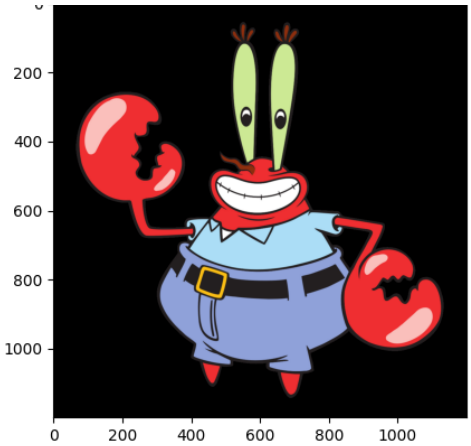
NIDA RAMZAN
CS-121

# LAB: 07

## ORIGNAL:

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
image = cv2.imread('D:/AI_labs/lab-7/Mr.krab.png')
image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
#plotting the image
plt.imshow(image)
plt.show()
```

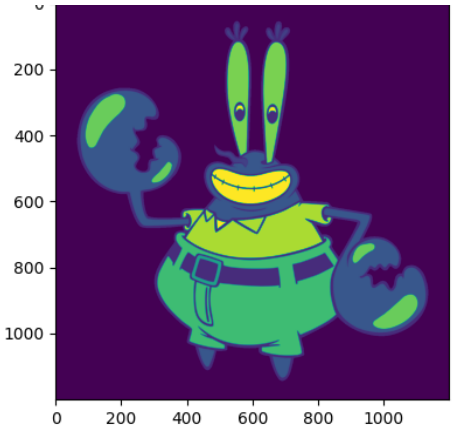
132 % No issues found




## GRAY:

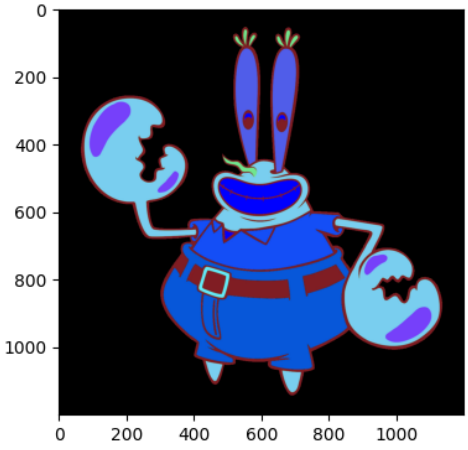
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
image = cv2.imread('D:/AI_labs/lab-7/Mr.krab.png')
image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
#plotting the image
plt.imshow(image)
#saving image
cv2.imwrite('D:/AI_labs/lab-7/new_krab.png',image)
#converting image to Gray scale
gray_image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
cv2.imwrite('D:/AI_labs/lab-7/gray_krab.png',gray_image)
#plotting the grayscale image
plt.imshow(gray_image)
plt.show()
```

thon 3.7 (64-bit) Interactive 4




### a) Plot in HSV

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
image = cv2.imread('D:/AI_labs/lab-7/Mr.krab.png')
image = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
#plotting the image
plt.imshow(image)
#saving image
cv2.imwrite('D:/AI_labs/lab-7/new_krab.png',image)
#converting image to HSV format
hsv_image = cv2.cvtColor(image,cv2.COLOR_BGR2HSV)
cv2.imwrite('D:/AI_labs/lab-7/hsv_krab.png',hsv_image)
#plotting the HSV image
plt.imshow(hsv_image)
plt.show()
```



NIDA RAMZAN
CS-121

## b) Apply static thresholding:

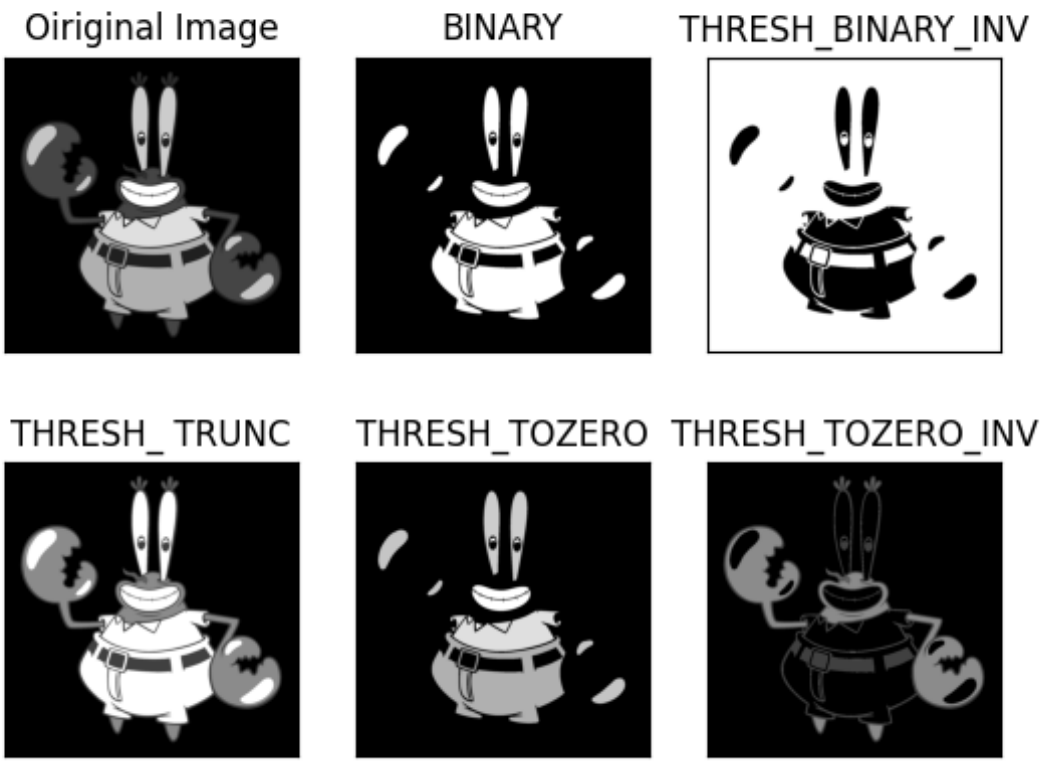
```python
import numpy as np
import matplotlib.pyplot as plt
import cv2
#STATIC THRESHOLDING
gray_image = cv2.imread('D:/AI_labs/lab-7/Mr.krab.png',0)

ret,thresh_binary = cv2.threshold(gray_image,127,255,cv2.THRESH_BINARY)
ret,thresh_binary_inv = cv2.threshold(gray_image,127,255,cv2.THRESH_BINARY_INV)
ret,thresh_trunc = cv2.threshold(gray_image,127,255,cv2.THRESH_TRUNC)
ret,thresh_tozero = cv2.threshold(gray_image,127,255,cv2.THRESH_TOZERO)
ret,thresh_tozero_inv = cv2.threshold(gray_image,127,255,cv2.THRESH_TOZERO_INV)

#DISPLAYING THE DIFFERENT THRESHOLDING STYLES
names = ['Oiriginal Image','BINARY', 'THRESH_BINARY_INV', 'THRESH_ TRUNC', 'THRESH_TOZERO', 'THRESH_TOZERO_INV']
images = gray_image ,thresh_binary ,thresh_binary_inv, thresh_trunc, thresh_tozero, thresh_tozero_inv
for i in range(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
    plt.title(names[i])

    plt.xticks([]),plt.yticks([])

plt.show()
```



NIDA RAMZAN
CS-121

## c) Best result thresholding

lab7.py

```python
import numpy as np
import matplotlib.pyplot as plt
import cv2
#STATIC THRESHOLDING
gray_image = cv2.imread('D:/AI_labs/lab-7/Mr.krab.png',0)
ret,thresh_global = cv2.threshold(gray_image,127,255,cv2.THRESH_BINARY)

#here 11 is the pixel neighbourhood that is used to calculate the threshold value
thresh_mean = cv2.adaptiveThreshold(gray_image,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.THRESH_BINARY,11,2)

thresh_gaussian = cv2.adaptiveThreshold(gray_image,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,2)

names = ['Original Image','Global Thresholding','Adaptive Mean Threshold','Adaptive Gaussian Thresholding']
images = [gray_image,thresh_global,thresh_mean,thresh_gaussian]

for i in range(4):
    plt.subplot(2,2,i+1),plt.imshow(images[i],'gray')
    plt.title(names[i])
    plt.xticks([]),plt.yticks([])
plt.show()
```