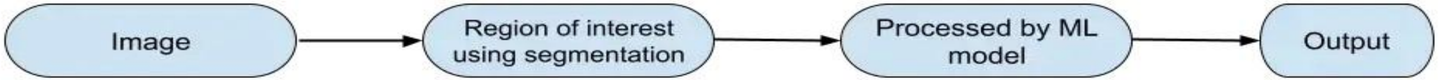
SEGMENTATION

Content

- What is Image Segmentation?
- Approaches of Segmentation
- Types of Image Segmentation
- Image Segmentation Techniques
- Image Segmentation using OpenCV-Python

What is Image Segmentation?

Image segmentation is a technique where a computerized picture is separated into different subgroups called segments which help in decreasing the intricacy of the picture to make further handling or investigation of the picture less difficult. Segmentation in simple words is allocating names to pixels. All image components or pixels having a place with a similar class have a typical mark allocated to them. Image segmentation can serve as a preprocessing step before applying a machine learning algorithm in order to reduce the time complexity required by the machine learning algorithm to process the image. For instance: The above-given image of a flower is an example of image segmentation using clustering where the colors of the image are segmented. This image then can be processed by any machine learning algorithm by only providing the region of interest, thereby reducing the time complexity of the algorithm.



Approaches of Image Segmentation

Similarity approach: This approach is based on detecting similarity between image pixels to form a segment, based on a threshold. ML algorithms like clustering are based on this type of approach to segment an image.

Discontinuity approach: This approach relies on the discontinuity of pixel intensity values of the image. Line, Point, and Edge Detection techniques use this type of approach for obtaining intermediate segmentation results which can be later processed to obtain the final segmented image.

Image Segmentation Techniques

- Threshold Based Segmentation
- Edge Based Segmentation
- Region-Based Segmentation
- Clustering Based Segmentation

Threshold Based Segmentation

Image thresholding segmentation is a simple form of image segmentation. It is a way to create a binary or multi-color image based on setting a threshold value on the pixel intensity of the original image.

In this thresholding process, we will consider the intensity histogram of all the pixels in the image. Then we will set a threshold to divide the image into sections. For example, considering image pixels ranging from 0 to 255, we set a threshold of 60. So all the pixels with values less than or equal to 60 will be provided with a value of 0(black) and all the pixels with a value greater than 60 will be provided with a value of 255(white).

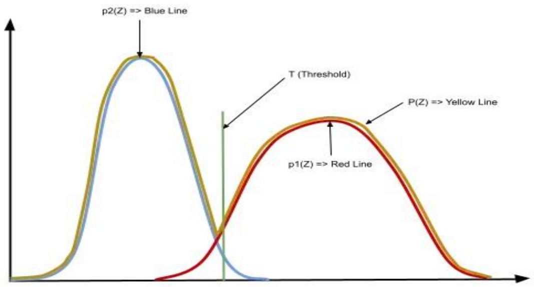
Considering an image with a background and an object, we can divide an image into regions based on the intensity of the object and the background. But this threshold has to be perfectly set to segment an image into an object and a background.

- Various thresholding techniques are :
  - Global thresholding: In this method, we use a bimodal image. A bimodal image is an image with 2 peaks of intensities in the intensity distribution plot. One for the object and one for the background. Then we deduce the threshold value for the entire image and use that global threshold for the whole image. A disadvantage of this type of threshold is that it performs really poorly during poor illumination in the image.
  - Manual thresholding: The following process goes as follows -

a. Choose the initial threshold value.
b. Segment the image into 2 regions G1 and G2 using the threshold value.
c. Find the mean of pixels of region of region G1 and G2 (μ_1 and μ_2 respectively).
d. T = (μ_1 + μ_2) / 2 and then go back to step 2.

Continue the process the $| T_i - T_{i+1} | <= T'$, Where T' is some value defined by the user.

- Adaptive Thresholding: To overcome the effect of illumination, the image is divided into various subregions, and all these regions are segmented using the threshold value calculated for all these regions. Then these subregions are combined to image the complete segmented image. This helps in reducing the effect of illumination to a certain extent.
- Optimal Thresholding: Optimal thresholding technique can be used to minimize the misclassification of pixels performed by segmentation.



Calculation of an optimal threshold using an iterative method by minimizing the misclassification loss of a pixel.
The probability of a pixel value is given by the following formula :

$$P(Z) = P_1 \cdot p_1(Z) + P_2 \cdot p_2(z) \Rightarrow P1 + P2 = 1$$

$$p_1(Z) = PDF\ of\ background\ pixels$$

$$p_2(Z) = PDF\ of\ object\ pixels$$

We consider a threshold value T as the initial threshold value. If the pixel value of the pixel to be determined is less or equal to T then it belongs to the background, else it belongs to the object.

$$f(x,y) > T \rightarrow (x,y) \in object$$

$$f(x,y) \leq T \rightarrow (x,y) \in background$$

E1: Error if a background pixel is misclassified as an object pixel.
E2: Error if an object pixel is misclassified as a background pixel.

$$E_1(T) = \int_{-\infty}^{\infty} p_1(Z)DZ$$

$$E_2(T) = \int_{T}^{\infty} p_2(Z)DZ$$

$$E(T) = P_2E_1(T) + P_1E_2(T)$$

E(T): Total Error in the classification of pixels as background or object. To obtain great segmentation, we have to minimize this error. We do so by taking the derivative of the following equation and equating it to zero.

$$\frac{\delta E(T)}{\delta T} = 0$$

Consider a gaussian pixel density, the value of P(Z) can be calculated as :

$$P(Z) = \left(\frac{P_1}{\sqrt{2\pi}\sigma_1} \cdot e^{-\frac{(z-\mu_1)^2}{2\sigma_1^2}}\right) + \left(\frac{P_2}{\sqrt{2\pi}\sigma_2} \cdot e^{-\frac{(z-\mu_2)^2}{2\sigma_2^2}}\right)$$

$$P_1, P_2 \Rightarrow A\,priori\,probability\,of\,background\,and\,object\,pixels.$$

$$\mu_1, \mu_2 \Rightarrow Mean\,of\,intensity\,of\,pixels\,in\,regions$$

$$\sigma_1, \sigma_2 \Rightarrow Standard\,deviation\,of\,density\,region.$$

After this the new value of T can be calculated by inputting it into the following equation:

$$AT^2 + BT + C = 0$$

$$A = \sigma_1^2 - \sigma_2^2$$

$$B = 2(\mu_1 \cdot \sigma_2^2 - \mu_2 \cdot \sigma_1^2)$$

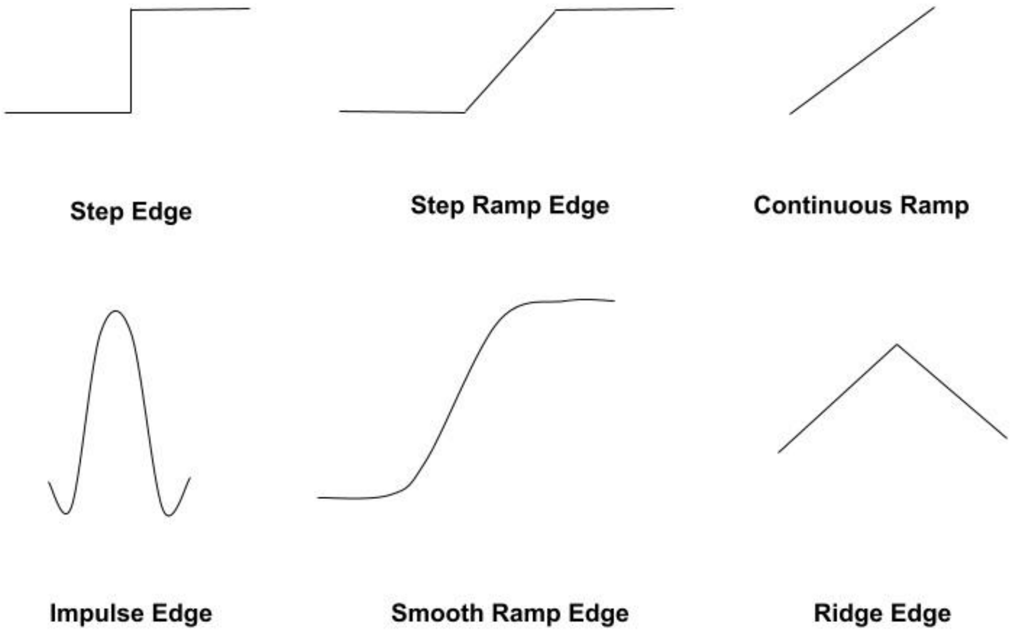
$$C = \sigma_1^2 \cdot \mu_2^2 - \sigma_2^2 \cdot \mu_1^2 + 2\sigma_1^2\sigma_2^2 ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right)$$

- Local Adaptive Thresholding: Due to variation in the illumination of pixels in the image, global thresholding might have difficulty in segmenting the image. Hense the image is divided into smaller subgroups and then adaptive thresholding of those individual groups is done. After individual segmentation of these subgroups, all of them are combined to form the completed segmented image of the original image. Hence, the histogram of subgroups helps in providing better segmentation of the image.

Edge Based Segmentation

Edge-based segmentation relies on edges found in an image using various edge detection operators. These edges mark image locations of discontinuity in gray levels, color, texture, etc. When we move from one region to another, the gray level may change. So if we can find that discontinuity, we can find that edge. A variety of edge detection operators are available but the resulting image is an intermediate segmentation result and should not be confused with the final segmented image. We have to perform further processing on the image to the segment it. Additional steps include combining edges segments obtained into one segment in order to reduce the number of segments rather than chunks of small borders which might hinder the process of region filling. This is done to obtain a seamless border of the object. The goal of edge segmentation is to get an intermediate segmentation result to which we can apply region-based or any other type of segmentation to get the final segmented image.

Types of Edges



Edges are usually associated with “Magnitude” and “Direction”. Some edge detectors give both directions and magnitude. We can use various edge detectors like Sobel edge operator, canny edge detector, Kirsch edge operator, Prewitt edge operator, Robert’s edge operator, etc.

-1	0	1
-2	0	2
-1	0	1

Gx

1	2	1
0	0	0
-1	-2	-1

Gy

Sobel operator with X and Y direction detector

5	5	5
-3	0	-3
-3	-3	-3

N

5	5	-3
5	0	-3
-3	-3	-3

NW

5	-3	-3
5	0	-3
5	-3	-3

W

-3	-3	-3
5	0	-3
5	5	-3

SW

-3	-3	-3
-3	0	-3
5	5	5

S

-3	-3	-3
-3	0	5
-3	5	5

SE

-3	-3	5
-3	0	5
-3	-3	5

E

-3	5	5
-3	0	5
-3	-3	-3

NE

Kirsch Operator

1	0
0	-1

Gx

0	1
-1	0

Gy

Roberts Cross Edge Detector

Any of the three formulae can be used to calculate the value of g. After calculation of g and theta, we obtain the edge vector with both magnitudes as well as direction.

$$g = (g_x^2 + g_y^2)$$

$$g = |g_x| + |g_y|$$

$$g = max(|g_x|, |g_y|)$$

$$\theta = arctan\frac{g_y}{g_x}$$

$$g_y \rightarrow Magnitude\ in\ y$$

$$g_x \rightarrow Magnitude\ in\ x$$

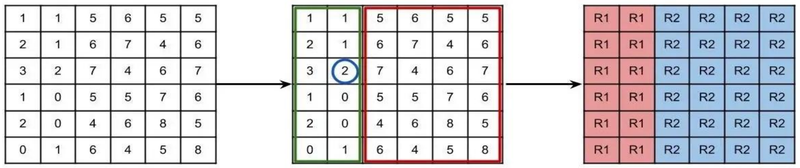
Region-Based Segmentation

A region can be classified as a group of connected pixels exhibiting similar properties. The similarity between pixels can be in terms of intensity, color, etc. In this type of segmentation, some predefined rules are present which have to be obeyed by a pixel in order to be classified into similar pixel regions. Region-based segmentation methods are preferred over edge-based segmentation methods in case of a noisy image. Region-Based techniques are further classified into 2 types based on the approaches they follow.

1. Region growing method
2. Region splitting and merging method

Region Growing Technique

In the case of the Region growing method, we start with some pixel as the seed pixel and then check the adjacent pixels. If the adjacent pixels abide by the predefined rules, then that pixel is added to the region of the seed pixel and the following process continues till there is no similarity left. This method follows the bottom-up approach. In case of a region growing, the preferred rule can be set as a threshold. For example: Consider a seed pixel of 2 in the given image and a threshold value of 3, if a pixel has a value greater than 3 then it will be considered inside the seed pixel region. Otherwise, it will be considered in another region. Hence 2 regions are formed in the following image based on a threshold value of 3.



Original Image

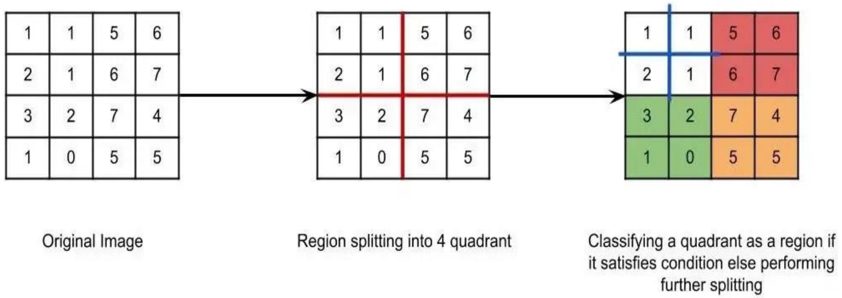
Region growing process with 2 as the seed pixel.

Splitting image into two regions based on a threshold.

Region Splitting and Merging

In Region splitting, the whole image is first taken as a single region. If the region does not follow the predefined rules, then it is further divided into multiple regions (usually 4 quadrants) and then the predefined rules are carried out on those regions in order to decide whether to further subdivide or to classify that as a region. The following process continues till there is no further division of regions required i.e every region follows the predefined rules. In Region merging technique, we consider every pixel as an individual region. We select a region as the seed region to check if adjacent regions are similar based on predefined rules. If they are similar, we merge them into a single region and move ahead in order to build the segmented regions of the whole image. Both region splitting and region merging are iterative processes. Usually, first region splitting is done on an image so as to split an image into maximum regions, and then these regions are merged in order to form a good segmented image of the original image.

In case of Region splitting, the following condition can be checked in order to decide whether to subdivide a region or not. If the absolute value of the difference of the maximum and minimum pixel intensities in a region is less than or equal to a threshold value decided by the user, then the region does not require further splitting.

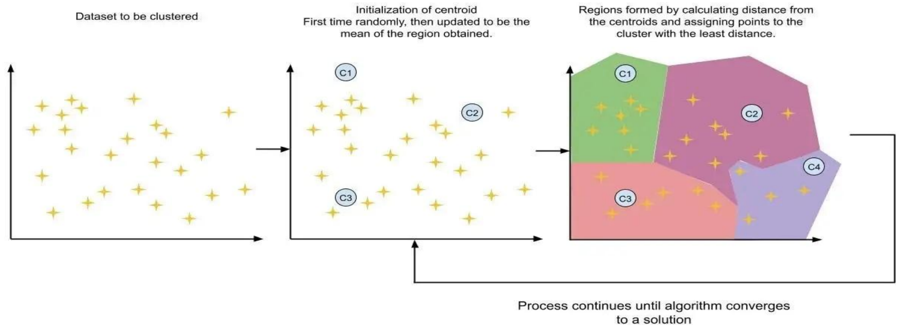


Clustering-Based Segmentation

Clustering is a type of unsupervised machine learning algorithm. It is highly used for the segmentation of images. One of the most dominant clustering-based algorithms used for segmentation is KMeans Clustering. This type of clustering can be used to make segments in a colored image.

KMeans Clustering

Let's imagine a 2-dimensional dataset for better visualization. First, in the dataset, centroids (chosen by the user) are first randomly initialized. Then the distance of all the points to all the clusters is calculated and the point is assigned to the cluster with the least distance. Then centroids of all the clusters are recalculated by taking the mean of that cluster as the centroid. Then again data points are assigned to those clusters. And the process continues till the algorithm converges to a good solution. Usually, the algorithm takes a very small number of iterations to converge to a solution and does not bounce.




Original Image
2 color image
4 color image
6 color image
8 color image
10 color image