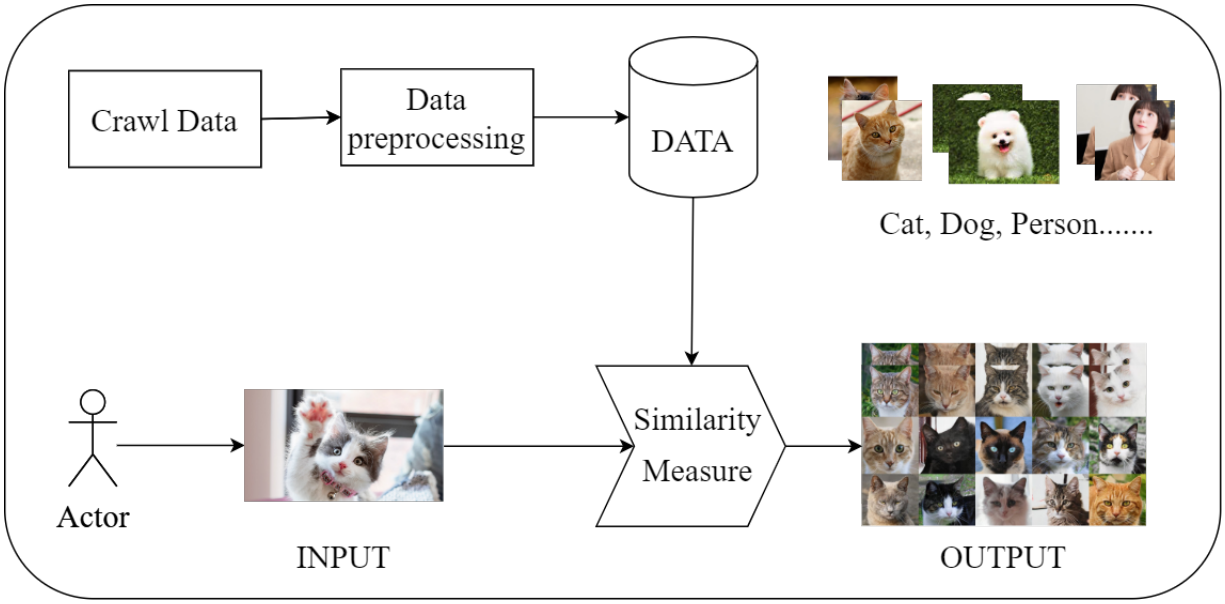
AI VIET NAM – COURSE 2022

Module 2 Project - Image Retrieval

Ngày 29 tháng 8 năm 2022

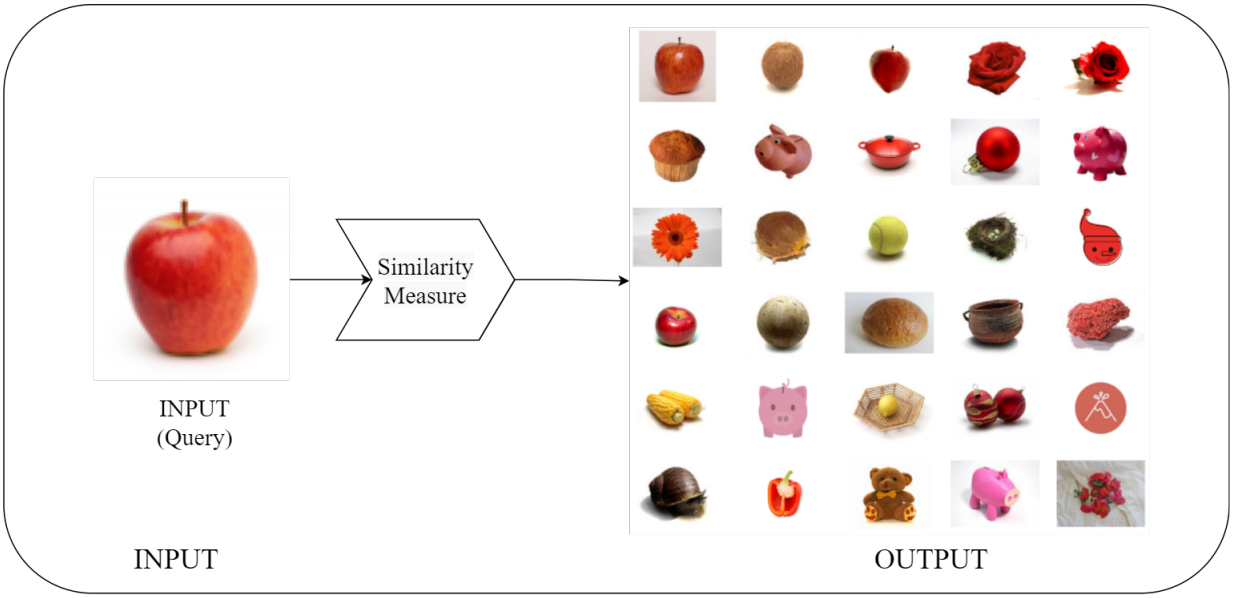
Truy vấn hình ảnh (Images Retrieval) là một bài toán thuộc lĩnh vực Truy vấn thông tin (Information Retrieval). Trong đó, nhiệm vụ của ta là xây dựng một chương trình trả về các hình ảnh (Images) có liên quan đến hình ảnh truy vấn (Query) đầu vào và các tài liệu được lấy từ một bộ dữ liệu hình ảnh cho trước như: Google Search Image, chức năng tìm kiếm sản phẩm bằng hình ảnh trên Shopee, Lazada, Tiki, ... Có rất nhiều cách thiết kế hệ thống truy vấn hình ảnh khác nhau, tuy nhiên về mặt tổng quát sẽ có một pipeline chung sau đây:



Hình 1: Pipeline tổng quan của một hệ thống Images Retrieval.

Dựa vào hình trên, có thể phát biểu Input/Output của một hệ thống truy vấn văn bản bao gồm:

- **Input:** Hình ảnh truy vấn **q** và bộ dữ liệu **C**.
- **Output:** Danh sách các hình ảnh **c** ($\mathbf{c} \in \mathbf{C}$) có sự tương quan đến hình ảnh truy vấn.



Hình 2: Demo về hệ thống truy vấn hình ảnh

Trong project này, chúng ta sẽ xây dựng một chương trình cho phép truy vấn hình ảnh sử dụng các phép đo độ tương đồng giữa các hình ảnh (Similarity Measure). Chúng ta sẽ xây dựng chương trình qua 3 giai đoạn sau:

- **Giai đoạn 1: Thu thập (Crawl Data) và Xử lý dữ liệu (Data Preprocessing).**
- **Giai đoạn 2: Xây dựng các hàm tính độ tương đồng các hình ảnh.**
- **Giai đoạn 3: Thử nghiệm và đánh giá kết quả.**

1. **Thu thập và xử lý dữ liệu:**

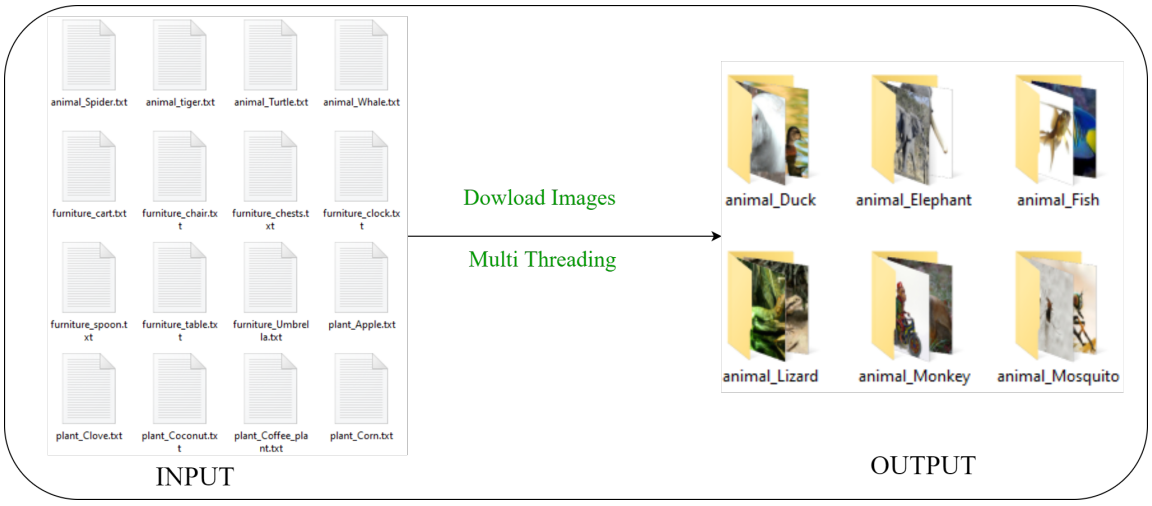
   (a) **Thu thập dữ liệu-Crawl data**

   Để xây dựng bộ dữ liệu cho bài toán Images Retrieval việc đầu tiên chúng ta làm là thu thập dữ liệu. Có rất nhiều cách thu thập dữ liệu: sử dụng google tìm kiếm, các thiết bị ghi hình, Image Downloader...Trong project này, chúng ta sẽ sử dụng thư viện BeautifulSoup và kỹ thuật xử lý đa luồng để thu thập hình ảnh từ trang freeimages.com. Bạn có thể tham khảo code tại file Crawler.ipynb, trong file code này thực hiện hai chức năng chính là:

   - **Get urls to txts: Gửi request đến trang freeimages.com, lưu các đường dẫn của ảnh vào file txt theo từng topic.**

Hình 3: Get urls to txts

- **Tải các hình ảnh theo từng topic vào các thư mục với tên tương ứng với từng topic**

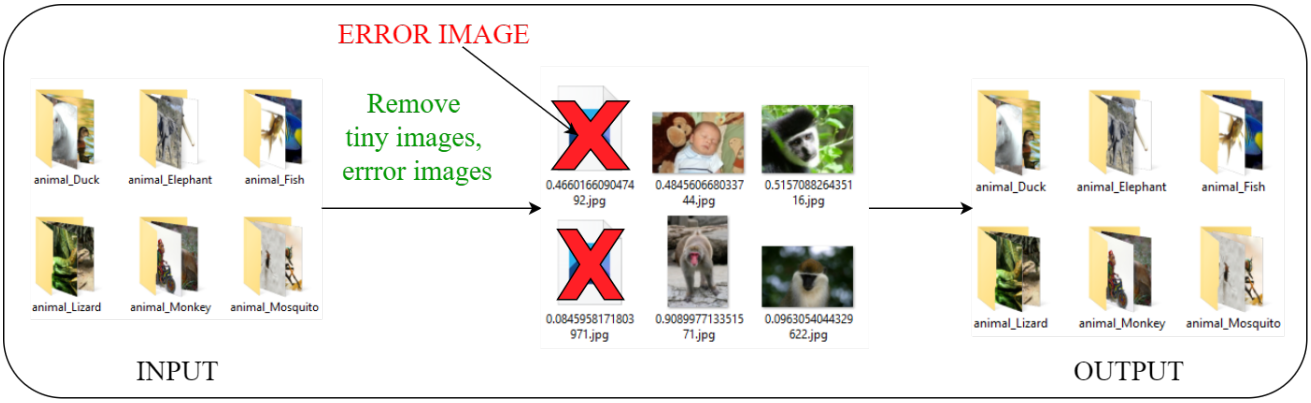


Hình 4: Get images from urls

(b) **Xử lí dữ liệu-Data preprocessing**

Trong thực tế sau khi thu thập dữ liệu về có thể xuất hiện các vấn đề như dữ liệu bị lỗi,ví dụ: hình ảnh quá mờ không chứa thông tin, kích thước hình ảnh quá nhỏ, hình ảnh bị hỏng, ... Vì vậy xử lí dữ liệu là bước tiền đề quan trọng để có được bộ dữ liệu tốt. Có rất nhiều phương pháp xử lí dữ liệu hình ảnh, trong dữ án này, chúng ta sẽ sử dụng một số phương pháp cơ bản như:

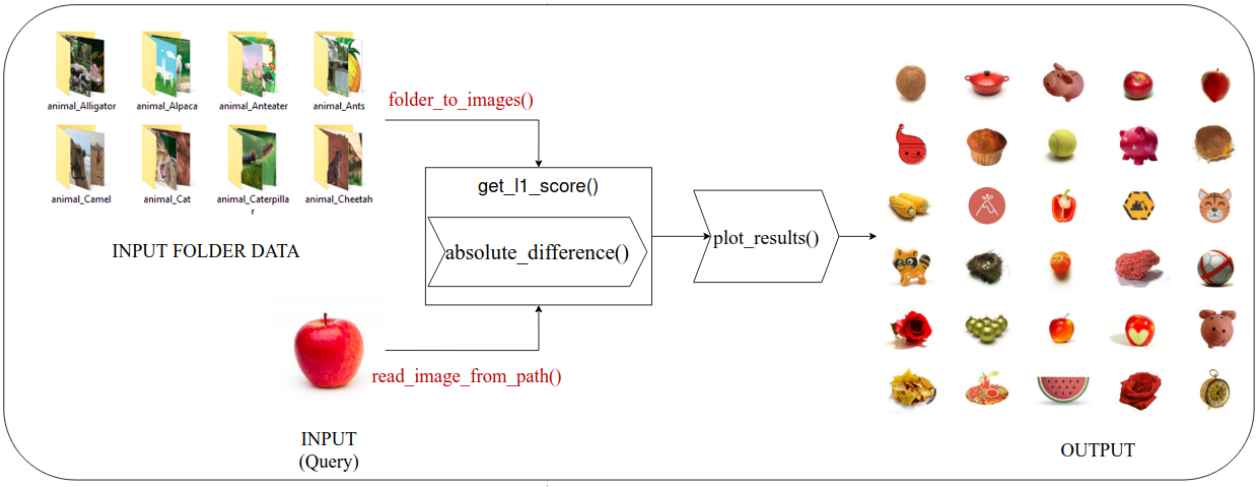
- Loại bỏ những hình ảnh lỗi (Không đọc được ảnh,...)
- Loại bỏ những hình ảnh có kích thước nhỏ hơn 10
- Loại bỏ hình ảnh có chanel khác 3 (chỉ nhận ảnh màu RGB)



Hình 5: Data preprocessing

2. **Xây dựng hàm tính độ tương đồng giữa hai hình ảnh:** Để có thể tìm được các hình ảnh có liên quan đến hình ảnh truy vấn, ta có thể sử dụng các công thức được dùng để đo sự tương đồng giữa hai ảnh, từ đó xây dựng một hàm $absolute\_difference()$ tính độ tương đồng giữa các hình ảnh. Trong ví dụ này chúng ta sẽ sử dụng hàm L1. Hàm L1 có công thức tính như sau:

$$l1\_norm(\vec{a}, \vec{b}) = \sum_{i=1}^{N} |a - b|$$



Hình 6: Image Retrieval với L1

Đầu tiên chúng ta sẽ import một số thư viện cần thiết:

```
from PIL import Image # Doc anh
import numpy as np # Xu ly ma tran
import os # Thao tac lay file, move file cua OS
import matplotlib.pyplot as plt
```

Sau đó chúng ta sẽ tạo hàm để đọc ảnh và thư mục ảnh:

```
def read_image_from_path(path, size):
    im = Image.open(path).resize(size)
    return np.asarray(im, dtype=np.float32)

def folder_to_images(folder,size):

    list_dir = [folder + '/' + name for name in os.listdir(folder) if
name.endswith((".jpg", ".png", ".jpeg"))]

    i = 0
    images_np = np.zeros(shape=(len(list_dir), *size, 3))
    images_path = []
    for path in list_dir:
        try:
            images_np[i] = read_image_from_path(path, size)
            images_path.append(path)
            i += 1

        except Exception:
            print("error: ", path)
#             os.remove(path)

    images_path = np.array(images_path)
    return images_np, images_path
```

Tiếp theo chúng ta sẽ tạo hàm sử dụng phép đo L1 $absolute\_difference()$:

```
def absolute_difference(query, X):
    axis_batch_size = tuple(range(1,len(X.shape)))
    return np.sum(np.abs(X - query), axis=axis_batch_size)
```

Đến đây chúng ta sẽ thực hiện tính toán để tính độ tương đồng giữa ảnh input và các hình ảnh trong bộ dữ liệu. Chúng ta sẽ tạo hàm $get\_l1\_score()$, hàm này sẽ trả về danh sách hình ảnh và giá trị độ tương đồng với từng ảnh.

```
def get_l1_score(root_img_path, query_path, size):
    dic_categories = ['scenery', 'furniture', 'animal', 'plant']
    query = read_image_from_path(query_path, size)
    ls_path_score = []
    for folder in os.listdir(root_img_path):
        if folder.split("_")[0] in dic_categories:
            path = root_img_path + folder
            # mang numpy nhieu anh, paths
            images_np, images_path = folder_to_images(path, size)
            rates = absolute_difference(query, images_np)
            ls_path_score.extend(list(zip(images_path, rates)))
    return query, ls_path_score
```

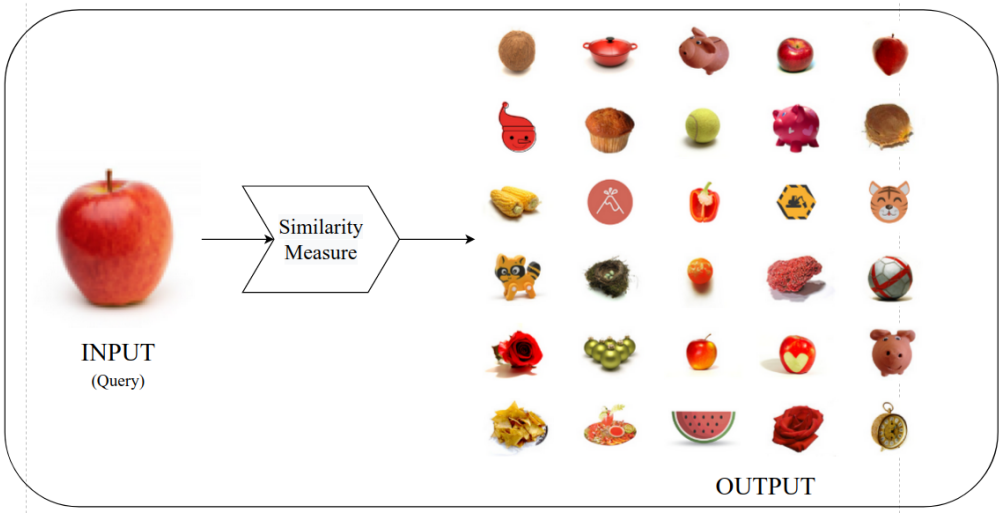
Chúng ta sẽ tạo hàm để hiển thị kết quả, kết quả trả về là 30 hình ảnh có độ tương đồng với hình ảnh Input cao nhất. Đối với mỗi độ đo khác nhau mà độ tương đồng cao nhất có thể có giá trị lớn nhất hoặc nhỏ nhất. Trong ví dụ này chúng ta sử dụng độ đo L1, vì vậy độ tương đồng cao nhất sẽ có giá trị nhỏ nhất.

```
def plot_results(query, ls_path_score):
    # Show query image
    plt.imshow(query/255.0)
    # Score
    fig = plt.figure(figsize=(15, 15))
    columns = 5
    rows = 6
    for i, path in enumerate(sorted(ls_path_score, key=lambda x : x[1])[:30], 1):
        img = np.random.randint(10, size=(10,10))
        fig.add_subplot(rows, columns, i)
        plt.imshow(plt.imread(path[0]))
        plt.axis("off")
    plt.show()
```

3. **Thử nghiệm và đánh giá kết quả:**
Cuối cùng là phần thú vị nhất, hãy xem thành quả mà chúng ta đạt được:

```
root_img_path = "/content/images/"
query_path = "apple.jpg"
size = (80, 80)
query, ls_path_score = get_l1_score(root_img_path, query_path, size)
plot_results(query, ls_path_score)
```

Và đây là kết quả chúng ta đạt được, nhìn chung khá thú vị:



Hình 7: Kết quả sử dụng Image Retrieval với L1

Phần đánh giá chi tiết kết quả trên mỗi độ đo mình sẽ nhường cho các bạn, bạn kéo xuống xem phần câu hỏi nhé!

Như vậy, trong project này các bạn sẽ thực hiện các công việc như sau:

1. Các bạn đọc hiểu và sử dụng file Crawler.ipynb, để thực hiện tạo bộ dữ liệu cho bài toán Images Retrieval.(Lưu ý: Nếu bạn sử dụng google colab,bộ dữ liệu sau khi xử lí cần nén lại, sau đó lưu vào drive để sử dụng cho các bước tiếp theo)

2. Bạn đọc hướng dẫn và hoàn thành code trong file PreprocessingData.ipynb , mình khuyến khích bạn sử dụng bộ dữ liệu từ câu 1 hoặc không bạn có thể tải về bộ dữ liệu có sẵn trong file PreprocessingData.ipynb. Sau khi có bộ dữ liệu bạn hãy thực hiện các yêu cầu sau:

    - Loại bỏ những hình ảnh lỗi (Không đọc được ảnh,...)
    - Loại bỏ những hình ảnh có kích thước nhỏ hơn 10
    - Loại bỏ hình ảnh có chanel khác 3 (chỉ nhận ảnh màu RGB)

```
for p in list_dir:
    try:
        ################### Your Code Here ####################
        #Step1:Open image, sử dụng Image của PIL để mở file ảnh theo path(biến p)
        #thu được biến img chứa ảnh(lưu ý: biến thu được chứa ảnh dạng PIL)


        #Step2: Verify image, Sau khi mở ảnh ở step1, thu được biến img chứa ảnh,
        #bạn sử dụng .verify() để phát hiện ảnh lỗi


        #Step3: Open image, Vì sau khi verify() hình ảnh sẽ bị đóng lại, vì vậy bạn cần mở lại hình ảnh như Step1.


        #Step4: Check width of image, nếu hình ảnh có width<10 thì xoá ảnh


        #Step5: Only 3 channle image (color image), convert ảnh từ PIL sang numpy
        #nếu hình ảnh có channel khác 3 thì xóa ảnh.
        ######################################################

    except Exception as e:
        print(e)
        count += 1
        print("error: ", p)
        os.remove(p) # Các trường hợp ngoại lệ, ảnh lỗi,... sẽ bị xóa
```

Hình 8: Hướng dẫn Data Preprocessing

- **Note 1**: Nếu bạn sử dụng google colab,bộ dữ liệu sau khi xử lí cần nén lại, sau đó lưu vào drive để sử dụng cho các bước tiếp theo
- **Note 2**: (Đọc thêm) Trong file PreprocessingData.ipynb ở session 1 là ví dụ export các topic và nội dung của từng topic sang file csv sử dụng pandas, giúp cho việc thống kê thông tin khi crawl data.

3. Các bạn sử dụng bộ dữ liệu từ câu 2 hoặc bạn có thể tải về bộ dữ liệu có sẵn tại Experiment-Methods.ipynb để hoàn thiện một chương trình truy vấn hình ảnh. File trên cũng đề cập về ví dụ sử dụng L1 với các bước chi tiết. Các bạn nên đọc và hiểu code ví dụ sử dụng L1 để thực hiện các yêu cầu bên dưới.

    - **Gợi Ý**: Các bạn đọc hiểu file ExperimentMethods.ipynb về ví dụ dùng L1. Sau đó các bạn thực hiện các yêu cầu bên dưới. (Các bạn có thể tùy ý làm hoặc làm theo những gợi ý sau: viết hàm để tính độ tương đồng ví dụ L1 là hàm $absolute\_difference$, sau đó viết hàm get_xx_score xx là tên measure method, cuối cùng viết hàm $plot\_results$ để show được input và 30 ảnh có độ tương đồng nhất với input (các bạn lưu ý vì mỗi measure method sẽ có độ lớn khác nhau để biểu thị độ tương đồng))
    - Thực hiện xếp hạng theo điểm tương đồng sử dụng phép đo L2 với công thức:

$$l2\_norm(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^{N} (a-b)^2}$$

    - Thực hiện xếp hạng theo điểm tương đồng sử dụng phép đo Cosine Similaritye với công thức:

$$cosine\_similarity(\vec{a}, \vec{b}) = \frac{a \cdot b}{\|a\|\|b\|} = \frac{\sum_{i=1}^{N} a_i b_i}{\sqrt{\sum_{i=1}^{N} a_i^2}\sqrt{\sum_{i=1}^{N} b_i^2}}$$

    - (Optional) Thực hiện xếp hạng theo điểm tương đồng sử dụng phép đo Correlation Coefficient với công thức:

$$p_{xy} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_x \sigma_y} = \frac{N(\Sigma_i x_i y_i) - (\Sigma_i x_i)(\Sigma_i y_i)}{\sqrt{N\Sigma_i x_i^2 - (\Sigma_i x_i)^2}\sqrt{N\Sigma_i y_i^2 - (\Sigma_i y_i)^2}}$$

4. (Optional) Các bạn hãy so sánh và đánh giá kết quả thu được của 4 measure ở trên.

- Hết -