

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**ooOoo**



---

# **CONTENT-BASED IMAGE RETRIEVAL**

---



**Môn học: Thị giác Máy tính nâng cao – CS331.N12**

**Giảng viên: TS. Mai Tiến Dũng**

**Nhóm thực hiện:**

- **Lê Ngọc Mỹ Trang – MSSV: 20520817**
- **Vương Vĩnh Thuận – MSSV: 20521997**
- **Đoàn Phương Khanh - MSSV: 20521443**

***TP. Hồ Chí Minh, ngày 3 tháng 2 năm 2023***

## MỤC LỤC

<b>I. Giới thiệu đề tài.....</b>	<b>3.</b>
1. Lý do chọn đề tài.....	3.
2. Ý nghĩa thực tiễn.....	3.
3. Khái quát bài toán .....	4.
<b>II. Dataset .....</b>	<b>4.</b>
<b>III. Phương pháp .....</b>	<b>5.</b>
1. Phương pháp trích xuất đặc trưng.....	5.
1.1. Color Histogram .....	5.
1.2. Histogram of oriented gradients (HOG) .....	7.
1.3. Resnet50 .....	11.
2. Phương pháp so sánh .....	13.
2.1. Cosine similarity .....	13.
2.2. So sánh tuần tự .....	13.
2.3. Gom nhóm bằng K-Nearest Neighbor.....	13.
3. Phương pháp đánh giá.....	14.
3.1. Precision at k ( $P@K$ ) .....	14.
3.2. Recall at k ( $R@K$ ) .....	15.
3.3. Average Precision (AP) .....	15.
3.4. Mean Average Precision (MAP) .....	16.
<b>IV. Kết quả và kết luận.....</b>	<b>16.</b>
1. Công cụ thực nghiệm .....	16.
2. Kết quả thực nghiệm .....	16.
3. Kết luận .....	17.
<b>V. Hướng phát triển .....</b>	<b>18.</b>
<b>VI. Tài liệu tham khảo .....</b>	<b>18.</b>
<b>VII. Phân công.....</b>	<b>18.</b>

# **I. Giới thiệu đề tài**

## **1. Lý do chọn đề tài**

- Với sự phát triển vượt bậc của xã hội, công nghệ thông tin nói chung và những ứng dụng thực tiễn của chúng đang đồng hành song song với cuộc sống của con người. Nhờ sự phát triển vượt bậc của công nghệ mà đời sống của con người được nâng cao hơn và hiện đại hơn.
- Để mang đến sự tiện lợi cho người dùng thì ngày càng nhiều các công cụ thông minh xuất hiện. Cụ thể là Google Lens, Google Assistant, Google Photos, Google Image Search,... Các công nghệ này ra đời và hỗ trợ lẫn nhau để có thể xử lý và đảm bảo cho công việc tìm kiếm hình ảnh của người dùng được tối ưu nhất. Tuy nhiên, với một số lượng lớn hình ảnh trên các trang mạng xã hội như hiện nay sẽ gây khó khăn và làm mất khá nhiều thời gian của người dùng khi họ muốn tìm ra một hình ảnh mang tính trực quan theo nhu cầu của mình.
- Chính vì thế, truy xuất hình ảnh dựa trên nội dung đã ra đời và khắc phục được toàn bộ những hạn chế của việc truy tìm ảnh dựa vào văn bản kí tự. Với mô hình này, dữ liệu trả về sẽ mang ngữ nghĩa gần đúng với ảnh ban đầu được đưa vào, vừa tiết kiệm thời gian mà hiệu quả mang lại được cao hơn. Vì vậy, nhóm chúng em quyết định lựa chọn đề tài “ Truy xuất hình ảnh dựa trên nội dung” làm đề tài nghiên cứu cho nhóm mình.

## **2. Ý nghĩa thực tiễn**

Để tối đa hóa giá trị dữ liệu hình ảnh lớn trong thời đại số hóa hiện nay thì việc phát triển phương pháp truy xuất hình ảnh dựa trên nội dung là điều hoàn toàn cần thiết. Tìm kiếm một hình ảnh mang tính trực quan sẽ mang lại sự dễ dàng cho người tìm kiếm thông tin đi kèm ảnh, phục vụ cho nhiều mục đích từ giải trí, đáp ứng cho nhu cầu cá nhân đến nhiều lĩnh vực khác trong đời sống. Đây là một trong những nhu cầu thiết yếu mà người dùng cần phải có trong xã hội hiện đại ngày nay.

### 3. Khái quát bài toán

- Tìm kiếm ảnh theo nội dung (Content Based Images Retrieval - CBIR) hay truy vấn theo nội dung ảnh (Query Based Image Content - QBIC) là một ứng dụng của thị giác máy tính đối với bài toán tìm kiếm ảnh. “Dựa vào nội dung ảnh (Content Based)” nghĩa là việc tìm kiếm sẽ phân tích nội dung thực sự của các bức ảnh. Nội dung ảnh ở đây được thể hiện bằng màu sắc, hình dạng, kết cấu (texture), các đặc trưng cục bộ (local features), ... hay bất cứ thông tin nào có từ chính nội dung ảnh.
- Bài toán của chúng em gồm có:
  - Input: 1 tấm ảnh chứa đối tượng cần tìm kiếm
  - Output: Top 10 bức ảnh tương đồng với input

## II. Dataset

- Tên bộ dữ liệu: corel\_images [1]
- Nguồn: <https://www.kaggle.com/datasets/elkamel/corel-images>
- Nội dung:
  - + Bộ dữ liệu gồm 1000 ảnh được chia vào 10 nhóm, mỗi nhóm gồm 100 ảnh, mỗi ảnh chứa một hoặc nhiều đối tượng của nhóm đó
  - + Danh sách tên các nhóm (nhãn) của bộ dữ liệu:



**Beaches**



**Bus**



**Dinosaurs**



**Elephants**



**Flowers**



**Foods**



**Horses**



**Monuments**



**Mountains\_and  
d\_snow**



**People\_and\_villages\_in\_Africa**

- Tiến hành chia các ảnh trong bộ dữ liệu thành 2 phần, mỗi phần có số nhóm giống như danh sách nhóm gốc theo tỷ lệ 9/1

- + Database: chiếm 9/10 tổng số ảnh => có tổng 900 ảnh với 90 ảnh trên mỗi nhóm.
- + Query: chiếm 1/10 tổng số ảnh => có tổng 100 ảnh với 10 ảnh trên mỗi nhóm.

### III. Phương pháp

#### 1. Phương pháp trích xuất đặc trưng

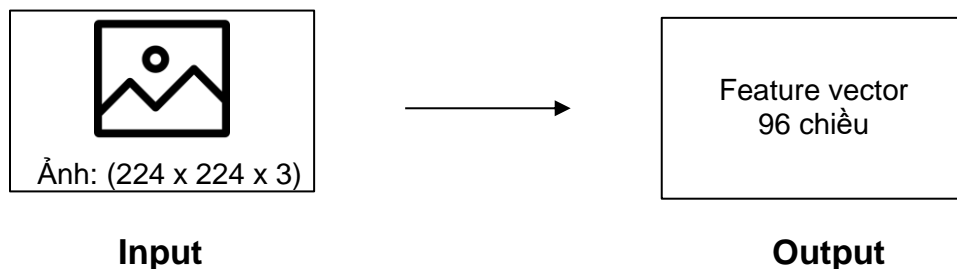
##### 1.1. Color Histogram

Là một trong những kỹ thuật CBIR đầu tiên, cho phép tìm kiếm thông qua các hình ảnh dựa trên **cấu hình màu** của chúng. Điều này có nghĩa là cho vào một hình ảnh, hình ảnh này sẽ được dịch sang biểu đồ dựa trên màu sắc, và chúng ta sẽ sử dụng biểu đồ này để truy vấn hình ảnh với những hình ảnh có cấu hình tương tự.

Hình ảnh bao gồm một mảng các điểm ảnh (pixel), và mỗi pixel thể hiện một màu sắc. Có nhiều không gian màu được sử dụng để tính toán các giá trị màu của pixel như: không gian màu RGB, không gian trực giác HSV,... Các đặc trưng được lưu giữ dưới dạng các vector biểu diễn cho các thông tin mô tả nội dung ảnh.

**Áp dụng cho bài toán:** Computer Vision (Thị giác máy tính), Image Processing (Xử lý ảnh)

##### 1.1.1. Input - Output



Hình 3.1. Hình minh họa input - output của color histogram.

### 1.1.2. Lược đồ màu

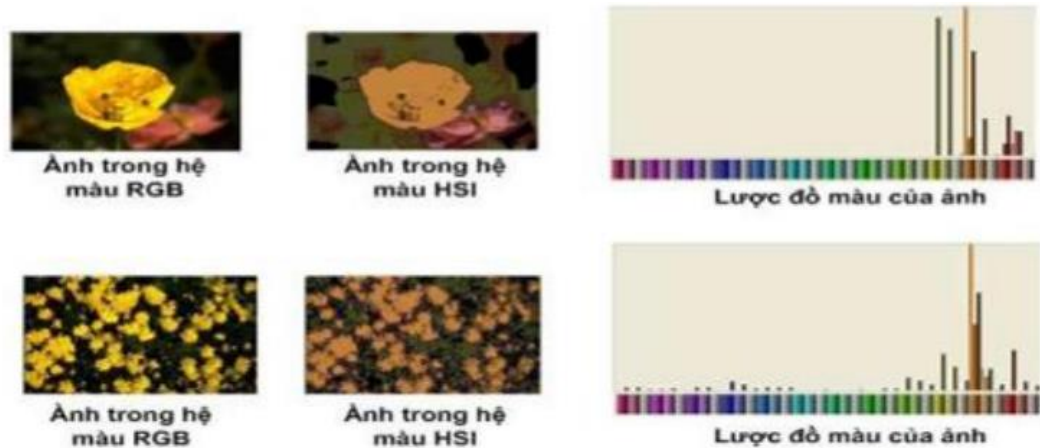
- Lược đồ màu (Histogram) là đại lượng đặc trưng cho phân bố màu cục bộ của ảnh. Được định lượng:

$$H(I_D, C_i) = \frac{m(I_D, C_i)}{n(I_D)}$$

- Trong đó:
  - +  $C_i$  : màu của điểm ảnh.
  - +  $n(I_D)$  : tổng số điểm ảnh trong ảnh.
  - +  $m(I_D, C_i)$  : Biểu diễn số điểm ảnh có giá trị màu
  - +  $H$  : lược đồ màu của ảnh.
- Độ đo tính tương tự về màu sắc giữa lược đồ màu của ảnh truy vấn  $H(I_Q)$  và lược đồ màu của ảnh trong CSDL ảnh  $H(I_D)$  được định nghĩa:

$$D_H(I_Q, I_D) = \frac{\sum_{j=1}^M \min(H(I_Q, j), H(I_D, j))}{\sum_{j=1}^M H(I_D, j)}$$

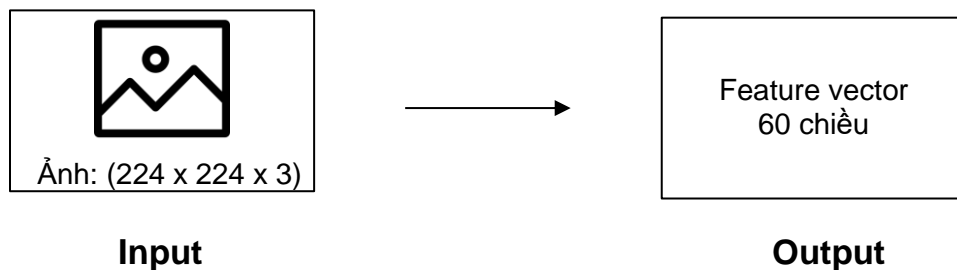
- Công thức trên cho ta thấy, tính tương tự về màu sắc được tính bằng phần giao của 2 lược đồ màu ảnh truy vấn  $H(I_Q)$  và ảnh trong CSDL ảnh  $H(I_D)$  . Kết quả sẽ là một lược đồ màu thể hiện độ giống nhau giữa 2 ảnh.
- Tuy nhiên vì lược đồ màu chỉ thể hiện phân bố màu sắc toàn cục của ảnh mà trong xét đến tính phân bố cục bộ của điểm ảnh nên có thể có 2 ảnh trông rất khác nhau nhưng lại cùng lược đồ màu.



Hình 3.2. Hình minh họa 2 ảnh có lược đồ màu giống nhau đến 70% nhưng khác nhau về ngữ nghĩa.

## 1.2. Histogram of oriented gradients (HOG)

### Bước 1: Tiền xử lý



- Trong bài toán HOG, để thuận tiện cho việc chia đều hình ảnh thành các khối, ô và tính toán đặc trưng ở các bước tiếp theo, chúng ta cần resize kích thước tất cả các hình ảnh trong tập dữ liệu về một kích thước chung.

### Bước 2: Tính Gradient

- Ở bước này được thực hiện bằng hai phép nhân chập ảnh gốc với 2 chiều, tương ứng với các toán tử lấy đạo hàm theo hai hướng  $O_x$  và  $O_y$ . Trong đó, 2 hướng tương ứng đó là:

$$D_x = [-1 \ 0 \ 1] \text{ và } D_y = [1 \ 0 \ -1]^T$$

(T là phép toán chuyển vị ma trận)

- Nếu ta có một ảnh input là  $I$ , ta sẽ có 2 ảnh đạo hàm riêng theo 2 hướng đó, theo công thức:

$$I_x = I * D_x \text{ và } I_y = I * D_y$$

- Khi đó, bạn có thể tính được Gradient bao gồm hai thành phần cường độ (Gradient Magnitude) và hướng (Gradient Direction) theo công thức:

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

**Bước 3:** Tính vector đặc trưng cho từng ô (cells)

- Để tính toán vector đặc trưng cho từng ô (cell), chúng ta cần chia hình ảnh thành các block, mỗi block lại chia đều thành các cell. Để xác định được số block, chúng ta sẽ sử dụng công thức sau:

$$n_{block\_image} = \left( \frac{W_{image} - W_{block} * W_{cell}}{W_{cell}} + 1 \right) * \left( \frac{H_{image} - H_{block} * H_{cell}}{H_{cell}} + 1 \right)$$

trong đó:

$W_{image}, W_{block}, W_{cell}$  : lần lượt là chiều rộng của ảnh, khối, ô

$H_{image}, H_{block}, H_{cell}$  : lần lượt là chiều dài của ảnh, khối, ô

- Sau khi xác định số block và kích thước mỗi block, cell, để tính toán vector đặc trưng cho từng cell, chúng ta cần:

1. Chia không gian hướng thành p bin (số chiều vector đặc trưng của ô).



2. Rời rạc hóa góc hướng nghiêng tại mỗi điểm ảnh vào trong các bin.
- Giả sử góc hướng nghiêng tại pixel ở vị trí (x,y) có độ lớn là  $\alpha(x,y)$ 
    1. Trường hợp rời rạc hóa unsigned-HOG với  $p=9$ :

$$B(x,y) = \text{round}\left(\frac{p \cdot \alpha(x,y)}{\pi}\right) \bmod p$$

2. Trường hợp rời rạc hóa signed-HOG với  $p=18$ :

$$B(x,y) = \text{round}\left(\frac{p \cdot \alpha(x,y)}{2\pi}\right) \bmod p$$

- Giá trị bin được định lượng bởi tổng cường độ biến thiên của các pixels thuộc về bin đó. Sau khi tính toán đặc trưng ô, ta sẽ nối các vector đặc trưng ô để thu được vector đặc trưng khối. Số chiều vector đặc trưng khối tính theo công thức:

$$size_{block} = n * size_{cell}$$

trong đó :

$n$  là số ô trong khối

$size_{cell}$  là số chiều của vector đặc trưng ô ( $size_{cell}=9$  nếu sử dụng “unsigned-HOG” và  $size_{cell}=18$  nếu sử dụng “signed-HOG”)

- Tiếp theo, tiến hành tính toán đặc trưng HOG tại mỗi cell *sử dụng không gian hướng 9 bin*, trường hợp “**unsigned-HOG**”. Hướng gradient sẽ chạy trong khoảng 0 độ đến 180 độ, trung bình 20 độ mỗi bin.
- Tại mỗi cell, xây dựng một biểu đồ cường độ gradient bằng cách vote các pixel vào biểu đồ. Trọng số vote của mỗi pixel phụ thuộc hướng và cường độ gradient (được tính toán từ bước 2) của pixel đó.

#### **Bước 4:** Chuẩn hóa khối (blocks)

- Có nhiều phương pháp có thể được dùng để chuẩn hóa khối. Gọi  $v$  là vector cần chuẩn hóa chứa tất cả các histogram của mọi khối.  $\|v(k)\|$  là giá trị chuẩn hóa của  $v$  theo các chuẩn  $k=1, 3$  và  $e$  là một hằng số nhỏ. Khi đó, các giá trị chuẩn hóa có thể tính bằng một trong những công thức sau:

$$\text{L2-norm: } f = \frac{v}{\sqrt{\|v_2\|^2 + e^2}}$$

$$\text{L1-norm: } f = \frac{v}{\|v_1\| + e}$$

$$\text{L1-sqrt: } f = \sqrt{\frac{v}{\|v_1\|^2 + e}}$$

- Ghép các vector đặc trưng khối sẽ thu được vector đặc trưng R-HOG cho ảnh. Số chiều vector đặc trưng ảnh tính theo công thức:

$$size_{image} = n * size_{block}$$

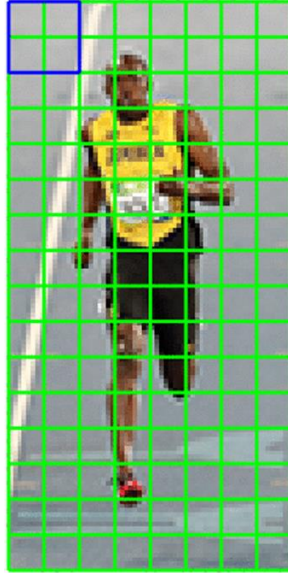
trong đó :

$n$  là số khối của hình ảnh.

$size_{block}$  là số chiều của vector đặc trưng khối

#### **Bước 5:** Tính toán vector đặc trưng HOG

- Giả sử hình ảnh sau khi được resize có kích thước là 64x128, chia thành các block 16x16 chồng nhau, sẽ có 7 block ngang và 15 block dọc, nên sẽ có  $7 \times 15 = 105$  blocks



Hình 3.3. (hình ảnh minh họa được lấy từ nguồn trên Google)

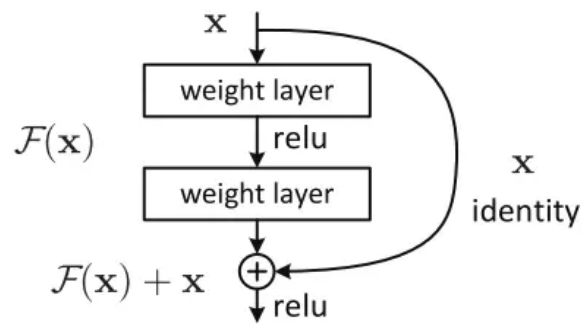
- Mỗi block gồm 4 cell. Khi áp dụng biểu đồ 9-bin cho mỗi cell, mỗi block sẽ được đại diện bởi một vector có kích thước  $36 \times 1$ .
- Vì vậy, khi nối tất cả các vector trong một block lại với nhau, ta sẽ thu được vector đặc trưng HOG của ảnh có kích thước  $105 \times 36 \times 1 = 3780 \times 1$ .

### 1.3. Resnet50

#### 1.3.1. Khái quát

Cùng với sự phát triển của Deep Learning, các mạng học sâu cũng có thể được sử dụng như phương pháp dùng để trích xuất đặc trưng của ảnh trong CBIR và ResNet50 là một đại diện tiêu biểu.

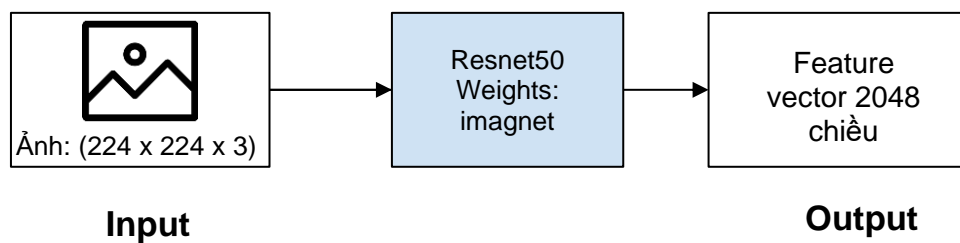
Mạng ResNet nói chung và ResNet50 nói riêng đều sử dụng kỹ thuật kết nối "tắt" để ánh xạ các đầu vào từ những layer trước đó tới những layer sau qua đó giải quyết vấn đề Vanishing Gradients khi tăng số lớp lên quá nhiều, khiến kết quả sau khi thực hiện Gradient Descent không thể hội tụ dẫn đến kết quả học không tốt.



Hình 3.4. Hình minh họa Residual Block của ResNet50

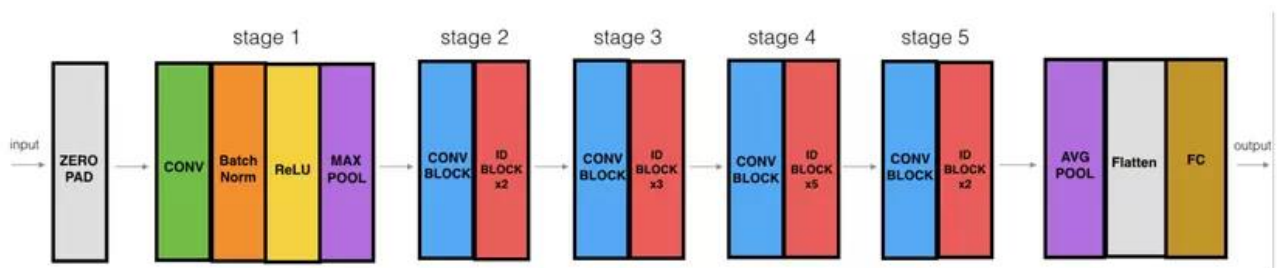
**Áp dụng cho dạng bài toán:** Localization, Detection, Segmentation.

### 1.3.2. Input - Output



Hình 3.5. Hình minh họa input - output của ResNet50

### 1.3.3. Tiến hành trích xuất đặc trưng



Hình 3.6. Hình minh họa các trạng thái của mạng ResNet50

Input của mạng là ảnh có kích thước (224,224,3), đặc trưng của ảnh sẽ được trích xuất qua từng lớp với các lớp đầu dùng để trích xuất các đặc trưng cấp thấp như:

góc, cạnh, hình khối vật thể. Các lớp cao hơn sẽ trích xuất các đặc trưng cấp cao (các đặc trưng riêng của từng đối tượng).

Output của mạng là 1 layer với 1000 node, tương ứng với 1000 class trong Imagenet (do nhóm có sử dụng weights của tập Imagenet mà không train lại), sử dụng Softmax với phân phối xác suất đầu ra ứng với từng class. Tuy nhiên, trong đề tài lần này, nhóm chỉ sử dụng output của các lớp gần cuối như vector đặc trưng của ảnh đó, cụ thể là kết quả của lớp average pooling với 2048 chiều thay vì sử dụng output với hàm softmax.

## 2. Phương pháp so sánh

### 2.1. Cosine similarity

- Cosine similarity so sánh độ tương đồng của hai vector khác 0 bằng cách tính cosine của góc giữa 2 vector đó
- Công thức:

$$\text{cosine similarity} = \frac{x \cdot y}{|x| \cdot |y|}$$

### 2.2. So sánh tuần tự

- Bước 1: Lần lượt tính cosine similarity của query đối với mỗi ảnh trong database
- Bước 2: Sắp xếp theo thứ tự giảm dần với độ tương đồng đã tính ở bước 1
- Bước 3: Chọn ra top-k kết quả từ bước 2, đây chính là kết quả cuối cùng.

### 2.3. Gom nhóm bằng K-Nearest Neighbor

- Đối với 1 tập hợp điểm trong không gian, các điểm dữ liệu tương tự nhau sẽ tồn tại gần nhau, dựa vào ý tưởng trên để tìm ra top-k ảnh trong database có độ tương đồng cosine similarity nhất với ảnh query.

	Thời gian	Kết quả
--	-----------	---------

<b>Tìm kiếm tuần tự</b>	Mất nhiều thời gian cụ thể là 1:33 cho 100 ảnh test	Giống nhau với cùng độ đo Cosine similarity
<b>KNN</b>	Mất ít thời gian, 26s cho 100 ảnh test	

### 3. Phương pháp đánh giá

#### 3.1. Precision at k (P@K)

- P@K cho biết có bao nhiêu ảnh trong top K output của 1 query có liên quan tới input.
- Công thức:

$$P@K = \frac{\text{true positives}@k}{(\text{true positives}@k) + (\text{false positives}@k)}$$

Trong đó:

- + true positives@k là số kết quả liên quan tới input trong số k kết quả của output
- + false positives@k là số kết quả không liên quan tới input trong số k kết quả của output
- Nhược điểm: hạn chế của P@K chính là chỉ xem xét số lượng kết quả đúng trong output mà không xét đến thứ tự của kết quả chính xác

Vd: cho ví dụ output dưới đây, với các ô màu vàng là ô chứa kết quả đúng

+ Output 1:

1	2	3	4	5
---	---	---	---	---

+ Output 2:

1	2	3	4	5
---	---	---	---	---

Áp dụng công thức P@K ở trên với K bằng 5 đối với 2 output này ta đều có được cùng một kết quả là  $P@5 = \frac{3}{5}$ , tuy nhiên, sự thật là các kết quả ở output 1 tốt hơn so với kết

quả ở output 2 vì nó trả về các ảnh liên quan tới ảnh query ở các vị trí đầu tiên đúng với mong muốn của người dùng hơn so với output 2.

### 3.2. Recall at k (R@K)

- R@K cho biết có bao nhiêu ảnh trong top K output có liên quan tới input trong số các ảnh có trong database.
- Công thức:

$$R@K = \frac{\text{true positives}@k}{(\text{true positives}@k) + (\text{false negatives}@k)}$$

Trong đó:

- + true positives@k là số kết quả liên quan tới input trong số k kết quả của output
- + false negatives@k là số kết quả liên quan tới input không có trong số k kết quả của output

### 3.3. Average Precision (AP)

- AP là độ chính xác trung bình của output đối với 1 input, cho biết các kết quả liên quan tới input có được xếp hạng lên đầu hay không, phương pháp này bổ sung cho hạn chế của việc P@K không xem xét đến thứ tự xuất hiện của các kết quả trong output
- Công thức:

$$AP = \frac{\sum nk = I(P(k) * rel(k))}{\text{number of relevant items}}$$

Trong đó:

- + P(k) là giá trị P@k của output
- + Rel(k) bằng 1 nếu như kết quả thứ k đang xét có liên quan tới input, ngược lại Rel(k) bằng 0 nếu như kết quả thứ k đang xét không liên quan tới input
- + false negatives@k là số kết quả liên quan tới input không có trong số k kết quả của output

### 3.4. Mean Average Precision (MAP)

- MAP độ chính xác trung bình của tập input..
- Công thức:

$$R@k = \frac{1}{Q} \sum Qq = IAP(q)$$

Trong đó:

- + Q là tổng số lượng query input
- + AP(q) là giá trị AP tại query thứ q

## IV. Kết quả và kết luận

### 1. Công cụ thực nghiệm

Để chạy ra kết quả thực nghiệm, nhóm đã sử dụng công cụ Google Colab.



Hình 4.1. Google Colab

### 2. Kết quả thực nghiệm

	<b>P@1</b>	<b>P@5</b>	<b>P@10</b>	<b>R@1</b>	<b>R@5</b>	<b>R@10</b>	<b>MAP</b>	<b>Memory</b>	<b>Time</b>
<b>HOG</b>	0.6	0.54	0.49	0.007	0.03	0.05	0.63	(900, 60)	0.23s/input



<b>Color Histogram</b>	00.7	0.57	0.51	0.078	0.032	0.06	0.67	(900, 96)	0.38s/input
<b>ResNet50</b>	<b>0.73</b>	<b>0.65</b>	<b>0.60</b>	<b>0.008</b>	<b>0.036</b>	<b>0.066</b>	<b>0.74</b>	(900, 2048)	0.67s/input

#### ❖ **Đánh giá:**

Qua bảng kết quả thực nghiệm trên, ta nhận thấy trong 3 phương pháp trích xuất đặc trưng:

- ResNet50 đạt P@1, P@5, P@10, R@1, R@5, R@10, MAP cao nhất, tuy nhiên thì thời gian train lâu nhất.
- HOG chiếm dung lượng ít nhất, ResNet50 chiếm dung lượng nhiều nhất.
- Thời gian thực thi của ResNet50 là lâu nhất, ngược lại thì của HOG là nhanh nhất.

### **3. Kết luận**

	<b>HOG</b>	<b>Color Histogram</b>	<b>ResNet50</b>
<b>Ưu điểm</b>	Khá hiệu quả khi ứng dụng tốt để phát hiện người với nhiều kích thước khác nhau - Tốn ít tài nguyên và chi phí tính toán	- Dễ thực hiện - Phù hợp để truy xuất tập trung vào màu sắc - Kết quả dễ hiểu và độ chính xác tương đối.	Độ chính xác cao

<b>Nhược điểm</b>	Độ chính xác chưa cao so với các thuật toán còn lại	Chỉ dựa vào cấu hình màu ảnh, nội dung ý nghĩa thực sự đằng sau nội dung hình ảnh chưa không được xem xét	<ul style="list-style-type: none"> <li>- Tốn nhiều thời gian</li> <li>- Ảnh có kích thước càng lớn thì thời gian cần để rút trích đặc trưng càng nhiều</li> </ul>
-------------------	---	---	---

## V. Hướng phát triển

Để mô hình được hoàn thiện hơn, nhóm sẽ tìm hiểu và giải quyết thêm một số vấn đề:

- Tăng cường dữ liệu, thu thập thêm nhiều hình ảnh để bộ dữ liệu đa dạng hơn.
- Sử dụng thêm các phương pháp tính khoảng cách khác như: L2 distance,... hoặc các phương pháp gom nhóm như: K-Means,...

## VI. Tài liệu tham khảo

[1]	AKIL ELKAMEL, corel_images   1K images, 2020. [Online]. Available: <a href="https://www.kaggle.com/datasets/elkamel/corel-images">https://www.kaggle.com/datasets/elkamel/corel-images</a>
[2]	<a href="#"><u>Histogram of Oriented Gradients</u></a>
[3]	<a href="#"><u>Evaluation Metrics For Information Retrieval</u></a>
[4]	<a href="#"><u>Giới thiệu mạng ResNet</u></a>

## VII. Phân công

Thành viên Phân việc		Lê Ngọc Mỹ Trang	Đoàn Phương Khanh	Vương Vĩnh Thuận
Dataset				
Code	Trích xuất đặc trưng			
	Đánh giá			
Báo cáo				
Slide				
Thuyết trình				