

Tên: Nguyễn Hoàng Minh  
Mssv: 18127265  
Lớp: 18CLC1  
Môn: Toán ứng dụng và thống kê

## Project 1

### Ý tưởng:

Giảm màu hình ảnh bằng thuật toán Kmeans:

- Bước 1: chọn random k pixels làm center
- Bước 2: phân lớp các pixel dựa vào khoảng cách của nó đến từng center, phân vào lớp có khoảng cách ngắn nhất
- Bước 3: Tính trung bình chỉ số R,G,B của các pixel trong từng phân lớp và chọn giá trị đó làm pixel center mới.
- Bước 4: Lặp lại từ bước 2 đến khi nào các pixel thuộc phân lớp trước và sau không thay đổi.

### Mô tả hàm:

*initcentroids*(string *mode*, np.array *img\_1d*, int *k\_clusters*)

- mode: "random" – chọn các pixel random từ (0-255), "in\_pixels" – chọn các pixel random từ hình ảnh input

Hàm trả về 1 mảng numpy có k phần tử, mỗi phần tử là 1 center [R,G,B]

*classify*(np.array *centroids*, np.array *img\_1d*):

- Hàm phân lớp các pixel trong mảng *img\_1d* và trả về list có chiều dài bằng số phần tử của mảng *img\_1d* , mỗi phần tử là trị số lớp của phần tử trong *img\_1d*.

*kmeans*(*img\_1d*, *k\_clusters*, *max\_iter*, *init\_centroids*='random'):

- truyền vào hình ảnh, số lượng màu mong muốn, và *max\_iter*: giới hạn số lần lặp.

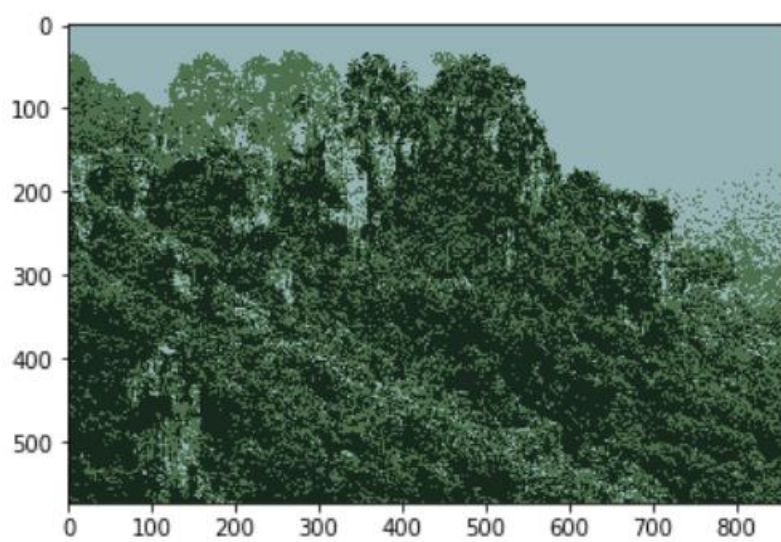
- Trả về các center cuối cùng thỏa điều kiện, và list chứa trị số lớp các pixel trong *img\_1d*

### Hình ảnh kết quả:

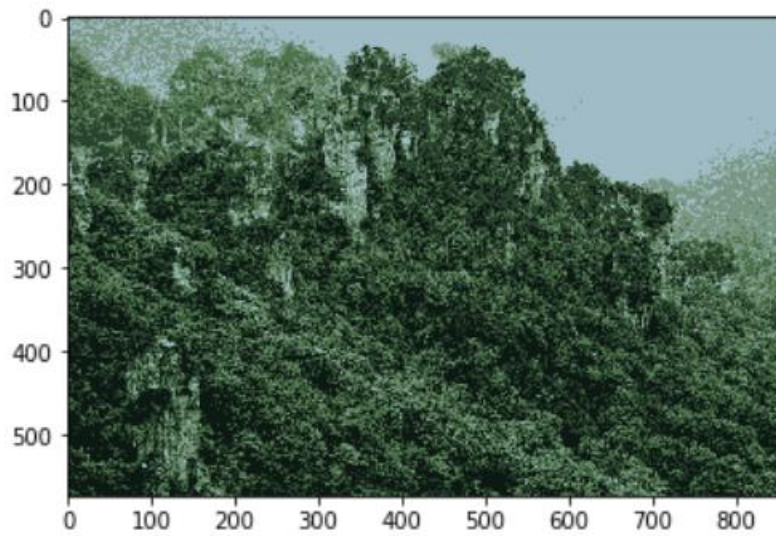
Input:



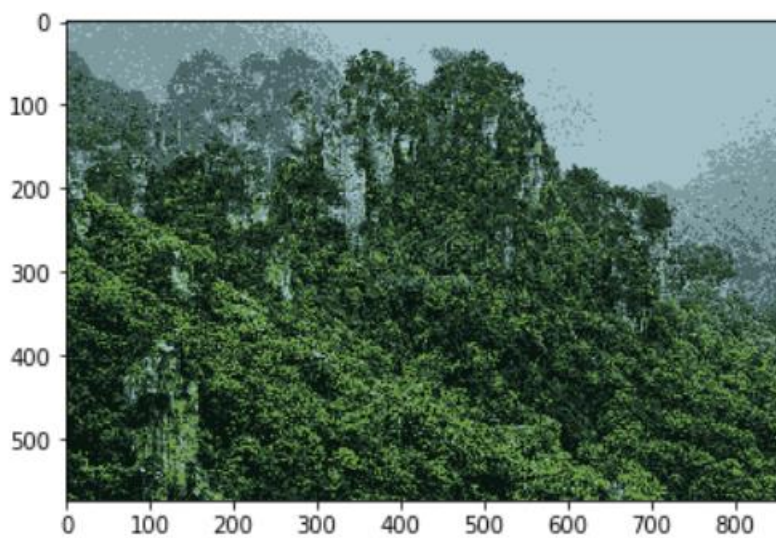
K=3:



K=5:



K=7:



Nhận xét:

- K càng lớn thì hình ảnh càng rõ ràng, thời gian xử lý càng chậm
- Với K thấp, một số màu sẽ bị triệt tiêu khiến cho hình ảnh khó có thể hình dung hoặc có thể thay đổi tính chất của hình ảnh, ví dụ: đèn giao thông.