

CAPSTONE PROJECT REPORT

Adaptive Thresholding: Edge Detection in Images

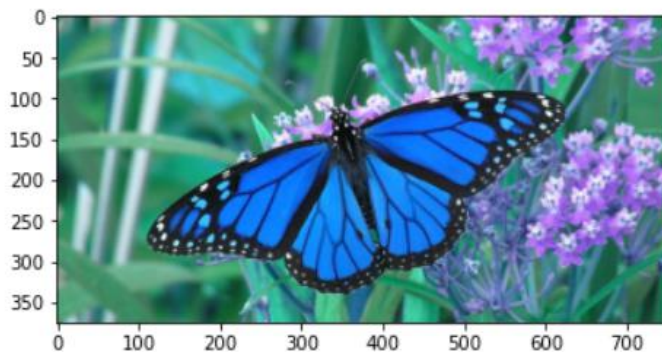
Name - Paleti Samuel Yashaswi
Course - Machine learning and AI
Duration - 24 months
Question - 5

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)

```
: import cv2
import numpy as np
import matplotlib.pyplot as plt
img = cv2.imread('her.jpg')
```

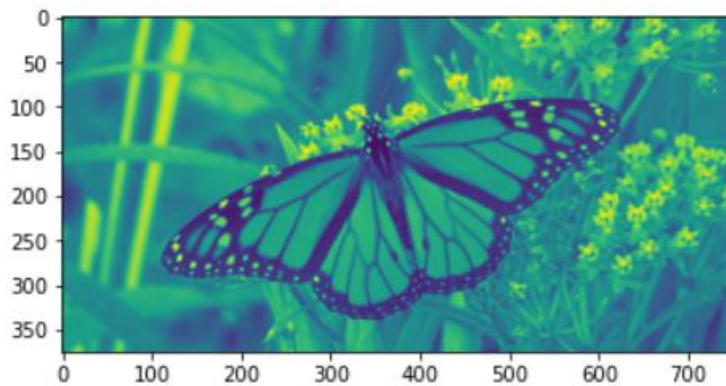
```
: plt.imshow(img)
```

```
: <matplotlib.image.AxesImage at 0xfc7daf0>
```



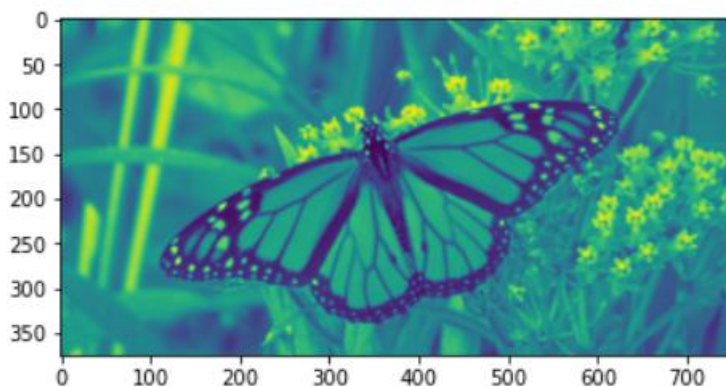
```
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(img_gray)
```

<matplotlib.image.AxesImage at 0xfce43d0>



```
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)
plt.imshow(img_blur)
```

<matplotlib.image.AxesImage at 0xfd26610>



```
edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200)
plt.imshow(edges)
```

<matplotlib.image.AxesImage at 0x1302358>

