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

Step1:

Load the necessary packages.

Step2:

Read the Image and convert to grayscale.

Step3:

Use Global thresholding to segment the image.

Step4:

Use Adaptive thresholding to segment the image.

Step5:

Use Otsu's method to segment the image and display the results.

Program

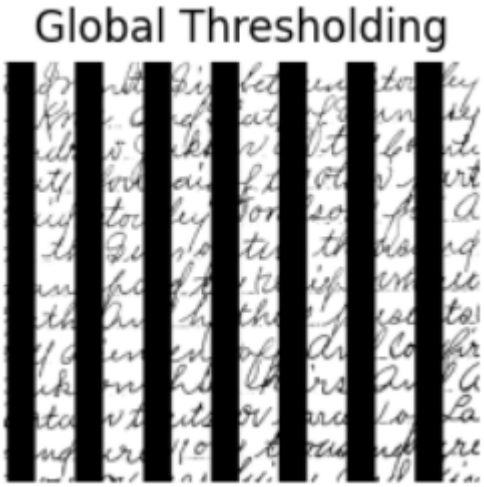
```
# Load the necessary packages

import cv2
import matplotlib.pyplot as plt

# Read the Image and convert to grayscale

image=cv2.imread('IMG.tif')
```

```
gray_img=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

# Original image

plt.subplot(2,2,1)
plt.imshow(cv2.cvtColor(image,cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

# Use Global thresholding to segment the image

_,global_thresholded = cv2.threshold(gray_img, 127, 255, cv2.THRESH_BINARY)

# Use Adaptive thresholding to segment the image

adaptive_thresholded = cv2.adaptiveThreshold(gray_img, 255, cv2.ADAPTIVE_THRESH_GAUSSI

# Use Otsu's method to segment the image

_,otsu_thresholded = cv2.threshold(gray_img, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OT

# Global Thresholding
plt.subplot(2, 2, 2)
plt.imshow(global_thresholded, cmap='gray')
plt.title("Global Thresholding")
plt.axis('off')

# Adaptive Thresholding
plt.subplot(2, 2, 3)
plt.imshow(adaptive_thresholded, cmap='gray')
plt.title("Adaptive Thresholding")
plt.axis('off')

# Otsu's Method
plt.subplot(2, 2, 4)
plt.imshow(otsu_thresholded, cmap='gray')
plt.title("Otsu's Method")
plt.axis('off')

# Show the plot
plt.tight_layout()
plt.show()
```

Output

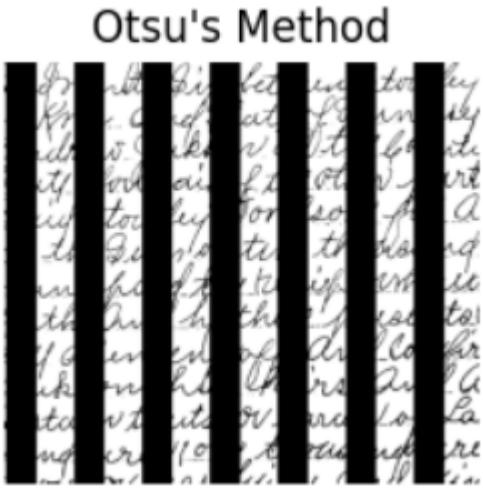
Original Image

Global Thresholding



Adaptive Thresholding

Optimum Global Thesholding using Otsu's Method



Result

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