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 MÔN HỌC

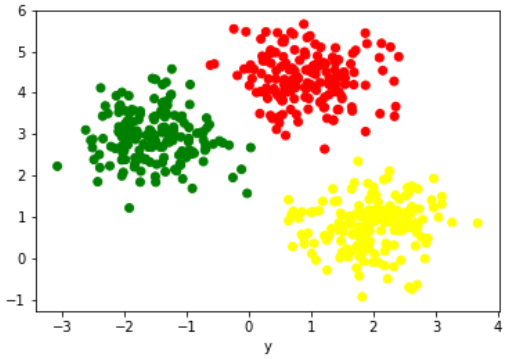
| Đề tài |

IMAGE COMPRESSING USING K-MEANS ALGORITHM

| Giảng viên hướng dẫn |

GV. Phan Thị Phương Uyên

Môn học: Toán ứng dụng và thống kê



Võ Trần Quang Tuấn - 18127248

Thành phố Hồ Chí Minh – 2020

I. Sinh viên thực hiện.

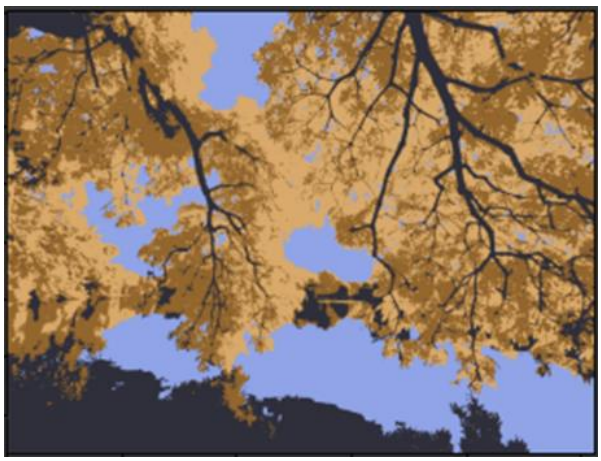
- Họ và tên: Võ Trần Quang Tuấn.
- MSSV: 18127248
- Lớp: 18CLC2.
- Môn học: Toán ứng dụng và thống kê.

II. Bài toán giải quyết.

1. Mô tả.

- Một bức ảnh với kích thước quá lớn và nhiều màu sẽ tăng kích thước lưu trữ trong bộ nhớ máy tính.
- Do đó chúng ta cần giảm số lượng màu của tấm ảnh về một số lượng màu chính, mục đích giảm kích thước lưu trữ nhưng vẫn giữ được nội dung chính của tấm ảnh.

➔ Ảnh gốc.



➔ Ảnh sau khi nén.

2. Hướng giải quyết và ý tưởng.

- Bài toán đưa về việc gom các điểm ảnh tương đồng với nhau theo nhiều nhóm màu chủ đạo, tương đương với viêc phân cụm màu.

- Mỗi điểm ảnh là một vector với 3 phần tử đại diện cho 3 kênh màu với ảnh RBG, 1 kênh màu với ảnh xám,…
- Do đó bài toán đưa về việc gom cụm các vector tương đồng, theo k nhóm nhất định. Theo suy luận này, chúng ta sẽ áp dụng thuật toán gom cụm K-means (K-means clustering) để gom cụm các điểm ảnh có màu gần giống nhau.

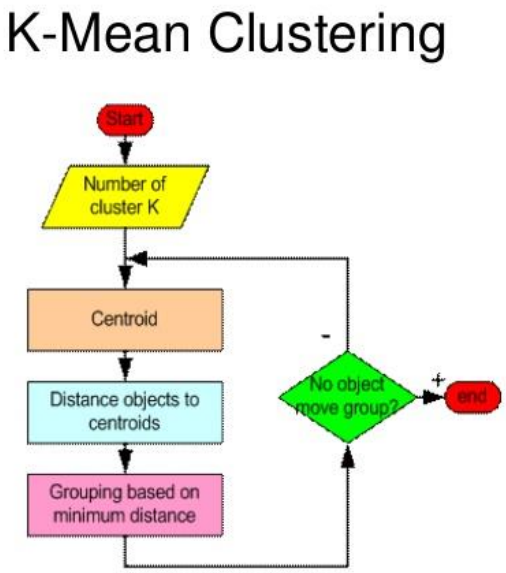


Figure 1. Thuật toán K-mean Clustering (source: internet).

III. Các hàm chính và chức năng.

1. Các thư viện sử dụng.

numpy: dùng cho tính toán trên ma trận.
matplotlib.pyplot: hiển thị ảnh
PIL: đọc ảnh.

2. Các hàm chính.

- normalize_image(image_1d):

  Vì mỗi điểm ảnh có giá trị các kênh màu từ [0, 255], vì vậy khi đưa nguyên các giá trị này vào xử lý thì sẽ tăng thời gian tính toán, làm quá trình chạy thuật toán diễn ra khá lâu. Do đó chúng ta cần đưa các giá trị điểm ảnh có trung bình về 0 và độ lệch chuẩn về 1.

  ➔ Input: ma trận các điểm ảnh được duỗi thẳng.

  ➔ Output: ma trận điểm ảnh được chuẩn hoá, trung bình và độ lệch chuẩn của toàn bộ điểm ảnh.

- kmeans_data_initialize(image_name):

Khởi tạo dữ liệu ban đầu cho quá trình phân cụm điểm ảnh.

➔ Input: tên ảnh

➔ Output: ma trận điểm ảnh ban đầu (height, width, channels) được duỗi thẳng thành (height*width, channels), shape của tấm ảnh.

- cdist(img_1d, centroids, form='euclidcian'):

Tính khoảng cách từng điểm ảnh đến ma trận các điểm ảnh trung tâm centroids

➔ Input: ma trận điểm ảnh duỗi thẳng, ma trận điểm ảnh trung tâm

➔ Output: ma trận khoảng cách từ các điểm ảnh đến ma trận điểm ảnh trung tâm

- assign_color(img_1d, centroids):

Phân nhóm các điểm ảnh trong ma trận ảnh gốc.

➔ Input: ma trận khoảng cách giữa điểm ảnh gốc và điểm ảnh phân nhóm, ma trận điểm ảnh phân nhóm.

➔ Output: ma trận chứa label của mỗi điểm ảnh gốc.

- update_centroids(img_1d, colors, k_clusters):

Cập nhật điểm ảnh trung tâm, bằng cách tính trung bình của các điểm ảnh thuộc nhóm trung tâm đó.

➔ Input: ma trận điểm ảnh ban đầu, nhóm của mỗi điểm ảnh, số nhóm.

➔ Output: ma trận chứa các điểm ảnh đại diện cho nhóm.

- kmeans(img_1d, k_clusters, max_iter, init_centroids='random'):

Phân nhóm dựa vào thuật toán Kmeans. Quét qua toàn bộ các điểm ảnh, phân nhóm, cập nhật nhóm mới. Thực hiện lặp max_iter lần thì dừng.

Nếu init_centroids='random' thì random trung tâm khởi tạo, còn init_centroids='in_pixels' thì random k_clusters trong ma trận điểm ảnh gốc. Nên sử dụng in_pixels vì random khó hội tụ trong môt vài trường hợp.

➔ Input: ma trận điểm ảnh gốc, số nhóm, số vòng lặp tối đa, cách khởi tạo trung tâm.

➔ Output: ma trận ảnh trung tâm, nhãn của các điểm ảnh gốc, ma trận ảnh gốc sau khi được cập nhật màu trung tâm.

- compress_image(image_name, img_1d, k_clusters, max_iter, init_centroids='random'):

Nén ảnh, sử dụng các hàm hỗ trợ phía trên và thuật toán Kmeans. Sau khi nén, chuẩn hoá lại ảnh và show ảnh.

➔ Input: Tên ảnh cần nén.

➔ Output: Tấm ảnh đã được nén.

IV. Kết quả thu được.

1. Kết quả.

- Thực hiện nén ảnh có tên sight_seeing.jpg, với lần lượt 3, 5 7 nhóm màu, số vòng lặp tối đa là 10, init_centroids='in_pixels', ta được kết quả như sau:

Figure 2: Ảnh phong cảnh gốc (source: internet)

Figure 3: Ảnh phong cảnh sau khi nén (k_clusters=7)

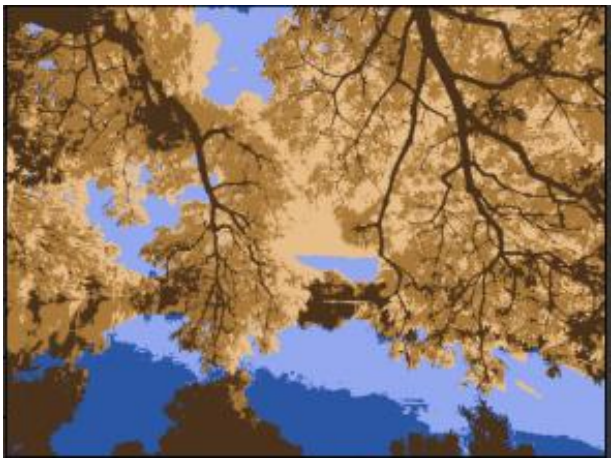


Figure 4: Ảnh phong cảnh sau khi nén (k_clusters=5)

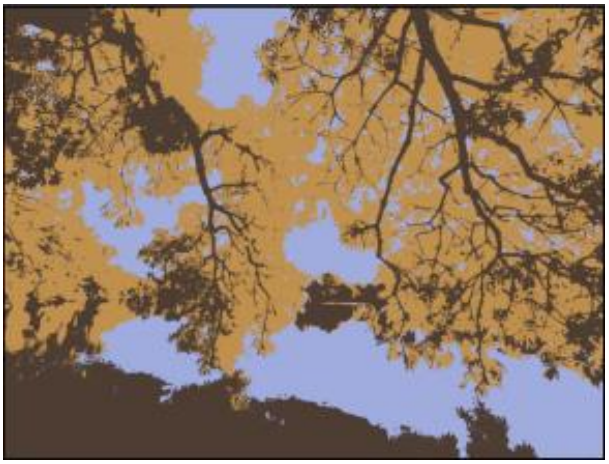


Figure 5: Ảnh phong cảnh sau khi nén (k_clusters=3)

2. Nhận xét.

- K-means chạy khá chậm, vì nó phải quét qua toàn bộ các điểm dữ liệu nhiều lần để cập nhật điểm trung tâm.
- max_iter càng lớn, thuật toán chạy càng lâu, nhưng khả năng phân nhóm điểm ảnh ít sai sót hơn -> ảnh rõ nét hơn.
- k_clusters càng lớn thì số màu đại diện càng nhiều, ảnh rõ nét hơn.

V. Tài liệu tham khảo.

https://en.wikipedia.org/wiki/K-means_clustering
https://machinelearningcoban.com/2017/01/01/kmeans/