



BÁO CÁO PROJECT 1

ĐỒ ÁN CÀI ĐẶT CHƯƠNG TRÌNH GIẢM SỐ LƯỢNG MÀU CHO ẢNH

Sinh viên thực hiện

Họ và tên: Lê Mỹ Khánh Quỳnh
Mssv: 21127681
Lớp: 21CLC01

HỌC PHẦN: TOÁN ỨNG DỤNG VÀ THỐNG KÊ
CHO CÔNG NGHỆ THÔNG TIN

MỤC LỤC

A. MÔ TẢ Ý TƯỞNG	2
B. MÔ TẢ HÀM.....	3
1. Xử lý dữ liệu input	3
2. K-Means Clustering algorithm	3
3. Xử lý dữ liệu output	4
4. Hàm main	5
5. Nearest Neighbor Interpolation	5
C. TESTCASES VÀ KẾT QUẢ	5
1. Case 1: 3 clusters	6
2. Case 2: 5 clusters	7
3. Case 3: 7 clusters	8
4. Resize.....	9
D. NHẬN XÉT	10
E. NGUỒN THAM KHẢO	12

A. MÔ TẢ Ý TƯỞNG

❖ Thuật toán K-means Clustering để xử lý vấn đề giảm số lượng màu ảnh

- K-means Clustering là một thuật toán phân cụm dữ liệu, bằng việc tạo và cập nhật các điểm trung tâm để phân nhóm các điểm dữ liệu cho trước và các nhóm khác nhau. Dựa trên nguyên lý hoạt động đó của thuật toán, để xử lý vấn đề giảm số lượng màu ảnh, em sử dụng việc tạo random các điểm trung tâm (centroid) đại diện cho từng nhóm màu, đồng thời các centroid liên tục được cập nhật bằng cách dựa vào trung bình của các điểm dữ liệu nằm trong phân cụm. Sau đó, thay thế mỗi điểm dữ liệu màu bằng màu sắc điểm trung tâm gần nhất dựa trên khoảng cách Euclidean để thu được hình ảnh giảm còn số lượng màu chỉ định.

• Step by steps

1. Khởi tạo các giá trị điểm trung tâm ngẫu nhiên.
2. Tính toán khoảng cách của từng điểm ảnh đến từng điểm trung tâm.
3. Dán nhãn điểm trung tâm có khoảng cách gần nhất cho điểm ảnh.
4. Cập nhật các điểm trung tâm mới bằng cách lấy trung bình các điểm ảnh nằm trong nhóm dữ liệu của điểm trung tâm trước đó.
5. Tiếp tục lặp lại quá trình 2,3,4 đến khi các điểm trung tâm không thay đổi, hoặc lặp lại với số lần tối đa được định sẵn.

❖ Thuật toán Nearest Neighbor Interpolation để thay đổi kích thước ảnh

- Dựa trên nguyên lý hoạt động của thuật toán, để có thể tăng giảm kích thước ảnh, ta tạo ra một mảng chứa dữ liệu mới có kích thước bằng với kích thước người dùng muốn thay đổi. Đồng thời, tính tỉ lệ thay đổi của chiều cao, chiều rộng của hình ảnh cũ so với hình ảnh mới. Đối với từng pixel ở mảng dữ liệu mới, ta lấy tọa độ pixel nhân với tỉ lệ chênh lệch từ đó tìm ra giá trị gần nhất với tọa độ tương ứng ở mảng dữ liệu cũ. Gán giá trị của điểm ảnh vừa tìm được từ hình ảnh cũ qua cho điểm ảnh tương ứng trong hình ảnh mới.

• Step by steps

1. Tạo numpy array với kích thước mới

2. Tính tỉ lệ của kích thước hình ảnh cũ so với kích thước mới
3. Với mỗi điểm ảnh trong hình ảnh mới, nhân tọa độ với tỉ lệ để tìm tọa độ điểm ảnh tương ứng trong hình ảnh cũ
4. Gán giá trị của điểm ảnh cũ vừa tìm được cho điểm ảnh mới tương ứng, lặp lại bước 3 cho đến khi tất cả điểm ảnh của hình ảnh mới đều có dữ liệu

B. MÔ TẢ HÀM

1. Xử lí dữ liệu input

*a. Function: **get_image***

- **Tham số đầu vào:** đường dẫn đến tập tin input.
- **Prototype:** get_image(path)
- **Mục đích:** Trả về dữ liệu ảnh dưới dạng mảng.
- **Mô tả:** Mở file và đọc dữ liệu từ ảnh, chuyển hình ảnh về dạng numpy array bằng hàm của thư viện numpy và trả về. Nếu quá trình mở ảnh xảy ra lỗi, hàm sẽ in ra thông báo và trả về None.

2. K-Means Clustering algorithm

*a. Function: **init_centroids***

- **Tham số đầu vào:** data array của image, số lượng cluster, loại khởi tạo.
- **Prototype:** init_centroids(img,k_cluster,type)
- **Mục đích:** Khởi tạo các điểm trung tâm.
- **Mô tả:** Ta có 2 cách khởi tạo giá trị. Nếu loại khởi tạo = ‘random’, hàm sẽ trả về k điểm trung tâm, với các kênh màu của mỗi điểm là giá trị random trong đoạn $[0,255]$. Trong trường hợp loại khởi tạo = ‘in_pixels’, các cluster sẽ được random từ những pixels trong ảnh.

*b. Function: **update_centroids***

- **Tham số đầu vào:** data array của image, labels, centroids cũ, số lượng cluster.
- **Prototype:** update_centroids(img,labels,old_centroids,k_cluster)
- **Mục đích:** Cập nhật các điểm trung tâm mới.
- **Mô tả:** Các điểm trung tâm mới được khởi tạo bằng cách lấy trung bình các điểm nằm trong phân cụm dữ liệu của trung tâm cũ, điều

kiện để cập nhật là phân cụm của centroid cũ phải có ít nhất 1 pixel để tránh trường hợp NaN value.

c. Function: *kmeans*

- **Tham số đầu vào:** data array của image, số lượng cluster, số lần thực thi vòng lặp, loại khởi tạo centroids.
- **Prototype:** `kmeans(img,k_cluster,max_iter,init_centroids)`
- **Mục đích:** Thực thi thuật toán K-Means Clustering và trả về mảng centroids, labels.
- **Mô tả:** Sử dụng lời gọi hàm khởi tạo các điểm trung tâm (`init_centroids`) và gán cho biến *centroids*. Hàm sẽ thực hiện chạy vòng lặp, trong mỗi lần chạy, ta sẽ thực hiện gán label cho từng điểm ảnh, với nhãn dán là index của điểm trung tâm gần với điểm ảnh nhất dựa trên khoảng cách Euclidean. Đồng thời cập nhật các điểm trung tâm bằng lời gọi hàm *update_centroids* để tìm ra các điểm trung tâm mới. Vòng lặp sẽ dừng lại nếu sau khi cập nhật các điểm ảnh trung tâm vẫn giữ nguyên, hoặc chạy hết số lần thực thi tối đa.

3. Xử lý dữ liệu output

a. Function: *restore_image*

- **Tham số đầu vào:** hình dạng hình ảnh gốc, centroids, labels.
- **Prototype:** `restore_image(shape,centroids,labels)`
- **Mục đích:** Khôi phục lại hình ảnh từ các điểm trung tâm, labels đã thực hiện phân cụm.
- **Mô tả:** Sử dụng labels để lấy các điểm trung tâm tương ứng từ mảng centroids, chuyển dữ liệu của giá trị các điểm trung tâm về dạng số nguyên bằng ép kiểu. Cuối cùng, reshape dựa theo hình dáng ảnh gốc để khôi phục hình dạng hình ảnh.

b. Function: *ouput_file*

- **Tham số đầu vào:** data array của image, tên file, phần mở rộng file, đường dẫn file.
- **Prototype:** `ouput_file(img,filename,ext,path)`
- **Mục đích:** Lưu file hình ảnh mới với định dạng và đường dẫn theo yêu cầu người dùng.

- **Mô tả:** Chuyển đổi mảng numpy chứa dữ liệu của hình ảnh thành một đối tượng hình ảnh PIL, để chuyển đổi mảng numpy cần được đổi thành kiểu dữ liệu 'uint8' để phù hợp với định dạng hình ảnh. Sử dụng phương thức `save()` của đối tượng hình ảnh để lưu ảnh vào tập tin với định dạng và đường dẫn được chỉ định. Nếu có lỗi xảy ra trong quá trình lưu, hàm sẽ in ra thông báo lỗi và trả về.

4. Hàm main

- **Mô tả:** Yêu cầu người dùng nhập đường dẫn link tập tin input, đồng thời gọi thực thi hàm `get_image` để lấy data array của ảnh. Từ mảng numpy 3D nhận được từ hàm `get_image`, thực hiện reshape hình ảnh để lấy mảng 2D của ảnh với hình dạng (chiều cao * chiều rộng, số kênh màu). Sử dụng mảng 2D này để thực thi thuật toán. Tiếp tục, cho người dùng nhập các thông tin về tên output file, đường dẫn để lưu và phần mở rộng của file (tùy chọn định dạng png/pdf). Sử dụng lời gọi hàm `kmeans` để thực thi phân cụm dữ liệu màu ảnh và gán dữ liệu trả về cho biến `centroids, labels`. Khôi phục hình ảnh đã được giảm màu bằng hàm `restore_image`. Cuối cùng, gọi hàm `ouput_file` để thực hiện lưu hình ảnh.

5. Nearest Neighbor Interpolation

a. Funtion: *nearest_neighbor_resize*

- **Tham số đầu vào:** data array của hình ảnh cần resize, chiều cao mới, chiều rộng mới.
- **Prototype:** `nearest_neighbor_resize(img,new_height,new_width)`
- **Mục đích:** Thay đổi kích thước hình ảnh
- **Mô tả:** Tạo numpy array có kích thước mới với full giá trị -1. Tính tỉ lệ của chiều rộng, chiều cao cũ so với kích thước mới gán cho biến `w_ratio, h_ratio`. Chạy vòng lặp đến từng pixel trong data array mới, nhân tọa độ với tỉ lệ để tìm tọa độ pixel tương ứng trong data array cũ. Gán giá trị của pixel cũ tương ứng qua pixel ở data array mới. Trả về mảng dữ liệu 3 chiều của hình ảnh có kích thước mới.

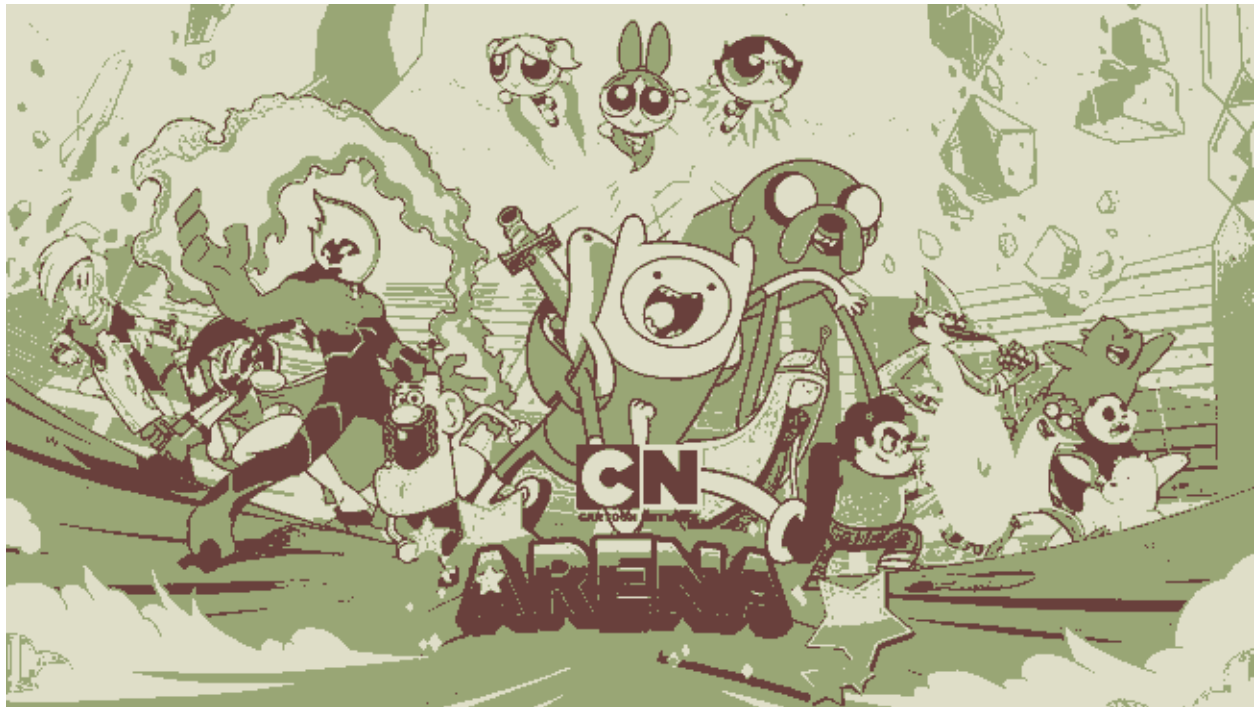
C. TESTCASES VÀ KẾT QUẢ

❖ Ảnh gốc

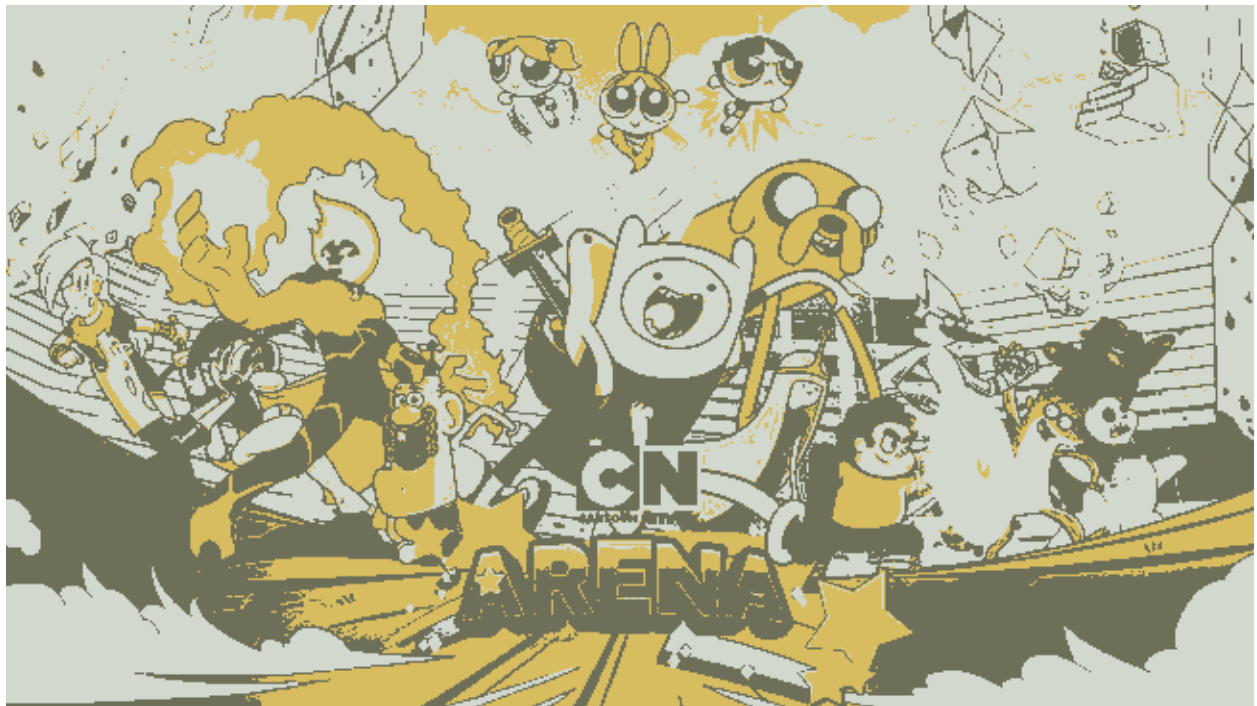


Picture 1 kích thước (366x650)

1. Case 1: 3 clusters



Picture 2 được khởi tạo centroids random



Picture 3 được khởi tạo centroids in pixels

2. Case 2: 5 clusters



Picture 4 được khởi tạo centroids random



Picture 5 được khởi tạo centroids in pixels

3. Case 3: 7 clusters



Picture 6 được khởi tạo centroids random



Picture 7 được khởi tạo centroids in pixels

4. Resize



Picture 8 được thu nhỏ với kích thước (250x550)



Picture 9 được phóng to với kích thước (450x700)

D. NHẬN XÉT

	Số lượng phân cụm lớn	Số lượng phân cụm nhỏ
Ưu điểm	<p>Có khả năng tạo ra phân cụm chi tiết hơn</p> <p>Các cụm màu sắc được phân chia rõ ràng và có tính tách biệt cao</p> <p>Hình ảnh đặc sắc và dễ dàng phản ánh thông tin ảnh gốc</p>	<p>Đối với số lượng phân cụm nhỏ vừa phải, vừa có thể đảm bảo ý nghĩa của vấn đề giảm màu ảnh vừa có thể phản ánh được thông tin ảnh</p> <p>Tốn ít dung lượng lưu trữ cũng như thời gian thực thi thuật toán</p>
Nhược điểm	<p>Nếu có quá nhiều phân cụm, có thể dẫn đến sự quá tách biệt về màu và làm mất ý nghĩa trong việc xử lý giảm màu ảnh</p>	<p>Nếu số lượng phân cụm quá nhỏ, có thể dẫn đến thiếu thông tin và biểu diễn không chính xác của ảnh</p> <p>Hình ảnh cho ra mờ nhạt, thiếu sự phong phú về màu sắc</p>
	Khởi tạo ngẫu nhiên các điểm trung tâm	Khởi tạo các điểm trung tâm dựa trên điểm ảnh của ảnh gốc
Ưu điểm	<ul style="list-style-type: none"> - Có độ đa dạng và tính ngẫu nhiên cao - Giúp hình ảnh có độ phong phú về màu sắc - Đơn giản và nhanh chóng để thực thi 	<ul style="list-style-type: none"> - Đảm bảo phân cụm màu sắc phản ánh đúng phân phối ảnh gốc - Giúp giữ được tính đặc trưng và thông tin ảnh - Màu sắc biểu diễn chính xác hơn
Nhược điểm	<ul style="list-style-type: none"> - Khả năng kết quả cho ra không phản ánh được chính xác phân phối màu của ảnh gốc - Có thể gây mất mát thông tin hoặc phân cụm không tốt do tính ngẫu nhiên 	<ul style="list-style-type: none"> - Giảm tính ngẫu nhiên và đa dạng của màu sắc - Có thể bị ảnh hưởng bởi nhiễu và các điểm ngoại lệ

❖ Nhận xét kết quả resize:

- Kết quả ảnh resize dựa theo thuật toán Nearest Neighbor Interpolation có thể đảm bảo được việc resize diễn ra đơn giản và nhanh chóng, giữ nguyên được sắc thái hình ảnh ban đầu, nhưng bên cạnh vẫn có nhiều hạn chế. Kết quả hình ảnh phóng to có thể phản ánh đầy đủ thông tin và sắc thái hình ảnh. Tuy nhiên do thuật toán chỉ chọn những giá trị gần nhất, không có độ chính xác cao và không có sự chọn lọc độ quan trọng của pixel cần loại bỏ khi thu nhỏ ảnh, dẫn đến trường hợp kết quả thu nhỏ có phần rời rạc và không được mượt mà.

E. NGUỒN THAM KHẢO

- Tài liệu giải thích thuật toán K-means Clustering: [link1](#) , [link2](#)
- Tài liệu cài đặt chương trình K-means Clustering: [link](#)
- Tài liệu về thuật toán Nearest Neighbor Interpolation: [link](#)
- [ChatGPT](#)