

```
In [ ]: import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

im=cv.imread('D:\Semester 4\Image processing\Homework2\spider.png',cv.IMREAD_GRAYSCALE)
assert im is not None

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

gamma = [0.2,0.8,1,1.2,2]

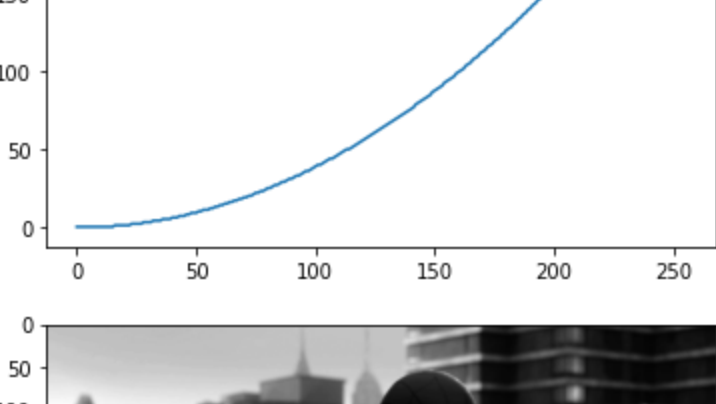
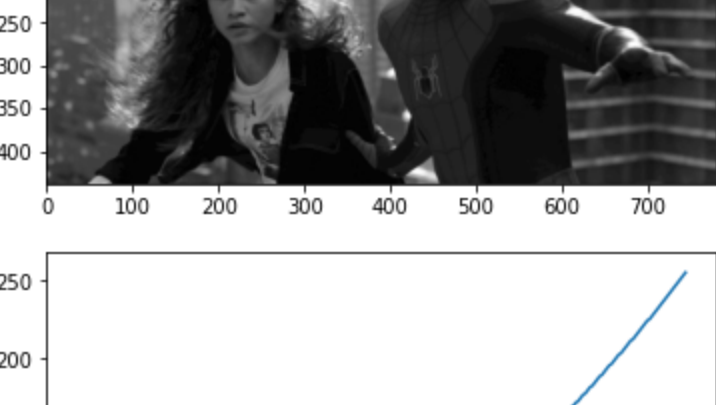
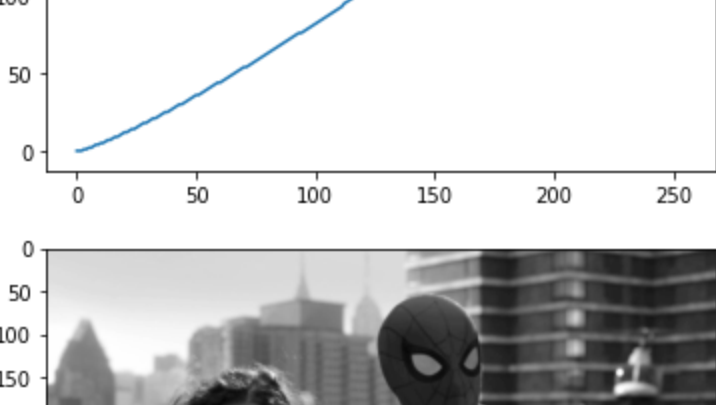
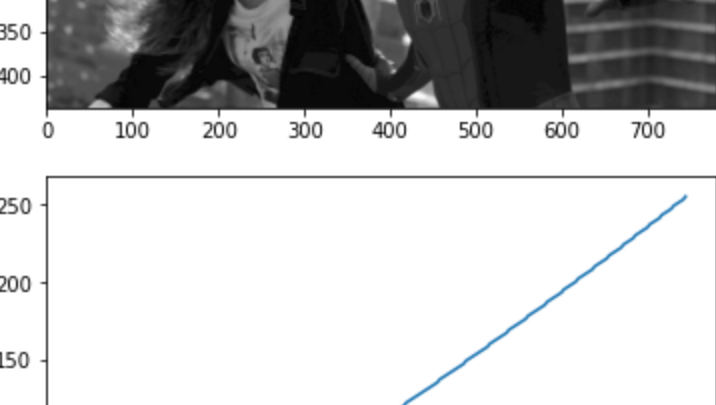
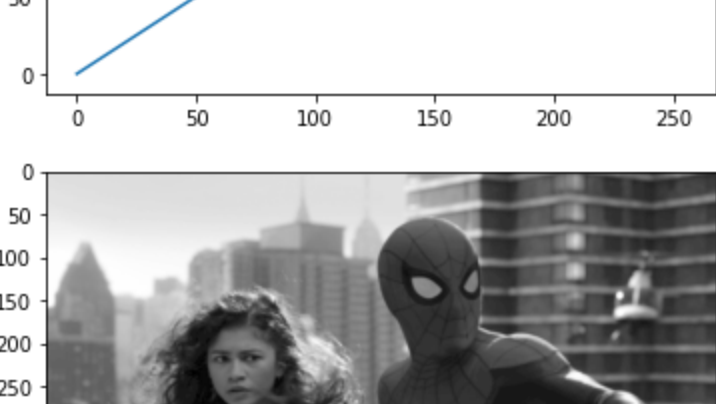
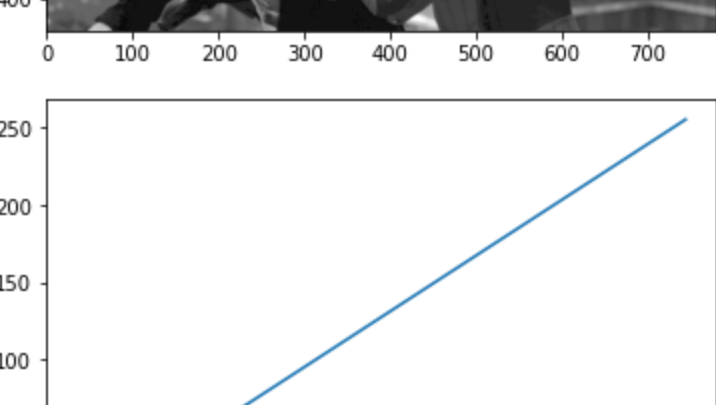
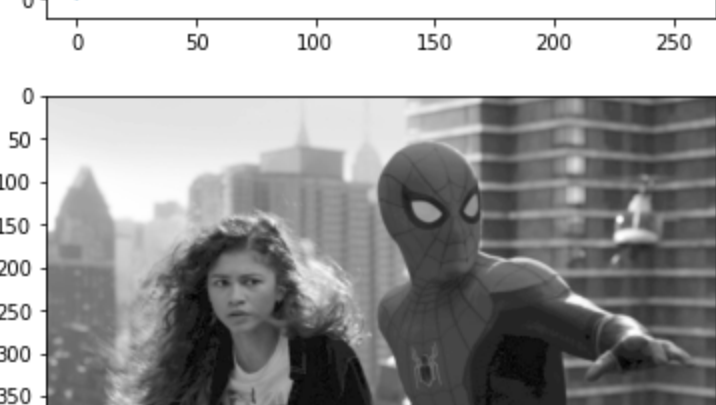
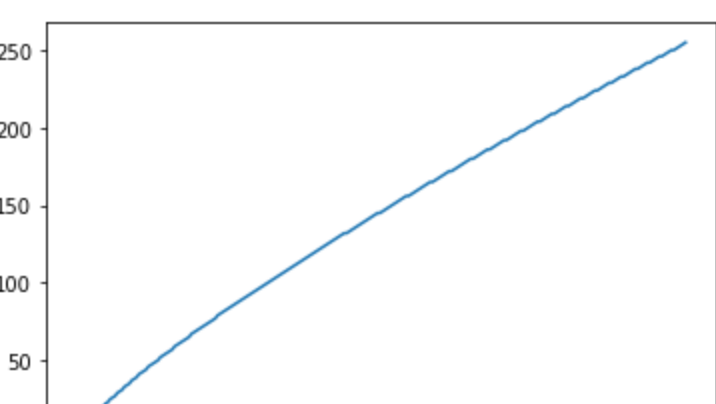
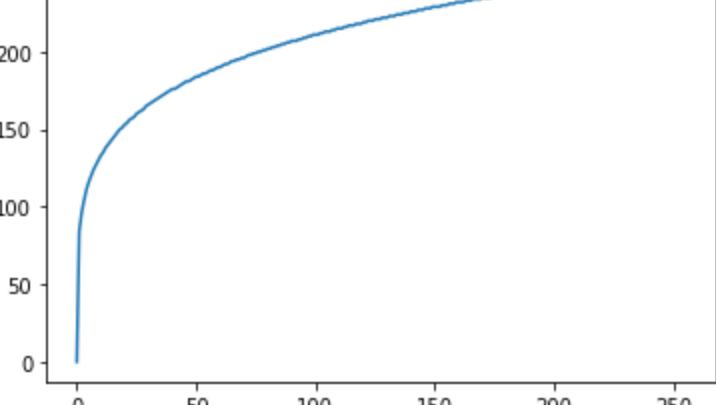
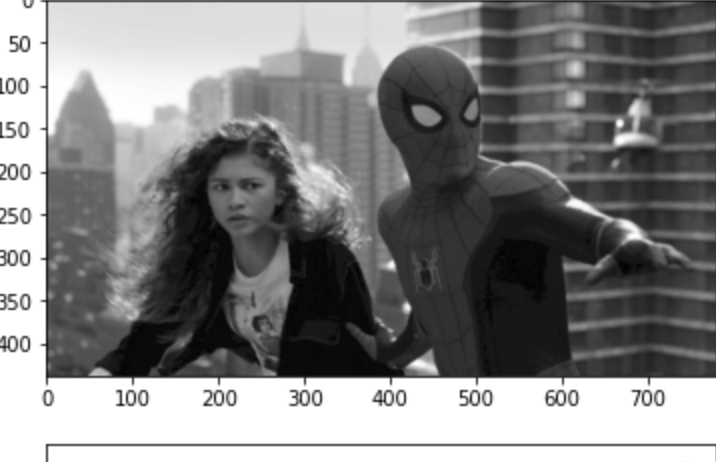
fig,ax=plt.subplots()
plt.imshow(im,'gray')
plt.title('Original image in grayscale')

for i in gamma:
    t = np.array([(p/255)**i*255 for p in range (0,256)]).astype(np.uint8)
    g = cv.LUT(im,t)

    fig,ax=plt.subplots()
    ax.plot(t)

    fig=plt.subplots()
    plt.imshow(g,'gray')
    plt.show()

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



```
In [ ]: t1=np.linspace (0,100,50)
t2= np.linspace (100,255,150)
t3= np.linspace (255,255,50)

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

fig,ax=plt.subplots()
ax.plot(t)
ax.set_aspect('equal')
```

```
assert len(t)==256
g= cv.LUT(im,t)
```

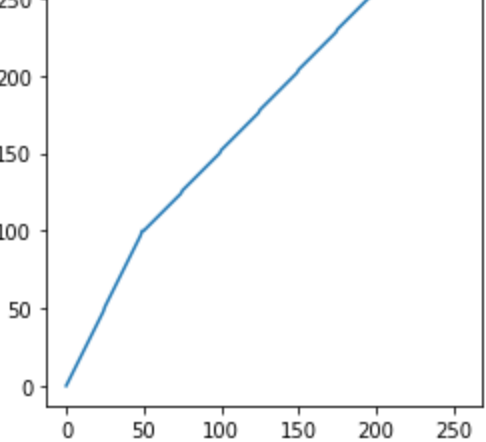
```
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()
```

```
fig,ax=plt.subplots(2,1,figsize=(10,8))
plt.subplot(2,1,1)
plt.imshow(im,'gray')
```

```
plt.title('Original')
plt.subplot(2,1,2)
plt.imshow(g,'gray')
```

```
plt.title('Intensity transform image')
```

```
Out[ ]: Text(0.5, 1.0, 'Intensity transform image')
```



Original



Intensity transform image



```
In [ ]: import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

im=cv.imread('D:\Semester 4\Image processing\Homework2\shell1.tif',cv.IMREAD_GRAYSCALE)
assert im is not None

hist_in = cv.calcHist([im], [0], None, [256], [0,256])
g = cv.equalizeHist(im)
hist_g = cv.calcHist([g], [0], None, [256], [0,256])

fig, ax = plt.subplots(2,1,figsize= (8,8))
ax[0].plot(hist_in)
ax[1].plot(hist_g)
```

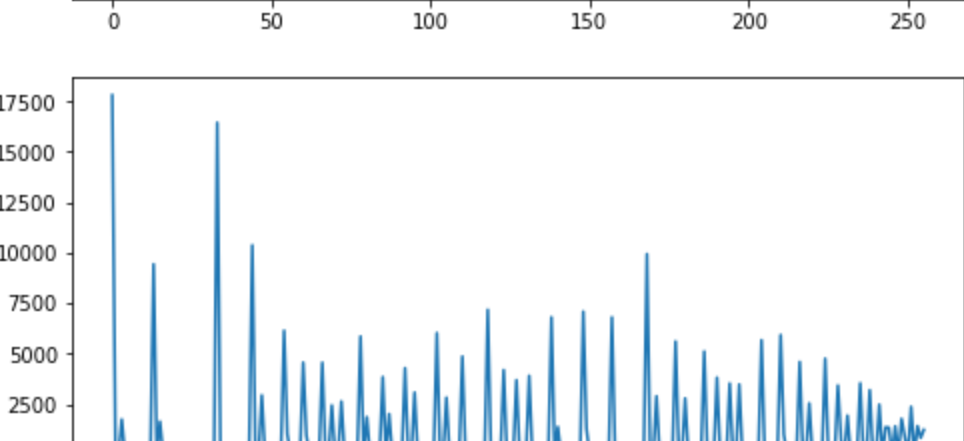
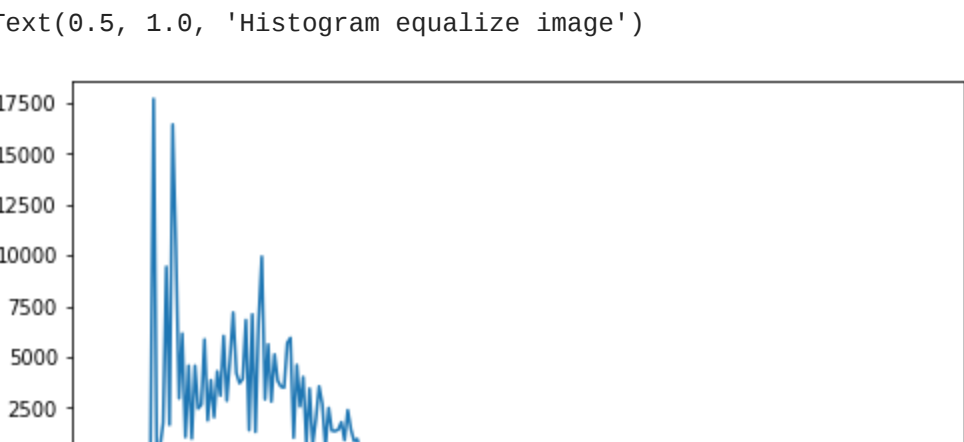
```
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',g)
cv.waitKey(0)
cv.destroyAllWindows()
```

```
fig=plt.subplots(2,1,figsize=(10,8))
plt.subplot(2,1,1)
plt.imshow(im,'gray')
```

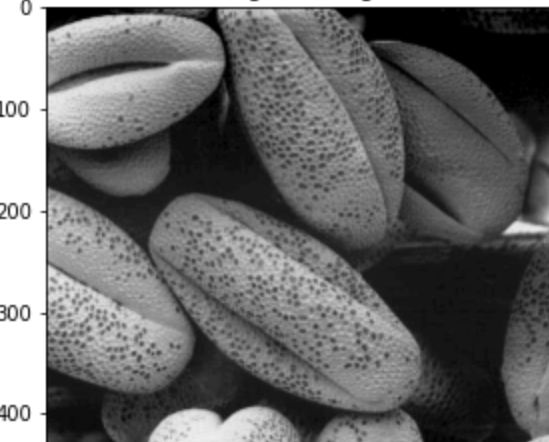
```
plt.title('Original image')
plt.subplot(2,1,2)
plt.imshow(g,'gray')
```

```
plt.title('Histogram equalize image')
```

```
Out[ ]: Text(0.5, 1.0, 'Histogram equalize image')
```



Original image



Histogram equalize image



```
In [ ]: im=cv.imread('D:\Semester 4\Image processing\Homework2\ Zion_pass.jpg')
```

```
assert im is not None
```

```
hsv0_in = cv.cvtColor(im, cv.COLOR_BGR2HSV)
```

```
(h0, s0, v0) = cv.split(hsv0_in)
```

```
s1= np.linspace (0,100,75)
```

```
s2= np.linspace (100,225,125)
```

```
s3= np.linspace (225,255, 56)
```

```
s= np.concatenate((s1,s2,s3), axis =0).astype(np.uint8)
```

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

```
ax.plot(s)
```

```
ax.set_aspect('equal')
```

```
assert len(s)==256
```

```
snew= cv.LUT(s0,s)
```

```
hsv1_in = cv.merge((h0,snew,v0))
```

```
bgr_in1 = cv.cvtColor(hsv1_in, cv.COLOR_HSV2BGR)
```

```
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
```

```
cv.imshow('Image',bgr_in1)
```

```
cv.waitKey(0)
```

```
cv.destroyAllWindows()
```

```
fig=plt.subplots(2,1,figsize=(10,8))
```

```
plt.subplot(2,1,1)
```

```
plt.imshow(cv.cvtColor(hsv0_in, cv.COLOR_HSV2RGB))
```

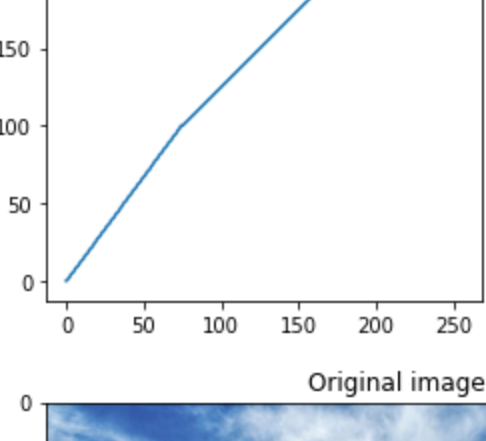
```
plt.title('Original image')
```

```
plt.subplot(2,1,2)
```

```
plt.imshow(cv.cvtColor(hsv1_in, cv.COLOR_HSV2RGB))
```

```
plt.title('Saturation transformed image')
```

```
Out[ ]: Text(0.5, 1.0, 'Saturation transformed image')
```



Original image



Saturation transformed image



```
In [ ]: im=cv.imread('D:\Semester 4\Image processing\Homework2\ Zion_pass.jpg')
```

```
assert im is not None
```

```
hsv0_in = cv.cvtColor(im, cv.COLOR_BGR2HSV)
```

```
(h0, s0, v0) = cv.split(hsv0_in)
```

```
h1= np.linspace (0,75,25)
```

```
h2= np.linspace (75,100,50)
```

```
h3= np.linspace (100,150, 100)
```

```
h4= np.linspace (150,255, 75)
```

```
h= np.concatenate((h1,h2,h3,h4), axis =0).astype(np.uint8)
```

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

```
ax.plot(h)
```

```
ax.set_aspect('equal')
```

```
assert len(h)==256
```

```
hnew= cv.LUT(h0,h)
```

```
hsv2_in = cv.merge((hnew,s0,v0))
```

```
bgr_in2 = cv.cvtColor(hsv2_in, cv.COLOR_HSV2BGR)
```

```
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
```

```
cv.imshow('Image',bgr_in2)
```

```
cv.waitKey(0)
```

```
cv.destroyAllWindows()
```

```
fig=plt.subplots(2,1,figsize=(10,8))
```

```
plt.subplot(2,1,1)
```

```
plt.imshow(cv.cvtColor(hsv0_in, cv.COLOR_HSV2RGB))
```

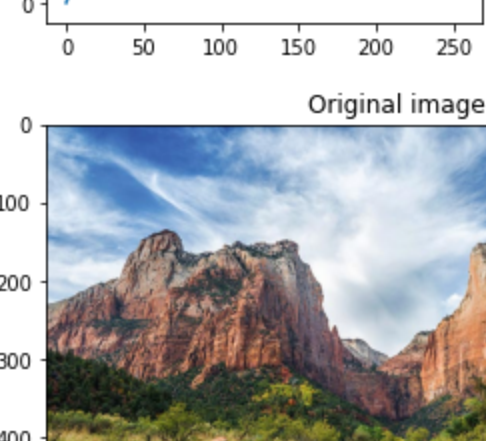
```
plt.title('Original image')
```

```
plt.subplot(2,1,2)
```

```
plt.imshow(cv.cvtColor(hsv2_in, cv.COLOR_HSV2RGB))
```

```
plt.title('Hue transformed image')
```

```
Out[ ]: Text(0.5, 1.0, 'Hue transformed image')
```



Original image

