

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN TỬ - VIỄN THÔNG**



**BÁO CÁO BÀI TẬP 6**  
**THUẬT TOÁN K-MEANS CLUSTERING**  
**XÂY DỰNG HỆ THỐNG PHÁT HIỆN TRÁI**  
**CHÍN ĐỎ TRONG ẢNH**

**NHÓM 10**

|                             |                          |       |
|-----------------------------|--------------------------|-------|
| <b>Sinh viên thực hiện</b>  | : Đinh Văn Quang         | 20DT1 |
|                             | Hà Phước Phúc            | 20DT2 |
|                             | Nguyễn Văn Quý           | 21DT2 |
| <b>Lớp học phần</b>         | : 20.38B                 |       |
| <b>Giảng viên hướng dẫn</b> | : TS. Hoàng Lê Uyên Thục |       |

**Đà Nẵng, 10/2024**

## MỤC LỤC

|   |           |
|---|-----------|
| <b>1. Giới thiệu:</b> .....                       | <b>3</b>  |
| <b>1.1 Tính cấp thiết</b> .....                   | <b>3</b>  |
| <b>1.2 Giới thiệu bài toán</b> .....              | <b>3</b>  |
| <b>1.3 Giới thiệu phương pháp thực hiện</b> ..... | <b>3</b>  |
| <b>2. Thực hiện:</b> .....                        | <b>4</b>  |
| <b>2.1 Các bước thực hiện</b> .....               | <b>4</b>  |
| <b>2.2 Kết quả</b> .....                          | <b>7</b>  |
| <b>a. Đối với ảnh RipeFruit1:</b> .....           | <b>7</b>  |
| <b>b. Đối với ảnh RipeFruit2:</b> .....           | <b>8</b>  |
| <b>3. Kết luận</b> .....                          | <b>8</b>  |
| <b>3.1 Nhận xét</b> .....                         | <b>8</b>  |
| <b>a. Đối với ảnh RipeFruit1:</b> .....           | <b>8</b>  |
| <b>b. Đối với ảnh RipeFruit2:</b> .....           | <b>8</b>  |
| <b>3.2 Đánh giá mô hình và phương pháp</b> .....  | <b>9</b>  |
| <b>a. Ưu điểm</b> .....                           | <b>9</b>  |
| <b>b. Nhược điểm</b> .....                        | <b>9</b>  |
| <b>c. Hướng khắc phục</b> .....                   | <b>9</b>  |
| <b>TÀI LIỆU THAM KHẢO</b> .....                   | <b>10</b> |

## **1. Giới thiệu:**

### **1.1 Tính cấp thiết**

Tự động hóa thu hoạch trong nông nghiệp hiện đại mang lại nhiều lợi ích về hiệu quả và chi phí. Tuy nhiên, việc phân biệt quả chín và chưa chín qua hình ảnh vẫn là một thách thức lớn. Nhận diện chính xác độ chín của quả bằng máy tính có thể tối ưu hóa quá trình thu hoạch và giảm hao hụt. Thuật toán K-means là một phương pháp phân cụm không giám sát, đã chứng tỏ hiệu quả trong phân tích ảnh nhờ khả năng phân loại dữ liệu dựa trên khoảng cách để nhóm những điểm gần nhau nhất để tạo thành những cụm. Nghiên cứu này tìm hiểu việc áp dụng K-means để phát hiện độ chín của quả thông qua ảnh, đề xuất cách sử dụng thuật toán này để phân biệt quả chín và chưa chín dựa trên màu sắc là một tiêu chí quan trọng trong quyết định thu hoạch nhiều loại trái cây.

### **1.2 Giới thiệu bài toán**

- Xây dựng hệ thống phát hiện trái chín đỏ trong ảnh sử dụng thuật toán gom cụm K-means tập trung vào việc áp dụng kỹ thuật xử lý ảnh để phát hiện các trái cây đã chín thông qua màu sắc. Bài toán được xét trong hai trường hợp:

+ Trường hợp 1: Chỉ có một trái chín đỏ trong ảnh.

+ Trường hợp 2: Có nhiều trái chín đỏ trong ảnh.

Mục tiêu chính của hệ thống là phân chia các điểm ảnh thành 3 cụm màu khác nhau ( $K=3$ ), trong đó một cụm đại diện cho màu sắc của trái chín đỏ. Thuật toán K-means sẽ được thực hiện với số vòng lặp khác nhau, lần lượt là 5 và 10 vòng, nhằm kiểm tra độ ổn định và hiệu quả của quá trình gom cụm. Đầu vào của bài toán là các ảnh chụp thực tế chứa một hoặc nhiều trái chín, từ đó hệ thống sẽ phân tích và xác định khu vực có màu sắc tương ứng với trái cây đã chín.

Việc ứng dụng thuật toán K-means trong bài toán này giúp phân loại các điểm ảnh theo màu sắc, hỗ trợ việc nhận diện các trái cây chín trong ảnh một cách nhanh chóng và hiệu quả.

### **1.3 Giới thiệu phương pháp thực hiện**

- Phương pháp phân cụm K-means là một thuật toán phổ biến trong lĩnh vực học máy và xử lý dữ liệu. Thuật toán K-means là một phương pháp phân cụm không giám sát, phân tích ảnh nhờ khả năng phân loại dữ liệu dựa trên khoảng cách để nhóm những điểm gần nhau nhất để tạo thành K cụm. Mục tiêu của K-means là chia tập dữ liệu thành K cụm (clusters) dựa trên sự tương đồng giữa các điểm dữ liệu. Các cụm này được hình thành sao cho các điểm dữ liệu trong cùng một cụm gần nhau hơn so với các điểm dữ liệu ở các cụm khác.

Quá trình thực hiện của thuật toán K-means gồm các bước chính sau:

**Bước 1.** Khởi tạo: phân chia các điểm dữ liệu thành K cụm ngẫu nhiên. Tìm các tâm (centroids) của mỗi cụm.

**Bước 2.** Phân cụm dữ liệu: đối với mỗi điểm dữ liệu, tính khoảng cách từ điểm dữ liệu của mỗi cụm, sau đó gán mỗi điểm dữ liệu vào cụm có tâm gần nhất.

Công thức Euclidean

$$\sqrt{\sum_{i=1}^n |x_i - y_i|^2}$$

**Bước 3.** Xác định tâm cụm: sau khi tất cả các điểm dữ liệu đã được gán vào các cụm, tính toán lại tâm cụm bằng cách lấy trung bình của tất cả các điểm trong cụm.

**Bước 4.** Lặp lại: quá trình gán điểm dữ liệu và cập nhật tâm cụm được lặp lại ở bước 2 và 3 cho đến khi không có sự thay đổi đáng kể về vị trí của các tâm hoặc khi đạt đến số lần lặp tối đa.

- Thuật toán K-means hoạt động dựa trên nguyên tắc tối ưu hóa, cố gắng giảm thiểu tổng khoảng cách giữa các điểm dữ liệu và tâm cụm tương ứng, từ đó tạo ra các cụm dữ liệu có sự liên kết chặt chẽ hơn.
- Phương pháp này có nhiều ứng dụng trong phân tích dữ liệu, nhận dạng mẫu, xử lý ảnh và nhiều lĩnh vực khác.

## 2. Thực hiện:

### 2.1 Các bước thực hiện

#### Bước 1: Khởi tạo

- Khởi tạo tâm cụm với 3 tâm cụm ban đầu tương ứng với màu đỏ, xanh lá, và xám.
  - + Màu đỏ: centroids[0] = [255, 0, 0], đại diện cho các vùng có màu đỏ, như quả chín.
  - + Màu xanh lá: centroids[1] = [0, 255, 0], đại diện cho các vùng có màu xanh lá, như quả chưa chín.
  - + Màu xám: centroids[2] = [128, 128, 128], đại diện cho các vùng trung tính hoặc nền trong ảnh.
- Điều này có nghĩa là ảnh sẽ được phân cụm theo ba màu sắc chủ đạo: đỏ, xanh lá và xám.

```
centroids = np.array([[255, 0, 0], # Đỏ
                       [0, 255, 0], # Xanh lá
                       [128, 128, 128]]) # Màu xám
```

#### Bước 2: Phân cụm dữ liệu

- Trong bước này, mỗi điểm ảnh trong ảnh đầu vào sẽ được gán vào một trong ba cụm dựa trên khoảng cách của nó đến các tâm cụm hiện tại. Khoảng cách được tính bằng công thức Euclid, đo sự tương đồng giữa giá trị màu sắc của điểm ảnh với các tâm cụm.
- Điểm ảnh sẽ được gán vào cụm có tâm cụm gần nhất với nó. Điều này giúp gom các điểm ảnh có màu sắc tương tự nhau vào cùng một cụm.

- Với  $K=3$  để chia ảnh thành ba cụm: một cụm đại diện cho trái chín đỏ, một cụm cho lá cây hoặc các vùng xanh, và một cụm cho nền ảnh hoặc các yếu tố khác.

```
def kmeans(pixels, iterations, k=3):
    global labels
    centroids = np.array([[255, 0, 0], # Đỏ
                          [0, 255, 0], # Xanh lá
                          [128, 128, 128]]) # Màu xám

    for _ in range(iterations):
        distances = np.sqrt(((pixels[:, np.newaxis] - centroids) ** 2).sum(axis=2))
        labels = np.argmin(distances, axis=1)

        for i in range(k):
            if np.any(labels == i):
                centroids[i] = pixels[labels == i].mean(axis=0)

    quantized_image = centroids[labels].reshape(-1, 3)

    return quantized_image.astype(np.uint8), labels, centroids
```

### Bước 3. Xác định tâm cụm

- Sau khi tất cả các điểm ảnh đã được gán vào các cụm tương ứng, tâm của mỗi cụm sẽ được cập nhật. Tâm cụm mới được tính bằng cách lấy giá trị trung bình của tất cả các điểm ảnh trong cụm đó. Đây là bước cực kỳ quan trọng, vì nó giúp điều chỉnh vị trí của tâm cụm sao cho nó phản ánh chính xác màu sắc trung bình của cụm.

- Trong hệ thống phát hiện trái chín, sau mỗi vòng lặp, tâm cụm đại diện cho trái cây chín sẽ được điều chỉnh dần dần dựa trên màu đỏ của các điểm ảnh trong ảnh, giúp phân chia các cụm một cách hiệu quả hơn.

```
for i in range(k):
    if np.any(labels == i):
        centers[i] = pixels[labels == i].mean(axis=0)
```

### Bước 4. Vòng lặp

- Quá trình phân cụm và cập nhật tâm cụm sẽ lặp lại cho đến khi tâm của các cụm không thay đổi hoặc sau một số vòng lặp nhất định. Trong bài toán này, chúng ta thực hiện K-means với 2 trường hợp khác nhau:

- + Trường hợp 1: Chạy thuật toán với 5 vòng lặp.
- + Trường hợp 2: Chạy thuật toán với 10 vòng lặp.

```

for _ in range(iterations):
    distances = np.sqrt(((pixels[:, np.newaxis] - centroids) ** 2).sum(axis=2))
    labels = np.argmin(distances, axis=1)

    for i in range(k):
        if np.any(labels == i):
            centroids[i] = pixels[labels == i].mean(axis=0)

quantized_image = centroids[labels].reshape(-1, 3)

return quantized_image.astype(np.uint8), labels, centroids

if __name__ == "__main__":
    image = cv2.imread('picture/RipeFruit2.png')

    # Phát hiện trái chín với 5 vòng lặp
    segmented_result_5, kmeans_result_5 = detect_ripe_fruits(image, iterations=5)

    # Phát hiện trái chín với 10 vòng lặp
    segmented_result_10, kmeans_result_10 = detect_ripe_fruits(image, iterations=10)

```

Đoạn mã trên lặp lại quá trình tính khoảng cách, gán cụm, và cập nhật tâm cụm trong 5 vòng lặp và 10 vòng lặp (được truyền vào từ tham số iterations).

### Bước 5. Phát hiện trái chín đỏ trong ảnh

- Sau khi thực hiện thuật toán K-means và phân cụm các điểm ảnh trong ảnh, kết quả sẽ cho ra một ảnh được phân loại thành ba cụm màu khác nhau, trong đó một cụm đại diện cho trái cây chín (màu đỏ).
- Để phát hiện khu vực chứa trái chín, sử dụng một mặt nạ màu đỏ. Mặt nạ này chỉ giữ lại các điểm ảnh có màu nằm trong khoảng màu đỏ (từ đỏ sáng đến đỏ đậm), tương ứng với màu sắc của trái chín.

```
red_mask = cv2.inRange(segmented_image, (160, 0, 0), (255, 100, 100))
```

#### - Giải thích

+ `cv2.inRange(segmented_image, (160, 0, 0), (255, 100, 100))`: Hàm này tạo mặt nạ bằng cách giữ lại các điểm ảnh trong ảnh `segmented_image` có giá trị màu trong khoảng từ (160, 0, 0) đến (255, 100, 100). Đây là khoảng màu đỏ dùng để phát hiện trái cây chín.

+ Kết quả là một mặt nạ nhị phân, trong đó các điểm ảnh màu đỏ được giữ lại (giá trị 1), các điểm ảnh khác được loại bỏ (giá trị 0).

### Bước 6. Tìm đường viền của khu vực chứa trái chín

- Sau khi có mặt nạ màu đỏ, chúng ta sử dụng hàm `cv2.findContours` để tìm các đường viền bao quanh các khu vực có màu đỏ (trái chín) trong ảnh. Các đường viền này được sử dụng để xác định vị trí và kích thước của các khu vực chứa trái chín.

```
contours = cv2.findContours(red_mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
```

## Bước 7. Loại bỏ các vùng nhiễu

- Để tránh nhận diện sai các vùng nhiễu nhỏ trong ảnh, chúng ta cần loại bỏ các đường viền có diện tích nhỏ hơn một giá trị ngưỡng (min\_area). Điều này giúp chỉ tập trung vào các vùng có kích thước đủ lớn để được coi là trái cây chín.

```
def remove_small_boxes(contours, min_area=1000):  
    filtered_contours = []  
    for cnt in contours:  
        area = cv2.contourArea(cnt)  
        if area > min_area:  
            filtered_contours.append(cnt)  
    return filtered_contours  
  
contours = remove_small_boxes(contours, min_area=1500)
```

## Bước 8. Vẽ hình chữ nhật bao quanh trái cây chín

Sau khi lọc bỏ các vùng nhiễu, chúng ta sử dụng danh sách các đường viền còn lại để vẽ các hình chữ nhật bao quanh các vùng chứa trái cây chín đỏ.

```
result_image = rgb_image.copy()  
for box in final_boxes:  
    x, y, w, h = box  
    cv2.rectangle(result_image, (x, y), (x + w, y + h), (255, 0, 0), 2)  
  
return result_image, segmented_image
```

## 2.2 Kết quả

- Dưới đây là kết quả thu được khi thử nghiệm với tập dữ liệu là ảnh quả chín đỏ, gồm 2 trường hợp là ảnh **RipeFruit1** có một quả chín và ảnh **RipeFruit2** có nhiều quả chín

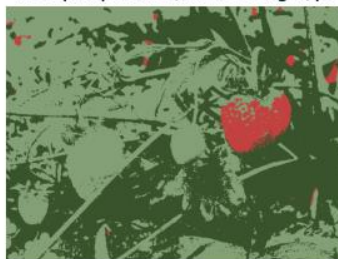
- Ảnh kết quả bao gồm (thứ tự từ trái sang phải) ảnh đầu tiên là ảnh gốc, ảnh thứ hai là kết quả phân cụm và ảnh thứ ba là ảnh đã phát hiện quả chín được khoanh vùng đỏ

### a. Đối với ảnh RipeFruit1:

Ảnh gốc



Kết quả phân cụm (5 vòng lặp)



Kết quả phát hiện trái chín (5 vòng lặp)



Kết quả phân cụm (10 vòng lặp)

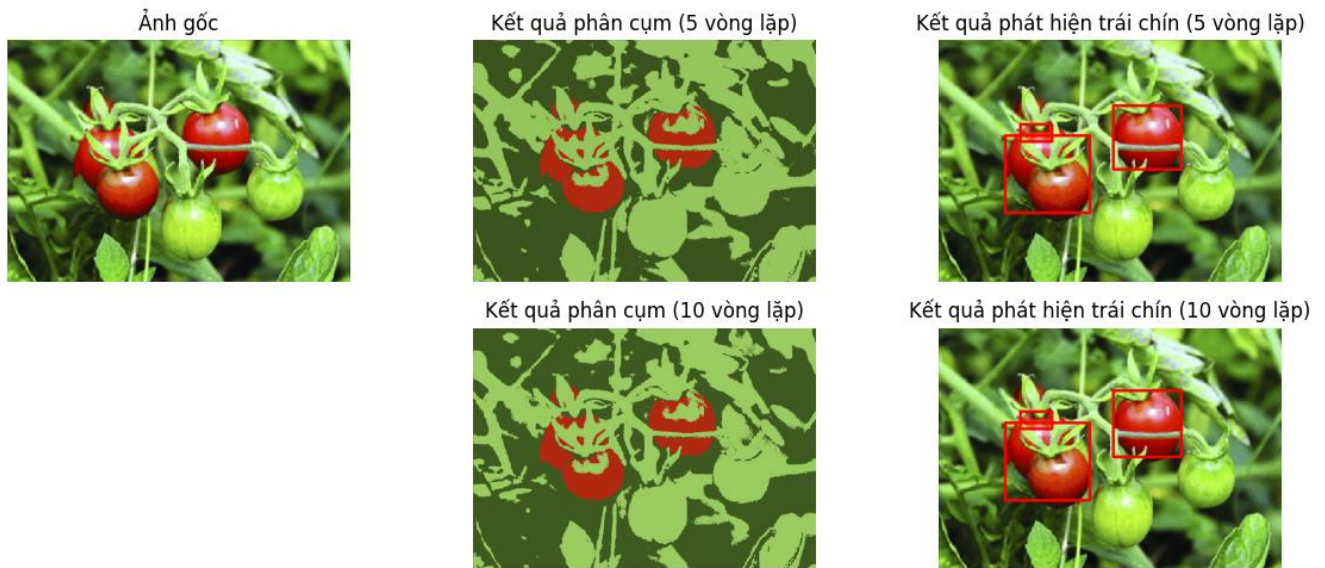


Kết quả phát hiện trái chín (10 vòng lặp)





## b. Đối với ảnh RipeFruit2:



## 3. Kết luận

### 3.1 Nhận xét

#### a. Đối với ảnh RipeFruit1:

- Kết quả phân cụm 5 vòng lặp (Hàng trên, giữa): Sau khi áp dụng thuật toán K-means với 5 vòng lặp, ảnh được chia thành 3 cụm màu chính. Cụm màu đỏ đại diện cho khu vực trái dâu đã chín, cụm màu xám/xanh đại diện cho lá cây và phần nền. Quả chín màu đỏ được phân tách khá rõ ràng, tuy nhiên có một số nhiễu (những chấm đỏ nhỏ ở các khu vực không phải là trái cây chín) do ảnh hưởng của ánh sáng hoặc màu nền.
- Kết quả phát hiện trái chín 5 vòng lặp (Hàng trên, phải): Hệ thống đã nhận diện chính xác khu vực chứa trái dâu tây chín màu đỏ, và vẽ một hình chữ nhật xanh bao quanh. Tuy nhiên, có thể thấy trong ảnh vẫn còn những chi tiết thừa, chẳng hạn như một số chấm nhỏ không liên quan ở nền.
- Kết quả phân cụm 10 vòng lặp (Hàng dưới, giữa): Với 10 vòng lặp, thuật toán K-means cho ra kết quả phân cụm ổn định hơn. Cụm màu đỏ đại diện cho trái chín rõ ràng hơn, các điểm nhiễu ở nền đã được giảm đi so với 5 vòng lặp. Điều này cho thấy việc tăng số vòng lặp đã giúp cải thiện độ chính xác của thuật toán.
- Kết quả phát hiện trái chín (10 vòng lặp) (Hàng dưới, phải): Vẫn giữ được độ chính xác cao, với hình chữ nhật bao quanh trái dâu chín. So với kết quả sau 5 vòng lặp, vùng trái chín được phát hiện với ít nhiễu hơn.

#### b. Đối với ảnh RipeFruit2:

- Kết quả phân cụm 5 vòng lặp (Hàng trên, giữa): Thuật toán K-means với 5 vòng lặp đã chia hình ảnh thành 3 cụm màu. Cụm màu đỏ tập trung chính xác vào các khu vực của trái cà chua chín, trong khi cụm màu xanh và xám bao phủ các khu vực của lá cây và



trái chưa chín. Tuy nhiên, một vài chi tiết nhỏ bị nhiễu (đặc biệt là trên phần lá) do ảnh hưởng của độ phức tạp về màu sắc trong ảnh.

- Kết quả phát hiện trái chín 5 vòng lặp (Hàng trên, phải): Hệ thống đã phát hiện các trái cà chua chín và vẽ hình chữ nhật đỏ bao quanh chúng. Mặc dù hệ thống đã xác định đúng khu vực trái cà chua, nhưng vị trí các hình chữ nhật vẫn còn chồng lấn một chút hoặc thiếu chính xác tại một vài quả.

- Kết quả phân cụm 10 vòng lặp (Hàng dưới, giữa): Với 10 vòng lặp, việc phân cụm màu sắc trở nên rõ ràng hơn. Khu vực trái chín đỏ được phân tách chính xác hơn và các chi tiết nhiễu từ lá cây được giảm thiểu đáng kể. Điều này cho thấy rằng việc tăng số vòng lặp đã cải thiện đáng kể độ chính xác của thuật toán.

- Kết quả phát hiện trái chín 10 vòng lặp (Hàng dưới, phải): Hệ thống tiếp tục phát hiện chính xác các trái cà chua chín, với các hình chữ nhật bao quanh từng trái. So với kết quả sau 5 vòng lặp, vị trí các hình chữ nhật đã được điều chỉnh chính xác hơn và ít bị chồng lấn hơn.

### **3.2 Đánh giá mô hình và phương pháp**

#### **a. Ưu điểm**

- Đơn giản và hiệu quả: Thuật toán K-means khá đơn giản và dễ triển khai. Nó hoạt động hiệu quả trong việc phân cụm dữ liệu dựa trên đặc trưng màu sắc và vị trí.

- Khả năng xử lý dữ liệu lớn: K-means có thể xử lý các tập dữ liệu ảnh lớn với số lượng điểm ảnh nhiều mà không bị ảnh hưởng đáng kể về hiệu suất.

- Không giám sát: K-means là một thuật toán phân cụm không giám sát, không yêu cầu dữ liệu đã được gán nhãn trước, phù hợp với bài toán phát hiện quả chín.

- Tính linh hoạt: Có thể điều chỉnh số lượng cụm K để phù hợp với các loại quả khác nhau hoặc điều kiện môi trường khác nhau.

#### **b. Nhược điểm**

- Nhạy cảm với nhiễu và giá trị ngoại lai (outlier): Thuật toán K-means có thể bị ảnh hưởng bởi nhiễu trong ảnh hoặc các giá trị ngoại lai, dẫn đến kết quả phân cụm không chính xác.

- Khó xác định số lượng cụm tối ưu: Việc lựa chọn số lượng cụm K phù hợp là một thách thức, và có thể ảnh hưởng đến kết quả phân cụm.

- Không phù hợp với cụm có hình dạng phức tạp: K-means hoạt động tốt nhất với các cụm có hình dạng gần đúng hình cầu. Đối với các cụm có hình dạng phức tạp, hiệu suất có thể giảm.

- Thuật toán khó phát hiện các quả chín có màu sắc tương tự với màu nền trong ảnh

#### **c. Hướng khắc phục**

- Kết hợp với kỹ thuật xử lý ảnh:

  - +Tách nền để giảm nhiễu và tăng độ tương phản

- +Làm mịn ảnh để giảm ảnh hưởng của điểm ngoại lai
- +Sử dụng KMeans để cải thiện việc khởi tạo tâm cụm
- Bổ sung đặc trưng:
  - +Phân tích hình dạng và kích thước đối tượng
  - +Kết hợp nhiều loại đặc trưng để tăng độ chính xác trong phát hiện quả chín
- Xem xét các phương pháp phân cụm thay thế:
  - +Phân cụm mật độ cho các hình dạng phức tạp
  - +Phân cụm phân cực để xử lý các trường hợp đặc biệt
- Tối ưu hóa quy trình:
  - +Điều chỉnh các tham số thuật toán
  - +Kết hợp nhiều phương pháp để tận dụng ưu điểm của từng kỹ thuật

## TÀI LIỆU THAM KHẢO

- [1]. <https://ndquy.github.io/posts/thuat-toan-phan-cum-kmeans/>
- [2]. [https://www.geeksforgeeks.org/image-segmentation-using-k-means-clustering/?ref=oin\\_asr1](https://www.geeksforgeeks.org/image-segmentation-using-k-means-clustering/?ref=oin_asr1)
- [3]. <https://machinelearningcoban.com/2017/01/01/kmeans/>
- [4]. <https://www.youtube.com/watch?v=22mpExWh1LY>
- [5] [StatQuest: K-means clustering](#)

\*\*\*Source Code: [Github](#)