ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN

COLOR COMPREHENSION

Môn học: **Toán ứng dụng và thống kê cho Công Nghệ Thông Tin**

Giảng viên hướng dẫn: **Phan Thị Phương Uyên**

Sinh viên thực hiện: **Đoàn Ngọc Mai**

Mã số sinh viên: **21127104**

Mục lục

1) Ý tưởng thực hiện

a. Đọc và chuẩn bị dữ liệu:

- Đầu tiên, cho phép người dung nhập tên tệp hình ảnh và số lượng cụm mong muốn.
- Hình ảnh được đọc và chuyển thành mảng numpy image. Hình ảnh cũng được chuyển đổi thành dạng phẳng (flat_image) để tiện xử lý.

b. Khởi tạo trung tâm cụm:

- Tạo hàm **initialize_centroids** được sử dụng để khởi tạo các trung tâm cụm. Có hai phương pháp khởi tạo:
    - **random**: Nếu người dùng chọn "random", các trung tâm sẽ được khởi tạo ngẫu nhiên trong khoảng từ 0 đến 255.
    - **in_pixels**: Nếu người dùng chọn "in_pixels", các trung tâm sẽ được lấy ngẫu nhiên từ các điểm ảnh trong hình ảnh.

c. Cập nhật trung tâm cụm và gán nhãn:

- Quá trình lặp max_iter lần được thực hiện:
- Các nhãn được gán cho từng điểm ảnh dựa trên trung tâm cụm hiện tại bằng cách tạo hàm **label_pixels**.
- Trung tâm cụm được cập nhật dựa trên các nhãn hiện tại và các điểm ảnh tương ứng trong cụm bằng cách sử dụng hàm **update_centroids**.
- Kiểm tra điều kiện dừng với hàm **check_convergence** để kiểm tra sự hội tụ, thông qua hàm này thì nếu không có sự thay đổi đáng kể trong trung tâm cụm, thuật toán dừng lại.

d. Thực thi thuật toán Kmeans

- Khởi tạo các centroids ban đầu:
    - Nhận đầu vào là img_1d (hình ảnh 1D) và số cụm k_clusters.
    - Sử dụng hàm initialize_centroids để khởi tạo các centroids ban đầu dựa trên img_1d và k_clusters.
    - Đặt nhãn ban đầu cho tất cả các điểm dữ liệu là -1.
- Lặp lại quá trình sau tối đa max_iter lần: Sao chép các giá trị centroids hiện tại vào old_centroids.
- Gán nhãn cho các điểm dữ liệu: Sử dụng phương thức label_pixels để gán nhãn cho mỗi điểm dữ liệu trong img_1d dựa trên các centroids hiện tại.
- Cập nhật centroids: Sử dụng phương thức update_centroids để cập nhật giá trị centroids dựa trên nhãn hiện tại và kích thước của centroids.
- Kiểm tra điều kiện dừng: Sử dụng phương thức check_convergence để kiểm tra xem centroids đã hội tụ hay chưa. Nếu hội tụ, thoát khỏi vòng lặp.

e. Hiển thị hình ảnh và lưu kết quả:

- Kết quả cuối cùng được tạo bằng cách thay thế mỗi điểm ảnh bằng trung tâm cụm tương ứng.
- Kết quả được hiển thị thông qua hai subplot: Hình ảnh gốc và kết quả của K-means.
- Nếu người dùng yêu cầu, kết quả cũng được lưu dưới dạng một tệp mới với định dạng được chỉ định.

2) Mô tả các hàm

a. Hàm initialize_centroids(img, k_clusters, init_type)

- Đầu vào:
    - img: Hình ảnh đầu vào dưới dạng mảng numpy.
    - k_clusters: Số lượng cụm mong muốn.
    - init_type: Loại phương pháp khởi tạo trung tâm cụm ("random" hoặc "in_pixels").
- Công dụng: Hàm này trả về các trung tâm cụm ban đầu dựa trên phương pháp khởi tạo được chọn. Hàm này được viết nhằm khởi tạo các trung tâm cụm ban đầu. Đây một bước quan trọng trong thuật toán K-means.
- Ưu điểm:
- Hỗ trợ hai phương pháp khởi tạo trung tâm cụm: "random" và "in_pixels".
- Cung cấp linh hoạt cho việc chọn trung tâm cụm ban đầu.

b. Hàm update_centroids(img, labels, old_centroids_shape):

- Đầu vào:
    - img: Hình ảnh đầu vào dưới dạng mảng numpy.
    - labels: Mảng chứa nhãn của các điểm ảnh.
    - old_centroids_shape: Kích thước của ma trận trung tâm cụm từ lần lặp trước.
- Công dụng: Hàm này tính toán và cập nhật trung tâm cụm dựa trên các nhãn hiện tại của các điểm ảnh trong cụm. Hàm này đóng vai trò quan trọng trong quá trình cập nhật trung tâm cụm của thuật toán K-means.
- Ưu điểm:
- Tính toán trung tâm cụm mới dựa trên các điểm ảnh trong cụm.
- Hỗ trợ xử lý các cụm không có điểm ảnh nào.
- Nhược điểm: Không xử lý trường hợp khi labels không hợp lệ.

c. Hàm label_pixels(img, centroids):

- Đầu vào:
    - img: Hình ảnh đầu vào dưới dạng mảng numpy.

    - centroids: Các trung tâm cụm hiện tại.
- Công dụng: Hàm này gán nhãn cho từng điểm ảnh dựa trên trung tâm cụm gần nhất. Hàm này là bước quan trọng để gán nhãn cho các điểm ảnh trong thuật toán K-means.
- Ưu điểm:
- Sử dụng tính toán vector hóa để tính toán khoảng cách giữa các điểm ảnh và trung tâm cụm.
- Hiệu suất tính toán cao hơn so với việc sử dụng vòng lặp.

d. Hàm kmeans(img_1d, k_clusters, max_iter, init_centroids)

- Đầu vào:
    - img_1d: Mảng 1D chứa các điểm ảnh.
    - k_clusters: Số lượng cụm mong muốn.
    - max_iter: Số lần lặp tối đa của thuật toán.
    - init_centroids: Loại phương pháp khởi tạo trung tâm cụm ("random" hoặc "in_pixels").
- Công dụng: Hàm này thực hiện thuật toán K-means để phân đoạn hình ảnh thành các cụm.
- Ưu điểm:
- Hỗ trợ cách khởi tạo trung tâm cụm linh hoạt.
- Kiểm tra điều kiện dừng để tăng hiệu suất và tránh việc lặp vô hạn.
- Nhược điểm: Không xử lý trường hợp khi init_centroids không hợp lệ.

e. Hàm load_image(input_file)

- Đầu vào: input_file, đại diện cho tên của tệp hình ảnh đầu vào mà người dùng muốn mở và chuyển đổi thành mảng numpy.
- Công dụng: Hàm này nhận một tên tệp đầu vào và trả về hình ảnh đã được chuyển đổi thành mảng numpy. Hàm này được viết để kiểm tra định dạng tệp đầu vào và chuyển đổi hình ảnh thành mảng numpy.
- Ưu điểm: Hàm kiểm tra và xử lý định dạng tệp đầu vào để đảm bảo tính hợp lệ và đáng tin cậy. Nó sử dụng vòng lặp vô hạn để cho phép người dùng nhập lại tên tệp nếu định dạng không hợp lệ.
- Nhược điểm: Vòng lặp vô hạn này có thể dẫn đến một vòng lặp vô hạn nếu không có tệp hợp lệ.

f. Hàm flatten_image(image)

- Đầu vào: image, đại diện cho hình ảnh mà người dung muốn nén sau khi đã tải ảnh lên.
- Công dụng: Hàm này được sử dụng để chuyển đổi hình ảnh từ dạng đa chiều sang dạng mảng 1D để áp dụng thuật toán K-means.

- Ưu điểm: Hàm giúp đơn giản hóa việc xử lý hình ảnh trong thuật toán K-means.
- Nhược điểm: Không kiểm tra giá trị đầu vào hợp lệ cho image.

g. Hàm export_image(image, k_clusters, output_format)

- Đầu vào:
  - image: Một mảng numpy biểu diễn hình ảnh đã được phân đoạn bằng thuật toán K-means.
  - k_clusters: Số lượng cụm trong thuật toán K-means. Đây là tham số được sử dụng trong quá trình phân đoạn hình ảnh. Nó xác định số lượng cụm hoặc điểm trung tâm mà các điểm ảnh được gán vào.
  - output_format: Định dạng tệp đầu ra cho hình ảnh phân đoạn. Tham số này xác định định dạng tệp (png hoặc pdf) để lưu hình ảnh kết quả. Nếu giá trị không hợp lệ được cung cấp, hàm sẽ in ra thông báo lỗi.
- Công dụng: Hàm này xuất kết quả hình ảnh đã được áp dụng thuật toán K-means vào một tệp đầu ra có định dạng được chỉ định.
- Ưu điểm: Hàm hỗ trợ xuất kết quả dưới dạng tệp PNG hoặc PDF.

h. Hàm main:

- Công dụng: Hàm main được sử dụng để chạy thuật toán K-means trên một hình ảnh và hiển thị kết quả. Nó cho phép người dùng tùy chỉnh số cụm, loại centroids ban đầu, định dạng hình ảnh đầu ra và cung cấp thời gian thực thi của thuật toán.

3) Kết quả của của random và in_pixels, so sánh và giải thích

a) Trường hợp 1: Hình ảnh có màu sắc đơn giản và hình ảnh có kích thước nhỏ (1000 x 1342)

Hình ảnh ban đầu:

Hình ảnh sau khi qua thuật toán K-means:

	Random	**In_pixels**
K=3:		
K=5:		
K=7:		

- **Nhận xét:**
  - Sự khác biệt giữa hình ảnh ban đầu và hình ảnh sau khi thực thi thuật toán K-means:
    - Hình ảnh ban đầu: Hiển thị chú mèo với màu sắc và chi tiết ban đầu.
    - Hình ảnh sau khi qua K-means: Đây là hình ảnh sau khi áp dụng thuật toán K-means để phân đoạn hình ảnh thành các cụm màu sắc khác nhau. Các cụm màu sắc tạo ra các vùng có màu sắc tương tự, tiêu biểu nhất như:
      - Vùng màu vàng từ lông chú mèo
      - Vùng trắng cho bức tường phía sau
      - Vùng đen nhạt cho chiếc bóng của chú mèo
    - Ảnh với **k_clusters=3** và **centroids_type=random**:
      - Thuật toán K-means sẽ phân đoạn hình ảnh thành 3 cụm khác nhau dựa trên màu sắc của các điểm ảnh và các điểm trung tâm được khởi tạo ngẫu nhiên.
      - Ảnh kết quả sẽ hiển thị sự phân đoạn với 3 màu sắc khác nhau tương ứng với 3 cụm được tìm thấy.
    - Ảnh với **k_clusters=5** và **centroids_type=random**:
      - Thuật toán K-means sẽ phân đoạn hình ảnh thành 5 cụm khác nhau dựa trên màu sắc của các điểm ảnh và các điểm trung tâm được khởi tạo ngẫu nhiên..
      - Ảnh kết quả sẽ hiển thị sự phân đoạn với 5 màu sắc khác nhau tương ứng với 5 cụm được tìm thấy.
    - Ảnh với **k_clusters=7** và **centroids_type=random**:

        - Thuật toán K-means sẽ phân đoạn hình ảnh thành 7 cụm khác nhau dựa trên màu sắc của các điểm ảnh và các điểm trung tâm được khởi tạo ngẫu nhiên.
        - Ảnh kết quả sẽ hiển thị sự phân đoạn với 7 màu sắc khác nhau tương ứng với 7 cụm được tìm thấy.
    - Với **k_clusters=3** và **centroid_type=in_pixels**:
        - Thuật toán K-means sẽ phân đoạn hình ảnh thành 3 cụm khác nhau dựa trên màu sắc của các điểm ảnh và chọn ngẫu nhiên 3 điểm ảnh trong hình ảnh làm các điểm trung tâm ban đầu.
        - Ảnh kết quả sẽ hiển thị sự phân đoạn với 3 màu sắc khác nhau tương ứng với 3 cụm được tìm thấy.
    - Với **k_clusters=5** và **centroid_type=in_pixels**:
        - Thuật toán K-means sẽ phân đoạn hình ảnh thành 5 cụm khác nhau dựa trên màu sắc của các điểm ảnh và chọn ngẫu nhiên 5 điểm ảnh trong hình ảnh làm các điểm trung tâm ban đầu.
        - Ảnh kết quả sẽ hiển thị sự phân đoạn với 5 màu sắc khác nhau tương ứng với 5 cụm được tìm thấy.
    - Với **k_clusters=7** và **centroid_type=in_pixels**:
        - Thuật toán K-means sẽ phân đoạn hình ảnh thành 7 cụm khác nhau dựa trên màu sắc của các điểm ảnh và chọn ngẫu nhiên 7 điểm ảnh trong hình ảnh làm các điểm trung tâm ban đầu.
        - Ảnh kết quả sẽ hiển thị sự phân đoạn với 7 màu sắc khác nhau tương ứng với 7 cụm được tìm thấy.
- Nguyên nhân của sự thay đổi trong màu sắc:
    - Kết quả là hình ảnh đã được phân đoạn bằng thuật toán K-means, trong đó các điểm ảnh đã được thay thế bằng màu sắc của trung tâm cụm tương ứng.
    - Các cụm màu sắc trong hình ảnh sau khi qua K-means được xác định dựa trên trung tâm cụm và nhãn của từng điểm ảnh. Các điểm ảnh thuộc cùng một cụm sẽ có màu sắc tương tự nhau và tạo ra các vùng có màu sắc giống nhau trên hình ảnh kết quả.
    - Trường hợp của **random**, nghĩa là các trung tâm cụm (centroids) được chọn ngẫu nhiên từ miền giá trị màu sắc của hình ảnh ban đầu.
    - Trường hợp của **in_pixels**, nghĩa là các trung tâm cụm (centroids) được chọn ngẫu nhiên trong các pixel của ảnh.
    - Khi áp dụng quá trình chọn điểm ảnh này lên hình ảnh đã qua thuật toán K-means, các trung tâm cụm cuối cùng được chọn để đại diện cho các vùng màu sắc khác nhau trong hình ảnh. Điểm ảnh của kết quả được gán nhãn dựa trên trung tâm cụm gần nhất và được thay thế bằng màu sắc của trung tâm cụm tương ứng. Kết quả là hình ảnh đã qua phân đoạn thành các vùng màu sắc tương tự dựa trên số lượng cụm và trung tâm cụm đã chọn.

- Với việc khởi tạo trung tâm cụm ban đầu:
  - Mỗi phần tử trong mảng centroids được lấy ngẫu nhiên từ miền giá trị 0 đến 255 bằng hàm np.random.randint.
  - Quá trình này đảm bảo rằng các trung tâm cụm ban đầu được khởi tạo ngẫu nhiên trên miền giá trị màu sắc của hình ảnh.
- Với việc cập nhật trung tâm cụm:
  - Quá trình cập nhật trung tâm cụm được thực hiện trong vòng lặp của thuật toán K-means.
  - Trong mỗi vòng lặp, các điểm ảnh được gán nhãn vào các cụm dựa trên khoảng cách Euclidean giữa điểm ảnh và trung tâm cụm.
  - Sau đó, các trung tâm cụm được cập nhật bằng cách tính trung bình của các điểm ảnh trong cùng một cụm.
  - Quá trình này lặp lại cho đến khi trung tâm cụm không thay đổi đáng kể hoặc đạt đến số lần lặp tối đa (**max_iter**).
  - Kết quả là một tập hợp các trung tâm cụm cuối cùng và nhãn của từng điểm ảnh.

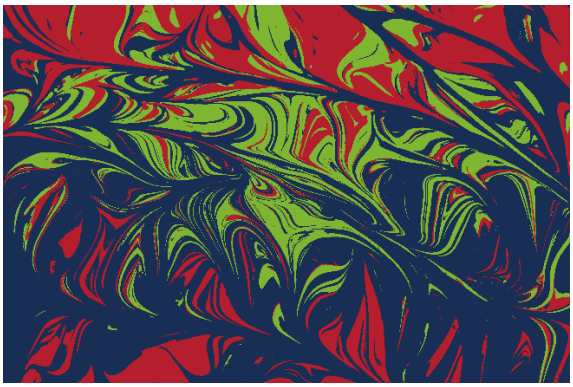
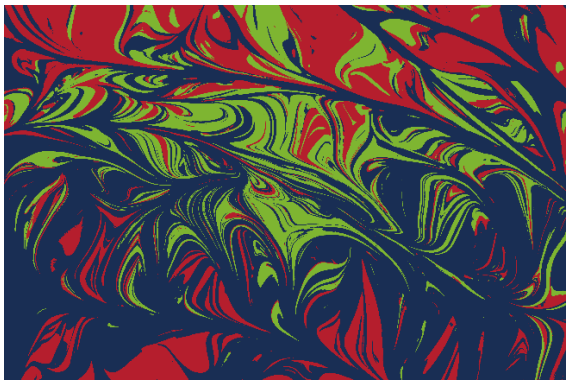
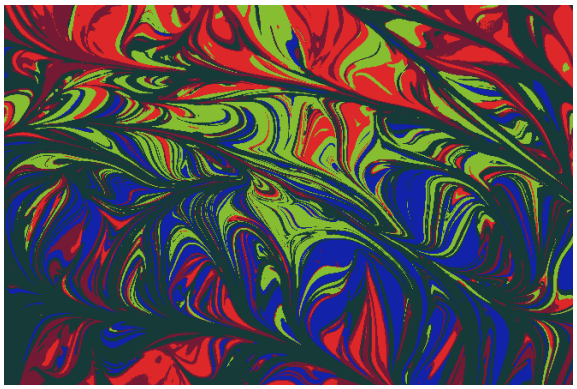
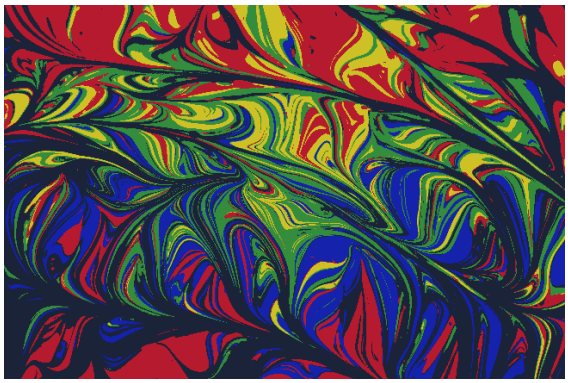
➢ **<u>Kết luận:</u>**

- Nguyên nhân của sự thay đổi là do quá trình phân đoạn và gán nhãn trong thuật toán K-means, cùng với trung tâm cụm được chọn. Quá trình phân đoạn tìm cách tối thiểu hóa tổng bình phương khoảng cách giữa các điểm ảnh và trung tâm cụm. Điều này có thể làm thay đổi màu sắc của các điểm ảnh ban đầu để tạo ra các vùng màu sắc tương tự.
- Nguyên nhân chính gây ra sự thay đổi là do lựa chọn trung tâm cụm ban đầu. Quá trình khởi tạo trung tâm cụm có thể là ngẫu nhiên hoặc dựa trên các điểm ảnh trong hình ảnh gốc.
- Tùy thuộc vào trung tâm cụm ban đầu và quá trình phân đoạn, màu sắc của các điểm ảnh ban đầu có thể thay đổi và tổ chức lại thành các cụm màu sắc khác nhau.
- Nhìn chung, sự phân đoạn màu sắc ở cùng các k_cluster nhưng với kiểu centroid là random hay in pixels, hình ảnh cho ra đều không có quá nhiều sự khác biệt.

b) Trường hợp 2: Hình ảnh có nhiều màu sắc và có kích thước hình ảnh lớn (5616 x 3744)

Hình ảnh ban đầu:

Hình ảnh sau khi phân đoạn thành các cụm:

	Random	**In_pixels**
K=3:		
K=5:		
K=7:		

➢ **Nhận xét:**

- Sự khác biệt giữa hình ảnh ban đầu và hình ảnh sau khi thực thi thuật toán K-means:
  - **Centroid_type=random**:
    - Với **k_clusters=3**:
      + Thuật toán K-means sẽ tìm 3 điểm trung tâm ban đầu ngẫu nhiên trong hình ảnh và gán các điểm ảnh vào 3 cụm tương ứng với các điểm trung tâm này.
      + Ảnh kết quả sẽ hiển thị sự phân đoạn với 3 màu sắc khác nhau tương ứng với 3 cụm được tìm thấy.
    - Với **k_clusters=5**:
      + Thuật toán K-means sẽ tìm 5 điểm trung tâm ban đầu ngẫu nhiên trong hình ảnh và gán các điểm ảnh vào 5 cụm tương ứng với các điểm trung tâm này.
      + Ảnh kết quả sẽ hiển thị sự phân đoạn với 5 màu sắc khác nhau tương ứng với 5 cụm được tìm thấy.
    - Với **k_clusters=7**:
      + Thuật toán K-means sẽ tìm 7 điểm trung tâm ban đầu ngẫu nhiên trong hình ảnh và gán các điểm ảnh vào 7 cụm tương ứng với các điểm trung tâm này.
      + Ảnh kết quả sẽ hiển thị sự phân đoạn với 7 màu sắc khác nhau tương ứng với 7 cụm được tìm thấy.
  - **Centroid_type=in_pixels**:
    - Với **k_clusters=3**:
      + Thuật toán K-means sẽ chọn 3 điểm ảnh ngẫu nhiên từ hình ảnh ban đầu làm các điểm trung tâm ban đầu.
      + Ảnh kết quả sẽ hiển thị sự phân đoạn với 3 màu sắc khác nhau tương ứng với 3 cụm được tìm thấy.
    - Với **k_clusters=5**:

+ Thuật toán K-means sẽ chọn 5 điểm ảnh ngẫu nhiên từ hình ảnh ban đầu làm các điểm trung tâm ban đầu.
+ Ảnh kết quả sẽ hiển thị sự phân đoạn với 5 màu sắc khác nhau tương ứng với 5 cụm được tìm thấy.

- Với **k_clusters=7**:
  + Thuật toán K-means sẽ chọn 7 điểm ảnh ngẫu nhiên từ hình ảnh ban đầu làm các điểm trung tâm ban đầu.
  + Ảnh kết quả sẽ hiển thị sự phân đoạn với 7 màu sắc khác nhau tương ứng với 7 cụm được tìm thấy.

- Nguyên do của sự thay đổi trong màu sắc:
  - Số lượng cụm (k_clusters): Khi tăng giá trị k_clusters, hình ảnh được phân chia thành nhiều cụm màu sắc khác nhau. Điều này dẫn đến việc tách biệt rõ ràng hơn giữa các vùng trong hình ảnh ban đầu. Khi giảm giá trị k_clusters, số lượng cụm màu sắc giảm, dẫn đến sự hòa trộn và gộp các vùng màu sắc lại với nhau.
  - Cách chọn điểm trung tâm ban đầu (centroid_type):
    - centroid_type=random: Khi các điểm trung tâm ban đầu được chọn ngẫu nhiên, việc khởi tạo các cụm và màu sắc ban đầu có tính ngẫu nhiên. Do đó, mỗi lần thực thi thuật toán có thể dẫn đến kết quả khác nhau. Sự khác biệt này phụ thuộc vào cách chọn ngẫu nhiên các điểm trung tâm ban đầu.
    - centroid_type=in_pixels: Khi các điểm trung tâm ban đầu được chọn từ hình ảnh ban đầu, sự khác biệt phụ thuộc vào vị trí và màu sắc của các điểm ảnh được chọn. Các điểm trung tâm ban đầu khác nhau sẽ dẫn đến việc khởi tạo các cụm ban đầu và kết quả cuối cùng khác nhau.
  - Quá trình hội tụ: Quá trình hội tụ của thuật toán K-means phụ thuộc vào điều kiện dừng và cách cập nhật các điểm trung tâm. Khi số lần lặp tới giới hạn (max_iter) hoặc khi khoảng cách giữa các điểm trung tâm không thay đổi đáng kể, thuật toán được coi là đã hội tụ. Sự khác biệt trong quá trình hội tụ có thể dẫn đến kết quả khác nhau giữa các lần chạy và giữa các giá trị k_clusters và centroid_type khác nhau.

➢ **Kết luận:**

- Thuật toán K-means tạo ra sự phân đoạn màu sắc trong hình ảnh, trong đó các điểm ảnh được gán vào các cụm màu tương ứng với các điểm trung tâm.
- Với số lượng cụm màu sắc khác nhau, thuật toán tạo ra các vùng màu sắc khác nhau và tách biệt hơn trong hình ảnh. Sự tách biệt giữa các vùng màu sắc phụ thuộc vào giá trị k_clusters.
- Sự lựa chọn ban đầu của các điểm trung tâm (centroid_type) cũng ảnh hưởng đến sự phân đoạn màu sắc và vị trí các cụm màu trong hình ảnh cuối cùng.

- Tùy thuộc vào trung tâm cụm ban đầu và quá trình phân đoạn, màu sắc của các điểm ảnh ban đầu có thể thay đổi và tổ chức lại thành các cụm màu sắc khác nhau.
- Nhìn chung, sự phân đoạn màu sắc ở cùng các k_cluster nhưng với kiểu centroid là random hay in pixels, hình ảnh cho ra đều không có quá nhiều sự khác biệt.

4) So sánh thời gian chạy với từng trường hợp ảnh và nhận xét:

a) Bảng thống kê thời gian thực thi dựa trên centroid type và số cluster (được đo theo đơn vị là giây):

Centroids type	Số k cluster	Trường hợp 1	Trường hợp 2
Random	K = 3	6.13	125.53
	K = 5	18.15	263.37
	K = 7	40.7	410.97
In_pixels	K = 3	5.40	205.46
	K = 5	15.96	122.18
	K = 7	45.97	446.23

b) So sánh giữa 2 trường hợp:

Lý do tại sao việc xuất ảnh ở trường hợp 2 lại mất nhiều thời gian hơn so với hình ảnh ở trường hợp 1 có thể phụ thuộc vào các yếu tố như sau:

- Kích thước của hình ảnh
- Số lượng cluster
- Loại khởi tạo centroid (random hoặc in_pixels) được chọn

Đặc biệt, việc xử lý hình ảnh có kích thước lớn và số lượng cluster nhiều hơn có thể yêu cầu thời gian tính toán lâu hơn, trong khi hình ảnh ở trường hợp 1 có thể có kích thước nhỏ hơn và số lượng cụm ít hơn, dẫn đến việc xuất ảnh nhanh hơn.

c) So sánh sự khác biệt giữa loại centroid được chọn:

- **Centroid_type=random:**
  - Thời gian thực thi thuật toán K-means với centroid_type=random tương đối nhanh hơn so với centroid_type=in_pixels.
  - Sự ngẫu nhiên trong việc chọn điểm trung tâm ban đầu dẫn đến sự phân tán ngẫu nhiên của các cụm màu ban đầu.

    - Điều này có thể giúp thuật toán hội tụ nhanh hơn do không yêu cầu quá nhiều lần lặp để đạt được kết quả tối ưu.
- **Centroid_type=in_pixels:**
    - Thời gian thực thi thuật toán K-means với centroid_type=in_pixels tương đối lâu hơn so với centroid_type=random.
    - Việc chọn điểm trung tâm ban đầu từ các điểm ảnh trong hình ảnh ban đầu tạo ra sự phụ thuộc và tính phức tạp hơn trong quá trình tính toán.
    - Điều này có thể dẫn đến việc thuật toán cần nhiều lần lặp hơn để hội tụ và đạt được kết quả tối ưu.

5) Tài liệu tham khảo

1. Vu, T. (2017, January 1). *Bài 4: K-Means Clustering*. Tiep Vu's Blog. https://machinelearningcoban.com/2017/01/01/kmeans/
2. Vu, T. (2017b, January 4). *Bài 5: K-Means Clustering: Simple Applications*. Tiep Vu's Blog. https://machinelearningcoban.com/2017/01/04/kmeans2/
3. Khan, S. S., & Ahmad, A. (2004, August 1). *Cluster center initialization algorithm for K-means clustering*. Pattern Recognition Letters. https://doi.org/10.1016/j.patrec.2004.04.007
4. StatQuest with Josh Starmer. (2018, May 23). *StatQuest: K-means clustering* [Video]. YouTube. https://www.youtube.com/watch?v=4b5d3muPQmA
5. First Principles of Computer Vision. (2021, May 26). *K-Means Segmentation | Image Segmentation* [Video]. YouTube. https://www.youtube.com/watch?v=22mpExWh1LY
6. Abubakr Shafique. (2019, November 13). *How to do K-Means Clustering on Images Using Python* [Video]. YouTube. https://www.youtube.com/watch?v=rfaVGfG3WBM