

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 01

Môn: Toán ứng dụng và thống kê cho Công nghệ thông tin



| Giảng viên hướng dẫn |

Phan Thị Phương Uyên

Vũ Quốc Hoàng

Nguyễn Văn Quang Huy

Nguyễn Ngọc Toàn

| Sinh viên thực hiện |

Họ và tên	Mã số sinh viên	Lớp
Nguyễn Tấn Lộc	22127236	22CLC03

Mục lục báo cáo

1. Giải thích các hàm:.....	3
2. Thử nghiệm và nhận xét:	6
3. Tài liệu tham khảo:.....	9

1. Giải thích các hàm:

read_image(img_path):

- Tham số: hàm nhận vào img_path, là đường dẫn đến tập tin cần đọc.
- Trả về: hàm trả về img_2d là một mảng Numpy 3 chiều để biểu diễn ảnh ở dạng con số.
- Giải thích: sử dụng thư viện PIL để đọc ảnh và thư viện Numpy để lưu ảnh ở dạng mảng 3 chiều. [1]

show_image(img_2d):

- Tham số: nhận vào mảng 3 chiều Numpy biểu diễn ảnh .
- Trả về: hiển thị hình ảnh đã được lưu.
- Giải thích: sử dụng thư viện matplotlib và hàm imshow để hiển thị ảnh. [2]

save_image(img_2d, img_path):

- Tham số: nhận vào mảng 3 chiều Numpy, và đường dẫn đến nơi cần lưu trữ ảnh.
- Trả về: tập tin ảnh được lưu trữ ở đường dẫn mong muốn.
- Giải thích: sử dụng thư viện PIL để tạo một đối tượng ảnh và lưu nó vào đường dẫn tương ứng. Để thay đổi định dạng của ảnh chỉ cần đổi phần đuôi của ảnh. Ví dụ: img_path = 'picture.pdf' ; img_path = 'picture.png' [3]

convert_img_to_1d(img_2d):

- Tham số: nhận vào mảng Numpy 3 chiều đang lưu ảnh.
- Trả về: mảng 2 chiều Numpy lưu ảnh, với số pixel bằng chiều dài * chiều rộng của ảnh và mỗi pixel chứa 3 thành phần ứng với 3 kênh màu.
- Giải thích: sử dụng hàm reshape để chuyển các pixel đang lưu ở dạng bảng thành một hàng. Hàm reshape nhận vào tham số (chiều dài * chiều rộng, số kênh màu) [4][5]

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

- Tham số:
 - img_1d: mảng 2 chiều Numpy lưu ảnh cần xử lý
 - k_cluster: số màu của ảnh muốn có sau khi hoàn tất thuật toán
 - max_iter: giới hạn số lần lặp trong trường hợp không thể hội tụ
 - init_centroids: quy định phương thức lấy các centroids ban đầu. Bao gồm 2 phương thức là 'random' – lấy ngẫu nhiên giá trị 0 -255 và 'in_pixels' – lấy ngẫu nhiên pixel thuộc ảnh gốc.
- Trả về:
 - centroids: mảng 2 chiều lưu k màu ứng với màu của từng centroids. Có k phần tử, mỗi phần tử gồm 3 thành phần tương ứng với 3 kênh màu Red Blue Green.
 - labels: mảng 1 chiều lưu nhãn của từng pixel của ảnh. Nhãn này cho biết pixel đang thuộc nhóm nào (sẽ mang màu của centroids nào)

- Ý tưởng thuật toán: [6] [7] [8] [9]

Bước 1: Khởi tạo các centroid ban đầu, khởi tạo mảng labels trống.

Bước 2: Tính khoảng cách Euclid từ mỗi pixel đến centroid.

Bước 3: Dựa vào kết quả khoảng cách vừa tính, xác định xem mỗi pixel thuộc nhóm nào và gán nhãn cho nó.

Bước 4: Tạo ra các centroids mới ứng với mỗi nhóm dựa vào cách xác định trung vị của các pixel thuộc nhóm đó. Bước này cần kiểm tra: nếu số lượng pixel nằm trong nhóm bằng 0, nghĩa là không có pixel nào nằm vào trong nhóm ứng với một centroid tương ứng thì ta cần xử lý chọn ngẫu nhiên lại centroid đó.

Bước 5: Lặp lại bước 4 cho đến khi các centroids đã hội tụ (centroids mới tìm ra giống với centroids đã tìm ra trước đó). Trường hợp không hội tụ, dừng lại khi đã lặp đủ max_iter lần.

Bước 6: Trả về centroids và labels.

Các hàm trong thuật toán K-Means:

a) initialize_centroids(img_1d, k_clusters, method):

- Tham số: mảng 2 chiều chứa ảnh, số k cluster, phương thức khởi tạo
- Trả về: các centroids được khởi tạo ban đầu.
- Giải thích: tùy thuộc vào phương thức ở method mà thực hiện khởi tạo. Nếu phương thức là 'random' thì chọn ngẫu nhiên giá trị 0 – 255 cho mỗi kênh màu Red Blue Green. Nếu phương thức là 'in_pixels' thì chọn ngẫu nhiên pixel nằm trong ảnh để làm centroid.

b) calculate_distances(img_1d, centroids):

- Tham số: mảng 2 chiều chứa ảnh, mảng chứa centroids đã được khởi tạo từ đầu
- Trả về: mảng lưu khoảng cách từ mọi pixel đến các centroids.
- Giải thích:
 - Sử dụng hàm newaxis trong thư viện numpy để tăng số chiều của mảng[10]. Điều này để chuẩn bị cho việc thực hiện broadcasting ở bước tiếp theo. Broadcasting cho phép thực hiện các phép toán giữa các phần tử của các mảng có hình dạng khác nhau bằng cách mở rộng mảng nhỏ hơn. [11]
 - Sử dụng hàm linalg.norm trong thư viện numpy để tính toán khoảng cách Euclid dọc theo trục cuối cùng (được xác định bởi axis=2). Điều này tính toán khoảng cách cho từng cặp pixel và tâm cụm. [12]

c) find_labels(distances):

- Tham số: mảng lưu khoảng cách từ mọi pixel đến các centroids.

- Trả về: mảng 1 chiều có cùng số phần tử với số pixel, trong đó mỗi phần tử là chỉ số của centroid gần nhất cho pixel tương ứng.
- Giải thích: Sử dụng hàm `argmin` trong thư viện Numpy. Với mỗi pixel tức hàng trong mảng `distance`, `argmin` sẽ tìm index của centroids có khoảng cách nhỏ nhất với nó. [13] [14]

d) `find_new_centroids(img_1d, labels, k_clusters)`:

- Tham số: nhận vào mảng lưu hình ảnh, mảng lưu các nhãn, và mảng cluster tương ứng với các centroid. Khi này các pixels của ảnh đã được gán nhãn
- Trả về: mảng 2 chiều lưu các centroid mới.
- Giải thích:
 - Thực hiện lấy số kênh màu từ `img_1d` được truyền vào. Khởi tạo mảng `new_centroids` chứa toàn 0 dựa vào số lượng cluster và số kênh màu.
 - Với mỗi cluster (mỗi nhóm): Nếu không có pixel nào thuộc vào cluster đó chứng tỏ xảy ra hiện tượng mất màu, ta thực hiện chọn random lại centroid ứng với cluster đó. Nếu có pixel thuộc cluster đó, thực hiện tìm centroid mới bằng cách lấy trung vị của các pixel trên.

`generate_2d_img(img_2d_shape, centroids, labels)`:

- Tham số: nhận vào shape ứng với định dạng ảnh (ở đây là shape tương tự như của ảnh trước khi xử lý). Centroids chứa các màu centroids đã chọn được từ thuật toán, labels lưu các nhãn của từng pixel của ảnh để cho biết nó thuộc centroid nào.
- Trả về: mảng 3 chiều Numpy lưu ảnh sau khi xử lý
- Giải thích:
 - Đọc và lưu vào chiều rộng, chiều cao và số kênh màu từ `img_2d_shape`
 - Tạo một ma trận 0 với shape ứng với chiều rộng, chiều cao, và số kênh màu. Cho `dtype = np.uint8` (cùng dtype với ảnh đầu vào) để thực hiện việc lưu trữ sau đó.
 - Với mỗi pixel của ảnh, đầu tiên tính ra index ứng với ứng với pixel đó trong danh sách nhãn labels từ đó biết được pixel thuộc nhóm nào. Sau đó chỉnh màu pixel đó thành màu của centroids tương ứng.
 - Sau khi việc gán màu cho tất cả pixel đã xong sẽ trả về mảng Numpy.

2. Thử nghiệm và nhận xét:

Ảnh ban đầu:



Với $k = 3$:



Với $k = 5$:



Với $k = 7$



Với $k = 10$



Với $k = 20$



Bảng dữ liệu:

K	Ban đầu	3	5	7	10	20
Dung lượng ảnh (KB)	212	90.7	119.3	132	192	287
Thời gian thực thi (giây)	-	3	4.3	6.2	16.9	33.8

Tất cả dữ liệu trên đang được tính toán dựa trên ảnh định dạng .png

Nhận xét:

- Sử dụng thuật toán K-Means có thể làm giảm đáng kể số lượng màu của ảnh từ đó làm giảm dung lượng.
- Khi số k càng nhỏ thì thời gian thực thi thuật toán càng ngắn và dung lượng ảnh sẽ càng nhỏ nhưng ảnh sẽ ít chi tiết và ít chính xác hơn.
- Khi k càng lớn thì thời gian thực thi thuật toán càng dài nhưng ảnh sẽ càng giống với ảnh gốc ban đầu.
- Tuy nhiên khi k lớn sẽ có trường hợp dung lượng ảnh sau khi thực hiện thuật toán sẽ lớn hơn ảnh ban đầu. Điều này do sự khác nhau trong việc lưu trữ của các định dạng ảnh khác nhau. [15] Từ dữ liệu trên, với file .png và $k = 20$ thì ảnh lúc sau có dung lượng lớn hơn. Tuy nhiên nếu đổi định dạng ảnh lúc sau sang .jpg cũng với $k = 20$ thì dung lượng ảnh còn 159KB nhỏ hơn ban đầu, cũng với thời gian thực thi tương đương.
- Một vài hạn chế của K-Means rút ra được sau khi cài đặt thuật toán: [16]
 - Không thể xác định được các Centroid ban đầu một cách chính xác mà chỉ có thể chọn ngẫu nhiên. Sau đó thực hiện phương pháp thử và sai.
 - Các centroids sẽ bị phụ thuộc vào giá trị ban đầu của chúng vì những giá trị khởi tạo khác nhau có thể dẫn đến những các phân cụm khác nhau dù cho có cùng số cụm.
 - Cần phải tính toán rất nhiều các khoảng cách từ mỗi pixel đến vị trí centroids. Nếu ảnh có kích thước lớn có thể vượt quá khả năng lưu trữ của RAM.
 - Đối với trường hợp các màu trong ảnh phân bố không đồng đều hoặc bị lệch thì thuật toán khó có thể hội tụ.

3. Tài liệu tham khảo:

[1] Importing Image Data into NumPy Arrays. PularalSight.

Truy cập: 15/6/2024

<https://www.pluralsight.com/resources/blog/guides/importing-image-data-into-numpy-arrays>

[2] How to Display an OpenCV image in Python with Matplotlib? – GeeksForGeeks.

Truy cập: 15/6/2024

<https://www.geeksforgeeks.org/how-to-display-an-opencv-image-in-python-with-matplotlib/>

[3] Python PIL | Image.save() method – GeeksForGeeks

Truy cập: 15/6/2024

<https://www.geeksforgeeks.org/python-pil-image-save-method/>

[4] Convert a NumPy array to an image. – GeeksForGeeks

Truy cập: 15/6/2024

<https://www.geeksforgeeks.org/convert-a-numpy-array-to-an-image/>

[5] NumPy Array Reshaping – W3School

Truy cập: 15/6/2024

https://www.w3schools.com/python/numpy/numpy_array_reshape.asp

[6] Cô Phan Thị Phương Uyên - Bài giảng Project 1 - 13/6/2024.

[7] K-Means Clustering in Python: A Practical Guide. - Real Python.

Truy cập: 16/6/2024

<https://realpython.com/k-means-clustering-python/>

[8] K means Clustering – Introduction. – GeeksForGeeks

Truy cập 16/6/2024

<https://www.geeksforgeeks.org/k-means-clustering-introduction/>

[9] K Means Clustering in Python. - Analytics Vidha.

Truy cập: 17/6/2024

<https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/>

[10] Mseifert. Answered Jan 13, 2018 at 17:48 - Stackoverflow

Truy cập: 16/7/2024

<https://stackoverflow.com/questions/29241056/how-do-i-use-np-newaxis>

[11] NumPy Array Broadcasting. GeeksForGeeks

Truy cập: 16/7/2024

<https://www.geeksforgeeks.org/numpy-array-broadcasting/>

[12] Calculate the Euclidean distance using NumPy. GeeksForGeeks.

Truy cập: 16/7/2024

<https://www.geeksforgeeks.org/calculate-the-euclidean-distance-using-numpy/>

[13] numpy.argmax() in Python – GeeksForGeeks

Truy cập: 16/7/2024

<https://www.geeksforgeeks.org/numpy-argmin-python/>

[14] NumPy argmin() – Programiz

Truy cập: 16/7/2024

<https://www.programiz.com/python-programming/numpy/methods/argmin>

[15] Christian Konig - Nov 19, 2018 at 8:37 – stackoverflow

Truy cập: 17/6/2024

<https://stackoverflow.com/questions/53369971/size-of-jpg-and-jpeg-increased-after-compression>

[16] Pham Dinh Khanh – DeepAIKhanhBlog

Truy cập: 17/6/2024

https://phamdinhkhanh.github.io/deepai-book/ch_ml/KMeans.html#han-che-cua-k-means