



Bài giảng môn học:

**Thị giác máy tính (7080518)**

# CHƯƠNG 4: PHÂN VÙNG ẢNH (Image Segmentation)

Đặng Văn Nam

dangvannam@humg.edu.vn

# Nội dung chương 4

1. Bài toán phân vùng ảnh (image segmentation)
2. Ứng dụng của phân vùng ảnh
3. Một số kỹ thuật phân vùng ảnh
  1. Phân vùng ảnh dựa trên ngưỡng
  2. Phân vùng ảnh dựa trên cạnh
  3. Phân vùng ảnh dựa trên kỹ thuật phân cụm

# 1. Bài toán phân vùng ảnh

# Giới thiệu

- Phân vùng ảnh (image segmentation) là một kỹ thuật quan trọng trong thị giác máy tính. Đây là tiền đề của quá trình xử lý dữ liệu hình ảnh.
- Phân vùng ảnh là được ứng dụng trong nhiều lĩnh vực khác nhau như lĩnh vực hình ảnh y tế (medical imaging), phát hiện và nhận dạng đối tượng, hệ thống camera giám sát, hệ thống điều khiển giao thông...
- Kết quả phân vùng tốt sẽ tạo điều kiện thuận lợi cho các khâu xử lý về sau, đảm bảo tính hiệu quả cao, gia tăng mức độ chính xác, đồng thời giảm thiểu nguồn lực tính toán.

Input Image

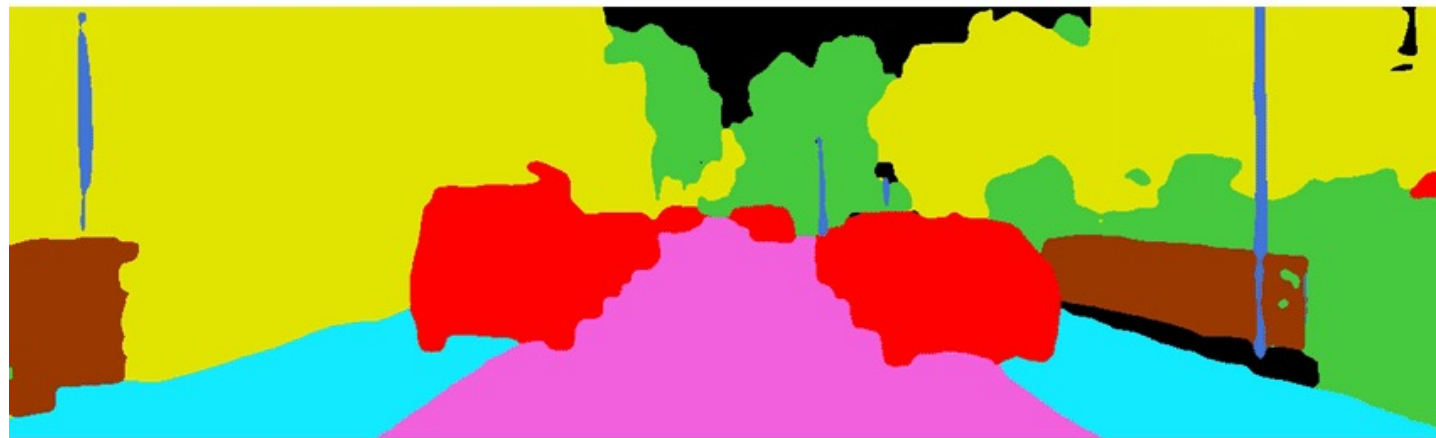


Semantic Segmentation



# Phân vùng ảnh là gì?

- **Phân vùng ảnh** là một phương pháp mà trong đó, hình ảnh kỹ thuật số được chia thành nhiều nhóm con khác nhau được gọi là segments.
- Một cách dễ hiểu, phân vùng ảnh là một quá trình gán nhãn (assigning a label) cho mỗi điểm ảnh trong một bức ảnh, các điểm ảnh trong cùng một nhãn sẽ có những đặc tính giống nhau về màu sắc, cường độ hoặc kết cấu của ảnh

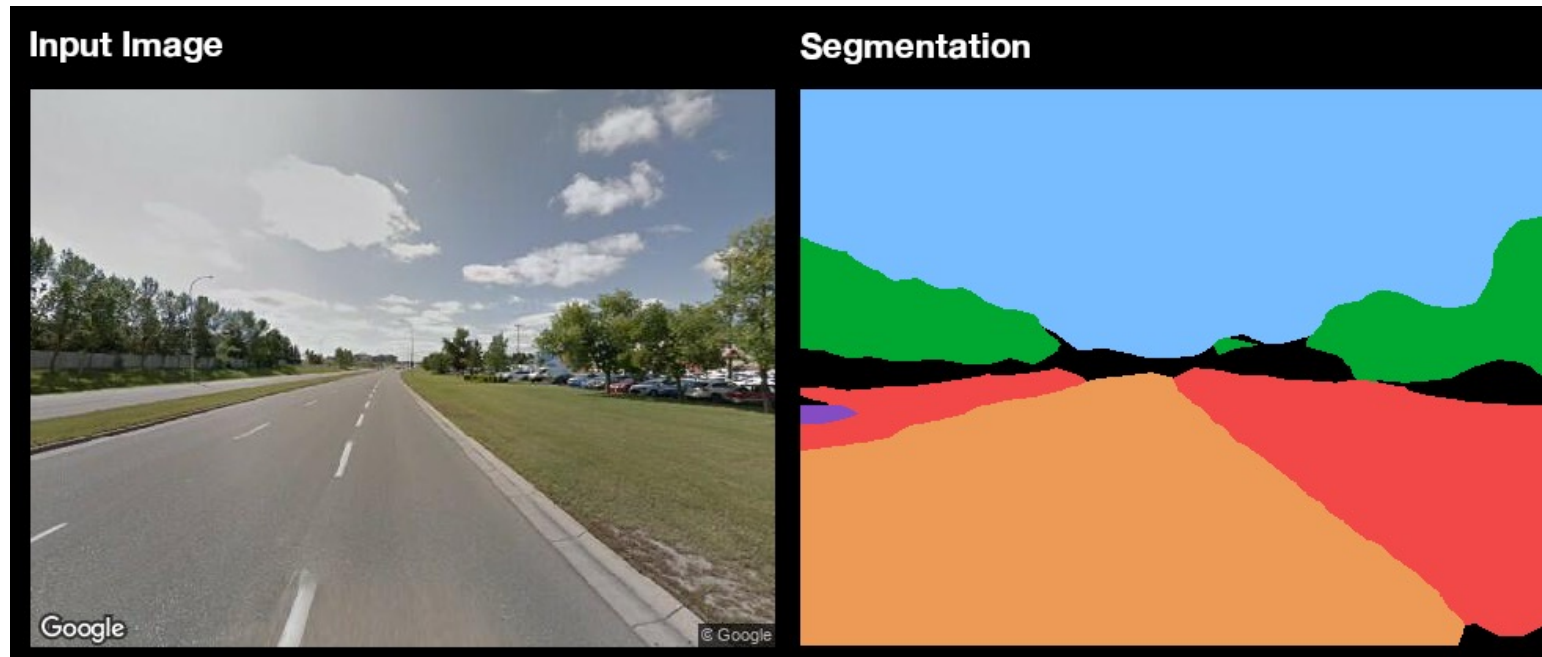


 Road	 Sidewalk	 Building	 Fence
 Pole	 Vegetation	 Vehicle	 Unlabel



# Phân vùng ảnh là gì?

- **Mục tiêu của phân vùng ảnh:** Làm giảm độ phức tạp của hình ảnh, giúp cho quá trình xử lý hoặc phân tích hình ảnh sau đó trở nên đơn giản hơn.
- **Kết quả của việc phân vùng ảnh là tập hợp các phân đoạn** (segments) bao gồm có thể là toàn bộ bức ảnh hoặc tập hợp các đường biên chiết xuất từ hình ảnh. Các điểm ảnh trong cùng một vùng có đặc tính tương tự nhau về màu sắc, cường độ hoặc kết cấu. Các vùng lân cận thì khác nhau đáng kể về các đặc trưng trên.



# Phân vùng ảnh là gì?

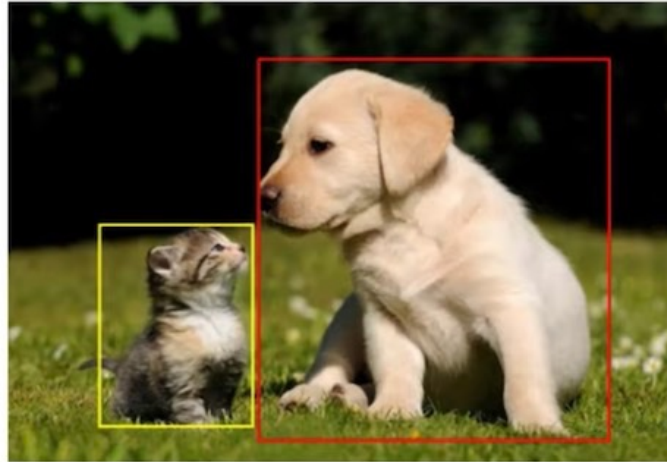
- Phân biệt một số bài toán trong thị giác máy tính

Is this a dog?



Image Classification

What is there in image and where?



Object Detection

Which pixels belong to which object?

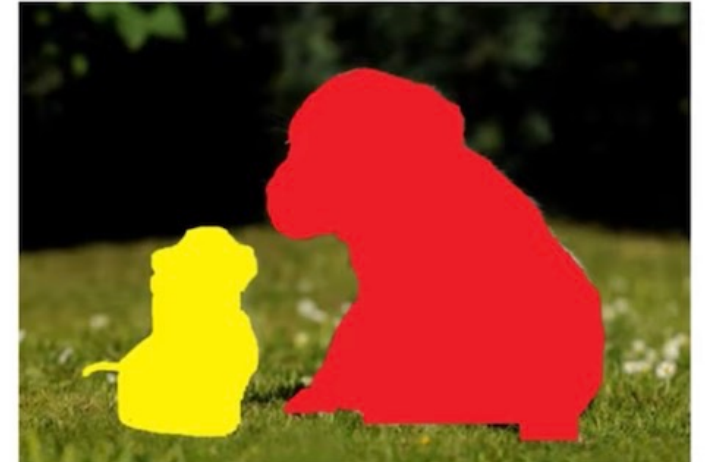


Image Segmentation

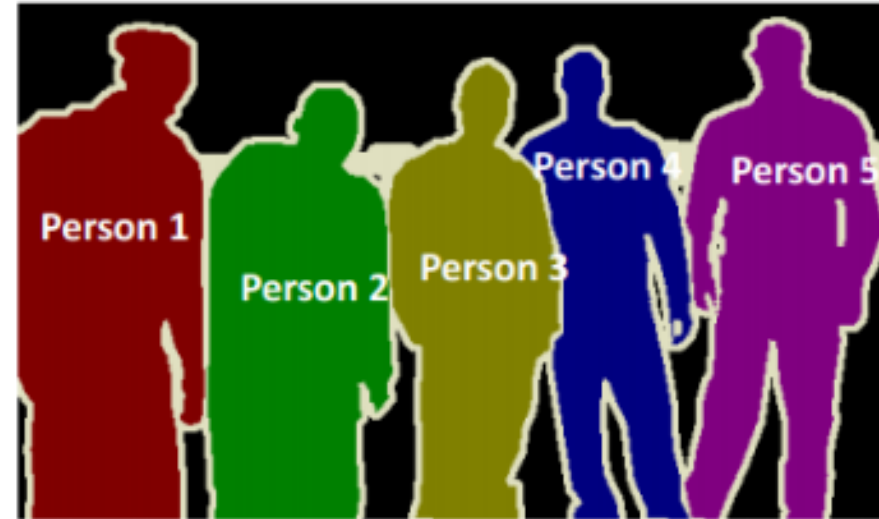
# Các dạng phân vùng ảnh

- Có 2 dạng bài toán phân vùng ảnh:



Semantic Segmentation

• Chúng ta phân đoạn (segment) các vùng ảnh theo những nhãn khác nhau mà không phân biệt sự khác nhau giữa các đối tượng trong từng nhãn. Ví dụ trong hình ảnh bên trái chúng ta phân biệt được pixel nào thuộc về người và pixel nào thuộc về background. Tuy nhiên trong bức ảnh xuất hiện 5 người, mức độ phân chia sẽ không xác định từng pixel thuộc về người nào.



Instance Segmentation

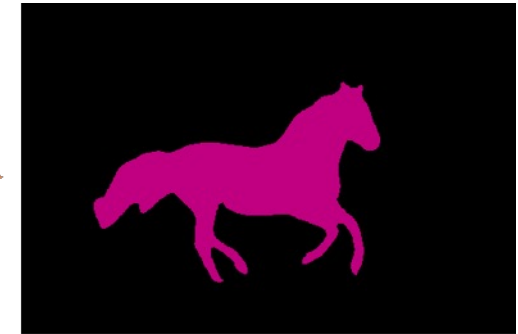
• Chúng ta phân đoạn các vùng ảnh chi tiết đến từng đối tượng trong mỗi nhãn. Ví dụ: ở hình ảnh bên phải đối với nhãn người sẽ được phân chia chi tiết tới từng người 1, 2, ... , 5.



## 2. Ứng dụng của phân vùng ảnh

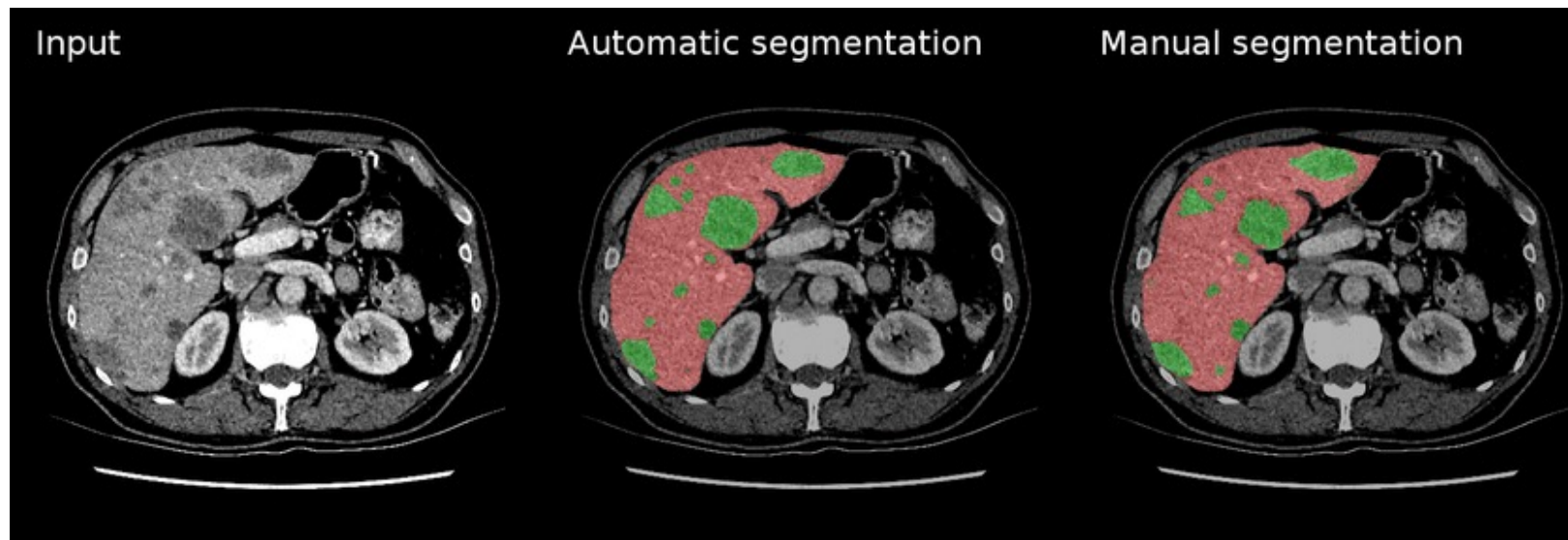
# Ứng dụng của phân vùng ảnh

- **Ứng dụng trong lĩnh vực nhận dạng đối tượng**, quá trình phân vùng ảnh sẽ tách đối tượng ra khỏi vùng nền. Đối tượng ở đây có thể là con người hoặc một vật gì đó chuyển động thuộc vùng tiền cảnh (Fg).
- Đối tượng sau khi được tách ra trong quá trình phân vùng có thể được xử lý trong các hệ thống như đếm số lượng người ra vào, nhận dạng cử chỉ tay, nhận dạng khuôn mặt.
- **Nhận dạng đối tượng:**
  - Phát hiện đi bộ
  - Phát hiện khuôn mặt
  - Phát hiện đèn dừng xe
  - Xác định vị trí đối tượng trong ảnh vệ tinh



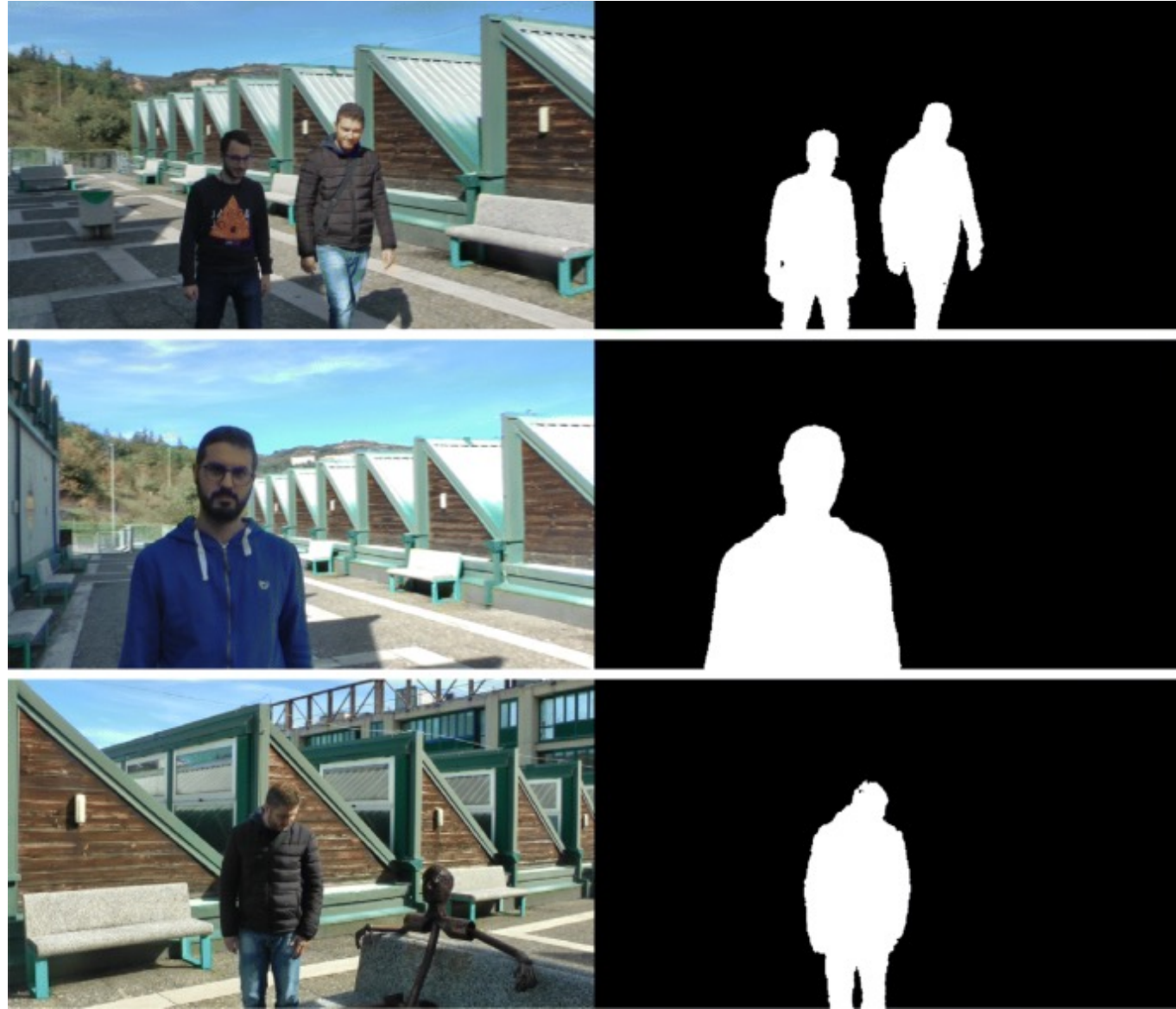
# Ứng dụng của phân vùng ảnh

- **Ứng dụng trong lĩnh vực hình ảnh y tế**, các kỹ thuật hình ảnh y tế như chụp CT (Computer Tomography), chụp MRI (Magnetic Resonance Imaging), chụp X-Quang, USG (Ultrasound) không thể thiếu để có thể phân tích chính xác nhiều bệnh lý khác nhau, qua đó đã hỗ trợ đáng kể bác sĩ trong việc chẩn đoán bệnh.
- Trong quá trình phân tích, người chẩn đoán cần trích xuất các đường biên cần thiết, các bề mặt hoặc các bộ phận cơ thể ra khỏi bức hình, kỹ thuật này được gọi là phân vùng (segmentation). Tuy nhiên, quá trình phân vùng thủ công là rất tốn thời gian và có thể không cho kết quả tốt. Các phân vùng và đường biên này là rất quan trọng đối với các bác sĩ.
- Chính vì vậy, trong vài thập kỷ qua, nhiều thuật toán phân vùng ảnh y tế được đề xuất nhằm tăng độ chính xác trong quá trình phân vùng ảnh.



# Ứng dụng của phân vùng ảnh

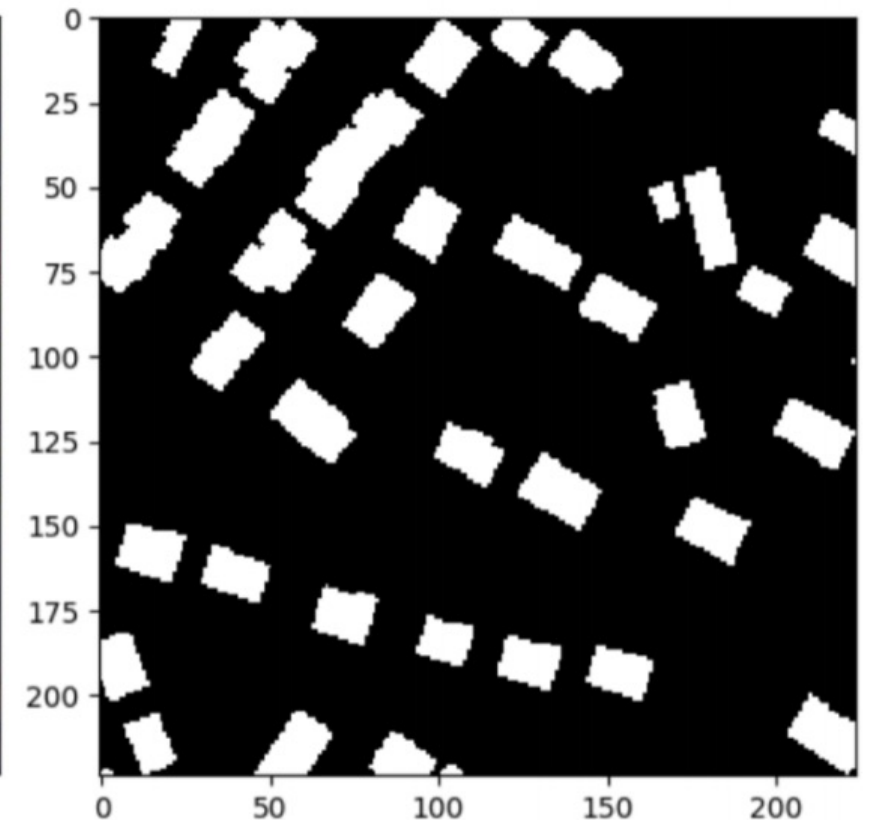
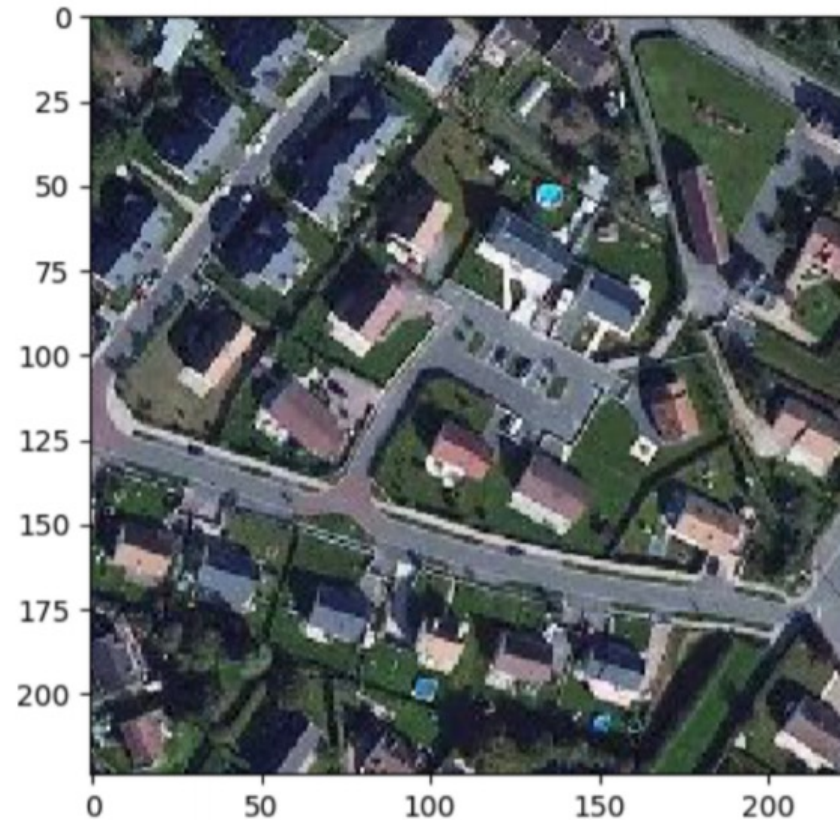
- Ứng dụng trong lĩnh vực camera giám sát, quá trình phân vùng ảnh có thể ứng dụng trong việc xác định, giám sát đối tượng đi vào vùng giám sát, cảnh báo chuyển động khi đối tượng di chuyển vào vùng giám sát.





# Ứng dụng của phân vùng ảnh

- **Ứng dụng trong xử lý ảnh vệ tinh,** Các vệ tinh quay quanh trái đất sẽ liên tục thu thập hình ảnh bề mặt trái đất ở những vùng khác nhau. Từ các bức ảnh vệ tinh, mô hình phân vùng ảnh sẽ phân hình ảnh thành các tuyến đường, khu phố, biển cả, cây cối.





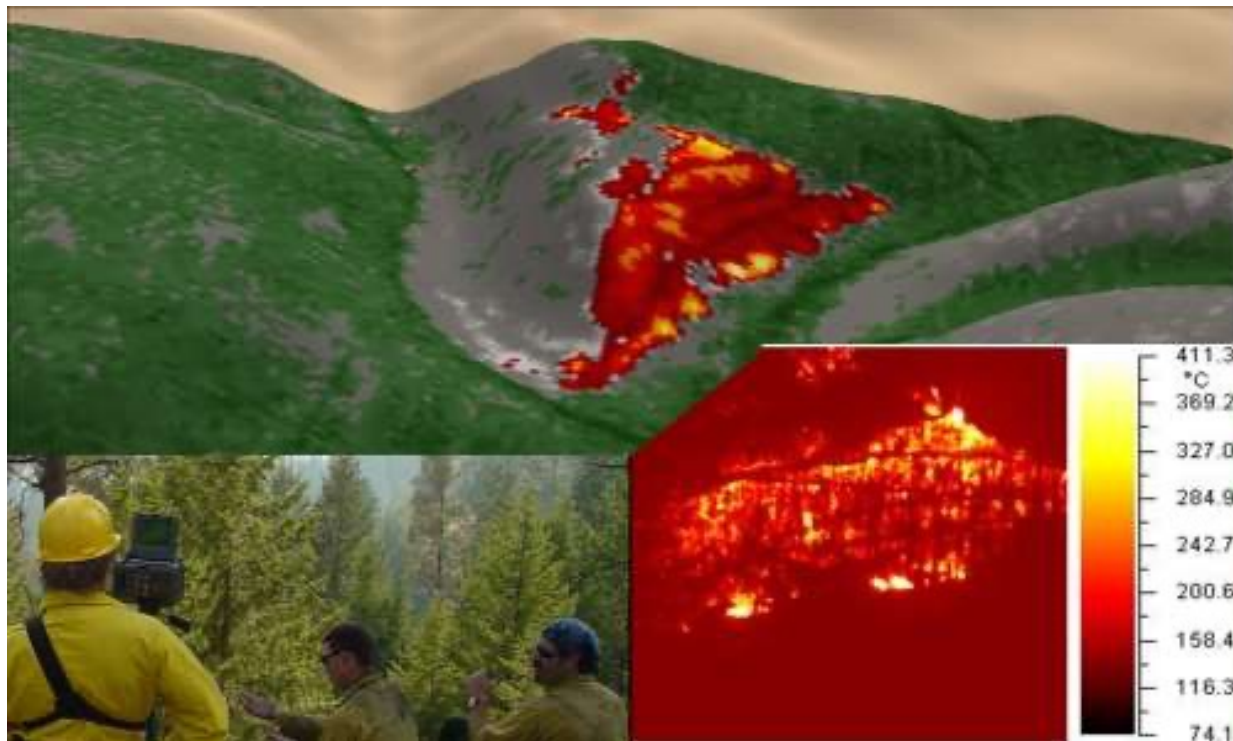
# Ứng dụng của phân vùng ảnh

- **Ứng dụng trong nông nghiệp**, Chúng ta có thể tiết kiệm được một lượng lớn thuốc trừ sâu, nước tưới trong nông nghiệp nhờ sử dụng hệ thống phun thuốc trừ sâu, tưới nước tự động có khả năng phân biệt được diện tích cỏ và cây trồng, đất trống dựa trên thuật toán Image Segmentation.



# Ứng dụng của phân vùng ảnh

- Ứng dụng trong cảnh báo cháy rừng**, những hệ thống cảnh báo cháy rừng có thể phân vùng được chính xác vị trí phát sinh đám cháy từ ảnh chụp vệ tinh. Từ đó đưa ra cảnh báo về qui mô và mức độ lây lan của các đám cháy trên diện rộng.





# Các cách tiếp cận phân vùng ảnh

- **Cách tiếp cận tương đồng (Similarity approach)**, có nghĩa là phát hiện sự tương đồng giữa các pixel hình ảnh để tạo thành một phân đoạn, dựa trên một ngưỡng. Các thuật toán học máy như phân cụm thường dựa trên kiểu tiếp cận này để phân vùng một hình ảnh.
- **Cách tiếp cận gián đoạn (Discontinuity approach)**: Cách tiếp cận này dựa trên sự gián đoạn của các giá trị cường độ pixel trong hình ảnh. Các kỹ thuật phát hiện đường, điểm và cạnh sử dụng kiểu tiếp cận gián đoạn để thu được các kết quả phân vùng trung gian. Kết quả này sau đó có thể được xử lý để cho ra hình ảnh được phân vùng cuối cùng.

### 3. Phương pháp phân vùng ảnh

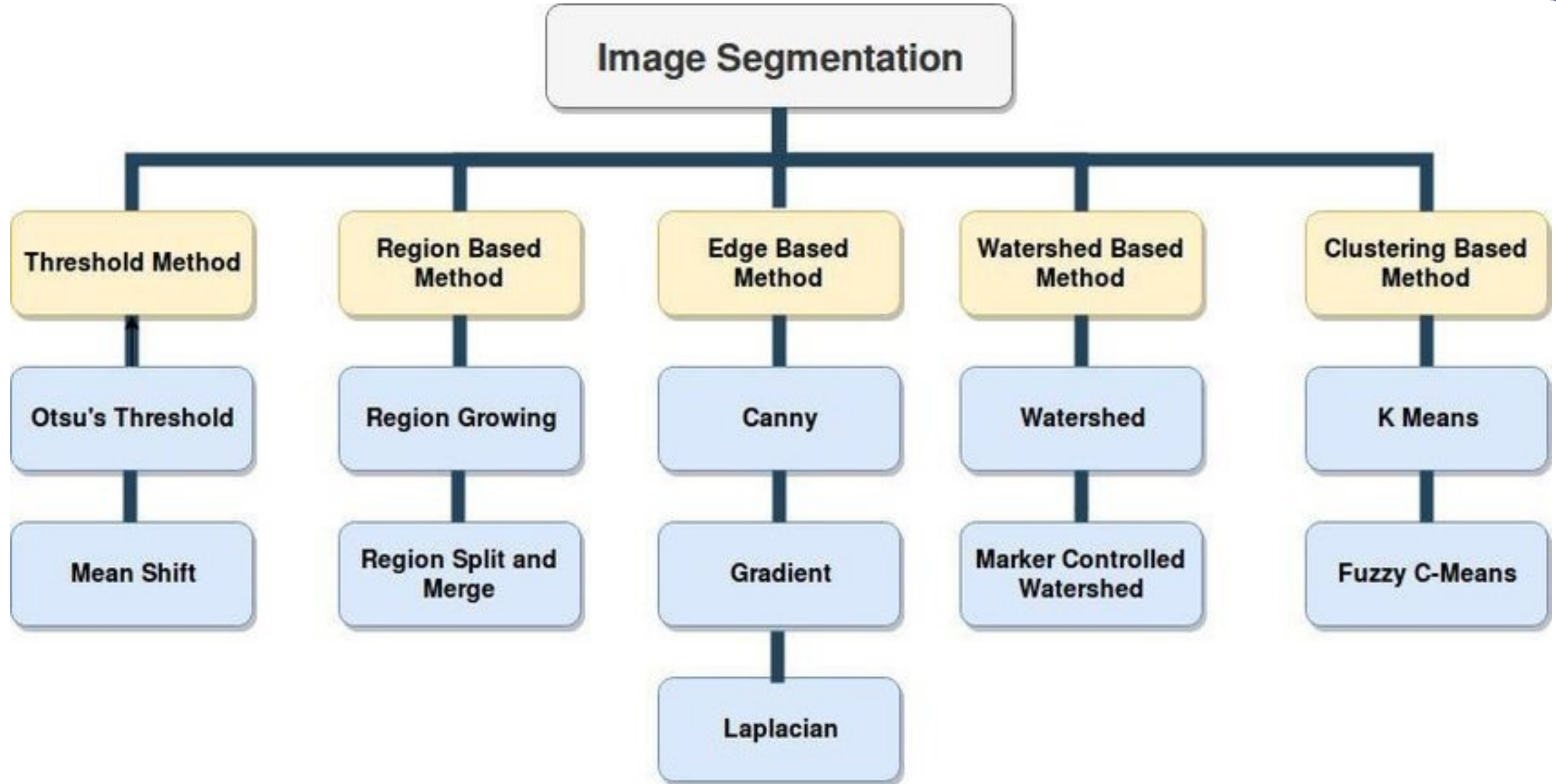
# Phương pháp phân vùng ảnh

**Có nhiều phương pháp phân vùng ảnh, một số phương pháp phổ biến bao gồm:**

1. Phân vùng dựa trên ngưỡng (Threshold Based Method)
2. Phân vùng dựa trên khu vực (Region Based Method)
3. Phân vùng dựa trên cạnh (Edge Based Method)
4. phân vùng dựa trên lưu vực (Watershed Based Method)
5. Phân vùng dựa trên kỹ thuật phân cụm (Clustering Based Method)
6. Phân vùng dựa trên mạng neural nhân tạo (Artificial Neural Network Based Method)



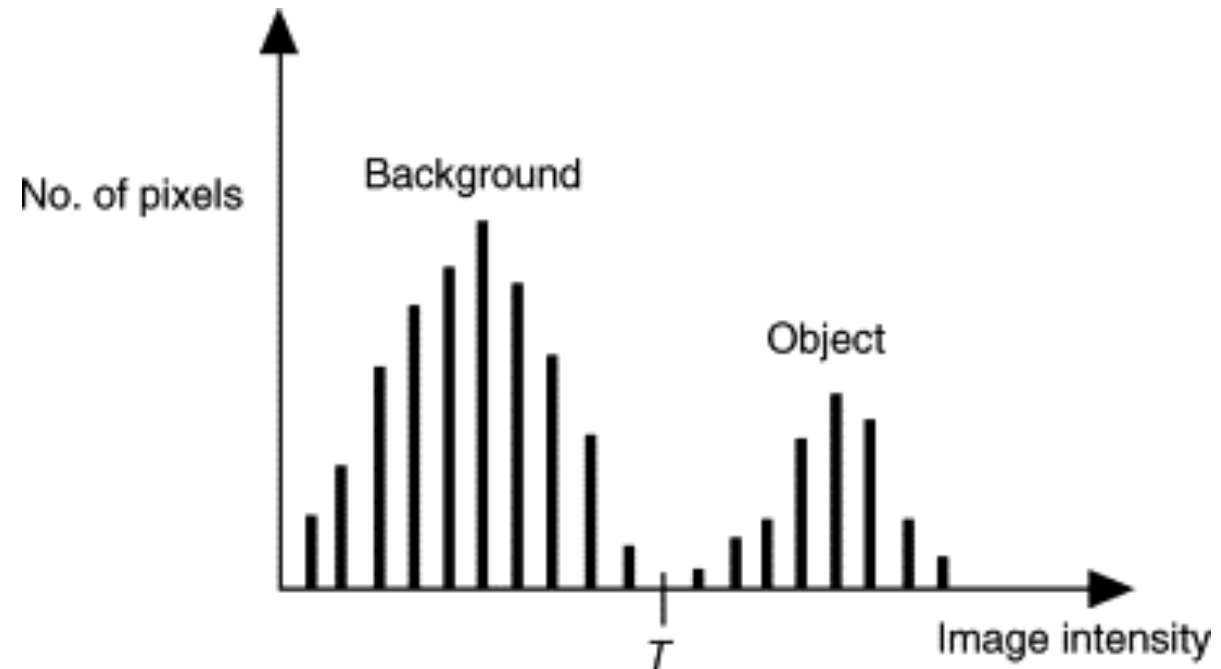
# Phương pháp phân vùng ảnh



### **3.1. Phân vùng ảnh dựa trên ngưỡng (Threshold Based Method)**

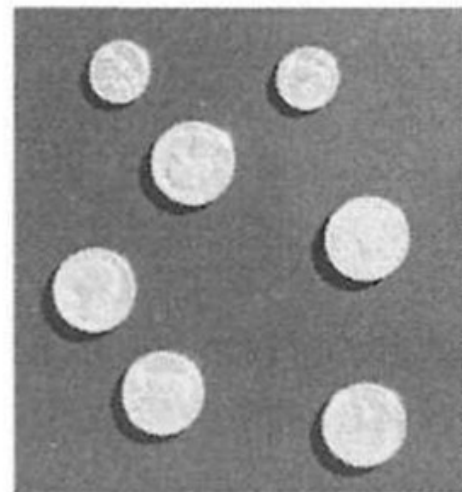
# Phân vùng ảnh dựa trên ngưỡng

- Đây là phương pháp phân vùng đơn giản, giúp tạo ra một hình ảnh nhị phân hoặc nhiều màu dựa trên việc đặt giá trị ngưỡng theo cường độ pixel của ảnh gốc.
- Đối với một ảnh có nền và đối tượng, có thể chia ảnh thành các vùng dựa trên cường độ của đối tượng và nền. Nhưng ngưỡng này phải được thiết lập hoàn hảo để phân đoạn hình ảnh thành một đối tượng và một nền.

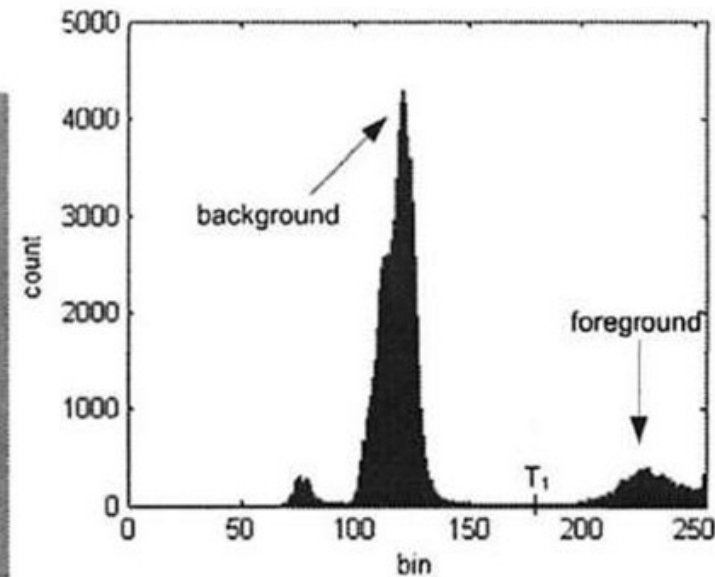


# Phân vùng ảnh dựa trên ngưỡng

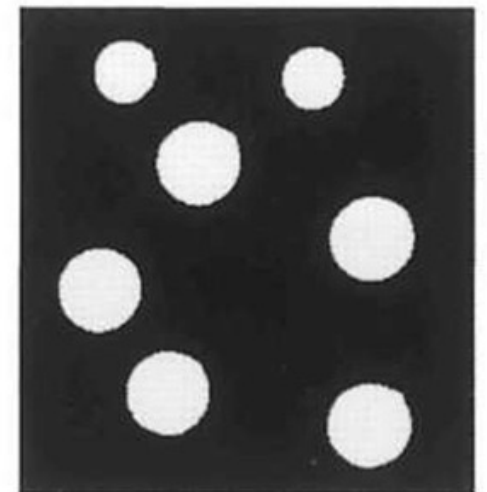
- Phân ngưỡng bao gồm các kỹ thuật như:
  - ngưỡng toàn cục (Global thresholding);
  - ngưỡng thủ công (Manual thresholding);
  - ngưỡng thích ứng (Adaptive Thresholding);
  - ngưỡng tối ưu (Optimal Thresholding);
  - ngưỡng thích ứng cục bộ (Local Adaptive Thresholding).



(a)



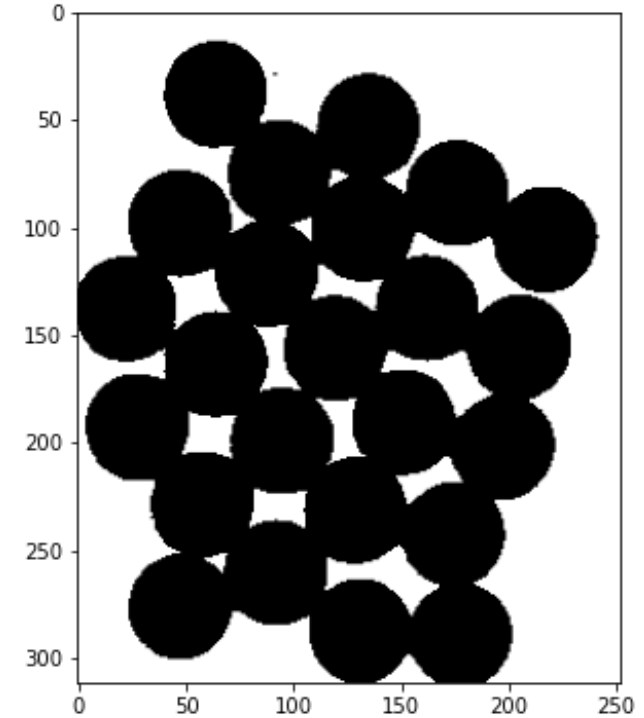
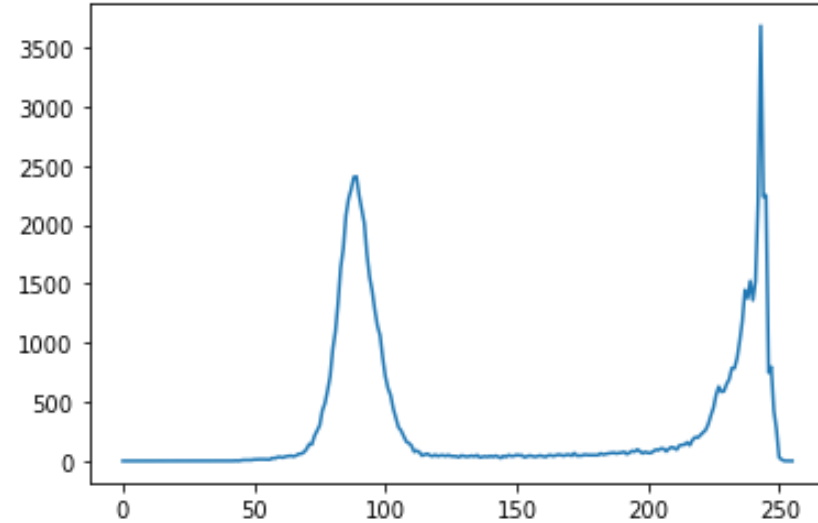
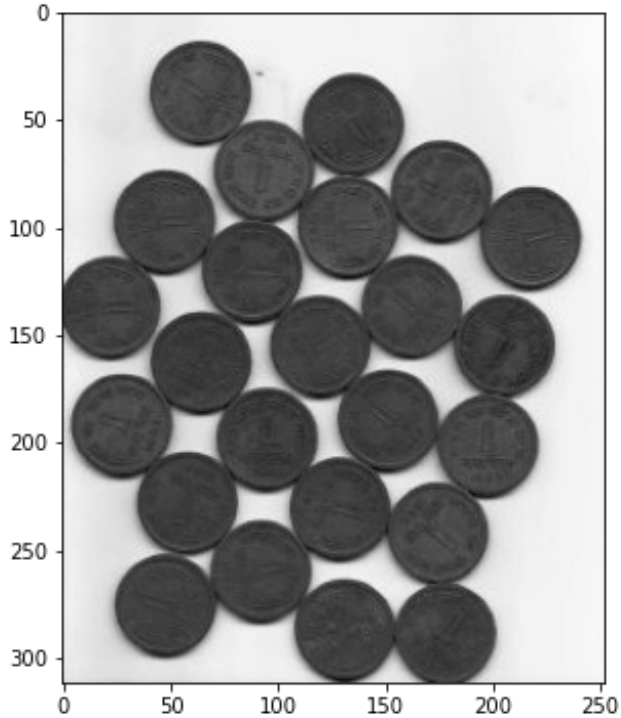
(b)



(c)

# Ngưỡng toàn cục (Global thresholding)

- Sử dụng một ngưỡng T chung cho tất cả các pixel trong một ảnh.

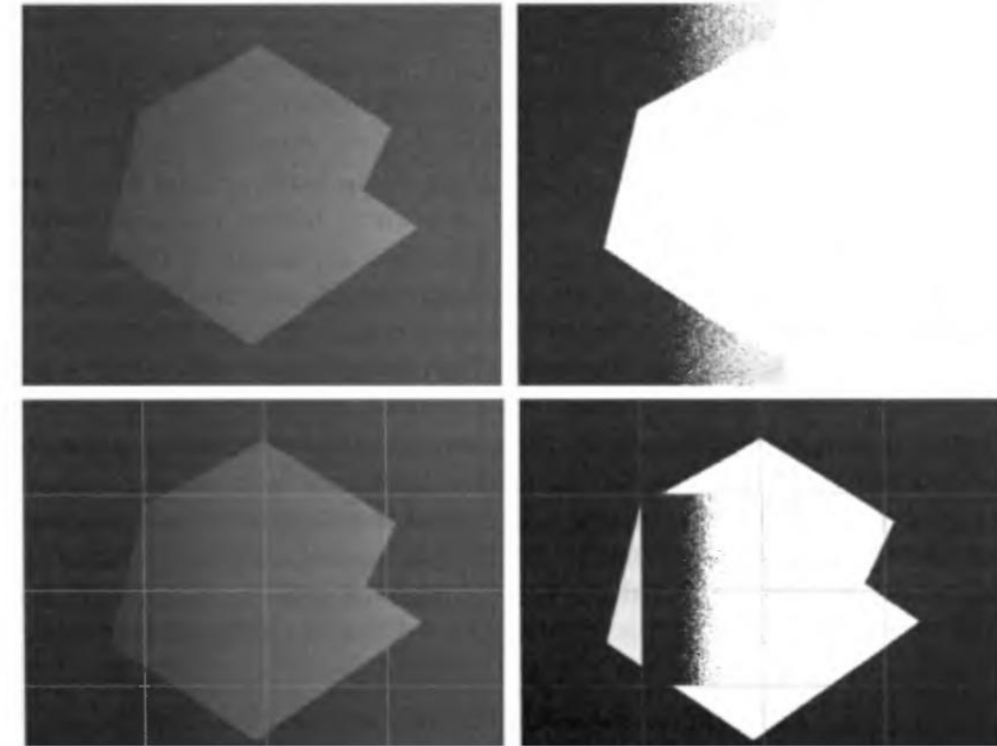


```
1 #Apply a threshold:
2 (T, thresh) = cv2.threshold(img_orignal, 170, 255, cv2.THRESH_BINARY)
3
4 plt.figure(figsize=(8,6))
5 plt.imshow(thresh,cmap='gray')
6 plt.show()
```



# Ngưỡng thích ứng (Adaptive Thresholding)

- Phương pháp phân ngưỡng toàn cục không phù hợp cho nhiều trường hợp như là với trường hợp ánh sáng không đồng đều trên ảnh. Khi đó phân ngưỡng thích là một giải pháp.
- OpenCV cung cấp phương thức `adaptiveThreshold` để thực hiện phân ngưỡng thích nghi. Phương thức này sẽ tính giá trị trung bình của  $n$  điểm ảnh xung quanh điểm ảnh đó rồi trừ đi  $C$  ( $n$  là thường là số lẻ,  $C$  là một số nguyên bất kỳ)



# Ngưỡng thích ứng (Adaptive Thresholding)

Original Image



Global Thresholding



Adaptive Thresholding



# Ngưỡng thích ứng (Adaptive Thresholding)

**pic = cv2.adaptiveThreshold(img, maxValue, thresholdType, type, n, C)**

Trong đó:

- img: ảnh gốc
- maxValue: giá thiết lập nếu > giá trị ngưỡng
- adaptiveMethod: phương pháp tính cho các điểm lân cận lấy trung bình hay theo Gaussian
  - ADAPTIVE\_THRESH\_MEAN\_C – threshold value is the mean of neighborhood area.
  - ADAPTIVE\_THRESH\_GAUSSIAN\_C – threshold value is the weighted sum of neighborhood values where weights are a Gaussian window.
- type: Loại xử lý cắt ngưỡng
  - THRESH\_BINARY: Có thể dịch là ngưỡng nhị phân. Ý nghĩa y hệt những gì mình đề cập ở trên.
  - THRESH\_BINARY\_INV: Ngưỡng nhị phân đảo ngược. Có thể hiểu là nó sẽ đảo ngược lại kết quả của THRESH\_BINARY.
- n: Số lượng điểm ảnh xung quanh sử dụng để tính giá trị cho điểm đang xét, là số lẻ.
- C: Giá trị số bất kỳ được sử dụng để trừ.

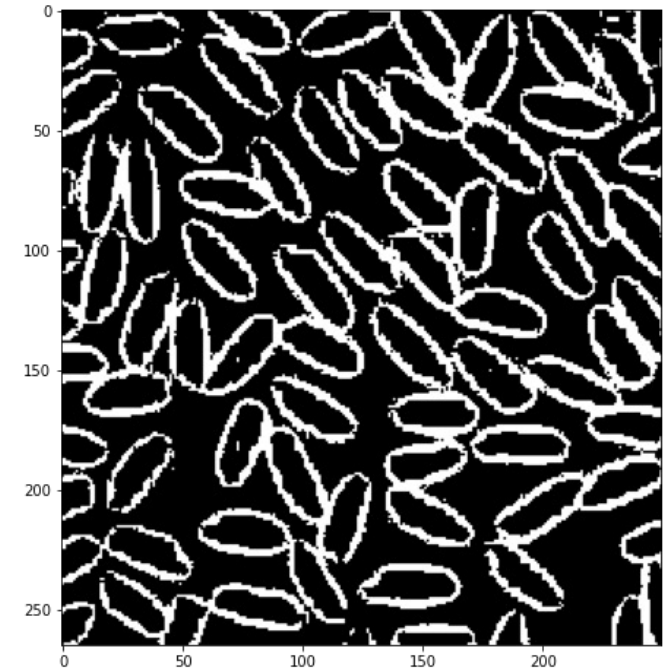
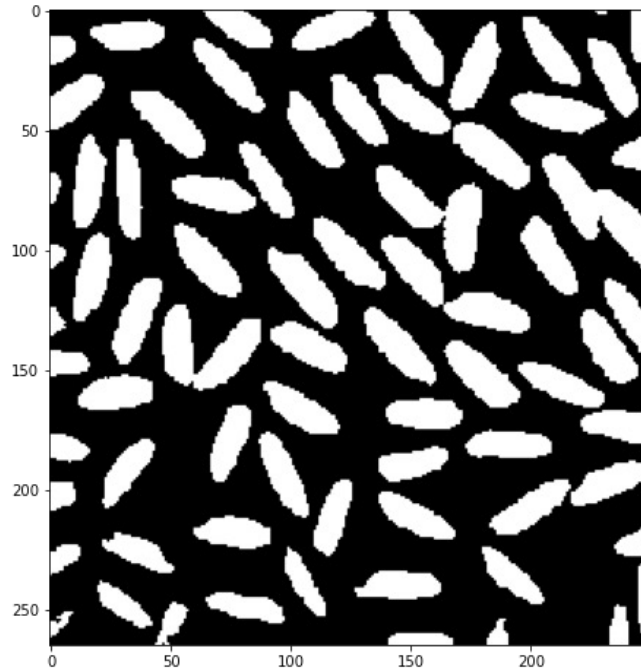
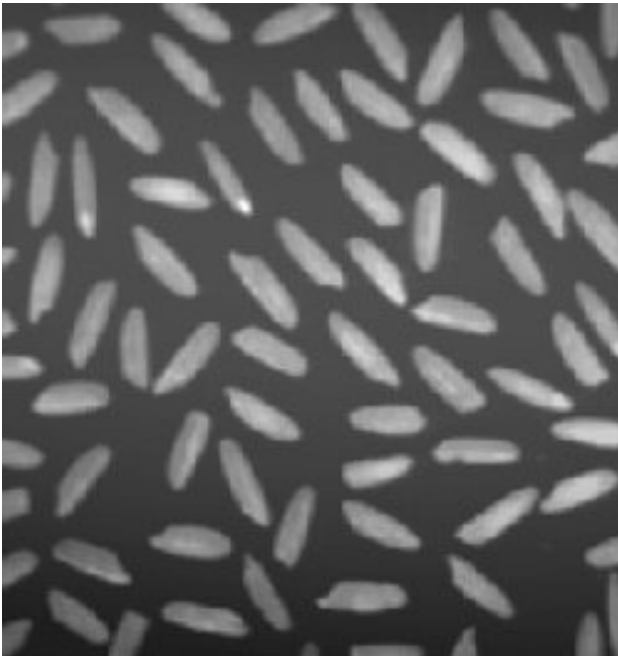


# Thực hành 4.1



# Thực hành

1. Đọc và hiển thị ảnh Thuchanh4\_1.jpeg ở dạng ảnh xám
2. Phân vùng ảnh sử dụng phương pháp phân ngưỡng toàn cục (Hãy xác định ngưỡng phù hợp để đạt kết quả tốt nhất)
3. Phân vùng ảnh sử dụng phương pháp phân ngưỡng thích nghi (Hãy xác định các tham số phù hợp để đạt kết quả tốt nhất)



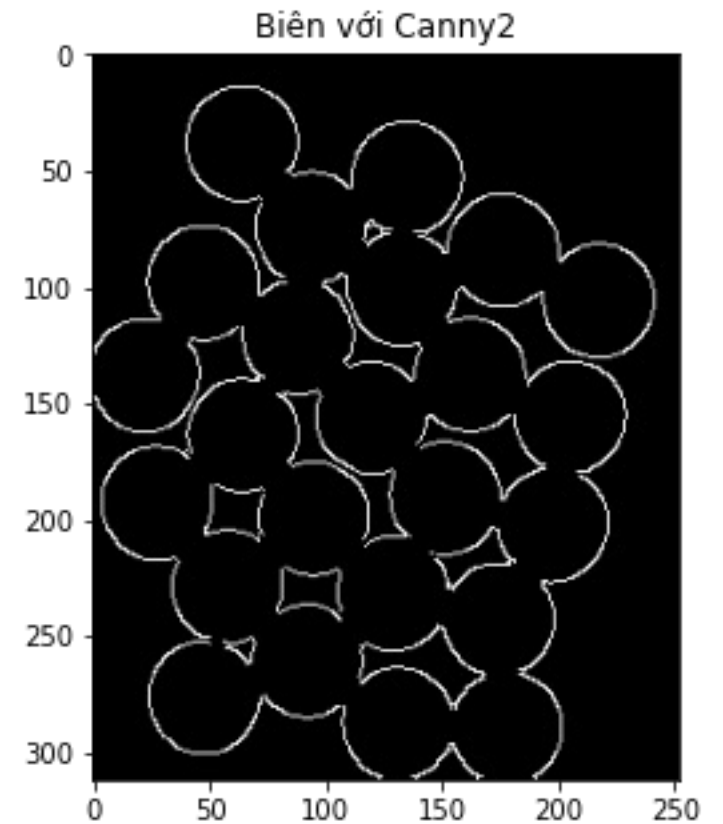
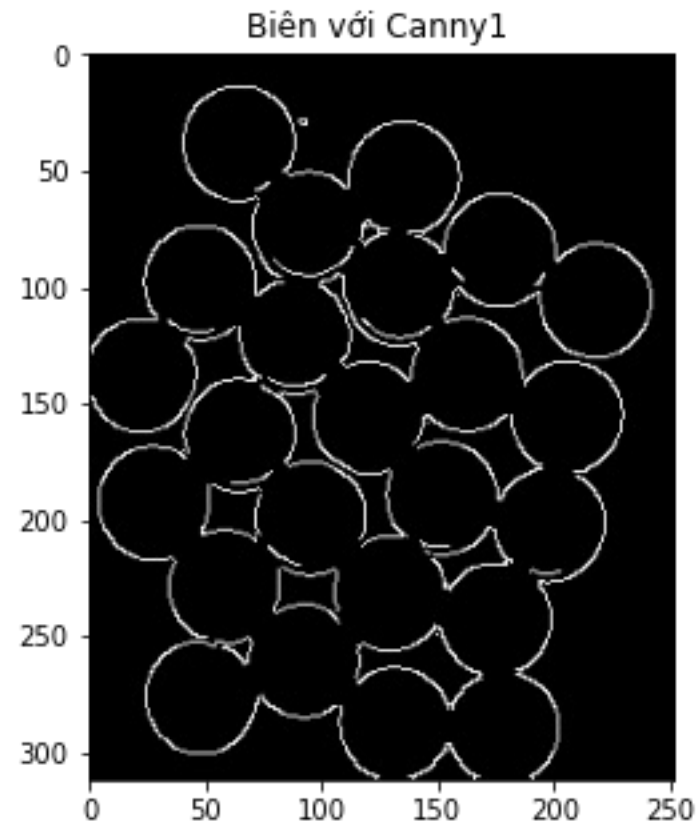
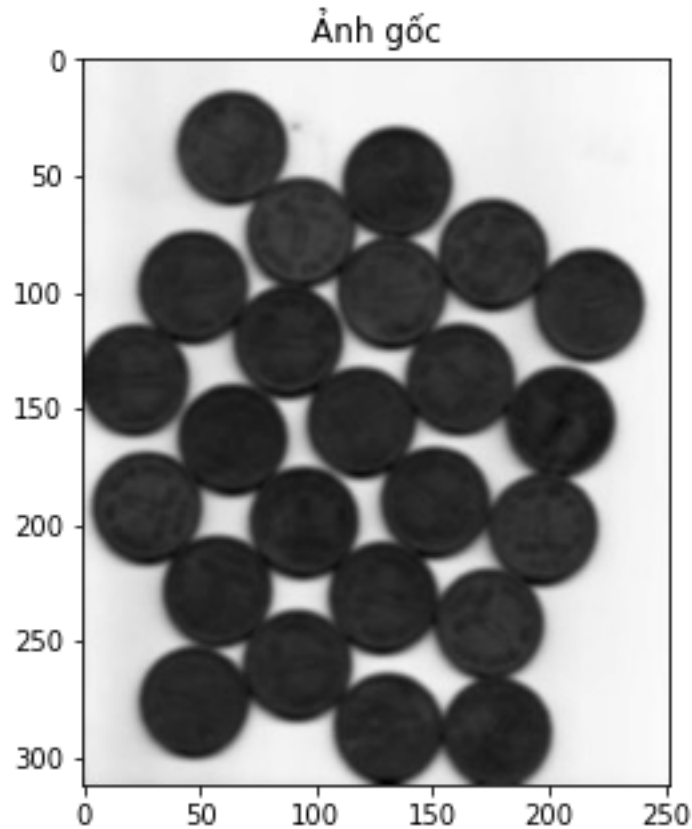
## 3.2. Phân vùng ảnh dựa trên cạnh, biên (Edge Based Method)

# Phân vùng ảnh dựa trên cạnh, biên

- Cạnh trong ảnh đánh dấu những vị trí hình ảnh không liên tục về mức xám, màu sắc, kết cấu, v.v. Khi di chuyển từ vùng này sang vùng khác, mức xám có thể thay đổi. Vì vậy, nếu tìm thấy sự gián đoạn đó, ta có thể tìm thấy cạnh. Thực tế, có nhiều toán tử phát hiện cạnh, nhưng hình ảnh thu được là kết quả phân vùng trung gian, và không nên nhầm lẫn với hình ảnh được phân vùng cuối cùng.
- Để ra được kết quả cuối, cần thực hiện một số bước bổ sung bao gồm: kết hợp các phân vùng cạnh thu được làm một, để giảm số lượng phân vùng và có được một đường viền liên mạch của đối tượng.
- Như vậy, có thể thấy, phân vùng cạnh đưa ra một kết quả phân vùng trung gian. Kết quả này sau đó có thể áp dụng theo vùng hoặc bất kỳ kiểu phân đoạn nào khác, nhằm có được hình ảnh được phân vùng cuối.

# Phân vùng ảnh dựa trên cạnh, biên

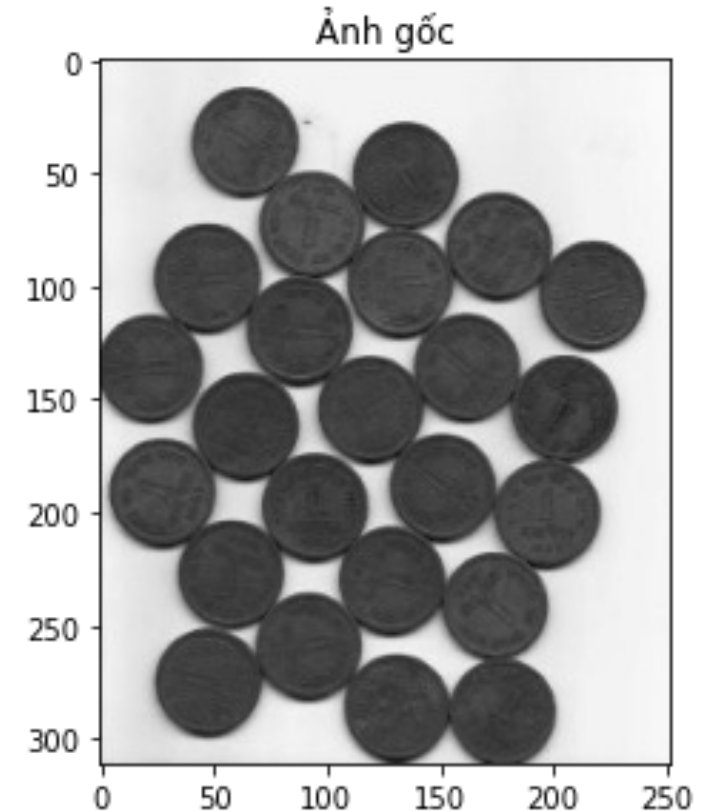
- Sử dụng Canny để phân vùng:



# Phân vùng ảnh dựa trên cạnh, biên

- Sử dụng các phương thức OpenCV cung cấp thực hiện đếm số lượng đồng xu trong ảnh

```
1  #Bước 1: Đọc ảnh ở chế độ ảnh màu:  
2  image = cv2.imread('images/pic_money.jpeg')  
3  
4  #Hiển thị ảnh gốc ở dạng xám:  
5  gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  
6  
7  plt.figure(figsize=(10,10))  
8  plt.subplot(2,2,1)  
9  plt.imshow(gray, cmap='gray');  
10 plt.title('Ảnh gốc')
```

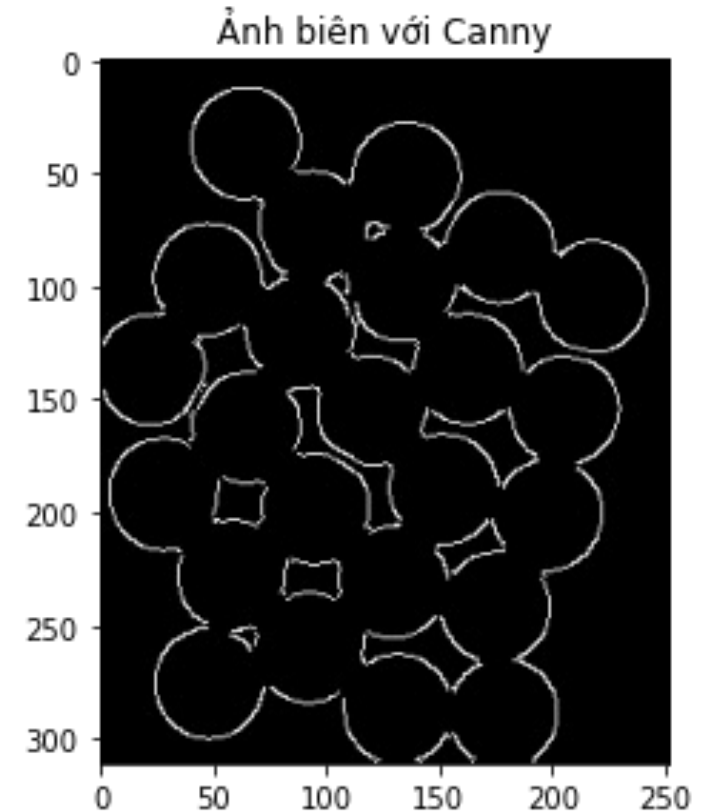




# Phân vùng ảnh dựa trên cạnh, biên

- Sử dụng các phương thức OpenCV cung cấp thực hiện đếm số lượng đồng xu trong ảnh

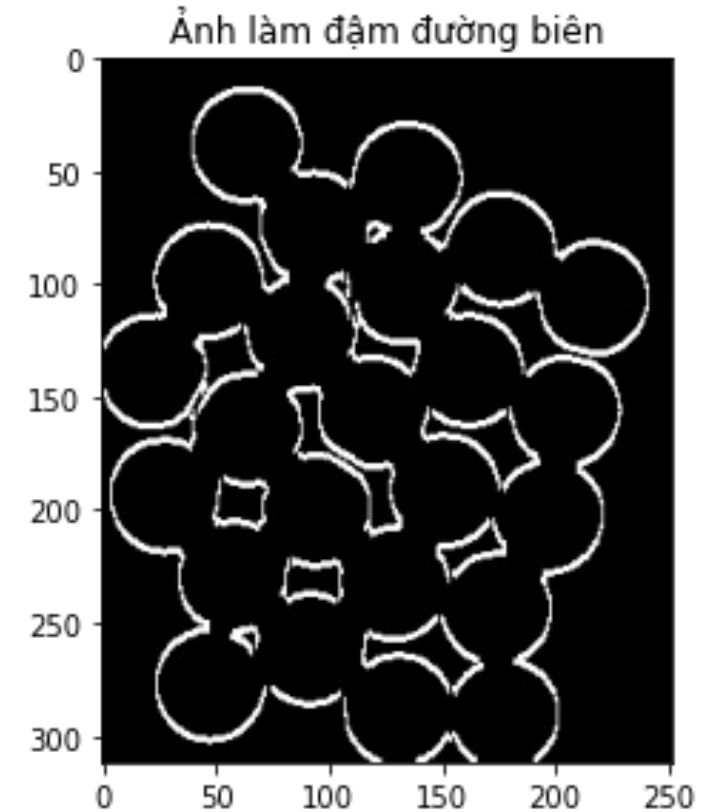
```
12 #Bước 2: Phát hiện biên với Canny:  
13 #Thực hiện làm mịn ảnh với Gaussian:  
14 blur = cv2.GaussianBlur(gray, (11,11), 0)  
15  
16 #Phát hiện biên với Canny  
17 canny = cv2.Canny(blur, 30, 150, 3)  
18 plt.subplot(2,2,2)  
19 plt.imshow(canny, cmap='gray')  
20 plt.title('Ảnh biên với Canny')
```



# Phân vùng ảnh dựa trên cạnh, biên

- Sử dụng các phương thức OpenCV cung cấp thực hiện đếm số lượng đồng xu trong ảnh

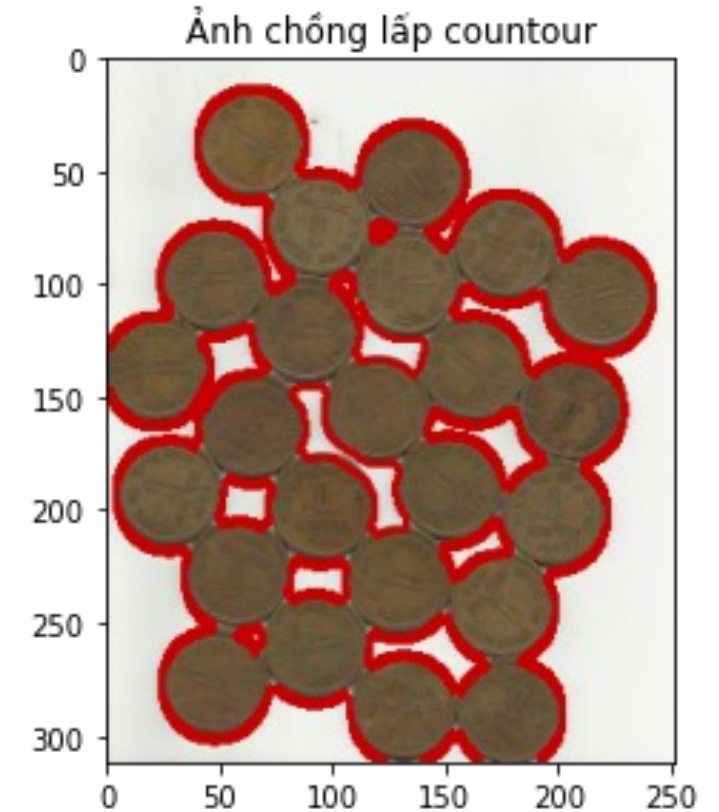
```
23 #Bước 3: Làm đậm đường biên:  
24 dilated = cv2.dilate(canny, (1,1), iterations = 2)  
25 plt.subplot(2,2,3)  
26 plt.imshow(dilated, cmap='gray')  
27 plt.title('Ảnh làm đậm đường biên')
```



# Phân vùng ảnh dựa trên cạnh, biên

- Sử dụng các phương thức OpenCV cung cấp thực hiện đếm số lượng đồng xu trong ảnh

```
29 #Bước 4: Lấy countour của ảnh và đếm số lượng:
30 (cnt, heirarchy) = cv2.findContours(dilated.copy(),
31                                     cv2.RETR_EXTERNAL,
32                                     cv2.CHAIN_APPROX_NONE)
33
34 rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
35 cv2.drawContours(rgb, cnt, -1, (200,0,0), 2)
36
37 plt.subplot(2,2,4)
38 plt.imshow(rgb)
39 plt.title('Ảnh chồng lấp countour')
40 plt.show()
41
42 print('Số lượng đồng xu trong ảnh là: ', len(cnt))
```



Số lượng đồng xu trong ảnh là: 24

## Thực hành 4.2

1. Đọc và hiển thị ảnh Thuchanh4\_2.jpeg ở dạng ảnh màu.
2. Sử dụng phương pháp Canny với các tham số phù hợp phát hiện biên của các đối tượng trong ảnh.
3. Đếm số lượng bóng bay trong bức ảnh với OpenCV





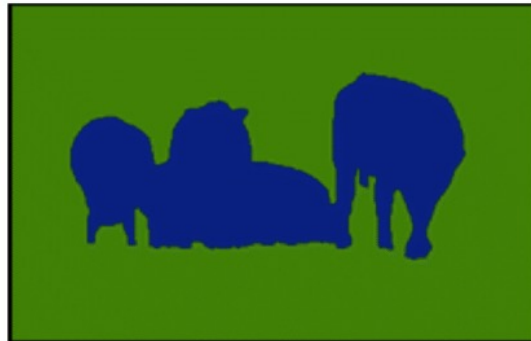
### 3.3. Phân vùng ảnh dựa trên kỹ thuật phân cụm

# Phân vùng ảnh dựa trên kỹ thuật phân cụm

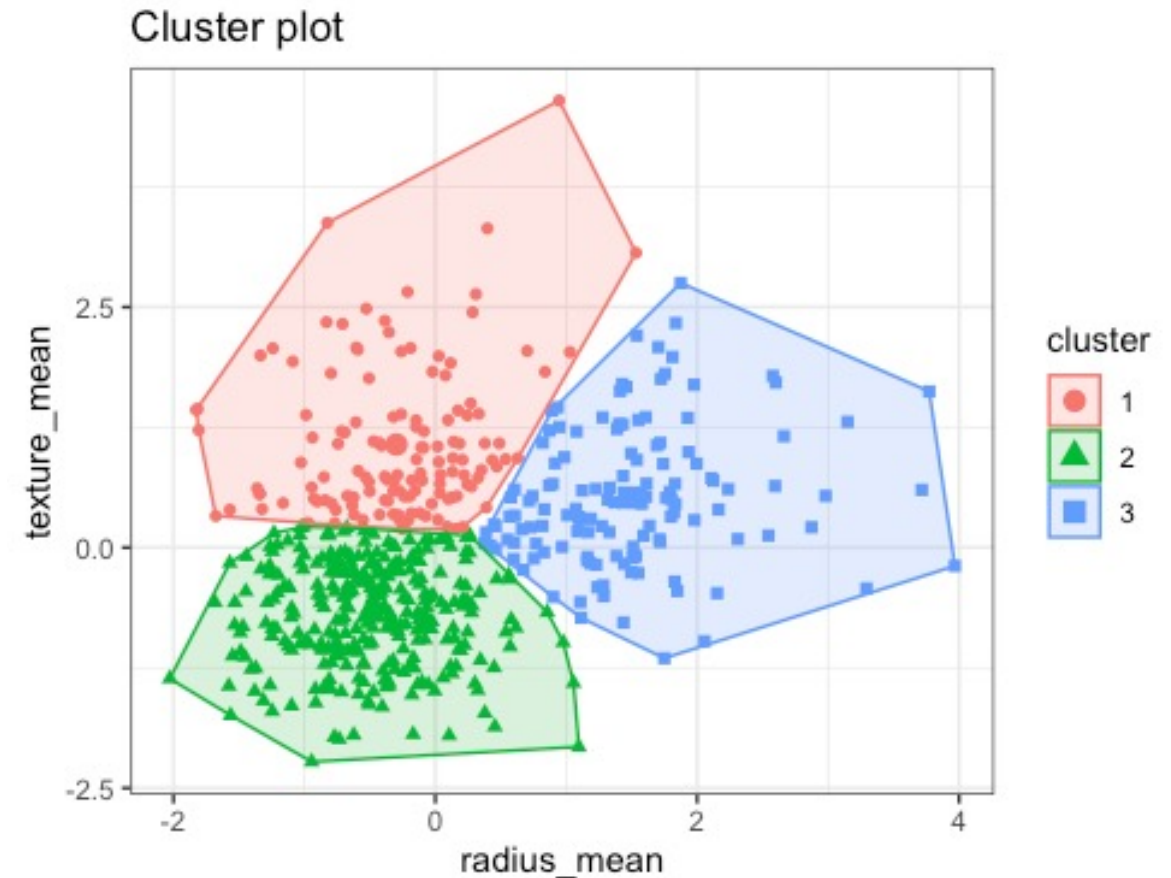
- Phân cụm (Clustering) là một loại thuật toán học máy không giám sát, được sử dụng phổ biến trong phân vùng ảnh.
- Một trong những thuật toán Clustering thường được ứng dụng cho tác vụ phân vùng ảnh là KMeans Clustering. Loại phân cụm này có thể được sử dụng để tạo các phân đoạn trong một hình ảnh có màu.



a



b



# Phân vùng ảnh dựa trên kỹ thuật phân cụm

- OpenCV cung cấp phương thức kmeans để thực hiện phân cụm

**ret,label,center=cv2.kmeans(samples, K, criteria, attempts, flags)**

## Tham số đầu vào:

- 1.samples : Mẫu thì nên là kiểu dữ liệu của np.float32, và mỗi đối tượng cần được đặt trong cột duy nhất.
- 2.nclusters (K): Số nhóm yêu cầu tại thời điểm cuối cùng
- 3.criteria : Đây là tiêu chí chấm dứt vòng lặp. Khi tiêu chí này được thỏa mãn, thuật toán sẽ dừng vòng lặp. Trên thực tế, nó phải là một nhóm 3 thông số (type, max\_iter, epsilon):
  1. 3.a - loại tiêu chí chấm dứt: Nó có 3 lá cờ như sau:
    1. cv2.TERM\_CRITERIA\_EPS - ngăn sự lặp lại thuật toán nếu đạt được độ chính xác nhất định - epsilon đạt được.
    2. cv2.TERM\_CRITERIA\_MAX\_ITER - dừng thuật toán sau khi số lượng nhất định được lặp đi lặp lại.
    3. cv2.TERM\_CRITERIA\_EPS max\_iter + cv2.TERM\_CRITERIA\_MAX\_ITER - ngăn chặn sự lặp đi lặp lại khi một trong các điều kiện trên nó được đáp ứng
  2. 3.b - max\_iter - Một số nguyên xác định số lượng tối đa vòng lặp.
  3. 3.c - Độ chính xác - epsilon
- 4.attempts: Cờ để xác định số lần các thuật toán được thực hiện bằng cách sử dụng việc đánh nhãn khởi tạo khác nhau.
- 5.flags : Cờ này được sử dụng để xác định trung tâm ban đầu được chọn như thế nào. Về cơ bản có 2 cờ được sử dụng là : cv2.KMEANS\_PP\_CENTERS và cv2.KMEANS\_RANDOM\_CENTERS.

## Tham số đầu ra:

- 1.compactness : Đây là tổng của bình phương khoảng cách từ mỗi điểm đến trọng tâm tương tự của họ.
- 2.labels : Đây là mảng các label trong đó mỗi phần tử của mảng được đánh dấu '0', '1' .....
- 3.centers : Đây là mảng trọng tâm của các nhóm.

# Phân vùng ảnh dựa trên kỹ thuật phân cụm

- Bước 1: Đọc ảnh màu cần thực hiện phân vùng

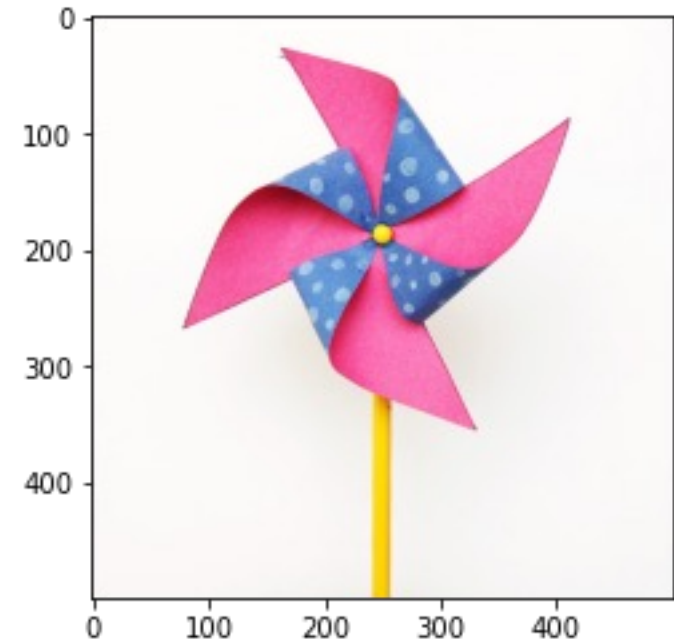
```
1 # Đọc ảnh màu
2 img = cv2.imread('images/pic2.jpeg')
3
4 #chuyển sang hệ màu RGB
5 img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
6 plt.imshow(img)
7 plt.show()
```

- Bước 2: Biến đổi dữ liệu

```
1 #Chuyển đổi mảng 3 chiều về ma trận, mỗi cột ứng với một kênh màu
2 twoDimage = img.reshape((-1,3))
3 twoDimage = np.float32(twoDimage)
```

```
1 twoDimage
```

```
array([[253., 252., 250.],
       [253., 252., 250.],
       [253., 252., 250.],
       ...,
       [246., 246., 246.],
       [246., 246., 246.],
       [246., 246., 246.]], dtype=float32)
```

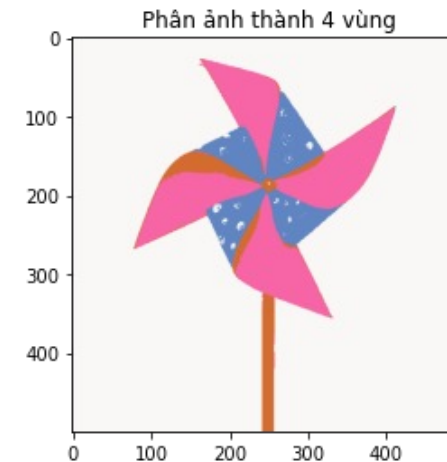
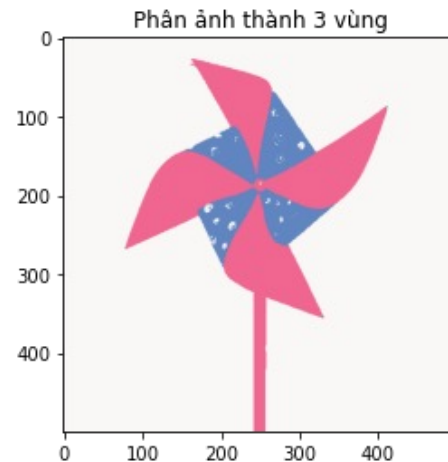
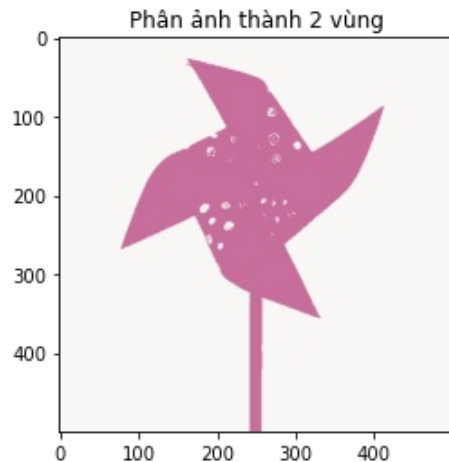


# Phân vùng ảnh dựa trên kỹ thuật phân cụm

- Bước 3: Thiết lập các tham số và thực hiện phân vùng ảnh với Kmeans

```
1 #Thiết lập các tham số của cụm
2 criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 1.0)
3 K = 2
4 attempts=50
```

```
1 #Sử dụng phương thức cv2.kmeans để phân cụm
2 ret,label,center=cv2.kmeans(twoDimImage,
3                             K,
4                             None,
5                             criteria,
6                             attempts,
7                             cv2.KMEANS_PP_CENTERS)
8 center = np.uint8(center)
9 res = center[label.flatten()]
10 result_image = res.reshape((img.shape))
```

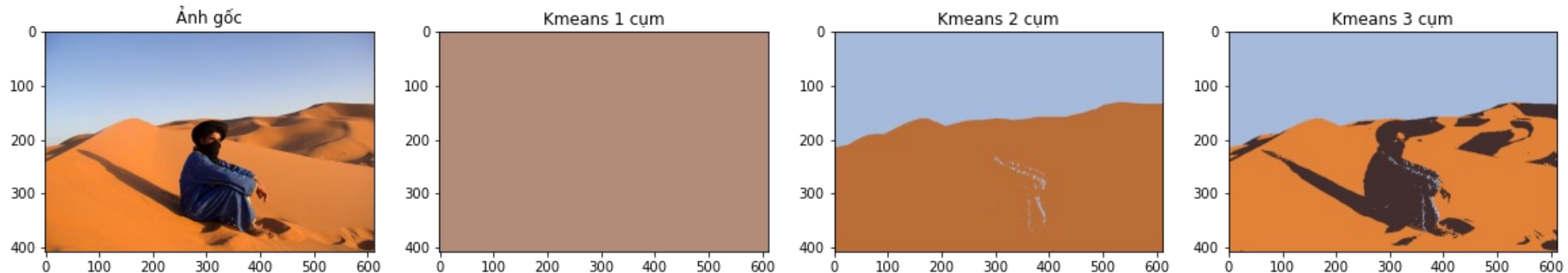




# Thực hành 4.3

# Thực hành

1. Đọc và hiển thị ảnh Thuchanh4\_3.jpeg ở dạng ảnh màu.
2. Sử dụng phương pháp phân vùng ảnh dựa trên kỹ thuật phân cụm Kmeans để phân ảnh ra thành:
  - \* 1 cụm
  - \* 2 cụm
  - \* 3 cụm
- \* Hiển thị kết quả: ảnh 1 cụm - ảnh 2 cụm - ảnh 3 cụm





QUESTIONS

**Q & A**

ANSWERS