

### 3. Gray level Slicing, Intensity slicing, Boxplot slicing, Bit plane slicing, Histogram Equalizer

```
In [1]: import cv2
import numpy as np
import matplotlib.pyplot as plt
img=cv2.imread("black.png",0)
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
```

```
In [ ]:
```

```
In [6]: #gray level

def gray_level(image,max_intensity,min_intensity):
    img1=image.copy()
    img1[(min_intensity<=image) & (image<=max_intensity)]=255

    img1[(min_intensity>image) | (image>max_intensity)]=0
    #img1=0
    return img1
gray=gray_level(img,200,100)
```

```
In [7]: #boxplot

def boxplot(img):
    x=np.quantile(img,[0.25,0.5,0.75,1])
    img1=img.copy()
    img1=cv2.resize(img1,(400,400))
    for i in range(200):
        for j in range(200):
            if img1[i,j]<100:
                img1[i,j]=x[0]
    for i in range(200):
        for j in range(200,400):
            if img1[i,j]<100:
                img1[i,j]=x[1]
    for i in range(200,400):
        for j in range(200):
            if img1[i,j]<100:
                img1[i,j]=x[2]
    for i in range(200,400):
        for j in range(200,400):
            if img1[i,j]<100:
                img1[i,j]=x[3]
    return img1

box=boxplot(img)
```

```
In [8]: #histogram equalizer

def histogram_equalizer(img):
    img1=cv2.equalizeHist(img)
    return img1
```

```
histo=hisatogram_equilizer(img)
```

In [ ]:

In [9]: *#bit plane*

```
def bit_plane(img):
    lst = []
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            lst.append(np.binary_repr(img[i][j] ,width=8))

    eight_bit_img = (np.array([int(i[0]) for i in lst],dtype = np.uint8) * 128).res
    seven_bit_img = (np.array([int(i[1]) for i in lst],dtype = np.uint8) * 64).resh
    six_bit_img = (np.array([int(i[2]) for i in lst],dtype = np.uint8) * 32).reshap
    five_bit_img = (np.array([int(i[3]) for i in lst],dtype = np.uint8) * 16).resha
    four_bit_img = (np.array([int(i[4]) for i in lst],dtype = np.uint8) * 8).reshap
    three_bit_img = (np.array([int(i[5]) for i in lst],dtype = np.uint8) * 4).resha
    two_bit_img = (np.array([int(i[6]) for i in lst],dtype = np.uint8) * 2).reshape
    one_bit_img = (np.array([int(i[7]) for i in lst],dtype = np.uint8) * 1).reshape

    finalr = cv2.hconcat([eight_bit_img,seven_bit_img,six_bit_img,five_bit_img])
    finalv =cv2.hconcat([four_bit_img,three_bit_img,two_bit_img,one_bit_img])

    final = cv2.vconcat([finalr,finalv])
    return final

bit=bit_plane(img)
```

In [27]: plt.figure(figsize=(20, 15))

```
plt.subplot(5, 2, 1), plt.imshow(img, cmap='gray'), plt.title('Original Image')
plt.subplot(5, 2, 2), plt.hist(img.flatten(), bins=256, range=[0, 256]), plt.title(

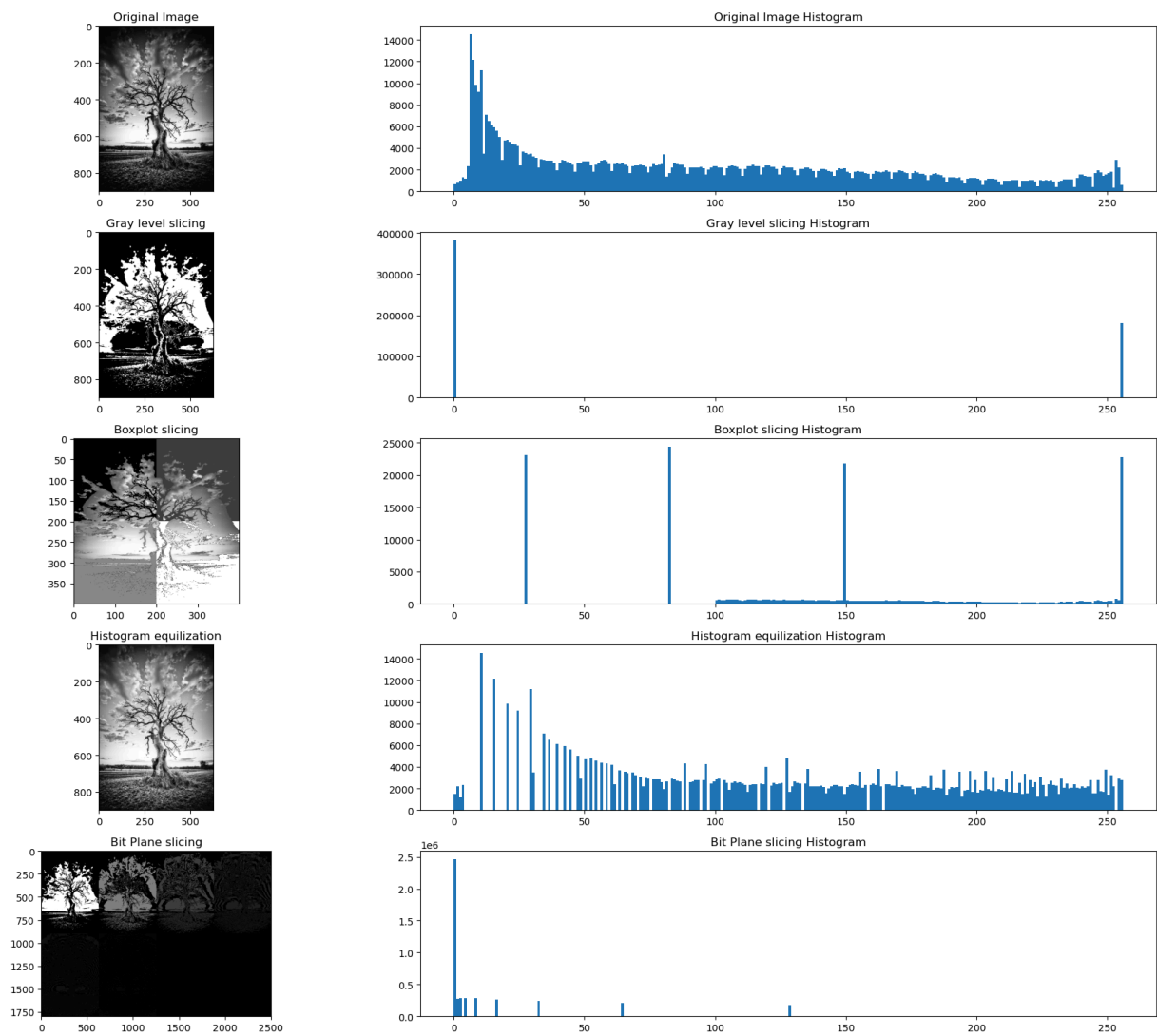
plt.subplot(5, 2, 3), plt.imshow(gray, cmap='gray'), plt.title('Gray level slicing')
plt.subplot(5, 2, 4), plt.hist(gray.flatten(), bins=256, range=[0, 256]), plt.title

plt.subplot(5, 2, 5), plt.imshow(box, cmap='gray'), plt.title('Boxplot slicing')
plt.subplot(5, 2, 6), plt.hist(box.flatten(), bins=256, range=[0, 256]), plt.title(

plt.subplot(5, 2, 7), plt.imshow(histo, cmap='gray'), plt.title('Histogram equiliza
plt.subplot(5, 2, 8), plt.hist(histo.flatten(), bins=256, range=[0, 256]), plt.titl

plt.subplot(5, 2, 9), plt.imshow(bit, cmap='gray'), plt.title('Bit Plane slicing')
plt.subplot(5, 2, 10), plt.hist(bit.flatten(), bins=256, range=[0, 256]), plt.title

plt.tight_layout()
plt.show()
```



In [ ]: