

Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

Khoa Công nghệ Thông tin

---o0o---



Đồ Án 1 – Color Compression

Môn: Toán Ứng Dụng Và Thống Kê

Giảng viên lý thuyết: - Vũ Quốc Hoàng

Giảng viên thực hành:

- Phan Thị Phương Uyên
- Nguyễn Văn Quang Huy
- Trần Thị Thảo Nhi

Sinh Viên:

- Nguyễn Trung Hiếu - 19127403

Table of Contents

<i>I.</i>	<i>Tổng quan và ý tưởng.</i>	4
<i>II.</i>	<i>Giải thích các hàm được sử dụng trong bài.</i>	5
1.	Convert_image_to_1D(url)	5
2.	Euclidean_Distance(x,y).....	5
3.	Init_centroid(img_1d, k_clusters, init_centroids)	5
4.	Distance_calc(img_1d, centroids).....	5
5.	Assign_centroid(matrix_distance).....	6
6.	Update_centroids(img_1d, labels, k_clusters).....	6
7.	Has_converged(centroids, new_centroids, init_centroids)	6
8.	Kmeans_func(img_1d, k_clusters, max_iter, init_centroids)	6
9.	Download_img(url, image_name)	7
<i>III.</i>	<i>Hướng dẫn sử dụng.....</i>	7
<i>IV.</i>	<i>Đánh giá.</i>	8
i.	Đối với set init_centroids là ‘in_pixels’	8
ii.	Đối với set init_centroids là ‘random’	12
<i>III.</i>	<i>Tài liệu tham khảo.</i>	17

I. Tổng quan và ý tưởng.

Ứng dụng thuật toán phân lớp K-means để giải quyết vấn đề giảm tải màu của một bức ảnh.

K-means là một thuật toán phân cụm cơ bản trong unsupervised learning (học không có giám sát). Được sử dụng trong trường hợp chúng ta không biết nhãn (label) của các điểm dữ liệu. Tương tự với bài nén màu sắc, chúng ta không có nhãn mà cần gom cụm các màu sắc lại của bức ảnh, ở vai trò này số lượng màu sắc chính là k-centroids.

Bức ảnh được sử dụng trong trường hợp này là:



Sau khi đã được giảm tải xuống còn 5 màu:



Với ý tưởng này hình thành nên các hướng đi sẽ ứng dụng thuật toán vào bài nén màu sắc:

Đầu tiên, ta cần một bức hình để đọc lên jupyter notebook và chuyển thành mảng 1 chiều với `shape=(height * width, numchannel)`

Sau đó, ta cần khởi tạo ngẫu nhiên các centroids, số lượng centroids tùy thuộc vào người dùng nhập và người dùng muốn dùng ngẫu nhiên các màu sắc có sẵn trong hình hoặc những màu sắc hoàn toàn ngẫu nhiên. Tiếp theo ta cần tính ma trận khoảng cách `matrix[i][j]` với ý nghĩa khoảng cách màu sắc dòng i tương ứng với dòng j centroids

Tiếp theo, ta cần gán nhãn các điểm dữ liệu còn lại dựa trên ma trận khoảng cách của từng điểm với từng centroids. Sau đó ta cập nhật lại các centroids bằng cách tính trung bình khoảng cách các điểm thuộc về mỗi centroids tạo ra một list `new_centroids` mới.

Thuật toán dừng lại khi vòng lặp đã đạt đến `max_iter` (được người dùng nhập vào) hoặc nếu như các điểm `new_centroids` giống với `centroids` cũ thì ta dừng lại.

II. Giải thích các hàm được sử dụng trong bài.

1. Convert_image_to_1D(url)

`url`: vị trí tương đối hoặc vị trí tuyệt đối của bức ảnh

Hàm này được sử dụng để chuyển đổi hình ảnh thành mảng 3 chiều với `shape(height, width, numchannel)` được reshape lại còn mảng 1 chiều với `shape(height * width, numchannel)`.

Return: trả về mảng 1 chiều.

2. Euclidean_Distance(x,y)

`x,y`: là một vector.

Hàm này được sử dụng để tính khoảng cách giữa 2 điểm hoặc 2 vector.

Return: trả về khoảng cách với giá trị là số thực.

3. Init_centroid(img_1d, k_clusters, init_centroids)

`img_1d`: là mảng vector hình ảnh 1 chiều.

`k_clusters`: số lượng clusters được sử dụng.

`init_centroids`: chọn ngẫu nhiên centroids dựa theo ‘random’ hoặc ‘in_pixels’.

+ `random`: được chọn ngẫu nhiên từ [0,255], số lần chọn tương ứng với kênh màu của bức hình.

+ `in_pixels`: được chọn ngẫu nhiên các màu sắc có sẵn trong bức ảnh.

Hàm được sử dụng để khởi tạo các centroids ban đầu.

Trả về số lượng centroids tương ứng với `k_clusters` được nhập.

4. Distance_calc(img_1d, centroids)

`img_1d`: mảng 1 chiều của hình ảnh.

`centroids`: các centroids (`k_clusters`).

Hàm được sử dụng để tính khoảng cách dựa trên các điểm dữ liệu của hình ảnh với từng điểm trong centroids. Với $\text{matrix}[i][j]$ là khoảng cách của dòng i với centroids j tương ứng.

Return: ma trận khoảng cách.

5. Assign_centroid(matrix_distance)

Matrix_distance: ma trận khoảng cách.

Hàm được sử dụng để gán nhãn (label) cho các điểm dữ liệu với centroids gần nó nhất (khoảng cách centroids gần nhất).

Return: trả về mảng 1 chiều tương ứng với các dòng với các giá trị là label của điểm.

6. Update_centroids(img_1d, labels, k_clusters)

Img_1d: mảng 1 chiều của hình ảnh.

Labels: các nhãn của các điểm dữ liệu

K_clusters: số lượng centroids được khởi tạo ban đầu

Hàm sử dụng để cập nhập lại các centroids mới dựa trên nhãn của các điểm dữ liệu và được tính trung bình.

Return: các điểm centroids mới

7. Has_converged(centroids, new_centroids, init_centroids)

Centroids: các điểm centroids ban đầu.

New_centroids: các điểm centroids sau khi update.

Init_centroids: cách khởi tạo centroids ngẫu nhiên ban đầu.

Hàm sử dụng để kiểm tra hội tụ của các centroids.

Returns: True nếu đã hội tụ, False nếu chưa hội tụ.

8. Kmeans_func(img_1d, k_clusters, max_iter, init_centroids)

Img_1d: mảng 1 chiều của hình ảnh.

K_clusters: số lượng centroids (màu sắc).

Max_iter: số vòng lặp tối đa.

Init_centroids: Cách tạo ngẫu nhiên centroids ban đầu. ‘random’ hoặc ‘in_pixels’.

Hàm ứng dụng k_means vào bài toán giảm màu.

Returns: centroids cuối cùng với labels tương ứng.

9. Download_img(url, image_name)

url: url của bức hình.

image_save: tên hình khi save lại.

Hàm tải hình ảnh dựa trên url vào thư viện jupyter notebook đang hoạt động.

III. Hướng dẫn sử dụng.

Nếu bạn đầu chưa có hình ảnh ta cần chạy hàm `download_img` để tải hình về.

url ảnh được sử dụng trong bài:

https://upload.wikimedia.org/wikipedia/commons/thumb/6/67/Inside_the_Batad_rice_terraces.jpg/400px-Inside_the_Batad_rice_terraces.jpg

Do tính chất là file .ipynb nên ta cần chạy các block trên trước khi chạy block chứa hàm **main** chính của bài.

Các yêu cầu cần nhập để có thể sử dụng thuật toán

Nhập địa chỉ tương đối hoặc tuyệt đối của bức ảnh

Các ảnh có trong folder

```
-rw-r--r-- 1 trunghieu staff 33136 Jun 15 16:58 Reduce_colored.  
png  
-rw-r--r-- 1 trunghieu staff 50264 Jun 14 13:41 terrace.png
```

Input image location:

terrace.png

Nhập cách random centroids: 0 cho random, 1 cho in_pixels.

Init centroids -> 0 for random, 1 for in_pixels:

1

Nhập số lượng màu.

Input number of k-clusters (color):

5

Nhập số lần lặp tối đa.

Input max iterations:

5

Hoàn chỉnh.

```
Các ảnh có trong folder
-rw-r--r-- 1 trunghieu staff  33136 Jun 15 16:58 Reduce_colored.png
-rw-r--r-- 1 trunghieu staff  50264 Jun 14 13:41 terrace.png
-----
Input image location:terrace.png
Init centroids -> 0 for random, 1 for in_pixels:1
Input number of k-clusters (color):5
Input max iterations:5
----- Bắt đầu tiến trình chạy thuật toán -----
Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4
Iteration: 5
Mất 16.811s để giảm màu
? .
```

IV. Đánh giá.

i. Đối với set init_centroids là ‘in_pixels’

Chạy thuật toán giảm còn 5 màu với 5 lần lặp.

Các ảnh có trong folder

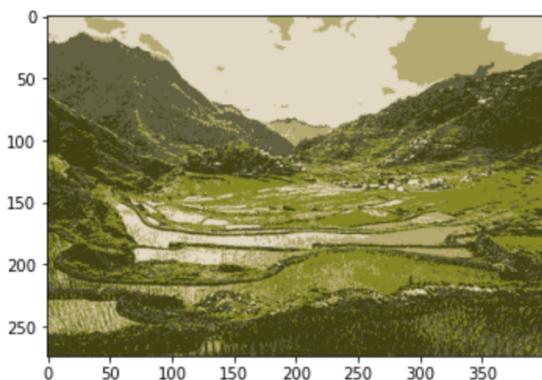
```
-rw-r--r-- 1 trunghieu staff 33136 Jun 15 16:58 Reduce_colored.png  
-rw-r--r-- 1 trunghieu staff 50264 Jun 14 13:41 terrace.png
```

```
-----  
Input image location:terrace.png  
Init centroids -> 0 for random, 1 for in_pixels:1  
Input number of k-clusters (color):5  
Input max iterations:5  
----- Bắt đầu tiến trình chạy thuật toán -----  
Iteration: 1  
Iteration: 2  
Iteration: 3  
Iteration: 4  
Iteration: 5  
Mất 16.811s để giảm màu  
Ảnh gốc
```

Ảnh gốc



Ảnh phục chế



mất 16s để giảm còn lại 5 màu với 5 lần lặp. Ta có thể nhận thấy màu sắc bị giảm một cách đáng kể nhưng bù lại khiến cho ảnh có dung lượng nhẹ hơn.

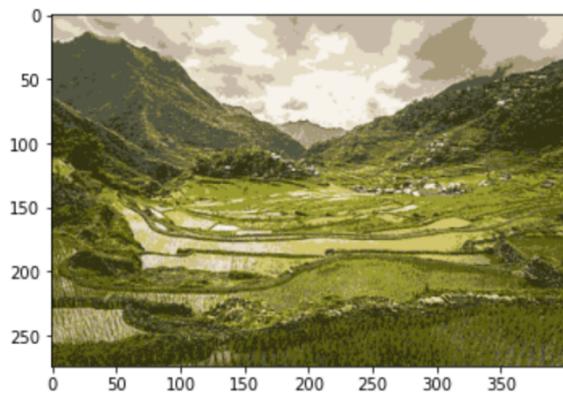
Chạy thuật toán giảm còn 10 màu với 5 lần lặp.

```
Các ảnh có trong folder
-rw-r--r-- 1 trunghieu staff 32599 Jun 15 19:34 Reduce_colored.
png
-rw-r--r-- 1 trunghieu staff 50264 Jun 14 13:41 terrace.png
-----
Input image location:terrace.png
Init centroids -> 0 for random, 1 for in_pixels:1
Input number of k-clusters (color):10
Input max iterations:5
----- Bắt đầu tiến trình chạy thuật toán -----
Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4
Iteration: 5
Mất 31.691s để giảm màu
```

Ảnh gốc



Ảnh phục chế



Với lần chạy này thì 10 màu với 5 lần lặp thì mất 31.6s để chạy. Hình vẫn còn giữ được bức tranh tổng thể phần nào đó so với ảnh gốc.

Chạy thuật toán giảm còn 100 màu với 2 lần lặp.

```
Các ảnh có trong folder
-rw-r--r-- 1 trunghieu staff 49801 Jun 15 19:39 Reduce_colored.
png
-rw-r--r-- 1 trunghieu staff 50264 Jun 14 13:41 terrace.png
```

```
Input image location:terrace.png
Init centroids -> 0 for random, 1 for in_pixels:1
Input number of k-clusters (color):100
Input max iterations:2
----- Bắt đầu tiến trình chạy thuật toán -----
Iteration: 1
Iteration: 2
Mất 151.527s để giảm màu
```



Ảnh phục chế



Với lần chạy cuối cùng này ảnh được giảm xuống 100 màu với 2 lần lặp. Mất 151s để hoàn thành, bức ảnh mặc dù được giảm màu nhưng vẫn đẹp hơn và gần ngang với màu gốc.

ii. Đối với set init_centroids là ‘random’

Khi chạy với centroids được set là random thì sẽ có 1 vài centroids sẽ không có labels nào bởi vì khoảng cách của nó lớn đối với tất cả các điểm trong hình ảnh. Vì vậy đối với số lượng centroids được cài đặt ban đầu có thể số lượng màu sắc sẽ không tương đương hay còn có thể nói là colors $\leq k_clusters$

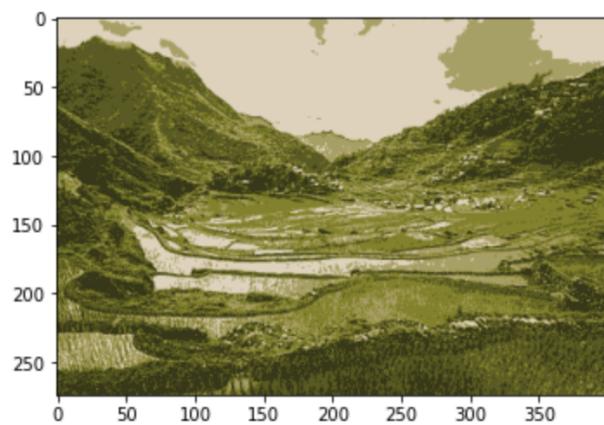
Chạy thuật toán giảm còn 5 màu với 5 lần lặp.

```
Các ảnh có trong folder  
-rw-r--r--@ 1 trunghieu staff 76296 Jun 15 19:51 Reduce_colored.png  
-rw-r--r--@ 1 trunghieu staff 50264 Jun 14 13:41 terrace.png
```

```
-----  
Input image location:terrace.png  
Init centroids -> 0 for random, 1 for in_pixels:0  
Input number of k-clusters (color):5  
Input max iterations:5  
----- Bắt đầu tiến trình chạy thuật toán -----  
Iteration: 1  
Iteration: 2  
Iteration: 3  
Iteration: 4  
Iteration: 5  
Mất 16.876s để giảm màu  
Số lượng màu sắc: 5  
Ảnh gốc
```



Ảnh gốc



Do tính chất chọn centroids random vì vậy khiến cho bức ảnh này màu sắc không đẹp bằng so với khởi tạo centroids in_pixels mặc dù có cùng 5 màu. Tuy nhiên ta vẫn có thể hình dung được bức ảnh này mô tả những gì.

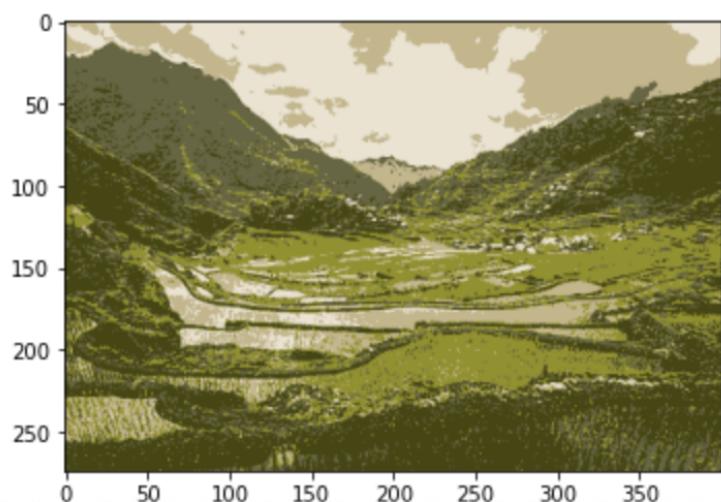
chạy lại lần 2

```
Các ảnh có trong folder
-rw-r--r--@ 1 trunghieu  staff    34861 Jun 15 20:02 Reduce_colored.png
-rw-r--r--@ 1 trunghieu  staff    50264 Jun 14 13:41 terrace.png
-----
Input image location:terrace.png
Init centroids -> 0 for random, 1 for in_pixels:0
Input number of k-clusters (color):5
Input max iterations:5
----- Bắt đầu tiến trình chạy thuật toán -----
Iteration: 1
Iteration: 2
Iteration: 3
Iteration: 4
Iteration: 5
Mất 16.447s để giảm màu
Số lượng màu sắc: 5
```

Ảnh gốc



Ảnh phục chế



Vì do chạy random màu sắc được chọn ngẫu nhiên, nên khi mỗi lần chạy sẽ cho ra các kết quả khác nhau. Với lần chạy lại này thì khiến cho kết quả khác với lần chạy trước. Có thể nói là đẹp hơn và bầu trời trông có vẻ là bầu trời.

Chạy thuật toán giảm còn 20 màu với 3 lần lặp.

Các ảnh có trong folder

```
-rw-r--r--@ 1 trunghieu staff      52375 Jun 18 10:25 Reduce_colored.png  
-rw-r--r--  1 trunghieu staff      50264 Jun 18 10:24 terrace.png
```

Input image location:terrace.png

Init centroids -> 0 for random, 1 for in_pixels:0

Input number of k-clusters (color):20

Input max iterations:1

----- Bắt đầu tiến trình chạy thuật toán -----

Iteration: 1

Mất 20.305s để giảm màu

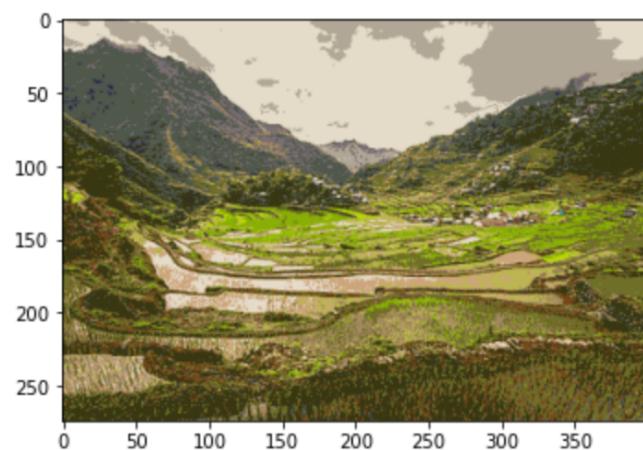
Số lượng màu sắc thật sự: 15

Số lượng màu sắc tinh lọc: 15

Ảnh gốc



Ảnh gốc



Với lần chạy random ở 20 màu thì màu sắc màu được cải thiện hơn so với 5 hoặc 10 màu của centroids ‘in_pixels’.

III. Tài liệu tham khảo.

<https://machinelearningcoban.com/2017/01/01/kmeans/>

https://en.wikipedia.org/wiki/K-means_clustering

<https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>