CLASS : B.E. E &TC **SUBJECT: DIVP**

EXPT. NO. : 8 **DATE:30-10-2020**

TITLE : TO PERFORM GLOBAL THRESHOLDING

CO 2:	Given a gray image, select an appropriate technique (similarity based or discontinuity based) to segment it. Derive the mask coefficients of First order Derivative (FoD) and Second order Derivative (SoD) to detect an edge in an image. Considering an appropriate test case, analyze and compare the performance of FoD and SoD using parameters like response to constant intensity and isolated intensities in an image.
CO4:	Carry out experiments as an individual and in a team, comprehend and write a laboratory recordand draw conclusions at a technical level.

AIM: To implement Global Thresholding

SOFTWARES REQUIRED: Matlab 7.0 or above or python

THEORY:

8.1 Thresholding:

Thresholding is one of the most important approaches to image segmentation.

It is the last step in image segmentation. Different types of thresholding are,

1. Global (depends on intensity value)
2. Adaptive (depends on intensity and local property)
3. Optimal (depends on intensity, position and local property)
4. Local Thresholding function is,

T = [f(x, y), p(x, y), (x, y)]

where f(x, y) = intensity of pixel

p(x, y) = local property (neighborhood)

(x, y) = position of pixel

- When T depends only on f(x, y) → global threshold
- When T depends on both f(x, y) and p(x, y) → local threshold

8.2 Segmentation by thresholding

Thresholding is the simplest segmentation method.

The pixels are partitioned depending on their intensity value I. Global thresholding, using an appropriate threshold T:

g(x, y) =1, if f (x, y) > T

=0, if f (x, y) _ T

Variable thresholding, if T can change over the image, Local or regional thresholding, if T depends on a neighborhood of (x, y). I adaptive thresholding, if T is a function of (x, y).

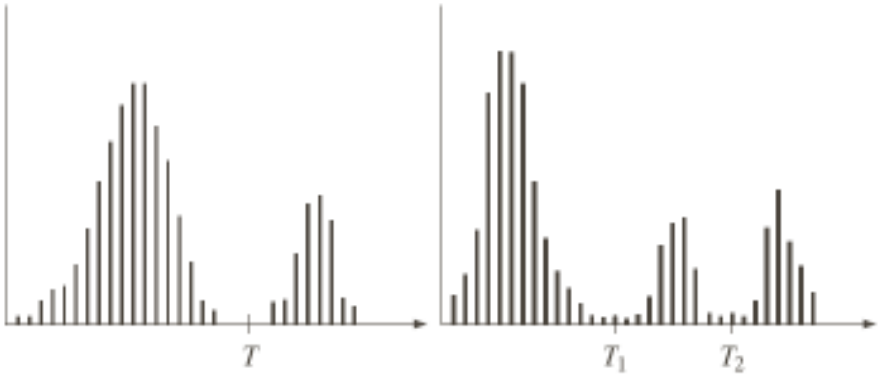
Multiple thresholding:

g(x, y) =a, if f (x, y) > T_2

=b, if T_1 < f (x, y) <T_2

=c, if f (x, y) <T_1

8.3 Choosing the thresholds



Peaks and valleys of the image histogram can help in choosing the appropriate value for the threshold(s). Some factors affect the suitability of the histogram for guiding the choice of the threshold:

- The separation between peaks;
- The noise content in the image;
- The relative size of objects and background;
- The uniformity of the illumination;
- The uniformity of the reflectance.

8.4 Global thresholding:

In practice global thresholding can be expected to be more successful in highly controlled environments such as industrial inspection application, where illumination control is feasible. Usually a successful segmentation is highly depends on the choice of thresholds.

Steps for Global Thresholding:

1) Initial estimate of T
2) Segmentation using T:
    - G_1, pixels brighter than T;
    - G_2, pixels darker than (or equal to) T.

3) Computation of the average intensities m_1 and m_2 of G_1 and G_2.

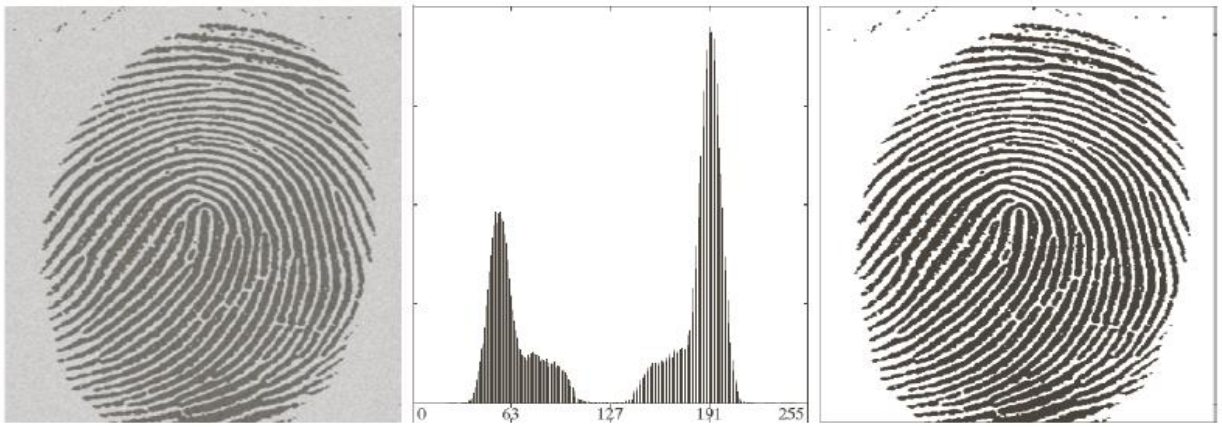
4) New threshold value:

$$T_{new} = (m_1 + m_2)/ 2$$

5) If $|T - T_{new}| >=$ delta (T), back to step 2, otherwise stop.

This simple steps works well in situations where there is a reasonably clear valley between the modes of the histogram related to objects and background. Parameter delta(T) is used to control the number of iterations in situations where speed is important. This initial threshold must be chosen greater than the minimum and less than maximum intensity level in the image.

Global thresholding : An example



8.5 Algorithm:

1) Read the image.

2) Input the threshold value and delta (T)(Delta T is error which can be tolerated).

3) Initial estimate of T

4) Segmentation using T:

- G_1, pixels brighter than T;
- G_2, pixels darker than (or equal to) T.

5) Computation of the average intensities m_1 and m_2 of G_1 and G_2.

6) New threshold value:

$$T_{new} = (m_1 + m_2)/ 2$$

7) If $|T - T_{new}| >= delta(T)$, back to step 4, otherwise stop. (T is of previous iteration)

8) Assign '0' for pixel values less than newly obtained threshold value of original image and '255' for the pixel value greater than threshold.

8.6 Conclusion:

1. Thresholding is a type of image segmentation, where we change the pixels of an image to make the image easier to analyze.
2. In thresholding, we convert an image from color or grayscale into a binary image, i.e., one that is simply black and white.
3. It is used to select areas of interest of an image, while ignoring the parts we are not concerned with hence it makes us easy for recognition and classification.
4. As the image is split into only two colors, we may lose some useful information from image during thresholding process.

8.7 References:

1." Digital Image Processing " , by Gonzalez and Woods.

2. "Digital Image Processing " , S. Jayaraman, S. Esakkirajan, T. Veerakumar.

3. Pictures taken from:

http://www.imageprocessingplace.com/root_files_V3/image_databases.html

(Course Teacher)

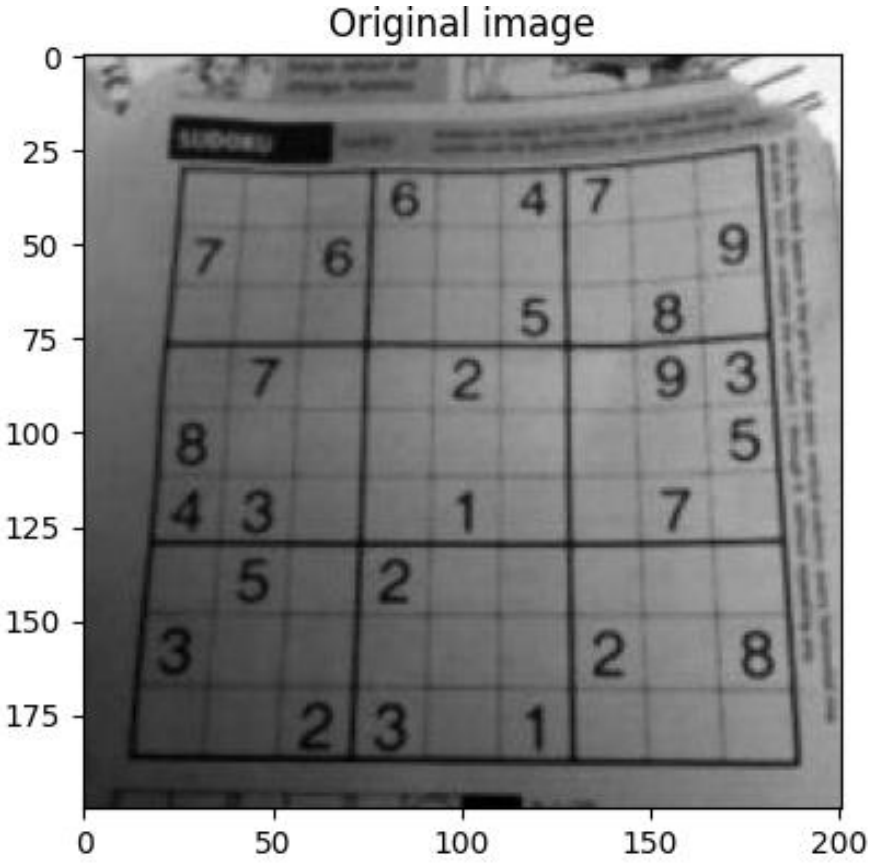
CLASS	**:**	**B.E (E &TC)**	**COURSE**	**:**	**DIVP**
AY	**:**	**2020-21 (SEM- I)**	**DATE**	**:**	**30-10-2020**
EXPT. NO.	**:**	**8**	**CLASS & ROLL NO**	**:**	**BE VIII 42428**
TITLE	**:**	**TO PERFORM GLOBAL THRESHOLDING**			

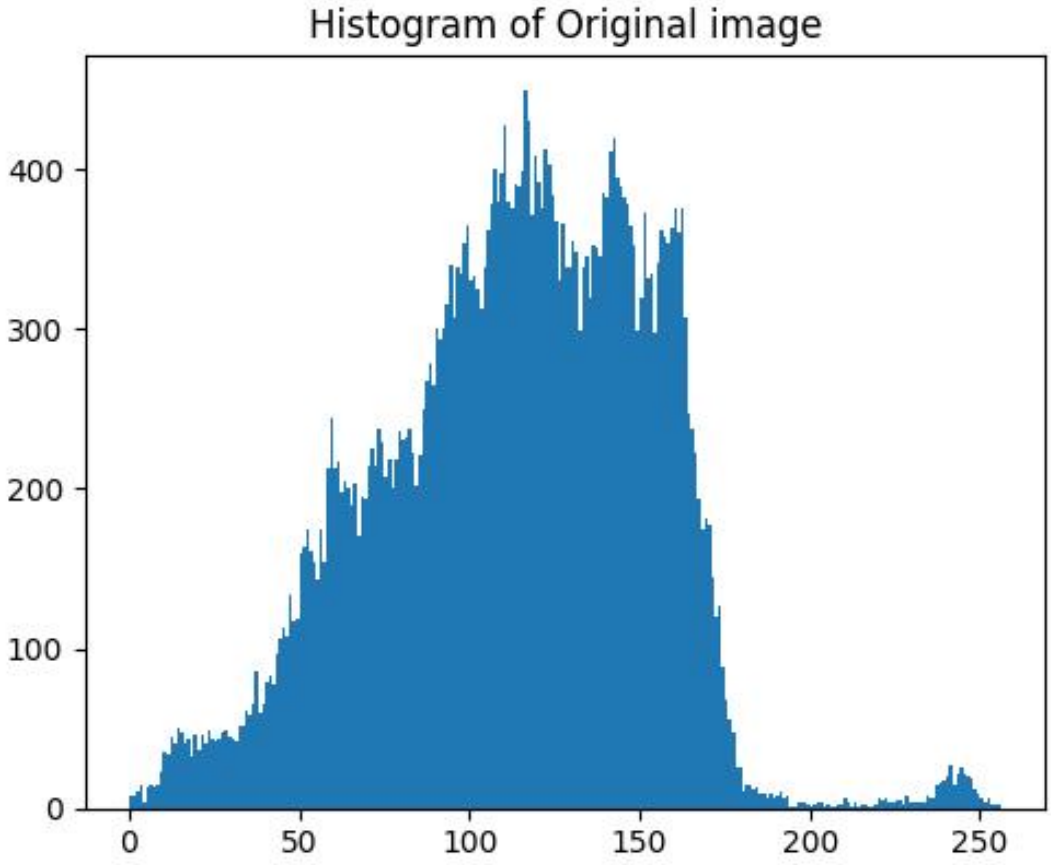
I. CODE:

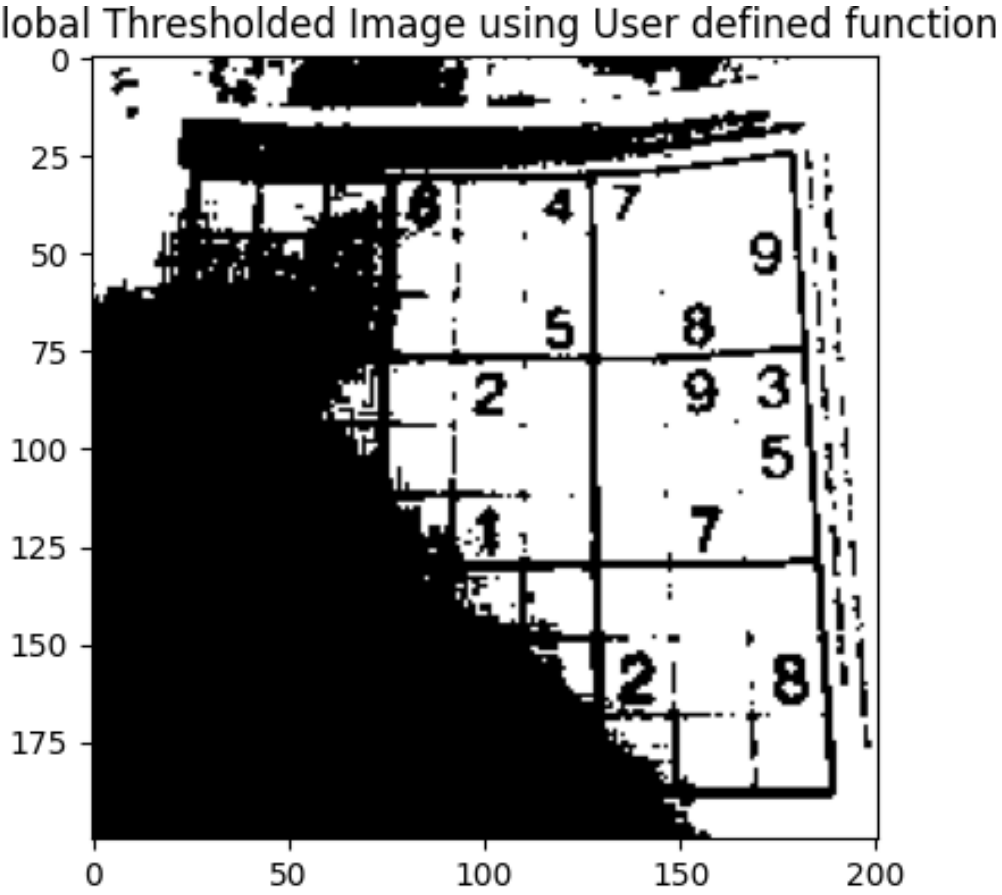
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

def global_threshold(img,T,delta):
    # Find width and height of image
    row, column = img.shape
    G1=0;
    G2=0;
    g1=0;
    g2=0;
    Tnew = 0;
    img1 = np.zeros((row, column),dtype = 'uint8')
    while 1:
        for i in range(row):
            for j in range(column):
                if img[i,j]>T:
                    G1+=img[i,j];
                    g1+=1;
                else:
                    G2+=img[i,j];
                    g2+=1;
        Tnew = ((G1/g1)+(G2/g2))/2;
        if(abs(T-Tnew) < delta):
            break;
        else:
            T=Tnew;
    print(T);
    for i in range(row):
        for j in range(column):
            if img[i,j]>T:
                img1[i,j] = 255;
            else:
                img1[i,j] = 0;
    return img1;

# Load the image
img = cv2.imread('images/threshold.jpg',0)

deltaT = 2
T = 127
```

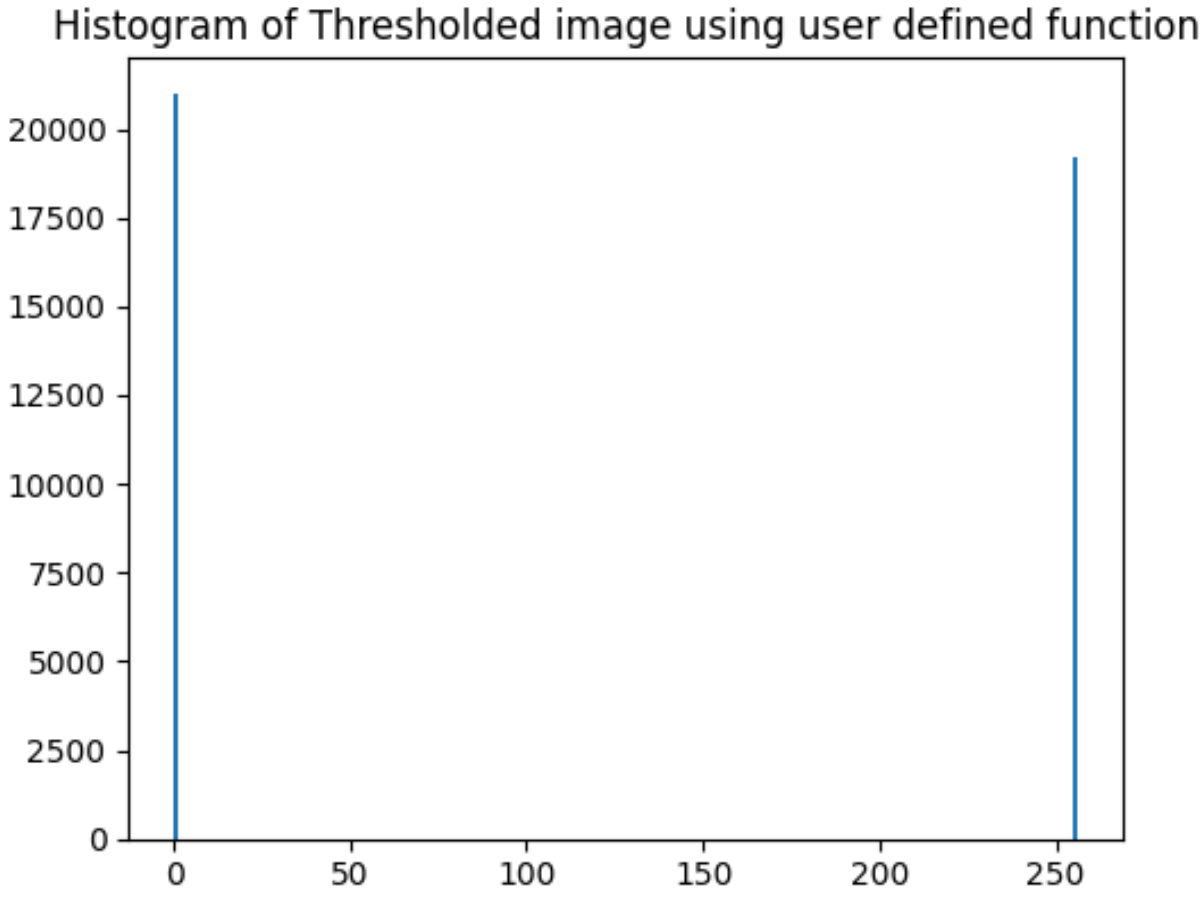
```
out = global_threshold(img,T,deltaT)

ret,th1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
print(ret)

adaptive_thresh1 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_MEAN_C,cv2.
THRESH_BINARY, 19, 3)

adaptive_thresh2 = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C
,cv2.THRESH_BINARY, 19, 3)

# Display the image
plt.subplot(2, 3, 1), plt.imshow(img, 'gray'),plt.title('Original image')
plt.subplot(2, 3, 2), plt.imshow(out, 'gray'),plt.title('Global Thresholded Image using User defined
 function')
plt.subplot(2, 3, 3), plt.imshow(th1, 'gray'),plt.title('Global Thresholded Image using inbuilt functi
on')
plt.subplot(2, 3, 4),plt.hist(img.ravel(),256,[0,256]),plt.title('Histogram of Original image')
plt.subplot(2, 3, 5),plt.hist(out.ravel(),256,[0,256]),plt.title('Histogram of Thresholded image usin
g user defined function')
plt.subplot(2, 3, 6),plt.hist(th1.ravel(),256,[0,256]),plt.title('Histogram of Thresholded image usin
g inbuilt function')
plt.show()

plt.subplot(2, 2, 1), plt.imshow(adaptive_thresh1, 'gray'),plt.title('Adaptive Mean Thresholded ima
ge')
plt.subplot(2, 2, 2), plt.imshow(adaptive_thresh2, 'gray'),plt.title('Adaptive Gaussian Thresholded
image')
plt.subplot(2, 2, 3),plt.hist(img.ravel(),256,[0,256]),plt.title('Histogram of Mean Thresholded ima
ge')
plt.subplot(2, 2, 4),plt.hist(img.ravel(),256,[0,256]),plt.title('Histogram of Adaptive Thresholded i
mage')
plt.show()
```

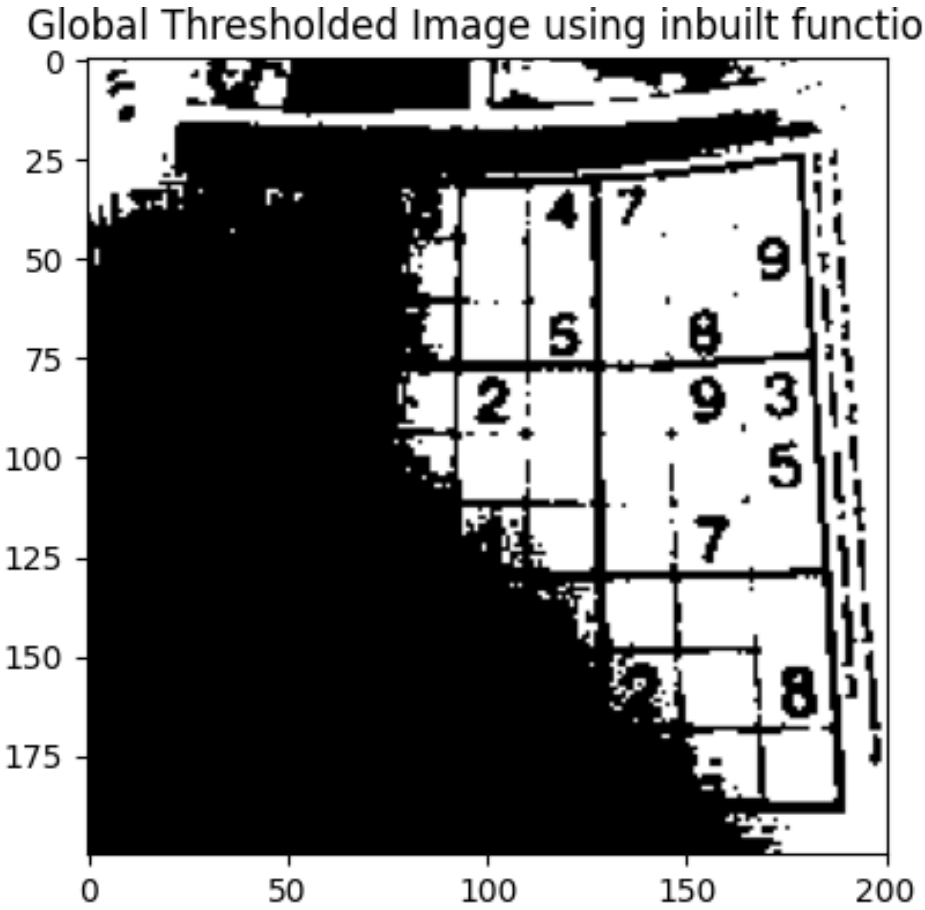
II. RESULTS:

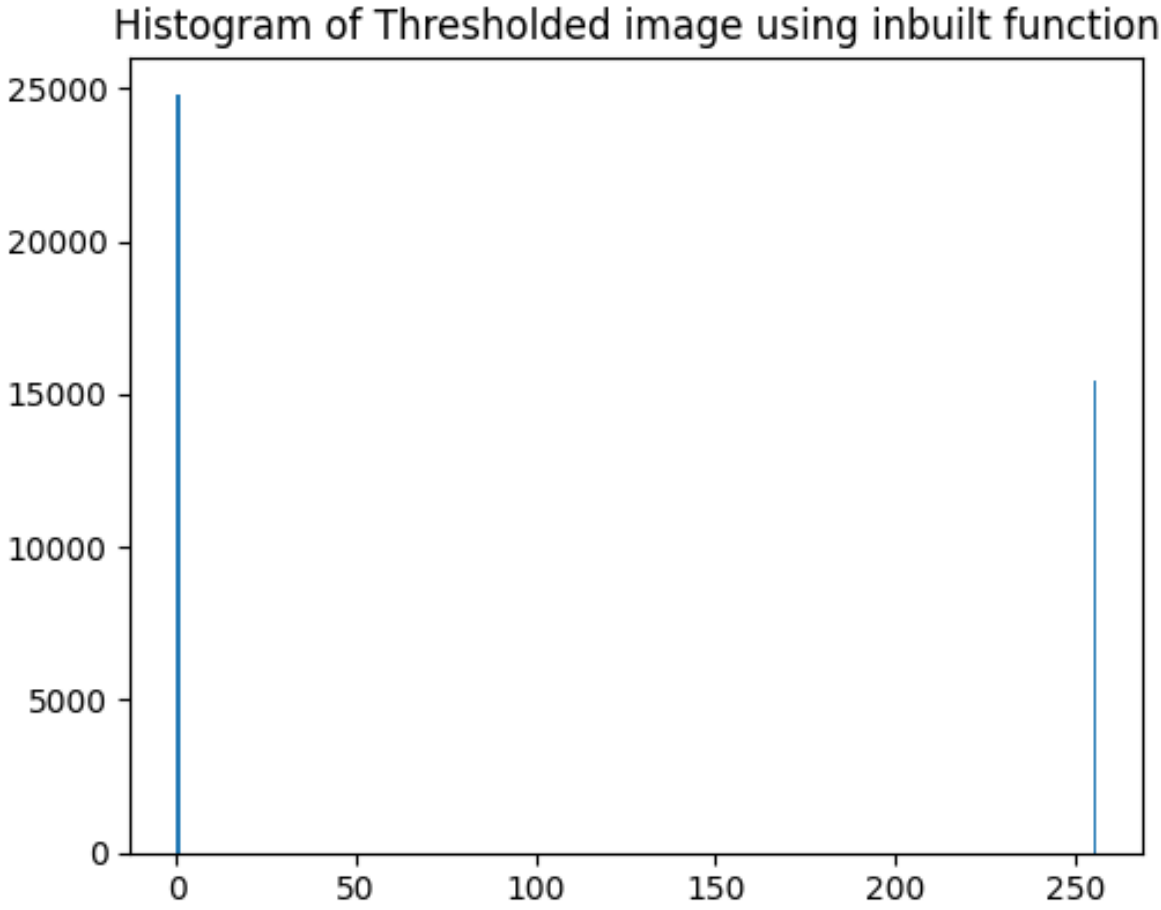

Original image
0
25
50
75
100
125
150
175
0
50
100
150
200

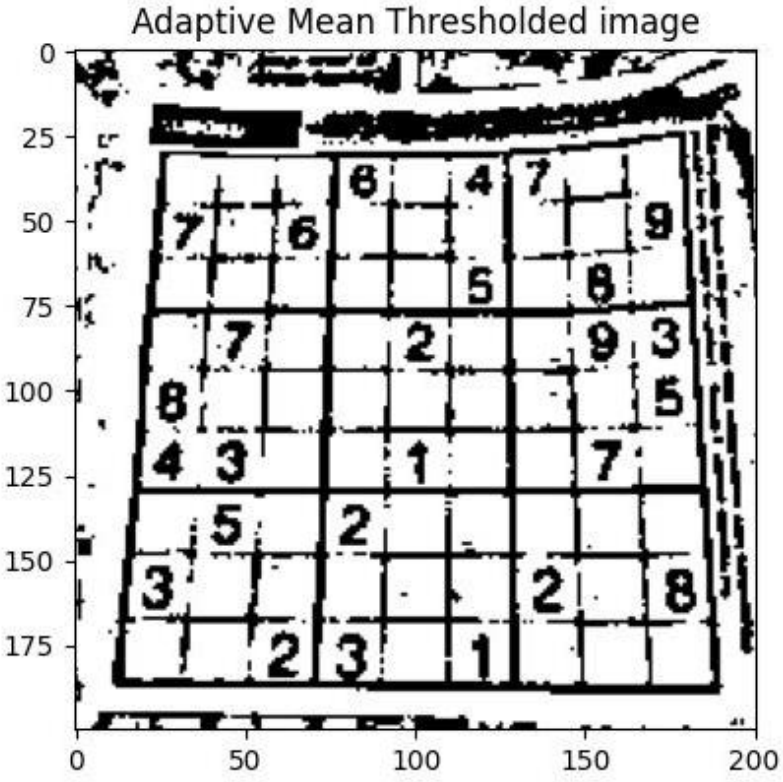

Histogram of Original image
400
300
200
100
0
0
50
100
150
200
250

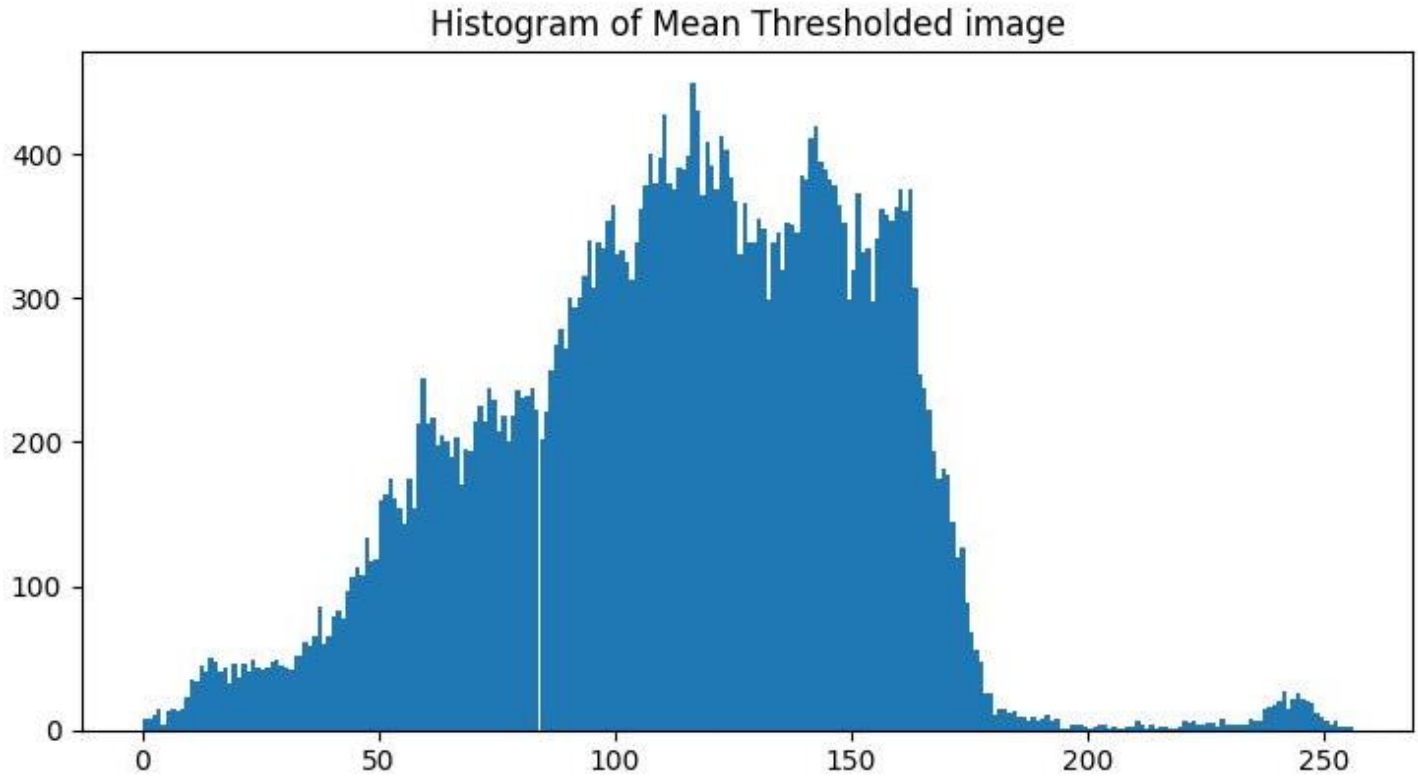

Global Thresholded Image using User defined function
0
25
50
75
100
125
150
175
0
50
100
150
200

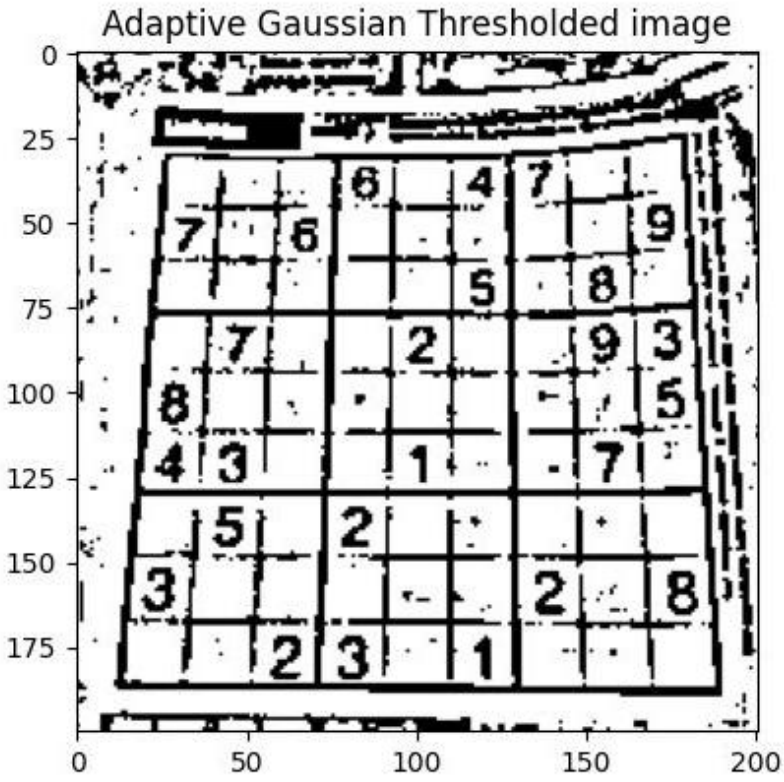

Histogram of Thresholded image using user defined function
20000
17500
15000
12500
10000
7500
5000
2500
0
0
50
100
150
200
250

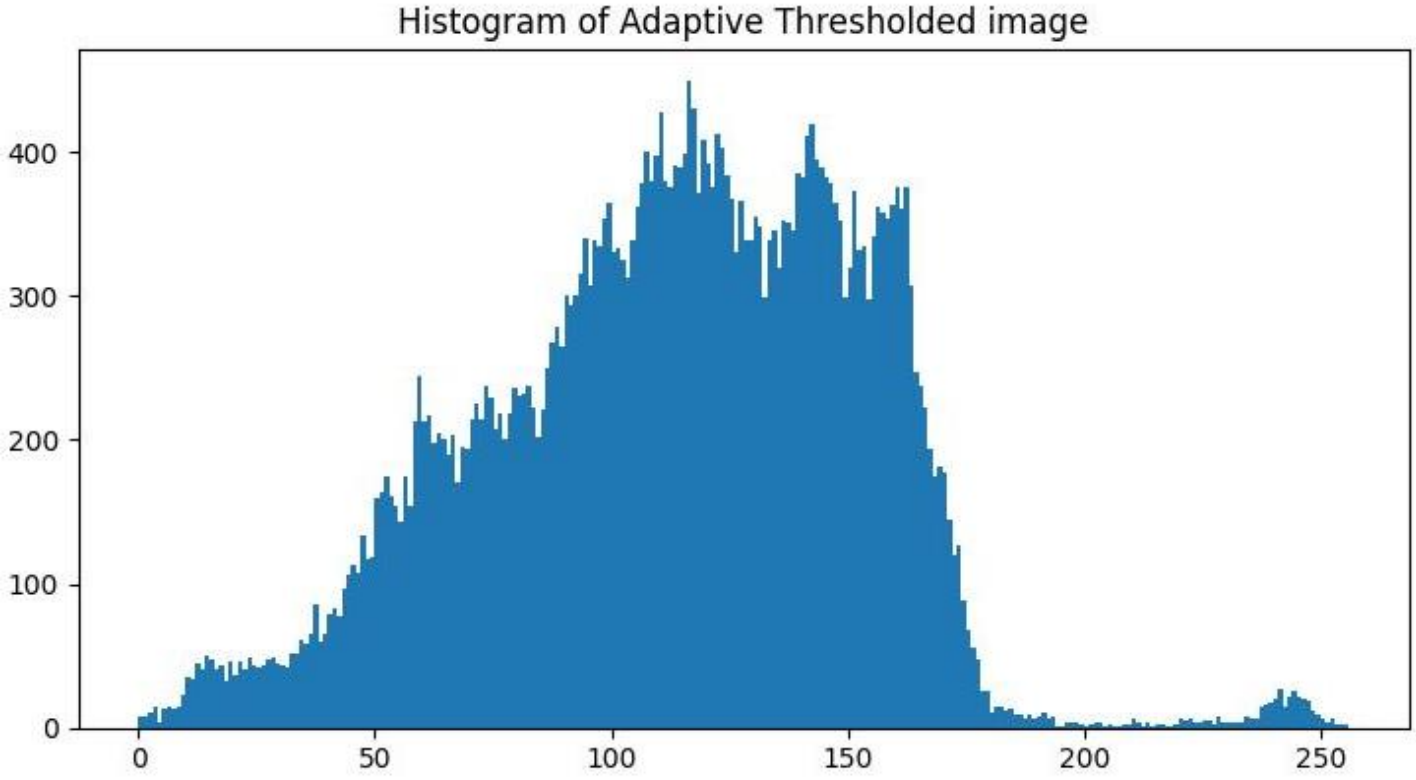

Global Thresholded Image using inbuilt function
0
25
50
75
100
125
150
175
0
50
100
150
200


Histogram of Thresholded image using inbuilt function
25000
20000
15000
10000
5000
0
0
50
100
150
200
250


Adaptive Mean Thresholded image
0
25
50
75
100
125
150
175
0
50
100
150
200


Histogram of Mean Thresholded image
400
300
200
100
0
0
50
100
150
200
250


Adaptive Gaussian Thresholded image
0
25
50
75
100
125
150
175
0
50
100
150
200


Histogram of Adaptive Thresholded image
400
300
200
100
0
0
50
100
150
200
250