EN2550 - Fundementals of Image Processing and Machine Vision

Name : R.G.S.M. RANATUNGA

Index No. : 190504H

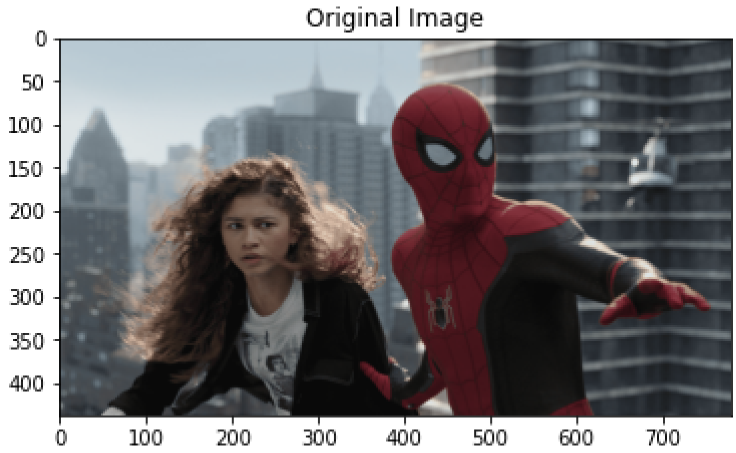
In [ ]:

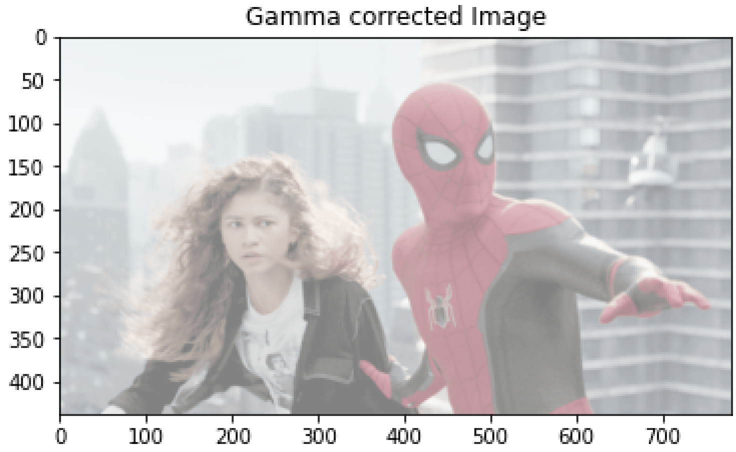
```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
```

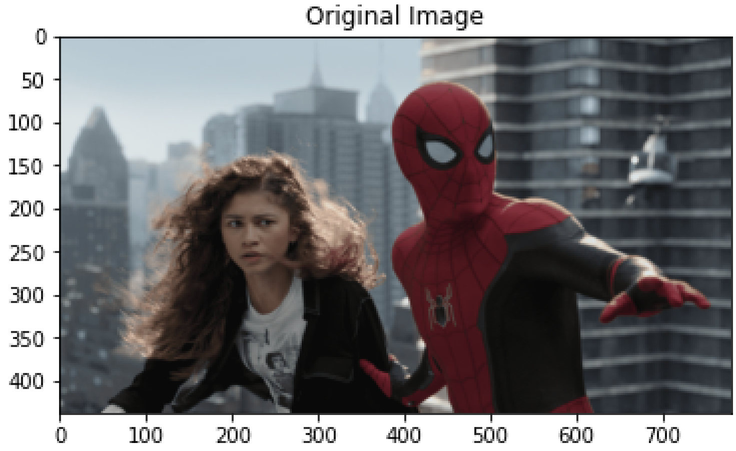
Part 1

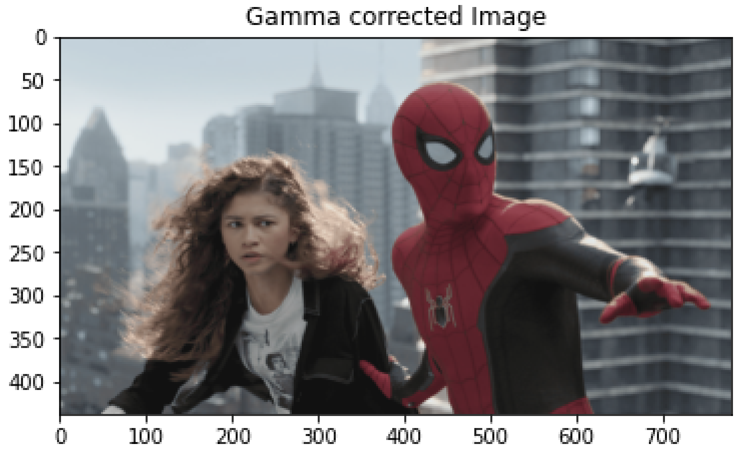
In [ ]:

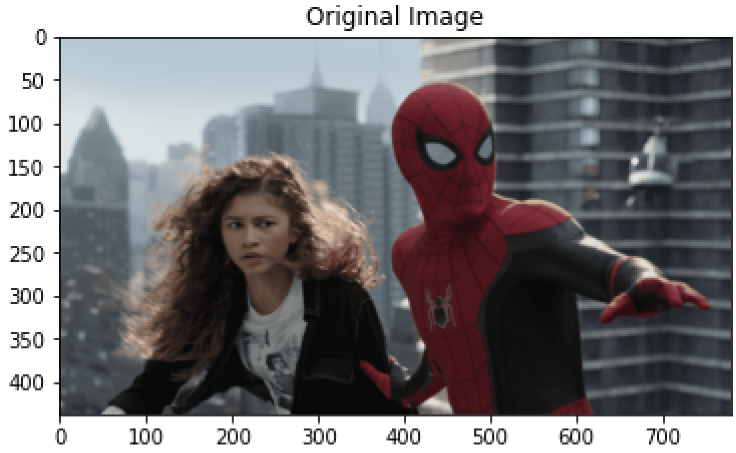
```
img_orig = cv.imread(r'./spider.png', cv.IMREAD_COLOR)
assert img_orig is not None
gamma = [0.2, 0.8, 1.2, 2]


for r in range(len(gamma)):
    table = np.array([(i/255.0)**(gamma[r])*255.0 for i in np.arange(0,256)]).astype

    img_gamma = cv.LUT(img_orig, table)

    fig,ax = plt.subplots()
    ax.imshow(cv.cvtColor(img_orig, cv.COLOR_BGR2RGB))
    plt.title("Original Image")
    plt.show()

    fig,ax = plt.subplots()
    ax.imshow(cv.cvtColor(img_gamma, cv.COLOR_BGR2RGB))
    plt.title("Gamma corrected Image")
    plt.show()

    cv.imshow('Image',img_orig)
    cv.waitKey(0)

    cv.imshow('Image',img_gamma)
    cv.waitKey(0)

    cv.destroyAllWindows()
```

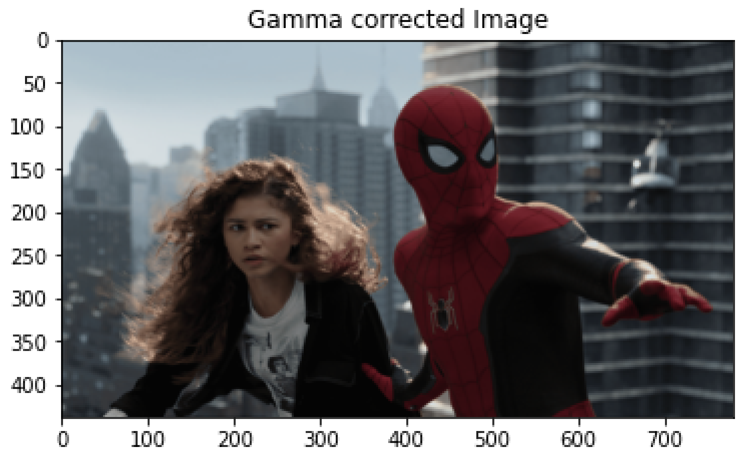


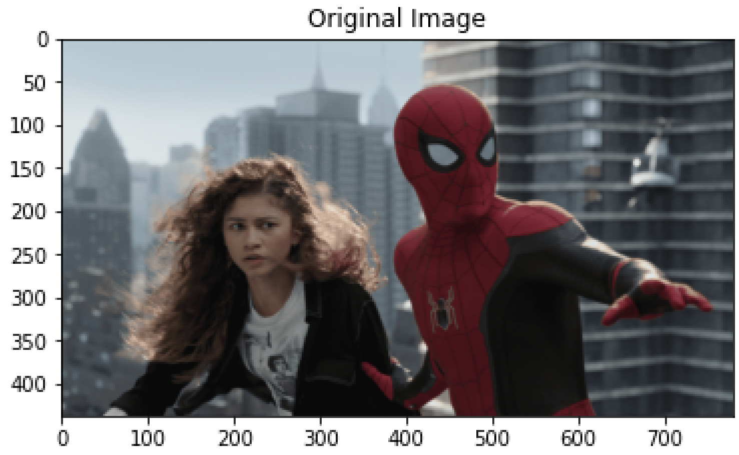

Gamma corrected Image
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700

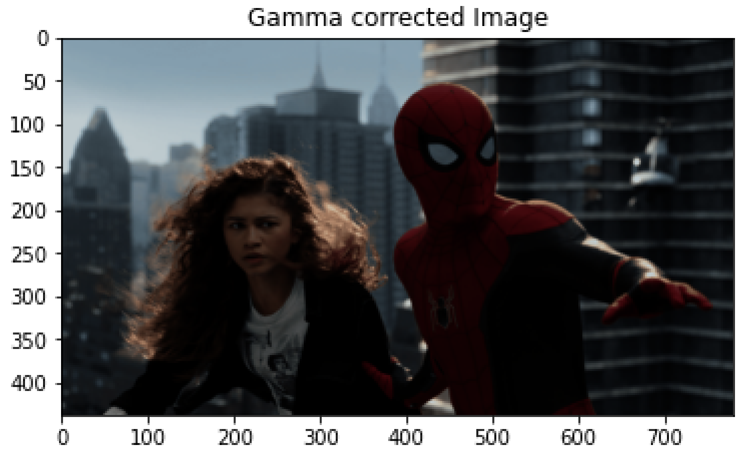

Original Image
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700


Gamma corrected Image
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700


Original Image
0
50
100
150
200
250
300
350
400
0
100
200
300
400
500
600
700





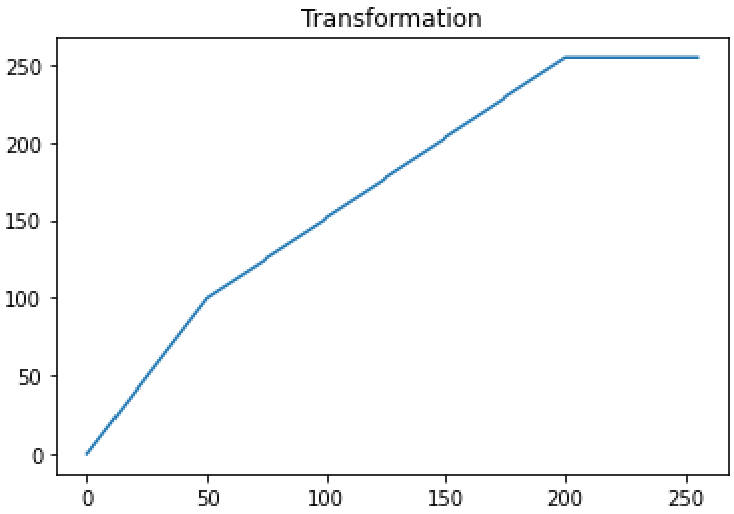


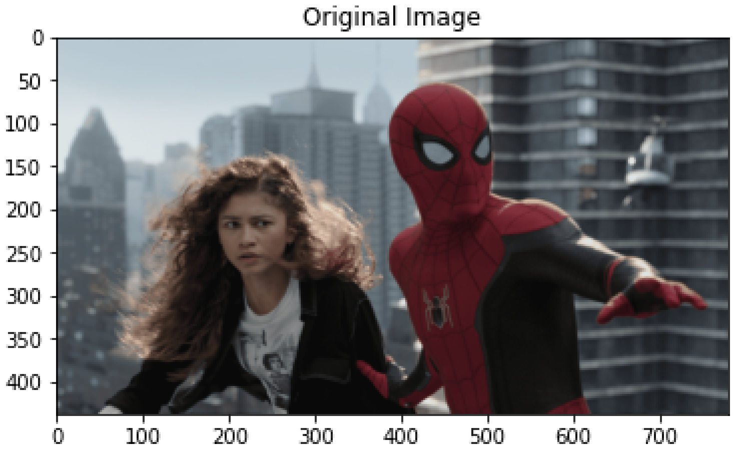
Part 2

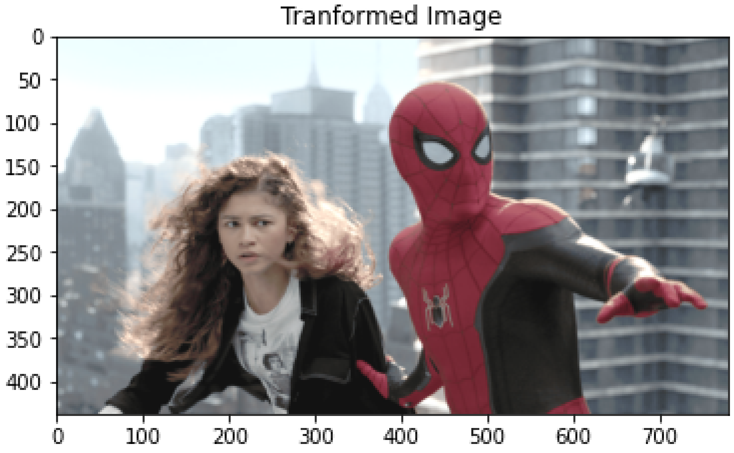
In [ ]:

```
img_orig = cv.imread(r'./spider.png', cv.IMREAD_COLOR)
assert img_orig is not None

t = np.arange(0,256,dtype=np.uint8)
t[:50] = np.linspace(0,100,50,endpoint=False)
t[50:200] = np.linspace(100,256,150,endpoint=False)
t[200:] = 255

fig,ax = plt.subplots()
ax.plot(t)
plt.title("Transformation")

img_t = cv.LUT(img_orig,t)

fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(img_orig, cv.COLOR_BGR2RGB))
```

```
plt.title("Original Image")
plt.show()

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

fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(img_t, cv.COLOR_BGR2RGB))
plt.title("Tranformed Image")
plt.show()

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



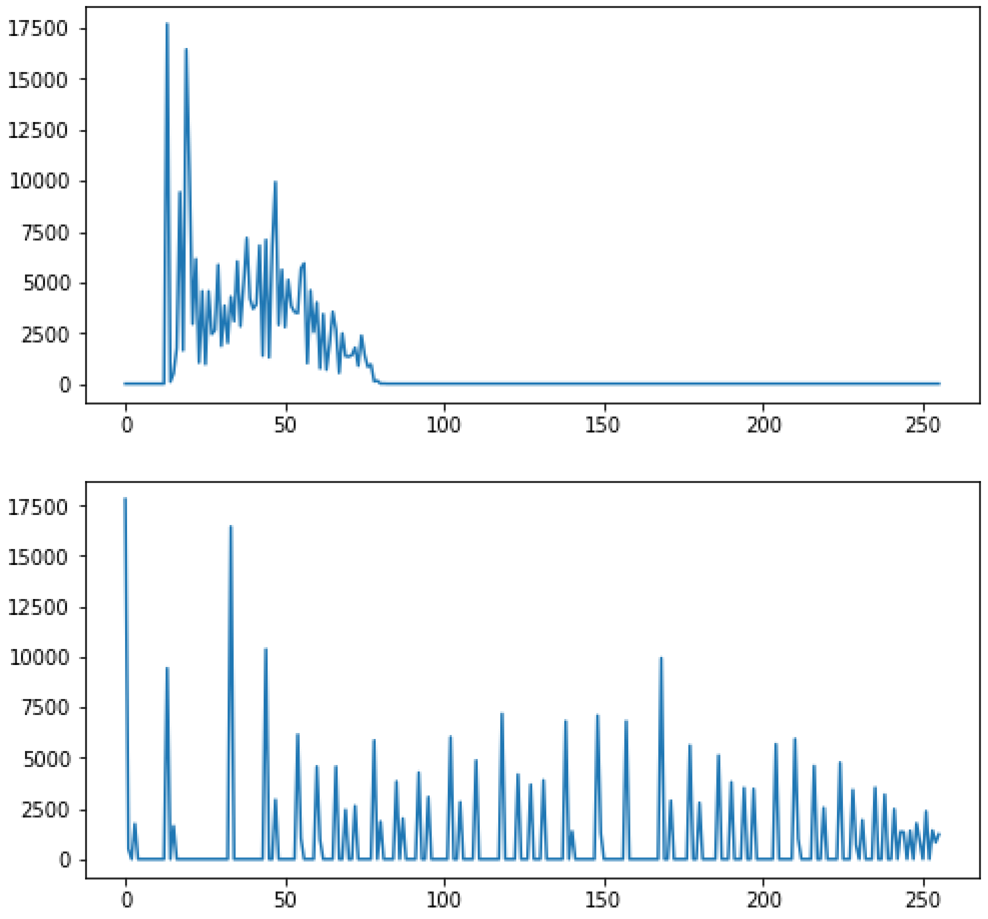


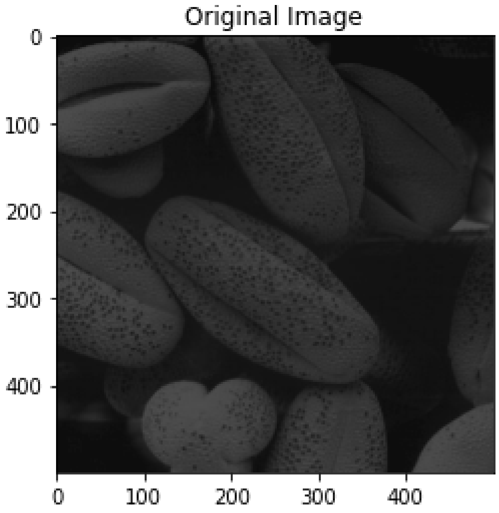


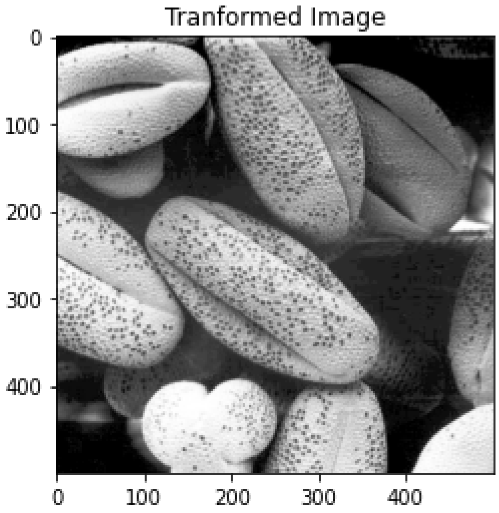
Part 3 a) & b)

In [ ]:

```
f = cv.imread(r'./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[1].plot(hist_g)

fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB))
plt.title("Original Image")
plt.show()

fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB))
plt.title("Tranformed Image")
plt.show()

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







Part 4 a) & b)

```
In [ ]:
img_orig = cv.imread(r'./zion_pass.jpg', cv.IMREAD_COLOR)
assert img_orig is not None

fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(img_orig, cv.COLOR_BGR2RGB))
plt.title("Original Image")
plt.show()

img_hsv = cv.cvtColor(img_orig, cv.COLOR_BGR2HSV).astype("float32")

(h, s, v) = cv.split(img_hsv)

s = s*4 #Saturation
s = np.clip(s,0,255)
imghsv_sat = cv.merge([h,s,v])
img_sat = cv.cvtColor(imghsv_sat.astype("uint8"), cv.COLOR_HSV2BGR)


h = h*3 #Hue
h = np.clip(h,0,255)
imghsv_hue = cv.merge([h,s,v])
img_hue = cv.cvtColor(imghsv_hue.astype("uint8"), cv.COLOR_HSV2BGR)

fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(img_sat, cv.COLOR_BGR2RGB))
plt.title("Saturated Image")
plt.show()
```

```
fig,ax = plt.subplots()
ax.imshow(cv.cvtColor(img_hue, cv.COLOR_BGR2RGB))
plt.title("Hued Image")
plt.show()
```