Capstone Project Report

Name: Sumit Patel

Course: AI and ML (Batch – AUG 2020)

Duration: 10 months

Adaptive Thresholding_Edge Detection in Images

Problem Statement:

Using OpenCV, first convert any image with varying High condition to a grayscale image. Now implement edge detection first using the canny edge detection. Then apply simple thresholding and also Adaptive/OTSU thresholding using OpenCV to see the working of each of these methods. Once you obtain good results, use the obtained edge detection result as a mask to give color to all the edges (if edges use the color from the original image, else leave it black only).

Prerequisites

What things you need to install the software and how to install them:

Python 3.6 This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-externalcommand/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic. Second and easier option is to download anaconda and use its anaconda prompt to run the commands.

To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.6 then run below commands in command prompt/terminal to install these packages:

pip install numpy

pip install cv2

pip install matplotlib

If you have chosen to install anaconda then run below commands in anaconda prompt to install these packages:

conda install -c anaconda numpy

conda install -c anaconda cv2

conda install -c anaconda matplotlib

Importing the libraries and loading dataset.

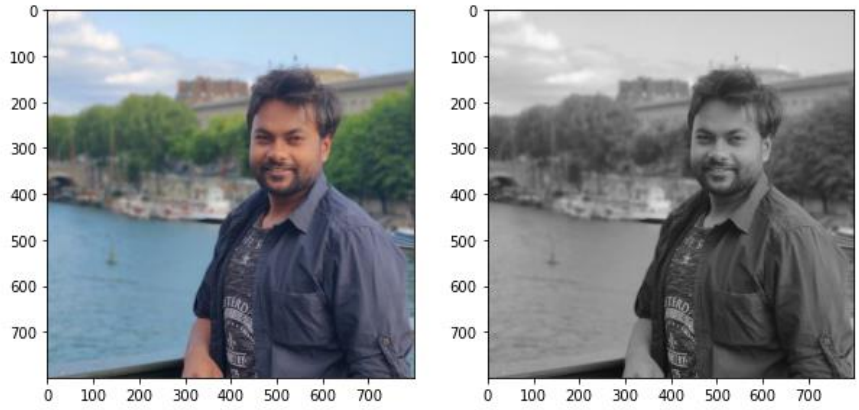
```
# Import libraries
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Loading and reading image

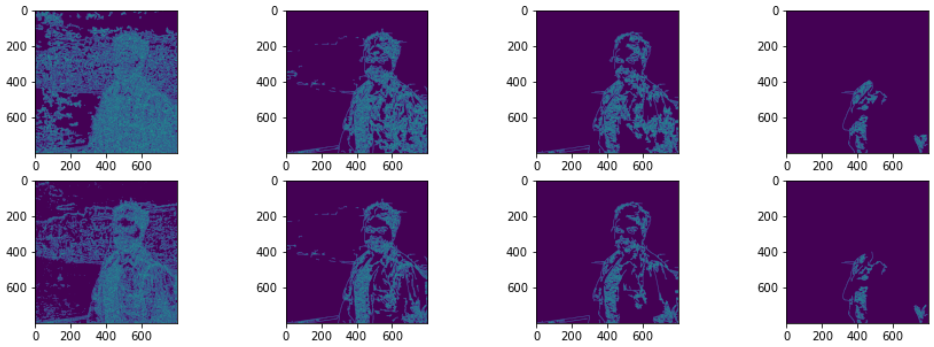
```
# Read color Image
img = cv2.imread("Image.jpg")
```

```
# Read grayscale Image
img_gray = cv2.imread("Image.jpg", 0)
```

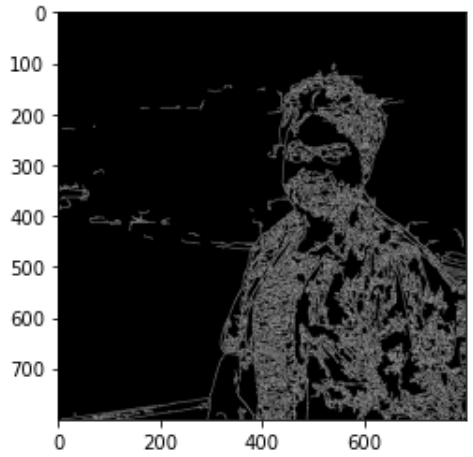
```
# Show Image
%matplotlib inline
fig, ax = plt.subplots(ncols = 2, figsize = (10,15))
ax[0].imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB));
ax[1].imshow(cv2.cvtColor(img_gray, cv2.COLOR_BGR2RGB));
```



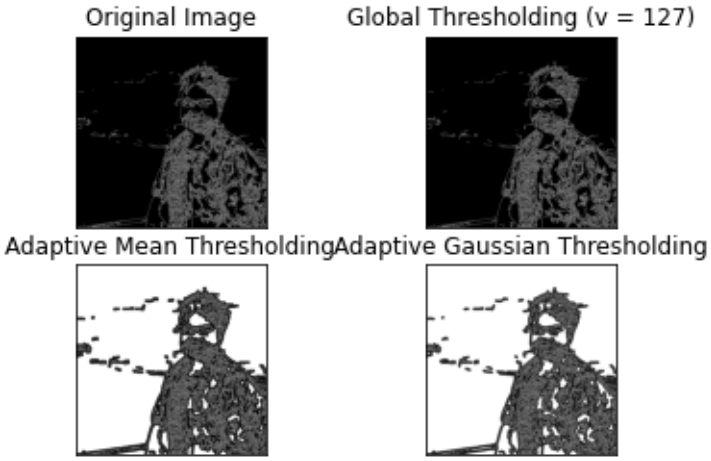

```
# Edge detection using Canny
minval = [1, 25]
maxval = [30, 100, 200, 500]
fig, ax = plt.subplots(ncols = len(maxval), nrows = len(minval), figsize = (15, 5))
for n, i in enumerate(minval):
    for m, j in enumerate(maxval):
        edge = cv2.Canny(img_gray, i, j)
        ax[n,m].imshow(edge);
```



```
# Simple Thresholding
ret, thres = cv2.threshold(edge_fin, 100, 255, cv2.THRESH_BINARY)
plt.imshow(thres);
```




```
# Adaptive thresholding
ret,th1 = cv2.threshold(edge_a,127,255,cv2.THRESH_BINARY)
th2 = cv2.adaptiveThreshold(edge_a,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,11,2)
th3 = cv2.adaptiveThreshold(edge_a,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,2)
titles = ['Original Image', 'Global Thresholding (v = 127)',
            'Adaptive Mean Thresholding', 'Adaptive Gaussian Thresholding']
images = [edge_a, th1, th2, th3]
for i in range(4):
    plt.subplot(2,2,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])
plt.show()
```



```
# OTSU's Thresholding
ret1, thresh1 = cv2.threshold(edge_a, 120, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)
plt.imshow(cv2.cvtColor(thresh1, cv2.COLOR_BGR2RGB));
```