Guided Projects: Unsupervised Learning

Adaptive Thresholding: Edge Detection in Images

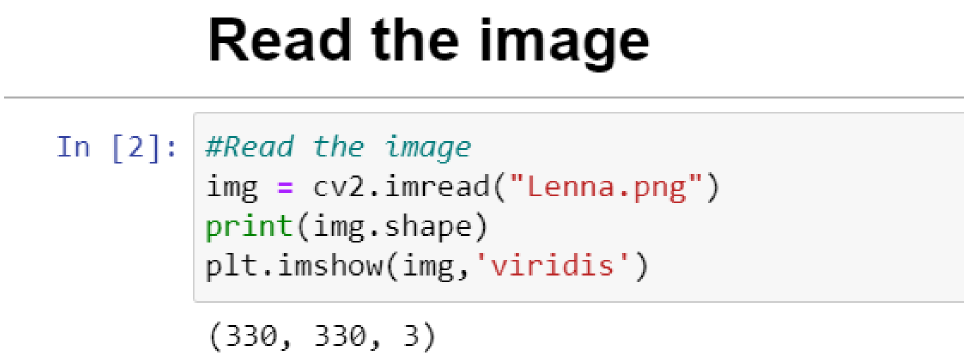
Name	**Lakshmi Thirunavukkarasu**
Course	**AI and ML (Batch 5)**
Problem Statement	Edges define the boundaries between different regions in an image, which helps in matching the pattern, segment, and recognize an object. Detects edges by applying various thresholds. Once best result is obtained, use the obtained edge detection result as a mask to give color to all the edges

Software requirements perquisites

1. Anaconda
2. Python 3.8
3. Python Packages
    - NumPy
    - OpenCV
    - Matplotlib

Steps

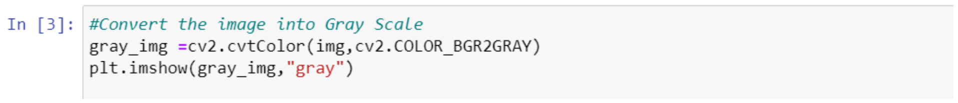
1. Read the image from the scripts folder.



Read the image

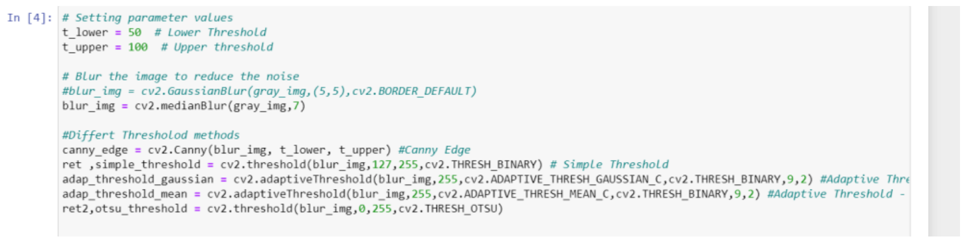
```
In [2]: #Read the image
        img = cv2.imread("Lenna.png")
        print(img.shape)
        plt.imshow(img,'viridis')

        (330, 330, 3)
```

2. Convert the image into gray scale image.



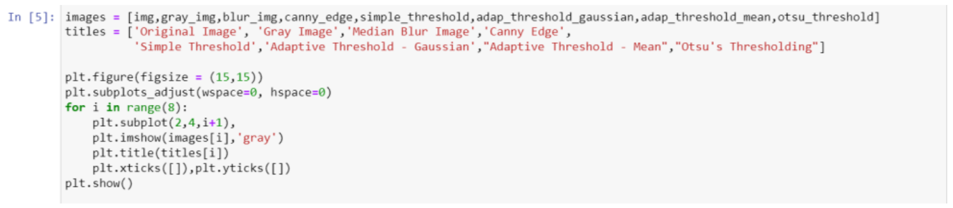
```
In [3]: #Convert the image into Gray Scale
        gray_img =cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        plt.imshow(gray_img,"gray")
```

3. Apply different thresholds on the image.



```
In [4]: # Setting parameter values
        t_lower = 50  # Lower Threshold
        t_upper = 100  # Upper threshold

        # Blur the image to reduce the noise
        #blur_img = cv2.GaussianBlur(gray_img,(5,5),cv2.BORDER_DEFAULT)
        blur_img = cv2.medianBlur(gray_img,7)

        #Differt Thresholod methods
        canny_edge = cv2.Canny(blur_img, t_lower, t_upper) #Canny Edge
        ret ,simple_threshold = cv2.threshold(blur_img,127,255,cv2.THRESH_BINARY) # Simple Threshold
        adap_threshold_gaussian = cv2.adaptiveThreshold(blur_img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,9,2) #Adaptive Thre
        adap_threshold_mean = cv2.adaptiveThreshold(blur_img,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,9,2) #Adaptive Threshold -
        ret2,otsu_threshold = cv2.threshold(blur_img,0,255,cv2.THRESH_OTSU)
```

4. Display image with different thresholds



```
In [5]: images = [img,gray_img,blur_img,canny_edge,simple_threshold,adap_threshold_gaussian,adap_threshold_mean,otsu_threshold]
        titles = ['Original Image', 'Gray Image','Median Blur Image','Canny Edge',
                 'Simple Threshold','Adaptive Threshold - Gaussian',"Adaptive Threshold - Mean","Otsu's Thresholding"]

        plt.figure(figsize = (15,15))
        plt.subplots_adjust(wspace=0, hspace=0)
        for i in range(8):
            plt.subplot(2,4,i+1),
            plt.imshow(images[i],'gray')
            plt.title(titles[i])
            plt.xticks([]),plt.yticks([])
        plt.show()
```

5. Apply the edge detection as a mask on the original image to color the edges.

In [6]:
```
import numpy as np
import numpy.ma as ma
masked_img = cv2.cvtColor(canny_edge, cv2.COLOR_GRAY2RGB)
pos_index = np.where((masked_img == [255,255,255]).all(axis=2))
masked_img[pos_index] = img[pos_index]
plt.imshow(masked_img,'viridis')
```