

Name: B.S.V.W. Munasinghe

Index Number: 190397E

```
In [ ]: #Importing Libraries
import numpy as np
import sympy as sy
import matplotlib.pyplot as plt
import cv2 as cv
%matplotlib inline
```

## Before Exercise

### Intensity Transformation example in class

```
In [ ]: f = cv.imread("spider.png",cv.IMREAD_GRAYSCALE)
assert f is not None

t = np.arange(255,-1,-1,dtype=np.uint8)
g = cv.LUT(f,t) #same as g = t[f]

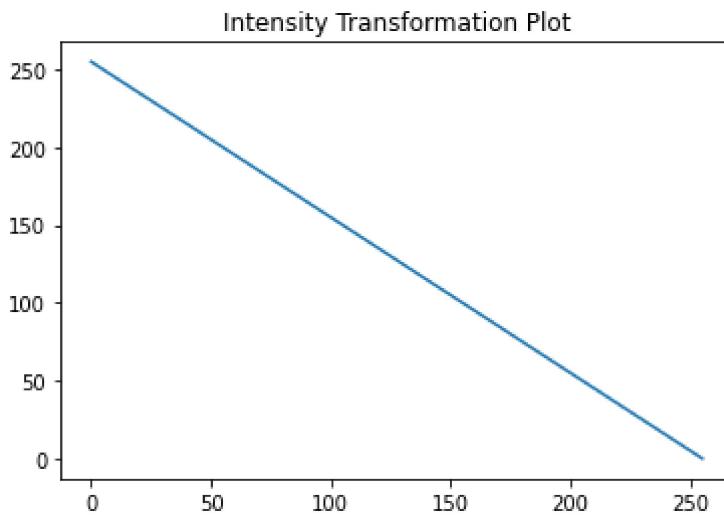
fig,ax = plt.subplots()
ax.plot(t)
ax.set_title("Intensity Transformation Plot")

#display using opencv
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',f)
cv.waitKey(0)
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()

#display using matplotlib
fig,ax = plt.subplots()
f_ = cv.cvtColor(f,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Original Image")
ax.imshow(f_)

fig,ax = plt.subplots()
g_ = cv.cvtColor(g,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Transformed Image")
ax.imshow(g_)
```

```
Out[ ]: <matplotlib.image.AxesImage at 0x231084130a0>
```



## Gemma Correction example in class

```
In [ ]: f = cv.imread("spider.png",cv.IMREAD_GRAYSCALE)
assert f is not None

gemma =0.5
t = np.array([(p/255)**gemma*255 for p in range(0,256)]).astype(np.uint8)
g = cv.LUT(f,t)

fig,ax = plt.subplots()
ax.plot(t)
ax.set_title("Intensity Transformation Plot")

#display using opencv
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',f)
cv.waitKey(0)
```

```

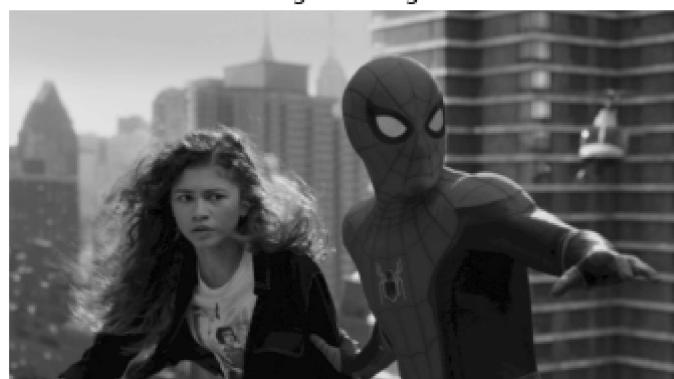
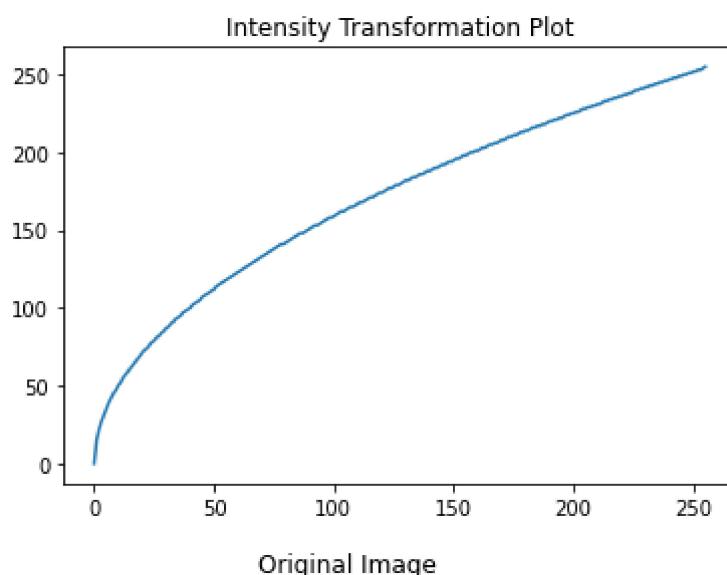
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()

#display using matplotlib
fig,ax = plt.subplots()
f_ = cv.cvtColor(f,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Original Image")
ax.imshow(f_)

fig,ax = plt.subplots()
g_ = cv.cvtColor(g,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Gemma corrected Image")
ax.imshow(g_)

```

Out[ ]: <matplotlib.image.AxesImage at 0x23107a61240>



# Exercise

## Question 1

```
In [ ]: f = cv.imread("spider.png",cv.IMREAD_COLOR)
assert f is not None

gemma =[0.2,0.8,1.2,2]
plt.rcParams['figure.figsize']=[200,50]
plt.rcParams['figure.autolayout']=True
fig,ax = plt.subplots()
f_ = cv.cvtColor(f,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Original Image")
ax.imshow(f_)

for i in range(1,5):
    t = np.array([(p/255)**gemma[i-1]*255 for p in range(0,256)]).astype(np.uint8)
    g = cv.LUT(f,t)
    fig,ax = plt.subplots()
    g_ = cv.cvtColor(g,cv.COLOR_BGR2RGB)
    ax.axis('off')
    ax.set_title("Gemma = "+str(gemma[i-1]))
    ax.imshow(g_)
```

Original Image



Gemma = 0.2



Gemma = 0.8



Gemma = 1.2



Gemma = 2



## Example by sir

```
In [ ]: f = cv.imread("spider.png",cv.IMREAD_GRAYSCALE)
assert f is not None

t1 = np.linspace(0,50,100)
t2 = np.linspace(50,200,50)
t3 = np.linspace(200,255,106)

t = np.concatenate((t1,t2,t3),axis = 0).astype(np.uint8)

fig,ax = plt.subplots()
ax.plot(t)
ax.set_title("Intensity Transformation Plot")
g = cv.LUT(f,t)

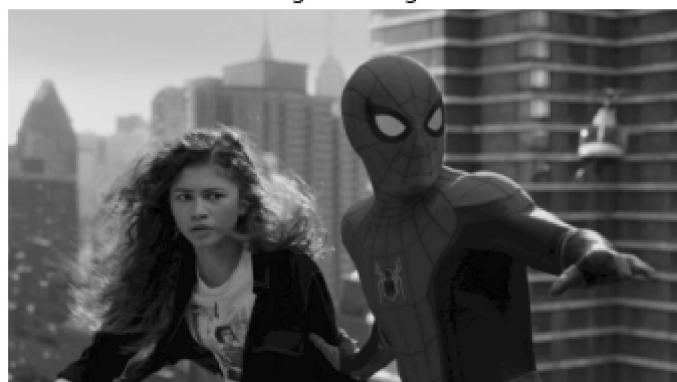
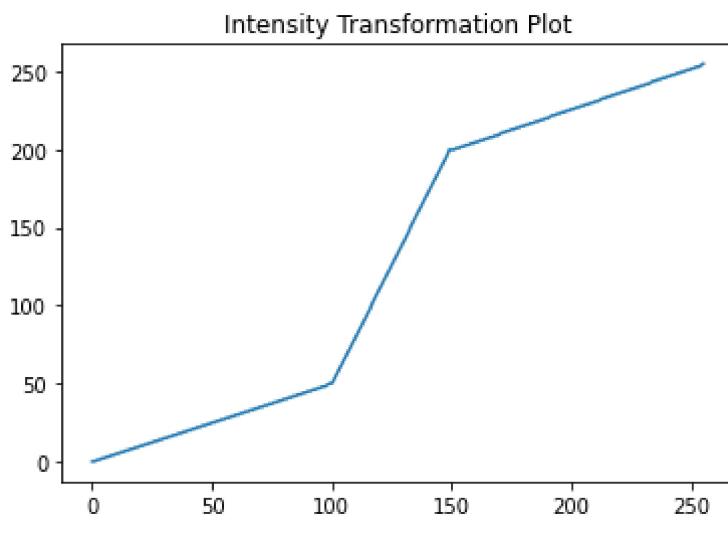
#display using opencv
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
```

```
cv.imshow('Image',f)
cv.waitKey(0)
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()

#display using matplotlib
fig,ax = plt.subplots()
f_ = cv.cvtColor(f,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Original Image")
ax.imshow(f_)

fig,ax = plt.subplots()
g_ = cv.cvtColor(g,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Transformed Image")
ax.imshow(g_)
```

Out[ ]: <matplotlib.image.AxesImage at 0x23106943790>



Transformed Image



## Question 2

```
In [ ]: f = cv.imread("spider.png",cv.IMREAD_GRAYSCALE)
assert f is not None

t1 = np.linspace(0,100,50)
t2 = np.linspace(100,255,150)
t3 = np.linspace(255,255,56)

t = np.concatenate((t1,t2,t3),axis = 0).astype(np.uint8)

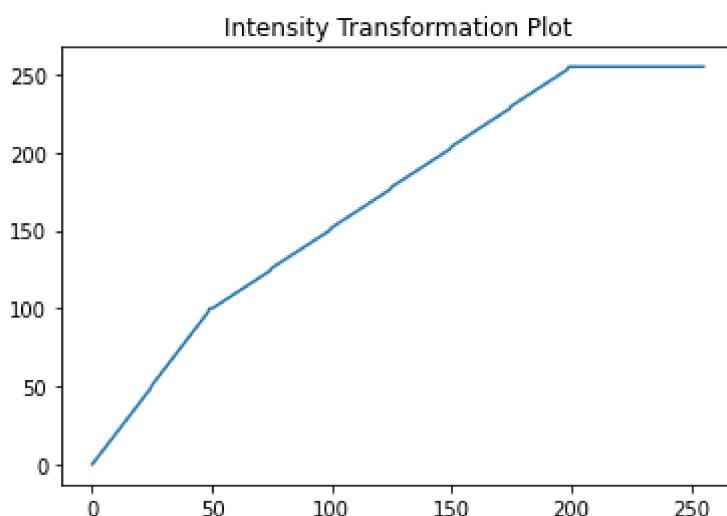
fig,ax = plt.subplots()
ax.plot(t)
ax.set_title("Intensity Transformation Plot")
g = cv.LUT(f,t)

#display using opencv
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',f)
cv.waitKey(0)
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()

#display using matplotlib
fig,ax = plt.subplots()
f_ = cv.cvtColor(f,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Original Image")
ax.imshow(f_)

fig,ax = plt.subplots()
g_ = cv.cvtColor(g,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Transformed Image")
ax.imshow(g_)
```

```
Out[ ]: <matplotlib.image.AxesImage at 0x231085ba5c0>
```



Original Image



Transformed Image



## Question 3

```
In [ ]: f = cv.imread("shells.tif",cv.IMREAD_GRAYSCALE)
assert f is not None

hist_f = cv.calcHist([f],[0],None,[256],[0,256])
g = cv.equalizeHist(f)
hist_g = cv.calcHist([g],[0],None,[256],[0,256])

fig,ax = plt.subplots(2,1,figsize=(8,8))
ax[0].plot(hist_f)
ax[0].set_title("Histogram of original Image")
ax[1].plot(hist_g)
ax[1].set_title("Histogram of equalized Image")

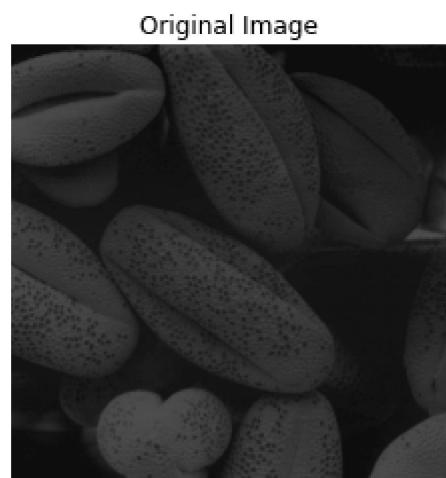
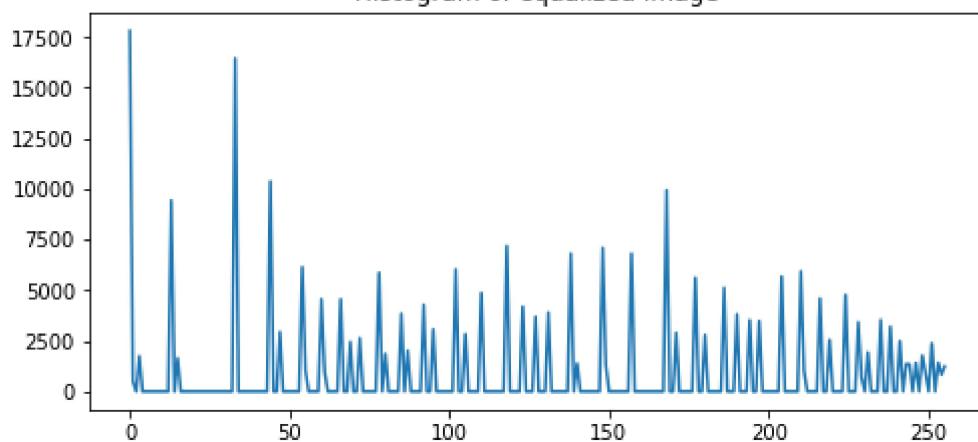
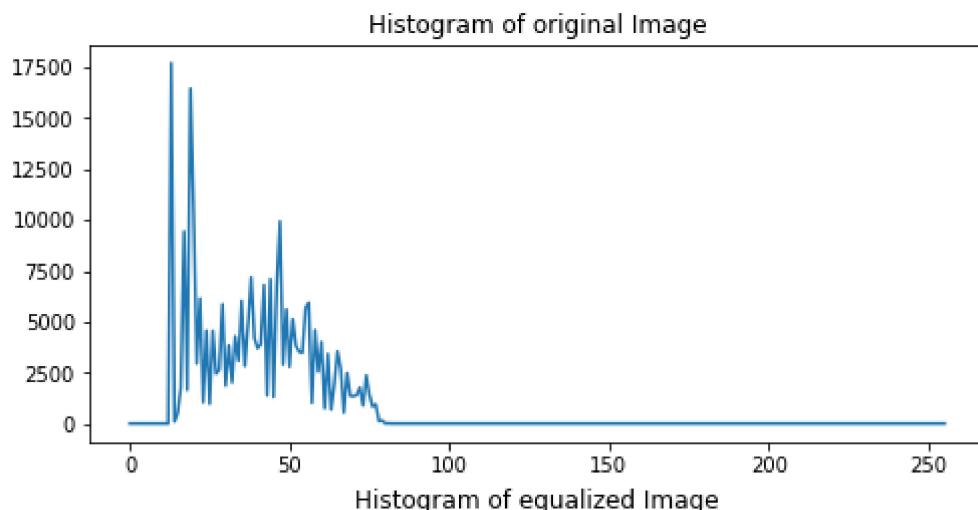
#display using opencv
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',f)
cv.waitKey(0)
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()

#display using matplotlib
fig,ax = plt.subplots()
f_ = cv.cvtColor(f,cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Original Image")
ax.imshow(f_)

fig,ax = plt.subplots()
g_ = cv.cvtColor(g,cv.COLOR_BGR2RGB)
ax.axis('off')
```

```
ax.set_title("Equalized Image")
ax.imshow(g_)
```

Out[ ]: <matplotlib.image.AxesImage at 0x2310971ad70>



Equalized Image



## Question 4

```
In [ ]: m = cv.imread("zion_pass.jpg",cv.IMREAD_COLOR)
assert f is not None

m_hsv = cv.cvtColor(m, cv.COLOR_BGR2HSV)
h = m_hsv[:, :, 0]
s = m_hsv[:, :, 1]
v = m_hsv[:, :, 2]

s_add = cv.add(s, 50)
h_add = cv.add(h, 142)

saturated_m = cv.cvtColor(cv.merge([h, s_add, v]), cv.COLOR_HSV2BGR)
hue_m = cv.cvtColor(cv.merge([h_add, s, v]), cv.COLOR_HSV2BGR)

#display using opencv
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',m)
cv.waitKey(0)
cv.destroyAllWindows()

cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',saturated_m)
cv.waitKey(0)
cv.destroyAllWindows()

cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',hue_m)
cv.waitKey(0)
cv.destroyAllWindows()

#display using matplotlib
fig,ax = plt.subplots()
m_ = cv.cvtColor(m, cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Original Image")
ax.imshow(m_)

fig,ax = plt.subplots()
saturated_m_ = cv.cvtColor(saturated_m, cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Saturation Increased Image")
ax.imshow(saturated_m_)

fig,ax = plt.subplots()
```

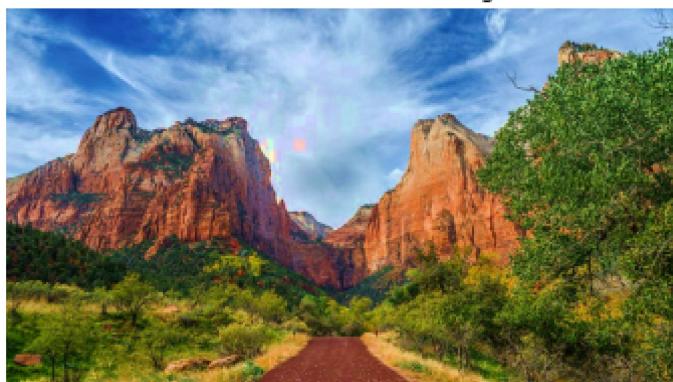
```
hue_m_ = cv.cvtColor(hue_m, cv.COLOR_BGR2RGB)
ax.axis('off')
ax.set_title("Hue Changed Image")
ax.imshow(hue_m_)
```

Out[ ]: <matplotlib.image.AxesImage at 0x2310b8e88e0>

Original Image



Saturation Increased Image



Hue Changed Image

