**TRƯỜNG ĐẠI HỌC** **HÀNG HẢI VIỆT NAM**

**KHOA** **CÔNG NGHỆ THÔNG TIN**

**-----\*\*\*-----**



**BÁO CÁO**

**BÀI TẬP LỚN HỌC PHẦN XỬ LÝ ẢNH**

**Mã HP: 17221**

**Lớp HP: N08**

**TÊN BÀI TẬP LỚN**

**XÂY DỰNG CHƯƠNG TRÌNH PHÂN LOẠI ẢNH TRONG MỘT THƯ MỤC ẢNH DỰA TRÊN THỜI GIAN VÀ ĐỘ TƯƠNG ĐỒNG THÀNH CÁC THƯ MỤC CON**

**Sinh viên thực hiện: Nguyễn Lê Trung Kiên**

**Mã SV: 90464**

**Hải Phòng, tháng 04 năm 2024**

**TRƯỜNG ĐẠI HỌC HÀNG HẢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN KHOA HỌC MÁY TÍNH**

**-----\*\*\*-----**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH**

**Mã đề tài: 20**

1. **Tên đề tài**

Xây dựng chương trình phân loại ảnh trong một thư mục ảnh dựa trên thời gian và độ tương đồng thành các thư mục con

1. **Mục đích**

Xây dựng chương trình phân loại ảnh trong một thư mục ảnh.

1. **Công việc cần thực hiện**

* Viết code chương trình đảm bảo chương trình thực hiện đúng yêu cầu bài toán
* Viết báo cáo bài tập lớn
* Bảo vệ bài tập lớn

1. **Yêu cầu**

* Kết quả làm bài tập lớn: Báo cáo bài tập lớn
* Báo cáo bài tập lớn phải được trình bày theo mẫu quy định , báo cáo có thể kết xuất thành tệp định dạng PDF và nộp qua email

**MỤC LỤC**

[I. GIỚI THIỆU VỀ BÀI TOÁN 1](#_Toc168582403)

[1. Sự cần thiết của việc giải quyết bài toán 1](#_Toc168582404)

[2. Các phương pháp giải quyết bài toán 1](#_Toc168582405)

[3. Mục đích của bài toán 2](#_Toc168582406)

[II. CƠ SỞ LÝ THUYẾT VÀ THUẬT TOÁN 3](#_Toc168582407)

[1. Cơ sở lý thuyết 3](#_Toc168582408)

[1.1. Phân cụm (Clustering) 3](#_Toc168582409)

[1.2. Mạng CNN 4](#_Toc168582410)

[1.3. K-means Clustering 11](#_Toc168582411)

[2. Cách thức hoạt động của thuật toán 13](#_Toc168582412)

[2.1. Cách thực hiện 13](#_Toc168582413)

[2.2. Chi tiết các bước 14](#_Toc168582414)

[III. XÂY DỰNG HỆ THỐNG 15](#_Toc168582415)

[1. Ngôn ngữ và nền tảng sử dụng 15](#_Toc168582416)

[1.1. Ngôn ngữ Python 15](#_Toc168582417)

[1.2. Nền tảng sử dụng: Pycharm 15](#_Toc168582418)

[2. Thư viện chính sử dụng 15](#_Toc168582419)

[2.1. Open CV 15](#_Toc168582420)

[2.2. Keras 16](#_Toc168582421)

[2.3. Scikit-Learn (Sklearn) 16](#_Toc168582422)

[3. Cài đặt chương trình 17](#_Toc168582423)

[3.1. Khai báo thư viện 17](#_Toc168582424)

[3.2. Khởi tạo model và tiền xử lý ảnh 18](#_Toc168582425)

[3.3. Trích xuất đặc trưng 18](#_Toc168582426)

[3.4. Phân cụm và sắp xếp ảnh vào thư mục con 19](#_Toc168582427)

[3.5. Giao diện chương trình 21](#_Toc168582428)

[IV. THỰC NGHIỆM 22](#_Toc168582429)

[1. Tập dữ liệu và kết quả thực nghiệm 22](#_Toc168582430)

[KẾT LUẬN 28](#_Toc168582431)

[TÀI LIỆU THAM KHẢO 29](#_Toc168582432)

**DANH MỤC HÌNH ẢNH**

[Hình 1: Ví dụ về phân cụm 3](#_Toc168527804)

[Hình 3: Cấu trúc của mạng CNN 5](file:///C:\Users\user\OneDrive\Desktop\pythonProject\thu_edit.docx#_Toc168527805)

[Hình 4: Kernel size 3 x 3 6](file:///C:\Users\user\OneDrive\Desktop\pythonProject\thu_edit.docx#_Toc168527806)

[Hình 5: Phép tính tích chập 6](file:///C:\Users\user\OneDrive\Desktop\pythonProject\thu_edit.docx#_Toc168527807)

[Hình 5: Ví dụ về stride = 2, padding = 1 7](#_Toc168527808)

[Hình 6: Convolution trên ảnh màu với kernel size 3 x 3 x 3 [5] 8](#_Toc168527809)

[Hình 8: Thực hiện phép tính convolution trên ảnh màu [5] 8](file:///C:\Users\user\OneDrive\Desktop\pythonProject\thu_edit.docx#_Toc168527810)

[Hình 8: Ví dụ về pooling layer 9](#_Toc168527811)

[Hình 9: Làm phẳng thành vector 9](#_Toc168527812)

[Hình 10: Vector sau khi được làm phẳng cho vào fully connected layer 10](#_Toc168527813)

[Hình 11: Kiến trúc của VGG16 [6] 10](#_Toc168527814)

[Hình 12: Kiến trúc của VGG16, conv = convolutional layer, pool= pooling layer, fc= fully connected layer 11](#_Toc168527815)

[Hình 13: Phân cụm K-means 12](#_Toc168527816)

[Hình 14: Ảnh minh họa các bước của thuật toán 13](#_Toc168527817)

[Hình 15: Cách thực hiện thuật toán 14](#_Toc168527818)

[Hình 16: Logo OpenCV 16](#_Toc168527819)

[Hình 17: Logo Keras 16](#_Toc168527820)

[Hình 18: Logo Scikit-learn 17](#_Toc168527821)

[Hình 19: Khai báo thư viện 17](#_Toc168527822)

[Hình 20: Hàm khởi tạo model và tiền xử lý ảnh 18](#_Toc168527823)

[Hình 21: Trích xuất đặc trưng của ảnh 18](#_Toc168527824)

[Hình 22: Phân loại ảnh đầu vào 20](#_Toc168527825)

[Hình 23: Cài đặt giao diện chương trình 21](#_Toc168527826)

[Hình 24: Giao diện khi chương trình chạy 21](#_Toc168527827)

[Hình 25: Hình ảnh về tập dữ liệu 22](#_Toc168527828)

[Hình 26: Một vài ảnh trong bộ dữ liệu 22](#_Toc168527829)

[Hình 27: Chỉ số Silhouette và k tối ưu được chọn 23](#_Toc168527830)

[Hình 28: Kết quả thực hiện dưới dạng biểu đồ 24](#_Toc168527831)

[Hình 29: Đầu ra sau khi thực hiện chương trình với k=12 24](#_Toc168527832)

[Hình 30: Tập ảnh tương đồng của thư mục con cluster10 sau khi phân loại 25](#_Toc168527833)

[Hình 31: Hình ảnh trong tập dữ liệu thực tế 25](#_Toc168527834)

[Hình 32: Chỉ số Silhouette và k tối ưu được chọn trong tập dữ liệu thực tế 26](#_Toc168527835)

[Hình 33: Kết quả thử nghiệm với bộ dữ liệu thực tế dưới dạng biểu đồ 26](#_Toc168527836)

[Hình 34: Kết quả với bộ dữ liệu thực tế 27](#_Toc168527837)

[Hình 35: Hình ảnh tương đồng trong cluster6 27](#_Toc168527838)

# GIỚI THIỆU VỀ BÀI TOÁN

## 1. Sự cần thiết của việc giải quyết bài toán

Sự phát triển của khoa học công nghệ kết hợp với vai trò to lớn của Xử lý ảnh và Thị giác máy tính đã giải quyết được rất nhiều bài toán trong cuộc sống, trong đó có bài toán phân loại ảnh. Về cốt lõi, phân loại ảnh là nhiệm vụ gán nhãn cho hình ảnh.

Hàng ngày, chúng ta không ít lần thực hiện việc phân loại như phân loại đồ vật, hay động vật, thực phẩm và để phân loại được chúng ta thường dựa vào đặc trưng của vật đó cùng với việc so sánh, đánh giá để đưa ra kết quả.

Việc ta thực hiện phân loại thủ công chỉ là một phần nhỏ nên ta có thể phân loại nhanh. Trong thực tế, việc phân loại thường được dùng với một bộ dữ liệu với số lượng rất lớn như phân loại sản phẩm lỗi trên dây truyền sản xuất hay với những dữ liệu phức tạp như chẩn đoán các bệnh thông qua X-quang. Nếu chỉ phân loại thủ công bằng mắt thường sẽ gây hao phí thời gian, sức lực và độ chính xác có thể không cao dẫn đến những sai sót và lại mất thời gian để xem xét lại. May thay với sự tiến bộ trong kỹ thuật thị giác máy tính và các thuật toán học máy mà ta có thể phân loại hình ảnh một cách tự động. Đề tài này tập trung nghiên cứu hệ thống phân loại ảnh tự động trong một thư mục ảnh dựa trên độ tương đồng để phân nhãn cho ảnh và xếp chúng vào các thư mục con, từ đó có thể hướng tới hệ thống phân loại tự động phục vụ vào các mục đích như phân loại hàng hóa thay vì là thư mục đầu vào thì ta có nguồn hàng hóa đủ các loại và cần phân loại chúng tới các đầu ra tương ứng. Vì vậy đề tài có tính cấp thiết, khoa học và thực tế.

## 2. Các phương pháp giải quyết bài toán

Có rất nhiều các tiếp cận trong việc giải quyết bài toán phân loại ảnh và chúng đều có điểm chung là trích chọn các đặc trưng của ảnh và sử dụng đặc trưng để phân loại. Về trích chọn các đặc trưng thì có các phương pháp trích chọn thủ công truyền thống như SIFT, SUF, HOG, .. cho tới sử dụng mạng nơ ron học sâu. Một trong những phương pháp hiệu quả nhất hiện nay là sử dụng các mô hình học sâu như Convolutional Neural Networks (CNN), có khả năng học các đặc trưng phức tạp từ dữ liệu hình ảnh và có thể đạt được độ chính xác cao. Sau khi có các đặc trưng của ảnh thì ta sử dụng các thuật toán phân loại ảnh. Có nhiều thuật toán khác nhau được ứng dụng trong việc phân loại hình ảnh. Các thuật toán này được chia thành hai nhóm chính là Học có giám sát (supervised learning) và Học không giám sát (unsupervised learning). Trong học máy có giám sát, thuật toán được huấn luyện trên một tập hình ảnh đã được dán nhãn. Từ dữ liệu mẫu này, thuật toán có thể trích xuất thông tin, phục vụ phân loại ngay cả những hình ảnh chưa từng nhìn thấy trước đó. Trong học máy không giám sát, thuật toán chỉ sử dụng dữ liệu thô để đào tạo. Các nhãn phân loại thường không xuất hiện trong kiểu học này và mô hình học bằng cách nhận dạng các mẫu trong tập dữ liệu huấn luyện.

## 3. Mục đích của bài toán

Đề tài nhằm xây dựng hệ thống phân loại ảnh tự động với đầu vào là một thư mục ảnh và ta phải phân loại ảnh trong thư mục đó dựa trên độ tương đồng, cùng lúc đó tạo các thư mục con tương ứng để sắp xếp ảnh sau khi phân loại. Kết quả cuối cùng thì thư mục ảnh đầu vào sẽ chứa các thư mục con chứa các ảnh đã phân loại.

Vì vậy, với đề tài **Xây dựng chương trình phân loại ảnh trong một thư mục ảnh dựa trên thời gian và độ tương đồng thành các thư mục con,**  em sẽ thực hiện việc phân loại ảnh trong thư mục ảnháp dụng mạng nơron tích chập (CNN) cụ thể là sử dụng mạng VGG-16 để trích xuất đặc trưng và thuật toán K-means Clustering để phân lớp nhãn ảnh và phân bố chúng vào k thư mục con tương ứng dựa trên độ tương đồng.

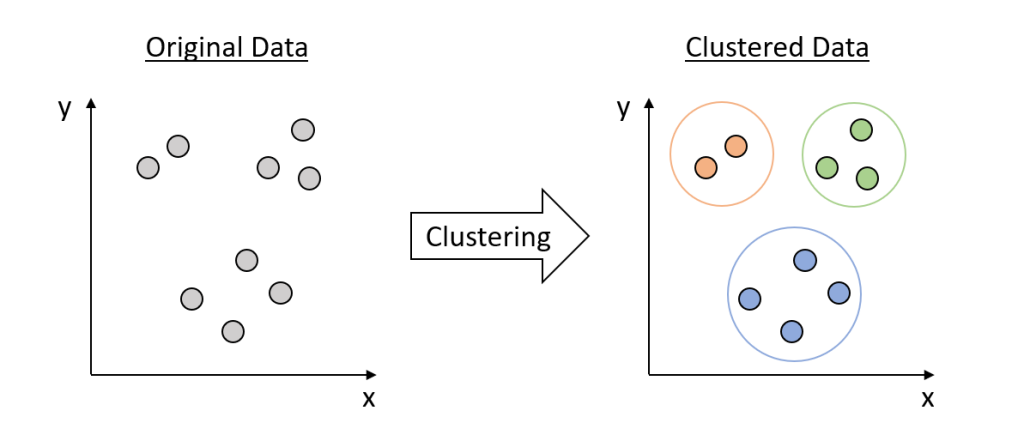
# CƠ SỞ LÝ THUYẾT VÀ THUẬT TOÁN

## Cơ sở lý thuyết

### 1.1. Phân cụm (Clustering)

#### 1.1.1. Giới thiệu

Bài toán phân cụm là 1 nhánh ứng dụng chính của lĩnh vực Unsupervised Learning (Học không giám sát), trong đó đầu vào là tập dữ liệu không có nhãn . Trong trường hợp này, thuật toán sẽ tìm cách phân cụm - chia dữ liệu thành từng nhóm có đặc điểm tương tự nhau, nhưng đồng thời đặc tính giữa các nhóm đó lại phải càng khác biệt càng tốt.



Hình : Ví dụ về phân cụm

Phân cụm nhằm mục đích hình thành các nhóm điểm dữ liệu đồng nhất từ ​​một tập dữ liệu không đồng nhất. Nó đánh giá sự tương đồng dựa trên một số liệu như Euclidean distance, Cosine similarity, Manhattan distance,… rồi nhóm các điểm có điểm tương tự cao nhất lại với nhau [1].

#### 1.1.2. Ứng dụng phân cụm

* Phân khúc thị trường : Các doanh nghiệp sử dụng phân cụm để nhóm khách hàng của họ và sử dụng quảng cáo được nhắm mục tiêu để thu hút nhiều đối tượng hơn [1].
* Phân tích mạng xã hội: Các trang truyền thông xã hội sử dụng dữ liệu của người dùng để hiểu hành vi duyệt web của người dùng và cung cấp cho người dùng các đề xuất nội dung hoặc đề xuất bạn bè [1].
* Ứng dụng trong y tế: Tìm ra các khu vực bị bệnh trong các hình ảnh X-quang [1].
* Sản xuất : Phân cụm được sử dụng để nhóm các sản phẩm tương tự lại với nhau, tối ưu hóa quy trình sản xuất và xác định các khiếm khuyết trong quy trình sản xuất [1].
* Phân đoạn hình ảnh.

#### 1.1.3. Một số phương pháp phân cụm phổ biến

* **Phân cụm dựa trên phân vùng** (Partition-based clustering): Đây là phương pháp phổ biến và được sử dụng nhiều trong các bài toán phân cụm. Mục tiêu là phân dữ liệu thành các phân vùng khác nhau.
* **Phân cụm thứ bậc** (Hierarchical clustering): Ngoài việc phân thành các cụm lớn, phương pháp này còn phân các cụm lớn thành những cụm nhỏ hơn dưới dạng thứ bậc.
* **Mô hình hỗn hợp** (Mixture models).
* **Phân cụm sâu** (Deep clustering): Sử dụng mạng nơ-ron học sâu để phân cụm.

#### 1.1.4. Đánh giá chất lượng mô hình cụm

Để đánh giá chất lượng mô hình phân cụm ta có thể đánh giá thông qua một số phương pháp như sau:

* Khoảng cách / sự khác biệt giữa hai cụm bất kỳ phải lớn. Giữa các cụm phải được tách biệt nhau hoàn toàn và sự khác biệt giữa 2 cụm phải đủ lớn để phân biệt 2 cụm với nhau [2].
* Chênh lệch giữa các điểm dữ liệu bên trong một cụm phải nhỏ. Chênh lệch ở đây thể hiện sự khác biệt với nhau về mặt tương đồng giữa 2 dữ liệu theo tiêu chí phân cụm [2].

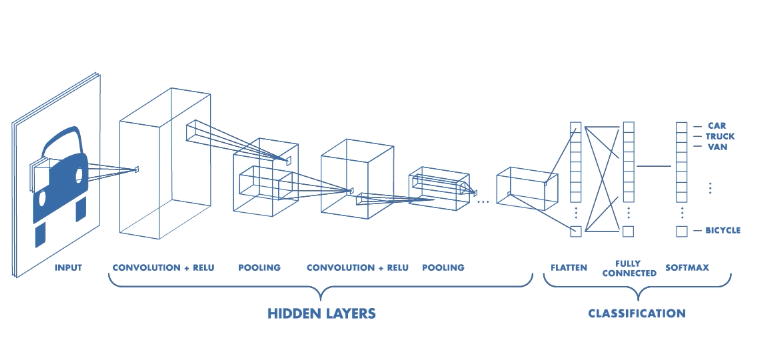
### 1.2. Mạng CNN

#### 1.2.1. Giới thiệu

Mạng nơ-ron tích chập (CNN) là một kiến trúc mạng nơ-ron đặc biệt dành cho việc xử lý dữ liệu hình ảnh và giúp máy tính hiểu và phân tích hình ảnh một cách tự động. Sự ra đời của CNN đã thúc đẩy quá trình của ngành thị giác máy tính nói riêng và ngành khoa học máy tính nói chung.

#### 1.2.2. Cấu trúc của mạng CNN

CNN bao gồm tập hợp các lớp cơ bản bao gồm: convolution layer, pooling layer, fully connected layer. Các lớp này liên kết với nhau theo một thứ tự nhất định. Thông thường, một ảnh sẽ được lan truyền qua tầng convolution layer, sau đó các giá trị tính toán được sẽ lan truyền qua pooling layer, convolution layer + pooling layer có thể được lặp lại nhiều lần trong network. Và sau đó được lan truyền qua tầng fully connected layer và softmax để tính xác suất ảnh đó chứa vật thế gì. [3]



H

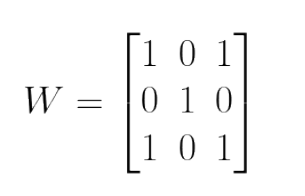
Hình : Cấu trúc của mạng CNN

**a, Một số khái niệm:**

**- Tensor**: Khi biểu diễn dữ liệu 1 chiều, người ta gọi là vector, với dữ liệu 2 chiều thì được gọi là matrix, với dữ liệu nhiều hơn 2 chiều được gọi là tensor. Ví dụ: tensor 3 chiều, …

**- Phép nhân chập (convolution)**: kỹ thuật quan trọng trong Xử Lý Ảnh, được sử dụng để làm mở ảnh, làm nét ảnh, xác định các đường, …

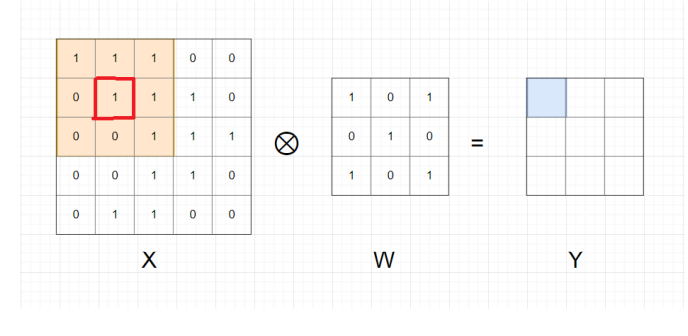
Cách tính : Ta định nghĩa kernel là một ma trận vuông kích thước k\*k trong đó k là số lẻ. k có thể bằng 1, 3, 5, 7, 9,... [4]



Hình : Kernel size 3 x 3

Kí hiệu phép tính convolution (⊗), kí hiệu Y = X ⊗W. [4]

Với mỗi phần tử xi, j trong ma trận X lấy ra một ma trận có kích thước bằng kích thước của kernel W có phần tử xi, j làm trung tâm (đây là vì sao kích thước của kernel thường lẻ) gọi là ma trận A. Sau đó tính tổng các phần tử của phép tính element-wise của ma trận A và ma trận W, rồi viết vào ma trận kết quả Y [4].



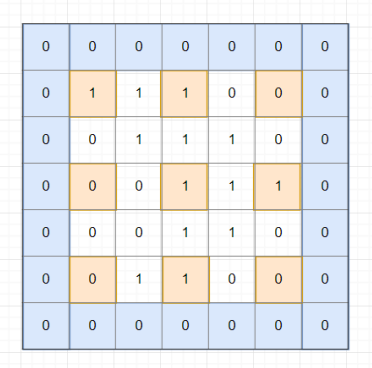
Hình : Phép tính tích chập

Ví dụ khi tính tại x22 (ô khoanh đỏ trong hình), ma trận A cùng kích thước với W, có x22 làm trung tâm có màu nền da cam như trong hình. Sau đó tính y11 = sum(A⊗W) = x11 ∗w11 +x12 ∗w12 + x13 ∗w13 +x21 ∗w21 +x22 ∗w22 +x23 ∗w23 +x31 ∗w31 +x32 ∗w32 +x33 ∗w33 = 4. Và làm tương tự với các phần tử còn lại trong ma trận. [4]

- **Padding**: Sau khi thực hiện phép nhân chập thì ma trận cho ra đều nhỏ hơn ma trận đầu vào (do viền không tính được). Để thu được ma trận với đầu ra bằng ma trận đầu vào thì ta cần tìm cách giải quyết các giá trị ở viền bằng cách bù các giá trị ở ngoài viền.

- **Bước nhảy (Stride)** : Giả sử stride =1 , thì ta thực hiện phép tính convolution trên từng phần tử trong ma trận.

Tuy nhiên nếu stride = k (k > 1) thì ta chỉ thực hiện phép tính convolution trên các phần tử x1 + i ∗ k, 1 + j ∗ k.



Hình : Ví dụ về stride = 2, padding = 1

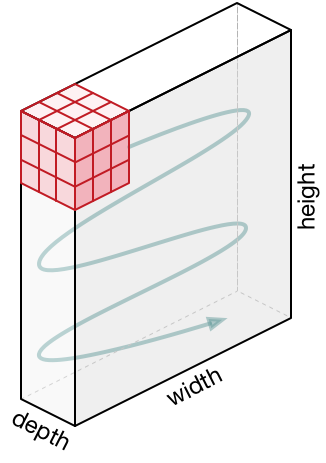
Stride thường được sử dụng để giảm kích thước ma trận sau phép tính convolution.

- **Hàm kích hoạt: ReLU**

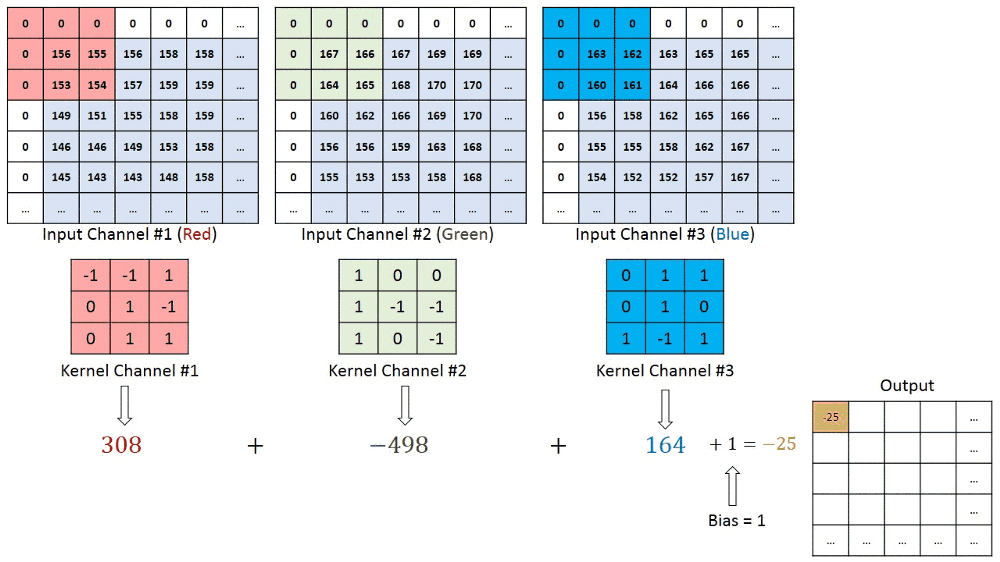
ReLU (Rectified Linear Units, f = max(0, x)) là hàm kích hoạt phổ biến nhất cho CNN. ReLu được ưa chuộng vì tính đơn giản và cho ra hiệu quả tốt. ReLu được đặt ngay sau tầng convolution layer và nó sẽ gán 0 với giá trị âm và giữ nguyên giá trị nếu giá trị đó lớn hơn 0.

**b, Convolution layer:**

Convolution layer (lớp tích chập) là lớp đầu tiên trích xuất các đặc tính từ hình ảnh. Tham số lớp này bao gồm một tập hợp các bộ lọc có thể học được. Các bộ lọc đều nhỏ thường có kích cỡ hai chiều đầu tiên khoảng 3x3 hoặc 5x5, .... và có độ sâu bằng với độ sâu của đầu vào. Bằng cách trượt dần bộ lọc theo chiều ngang và dọc trên ảnh, chúng thu được một Feature Map chứa các đặc trưng được trích xuất từ trên hình ảnh đầu vào.



Hình : Convolution trên ảnh màu với kernel size 3 x 3 x 3 [4]

****

Hình : Thực hiện phép tính convolution trên ảnh màu [4]

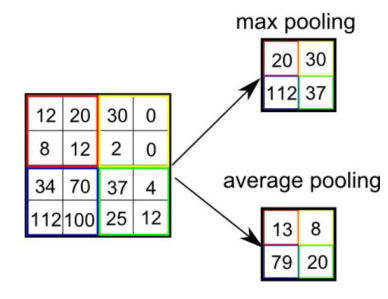
Với mỗi kernel khác nhau thì sẽ thu được các đặc trưng khác của ảnh. Vì vậy, trong convolution layer thường sử dụng nhiều kernel để học được nhiều thuộc tính của ảnh.

**c, Pooling layer**

Pooling layer thường được dùng giữa các convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Việc giảm kích thước dữ liệu giúp giảm các phép tính toán trong model. [4]

Thông thường khi pooling thì dùng kernel kích thước (2x2).

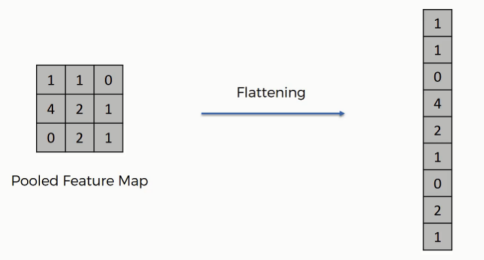
Có 2 loại pooling layer phổ biến là: max pooling và average pooling.

****

Hình : Ví dụ về pooling layer

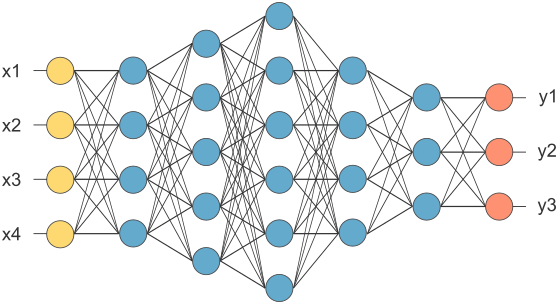
**d, Fully connected layer:**

Sau khi ảnh được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của ảnh (ví dụ mắt, mũi, khung mặt,...) thì tensor của output của layer cuốicùng, kích thước H\*W\*D, sẽ được chuyển về 1 vector kích thước (H\*W\*D). [4]

****

Hình : Làm phẳng thành vector

Sau đó ta dùng các fully connected layer để kết hợp các đặc điểm của ảnh để ra được output của model.

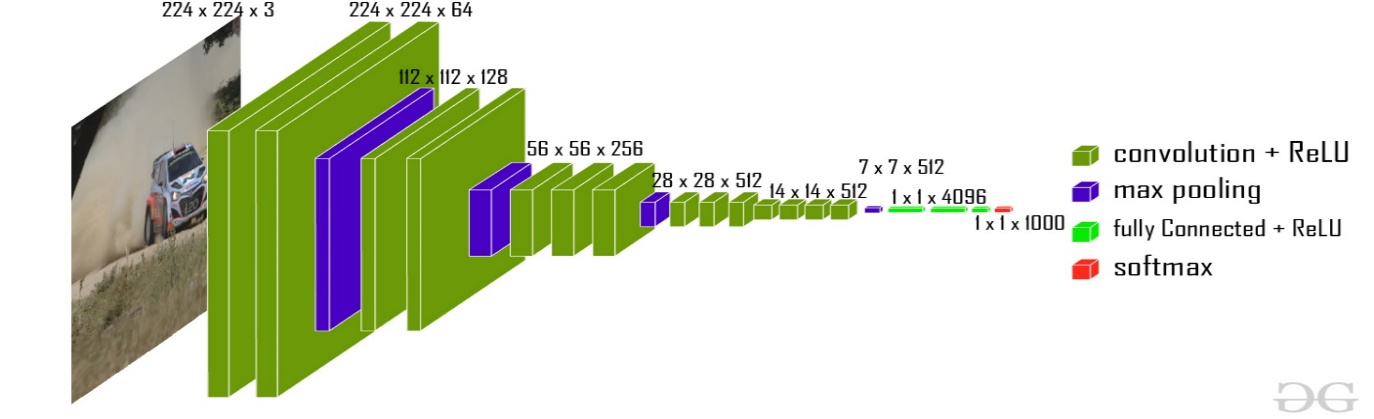


Hình : Vector sau khi được làm phẳng cho vào fully connected layer

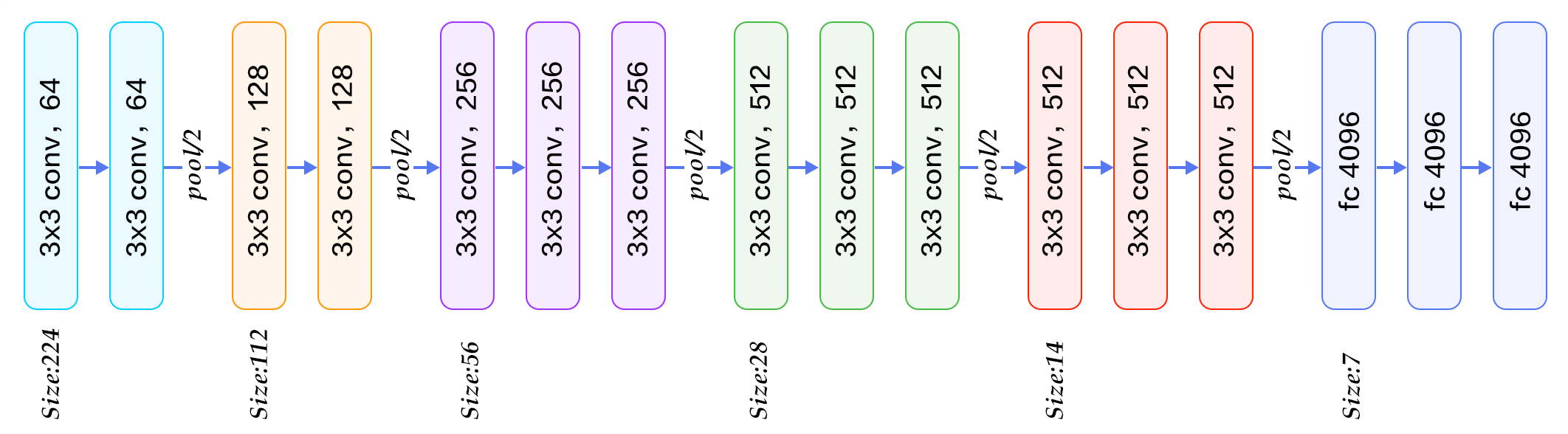
Trong bài toán phân loại, layer cuối cùng được áp dụng với hàm softmax phục vụ phân loại cho output đầu ra có kích thước n tương đương với n lớp mà ta muốn phân lớp. Vì vậy chỉ cần bỏ layer cuối cùng thì ta sẽ thu được vector đặc trưng được sử dụng trong bài này.

**e, VGG-16:**

VGG-16 là mô hình mạng CNN được đề xuất bởi Karen Simonyan và Andrew Zisserman thuộc Phòng thí nghiệm hình học Trực quan của đại học Oxfold vào năm 2014. Mô hình này bao gồm 16 lớp gồm 13 lớp tích chập và 3 lớp kết nối đầy đủ. VGG-16 nổi tiếng vì tính đơn giản và hiệu quả cũng như khả năng đạt được hiệu suất mạnh mẽ trong các tác vụ thị giác máy tính khác nhau, bao gồm phân loại hình ảnh và nhận dạng đối tượng.



Hình : Kiến trúc của VGG16 [5]



Hình : Kiến trúc của VGG16, conv = convolutional layer, pool= pooling layer, fc= fully connected layer

Với VGG-16 , input đầu vào có kích thước (224, 224, 3) đi qua các khối với mỗi khối gồm nhiều lớp convolutional layer với size 3 x 3 áp dụng same padding rồi sau đó đi qua max poling layer với size 2 x 2 . Càng các convolutional layer sau thì kích thước width, height càng giảm nhưng depth hay số kernel áp dụng trong layer đó ngày càng tăng (64, 128, 256, 512). Sau khá nhiều convolutional layer và pooling layer thì dữ liệu được flatten và cho vào fully connected layer.

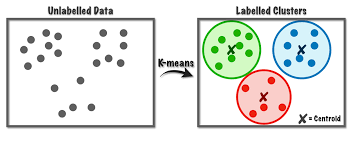
Mặc dù đơn giản so với các kiến ​​trúc gần đây hơn, VGG-16 vẫn là lựa chọn phổ biến cho nhiều ứng dụng deep learning do tính linh hoạt và hiệu suất tuyệt vời của nó. VGG-16 đã được tích hợp trong thư viện kerras và được sử dụng trong bài để trích xuất đặc trưng của ảnh.

### 1.3. K-means Clustering

#### 1.3.1. Tổng quan

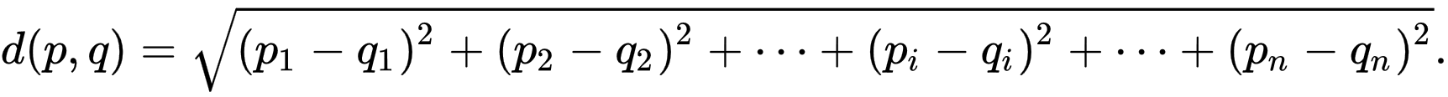
Thuật toán phân cụm K-means được giới thiệu năm 1957 bởi Lloyd K-means và là phương pháp phổ biến nhất cho việc phân cụm, dựa trên việc phân vùng dữ liệu.  
Đầu vào là tập dữ liệu: D = {x1, x2, …, xr}, với xi là vector n chiều trong không gian Euclidean. K-means phân cụm D thành K cụm dữ liệu:

* Mỗi cụm dữ liệu có một điểm trung tâm gọi là centroid.
* K là một hằng số cho trước.

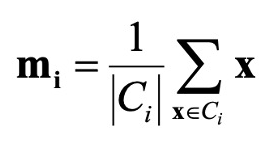


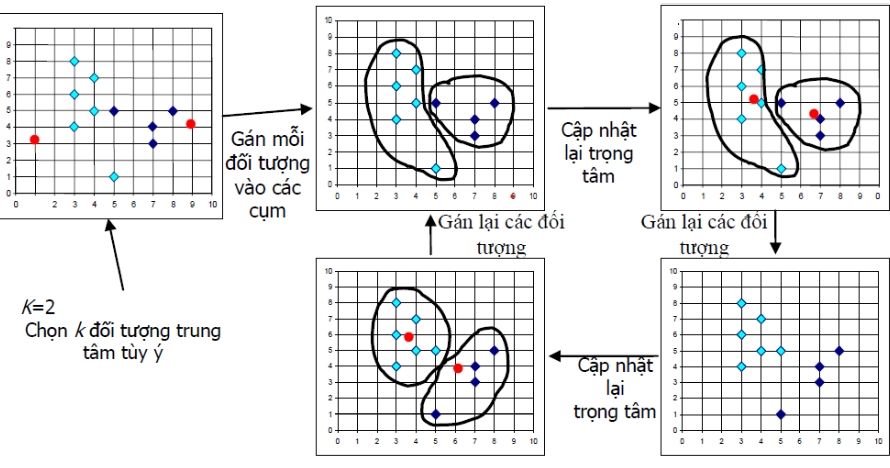
Hình : Phân cụm K-means

#### 1.3.2. Các bước trong thuật toán K-Means

* **Đầu vào**: Cho tập dữ liệu D, với K là số cụm
* **Khởi tạo**: Khởi tạo K điểm dữ liệu ngẫu nhiên thành centroid ( K điểm này có thể trong D)
* **Lặp lại** các bước sau đến khi **hội tụ**:
  + **Bước 1**: Với mỗi điểm dữ liệu, gán điểm dữ liệu đó vào cluster có khoảng cách đến điểm trung tâm của cluster là nhỏ nhất. Trong bài này ta sử dụng khoảng cách Euclid. Giả sử điểm p có tọa độ là (p1, p2), q có tọa độ là (q1, q2). Khi đó khoảng cách Euclid giữa điểm p và q là:

Ngoài ra ta có thể sử dụng các phép đo khoảng cách khác.

* + **Bước 2*:*** Với mỗi cluster, xác định lại điểm trung tâm của tất cả các điểm dữ liệu được gán vào cluster đó. Ta sẽ tính điểm trung tâm của 1 cụm bằng cách tính trung bình cộng tất cả các điểm dữ liệu trong cụm đó. Trong công tính điểm trung tâm dưới thì mi là điểm trung tâm của cluster Ci và x là điểm dữ liệu thuộc cluster đó.



Hình : Ảnh minh họa các bước của thuật toán

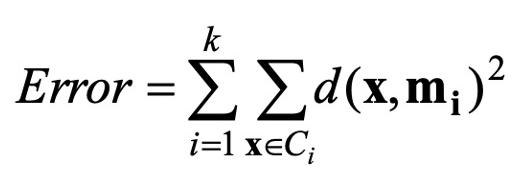
#### 1.3.3. Điểu kiện hội tụ ( điều kiện dừng của thuật toán )

Ta sẽ xác định điều kiện dừng thuật toán theo một số cách như sau:

+, Tại 1 vòng lặp: có ít các điểm dữ liệu được gán sang cluster khác [2].

+, Điểm trung tâm (centroid) không thay đổi nhiều [2].

+, Giá trị hàm mất mát không thay đổi nhiều, công thức dưới tính giá trị hàm mất mát là tổng bình phương khoảng cách của tất cả các điểm với cluster của nó [2]:

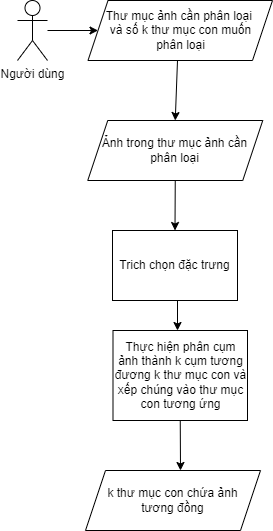


Trong đó Ci là cluster thứ i, mi­ là điểm trung tâm của cluster tương ứng [2].

Nhìn chung về điều kiện hội tụ có thể thấy mối liên hệ giữa các điều kiện là gần tương đồng như nhau. Khi có ít điểm dữ liệu được gán sang cluster khác có thể khiến điểm trung tâm không thay đổi nhiều và từ đó hàm mất mát cũng sẽ ít bị ảnh hưởng. Vậy nên chúng ta có thể sử dụng 1 trong 3 cách trên để xác định điều kiện dừng của thuật toán [2].

## 2. Cách thức hoạt động của thuật toán

### 2.1. Cách thực hiện



Hình : Cách thực hiện thuật toán

### Chi tiết các bước

Người dùng sẽ đưa input đầu vào gồm số k là số thư mục con mà người dùng muốn sau khi phân loại ảnh và đường dẫn của thư mục ảnh đầu vào.

+ **Trích chọn đặc trưng** : Ta trích xuất vector đặc trưng của ảnh trong thư mục đầu vào thông quan CNN model, cụ thể là sử dụng mạng VGG-16. Kết quả là thư mục ảnh có N ảnh thì ta sẽ thu được N vector đặc tương ứng.

+ **Phân cụm ảnh và xếp ảnh vào thư mục tương ứng** : Ta thực hiện thuật toán K- means Clustering với đầu vào là một danh sách chứa vector đặc trưng của thư mục ảnh đầu vào và giá trị k do người dùng nhập để tạo ra k thư mục con. Sau đó ta thực hiện xếp ảnh vào thư mục con tương ứng theo nhãn mà ảnh được nhận sau khi thực hiện thuật toán K-means Clustering.

# XÂY DỰNG HỆ THỐNG

## 1. Ngôn ngữ và nền tảng sử dụng

### 1.1. Ngôn ngữ Python

Python là ngôn ngữ lập trình bậc cao, được ra mắt vào năm 1991 do Guido van Rossum tạo ra. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Một số cách cài đặt và sử dụng Python:

* Tải file cài đặt python tại: <https://www.python.org/downloads/> , chọn version bạn muốn cài đặt và tải xuống, sau khi tải xong thực hiện chạy python và sử dụng.
* Tải các môi trường IDE tích hợp python như : Pycharm, Eclipse – Pydev,.…

### 1.2. Nền tảng sử dụng: Pycharm

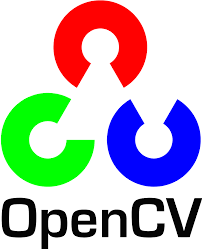
PyCharm là một phần mềm được phát triển bởi JetBrains, cung cấp các công cụ cần thiết giúp các lập trình viên Python tăng năng suất làm việc. Ngoài ra, PyCharm còn được tích hợp nhiều yếu tố mở rộng khác như: biên dịch mã, tô sáng cú pháp, điều hướng project nhanh chóng, công cụ cơ sở dữ liệu và trình soạn thảo văn bản có tích hợp lập trình nhằm mục đích thúc đẩy quá trình phát triển Website.

## 2. Thư viện chính sử dụng

### 2.1. Open CV

OpenCV là một thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), xử lý ảnh và máy học, và các tính năng tăng tốc GPU trong hoạt động thời gian thực.

OpenCV hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot.



Hình : Logo OpenCV

### 2.2. Keras

Keras là một thư viện mã nguồn mở được sử dụng rộng rãi trong lĩnh vực deep learning (học sâu) và mạng nơ-ron. Nó được thiết kế để giúp các nhà phát triển xây dựng và thử nghiệm các mô hình học sâu một cách dễ dàng và linh hoạt.

Mục tiêu chính của Keras là cung cấp một API đơn giản, trực quan và dễ sử dụng, giúp người dùng tập trung vào việc xây dựng mô hình mà không cần lo lắng về chi tiết kỹ thuật phức tạp.

Một trong những điểm mạnh của Keras là khả năng tích hợp với các framework học sâu khác như TensorFlow và Theano, cho phép người dùng tận dụng sức mạnh của cả hai thế giới. Đây chính là điều giúp Keras trở thành một lựa chọn hàng đầu cho các dự án học sâu từ phân loại ảnh, dự đoán chuỗi thời gian, đến các ứng dụng trong lĩnh vực ngôn ngữ tự nhiên và nhiều lĩnh vực khác.



Hình : Logo Keras

### 2.3. Scikit-Learn (Sklearn)

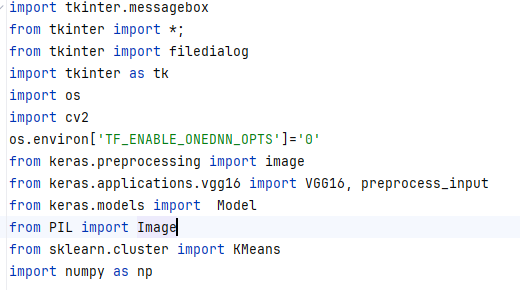
Scikit-learn (Sklearn) là thư viện hữu ích và mạnh mẽ nhất cho Machine Learning bằng Python. Nó cung cấp một loạt các công cụ hiệu quả để học máy và lập mô hình thống kê bao gồm phân loại, hồi quy, phân cụm và giảm kích thước thông qua giao diện nhất quán trong Python. Thư viện này, phần lớn được viết bằng Python, được xây dựng dựa trên NumPy, SciPy và Matplotlib .



Hình : Logo Scikit-learn

## 3. Cài đặt chương trình

### 3.1. Khai báo thư viện



Hình : Khai báo thư viện

* tkinter: Xử lý về giao diện chương trình
* os : Thư viện thao tác tệp, thư mục như tạo file, thư mục kiểm tra file, thư mục có tồn tại hay không
* cv2: Thư viện của OpenCV thực hiện các thao tác đọc, ghi, xử lý ảnh
* keras: Thư viện keras phục vụ tạo các model, trong applications của keras đã tích hợp sẵn các mạng như VGG16, VGG19,…
* PIL: Thư viện xử lý hình ảnh
* numpy: Thư viện toán học phổ biến và mạnh mẽ của python, làm việc hiệu quả với mảng, ma trận lớn, đa chiều.
* sklearn: thư viện cung cấp các thuật toán trong machine learning.

### 3.2. Khởi tạo model và tiền xử lý ảnh

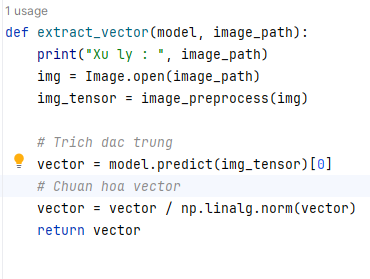


Hình : Hàm khởi tạo model và tiền xử lý ảnh

Hàm get\_extract\_model () : Thực hiện khởi tạo và trả về model VGG16 với đầu vào tensor có kích thước ( 224, 224, 3) và đầu ra là layer gần cuối ( sau khi bỏ layer cuối cùng).

Hàm image\_preprocess () : Tiền xử lý ảnh thành tensor phù hợp với yêu cầu của model. Tham số img được chuyền là ảnh được đọc bởi thư viện PIL, kích thước ảnh được chuyển thành (224,224), convert sang RGB và sau đó chuyển về mảng numpy và lưu trữ tại biến x. Kết quả trả về là tensor phục vụ làm input cho model.

### 3.3. Trích xuất đặc trưng



Hình : Trích xuất đặc trưng của ảnh

Hàm trả về vector đặc trưng với đầu vào là model đã khởi tạo và đường dẫn của ảnh. Ảnh được đọc bởi đối tượng Image trong thư viện PIL, tiền xử lý thành tensor và thực hiện trích xuất đặc trưng (model.predict) thành vector đặc trưng. Sau đó vector đặc trưng được chuẩn hóa theo phương pháp Unit Length.

### 3.4. Phân cụm và sắp xếp ảnh vào thư mục con

Hàm phân loại thực hiện nhận đầu vào là đường dẫn của thư mục cần phân loại và số k tương ứng với số thư mục con đầu ra sau khi phân loại.

+, Tại vòng lặp đầu tiên, ta thực hiện duyệt file ảnh trong thư mục ảnh đầu vào thông qua hàm os.lisdir với tham số truyền vào là đường dẫn thư mục:

Trước khi bắt đầu vòng lặp, khởi tạo ba danh sách chứa tên ảnh, đường dẫn của ảnh và vector đặc trưng của ảnh. Với mỗi hình ảnh được duyệt trong vòng lặp, ta trích xuất đặc trưng của hình ảnh và lưu trữ trong danh sách chứa vector đặc trưng, đồng thời lưu trữ tên và đường dẫn hình ảnh. Kết thúc vòng lặp ta thu được vector đặc trưng có số lượng là N tương đương với N ảnh và tham số k đã được truyền vào làm input đầu vào để thực hiện thuật toán K-means Clustering.

Lúc này ta thực hiện thuật toán K-means của thư viện sklearn có các tham số như n\_cluster = số lượng cụm, init ={k-mean++, random}: thiết lập trọng tâm ban đầu, max\_iter: số lần lặp tối đa cho 1 lần chạy thuật toán, n\_init: số lần chạy thuật toán với các centroid khác nhau và sẽ đưa ra kết quả tốt nhất. Sau khi thực hiện xong k-means ta sẽ lấy các thuộc tính như tọa độ của k centroid hoặc vector chứa n nhãn cụm của ảnh.

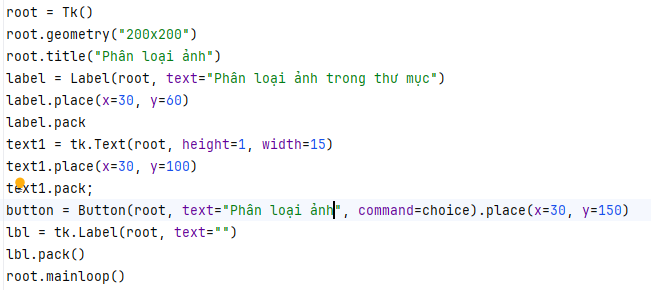
+, Tại vòng lặp thứ 2, ta thực hiện sắp xếp ảnh vào thư mục con, nếu label của ảnh được xếp chưa có thư mục con thì ta thực hiện khởi tạo thư mục thông qua hàm os.makedirs(). Sau đó, thực hiện việc đọc và ghi ảnh vào thư mục con này thông qua thư viện OpenCV ( cv2.imread(), cv2.imwrite()).





Hình : Phân loại ảnh đầu vào

### 3.5. Giao diện chương trình



Hình : Cài đặt giao diện chương trình

Sử dụng thư viện tkinter

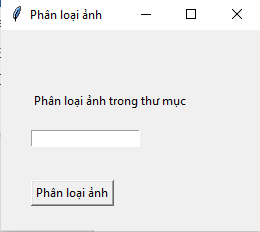
Tk(): thực hiện tạo cửa sổ.

Geometry(): Thiết lập kích thước cho cửa sổ.

Title(): Tạo tiêu đề cho cửa sổ.

Label(), Button(), Text (): Tạo các đối tượng hiển thị trong cửa sổ, trong label có thuộc tính text chứa tên của label sẽ được hiển thị và button chứa command = function có ý nghĩa khi button được nhấn thì function sẽ được thực hiện. Đối tượng Text giúp người dùng nhập các kí tự từ bàn phím và trong đề tài này đối tượng này sẽ giúp người nhập số k thư mục đầu ra mong muốn.

Việc chạy ứng dụng được thực hiện thông qua phương thức mainloop().



Hình : Giao diện khi chương trình chạy

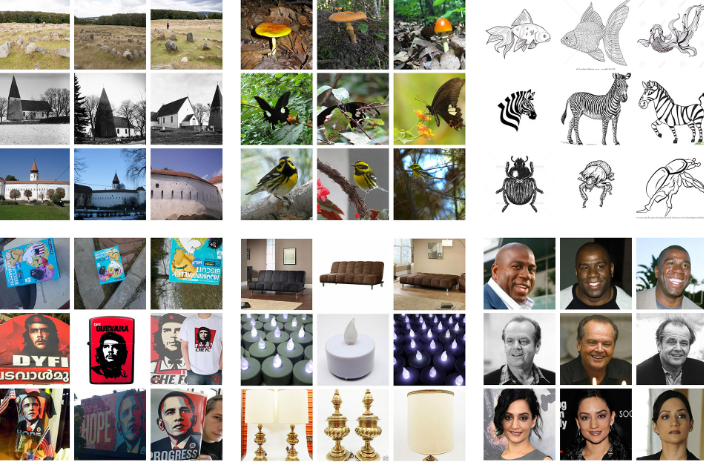
# THỰC NGHIỆM

## 1. Tập dữ liệu và kết quả thực nghiệm

Tập dữ liệu phục vụ cho bài toán được xây dựng từ bộ GPR1200 Dataset do MATHURIN ACHÉ update 2 năm trước tính tới thời điểm hiện tại trên Kaggle.

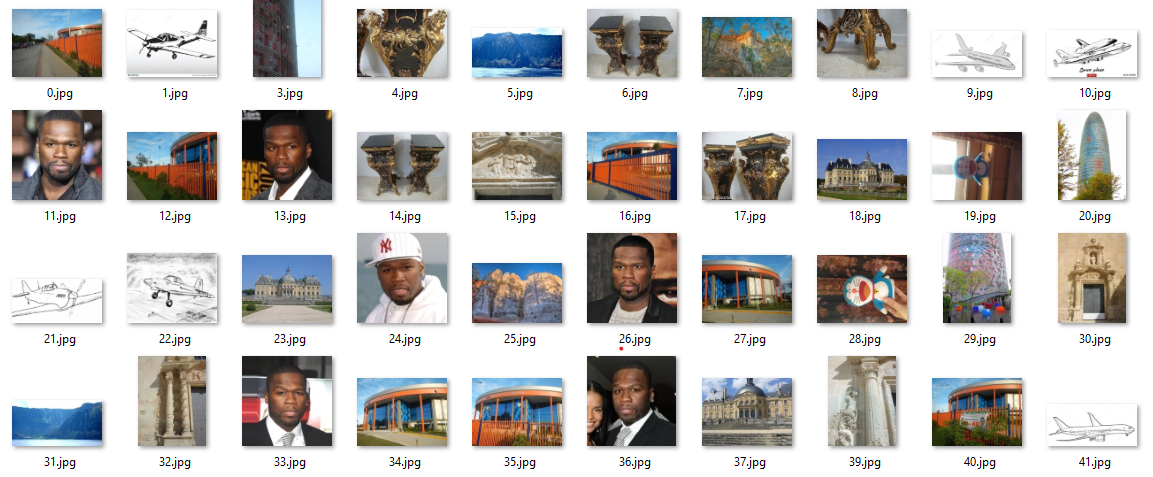
Link download: <https://www.kaggle.com/datasets/mathurinache/gpr1200-dataset/data>

Tập dữ liệu bao gồm 1200 đối tượng và mỗi đối tượng có 10 hình ảnh tương đồng về đối tượng đó.



Hình : Hình ảnh về tập dữ liệu

Ta xây dựng và thực nghiệm bộ dữ liệu nhỏ gồm 120 ảnh của 12 đối tượng.



Hình : Một vài ảnh trong bộ dữ liệu

Với k-means, việc xác định bao nhiêu cụm được phân tương đương với số thư mục con đầu ra trong bài toán rất quan trọng sao cho các cụm có được sau khi kết thúc thuật toán được đánh giá phù hợp, chính xác. Việc lựa chọn tham số k ban đầu để kiểm soát quá trình phân cụm và phân tích cụm sao cho tối ưu [7].

Có rất nhiều cách chọn k hiểu quả, nó kết hợp không chỉ dựa trên phương pháp mà còn kinh nghiệm của chuyên gia phân tích [7]. Trong bài, ta sẽ thử sử dụng Silhouette Coefficient để tìm số k sao cho hiệu quả. Cách tính như sau:

Cho điểm I:

Tính a = Trung bình khoảng cách từ điểm I tới các điểm khác trong cụm

Tính b = min ( Trung bình khoảng cách từ điểm I tới các điểm ở trong cụm khác)

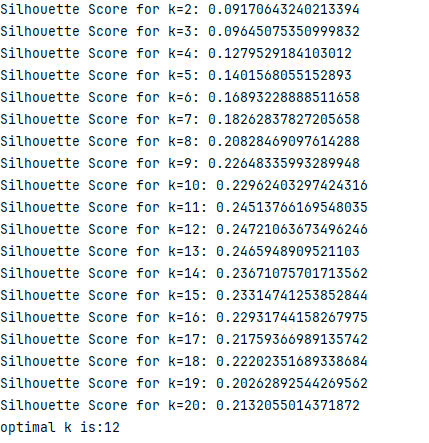
Silhouette score của điểm I là s =

Hệ số này trong khoảng từ [0, 1], càng gần 1 thì chất lượng phân cụm càng tốt [8].

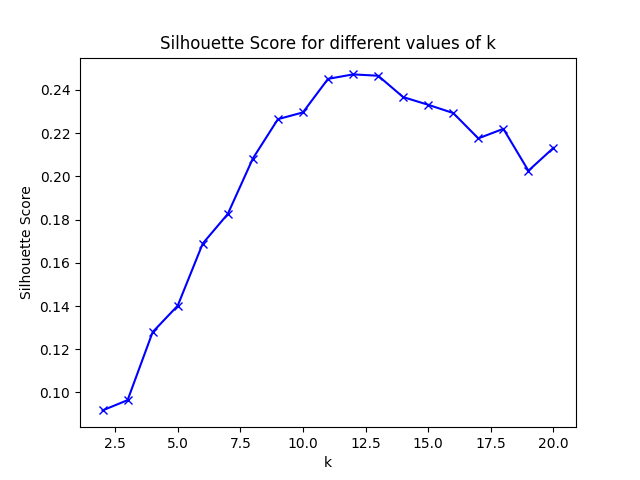
Có thể tính silhouette score trung bình của 1 cụm hoặc cả phân cụm [8].

Do càng gần 1 thì chất lượng phân cụm càng tốt thì mỗi k khác nhau ta sẽ tính silhouette score và chọn k sao cho silhouette score là lớn nhất. Trong thư viện sklearn đã có sẵn hàm silhouette\_score để hỗ trợ trong việc tính toán.

Thực nghiệm thuật toán với k từ 2 tới 20:



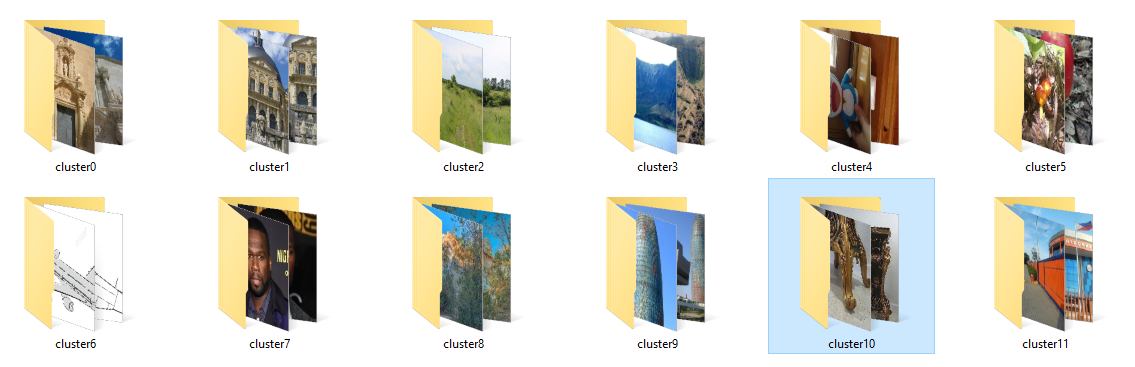
Hình : Chỉ số Silhouette và k tối ưu được chọn



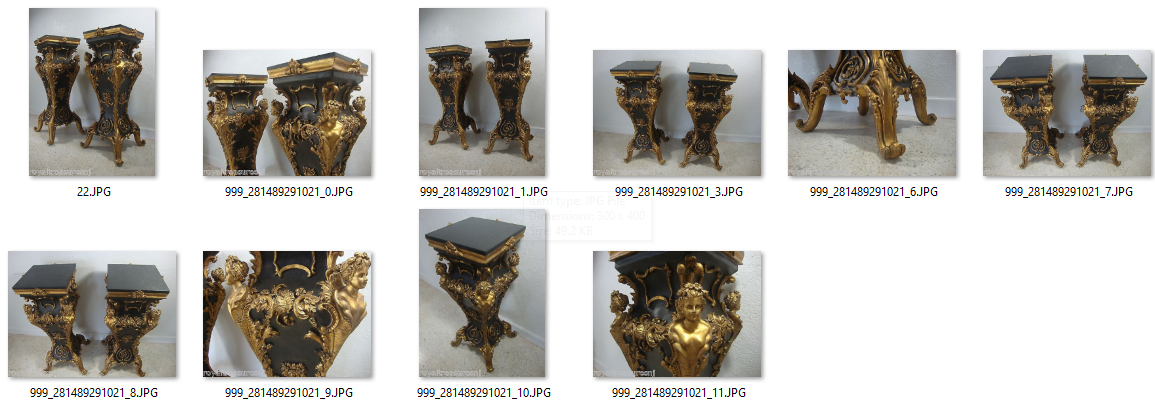
Hình : Kết quả thực hiện dưới dạng biểu đồ

Dựa vào kết quả ta nhận được k = 12 và kết quả này đúng với số đối tượng mà ta đã khai báo trước đó, vì vậy ta sẽ thực nghiệm thuật toán với tham số trên.

Kết quả sau khi thực hiện phân loại với k = 12 tương đương với số thư mục đầu ra:

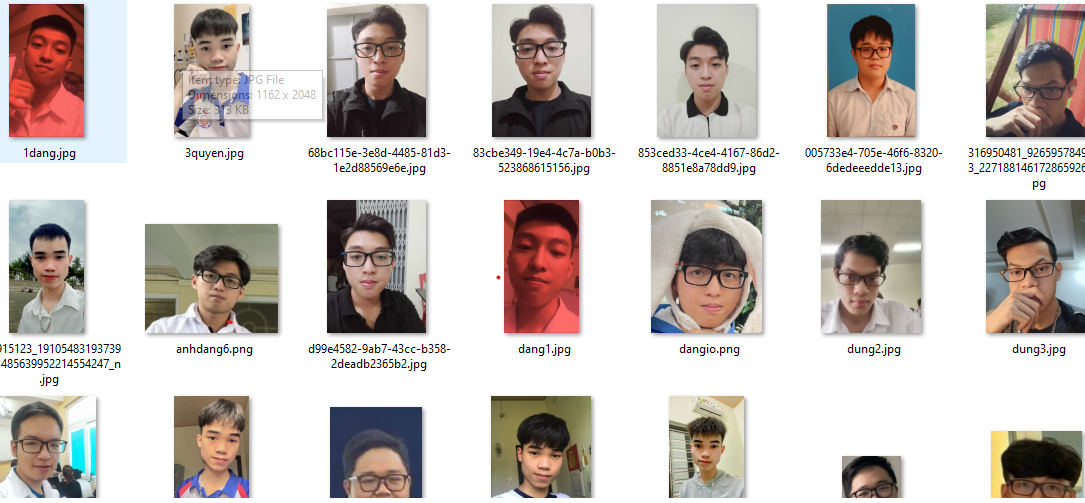


Hình : Đầu ra sau khi thực hiện chương trình với k=12



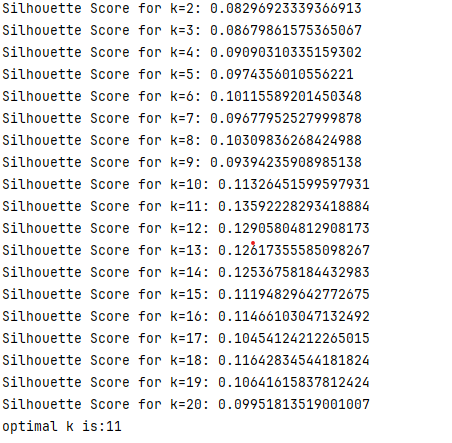
Hình : Tập ảnh tương đồng của thư mục con cluster10 sau khi phân loại

Với bộ dữ liệu đầu vào này, chương trình mất 95 giây để phân loại. Chương trình cho ra kết quả khá tốt với độ chính xác cao dựa trên quan sát. Tuy nhiên đây là những hình được thu thập trên mạng và có chất lượng tốt nên thuật toán phân cụm hiệu quả. Tiếp theo, chúng ta sẽ thử nghiệm với bộ dữ liệu thực tế. Bộ dữ liệu này gồm những hình ảnh của học sinh K62 Khoa Công nghệ thông tin với số lượng là 29 hình ảnh.

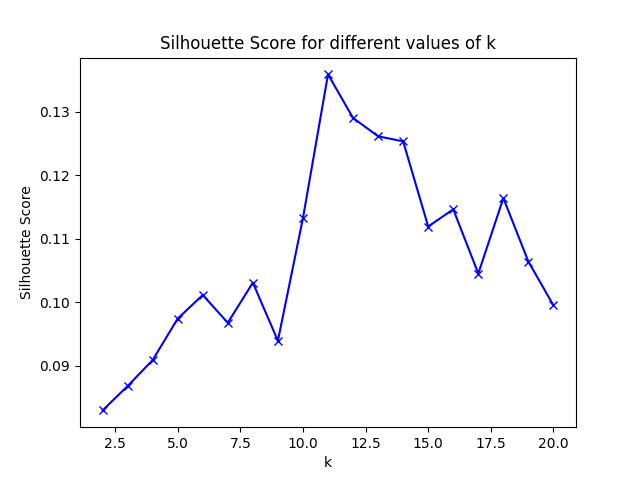


Hình : Hình ảnh trong tập dữ liệu thực tế

Ta tiếp tục thử nghiệm và chọn k tối ưu:



Hình : Chỉ số Silhouette và k tối ưu được chọn trong tập dữ liệu thực tế



Hình : Kết quả thử nghiệm với bộ dữ liệu thực tế dưới dạng biểu đồ

Ta xác định k = 11 cho ra chất lượng các cụm là tốt nhất nên ta sẽ chạy thuật toán cho ra kết quả đầu ra là :



Hình : Kết quả với bộ dữ liệu thực tế



Hình : Hình ảnh tương đồng trong cluster6

So với bộ dữ liệu xây dựng trên mạng thì lần này với bộ dữ liệu thực tế độ chính xác thấp hơn do gặp phải trường hợp nhiễu ảnh dẫn tới độ chính xác không cao. Do số lượng ảnh ít nên thuật toán chạy chỉ mất 25 giây.

Chương trình chạy tiêu tốn bộ nhớ khá lớn ( 840MB ) và việc phải trích xuất đặc trưng dựa trên số lượng ảnh nên chương trình chạy ổn trên bộ dữ liệu nhỏ còn đối với bộ dữ liệu lớn thì chương trình tỏ ra khá chậm. Thuật toán K-means lặp lại việc tính khoảng cách để xác định cụm cho các điểm lặp đi lặp lại cũng là một trong nguyên nhân khiến chương trình chạy chậm với những bộ dữ liệu lớn.

# KẾT LUẬN

Bài toán xây dựng chương trình phân loại ảnh trong một thư mục ảnh dựa trên thời gian và độ tương đồng thành các thư mục con nhằm mục đích hỗ trợ người dùng tự động phân loại ảnh trong thư mục ảnh thành các thư mục con giúp tiết kiệm công sức, thời gian thay vì phải phân loại ảnh thủ công bằng mắt thường.

Các bước giải quyết bài toán bao gồm: thực hiện trích xuất đặc trưng của từng ảnh trong thư mục đầu vào sử dụng mạng nơron tích chập (CNN) để trích xuất vector đặc trưng, áp dụng thuật toán K-means để phân cụm ảnh thành các nhóm thư mục con có đặc điểm tương tự nhau và cuối cùng sắp xếp hình ảnh vào thư mục con tương ứng.

Kết quả thử nghiệm cho thấy hệ thống cơ bản đã thực hiện được yêu cầu của bài toán với độ chính xác khá tốt, tuy nhiên hệ thống vẫn còn nhiều hạn chế như thời gian chạy phụ thuộc vào số lượng ảnh cho nên hệ thống khá chậm với bộ dữ liệu lớn, giao diện chương trình còn thô sơ.

Trong tương lai, hệ thống sẽ được thử nghiệm với mô hình CNN hiện đại hơn, sử dụng thử các thuật toán phân loại khác để nâng cao khả năng phân loại của hệ thống, cải thiện tốc độ xử lý và tích hợp thử vào ứng dụng. Đồng thời, hệ thống sẽ sử dụng các bộ dữ liệu lớn để đánh giá hệ thống chính xác hơn.

Thông qua bài tập lớn, em xin chân thành cảm ơn thầy Nguyễn Hữu Tuân đã giúp đỡ em trong quá trình hiểu bài toán, gợi ý một số phương pháp giúp em có điểu kiện tìm hiểu về bài toán phân loại, mạng CNN, các thuật toán được dùng để phân loại ảnh và tiếp xúc với các thư viện hỗ trợ xử lý ảnh như OpenCV hay thư viện liên quan về machine learning, deep learning như Keras, Sklearn để từ đó em có thể ứng dụng được và áp dụng chúng với những bài toán sau này.

# TÀI LIỆU THAM KHẢO

[1]. GeeksforGeeks, Clustering in Machine Learning.

<https://www.geeksforgeeks.org/clustering-in-machine-learning/>

[2]. Nguyễn Đình Quý, Blog về bài toán phân cụm và thuật toán K-Means.

<https://ndquy.github.io/posts/thuat-toan-phan-cum-kmeans/>

[3]. Quoc Pham, Tìm Hiểu Convolutional Neural Networks Cho Phân Loại Ảnh.

<https://pbcquoc.github.io/cnn/>

[4]. Nguyễn Thanh Tuấn, Sách Deep Learning cơ bản.

[5]. Geeksforgeeks, VGG16 – CNN model

<https://www.geeksforgeeks.org/vgg-16-cnn-model/>

[6]. Mì AI, Video hướng dẫn cài đặt mạng VGG16.

<https://www.youtube.com/watch?v=C42lHmnNFe8&t=1052s>

[7]. Các phương pháp đánh giá thuật toán Clustering.

<https://bigdatauni.com/tin-tuc/cac-phuong-phap-danh-gia-trong-thuat-toan-clustering.html>

[8]. Dang Thi Thu Hien, Cluster Analysis.