```python
import cv2
import numpy as np
import matplotlib.pyplot as plt
fig, ax = plt.subplots(2,2,figsize=(8, 8))
img = cv2.imread('spider.png')
for gamma in [0.2, 0.8, 1.2, 2]:
    gamma_corrected = np.array(255*(img / 255) ** gamma, dtype = 'uint8')
    cv2.imwrite('gamma_transformed'+str(gamma)+'.jpg', gamma_corrected)
img1 = cv.imread ( 'gamma_transformed0.2.jpg',cv.IMREAD_COLOR )
img2 = cv.imread ( 'gamma_transformed0.8.jpg',cv.IMREAD_COLOR )
img3 = cv.imread ( 'gamma_transformed1.2.jpg',cv.IMREAD_COLOR )
img4 = cv.imread ( 'gamma_transformed2.jpg',cv.IMREAD_COLOR )
ax[0,0].imshow(cv.cvtColor(img1, cv.COLOR_BGR2RGB))
ax[0,1].imshow(cv.cvtColor(img2, cv.COLOR_BGR2RGB))
ax[1,0].imshow(cv.cvtColor(img3, cv.COLOR_BGR2RGB))
ax[1,1].imshow(cv.cvtColor(img4, cv.COLOR_BGR2RGB))
plt.show()
```
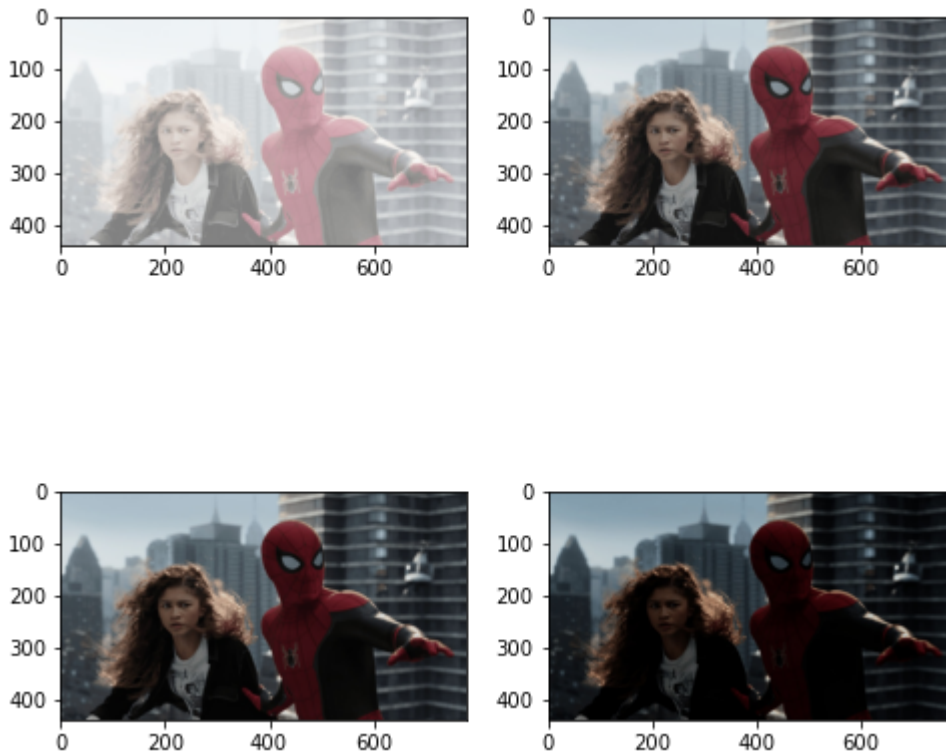
```python
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
c = np.array ( [ ( 50 , 100) , ( 200 , 255) ] )
t1 = np.linspace(0, c[0,1], c[0,0] + 1 - 0).astype('uint8')
t2 = np.linspace(c[0,1] + 1, c[1,1], c[1,0] - c[0,0]).astype('uint8')
t3 = np.linspace(c[1,1] , 255, 255 - c[1,0]).astype('uint8')
transform = np.concatenate((t1, t2), axis=0).astype('uint8')
transform = np.concatenate((transform, t3), axis=0).astype('uint8')
fig , ax = plt.subplots( )
ax.plot( transform )
ax.set_title('Intensity transformation')
ax.set_xlabel( r'Input, $f(\mathbf{x})$' )
ax.set_ylabel('Output, $\mathrm{T}[f(\mathbf{x})]$')
```

```
ax.set_xlim( 0 , 255 )
ax.set_ylim( 0 , 255 )
ax.set_aspect( 'equal' )
plt.savefig( 'transform.png' )
plt.show( )
img_orig = cv.imread ( 'spider.png' , cv.IMREAD_GRAYSCALE )
cv.namedWindow( " Image " , cv.WINDOW_AUTOSIZE )
cv.imshow( " Image " , img_orig )
cv.waitKey( 0 )
image_transformed = cv.LUT( img_orig , transform )
cv.imshow( " Image " , image_transformed )
cv.imwrite('image_transformed.jpg', image_transformed)
cv.waitKey( 0 )
cv.destroyAllWindows( )
fig , ax = plt.subplots( )
img1 = cv.imread ( 'image_transformed.jpg',cv.IMREAD_COLOR )
ax.imshow(cv.cvtColor(img1, cv.COLOR_BGR2RGB))
ax.set_title('Image_transformed')
plt.show()
```


Intensity transformation

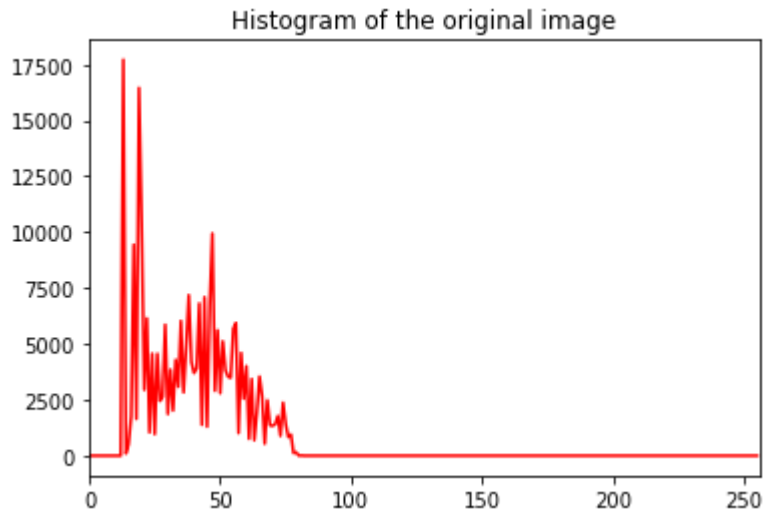Image_transformed
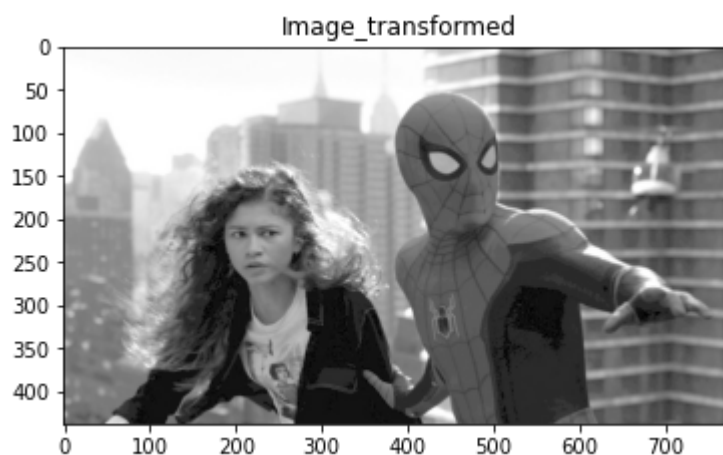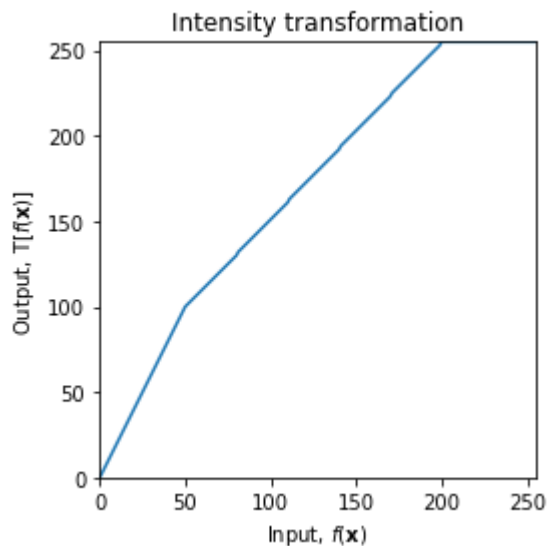
In [64]:

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
img = cv . imread ( 'shells.tif' ,cv.IMREAD_COLOR )
color = ( 'b' , 'g' , 'r' )
for i, c in enumerate(color):
```

```
        hist = cv.calcHist([img], [i], None, [256], [0,256])
plt.plot( hist , color = c )
plt.xlim( [ 0, 256 ] )
plt.title('Histogram of the original image')
plt.show( )
```
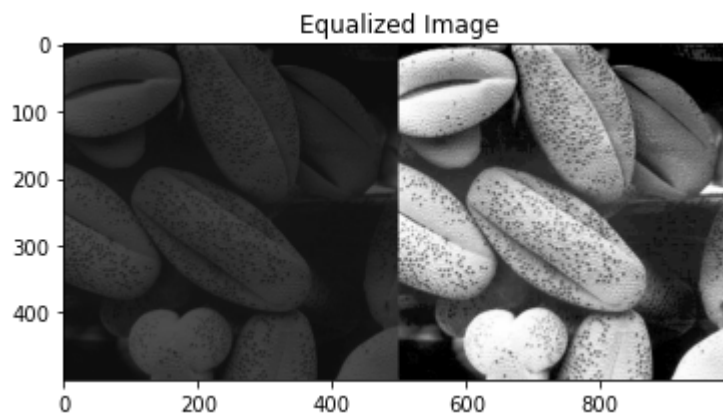

Histogram of the original image

```
import cv2 as cv
img = cv.imread('shells.tif',0)
equ = cv.equalizeHist(img)
histeq = np.hstack((img,equ))
cv.imwrite('histeq.png',histeq)
fig , ax = plt.subplots( )
img1 = cv.imread ( 'histeq.png',cv.IMREAD_COLOR )
ax.imshow(cv.cvtColor(img1, cv.COLOR_BGR2RGB))
ax.set_title('Equalized Image')
plt.show()
```
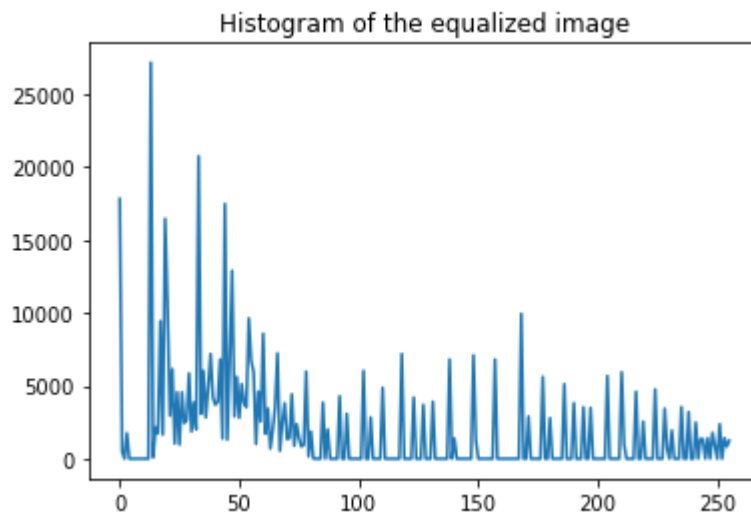

Equalized Image

```
import cv2
from matplotlib import pyplot as plt

img = cv2.imread('histeq.png',0)
histr = cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(histr)
plt.title('Histogram of the equalized image')
plt.show()
```
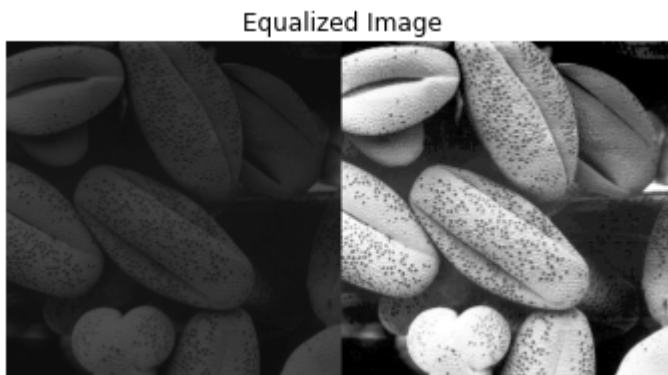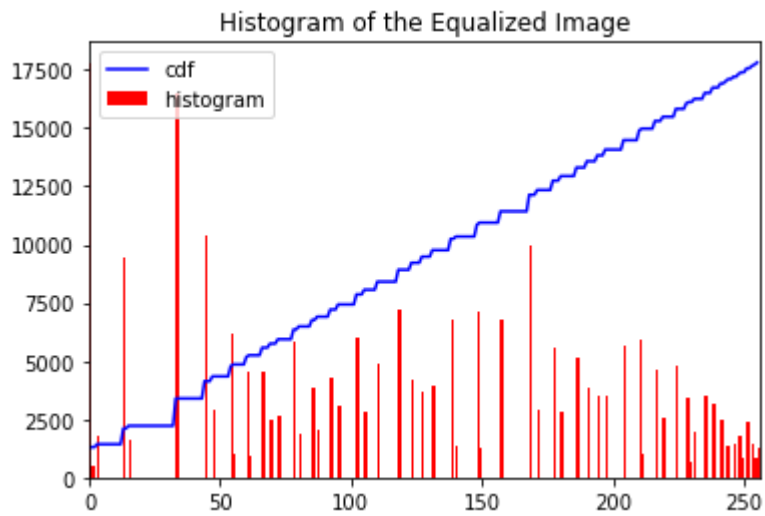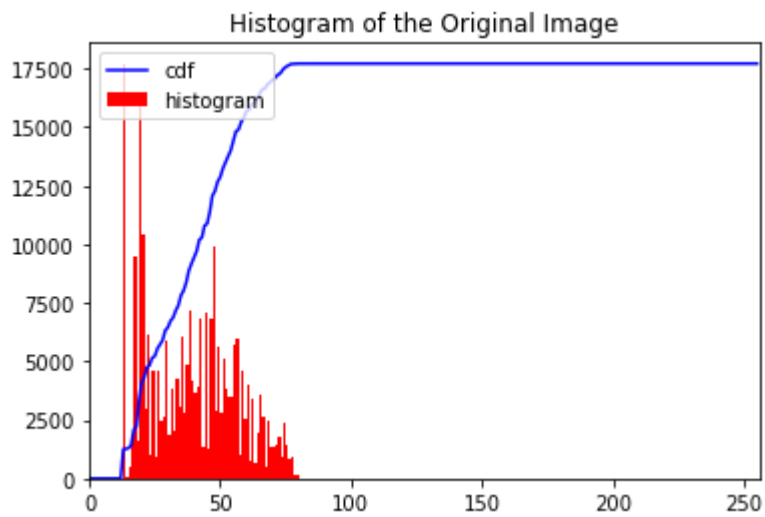
Histogram of the equalized image

```python
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

img = cv.imread("shells.tif",cv.IMREAD_GRAYSCALE)
hist, bins = np.histogram(img.ravel(),256,[0,256])

cdf = hist.cumsum()
cdf_normalized = cdf*hist.max() / cdf.max()
plt.plot(cdf_normalized, color = 'b')
plt.hist(img.flatten(),256,[0,256],color='r')
plt.xlim ([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.title ('Histogram of the Original Image')
plt.show()
equ = cv.equalizeHist(img)
hist, bins = np.histogram (equ.ravel(),256,[0,256])
cdf = hist.cumsum()
cdf_normalized = cdf * hist.max()/cdf.max()
plt.plot (cdf_normalized, color = 'b')
plt.hist(equ.flatten (),256,[0,256], color = 'r')
plt.xlim ([0,256])
plt.legend (('cdf', 'histogram'), loc = 'upper left')
plt.title('Histogram of the Equalized Image')
plt.show()
res = np.hstack ((img, equ))
plt.axis('off')
plt.imshow (res, cmap= 'gray')
plt.title("Equalized Image")
plt.show()
```

Histogram of the Original Image

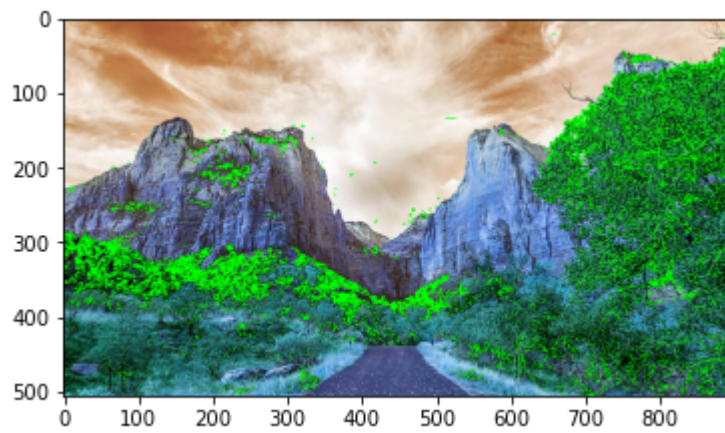Histogram of the Equalized Image

Equalized Image

In [74]:
```python
import cv2

img = cv2.imread("zion_pass.jpg",cv.IMREAD_COLOR)

hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
greenMask = cv2.inRange(hsv, (26, 10, 30), (97, 100, 255))

img[greenMask == 255] = (0, 255, 0)

cv2.imshow('test', img)
plt.imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [76]:
```python
import cv2

img = cv2.imread("zion_pass.jpg",cv.IMREAD_COLOR)

hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
greenMask = cv2.inRange(hsv, (20, 30, 50), (90, 200, 55))

img[greenMask == 255] = (0, 255, 0)

cv2.imshow('test', img)
plt.imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```
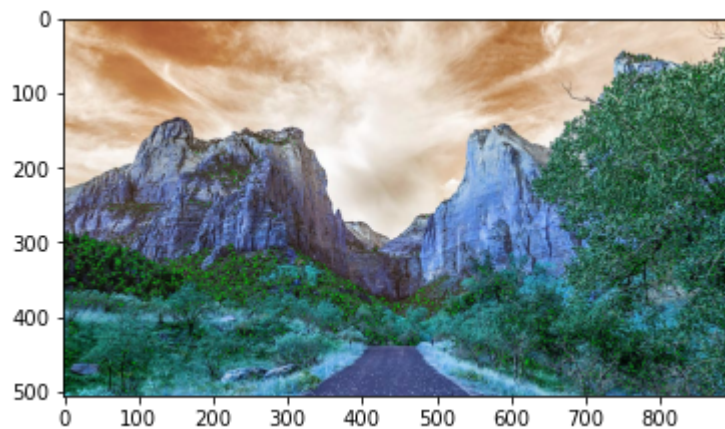

In [ ]: