát với Gradient descent. [11]

## Mini-batch Gradient Descent

Trong giải thuật Gradient Descent, khi tính đạo hàm của hàm loss function, ta dùng tất cả các dữ liệu để tính đạo hàm rồi cập nhật lại các tham số. Giải thuật Gradient Descent chạy tốt với những tập dữ liệu nhỏ, nhưng nếu với tập dữ liệu ảnh cộng với số lượng dữ liệu lớn, việc tính đạo hàm trên toàn bộ dữ liệu rất tốn thời gian. Dựa vào số lượng dữ liệu được tính trong bước tính đạo hàm, người ta chia ra thành:

- Batch gradient descent: Dùng tất cả dữ liệu trong training set cho mỗi lần thực hiện bước tính đạo hàm.
- Mini-batch gradient descent: Dùng một phần dữ liệu trong training set cho mỗi lần thực hiện bước tính đạo hàm.
- Stochastic gradient descent: Chỉ dùng một dữ liệu trong training set cho mỗi lần thực hiện bước tính đạo hàm.



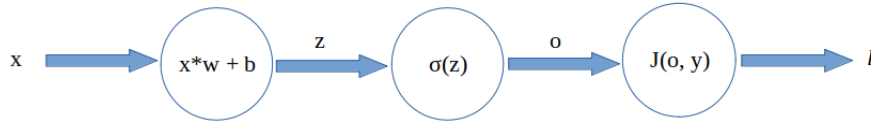
Hình 2.8: Minh họa cập nhật tham số của ba loại giải thuật Gradient descent. [12]

Hình 2.8 biểu diễn việc cập nhật trọng số trong gradient descent, điểm đỏ là giá trị tối ưu ta cần tìm, các điểm ở ngoài cùng là giá trị ban đầu của các trọng số trong gradient descent. Vì không có nhiễu nên trong batch gradient descent, trọng số cập nhật trực tiếp theo một đường thẳng. Mini-batch thì mất nhiều thời gian hơn và còn đi chệch hướng tuy nhiên thì vẫn đến được điểm đỏ. Còn stochastic thì đi khá lòng vòng để đến được điểm đỏ và vì dữ liệu quá nhiều nên có thể thuật toán gradient descent chỉ quanh điểm đỏ mà không đến được điểm đỏ. Tuy mất nhiều thời gian để đến được điểm cực tiểu tối ưu, nhưng Mini-batch rất thích hợp với dữ liệu có số lượng lớn do giảm chi phí tính đạo hàm cho toàn bộ dữ liệu, và thường được sử dụng trong deep learning.

## Giải thuật lan truyền ngược (Backpropagation)

Giải thuật chính để thực thi gradient descent trong mạng nơ-ron. Giải thuật này bao gồm 2 phần, forward pass và backward pass. Ở forward pass, các giá trị đầu vào sẽ được truyền qua từng lớp nơ-ron để tính toán đầu ra tương ứng. Ở mỗi lớp, các giá trị này sẽ được lưu lại để phục vụ cho việc tính toán ở backward pass. Ở backward pass, mục tiêu là tính đạo hàm riêng của hàm mất mát theo từng tham số trong mỗi lớp nơ-ron. Tuy nhiên, việc tính toán này không thể thực hiện một cách trực tiếp, bởi vì các tham số này không xuất hiện trực tiếp trong công thức tính hàm mất mát.

Để minh họa cho giải thuật này, ta sẽ xem xét một ví dụ với một mạng nơ-ron đơn giản chỉ gồm một lớp đầu vào, một lớp đầu ra cũng một lớp ẩn, mỗi lớp chỉ gồm một nơ-ron. Ở ví dụ này, ta sẽ sử dụng hàm kích hoạt là sigmoid ( $\sigma$ ), và hàm mất mát ( $J$ ) là mean square error. Quá trình tính toán ở forward pass diễn ra như biểu diễn ở hình 2.9:



Hình 2.9: Forward pass.

Trong đó:

$x$ : giá trị đầu vào

$z$ : giá trị sau bước nhân tuyến tính,  $z = x * w + b$

$o$ : giá trị đầu ra của nơ-ron:  $o = \sigma(z) = \frac{1}{1+e^{-z}}$

$y$ : giá trị đầu ra chuẩn

$l$ : giá trị mất mát,  $l = J(o, y) = \frac{1}{2} * (y - o)^2$

Viết ngắn gọn, ta có giá trị mất mát:  $l = J(\sigma(x * w + b), y)$

Để có thể tính được đạo hàm của hàm  $J$  theo các tham số  $w$  và  $b$  ta cần áp dụng Chain Rule như sau:

$$\Delta w = \frac{\partial l}{\partial w} = \frac{\partial l}{\partial o} * \frac{\partial o}{\partial z} * \frac{\partial z}{\partial w}$$

$$\Delta b = \frac{\partial l}{\partial b} = \frac{\partial l}{\partial o} * \frac{\partial o}{\partial z} * \frac{\partial z}{\partial b}$$

Thực hiện các bước biến đổi cho công thức tính  $\Delta w$  ở trên:

$$\begin{aligned} \Delta w &= \frac{\partial l}{\partial o} * \frac{\partial o}{\partial z} * \frac{\partial z}{\partial w} \\ &= (o - y) * (o * (1 - o)) * x \end{aligned}$$

Tương tự, với tham số  $b$  của nơ-ron, ta cũng tính được  $\Delta b$ :

$$\begin{aligned} \Delta b &= \frac{\partial l}{\partial o} * \frac{\partial o}{\partial z} * \frac{\partial z}{\partial b} \\ &= (o - y) * (o * (1 - o)) \end{aligned}$$

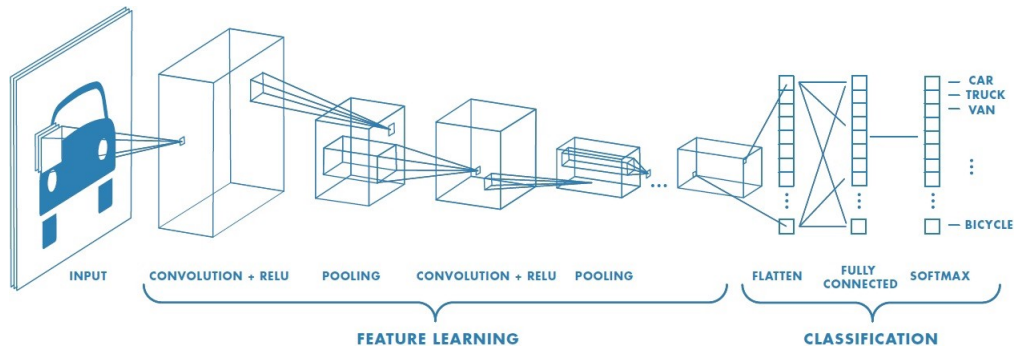
Đối với ví dụ trên, nếu gọi  $\Delta o$  là đạo hàm riêng của hàm mất mát theo  $o$ ,  $\phi(x * w + b)$  là hàm kích hoạt của nơ-ron, ta sẽ có công thức tổng quát cho việc tính  $\Delta w$ ,  $\Delta b$  là:

$$\Delta w = \Delta o * \phi'(x * w + b) * x$$

$$\Delta b = \Delta o * \phi'(x * w + b)$$

Với mạng nơ-ron có nhiều lớp hơn, ta cũng thực hiện tương tự các phép biến đổi này sẽ giúp tính được đạo hàm của hàm mất mát theo các tham số trong từng lớp nơ-ron. Ngoài ra, nếu gọi  $x_i$ ,  $o_i$  tương ứng là đầu vào và đầu ra của lớp nơ-ron thứ  $i$ , ta dễ dàng nhận thấy  $\Delta o_i = \Delta x_{i+1} + 1$ , điều này giúp chúng ta tiết kiệm quá trình tính toán bằng cách lan truyền ngược giá trị  $\Delta x$  từ lớp sau về lớp trước. Giá trị  $\Delta x$  này cũng có thể được tính bằng phương pháp tương tự như trình bày ở trên.

Qua các phần trình bày ở trên, ta có thể hiểu được cơ chế giúp cho mạng nơ-ron sâu có thể học được bài toán phân loại với dữ liệu đầu vào và đầu ra chuẩn. Ngoài ra, còn nhiều biến thể khác cho mạng học sâu và các giải thuật liên quan, tuy nhiên trong phạm vi đề tài, nhóm sẽ không đề cập tới.



Hình 2.10: Ví dụ kiến trúc mạng nơ-ron tích chập.

## Mạng nơ-ron tích chập

Là một dạng mạng nơ-ron truyền thẳng, trong đó có ít nhất một lớp tích chập. Mạng nơ-ron tích chập đã đạt được các kết quả rất tốt trong một số lĩnh vực cụ thể như nhận diện hình ảnh. Mạng này cũng được dùng như là một bộ trích xuất đặc trưng hình ảnh và thu về những kết quả khả quan. Mạng nơ-ron tích chập hiệu quả hơn mạng nơ-ron thông thường vì mạng sử dụng phép tích chập lên các hình ảnh đầu vào, qua đó giảm chi phí tính toán của mạng.

Kiến trúc của một mạng nơ-ron tích chập thường bao gồm các lớp tích chập, theo sau là các lớp pooling, cuối cùng sẽ là các lớp fully-connected dùng để kết hợp các giá trị từ những lớp trước lại để cho ra kết quả dự đoán.

## Lớp tích chập

Thay vì sử dụng phép nhân ma trận giữa các giá trị đầu vào và ma trận trọng số như mạng nơ-ron thông thường, lớp tích chập sử dụng phép tính chập để tính giá trị đầu ra. Các ma trận trọng số, hay còn gọi là filter này sẽ được sử dụng để quét qua từng phần của hình ảnh đầu vào để tạo ra các giá trị trong ma trận đầu ra. Mỗi ô của ma trận đầu ra được tính theo công thức như sau:

$$Y(u, v) = X * W = \sum_{i=-r}^r \sum_{j=-r}^r X(u-i, v-j)W(i, j)$$

Trong đó:

Y: Ma trận đầu ra

X: Ma trận đầu vào (hình ảnh)

W: Ma trận filter được sử dụng cho phép tích chập

Ngoài ra, filter có thể được quét qua ảnh theo bước trượt (stride) khác nhau, và ảnh đầu vào cũng có thể được thêm các giá trị đệm ở viền (padding), thường là 0 nhằm điều chỉnh kích thước ma trận. Dựa vào các thông số đầu vào, ta có thể tính kích thước ma trận đầu ra tương ứng như sau:

$$W_o = (W_i - F + 2P)/S + 1$$

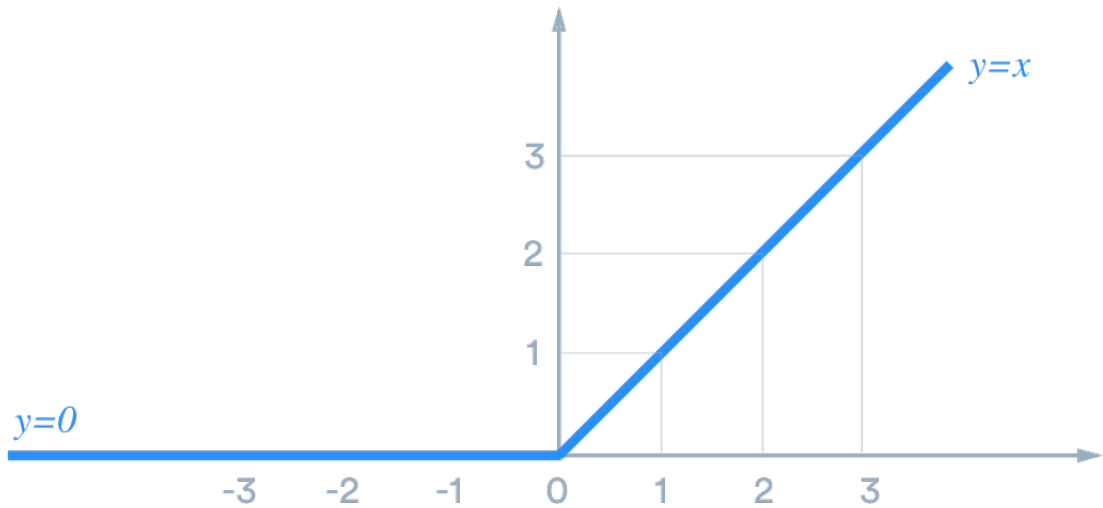
$$H_o = (H_i - F + 2P)/S + 1$$

$$D_o = K$$

Trong đó:

$W_i, H_i$ : chiều rộng, chiều cao của ma trận đầu vào





Hình 2.11: Đồ thị hàm ReLU

$W_o, H_o, D_o$ : chiều rộng, chiều cao, chiều sâu của ma trận đầu ra

$F$ : kích thước một cạnh của filter

$S$ : bước trượt

$P$ : padding

$K$ : số lượng filter

Sau khi được tính bằng phép tích chập, ma trận đầu ra sẽ được đi qua hàm kích hoạt ReLU nhằm tạo tính phi tuyến cho mạng nơ-ron. Công thức của hàm ReLU như sau:

$$y = \max(x, 0).$$

So với các hàm kích hoạt thường dùng khác như sigmoid hay tanh, hàm ReLU cho khả năng tính toán nhanh hơn, cùng với đó là giải quyết vấn đề triệt tiêu đạo hàm khi  $x > 0$ .

### Lớp pooling

Dùng để giảm kích thước của ma trận được tạo ra từ lớp nơ-ron phía trước, giúp giảm độ phức tạp tính toán của mạng cũng như hạn chế overfitting.

Tương tự như lớp convolutional, lớp pooling cũng sử dụng các filter để di chuyển qua từng vùng của đầu vào và tính giá trị đầu ra. Giá trị đầu ra thường được tính bằng cách lấy giá trị trung bình (average) hoặc giá trị lớn nhất (maximum) của các giá trị đầu vào.

Kích thước của filter thường dùng là  $2 \times 2$ , bước trượt thường dùng 2. Trên hình 2.12 là ví dụ với filter size =  $2 \times 2$ , stride = 1.

Kích thước ma trận đầu ra của lớp Pooling cũng có thể tính như sau:

$$W_o = (W_i - F) / S + 1$$

$$H_o = (H_i - F) / S + 1$$

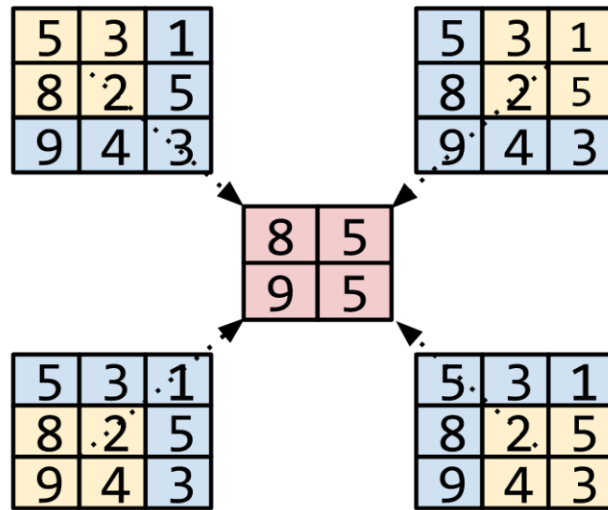
$$D_o = D_i$$

Trong đó:

$W_i, H_i, D_i$ : chiều rộng, chiều cao, chiều sâu của ma trận đầu vào

$W_o, H_o, D_o$ : chiều rộng, chiều cao, chiều sâu của ma trận đầu ra

$F$ : kích thước một cạnh của filter



Hình 2.12: Max pooling với filter size = 2 x 2, stride = 1

S: bước trượt

### Lớp fully-connected

Lớp này có nhiệm vụ kết hợp tất cả giá trị ở các nơ-ron ở lớp trước lại, tương tự như mạng nơ-ron thông thường. Ở lớp này, phép nhân ma trận được sử dụng để tạo các giá trị đầu ra. Lớp cuối cùng của một mạng nơ-ron tích chập là một lớp fully-connected có số nơ-ron bằng với số class trong bài toán phân loại, hàm kích hoạt của lớp này thường là hàm softmax, đây là một hàm số giúp phân hóa giá trị đầu ra về gần 2 điểm 0 và 1 hơn.

### Học chuyển giao

Học chuyển giao (transfer learning) là phương pháp sử dụng các mô hình mạng nơ-ron được huấn luyện sẵn để huấn luyện lại trên một tập dữ liệu mới. Việc huấn luyện lại này có thể diễn ra trên toàn bộ mạng, hoặc một vài lớp nơ-ron nhất định trong mạng. Ý tưởng của phương pháp này là sử dụng các trọng số đã được huấn luyện sẵn sẽ giúp mô hình học được nhanh hơn so với việc khởi tạo các giá trị trọng số này một cách ngẫu nhiên. Tuy nhiên khi thực hiện học chuyển giao trên một tập dữ liệu mới, chúng ta cũng phải lưu ý lựa chọn các lớp nơ-ron nào cần được huấn luyện lại, và chọn hệ số học của mô hình phù hợp để tránh hiện tượng overfitting.

Ngày nay, trong các bài toán sử dụng mạng nơ-ron tích chập để trích xuất đặc trưng từ ảnh, người ta thường dùng các mô hình đã được huấn luyện trên tập ImageNet để thực hiện học chuyển giao, giúp cho quá trình huấn luyện diễn ra nhanh hơn và cũng hiệu quả hơn.

## 2.1.6 Các mô hình mạng neural

### Kiến trúc mạng dựa trên Triplet Loss

#### Triplet Loss

Với các mô hình deep learning thông thường, ta cho từng hình ảnh vào mô hình, tuy nhiên với mô hình sử dụng Triplet Loss ta sẽ cho vào mô hình từng bộ ba (triplet).

Một bộ ba (triplet):  $t_i = (p_i, p_i^+, p_i^-)$  với  $p_i$  là ảnh query,  $p_i^+$  là ảnh positive (ảnh tương đồng với ảnh query),  $p_i^-$  là ảnh negative (ảnh khác hoàn toàn với ảnh query).

Các định nghĩa liên quan:

Ta định nghĩa sự tương đồng giữa hai hình ảnh P và Q dựa vào khoảng cách Euclid giữa chúng:

$$D(f(P), f(Q)) = \|f(P) - f(Q)\|_2^2$$

Với  $f(.)$  là hàm chuyển một hình ảnh sang một điểm trong không gian Euclid, và  $D(....)$  là hàm tính khoảng cách giữa 2 điểm trong không gian Euclid, giá trị hàm D càng nhỏ thì độ tương đồng giữa 2 hình càng lớn.

Giả sử ta có tập hợp các hình ảnh  $\mathcal{P}$ , và  $r_{i,j} = r(p_i, p_j)$  là độ đo tương đồng giữa hình ảnh  $p_i \in \mathcal{P}$  và  $p_j \in \mathcal{P}$ . Hai hình ảnh càng tương đồng nhau thì r càng lớn.

$$D(f(p_i), f(p_i^+)) < D(f(p_i), f(p_i^-)),$$

$$\forall p_i, p_i^+, p_i^-, r(p_i, p_i^+) > r(p_i, p_i^-)$$

Hàm loss cho một bộ ba (triplet) được định nghĩa:

$$l(p_i, p_i^+, p_i^-) = \max\{0, g + D(f(p_i), f(p_i^+)) - D(f(p_i), f(p_i^-))\}$$

Với g là một tham số hợp thức hoá khoảng cách giữa 2 cặp hình ảnh  $(p_i, p_i^+)$  và  $(p_i, p_i^-)$ . Hàm loss có giá trị trong khoảng  $[0,1]$ , và hàm mục tiêu là:

$$\min \sum_i \xi_i + \lambda \|\mathbf{W}\|_2^2$$

trong đó:

$$\max\{0, g + D(f(p_i), f(p_i^+)) - D(f(p_i), f(p_i^-))\} \leq \xi_i,$$

$$\forall p_i, p_i^+, p_i^-, r(p_i, p_i^+) > r(p_i, p_i^-)$$

$\lambda$  là tham số kiểm soát sự xếp hạng để cải thiện tính tổng quát của nó.  $\mathbf{W}$  là những tham số của hàm  $f(.)$ . Trong mô hình sử dụng triplet loss, việc quan trọng nhất là học và xác định tham số của hàm  $f(.)$  một cách tốt nhất có thể.

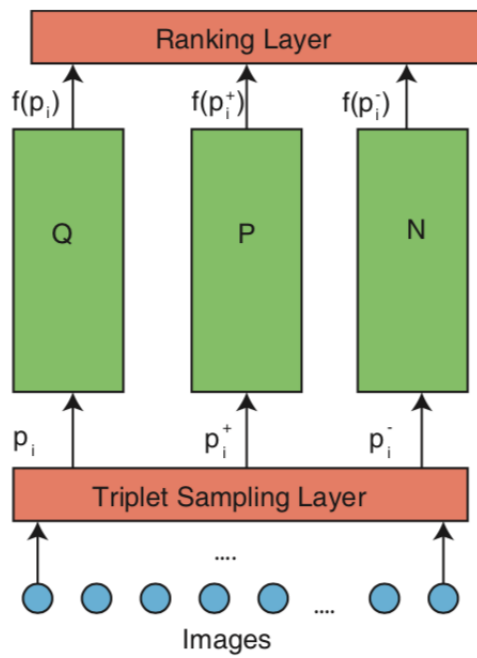
### Kiến trúc mạng dựa trên Triplet Loss

Kiến trúc mạng dựa trên Triplet Loss ở hình 2.13 [20] được đề xuất cho hàm mất mát dùng để ranking. Mạng này lấy input là những triplet. Một triplet bao gồm  $p_i$  là ảnh query,  $p_i^+$  là ảnh positive (ảnh tương đồng với ảnh query),  $p_i^-$  là ảnh negative (ảnh khác hoàn toàn với ảnh query). Một triplet đặc trưng cho mỗi quan hệ tương đồng giữa 3 hình ảnh đó. Chúng được vào 3 mạng neural giống nhau về kiến trúc và tham số  $f(.)$ . Mạng neural  $f(.)$  tính toán ra vector đặc trưng của hình ảnh  $p_i : f(p_i) \in \mathcal{R}^d$ , với d là số chiều của vector đặc trưng của hình ảnh.

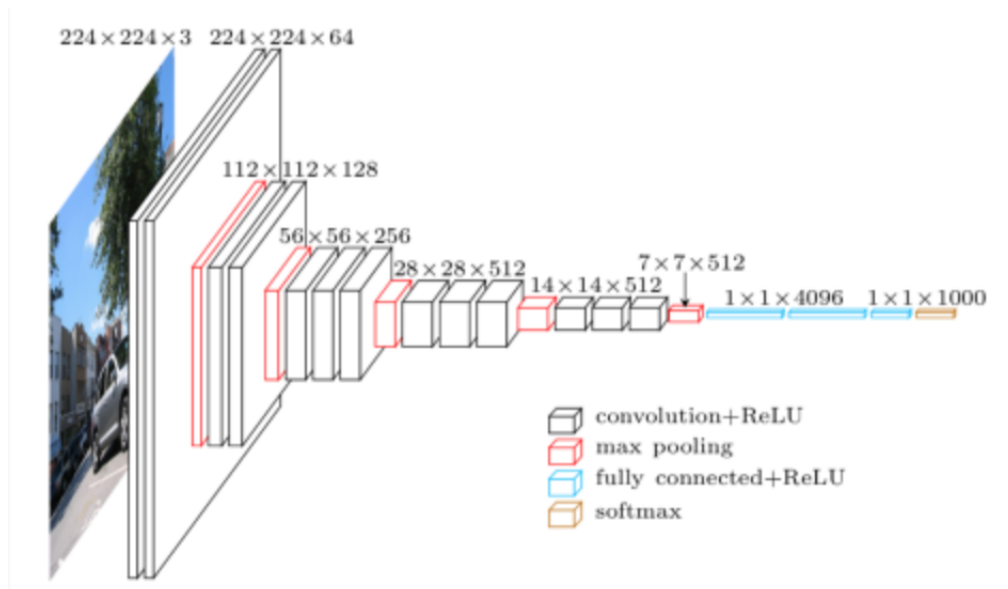
Lớp ranking trên cùng của mạng tính toán Loss của một triplet. Lớp ranking không có tham số nào. Trong suốt quá trình học, lớp ranking dùng giải thuật lan truyền ngược các gradient đến các lớp trước để các lớp trước có thể điều chỉnh tham số của chúng để giảm chi phí (loss).

### Kiến trúc mạng VGG

Kiến trúc mạng VGG trong hình 2.14 được giới thiệu bởi Simonyan và Zisserman vào năm 2014 trong bài báo "Very Deep Convolutional Networks for Large Scale Image Recognition". Đặc trưng của cấu trúc mạng này là sự đơn giản của nó, chỉ sử dụng những lớp tích chập 3x3 chồng lên nhau khi chiều sâu mạng tăng dần. Việc giảm kích thước được xử lý bởi hàm max



Hình 2.13: Kiến trúc mạng dựa trên Triplet Loss  
[20]



Hình 2.14: Kiến trúc VGG  
[13]

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Hình 2.15: Cấu trúc mạng VGG  
[14]

---

pooling. Hai lớp fully-connected, mỗi lớp có 4096 node, theo sau hai lớp fully-connected là lớp kích hoạt softmax.

Trong mạng VGG16 và VGG19, số 16 và 19 dựa trên số lớp trọng số trong mạng (cột D và E trong hình 2.15 [14]).

Simonyan và Zisserman nhận thấy việc training VGG16 và VGG19 là đầy thách thức (cụ thể là về sự hội tụ trên các mạng sâu hơn), vì vậy để làm cho việc training dễ dàng hơn, trước tiên họ đã train các phiên bản VGG nhỏ hơn với các lớp trọng số ít hơn (cột A và C).

Các mạng nhỏ hơn hội tụ và sau đó được sử dụng làm khởi tạo cho các mạng lớn hơn, sâu hơn - quá trình này được gọi là việc train trước. Mặc dù có ý nghĩa logic, việc train trước là một công việc rất tốn thời gian, tẻ nhạt, đòi hỏi toàn bộ mạng phải được train trước khi nó có thể đóng vai trò khởi tạo cho một mạng sâu hơn.

VGG có hai nhược điểm:

- Việc train chậm.
- Các trọng số của kiến trúc mạng này khá lớn (về mặt đĩa / băng thông).

## 2.1.7 Giải thuật phân cụm K-means

### Định nghĩa giải thuật K-means

Giải thuật K-means dùng để phân cụm dữ liệu, với điểm trung tâm của mỗi cụm là giá trị trung bình của những đối tượng trong cụm đó.

Input:

- k: Số lượng cụm dữ liệu.
- D: một dataset chứa n đối tượng.

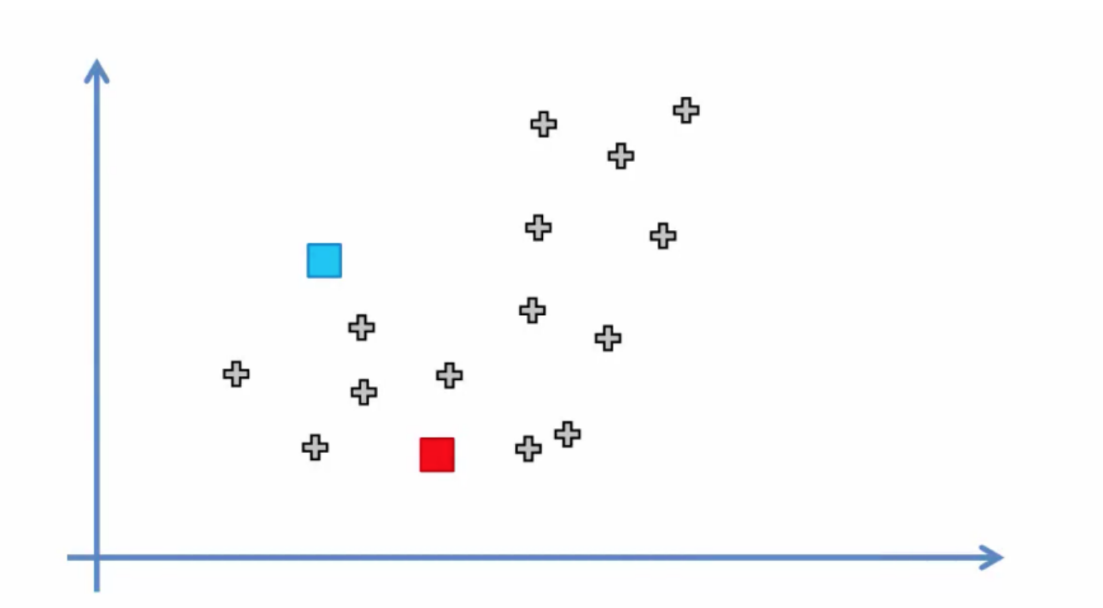
Output: Tập dữ liệu được phân thành k cụm các đối tượng.

Giải thuật:

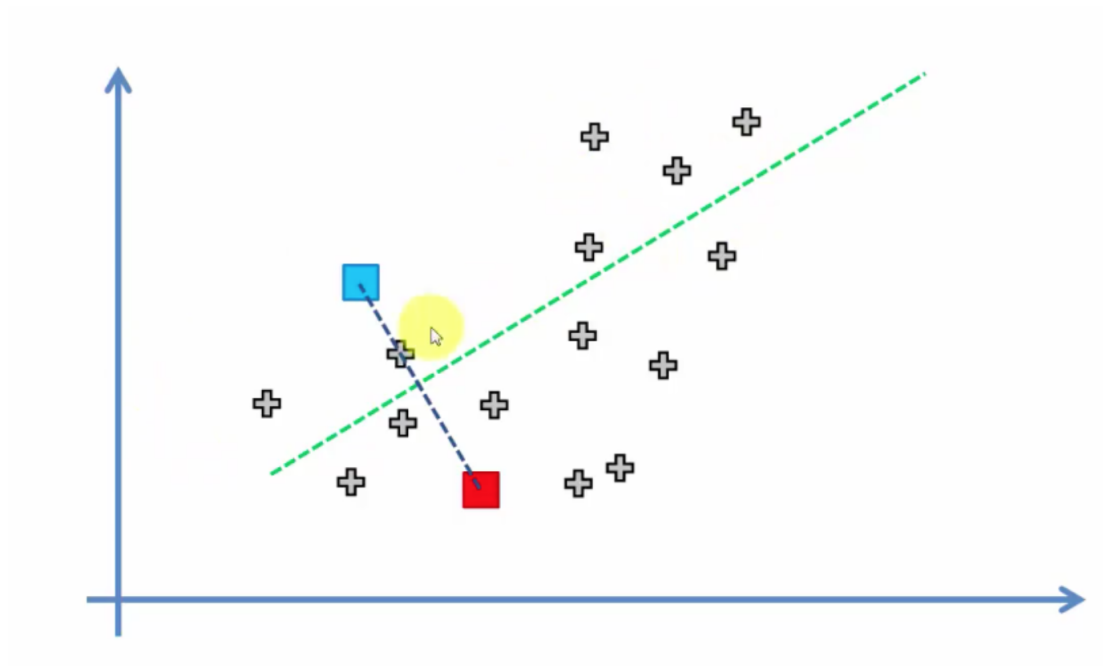
- Bước 1: Chọn k điểm bất kỳ trong tập dữ liệu làm center ban đầu.
- Bước 2: Gán mỗi điểm dữ liệu vào cluster có center gần nó nhất.
- Bước 3: Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.
- Bước 4: Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất cả các điểm dữ liệu đã được gán vào cluster đó sau bước 2.
- Bước 5: Quay lại bước 2.



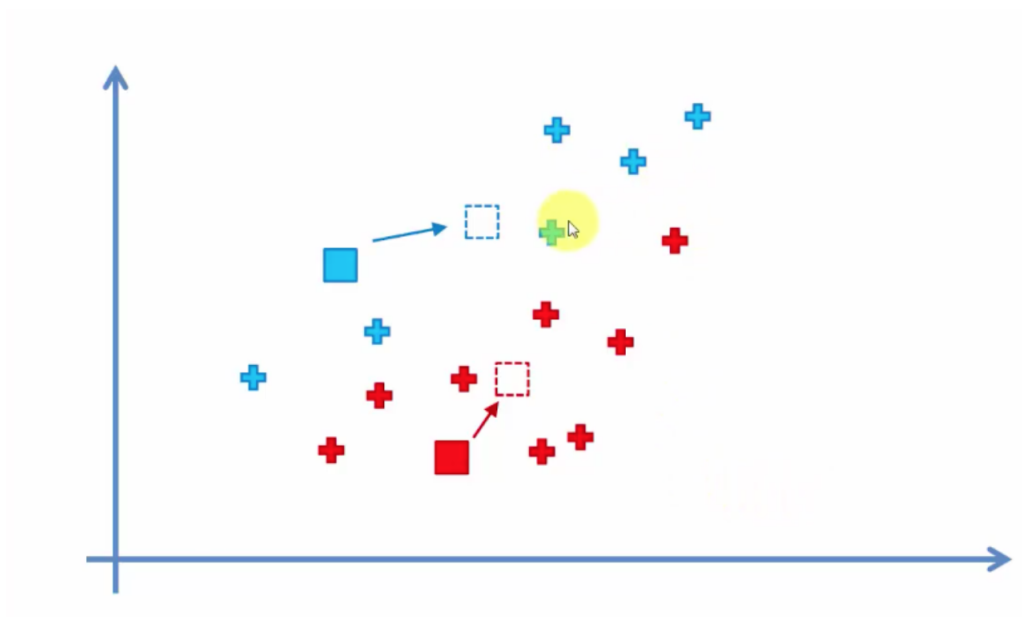
Hình 2.16: Chọn số cụm  $k = 2$   
[15]



Hình 2.17: Chọn ngẫu nhiên 2 điểm làm center  
[15]

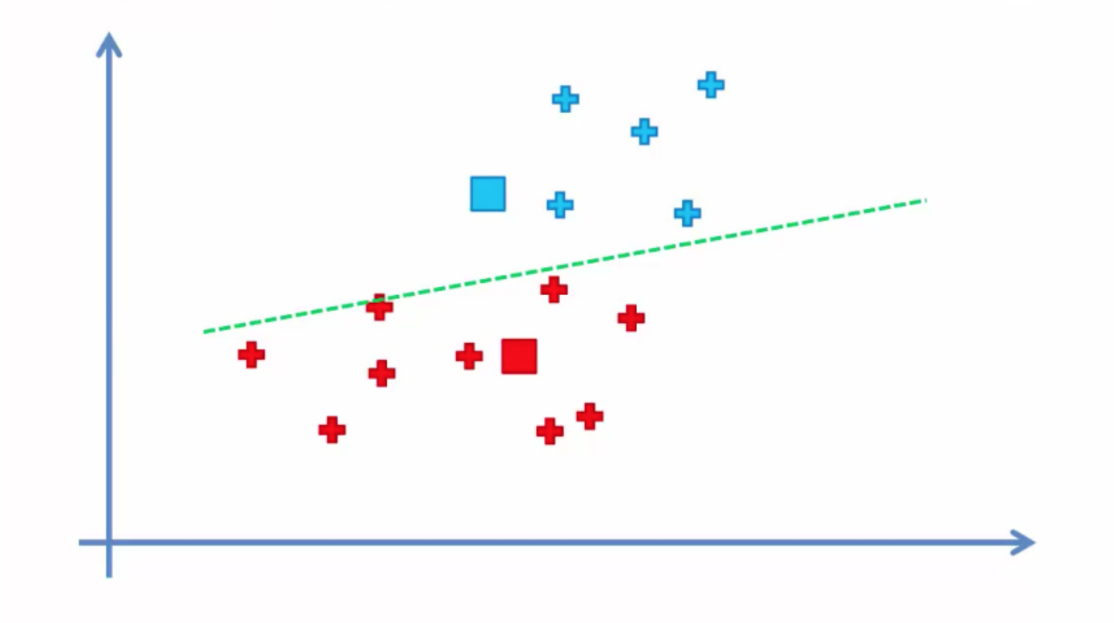


Hình 2.18: Lần lượt gán các điểm vào center gần nhất  
[15]

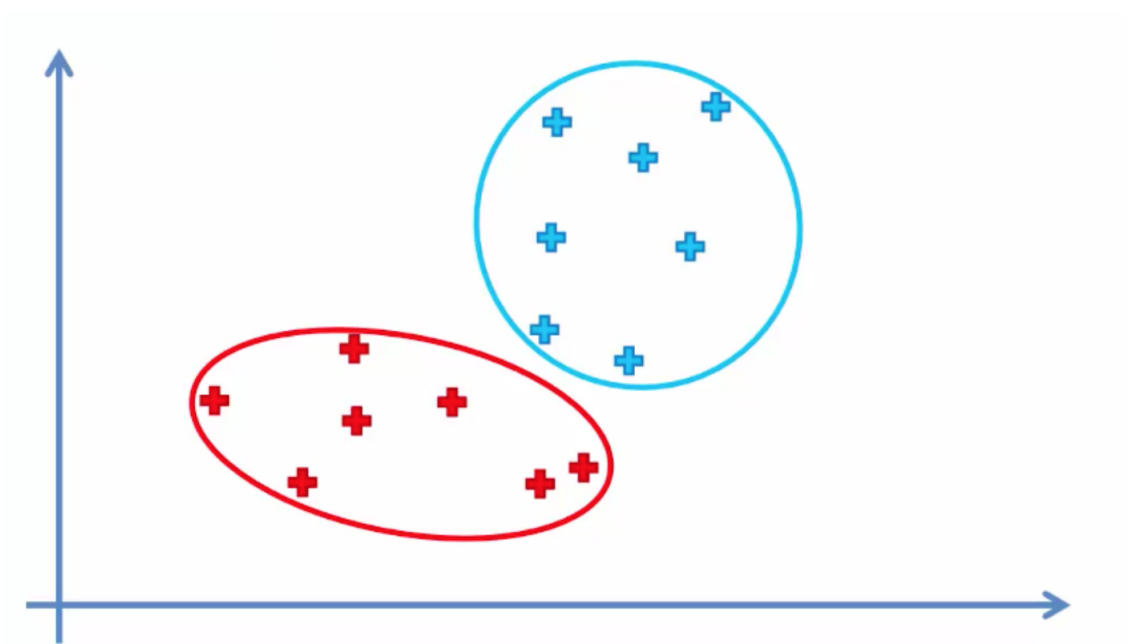


Hình 2.19: sau khi phân cụm với center cũ, tiến hành chọn center mới  
[15]





Hình 2.20: Phân cụm với center mới  
[15]



Hình 2.21: Sau các bước lặp mà không có sự thay đổi, ta nhận được 2 cụm kết quả cuối cùng  
[15]

---

### Đặc điểm của giải thuật K-means

- Bài toán tối ưu hoá với cực trị cục bộ.
- Mỗi cụm đặc trưng hoá bởi center của cụm (giá trị trung bình của cụm) nên bài toán đặt ra là số cụm (k) nên là bao nhiêu?
- Giải thuật ảnh hưởng bởi nhiễu.
- Không phù hợp với việc khai phá các cụm dữ liệu không lồi hoặc các cụm có kích thước khác nhau.

**Chất lượng của các cụm  $C_i$  được tính bởi công thức sau:**

$$E = \sum_{i=1}^k \sum_{p \in C_i} \text{dist}(p, C_i)^2$$

E: Tổng bình phương sai số (SSE) của tất cả đối tượng trong tập dữ liệu.

p: Là điểm trong không gian tập dữ liệu đại diện cho một đối tượng nhất định

$c_i$  là trung tâm của cụm  $C_i$

dist là độ đo khoảng cách giữa hai đối tượng trong không gian Euclide:

$$\text{dist}(x, y) = \left( \sum_{k=1}^n |x_k - y_k|^2 \right)^{1/2}$$

---

## 2.2 Các nghiên cứu liên quan

### 2.2.1 Bài báo: "Deep learning for content-based image retrieval: A comprehensive study." [16]

Bài báo nghiên cứu về việc áp dụng Deep Learning cho CBIR, thông qua 2 quá trình chính:

- Giai đoạn 1: đào tạo một mô hình Deep Learning từ một tập lớn dữ liệu đào tạo.
- Áp dụng mô hình sâu đã được training cho việc học biểu diễn đặc trưng của CBIR trong một miền mới.

Và kết quả thu được rất khả quan:

- Deep CNN model được train từ trước trên tập data lớn có thể được sử dụng trực tiếp cho việc rút trích đặc trưng trong nhiệm vụ CBIR mới.
- Các đặc trưng được rút ra bởi mô hình CNN được đào tạo trước có thể hoặc có thể không tốt hơn các đặc trưng thủ công truyền thống, nhưng với các lược đồ tinh chỉnh đặc trưng phù hợp, biểu diễn đặc trưng học sâu luôn vượt trội so với các đặc trưng thủ công truyền thống trên tất cả các tập dữ liệu.
- Khi được áp dụng biểu diễn đặc trưng trong một domain mới, thấy rằng học tương đồng có thể tăng cường hơn nữa hiệu suất truy xuất của các đặc trưng trực tiếp của các mô hình sâu được đào tạo trước.
- Cuối cùng, bằng cách đào tạo lại các mô hình sâu với phân loại hoặc mục tiêu học tương tự trên domain mới, Nhận thấy rằng hiệu suất truy xuất có thể được tăng lên đáng kể, tốt hơn nhiều so với các cải tiến do học tập tương tự "nông".

### 2.2.2 Bài báo: "A simple texture feature for retrieval of medical images." [17]

Bài báo đề xuất 1 cách tiếp cận đơn giản để sử dụng các đặc trưng kết cấu của hình ảnh y tế để cho việc trích xuất. Cách tiếp cận đầu tiên tiến hành lọc những hình ảnh y tế sử dụng bộ lọc Gabor và bộ lọc Schmid và sau đó phân vùng các hình ảnh được lọc thành các bản vá không chồng lên nhau. Những hoạt động này cung cấp thông tin đặc trưng kết cấu của những hình ảnh y tế. Mô hình bag-of-words cuối cùng được sử dụng để có được những biểu diễn đặc trưng của những hình ảnh y tế. So với một số đặc trưng hiện có, đặc trưng được đề xuất có tính phân biệt và hiệu quả hơn. Các thí nghiệm trên hai cơ sở dữ liệu hình ảnh CT y tế đã chứng minh tính hiệu quả của phương pháp được đề xuất.

### 2.2.3 Bài báo: "A new method of content based medical image retrieval and its applications to CT imaging sign retrieval" [18]

Bài báo đề xuất một phương pháp mới để thu thập hình ảnh y tế dựa trên nội dung thông qua việc xem sự tương đồng nhạy cảm ngữ cảnh, đồng nhất. Thứ nhất, nhóm nghiên cứu kết hợp các điểm tương đồng ngữ nghĩa và hình ảnh giữa hình ảnh truy vấn và từng hình ảnh trong cơ sở dữ liệu dưới dạng các điểm giống nhau của chúng. Sau đó, xây dựng một biểu đồ có trọng số có các nút đại diện cho các hình ảnh và các cạnh đo các điểm giống nhau của chúng. Bằng cách sử dụng thuật toán đường đi ngắn nhất trên biểu đồ trọng số, nhóm nghiên cứu có được một thước đo tương tự mới, đo lường tương tự về ngữ cảnh, giữa hình ảnh truy vấn và từng hình ảnh cơ sở dữ liệu để hoàn tất quá trình truy xuất. Trên thực tế, nhóm nghiên cứu sử dụng tính tương tự của cặp tương tự để thu hẹp khoảng cách ngữ nghĩa để có được một phép đo tương tự cặp đôi

---

chính xác hơn và trải rộng trên phạm vi dữ liệu nội tại để đạt được sự giống nhau về ngữ cảnh cho hiệu suất truy xuất tốt hơn. Phương pháp đề xuất đã được đánh giá dựa trên việc rút trích các dấu hiệu hình ảnh CT phổ biến của bệnh phổi (CISLs) và đạt được không chỉ kết quả thu hồi tốt hơn mà còn đạt hiệu quả tính toán thỏa đáng.

#### 2.2.4 Bài báo: "Deep convolutional learning for Content Based Image Retrieval" [19]

Bài báo đề xuất một phương pháp huấn luyện lại mô hình để tìm hiểu các biểu diễn convolutional hiệu quả hơn cho việc truy xuất hình ảnh dựa trên nội dung. Sử dụng mô hình Deep CNN đã được huấn luyện trước để thu được các biểu diễn tính năng từ việc kích hoạt các lớp convolutional bằng cách sử dụng max-pooling, và sau đó điều chỉnh và huấn luyện lại mạng, để tạo ra các mô tả hình ảnh nhỏ gọn hiệu quả hơn, cải thiện hiệu suất truy xuất và yêu cầu bộ nhớ, dựa vào thông tin có sẵn.

Các tác giả gợi ý ba phương pháp huấn luyện mô hình cơ bản. Đó là:

- Huấn luyện lại hoàn toàn không được giám sát khi không có thông tin ngoại trừ từ tập dữ liệu có sẵn.
- Huấn luyện lại với thông tin liên quan khi các nhãn của tập dữ liệu huấn luyện có sẵn.
- Huấn luyện lại dựa trên phản hồi có liên quan nếu có phản hồi từ người sử dụng.

Việc đánh giá thực nghiệm trên ba tập dữ liệu truy xuất hình ảnh công khai cho thấy hiệu quả của phương pháp được đề xuất trong việc học các biểu diễn hiệu quả hơn cho việc trích xuất, vượt trội hơn các kỹ thuật trích xuất dựa trên CNN khác, cũng như các phương pháp dựa trên đặc điểm thủ công truyền thống trong tất cả bộ dữ liệu đã sử dụng.

#### 2.2.5 Bài báo: "Learning Fine-grained Image Similarity with Deep Ranking" [20]

Bài báo đề xuất việc học tương tự hình ảnh với mô hình deep ranking, mô hình này sử dụng hàm loss function dựa trên các bộ ba (triplet) để mô tả sự tương đồng giữa các hình ảnh. Một bộ ba (triplet) bao gồm một hình ảnh truy vấn, một ảnh positive và một ảnh negative với ảnh positive thì tương đồng với ảnh truy vấn nhiều hơn ảnh negative. Mô hình cũng sử dụng kiến trúc mạng neural đa tầng để lấy những thuộc tính của hình ảnh.

Trong quá trình nghiên cứu, các tác giả đã đề xuất ra một phương pháp bootstrap mới để tạo ra training data, phương pháp này có thể tạo ra số lượng training data không giới hạn. Để sử dụng dữ liệu một cách hiệu quả, một thuật toán lấy mẫu bộ ba trực tuyến được đề xuất để tạo ra các bộ ba (triplet) phân biệt và sử dụng thuật toán stochastic gradient không đồng bộ trong việc tối ưu hóa hàm xếp hạng dựa trên bộ ba. Một cấu trúc mạng đa tầng mới đã được phát triển, trong đó có mạng neural tích chập, mạng neural này cho thấy cấu trúc mạng đa tầng này có thể hoạt động hiệu quả cho xếp hạng hình ảnh tương tự. Ngoài ra các tác giả còn xuất bản một bộ dữ liệu đánh giá với thông tin xếp hạng tương tự cho các hình ảnh cùng loại.

Kết quả nghiên cứu cho thấy mô hình deep ranking đạt được hiệu suất tốt hơn nhiều so với mô hình dựa trên đặc trưng và mô hình phân loại sâu. Các mô hình mô phỏng hình ảnh có thể được áp dụng cho nhiều ứng dụng vì tính máy tính khác, chẳng hạn như nhận dạng/phát hiện đối tượng dựa trên mẫu và sao chép hình ảnh.

## Chương 3

# Phương pháp đề xuất

### 3.1 Yêu cầu bài toán

Yêu cầu của bài toán là hiện thực một phương pháp giúp tìm kiếm đối tượng trong một tập dữ liệu ảnh cho trước. Để có thể tìm kiếm đối tượng theo đặc trưng, bài toán cũng tạo ra một yêu cầu phụ là phải tạo ra cơ sở dữ liệu đặc trưng tương ứng với từng ảnh trong cơ sở dữ liệu hình ảnh, thông qua đó mới có thể so sánh với hình ảnh đầu vào.

### 3.2 Phương pháp đề xuất

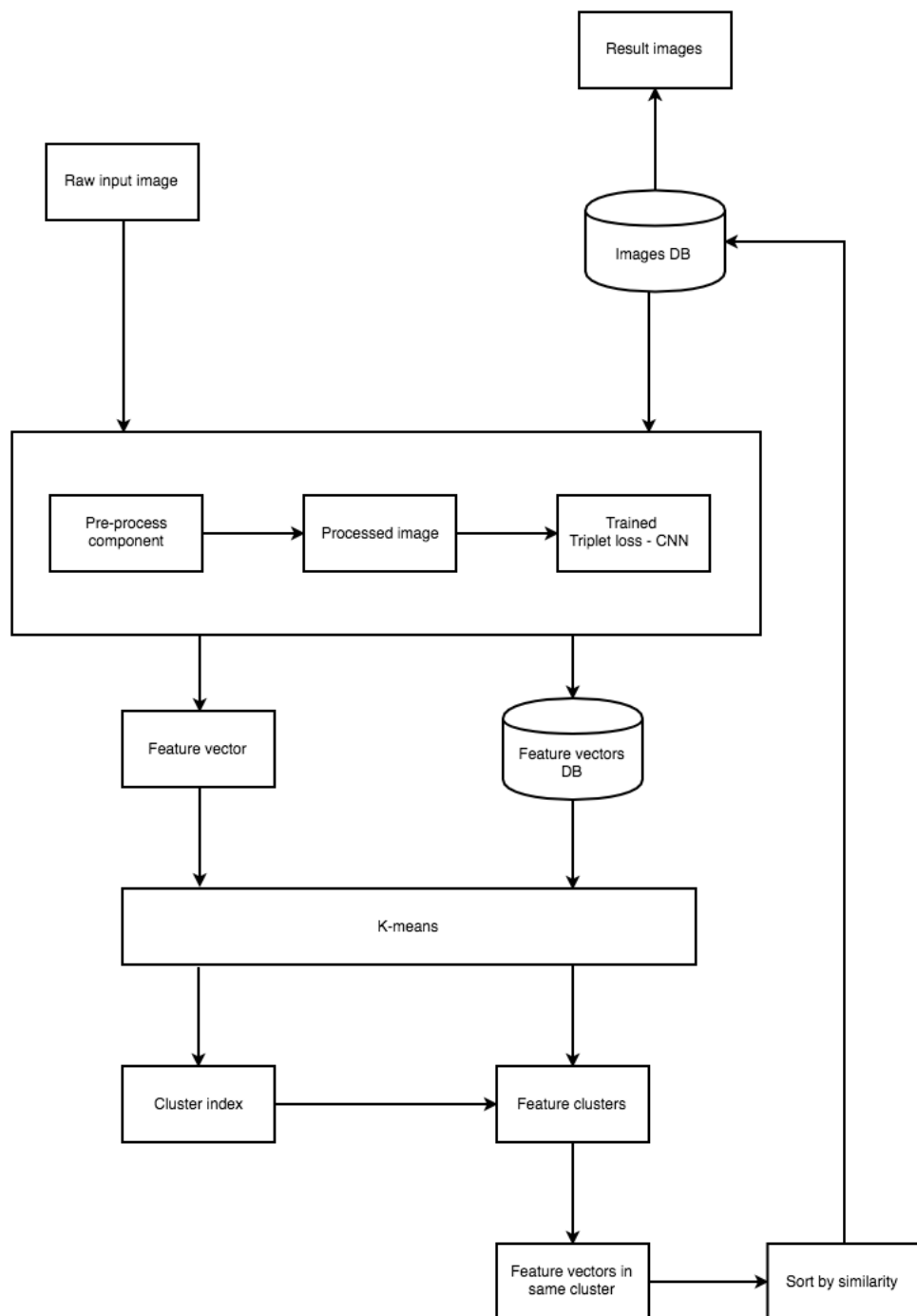
Trong phần này, nhóm sẽ đánh giá kết quả thu được từ việc trích xuất dựa trên một mô hình mà nhóm đề xuất. Nhiều bài báo sử dụng mạng nơ-ron tích chập để làm bộ trích xuất đặc trưng và sử dụng những đặc trưng đó để ứng dụng vào các bài toán khác nhau. Trong luận văn này nhóm sử dụng mô hình của nhóm để làm bộ trích xuất đặc trưng. Mô hình của nhóm là sự kết hợp giữa một mạng CNN và một mạng triplet loss.

Phương pháp này dựa trên ý tưởng mạng nơ-ron sau khi được huấn luyện để có thể phân loại ảnh thì cũng sẽ cho ra các đặc trưng tương tự với các bức ảnh tương tự nhau. Dựa vào ý tưởng này, các bức ảnh sau khi đi qua mô hình nhóm đề xuất cũng sẽ cho ra các đặc trưng tương tự với các bức ảnh tương tự nhau.

Input của phương pháp này là ảnh một vật thể đã được điều chỉnh kích thước phù hợp với kích thước đầu vào của mạng nơ-ron tích chập được huấn luyện. Output của phương pháp là tập các ảnh trong cơ sở dữ liệu có độ tương đồng cao nhất với ảnh đầu vào.

Sơ đồ xử lý của giải thuật được mô tả như hình 3.1, trong đó, ảnh đầu vào sẽ trải qua bước tiền xử lý để thu về ảnh có kích thước phù hợp với đầu vào của mô hình Triplet loss - CNN. Ảnh sau đó sẽ được truyền qua mô hình để thu về vector đặc trưng. Các hình ảnh trong database cũng qua mô hình để tạo ra những vector đặc trưng tương ứng. Những vector đặc trưng đó được tiến hành gom cụm để tạo ra các cluster dựa trên giải thuật K-mean. Sau đó, vector đặc trưng của hình ảnh truy vấn sẽ xác định được nằm ở cluster nào, lấy những vector đặc trưng nằm trong những cluster đó ra và tiến hành sắp xếp theo mức độ tương đồng từ cao tới thấp. Những vector tương đồng nhất tương ứng với những hình ảnh tương đồng với ảnh truy vấn nhất và những hình ảnh tương đồng đó sẽ được trích xuất từ database dựa theo index.

Ưu điểm của phương pháp này là cho kết quả nhanh chóng, đồng thời có thể áp dụng trực tiếp lên một tập dữ liệu mới mà không cần phải huấn luyện lại. Tuy nhiên có thể dễ thấy rằng việc áp dụng một mô hình được huấn luyện sẵn vào một tập dữ liệu khác với tập dữ liệu huấn luyện có thể sẽ mang đến kết quả không thực sự tốt.



Hình 3.1: Sơ đồ xử lý của giải thuật sử dụng mạng nơ-ron tích chập.

---

## 3.3 Các phương pháp đánh giá

### 3.3.1 Accuracy

Để đánh giá độ tốt của mạng CNNs sử độ đo lường độ chính xác của các hình ảnh được tìm thấy dựa theo nhãn của hình ảnh.

$$\text{accuracy} = \frac{\text{số lượng hình ảnh tìm được với nhãn đúng}}{\text{tổng số lượng hình ảnh tìm được}}$$

### 3.3.2 Precision-Recall[21]

$$\text{Precision} = \frac{\text{n. of Relevant Retrieved Images}}{\text{n. of Retrieved Images}}$$

$$\text{Recall} = \frac{\text{n. of Relevant Retrieved Image}}{\text{n. of Relevant Images}}$$

### 3.3.3 Mean Average Precision (mAP)[21]

**MAP** được dùng để đánh giá các giải thuật phân loại và phát hiện vật thể. Giá trị mAP nằm trong khoảng từ 0 đến 100. Giá trị càng cao thì chứng tỏ hệ thống càng tốt. **MAP** là giá trị trung bình của Average Precision (AP) trên toàn bộ truy vấn. Ta có công thức tính của AP cho truy vấn thứ  $i$  được như sau:

$$AP_i = \frac{1}{Q_i} \sum_{n=1}^N \frac{R_i^n}{n} t_i^n$$

Trong đó:

- $Q_i$  là tổng số relevant images cho truy vấn thứ  $i$
- $N$  là tổng số hình ảnh của tập tìm kiếm
- $R_i^n$  là số hình ảnh liên quan đã truy vấn trong  $n$  hình ảnh đang xét.
- $t_i^n$  là indicator function,  $t_i^n = 1$  nếu hình ảnh được truy xuất thứ  $n$  là relevant image trong truy vấn thứ  $i$  và  $t_i^n = 0$  ngược lại.

# Chương 4

## Hiện thực và đánh giá kết quả

Với phương pháp đã đề xuất nhóm đã sử dụng các tài nguyên cần thiết, tìm kiếm tập dữ liệu để hiện thực phương pháp đó. Phần này sẽ trình bày chi tiết tài nguyên, tập dữ liệu cũng như quá trình thực hiện model.

### 4.1 Tài nguyên cần thiết

#### 4.1.1 Ngôn ngữ Python

Python là ngôn ngữ lập trình thông dịch, hướng đối tượng, và là ngôn ngữ lập trình cấp cao được tạo ra bởi Guido van Rossum tại Centrum Wiskunde và Informatica (CWI) ở Hà Lan. Python là một ngôn ngữ có cú pháp đơn giản, dễ đọc hiểu, cách viết linh động, được biết như là một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người bắt đầu lập trình và hiện nay đang được sử dụng rộng rãi trong nhiều lĩnh vực với nhiều mục đích khác nhau.

Đặc điểm của ngôn ngữ Python

- Ngôn ngữ lập trình đơn giản, dễ học: Python có cú pháp rất đơn giản. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, ...Python cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.
- Miễn phí, mã nguồn mở: Ta có thể tự do sử dụng và phân phối Python, Python có một cộng đồng rộng lớn hỗ trợ và không ngừng phát triển, cải thiện.
- Là ngôn ngữ dynamic type, quản lý vùng nhớ tự động và hỗ trợ nhiều mô hình lập trình khác nhau như: lập trình hướng đối tượng, lập trình thủ tục và lập trình hàm. Là một ngôn ngữ rất phổ biến và có một lượng lớn các thư viện hỗ trợ như OpenCV, Numpy, Scikit, TensorFlow,...
- Python là ngôn ngữ có thể chạy trên nhiều nền tảng: Python có trên mọi hệ điều hành như Windows, Linux/Unix, OS/2, Mac, Amiga, và những hệ điều hành khác.

Ngoài ra, lý do quan trọng nhất để nhóm chọn ngôn ngữ Python là vì ngôn ngữ Python có các framework cho deep learning như Tensorflow, Keras, các framework vô cùng mạnh mẽ và tiện lợi cho việc hiện thực và đánh mô hình mà nhóm đã lựa chọn.

#### 4.1.2 Các Framework sử dụng trong Luận văn

##### Framework TensorFlow

TensorFlow là một framework mã nguồn mở được sử dụng để xây dựng nên mạng neural do Google phát triển. TensorFlow sử dụng mô hình đồ thị với mỗi nút trong đồ thị thể hiện một



---

hàm toán học và mỗi cạnh thể hiện một ma trận đa chiều. Do việc sử dụng mô hình đồ thị nên framework TensorFlow rất linh hoạt, có thể được triển khai trên nhiều thiết bị như máy chủ, máy tính để bàn, thiết bị di động.

Ngoài ra TensorFlow còn bao gồm Tensorboard là một công cụ rất mạnh giúp ghi lại kết quả và dựng lên biểu đồ từ kết quả ghi lại được. Tensorboard còn đồ thị hoá cấu trúc mô hình được xây dựng từ TensorFlow để giúp người sử dụng dễ dàng kiểm tra lại cấu trúc mô hình đã xây dựng.

## Framework Keras

Keras là một framework mã nguồn mở cho deep learning được viết bằng Python. Nó có thể chạy trên nền của các deep learning framework khác như: TensorFlow (cũng như Theano, CNTK). Keras là một API cấp cao, được phát triển với mục đích để cho các sản phẩm phát triển nhanh. Vì vậy, nếu muốn có kết quả nhanh chóng, Keras sẽ tự động đảm nhiệm các nhiệm vụ cốt lõi và tạo đầu ra. Một trong những khó khăn thông thường với người bắt đầu nghiên cứu deep learning là họ khó có thể nắm bắt được các model phức tạp, thì Keras là lựa chọn phù hợp nhất.

### 4.1.3 Google Colaboratory

Colaboratory là một môi trường Jupyter notebook miễn phí, không yêu cầu thiết lập và chạy hoàn toàn trên cloud. Với Colaboratory, ta có thể viết và thực thi code, lưu và chia sẻ các phân tích của mình và truy cập các tài nguyên tính toán mạnh mẽ được hỗ trợ một server, tất cả đều miễn phí từ trình duyệt.

Môi trường Colaboratory cho những tài nguyên sau:

- 2vCPU @ 2.2GHz.
- 13GB RAM.
- 33GB Free Space.
- Hỗ trợ Python 2 và Python 3.
- Có hỗ trợ tăng tốc bằng GPU miễn phí.
- Những thư viện được cài đặt sẵn. Tất cả các thư viện Python lớn như TensorFlow, Scikit-learn, Matplotlib và nhiều thư viện khác được cài đặt sẵn và sẵn sàng để được import vào.
- Tính năng cộng tác (hoạt động với một nhóm giống như Google Docs): Google Colab cho phép các nhà phát triển sử dụng và chia sẻ sổ Jupyter Notebook với nhau mà không phải tải xuống, cài đặt hoặc chạy bất cứ thứ gì ngoài trình duyệt.
- Hỗ trợ bash command.
- Google Colab notebooks được lưu trên drive.

## 4.2 Hiện thực các module

### 4.2.1 Load Dataset và xử lý Dataset

#### Dataset

Nhóm sử dụng dataset xxx để tiến hành huấn luyện mô hình của mình. Dataset xxx gồm những hình ảnh xxx với tổng cộng xxx hình ảnh. Mỗi hình ảnh trong dataset có kích thước xxx, để phù hợp với mô hình nhóm đề xuất, nhóm tiến hành thu giảm kích thước hình ảnh về kích

---

thước 224 x 224. Nhóm nhận định rằng với xxx hình ảnh là quá ít dữ liệu, không mang tính tổng quát, dễ dẫn đến overfit khi huấn luyện mô hình. Vì vậy nhóm đã tiến hành làm giàu thêm hình ảnh (Agumentation Dataset).

### **Agumentation Dataset**

Sau khi làm giàu hình ảnh xong. Các dataset hiện tại của mô hình gồm:

Tên tập dữ liệu	Kích thước	Số lượng
Train dataset	224 x 224	100
Validation dataset	224 x 224	100
Test dataset	224 x 224	100

Bảng 4.1: Các Dataset trong quá trình huấn luyện mô hình

#### **4.2.2 Huấn luyện mạng CNN**

#### **4.2.3 Tinh chỉnh với Triplet loss**

#### **4.2.4 Trích xuất hình ảnh**

### **4.3 Kết quả**

# Chương 5

## Kết luận

### 5.1 Kết quả đạt được

Với các nội dung trình bày ở các phần trên, nhóm đã hoàn thành được các mục tiêu trong giai đoạn đề cương luận văn này:

- Tìm hiểu được cơ sở lý thuyết về hình ảnh số.
- Tìm hiểu các đặc trưng của hình ảnh số cũng như các giải thuật trích xuất đặc trưng này.
- Tìm hiểu được cơ sở lý thuyết cho mạng nơ-ron, thông qua đó hiểu được hướng áp dụng mạng nơ-ron tích chập vào bài toán
- Hiện thực được một ứng dụng đơn giản để quan sát các kết quả bước đầu
- Biết được các ưu nhược điểm của phương pháp, qua đó hiểu rằng cần phải tìm hiểu thêm để tìm cách khắc phục.

### 5.2 Ưu và nhược điểm của phương pháp đề xuất

#### 5.2.1 Ưu điểm

- Không sử dụng các phương pháp trích xuất đặc trưng thủ công.
- Kết quả thử nghiệm tương đối cao.

#### 5.2.2 Nhược điểm

- Thời gian tính toán và không gian bộ nhớ cần để lưu trữ vector đặc trưng cao.

### 5.3 Hướng mở rộng trong tương lai

Tiếp tục tìm hiểu về lĩnh vực này. Hiện thực các phương pháp đề xuất và thực hiện so sánh với các phương pháp truyền thống. Hiện thực website hoàn thiện để minh họa cho ứng dụng.

# Tài liệu tham khảo

- [1] Isaac Weiss. Digital Images. Truy xuất từ: <https://www.encyclopedia.com/computing/news-wires-white-papers-and-books/digital-imageas>
- [2] Nguyễn Kiên. Tìm hiểu về các định dạng hình ảnh. Truy xuất từ: <https://blogchiasekienthuc.com/thu-thuat-hay/tim-hieu-ve-cac-dinh-dang-hinh-anh.html>
- [3] Vipin Tyagi, *ContentBased Image Retrieval: Ideas, Influences, and Current Trends*, trang 4, 2017.
- [4] Kranthi Kumar. CBIR: Content Based Image Retrieval. *National Conference on Advances in Information Security(NCAIS-2010)*
- [5] Ying Liua, Dengsheng Zhanga. *A survey of content-based image retrieval with high-level semantics*, April 2006
- [6] Adrien Ulens. *CMYK vs. RGB - what is the difference?* Truy xuất từ: <https://www.gogoprint.sg/blog/cmyk-vs-rgb-sg/>
- [7] Vipin Tyagi, *ContentBased Image Retrieval: Ideas, Influences, and Current Trends*, trang 11, 2017
- [8] Vipin Tyagi, *ContentBased Image Retrieval: Ideas, Influences, and Current Trends*, trang 16, 2017
- [9] BLog Code từ đầu. *Neural Network và Deep Learning là gì?* Truy xuất từ: <https://codetudau.com/neural-network-va-deep-learning-la-gi/index.html>
- [10] BLog Deep Learning. *Giới thiệu keras và bài toán phân loại ảnh* Truy xuất từ: <https://nttuan8.com/bai-7-gioi-thieu-keras-va-bai-toan-phan-loai-anh/>
- [11] Imad Dabbura. *Gradient Descent Algorithm and Its Variants*. Truy xuất từ: <https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f66>
- [12] Andrew Ng. *Machine Learning course* Truy xuất từ: <https://www.coursera.org/learn/machine-learning>
- [13] Davi Frossard. VGG in TensorFlow. Truy xuất từ: <https://www.cs.toronto.edu/~frossard/post/vgg16/>
- [14] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large Scale Image Recognition. *ICLR*, 2015.
- [15] Luca Mel. K-means Clustering in Python . Truy xuất từ: <https://medium.com/data-py-blog/k-means-clustering-in-python-4fc8caa0e0d9>
- [16] Ji WAN, Dayong WANG, Steven C. H. HOI, Pengcheng WU, Jianke ZHU, Yongdong ZHANG, and Jintao LI. Deep learning for content-based image retrieval: A comprehensive study. *Research Collection School Of Information Systems, Institutional Knowledge at Singapore Management University*, 2014.
- [17] Rushi Lan, Si Zhong, Zhenbing Liu, Zhuo Shi, and Xiaonan Luo. A simple texture feature for retrieval of medical images. *Multimedia Tools and Applications*, 2017.

- 
- [18] Ling Ma, Xiabi Liu, Yan Gao, Yanfeng Zhao, Xinming Zhao, Chunwu Zhou. A new method of content based medical image retrieval and its applications to CT imaging sign retrieval. *Journal of Biomedical Informatics*, 2017.
  - [19] Maria Tzelepi and Anastasios Tefas. Deep convolutional learning for Content Based Image Retrieval. *Neurocomputing*, 2017.
  - [20] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, Ying Wu. Learning Fine-grained Image Similarity with Deep Ranking. *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
  - [21] Maria Tzelepi, Anastasios Tefas. *Deep convolutional learning for Content Based Image Retrieval*, 16 November 2017