

```
In [100...
import numpy as np
from skimage import io
from PIL import Image
from matplotlib import pyplot as plt
```

```
In [101...
img_filename = 'C:/Users/91893/Desktop/Anshu/img1.jpg'
save_filename = 'C:/Users/91893/Desktop/Anshu/output.jpg'
```

```
In [102...
#loading file as pillow image
img = Image.open(img_filename)
print(io.imshow(img_filename))
```



```
In [103...
#converting into grayscale
imggray = img.convert(mode='L')

#converting into NumPy array
img_array = np.asarray(imggray)
```

```
In [104...
#STEP 1: Normalized cumulative histogram

#flattening image array and calculating histogram via binning method
pdf = np.bincount(img_array.flatten(), minlength=256)
```

```
In [105...
#normalizing pixel values
tot_sum = np.sum(nk)
pdf = nk/tot_sum

#normalized cumulative histogram
cdf = np.cumsum(pdf)
```

```
In [106...
#STEP 2: Pixel mapping lookup table

transform_map = np.floor(255 * cdf).astype(np.uint8)
```

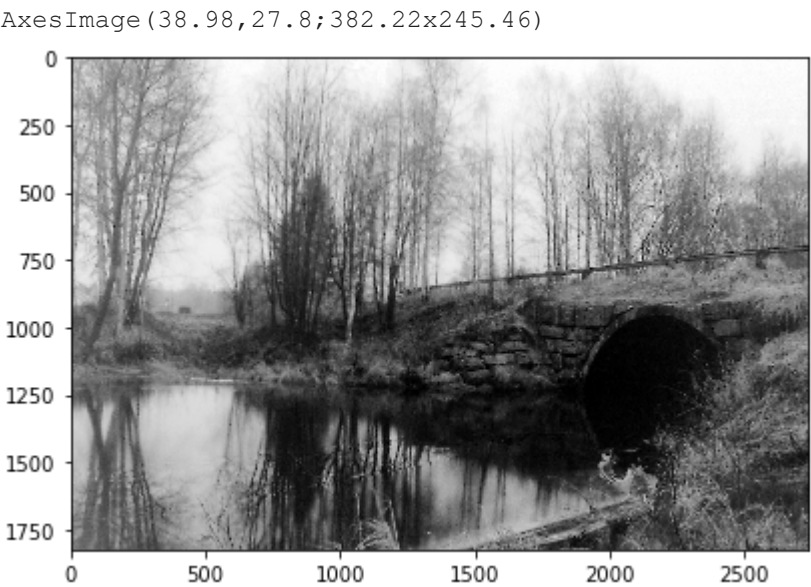
```
In [107...
#STEP 3: Transformation

# flattening image array into 1D list
img_list = list(img_array.flatten())
```

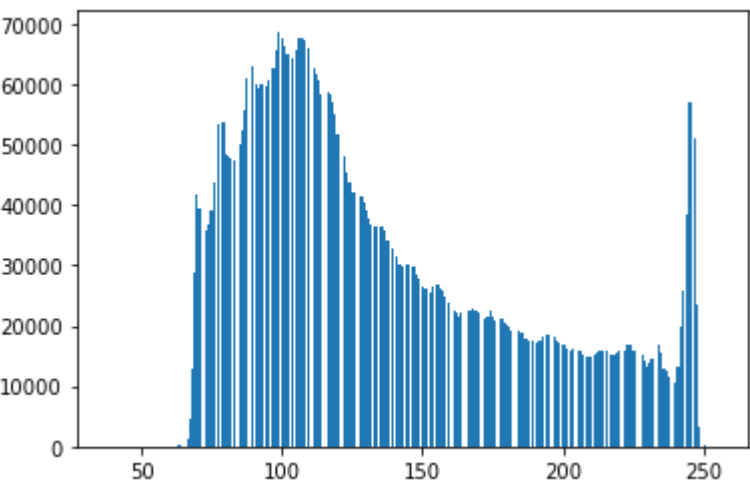
```
In [108...
# mapping the transformed pixel values to original image
eq_img_list = [transform_map[p] for p in img_list]
```

```
In [109...
# reshape and write back into img_array
eq_img_array = np.reshape(np.asarray(eq_img_list), img_array.shape)
```

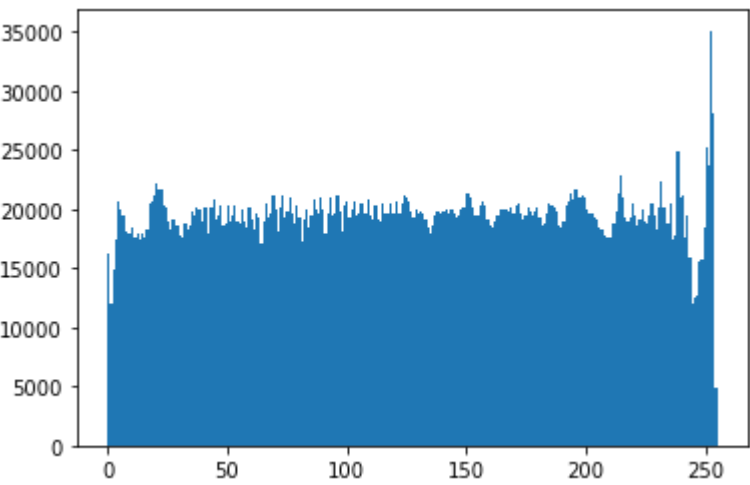
```
In [110...
#convert NumPy array to pillow Image and write to file
eq_img = Image.fromarray(eq_img_array, mode='L')
eq_img.save(save_filename)
print(io.imshow(save_filename))
```



```
In [111...
#histogram of original image
im = io.imread('C:/Users/91893/Desktop/Anshu/img1.jpg')
ax = plt.hist(im.ravel(), bins = 256)
plt.show()
```



```
In [112...
#histogram of equalized image
im = io.imread('C:/Users/91893/Desktop/Anshu/output.jpg')
ax = plt.hist(im.ravel(), bins = 256)
plt.show()
```



In []: