**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**LỚP: 21CLC07**

**Ảnh có chứa Phông chữ, biểu tượng, Đồ họa, Xanh điện

Mô tả được tạo tự động**

**BÁO CÁO PROJECT 1:**

**COLOR COMPRESSION**

Thông tin sinh viên thực hiện:

* Họ và Tên: Nguyễn Tấn Khiêm
* MSSV: 21127726

Giảng viên hướng dẫn:

• Giảng viên: Nguyễn Văn Quang Huy  
• Giảng viên: Ngô Đình Hy  
• Giảng viên: Trần Hà Sơn  
• Giảng viên: Nguyễn Đình Thúc

Nội dung

[**1.** **Ý tưởng thực hiện** 2](#_Toc140789237)

[**2.** **Miêu tả thuật toán:** 2](#_Toc140789238)

[**3.** **Test case:** 5](#_Toc140789239)

[**4.** **Kết luận:** 7](#_Toc140789240)

[**5.** **Tài liệu tham khảo:** 7](#_Toc140789241)

1. **Ý tưởng thực hiện**

* Bước 1: Chọn số k (int) ngẫu nhiên.
* Bước 2: Ngẫu nhiên gán một trọng tâm cho mỗi trong số k cụm.
* Bước 3: Tính khoảng cách từ tất cả các quan sát tới mỗi trọng tâm của k cụm.
* Bước 4: Gán các quan sát vào trọng tâm gần nhất.
* Bước 5: Tìm vị trí mới của trọng tâm bằng cách lấy giá trị trung bình của tất cả các quan sát trong mỗi cụm.
* Bước 6: Lặp lại các bước 3-5 cho đến khi các trọng tâm không thay đổi vị trí.

1. **Miêu tả thuật toán:**

* Đọc ảnh bằng thư viện PIL.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình : hàm đọc ảnh.

* Hàm lưu ảnh dưới dạng png:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình : Hàm dùng để lưu ảnh dưới dạng png.

* Chuyển đổi ảnh thành ma trận bằng thư viện numpy. Trả về ma trận và số hàng, số cột của ma trận đó.

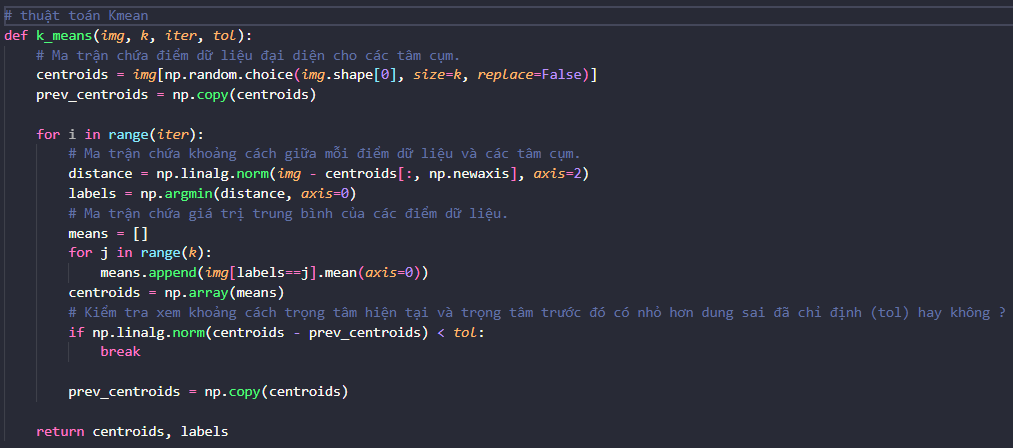
**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

Hình : hàm reshape\_img.

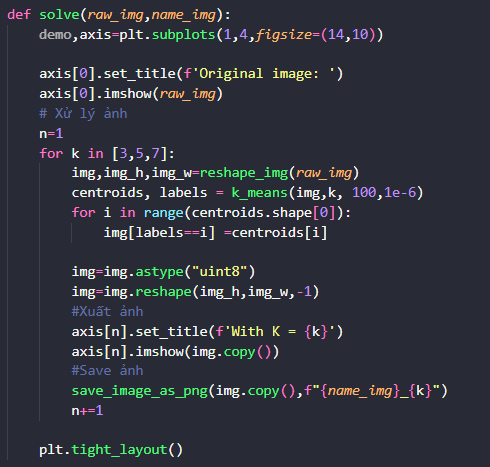
* Thuật toán K-means: trả về các điểm tâm cụm cuối cùng và nhãn của mỗi điểm dữ liệu tương ứng với cụm của nó. Các điểm tâm cụm cuối cùng đại diện cho các cụm cuối cùng của dữ liệu.

+ Khởi tạo: Chọn ngẫu nhiên k điểm từ dữ liệu ban đầu làm điểm tâm cụm ban đầu. Các điểm này sẽ đại diện cho các cụm ban đầu của dữ liệu.  
+ Gán nhãn: Tính khoảng cách giữa từng điểm dữ liệu và các điểm tâm cụm đã chọn. Gán mỗi điểm dữ liệu vào cụm có điểm tâm gần nhất dựa trên khoảng cách đã tính. Điều này dẫn đến việc tạo ra các cụm ban đầu.  
+ Cập nhật điểm tâm cụm: Tính giá trị trung bình của các điểm dữ liệu trong từng cụm để có được các điểm tâm cụm mới. Cập nhật các điểm tâm cụm ban đầu bằng các điểm tâm cụm mới.  
+ Lặp lại: Lặp lại bước 2 và 3 cho một số lần lặp (iter) cho trước hoặc cho đến khi điều kiện dừng được đáp ứng. Thông thường, điều kiện dừng là khi các điểm tâm cụm không thay đổi đáng kể sau mỗi lần cập nhật.



Hình : Thuật toán K means.

* Hàm Solve(): dùng để xử lý ảnh ở K=[3,5,7], xuất ảnh đó ra và lưu ảnh ở dạng png.



Hình : Hàm solve.

* Hàm main(): Người dùng chỉ cần nhập vào tên ảnh, không cần nhập định dạng ảnh(.png,.jpeg,…).

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

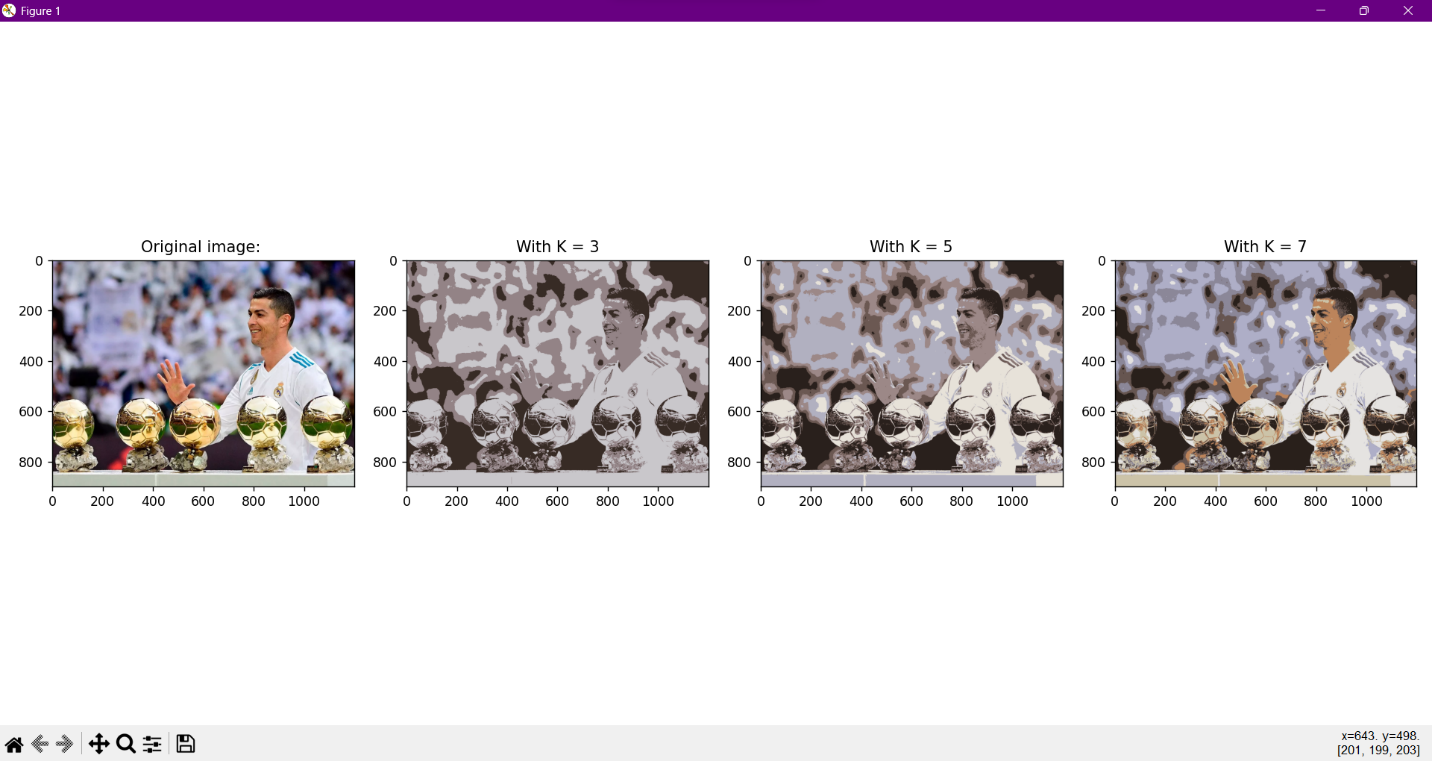
Mô tả được tạo tự động

Hình : hàm main.

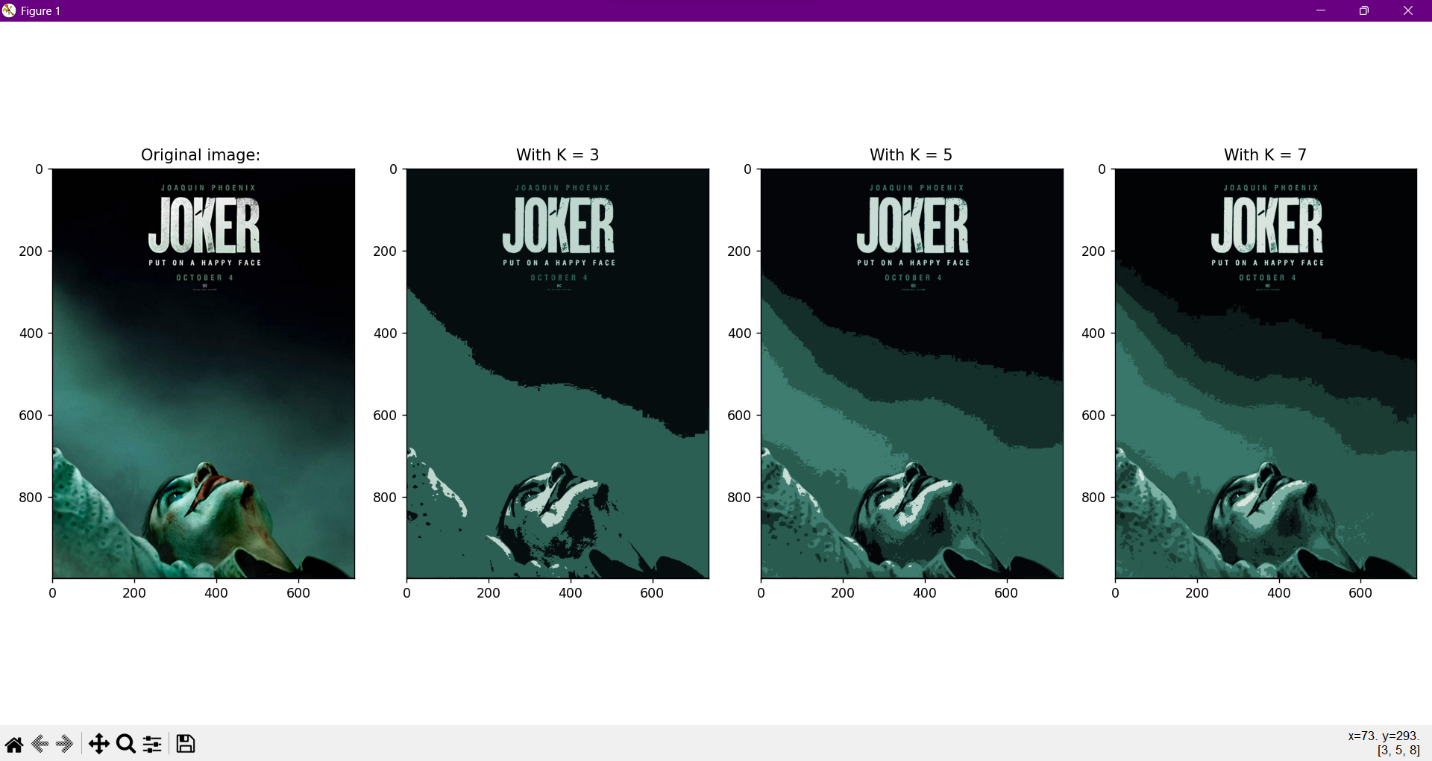
1. **Test case:**



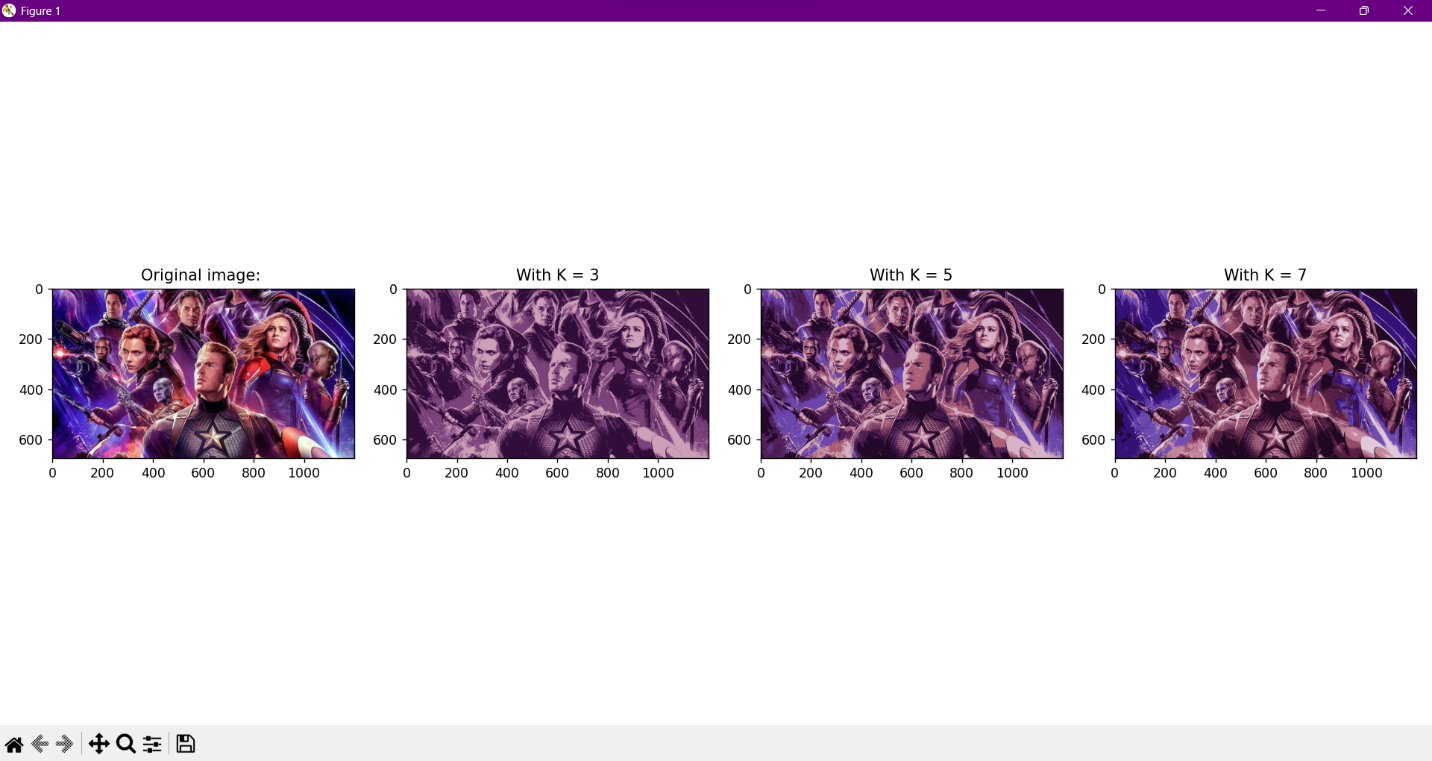
Hình : Test case 1.



Hình : Test case 2.



Hình : Test case 3.



Hình : Test case 4.

1. **Kết luận:**

* K càng lớn, ảnh càng được phân chia chi tiết: K-means sẽ tách dữ liệu thành k cụm khác nhau. Nếu k càng lớn, điều này dẫn đến việc phân chia dữ liệu thành nhiều nhóm nhỏ hơn và chi tiết hơn. Trong hình ảnh, điều này có thể thể hiện dưới dạng việc phân chia ảnh thành nhiều phần nhỏ, mỗi phần biểu diễn một cụm.
* K càng lớn, ảnh càng được phân chia chi tiết: K-means sẽ tách dữ liệu thành k cụm khác nhau. Nếu k càng lớn, điều này dẫn đến việc phân chia dữ liệu thành nhiều nhóm nhỏ hơn và chi tiết hơn. Trong hình ảnh, điều này có thể thể hiện dưới dạng việc phân chia ảnh thành nhiều phần nhỏ, mỗi phần biểu diễn một cụm.

1. **Tài liệu tham khảo:**

* <https://www.dominodatalab.com/blog/getting-started-with-k-means-clustering-in-python>
* https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html