**EXPERIMENT NO. 6**

**TO PERFORM IMAGE SEGMENTATION USING GLOBAL THRESHOLDING ALGORITHM**

**EXPERIMENT NO. 6: Segmentation**

**AIM: -** To perform image segmentation using global Thresholding Algorithm

**OBJECTIVES:**

1. To obtain a threshold for the input image.
2. To perform thresholding on that image using the generated threshold.

**EQUIPMENTS/SOFTWARE:** Python

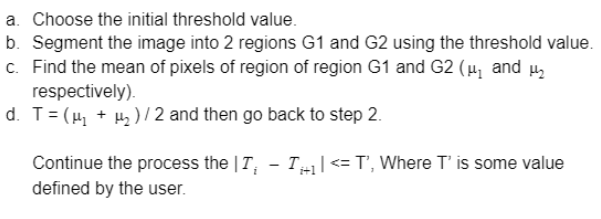
**THEORY: -**

Image segmentation is a method in which a digital image is broken down into various subgroups called Image segments which helps in reducing the complexity of the image to make further processing or analysis of the image simpler. Segmentation in easy words is assigning labels to pixels.

**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.

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.

****

**Code:-**

import numpy as np  
import cv2  
  
def thres\_finder(img, thres=20, delta\_T=1.0):  
  
 # Step-2: Divide the images in two parts  
 x\_low, y\_low = np.where(img<=thres)  
 x\_high, y\_high = np.where(img>thres)  
  
 #Step-3: Find the mean of two parts  
 mean\_low = np.mean(img[x\_low,y\_low])  
 mean\_high = np.mean(img[x\_high,y\_high])  
  
 #Step-4: Calculate the new threshold  
 new\_thres = (mean\_low + mean\_high)/2  
  
 # Step-5: Stopping criteria, otherwise iterate  
 if abs(new\_thres-thres)< delta\_T:  
 return new\_thres  
 else:  
 return thres\_finder(img, thres=new\_thres,delta\_T=1.0)  
  
img = cv2.imread('D:\\college related\\Third year\\sem6\\IPMV\\CODE\\cameraman.png',cv2.IMREAD\_GRAYSCALE)  
  
# apply threshold finder  
vv1 = thres\_finder(img, thres=30,delta\_T=1.0)  
  
# threshold the image  
ret, thresh = cv2.threshold(img,vv1,255,cv2.THRESH\_BINARY)  
  
# Display the image side by side  
out = cv2.hconcat([img, thresh])  
cv2.imshow("Output",out)  
cv2.waitKey(0)

**Output:-**

****

**CONCLUSION** :-

Image segmentation using global Thresholding Algorithm was performed successfully. We obtained threshold for input image. The output was black and white image.