

**Trường Đại Học Điện Lực**  
**Khoa Công Nghệ Thông Tin**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**  
**TRA CỨU ẢNH DỰA VÀO NỘI DUNG**

**ĐỀ TÀI:**

**TRA CỨU ẢNH THỰC PHẨM**

<i>Sinh viên thực hiện:</i>	<b>NGUYỄN MẠNH HÙNG</b>
<i>Lớp:</i>	<b>D15TTNT&amp;TGMT</b>
<i>Giảng Viên Hướng Dẫn:</i>	<b>HOÀNG VĂN QUÝ</b>
<i>Ngành:</i>	<b>CÔNG NGHỆ THÔNG TIN</b>
<i>Chuyên Ngành:</i>	<b>TTNT &amp; TGMT</b>
<i>Khóa:</i>	<b>2020-2025</b>

*Hà Nội, Tháng 5 Năm 2023*

## PHIẾU CHẤM ĐIỂM

**Sinh viên thực hiện**

Họ và tên	Điểm	Chữ ký	Ghi Chú
Nguyễn Mạnh Hưng 20810310347			

**Giảng viên chấm**

Họ và tên	Chữ ký	Ghi chú
Giảng viên chấm 1		
Giảng viên chấm 2		

# MỤC LỤC

LỜI NÓI ĐẦU .....	1
CHƯƠNG 1: TỔNG QUAN VỀ TRA CỨU ẢNH DỰA VÀO NỘI DUNG.....	2
1.1. Giới thiệu .....	2
1.2. Các thành phần chính của hệ thống.....	3
1.2.1. Tính toán độ tương đồng giữa các ảnh .....	3
1.2.2. Giao diện truy vấn .....	6
1.3. Một số phương pháp trích chọn đặc trưng .....	6
1.3.1. Đặc trưng màu sắc .....	7
1.3.2. Đặc trưng kết cấu.....	11
1.3.3. Đặc trưng hình dạng .....	11
CHƯƠNG 2: MÔ HÌNH CNN TRONG BÀI TOÁN TRA CỨU ẢNH .....	13
2.1. Tổng quan về mô hình CNN – Convolutional Neural Network .....	13
2.1.1. Convolutional là gì?.....	13
2.1.2. Kiến trúc mạng CNN.....	14
2.2. Tổng quan về mạng VGG .....	16
2.3. Ứng dụng mô hình VGG19 để trích chọn đặc trưng ảnh .....	18
2.3.1. Độ đo khoảng cách sử dụng trong chương trình .....	19
2.3.2. Số chiều của vector đặc trưng trong mô hình VGG19 .....	19
CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ .....	20
3.1. Cơ sở dữ liệu .....	20
3.1.1. Cơ sở dữ liệu ảnh .....	20
3.1.2. Trích chọn đặc trưng .....	20
3.2. Chương trình thực nghiệm .....	21
3.2.1. Giao diện.....	21
3.2.2. Các bước thực hiện truy vấn ảnh .....	22
3.3. Đánh giá kết quả.....	23
KẾT LUẬN .....	24
TÀI LIỆU THAM KHẢO.....	25

## LỜI NÓI ĐẦU

Để tìm kiếm một vài hình ảnh trong một kho dữ liệu ảnh chứa tới vài trăm nghìn hình ảnh không phải là chuyện đơn giản. Nếu chúng ta sử dụng cách tìm kiếm thủ công như xem lần lượt từng bức ảnh cho tới khi tìm được bức ảnh có nội dung cần tìm sẽ khiến ta mất rất nhiều thời gian và kết quả có thể không chính xác. Một phương pháp truyền thống khác được sử dụng là tra cứu ảnh dựa vào các từ khóa được chú thích. Để chú thích các hình ảnh tốn rất nhiều thời gian bởi vì rất khó để có thể mô tả nội dung của ảnh chỉ bằng các từ khóa.

Ngày nay, với sự phát triển của các phương tiện kỹ thuật số, số lượng ảnh sẽ còn tăng lên liên tục và nhanh chóng. Vì vậy, rất cần có các phương pháp hỗ trợ tra cứu ảnh từ một cơ sở dữ liệu ảnh khổng lồ một cách nhanh chóng và thuận tiện nhất. Chính vì vậy, em chọn đề tài “Tra cứu ảnh thực phẩm” nhằm tìm hiểu rõ hơn về hệ thống CBIR, cũng như những lợi ích mà hệ thống này đem lại. Trong tra cứu ảnh dựa vào nội dung (CBIR), chú thích ảnh không được yêu cầu, bằng việc sử dụng phương pháp này ta có thể tra cứu các hình ảnh mong muốn từ cơ sở dữ liệu ảnh lớn dựa vào các đặc trưng được trích xuất từ các ảnh.

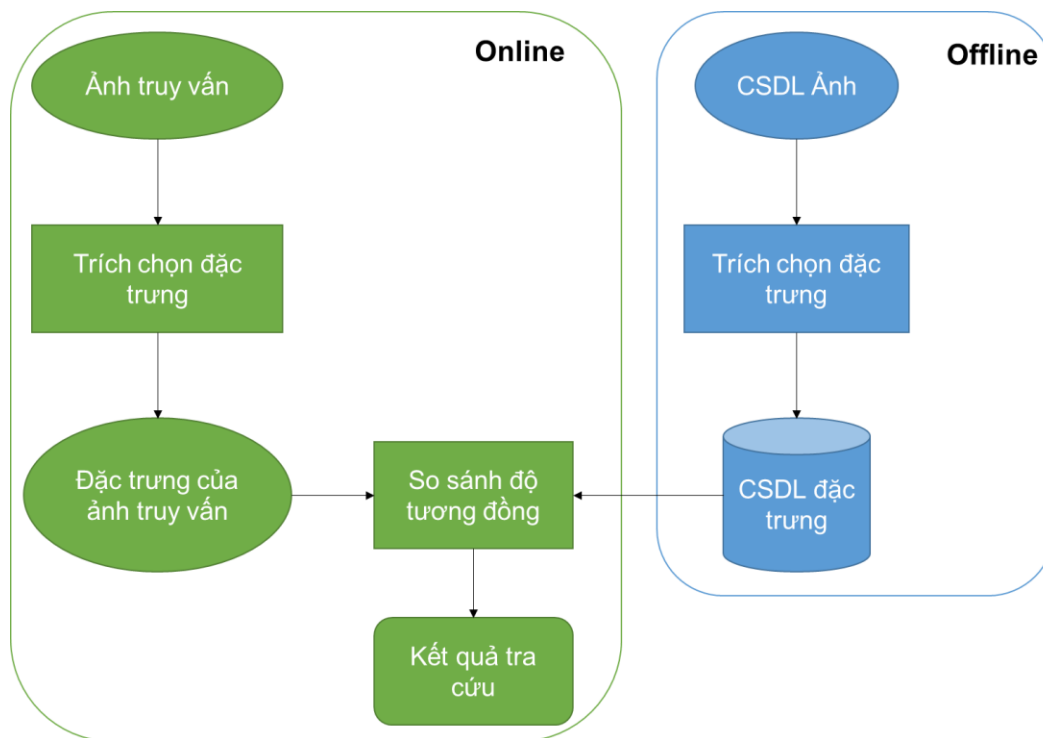
# CHƯƠNG 1: TỔNG QUAN VỀ TRA CỨU ẢNH DỰA VÀO NỘI DUNG

## 1.1. Giới thiệu

Tra cứu ảnh dựa vào nội dung được thực hiện theo nguyên tắc sau: Sử dụng các vector đặc trưng được trích xuất từ ảnh như màu sắc, kết cấu, hình dạng,... để biểu diễn nội dung của mỗi ảnh trong cơ sở dữ liệu ảnh và ảnh cần tìm. Sau đó, tính toán độ tương đồng giữa ảnh cần tìm và ảnh trong cơ sở dữ liệu, kết quả trả về là hình ảnh có độ tương đồng cao nhất.

Một hệ thống tra cứu ảnh dựa vào nội dung (CBIR – Content Based Image Retrieval) tiêu biểu không chỉ xử lý các nguồn thông tin khác nhau ở các định dạng khác nhau (văn bản, hình ảnh và video) mà còn giải quyết các nhu cầu của người sử dụng. Về cơ bản, hệ thống phân tích các nội dung của nguồn thông tin cũng như các nội dung của truy vấn sử dụng, sau đó so sánh các nội dung này để tra cứu và đưa ra các nội dung liên quan. Các chức năng chính của một hệ thống tra cứu ảnh dựa vào nội dung như sau:

- Phân tích các nội dung của nguồn thông tin, và biểu diễn các nội dung của các nguồn được phân tích theo cách thích hợp cho so sánh các truy vấn sử dụng (không gian của nguồn thông tin được biến đổi thành không gian đặc trưng cho mục tiêu so sánh nhanh trong bước tiếp theo). Bước này thông thường cần rất nhiều thời gian do nó phải xử lý lần lượt tất cả thông tin nguồn (các ảnh) trong cơ sở dữ liệu. Tuy nhiên, bước này được thực hiện chỉ một lần và có thể được thực hiện ngoại tuyến.
- Phân tích các truy vấn người sử dụng và biểu diễn chúng ở dạng thích hợp cho so sánh với cơ sở dữ liệu nguồn. Bước này là tương tự với bước trước, nhưng chỉ được áp dụng với ảnh truy vấn.
- Nghĩa một chiến lược để so sánh các truy vấn tìm kiếm với thông tin trong cơ sở dữ liệu được lưu trữ. Tra cứu thông tin liên quan theo một cách hiệu quả. Bước này được thực hiện trực tuyến và được yêu cầu rất nhanh. Các kỹ thuật đánh chỉ số hiện đại có thể được sử dụng để tổ chức lại không gian đặc trưng để tăng tốc quá trình đối sánh.
- Thực hiện các điều chỉnh cần thiết trong hệ thống (thường bằng điều chỉnh các tham số trong máy đối sánh) dựa trên phản hồi từ người sử dụng và/hoặc các ảnh được tra cứu.



*Hình 1.1. Kiến trúc của hệ thống tra cứu ảnh*

Nhìn vào mô hình trên ta nhận thấy rằng trên một mặt của một hệ thống tra cứu ảnh dựa vào nội dung có các nguồn thông tin trực quan ở các khuôn dạng khác nhau, trên mặt kia có các truy vấn người sử dụng. Hai mặt này đều được liên kết thông qua một chuỗi các tác vụ. Hai tác vụ phân tích truy vấn người sử dụng và đánh chỉ số nhiều chiều được tóm lược mục 1.1.2 và 1.1.3 trong khi hai tác vụ quan trọng nhất: “Phân tích các nội dung của thông tin nguồn” (trích rút đặc trưng) và “Định nghĩa một chiến lược để đối sánh các truy vấn tìm kiếm với thông tin trong cơ sở dữ liệu được lưu trữ” (các độ đo tương tự), sẽ được mô tả chi tiết hơn trong phần chuyên dụng (mục 1.2 và mục 1.3) trong đó màu được nhấn mạnh.

## **1.2. Các thành phần chính của hệ thống**

### **1.2.1. Tính toán độ tương đồng giữa các ảnh**

#### **\*Độ đo màu sắc:**

Một số độ đo tương đồng được sử dụng như: Độ đo khoảng cách Oclit, độ đo Jensen-Shannon divergence (JSD).

Gọi  $h(I)$  và  $h(M)$  tương ứng là 2 lược đồ màu của hai ảnh  $I$  và ảnh  $M$ . Khi đó các loại độ đo màu được định nghĩa là một số nguyên (hoặc số thực) theo các loại độ đo tương ứng như sau:

- **Khoảng cách Öclit**

Đây là khoảng cách Öclit thông thường giữa các  $K$  bin:

$$Intersection(h(I), h(M)) = \sum_{j=1}^k \sqrt{(h(I) - h(M))^2} \sum_{j=1}^k \sqrt{(h(I) - h(M))^2}$$

Hoặc

$$Intersection(h(I), h(M)) = \sum_{j=1}^k \sqrt{|h(I) - h(M)|}$$

- **Độ đo Jensen-Shannon divergence (JSD)**

Độ đo Jensen-Shannon divergence sử dụng lược đồ màu RGB để tính toán độ tương đồng về màu sắc giữa 2 ảnh:

$$d_{JSD}(H, H') = \sum_{m=1}^M H_m \log \frac{2H_m}{H_m + H'_m} + H'_m \log \frac{2H'_m}{H'_m + H_m}$$

$$\sum_{m=1}^M H_m \log \frac{2H_m}{H_m + H'_m} + H'_m \log \frac{2H'_m}{H'_m + H_m}$$

Trong đó:  $H$  và  $H'$  là 2 biểu đồ màu được so sánh,  $H_m$  là dải thứ  $m$  của biểu đồ  $H$ .

**\*Độ đo tương đồng hình dạng:**

Độ đo về hình dạng rất nhiều trong phạm vi lý thuyết của bộ môn xử lý ảnh. Chúng trải rộng từ những độ đo toàn cục dạng thô với sự trợ giúp của việc nhận dạng đối tượng, cho tới những độ đo chi tiết tự động tìm kiếm những hình dạng đặc biệt. Lược đồ hình dạng là một ví dụ của độ đo đơn giản. Kỹ thuật dùng đường biên hiệu quả hơn phương pháp trước, chúng tìm kiếm những hình dạng đối tượng gần giống với đường biên nhất. Phương pháp vẽ phác họa là phương pháp có nhiều đặc trưng rõ ràng hơn, không chỉ tìm kiếm những đường biên đối tượng đơn, mà còn đối với tập những đối tượng đã được phân đoạn trong một ảnh mà người dùng vẽ hay cung cấp.

### **\*Độ đo tương đồng kết cấu ảnh:**

Để đo độ tương đồng theo kết cấu giữa các ảnh, người ta thường sử dụng độ đo Oclit. Kết cấu được trích xuất từ các bức ảnh sẽ được biểu diễn thành các véc tơ nhiều chiều và khoảng cách Oclit được dùng để đo độ tương đồng giữa các đặc trưng của ảnh truy vấn với đặc trưng của ảnh trong cơ sở dữ liệu.

### **\*Độ đo tương đồng đặc trưng phân đoạn:**

Một số độ đo tương đồng:

- **Độ đo Cosin distance (Cosine similarity)** là độ đo khoảng cách được sử dụng phổ biến trong CBIR. Nó tính toán góc giữa hai vector và đo lường độ tương đồng giữa chúng. Độ đo này không bị ảnh hưởng bởi độ lớn của vector và cho phép tìm các ảnh giống nhau dựa trên hướng của vector đặc trưng. Cosine similarity thường được sử dụng để giải quyết vấn đề của Euclidean distance ở không gian nhiều chiều. Ý tưởng đơn giản là tính góc tạo thành giữa hai vector. Giá trị sẽ tương đương với phép dot product nếu cả hai vector được norm về giá trị 1. Hai vector cùng hướng sẽ có cosine similarity bằng 1 và ngược hướng sẽ có giá trị -1. Lưu ý rằng, chiều dài không được sử dụng do đây là phương pháp tính theo hướng.

$$d(x,y) = \frac{x.y}{\|x\|.\|y\|}$$

Nhược điểm của độ đo Cosine distance là không tận dụng độ lớn của vector, chỉ tính theo hướng. Điều này vô tình làm mất mát thông tin so sánh. Use case thường được sử dụng trong các dữ liệu đa chiều và không quá phụ thuộc vào độ lớn của vector. Các độ đo khoảng cách này được sử dụng để tính toán khoảng cách giữa vector đặc trưng của ảnh ban đầu và các vector đặc trưng của các ảnh trong tập dữ liệu. Các ảnh có khoảng cách gần với ảnh ban đầu sẽ được xem là các ảnh giống nhau.

- **Khoảng cách góc**

$$d(x,y) = \cos^{-1}(x.y)$$

- **Độ đo Euclid distance** còn được biết đến với cái tên L2 distance, là độ đo khoảng cách cơ bản nhất được sử dụng trong CBIR. Nó tính khoảng cách giữa hai vector bằng cách tính căn bậc hai của tổng bình phương các hiệu của các thành phần tương ứng trong hai vector. Khi tính toán khoảng cách giữa hai vector đặc trưng của hai ảnh, Euclidean distance đo lường khoảng cách giữa chúng bằng cách tính căn bậc hai của tổng bình phương của các hiệu giữa các



thành phần tương ứng của hai vector đó. Giá trị khoảng cách được tính toán sẽ cho biết mức độ khác biệt giữa các vector đặc trưng này, giúp xác định sự tương đồng giữa các ảnh.

$$d(x,y) = \sqrt{\sum_{i=1}^m |x_i - y_i|^2}$$

Cách tính toán không có gì phức tạp, tính căn bậc hai tổng bình phương hiệu của các điểm. Ưu điểm của độ đo là phổ biến, dễ hiểu, dễ implement, kết quả tốt trong nhiều usecase. Đặc biệt hiệu quả với dữ liệu ít chiều. Nhược điểm là Euclidean distance có thể bị ảnh hưởng bởi đơn vị của feature. Chính vì vậy cần phải normalize trước khi tính toán. Khi số chiều vector space tăng lên, Euclidean Distance trở nên kém hiệu quả. Một phần nguyên nhân do dữ liệu thực tế thường không chỉ nằm trong Euclidean Metric Space.

### 1.2.2. Giao diện truy vấn

Để biểu diễn ảnh tra cứu từ CSDL cho người dùng thì có rất nhiều cách. Trong đó, những cách thông thường nhất được sử dụng là: Duyệt qua mục; Truy vấn bởi khái niệm; Truy vấn với bản phác thảo và truy vấn với ví dụ...

- Duyệt qua mục là phương pháp duyệt qua toàn bộ CSDL theo danh mục các ảnh. Mục đích của phương pháp này là ảnh trong CSDL được phân loại thành nhiều mục khác nhau theo ngữ nghĩa hoặc nội dung trực quan.
- Truy vấn bởi khái niệm là tra cứu ảnh theo mô tả khái niệm liên quan với từng ảnh trong CSDL.
- Truy vấn với bản phác thảo và truy vấn với ví dụ là vẽ ra một bản phác thảo hoặc cung cấp một hình ảnh ví dụ từ những ảnh với độ tương đồng đặc trưng trực quan sẽ được trích chọn từ CSDL.

Trong số các phương pháp trên, phương pháp truy vấn với bản phác thảo và truy vấn với ví dụ là phương pháp quan trọng và khó khăn nhất. Phần lớn các nghiên cứu tra cứu ảnh dựa trên nội dung tập trung đi sâu vào phương pháp này.

### 1.3. Một số phương pháp trích chọn đặc trưng

Trích chọn đặc trưng là cơ sở của tra cứu ảnh dựa vào nội dung. Các đặc trưng của hình ảnh bao gồm: các đặc trưng dựa vào văn bản và các đặc trưng trực quan như màu, kết cấu, hình dạng..., được định lượng trong tự nhiên, chúng có thể được trích xuất tự động hoặc bán tự động. Trong phạm vi đặc trưng trực quan, các đặc

trung có thể được phân loại tiếp thành các đặc trưng chung hoặc các đặc trưng lĩnh vực cụ thể.

### **1.3.1. Đặc trưng màu sắc**

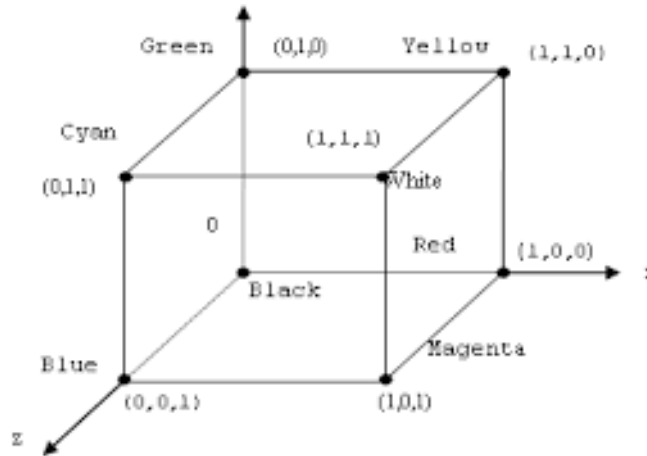
Màu sắc là đặc trưng trực quan quan trọng đầu tiên, đơn giản nhất cho đánh chỉ số và tra cứu các ảnh. Nó cũng là đặc trưng được sử dụng phổ biến nhất trong lĩnh vực tra cứu ảnh. Các đặc trưng màu sắc có thể cung cấp những thông tin rất hữu hiệu cho việc phân loại ảnh và chúng cũng rất hữu ích cho việc tra cứu ảnh. Chính vì thế mà tra cứu ảnh dựa trên màu sắc được sử dụng rộng rãi trong các hệ thống tra cứu ảnh dựa trên nội dung. Lược đồ màu cung cấp, biểu diễn hiệu quả nội dung màu của một ảnh nếu mẫu màu là duy nhất so với phần còn lại của tập dữ liệu. Lược đồ màu dễ dàng để tính toán và hiệu quả trong mô tả cả phân bố màu toàn cục và cục bộ trong ảnh. Hơn nữa, nó mạnh với quay và dịch chuyển về trực quan sát và thay đổi chậm với tỷ lệ và góc quan sát. Vì vậy trước khi sử dụng lược đồ màu chúng ta cần phải lựa chọn và xác định kiểu không.

#### **\*Không gian màu**

Không gian màu là sự biểu diễn tập các màu, một số không gian màu được sử dụng rộng rãi trong đồ họa máy tính và màu sắc thường được xác định trong không gian màu 3 chiều.

- **Không gian màu RGB**

Không gian màu RGB mô tả màu sắc bằng 3 thành phần chính là Red, Green và Blue. Không gian này được xem như một khối lập phương 3 chiều và màu red là trục x, màu Green là trục y, màu Blue là trục z. Mỗi màu trong không gian này được xác định bởi 3 thành phần R, G, B. tương ứng với các tổ hợp khác nhau của 3 màu này sẽ cho ta một màu mới.



Hình 1.2. Không gian màu RGB

Ở hình lập phương trên, mỗi màu gốc (R, G, B) có các góc đối diện là các màu bù với nó. Hai màu được gọi là bù nhau khi kết hợp hai màu này lại với nhau ra màu trắng. Ví dụ: Green - Magenta, Red - Cyan, Blue - Yellow.

- **Không gian màu CMY (Cyan - Magenta - Yellow)**

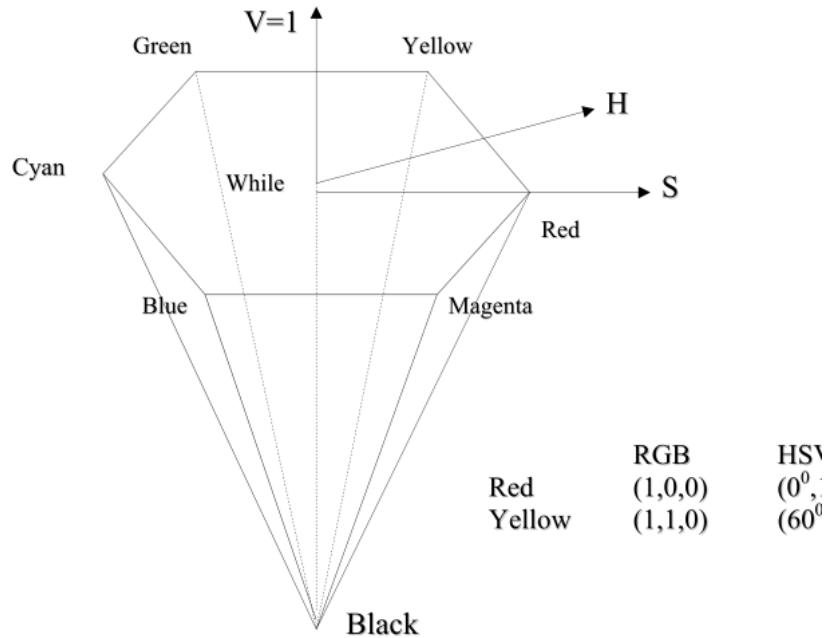
Tương tự như không gian màu RGB nhưng 3 thành phần chính là Cyan, Magenta và Yellow. Do đó, tọa độ các màu trong không gian CMY trái ngược với không gian RGB. Ví dụ: màu White có các thành phần là (0,0,0), màu Black (1,1,1), màu Cyan (1,0,0), ....

- **Không gian màu HSV (Hue - Saturation - Value)**

Thực chất của không gian này là sự biến đổi của không gian RGB. Không gian HSV được mô tả bằng lệnh lập phương RGB quay trên đỉnh Black. H (Hue) là góc quay trục V (value) qua 2 đỉnh Black và White.

Các giá trị biến thiên của H, S, V được mô tả như sau:

- H (Hue) chỉ sắc thái có giá trị từ 00 - 3600.
- S (Saturation) chỉ độ bão hòa.
- V (Value) có giá trị từ 0 - 1. Các màu đạt giá trị bão hòa khi  $s=1$  và  $v=1$ .



*Hình 1.3. Không gian màu HSV*

Sắc độ là thành phần của không gian màu HSx. Sắc độ là góc giữa những đường tham chiếu và điểm gốc màu trong không gian màu RGB. Vùng giá trị này từ 00 đến 3600. Theo uỷ ban quốc tế về màu sắc CIE (Commission International d'Eclairage) thì sắc độ là thuộc tính của cảm giác có liên quan đến thị giác, qua đó một vùng xuất hiện tương tự với một màu được cảm nhận như red, green, blue hoặc là sự kết hợp của hai trong số những màu được cảm nhận. Cũng theo CIE độ bão hoà là màu được đánh giá theo tỷ lệ độ sáng của nó. Trong hình nón độ bão hoà là khoảng cách từ tâm đến cạnh hình nón. Chiều cao của đường cắt chính là giá trị (value) đây chính là độ sáng hoặc độ chói của màu. Khi độ bão hoà S bằng 0 thì H không xác định, giá trị nằm trên trục V biểu diễn ảnh xám. Không gian màu HSV dễ dàng lượng tử hoá. Mức lượng tử hoá thông dụng trong không gian màu này là 162 mức với H nhận 18 mức, S và V nhận 3 mức.

### **\*Lược đồ màu**

Lược đồ màu cung cấp một biểu diễn hiệu quả của nội dung màu của một ảnh nếu mẫu màu là duy nhất so với phần còn lại của tập dữ liệu. Lược đồ màu dễ dàng để tính toán và hiệu quả trong mô tả cả phân bố màu toàn cục và cục bộ trong ảnh.

Nó mạnh với quay và dịch chuyển về trực quan sát và thay đổi chậm với tỷ lệ và góc quan sát.

Lược đồ màu được xác định bằng một tập các bin, trong đó mỗi bin biểu thị xác suất của các pixel trong ảnh. Một lược đồ màu  $H$  của một ảnh đã cho được xác định bởi véc tơ:

$$H=\{H[0], H[1], H[2], ..., H[i],... H[N]\},$$

Ở đây  $i$  biểu diễn một màu trong lược đồ màu và tương ứng với một khối con trong không gian màu RGB,  $H[i]$  là số các pixel có màu  $i$  trong ảnh và  $N$  là số các bin trong lược đồ màu.

Mỗi pixel trong ảnh sẽ thuộc về một bin của lược đồ màu của ảnh, vì thế với lược đồ màu của một ảnh, giá trị của mỗi bin là số các pixel cùng màu. Để so sánh các ảnh có các kích cỡ khác nhau, các lược đồ màu được chuẩn hóa. Lược đồ màu chuẩn hóa  $H'$  được xác định bằng:

$$H'=\{H'[0], H'[1], H'[2], ..., H'[i],... H'[N]\},$$

Ở đây,  $P$  là tổng số các pixel trong ảnh.

Trong lược đồ lượng hóa không gian màu lý tưởng, các màu riêng biệt không được định vị trong cùng hình khối con và các màu tương tự được gán vào cùng hình khối con. Sử dụng một số màu sẽ giảm khả năng các màu tương tự được gán vào các bin khác nhau, nhưng cũng tăng khả năng các màu riêng biệt được gán vào cùng các bin, nội dung thông tin của các ảnh sẽ giảm. Mặt khác, các lược đồ màu với một số lượng lớn các bin sẽ chứa nhiều thông tin về nội dung của ảnh, theo đó giảm khả năng các màu riêng biệt sẽ được gán vào cùng các bin. Tuy nhiên, chúng tăng khả năng các màu tương tự sẽ được gán vào các bin khác nhau và tăng không gian lưu trữ biểu diễn ảnh và thời gian tính toán khoảng cách giữa các lược đồ màu. Do đó, cần có sự thỏa hiệp trong việc xác định số lượng các bin sẽ được sử dụng trong các lược đồ màu.

### 1.3.2. Đặc trưng kết cấu

Kết cấu là một đặc tính quan trọng khác của ảnh. Các biểu diễn kết cấu đa dạng đã được nghiên cứu trong nhận dạng mẫu và thị giác máy tính. Về cơ bản, các phương pháp biểu diễn kết cấu có thể được phân ra thành hai loại: cấu trúc và thống kê.

Kết cấu là một mô tả vùng trợ giúp tốt trong quá trình tra cứu. Kết cấu không có khả năng tìm ra các ảnh tương tự, nhưng nó có thể được sử dụng để phân lớp các ảnh kết cấu từ các ảnh không kết cấu và sau đó được kết hợp với các thuộc tính đặc trưng khác như màu để làm cho tra cứu hiệu quả hơn. Một trong những biểu diễn phổ biến nhất của đặc trưng kết cấu là ma trận đồng khả năng được đề xuất bởi Haralick và cộng sự. Ma trận dựa trên hướng và khoảng cách pixel. Các thống kê từ ma trận đồng khả năng được trích rút và được biểu diễn như thông tin kết cấu. Tamura và cộng sự đã đề xuất phương pháp trích rút sáu thuộc tính kết cấu đặc trưng coarseness, contrast, directionality, likeliness, regularity and roughness.

### 1.3.3. Đặc trưng hình dạng

So với các đặc trưng màu và kết cấu, các đặc trưng hình dạng thường được mô tả sau khi các ảnh được phân đoạn thành các vùng hoặc các đối tượng. Hình dạng được xem như là một đặc trưng quan trọng trong mô tả các đối tượng nổi bật trong ảnh và có thể giúp phân biệt giữa hai ảnh.

Đầu tiên là sử dụng một phương pháp phân đoạn thích hợp để chia ảnh thành các vùng. Các kỹ thuật phân đoạn có thể được phân lớp thành ba loại: dựa vào vùng, dựa vào đường biên và dựa vào pixel. Sau khi ảnh được phân đoạn và thu được các vùng, các đặc trưng thuộc về các vùng thu được sẽ được ghi lại. Các mã xích sử dụng 4 hoặc 8 liên thông để biểu diễn các đoạn thẳng tạo thành đường biên của vùng. Các dấu hiệu, số các hình và đa giác là các lược đồ biểu diễn khác.

Tiếp theo là sử dụng các mô tả thích hợp cho các vùng này sao cho chúng có thể được sử dụng trong khi đối sánh các vùng của các ảnh khác nhau. Các mô tả hình được chia thành ba loại: Các mô tả dựa vào đường biên xác định các thuộc tính của đường biên. Các kỹ thuật dựa vào đường biên sử dụng các phác thảo vùng để tính

toán hình. Mô tả Fourier là một trong những phương pháp phổ biến thuộc về loại này. Trong kỹ thuật này, đường biên của một vùng đã cho được thu và được biến đổi Fourier. Các hệ số Fourier trội được sử dụng như các mô tả hình. Các mô tả khác trong loại này là các mô men hình.

Nếu một vùng có một hình phức hợp, thì nó có thể được phân tách tiếp thành các hình đơn giản hơn như các hình chữ nhật hoặc các hình tròn, một số thuộc tính của các hình đơn giản này và các quan hệ của chúng có thể được sử dụng cho các mô tả hình. Các mô tả vùng khác gồm màu và kết cấu. Một vài đặc tính của vùng như trọng tâm, không cụ thể đối với đường biên hoặc nội dung của vùng.

## CHƯƠNG 2: MÔ HÌNH CNN TRONG BÀI TOÁN TRA CỨU ẢNH

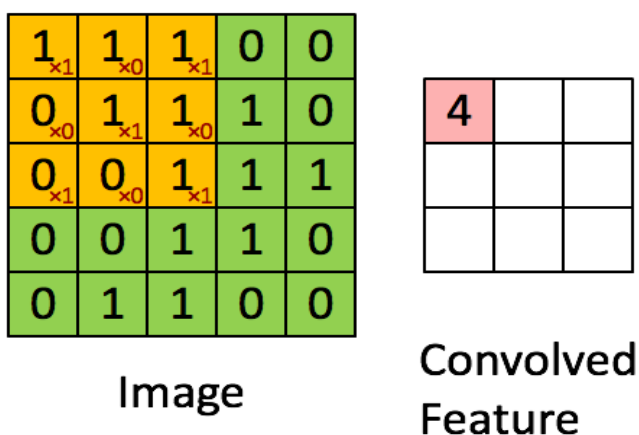
### 2.1. Tổng quan về mô hình CNN – Convolutional Neural Network

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

CNN cũng có lịch sử khá lâu đời. Kiến trúc gốc của mô hình CNN được giới thiệu bởi một nhà khoa học máy tính người Nhật vào năm 1980. Sau đó, năm 1998, Yan LeCun lần đầu huấn luyện mô hình CNN với thuật toán backpropagation cho bài toán nhận dạng chữ viết tay. Tuy nhiên, mãi đến năm 2012, khi một nhà khoa học máy tính người Ukraine Alex Krizhevsky (đệ của Geoffrey Hinton) xây dựng mô hình CNN (AlexNet) và sử dụng GPU để tăng tốc quá trình huấn luyện deep nets để đạt được top 1 trong cuộc thi Computer Vision thường niên ImageNet với độ lỗi phân lớp top 5 giảm hơn 10% so với những mô hình truyền thống trước đó, đã tạo nên làn sóng mãnh mẽ sử dụng deep CNN với sự hỗ trợ của GPU để giải quyết càng nhiều các vấn đề trong Computer Vision.

#### 2.1.1. Convolutional là gì?

Là một cửa sổ trượt (Sliding Windows) trên một ma trận.



Hình 2.1: Cửa sổ trượt trên ma trận



Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature.

Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước  $5 \times 5$  và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột.

Convolution hay tích chập là nhân từng phần tử trong ma trận  $3 \times 3$ . Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có kích thước nhỏ như trong ví dụ trên là  $3 \times 3$ .

Convolution hay tích chập là nhân từng phần tử bên trong ma trận  $3 \times 3$  với ma trận bên trái. Kết quả được một ma trận gọi là Convoled feature được sinh ra từ việc nhân ma trận Filter với ma trận ảnh  $5 \times 5$  bên trái.

### **2.1.2. Kiến trúc mạng CNN**

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

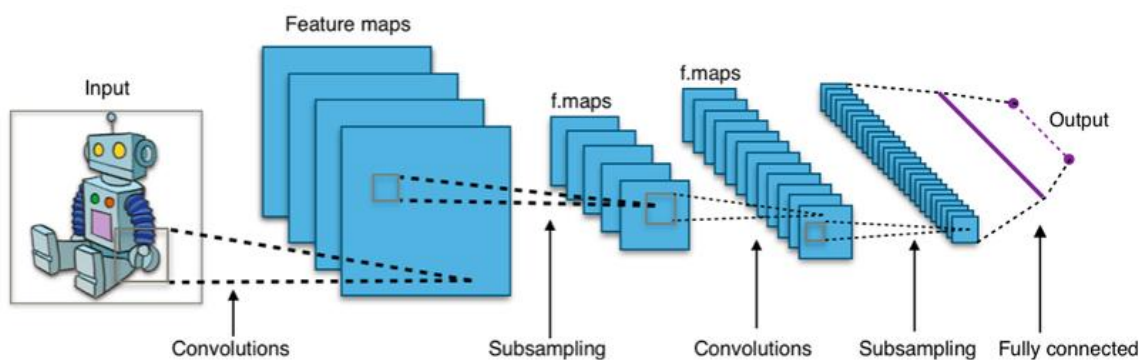
Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer

khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (training) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



Hình 2.2: Các lớp trong Convolutional Layer

Trong mô hình CNN có 2 khía cạnh cần quan tâm là **tính bất biến** (Location Invariance) và **tính kết hợp** (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter. Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

Mạng CNN sử dụng 3 ý tưởng cơ bản:

- **Các trường tiếp nhận cục bộ** (local receptive field).
- **Trọng số chia sẻ** (shared weights).
- **Tổng hợp** (pooling).

## 2.2. Tổng quan về mạng VGG

Sau những thành công của AlexNet và sau đó là ZFNet, giới nghiên cứu bị thuyết phục bởi những kết quả ấn tượng từ CNN. Tuy nhiên các mô hình nhìn chung được chỉnh sửa mang nhiều công đoạn code phức tạp, chưa có một mẫu chung để lựa chọn các tham số cho các lớp như kernel size, stride, padding, ...

VGG được xây dựng và đào tạo bởi Karen Simonyan và Andrew Zisserman tại Đại học Oxford vào năm 2014. VGG ra đời đã tạo nên một kiểu mẫu tạo ra mô hình mà chúng ta đều có thể áp dụng được dễ dàng và tạo ra các mô hình large scale với hàng triệu tham số. VGG được đào tạo bằng cách sử dụng hơn 1 triệu hình ảnh từ cơ sở dữ liệu ImageNet, trong đó có các ảnh màu  $224 \times 224$  pixel.

### **\*Cách chọn kernel size, filter, stride, padding cho Convolution**

Thay vì sử dụng một kernel size lớn, VGG chỉ sử dụng các kernel size  $3 \times 3$  cho tất cả lớp Convolution, với stride 1 và padding 1. Khi đưa qua các lớp convolution này, kích thước ảnh  $W \times H$  không thay đổi.

Ý tưởng cơ bản của tác giả VGG đưa ra đó là việc sử dụng 3 lớp convolution  $3 \times 3$  hiệu quả hơn so với sử dụng 1 lớp convolution  $7 \times 7$  của ZFNet.

Đầu tiên về mặt đặc trưng, 3 lớp  $3 \times 3$  với các channel lớn sẽ tối đa hóa sự đa dạng các đặc trưng học được hơn vì ta sử dụng tới 3 lần biến đổi phi tuyến. Thứ hai, về số lượng tham số kernel, 1 lớp convolution  $7 \times 7$  có số tham số  $1 \times (7 \times 7) \times C = 49C$ , với  $C$  là số lượng filter của convolution, đối với 3 lớp convolution  $3 \times 3$  ta chỉ có  $3 \times (3 \times 3) \times C = 27C$ , ít hơn hẳn nhưng vẫn giữ nguyên vùng nhận thức.

### **\*Max Pooling**

VGG sử dụng các lớp Convolution để học đặc trưng, và các lớp Max Pooling để downsampling, các lớp Max Pooling có kernel size 2, stride 2. Khi đưa qua lớp Max Pooling này, kích thước của ảnh sẽ giảm đi  $1/2$ .

### **\*Scale up các đặc trưng**

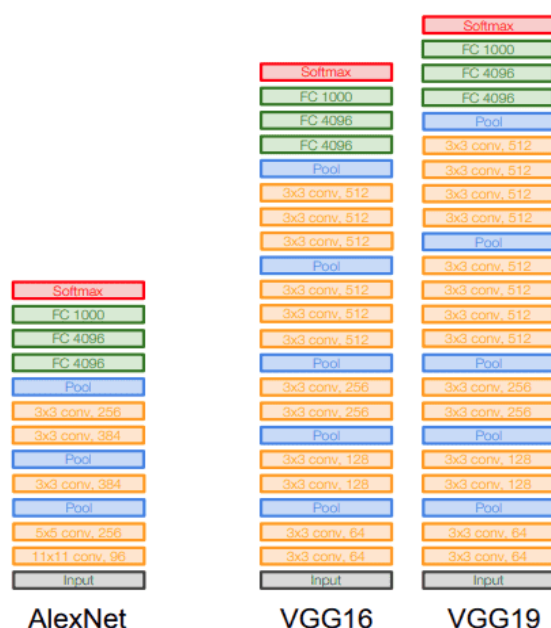
Đây là mô hình đầu tiên sử dụng các lớp Convolution liên tiếp nhau trước khi sử dụng Max Pooling, VGG dùng số lượng filter của các lớp Convolution lũy thừa của 2 như 64, 128, 256, 512. Mô hình VGG nhìn chung là một mô hình có số lượng tham số cực lớn với hơn 128 triệu tham số, không gian lưu trữ của mô hình ước tính khoảng trên 600Mb.

## \*Kiến trúc tổng quan

Về cơ bản, mẫu thiết kế của VGG khá đơn giản vì chúng lặp lại nhau, khi đi vào các lớp sâu hơn, kích thước channel sẽ tăng theo lũy thừa 2, và kích thước ảnh sẽ được giảm đi một nửa dần dần.

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 × 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 2.3: Các cấu hình của mạng VGG



Hình 2.4: Mô hình so sánh giữa mạng VGG16 và VGG19 với AlexNet

### 2.3. Ứng dụng mô hình VGG19 để trích chọn đặc trưng ảnh

Mô hình VGG19 có kiến trúc mạng sâu hơn VGG16 với 16 lớp tích chập, 3 lớp fully connected, 5 lớp max pooling và 1 lớp Softmax để phân loại. Kiến trúc mạng này với có khoảng 143 triệu tham số, trong đó các tham số được học từ bộ dữ liệu ImageNet chứa 1,2 triệu đối tượng chung của 1.000 loại đối tượng khác nhau để đào tạo.

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg19 import VGG19, preprocess_input
from tensorflow.keras.models import Model
from pathlib import Path
from PIL import Image
import numpy as np

class FeatureExtractor:
    def __init__(self):
        base_model = VGG19(weights='imagenet')
        self.model = Model(inputs=base_model.input, outputs=base_model.get_layer('fc1').output)

    def extract(self, img):
        # Resize the image
        img = img.resize((224, 224))
        # Convert the image color space
        img = img.convert('RGB')
        # Reformat the image
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        x = preprocess_input(x)
        # Extract Features
        feature = self.model.predict(x)[0]
        return feature / np.linalg.norm(feature)
```

Hình 2.5: Code khai báo sử dụng mạng VGG19

Mô hình VGG19 từ thư viện *tensorflow.keras* là thứ đầu tiên chúng ta cần import. Tiếp theo, module *preprocess\_input* được sử dụng để chia tỷ lệ các giá trị pixel phù hợp với mô hình VGG19, module *Image* được import để xử lý các đối tượng hình ảnh.

Để xử lý mảng, module *numpy* được import. Các trọng số tiền xử lý của bộ dữ liệu *ImageNet* sau đó được tải vào mô hình VGG19. Mô hình VGG19 được tạo thành từ các lớp tích chập, theo sau là một vài lớp dày đặc được kết nối đầy đủ. Chúng ta có thể lựa chọn sử dụng lớp dày đặc cuối cùng với *Include\_top* hay không.

Thành phần trích xuất đặc trưng của mô hình được mở rộng từ lớp đầu vào cho đến lớp tổng hợp tối đa cuối cùng, được biểu thị bằng số  $7 \times 7 \times 512$ , thành phần phân loại của mô hình mở rộng khắp mạng. Sau khi xác định mô hình, hình ảnh đầu vào phải đúng với kích thước mà mô hình yêu cầu, trong trường hợp này là  $224 \times 224$  pixel.

Các lớp tích chập sẽ trích xuất các đặc trưng mức thấp của ảnh, như cạnh, góc, hay chấm tròn, lớp fully-connected sẽ giúp tổng hợp các đặc trưng này để tạo ra một vector đặc trưng duy nhất cho ảnh.

### 2.3.1. Độ đo khoảng cách sử dụng trong chương trình

Trong chương trình này, em sử dụng độ đo khoảng cách Euclid distance để tính khoảng cách giữa các ảnh. Độ đo này còn được biết đến với cái tên L2 distance, là độ đo khoảng cách cơ bản nhất được sử dụng trong CBIR. Nó tính khoảng cách giữa hai vector bằng cách tính căn bậc hai của tổng bình phương các hiệu của các thành phần tương ứng trong hai vector. Khi tính toán khoảng cách giữa hai vector đặc trưng của hai ảnh, Euclidean distance đo lường khoảng cách giữa chúng bằng cách tính căn bậc hai của tổng bình phương của các hiệu giữa các thành phần tương ứng của hai vector đó. Giá trị khoảng cách được tính toán sẽ cho biết mức độ khác biệt giữa các vector đặc trưng này, giúp xác định sự tương đồng giữa các ảnh.

$$d(x, y) = \sqrt{\sum_{i=1}^m |x_i - y_i|^2}$$

### 2.3.2. Số chiều của vector đặc trưng trong mô hình VGG19

Để trích xuất đặc trưng của ảnh, ta chỉ quan tâm đến phần mạng tích chập và lớp fully connected thứ nhất (fc1) của mô hình. Về cơ bản, lớp fc1 này nhận đầu vào từ lớp flatten (làm phẳng) và tạo ra một vector đặc trưng có kích thước 4096. Điều này có nghĩa là với mỗi bức ảnh đầu vào, đầu ra của lớp fc1 sẽ là một vector 4096 chiều, mỗi chiều trong đó đại diện cho một đặc trưng của bức ảnh đó. Các đặc trưng này sẽ được sử dụng để so sánh với các đặc trưng của các bức ảnh khác trong cơ sở dữ liệu để tìm kiếm các ảnh tương tự trong hệ thống CBIR.

## CHƯƠNG 3: THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

### 3.1. Cơ sở dữ liệu

#### 3.1.1. Cơ sở dữ liệu ảnh

Tập dữ liệu được sử dụng trong thử nghiệm là tập dữ liệu Vegetables bao gồm 900 ảnh được tổ chức thành 9 lớp khác nhau: Radish, Bottle\_Gourd, Capsicum, Cabbage, Bean, Carrot, Papaya, Broccoli, Cauliflower. Mỗi lớp có 100 ảnh, tất cả đều là ảnh màu, định dạng .jpg, kích thước mỗi ảnh là 224x224. Hình 3.1 minh họa cho một số mẫu ảnh tương ứng với 9 lớp trong tập dữ liệu này:



*Hình 3.1. Ảnh minh họa cho 9 lớp trong tập dữ liệu Vegetables*

#### 3.1.2. Trích chọn đặc trưng

Trong lần thử nghiệm này, em sử dụng mô hình mạng CNN – VGG19 để trích chọn đặc trưng của từng hình ảnh. Quá trình tiền xử lý diễn ra như sau:

- B1: Resize ảnh, đưa ảnh về chuẩn màu RGB.
- B2: Trích chọn đặc trưng của từng ảnh trong CSDL ảnh.
- B3: Đặc trưng của mỗi hình ảnh sau khi trích chọn được mô tả thành một vector đặc trưng tương ứng.

Sau khi thực hiện trích chọn đặc trưng, kết quả trả về là tập tin vector đặc trưng có định dạng “.npz”, số lượng tập tin “.npz” trả về tương ứng với số lượng ảnh ta có trong cơ sở dữ liệu, quá trình này được thực hiện offline.



```

# Import the libraries
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg19 import VGG19, preprocess_input
from tensorflow.keras.models import Model
from pathlib import Path
from PIL import Image
import numpy as np

class FeatureExtractor:
    def __init__(self):
        base_model = VGG19(weights='imagenet')
        self.model = Model(inputs=base_model.input, outputs=base_model.get_layer('fc1').output)

    def extract(self, img):
        # Resize the image
        img = img.resize((224, 224))
        # Convert the image color space
        img = img.convert('RGB')
        # Reformat the image
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        x = preprocess_input(x)
        # Extract Features
        feature = self.model.predict(x)[0]
        return feature / np.linalg.norm(feature)

count = 0
if __name__ == '__main__':
    fe = FeatureExtractor()

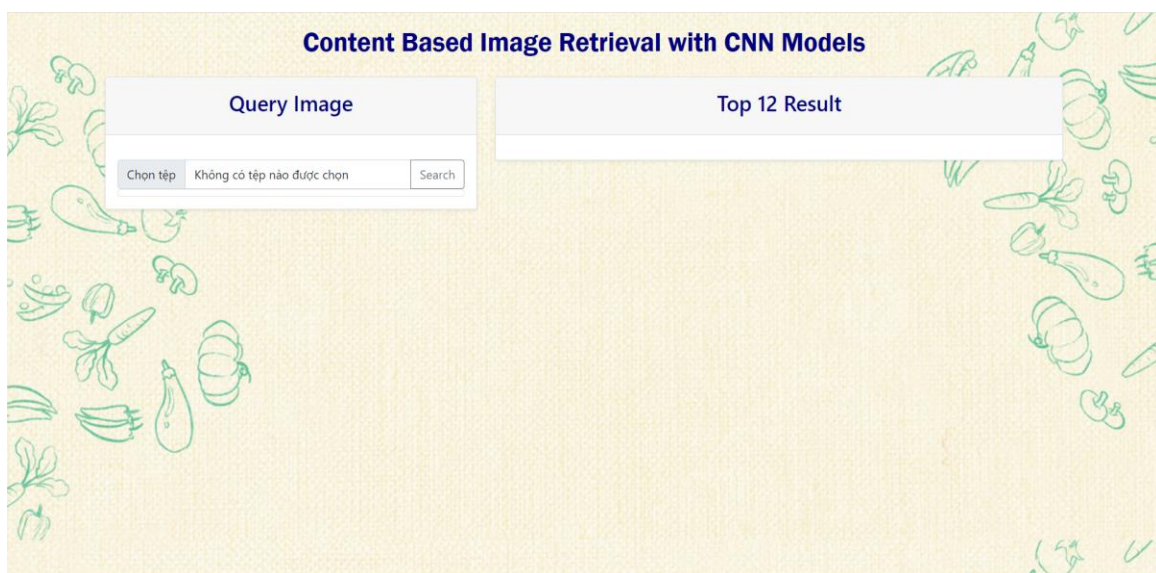
    for img_path in sorted(Path("./static/database").glob("*.jpg")):
        count+=1
        feature = fe.extract(image.open(img_path))
        feature_path = Path("./static/FeaturesVGG19") / (img_path.stem + ".npy")
        np.save(feature_path, feature)
    print('Number of images:', count)

```

Hình 3.2. Code thực hiện trích chọn đặc trưng sử dụng mô hình VGG19

## 3.2. Chương trình thực nghiệm

### 3.2.1. Giao diện



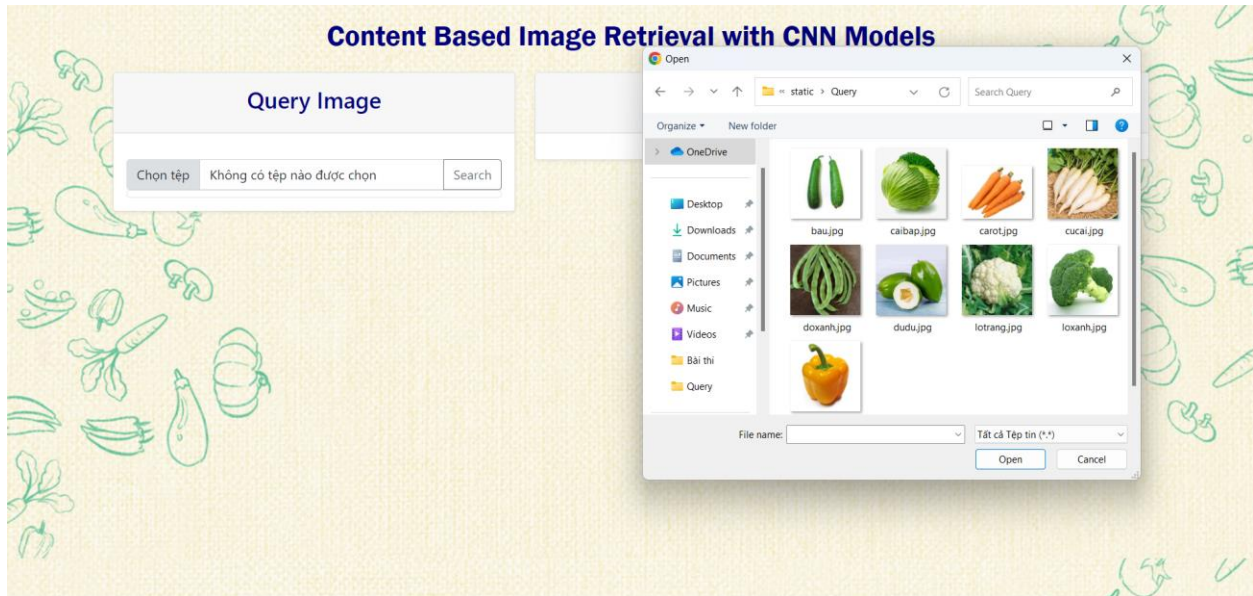
Hình 3.3. Hình ảnh giao diện chương trình thử nghiệm



### 3.2.2. Các bước thực hiện truy vấn ảnh

#### **Bước 1.** Chọn hình ảnh cần truy vấn

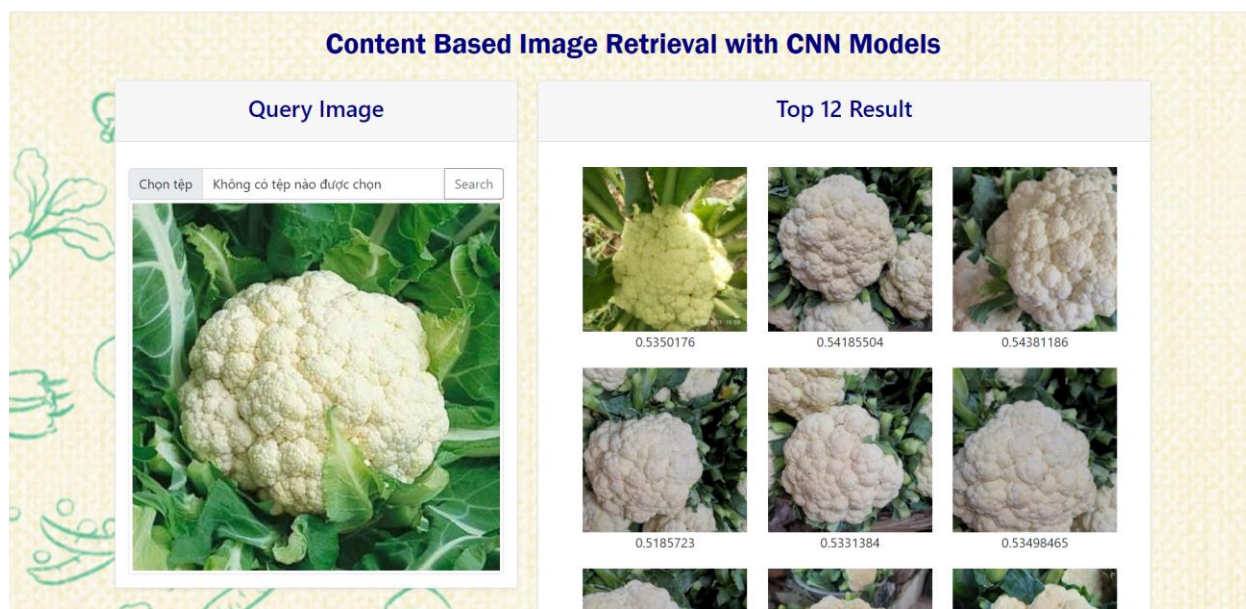
- Người dùng chọn hình ảnh cần truy vấn bằng cách bấm vào nút *Chọn tệp* -> *Chọn hình ảnh cần truy vấn* -> *Open*.
- Chương trình tự động trích chọn đặc trưng ảnh cần truy vấn.
- Hiện thị tên ảnh cần truy vấn ở mục chọn tệp.



Hình 3.4. Ảnh minh họa bước chọn ảnh cần truy vấn

#### **Bước 2.** Tra cứu ảnh

- Người dùng tiến hành truy vấn hình ảnh bằng cách bấm vào nút *Search*.
- Chương trình tính khoảng cách giữa ảnh cần truy vấn với tất cả các ảnh trong CSDL bằng hàm ***distance với độ đo khoảng cách L2***, sau đó sắp xếp theo thứ tự giảm dần các khoảng cách đồng thời cập nhật chỉ số của các ảnh trong CSDL theo thứ tự khoảng cách đã sắp xếp, hiển thị 12 hình ảnh có khoảng cách nhỏ nhất ở mục *Top 12 Result*.



*Hình 3.5. Ảnh minh họa kết quả truy vấn*

### 3.3. Đánh giá kết quả

Qua kết quả thực nghiệm, ta thấy được việc sử dụng trích xuất đặc trưng CNN VGG19 cùng độ đo khoảng cách L2, về cơ bản cho kết quả và độ chính xác tương đối tốt. Tuy nhiên, bên cạnh đó vẫn còn một số hạn chế như:

- Nhạy cảm với ánh sáng: Một hình ảnh có thể có cùng màu sắc nhưng khác nhau về ánh sáng. Do đó, các đặc trưng có thể không phản ánh chính xác màu sắc thực tế của hình ảnh.
- Không phân biệt được hình dạng giữa các vật thể khác nhau: Các đặc trưng chỉ sử dụng thông tin màu sắc để trích xuất đặc trưng hình ảnh, do đó không phân biệt được giữa hình dạng của các vật thể khác nhau có cùng màu sắc.
- Không phân biệt được sự khác biệt trong cường độ sáng: Các đặc trưng không đưa ra thông tin về cường độ sáng của hình ảnh, do đó không phân biệt được sự khác biệt trong cường độ sáng giữa các pixel.

## KẾT LUẬN

Thông qua bài báo cáo này, chúng ta đã có một cái nhìn cụ thể hơn về phương pháp tìm kiếm hình ảnh dựa trên nội dung (Content-Based Image Retrieval - CBIR). Về cơ bản, CBIR là một phương pháp tra cứu ảnh dựa trên nội dung của ảnh, trong đó thông tin đặc trưng của ảnh như màu sắc, hình dạng, kích thước, texture..., được sử dụng để phân loại và tra cứu hình ảnh.

Hệ thống tra cứu ảnh CBIR được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm y tế, giáo dục, truyền thông, quảng cáo, thương mại điện tử và nhiều lĩnh vực khác. Các ứng dụng CBIR giúp người dùng tìm kiếm hình ảnh một cách nhanh chóng và hiệu quả hơn so với các phương pháp tra cứu ảnh truyền thống. Bên cạnh đó, CBIR cũng có những hạn chế nhất định, các hệ thống CBIR đòi hỏi một lượng lớn dữ liệu huấn luyện để có thể đưa ra kết quả chính xác và phù hợp. Ngoài ra, việc phân tích thông tin đặc trưng của ảnh cũng có thể gặp phải nhiều thách thức về tính toán và đánh giá.

Tổng kết lại, hệ thống tra cứu ảnh CBIR là một công nghệ quan trọng trong việc quản lý và tìm kiếm ảnh. Mặc dù có những hạn chế nhất định. Tuy nhiên, với sự phát triển của công nghệ và thuật toán, CBIR đang trở thành một phương pháp tra cứu ảnh với nhiều ứng dụng thực tiễn và tiềm năng trong tương lai.

## **TÀI LIỆU THAM KHẢO**

- [1] Luca Piras and G. Giacinto, Neighborhood-based feature weighting for relevance feedback in content-based retrieval. 2009, IEEE Computer Society: In WIAMIS, pages. 238-241.
- [2] Luca Piras, Interactive search techniques for content-based retrieval from archives of images, in Electronic and Computer Engineering. 2011: Electrical and Electronic Engineering University of Cagliari, pages. 63-68.
- [3] D. N. F. Awang Iskandar, James A. Thom, and S.M.M. Tahaghoghi, Content-based Image Retrieval Using Image Regions as Query Examples. 2008: CRPIT Volume 75- Database technologies
- [4] Müller, H., Müller, W., & Squire, D. M. (2007). Knowledge-assisted content-based image retrieval. In Knowledge-Assisted Multimedia Analysis
- [5] C. Huang, Y. Han, Y. Zhang - A Method for Object-based Color Image Retrieval, Inter. Conference on Fuzzy Systems and Knowledge Discovery, IEEE, (2012) pp.1659-1663.
- [6] E. A. El-Kwae, M. R. Kabuka - Efficient Content-Based Indexing of Large Image Databases, ACM Trans. on Infor. Systems, 18(2) (2000) pp.171-210.