

Name: Treshan Ayesh

Index: 190443T

In [ ]:

```
#Q1)
%matplotlib inline
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

f = cv.imread('spider.png')
assert f is not None

gamma = [0.2, 0.8, 1.2, 2]
corrected_images = []
for i in gamma:
    t = np.array([(p/255)**i*255 for p in range(0,256)]).astype('uint8')
    corrected_images.append(t)

new_images = []
for j in corrected_images:
    image_transformed = cv.LUT(f, j) #LUT - LookUpTable
    new_images.append(image_transformed)
#g = t[f] #same as above using numpy arrays

fig,ax = plt.subplots(4,3, figsize = (20,20))
for val,image in enumerate(new_images):

    ax[val][0].plot(corrected_images[val])

    im_rgb = cv.cvtColor(image, cv.COLOR_BGR2RGB)
    oimg_rgb = cv.cvtColor(f, cv.COLOR_BGR2RGB)

    ax[val][1].imshow(oimg_rgb)
    ax[val][2].imshow(im_rgb)

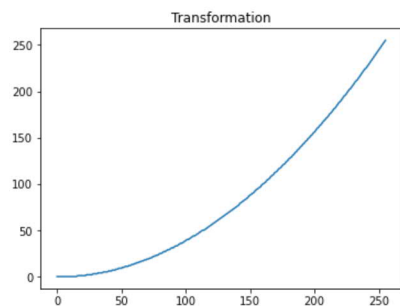
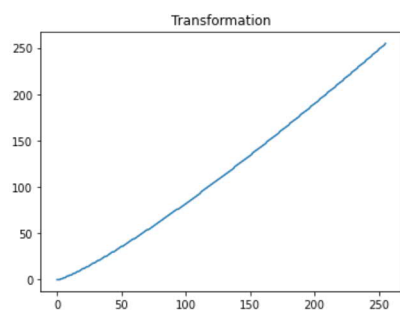
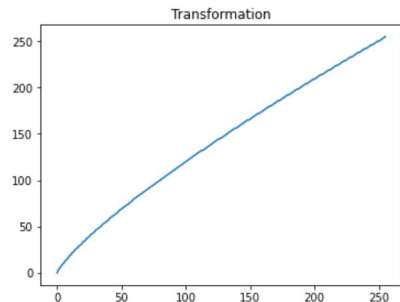
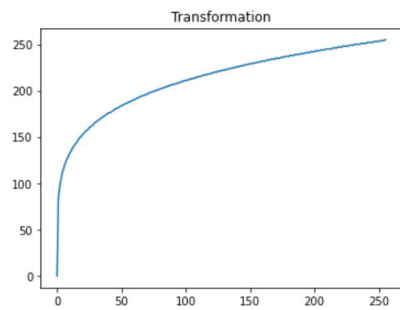
    ax[val][0].set_title('Transformation')
    ax[val][1].set_title('Original Image')
    ax[val][2].set_title('Gamma corrected Image  $\gamma=' + \text{str}(\text{gamma}[\text{val}])$ )

    for i in range(2):
        for j in range(4):
            ax[j][i+1].set_xticks([])
            ax[j][i+1].set_yticks([])

    # cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
    # cv.namedWindow('Image2', cv.WINDOW_AUTOSIZE)

    # cv.imshow('Image', f)
    # cv.imshow('Image2', image)

    # cv.waitKey(0)
    # cv.destroyAllWindows()
```



In [ ]:

```
#Q2)
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

f = cv.imread('spider.png')
assert f is not None

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

transform = np.concatenate((t1,t2,t3), axis = 0).astype(np.uint8)
transformed_img = cv.LUT(f, transform) #LUT - LookUpTable
#g = t[f] #same as above using numpy arrays

fig,ax = plt.subplots(1,3, figsize = (20,20))
ax[0].plot(transform)
```

```

ax[1].imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB))
ax[2].imshow(cv.cvtColor(transformed_img, cv.COLOR_BGR2RGB))

ax[0].set_aspect('equal')
ax[1].set_aspect('equal')
ax[2].set_aspect('equal')

ax[0].set_title('Transformation')
ax[1].set_title('Original Image')
ax[2].set_title('Transformed Image')

ax[1].set_xticks([])
ax[1].set_yticks([])
ax[2].set_xticks([])
ax[2].set_yticks([])

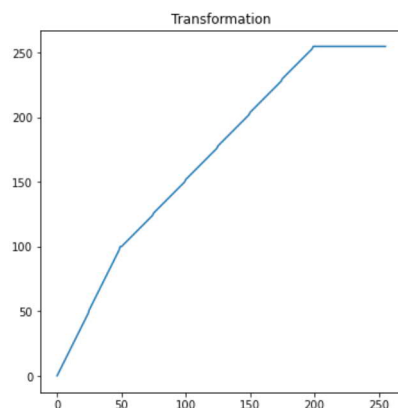
# cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
# cv.namedWindow('Image2', cv.WINDOW_AUTOSIZE)

# cv.imshow('Image', f)
# cv.imshow('Image2', g)

# cv.waitKey(0)
# cv.destroyAllWindows()

```

Out[ ]: []



In [ ]:

```

#Q3)
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

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(2,2, figsize = (16,16))
ax[0][0].plot(hist_f)
ax[1][0].plot(hist_g)

ax[0][1].imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB))

```

```
ax[1][1].imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB))

ax[0][0].set_title('Histogram of original image')
ax[1][0].set_title('Histogram of equalized image')
ax[0][1].set_title('Original image')
ax[1][1].set_title('Equalized image')

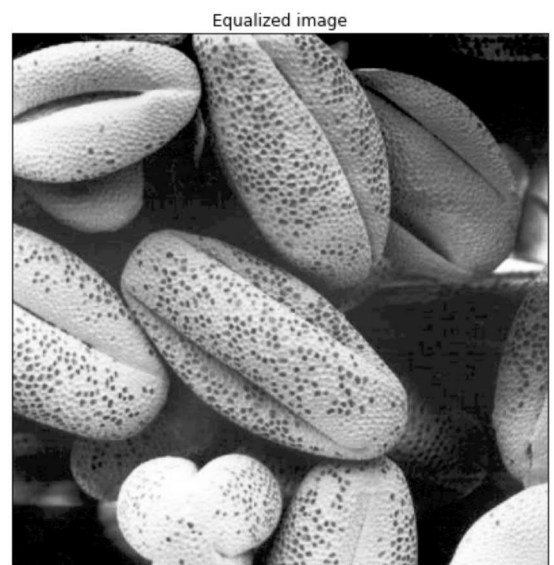
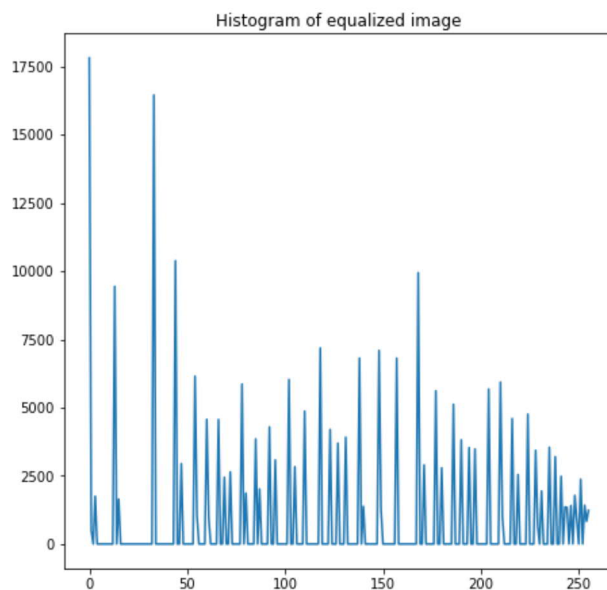
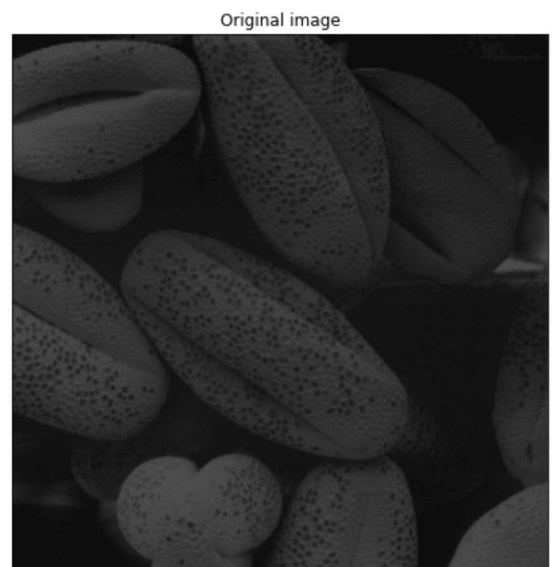
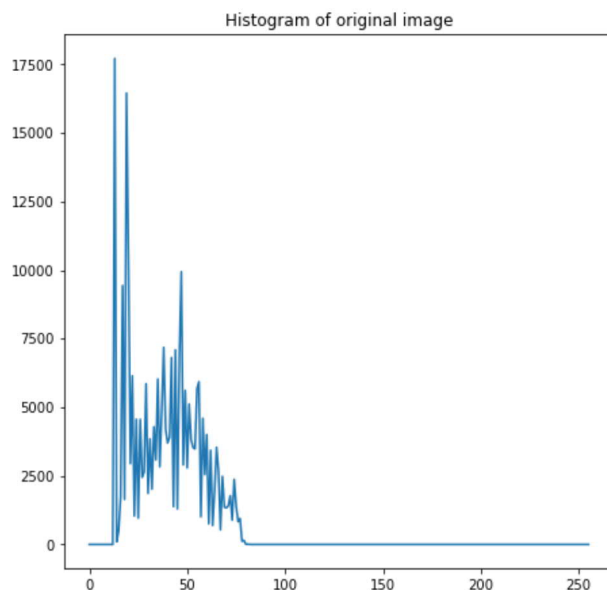
ax[0][1].set_xticks([])
ax[0][1].set_yticks([])
ax[1][1].set_xticks([])
ax[1][1].set_yticks([])

# cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
# cv.namedWindow('Image2', cv.WINDOW_AUTOSIZE)

# cv.imshow('Image', f)
# cv.imshow('Image2', g)

# cv.waitKey(0)
# cv.destroyAllWindows()
```

Out[ ]: []



In [ ]:

```
#Q4)
import cv2
import numpy as np

image = cv2.imread('zion_pass.jpg')
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

h,s,v = cv2.split(hsv)

s_new = cv2.add(s,20)
sat_increased = cv2.merge([h,s_new,v])

h_new = cv2.add(h,20)
hue_increased = cv2.merge([h_new,s,v])

fig,ax = plt.subplots(3,1, figsize = (16,16))
ax[0].imshow(cv.cvtColor(image, cv.COLOR_BGR2RGB))
ax[1].imshow(cv.cvtColor(sat_increased, cv.COLOR_HSV2RGB))
ax[2].imshow(cv.cvtColor(hue_increased, cv.COLOR_HSV2RGB))
```

```
ax[0].set_title('Original image')
ax[1].set_title('Increased Saturation')
ax[2].set_title('Increased Hue')

for i in range(3):
    ax[i].set_xticks([])
    ax[i].set_yticks([])
```



Original image



Increased Saturation



Increased Hue

