Họ và tên : Bùi Nguyễn Nhật Minh

MSSV : 21127105

Ý tưởng thực hiện

Nén hình ảnh sử dụng phương pháp Kmeans, gồm 4 bước chính :

1. Khởi tạo centroids (trung tâm)
2. Dán label cho từng điểm ảnh
3. Cập nhật lại centroids
4. Lặp lại bước 2, 3 cho tới khi centroids ít thay đổi hoặc đạt số lượt cho trước

Mô tả các hàm

Hàm main

- Sử dụng hàm get_user_inputs() để lấy được : ma trận màu, số lượng cluster, max_iter, cách khởi tạo trung tâm (centroids), tên ảnh, định dạng file ảnh sau khi nén màu, ma trận màu ban đầu
- Chạy hàm kmeans và trả về kết quả là centroids, labels
- Thực hiện hàm output để xuất ra hình sau khi nén

Hàm get_user_inputs

Hàm get_user_inputs được sử dụng để nhận đầu vào từ người dùng cho các thông số của thuật toán K-means và định dạng lưu trữ của hình ảnh.

Đầu vào :

- Không có đầu vào.

Đầu ra :

- flat_image: Ma trận 1 chiều của hình ảnh.
- k_clusters: Số cụm (k_clusters) cho thuật toán K-means.
- max_iter: Số lần lặp tối đa (max_iter) cho thuật toán K-means.
- init_centroids: Phương pháp khởi tạo điểm trung tâm (init_centroids) cho thuật toán K-means.
- image_name : Tên ảnh.
- save_format: Định dạng lưu trữ (save_format) cho tệp hình ảnh kết quả.
- image : Ma trận ảnh ban đầu.

Mô tả :

- Hàm get_user_inputs yêu cầu người dùng nhập tên của tệp hình ảnh, số cụm (k_clusters), số lần lặp tối đa (max_iter), phương pháp khởi tạo điểm trung tâm (init_centroids), và định dạng lưu trữ (save_format).
- Hàm kiểm tra tính hợp lệ của các đầu vào, bao gồm kiểm tra tệp hình ảnh tồn tại, số cụm (k_clusters) và số lần lặp tối đa (max_iter) là số nguyên dương, phương pháp khởi tạo điểm trung tâm (init_centroids) là 'random' hoặc 'in_pixels', và định dạng lưu trữ (save_format) chỉ chứa các ký tự chữ cái.

- Nếu người dùng nhập một giá trị không hợp lệ, một thông báo lỗi sẽ được hiển thị và yêu cầu người dùng nhập lại giá trị đúng.

Hàm label_pixels

Đầu vào:

- img: Một mảng numpy có kích thước (num_pixels, num_channels) chứa các giá trị màu RGB của các điểm ảnh trong hình ảnh.
- centroids: Một mảng numpy có kích thước (num_centroids, num_channels) chứa các giá trị màu RGB của các điểm trung tâm.

Đầu ra:

- labels: Một mảng numpy có kích thước (num_pixels,) chứa chỉ số của điểm trung tâm gần nhất cho mỗi điểm ảnh trong img.

Mô tả :

- Tính toán số lượng điểm ảnh (num_pixels) và số lượng điểm trung tâm (num_centroids) từ kích thước của img và centroids.
- Tạo một mảng distances với kích thước (num_centroids, num_pixels) để lưu trữ khoảng cách giữa các điểm ảnh và các điểm trung tâm.
- Sử dụng broadcasting, ánh xạ lại img thành reshaped_img có kích thước (num_pixels, 1, num_channels) để phù hợp với kích thước của centroids.
- Tính toán khoảng cách giữa mỗi điểm ảnh trong reshaped_img và các điểm trung tâm trong centroids bằng cách trừ các giá trị tương ứng và tính norm theo trục thứ 2.
- Sử dụng np.argmin để tìm chỉ số của điểm trung tâm gần nhất cho mỗi điểm ảnh, từ đó xác định nhãn cho mỗi điểm ảnh và lưu vào mảng labels.
- Trả về mảng labels đã được gán nhãn.

Hàm initialize_centroids

Hàm initialize_centroids được sử dụng để khởi tạo các điểm trung tâm cho thuật toán K-means dựa trên phương pháp được chỉ định. Đây là một phần quan trọng trong quá trình khởi tạo các cụm màu ban đầu cho thuật toán.

Đầu vào:

- img: Một mảng numpy có kích thước (num_pixels, num_channels) chứa các giá trị màu RGB của các điểm ảnh trong hình ảnh.
- k_clusters: Số lượng cụm (k_clusters) được yêu cầu cho thuật toán K-means.
- init_type: Phương pháp khởi tạo điểm trung tâm được chỉ định, có thể là 'random' hoặc 'in_pixels'.

Đầu ra:

- centroids: Một mảng numpy có kích thước (k_clusters, num_channels) chứa các giá trị màu RGB của các điểm trung tâm.

Mô tả:

- Trong trường hợp phương pháp khởi tạo là 'random':
    - Sử dụng hàm np.unique để tính toán các điểm ảnh duy nhất trong hình ảnh (unique_pixels).
    - Nếu số lượng điểm ảnh duy nhất (len(unique_pixels)) nhỏ hơn k_clusters, sẽ raise một ValueError để thông báo rằng số lượng điểm ảnh duy nhất không đủ.
    - Sử dụng np.random.choice để tạo ra các chỉ số ngẫu nhiên (indices) đảm bảo các điểm trung tâm duy nhất từ các điểm ảnh duy nhất có sẵn.
    - Cuối cùng, trả về các điểm ảnh tương ứng với các chỉ số ngẫu nhiên để tạo thành các điểm trung tâm (centroids).
- Trong trường hợp phương pháp khởi tạo là 'in_pixels':
    - Kiểm tra xem số lượng điểm ảnh (len(img)) có lớn hơn hoặc bằng k_clusters hay không. Nếu không, raise một ValueError để thông báo rằng số lượng điểm ảnh không đủ.
    - Tính toán các chỉ số duy nhất từ np.arange(len(img)) (unique_indices). Nếu số lượng chỉ số duy nhất nhỏ hơn k_clusters, raise một ValueError để thông báo rằng số lượng chỉ số duy nhất không đủ.
    - Sử dụng np.random.choice để tạo ra các chỉ số ngẫu nhiên (indices) từ các chỉ số duy nhất có sẵn và chọn các điểm ảnh tương ứng từ img.
    - Cuối cùng, trả về các điểm ảnh đã chọn để tạo thành các điểm trung tâm (centroids).

Hàm update_centroids

Hàm update_centroids được sử dụng để cập nhật các điểm trung tâm dựa trên các nhóm hiện tại của thuật toán K-means.

Đầu vào:

- img: Một mảng numpy có kích thước (num_pixels, num_channels) chứa các giá trị màu RGB của các điểm ảnh trong hình ảnh.
- labels: Một mảng numpy có kích thước (num_pixels,) chứa chỉ số của điểm trung tâm gần nhất cho mỗi điểm ảnh trong img.
- old_centroids_shape: Kích thước của mảng centroids trước khi cập nhật.

Đầu ra:

- centroids: Một mảng numpy có kích thước (k_clusters, num_channels) chứa các giá trị màu RGB của các điểm trung tâm đã được cập nhật.

Mô tả:

- Mảng centroids được khởi tạo với kích thước ban đầu là old_centroids_shape và các giá trị ban đầu là 0.
- Với mỗi điểm trung tâm:
    - Lấy ra các điểm ảnh trong cụm tương ứng bằng cách sử dụng img[labels == i].
    - Nếu số lượng điểm ảnh trong cụm khác 0, tính toán giá trị trung bình trên từng kênh màu của các điểm ảnh trong cụm bằng cách sử dụng np.mean(pixels, axis=0).
    - Cuối cùng, cập nhật giá trị của điểm trung tâm tương ứng trong mảng centroids.
    - Trả về mảng centroids đã được cập nhật.

Hàm Kmeans

Hàm kmeans thực hiện thuật toán K-means để giảm số lượng màu trong một hình ảnh.

Đầu vào:

- img_1d: Một mảng numpy 1 chiều có kích thước (num_pixels,) chứa giá trị màu RGB của từng điểm ảnh trong hình ảnh.
- k_clusters: Số lượng cụm (k_clusters) được yêu cầu cho thuật toán K-means.
- max_iter: Số lần lặp tối đa (max_iter) cho thuật toán K-means.
- init_centroids: Phương pháp khởi tạo điểm trung tâm (init_centroids) cho thuật toán K-means. Giá trị mặc định là 'random'.

Đầu ra:

- centroids: Một mảng numpy có kích thước (k_clusters, num_channels) chứa các giá trị màu RGB của các điểm trung tâm cuối cùng.
- labels: Một mảng numpy có kích thước (num_pixels,) chứa chỉ số của điểm trung tâm gần nhất cho mỗi điểm ảnh trong hình ảnh.

Mô tả:

- Hàm kmeans bắt đầu bằng việc khởi tạo các điểm trung tâm ban đầu bằng cách sử dụng hàm initialize_centroids.
- Mảng labels được khởi tạo với giá trị -1 cho mỗi điểm ảnh trong hình ảnh.
- threshold là ngưỡng để kiểm tra sự hội tụ của thuật toán. Nếu khoảng cách giữa các điểm trung tâm hiện tại và điểm trung tâm trước đó nhỏ hơn threshold, thuật toán sẽ dừng lại.
- Trong vòng lặp max_iter:
    - Hàm label_pixels được sử dụng để gán nhãn cho mỗi điểm ảnh dựa trên điểm trung tâm gần nhất.
    - Các điểm trung tâm trước đó được sao chép vào mảng old_centroids.
    - Hàm update_centroids được sử dụng để cập nhật các điểm trung tâm dựa trên nhãn hiện tại.
    - Khoảng cách Euclidean giữa các điểm trung tâm hiện tại và điểm trung tâm trước đó được tính bằng cách sử dụng np.sum((centroids - old_centroids) ** 2).
    - Nếu khoảng cách này nhỏ hơn ngưỡng threshold, thuật toán dừng lại.
- Kết quả cuối cùng là các điểm trung tâm cuối cùng và nhãn của từng điểm ảnh.

Hàm output

Hàm output được sử dụng để tạo và lưu hình ảnh kết quả sau khi áp dụng thuật toán K-means để giảm số lượng màu.

Đầu vào:

- centroids: Một mảng numpy có kích thước (k_clusters, num_channels) chứa các giá trị màu RGB của các điểm trung tâm.
- labels: Một mảng numpy có kích thước (num_pixels,) chứa chỉ số của điểm trung tâm gần nhất cho mỗi điểm ảnh trong hình ảnh gốc.

- name: Tên tệp đầu ra được tạo.
- extension: Phần mở rộng của tệp đầu ra (ví dụ: 'jpg', 'png',...).
- image: Mảng numpy chứa hình ảnh gốc.
- k_clusters: Số lượng cụm (k_clusters) được sử dụng trong thuật toán K-means.

Đầu ra :

- Không có đầu ra

Mô tả:

- Hàm output bắt đầu bằng việc thay thế từng điểm ảnh trong hình ảnh gốc bằng điểm trung tâm tương ứng. Điều này được thực hiện bằng cách sử dụng mảng labels để truy cập các điểm trung tâm từ centroids.
- Kết quả được chuyển đổi lại thành hình dạng ban đầu của hình ảnh và được ép kiểu thành np.uint8 để đảm bảo định dạng đúng cho hình ảnh.
- Tệp đầu ra được tạo bằng cách kết hợp tên tệp và số lượng cụm trong định dạng tên tệp, phần mở rộng và số lượng cụm.
- Cuối cùng, hình ảnh kết quả được lưu xuống tệp đầu ra.

Kết quả minh họa

Figure 1: Hình mẫu

	Random	In_pixels

K = 3		
K = 5		
K = 7		
K = 10		

K = 20		

Nhận xét :

- Hình gốc có kích thước 1024x768 pixels.
- Trong 10 hình với k = 3,5,7,10,20, màu ảnh và nội dung ảnh vẫn giữ được chi tiết hình ảnh.
- Với K = 3, 5, vẫn chưa thấy sự khác biệt giữa việc chọn màu của 2 lựa chọn.
- Với K = 7, 10, ta thấy sự khác biết ở phần bầu trời và chi tiết. Hình 'in_pixels' thể hiện màu tốt hơn của 'random'.
- Với K = 20 thì màu, bóng và các chi tiết ở xa đã được làm rõ, và cả 2 bản đều chi tiết.

Tài liệu tham khảo

- Github kieuconghau
- Github phungvhbui