

# Thresholding

## › AIM:

To segment the image using global thresholding, adaptive thresholding and Otsu's thresholding using python and OpenCV.

## › SOFTWARE REQUIRED:

1. Anaconda - Python 3.7
2. OpenCV

## › ALGORITHM:

### › Step 1:

Load the necessary packages.

### › Step 2:

Read the Image and convert to grayscale.

### › Step 3:

Use Global thresholding to segment the image.

### › Step 4:

Use Adaptive thresholding to segment the image.

### › Step 5:

Use Otsu's method to segment the image.

### › Step 6:

Display the results.

## › PROGRAM:

```

# Load the necessary packages
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Read the Image and convert to grayscale
image=cv2.imread("jk.jpg",1)
image=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
image_gray=cv2.imread("jk.jpg",0)

# Use Global thresholding to segment the image
ret,thresh_jk1=cv2.threshold(image_gray,86,255,cv2.THRESH_BINARY)
ret,thresh_jk2=cv2.threshold(image_gray,86,255,cv2.THRESH_BINARY_INV)
ret,thresh_jk3=cv2.threshold(image_gray,86,255,cv2.THRESH_TOZERO)
ret,thresh_jk4=cv2.threshold(image_gray,86,255,cv2.THRESH_TOZERO_INV)
ret,thresh_jk5=cv2.threshold(image_gray,100,255,cv2.THRESH_TRUNC)

# Use Adaptive thresholding to segment the image
thresh_jk7=cv2.adaptive
Threshold(image_gray,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,11,2)
thresh_jk8=cv2.adaptive
Threshold(image_gray,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,2)

# Use Otsu's method to segment the image
ret,thresh_jk6=cv2.threshold(image_gray,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

# Display the results
titles=["Gray Image","Threshold Image (Binary)","Threshold Image (Binary
Inverse)","Threshold Image (To Zero)"
        ,"Threshold Image (To Zero-Inverse)","Threshold Image (Truncate)","Otsu","Adaptive
Threshold (Mean)","Adaptive Threshold (Gaussian)"]
images=
[image_gray,thresh_jk1,thresh_jk2,thresh_jk3,thresh_jk4,thresh_jk5,thresh_jk6,thresh_jk7,thresh

for i in range(0,9):
    plt.figure(figsize=(10,10))
    plt.subplot(1,2,1)
    plt.title("Original Image")
    plt.imshow(image)
    plt.axis("off")
    plt.subplot(1,2,2)
    plt.title(titles[i])
    plt.imshow(cv2.cvtColor(images[i],cv2.COLOR_BGR2RGB))
    plt.axis("off")
    plt.show()

```

’ OUTPUT:

’ Original Image and Grayscale Image

Original Image



Gray Image



’ Global Thresholding

Original Image



Threshold Image (Binary)



Original Image



Threshold Image (Binary Inverse)



Original Image



Threshold Image (To Zero)



Original Image



Threshold Image (To Zero-Inverse)



Original Image



Threshold Image (Truncate)



## ' Adaptive Thresholding

Original Image



Adaptive Threshold (Mean)



Original Image



Adaptive Threshold (Gaussian)



## ' Optimum Global Thesholding using Otsu's Method

Original Image



Otsu



## ' RESULT:

Thus the images are segmented using global thresholding, adaptive thresholding and optimum global thresholding using python and OpenCV.