**Experiment 5**

**Aim:** Histogram Equalization

**Code for histogram viewing:**

from scipy import misc

from skimage import io

import matplotlib.pyplot as plt

img = misc.face()

io.imshow(img)

plt.figure()

plt.figure(figsize=(15,5))

plt.subplot(1,3,1)

img1=img[:,:,0].ravel() #converts into a flattened 1D array

ax = plt.hist(img1, bins=256,color='r')

plt.xlabel ("Intensity-->")

plt.ylabel ("Number of pixels-->")

plt.title ("Histogram for Red Plane")

plt.subplot(1,3,2)

img1=img[:,:,1].ravel() #converts into a flattened 1D array

ax = plt.hist(img1, bins=256,color='g')

plt.xlabel ("Intensity-->")

plt.ylabel ("Number of pixels-->")

plt.title ("Histogram for Green Plane")

plt.subplot(1,3,3)

img1=img[:,:,2].ravel() #converts into a flattened 1D array

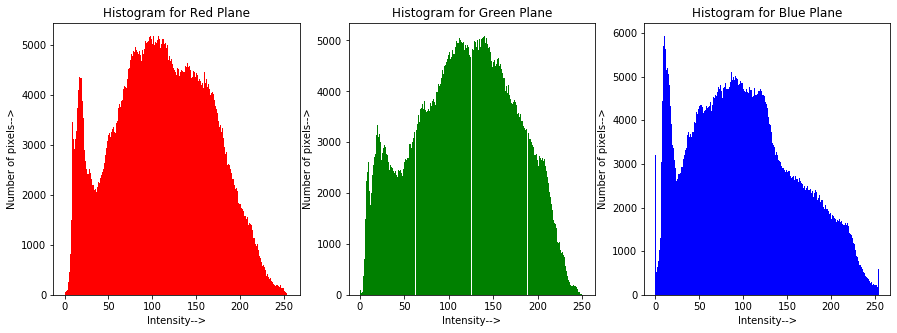
ax = plt.hist(img1, bins=256,color='b')

plt.xlabel ("Intensity-->")

plt.ylabel ("Number of pixels-->")

plt.title ("Histogram for Blue Plane")

**Outputs for Histogram:**



**Code for histogram equalization:**

import numpy as np

image = io.imread ('C:/Users/dhruv/Desktop/College/NOTES/IVP/Labs/pollen\_very\_dark.tif')

# io.imshow (image)

sh=image.shape

row = sh[0]

col = sh[1]

num\_pix = row\*col #number of pixels

imhist,bins = np.histogram(image.ravel(),256) #create histogram

cdf = imhist.cumsum() #cumulative sum in an array form

cdf\_modified = (cdf\*255)/num\_pix

s=cdf\_modified.astype(int)

im\_eq = image.copy()

for r in range (0,row):

for c in range (0,col):

temp= image[r][c]

im\_eq[r][c]= s[temp]

plt.figure()

plt.figure(figsize=(7,7))

plt.subplot(2,2,1)

io.imshow (image)

plt.title ("Original Image")

plt.subplot(2,2,2)

io.imshow (im\_eq)

plt.title ("Equalized Image")

plt.subplot(2,2,3)

ax=plt.hist(image.ravel(),bins=256)

plt.xlabel ("Intensity-->")

plt.ylabel ("Number of pixels-->")

plt.title ("Histogram for Original Image")

plt.subplot(2,2,4)

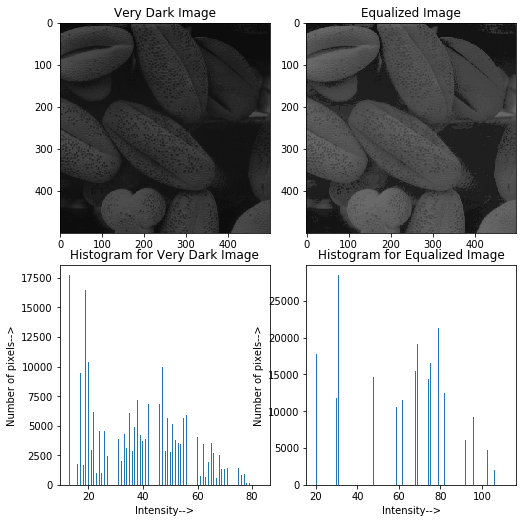
ax=plt.hist(im\_eq.ravel(),bins=256)

plt.xlabel ("Intensity-->")

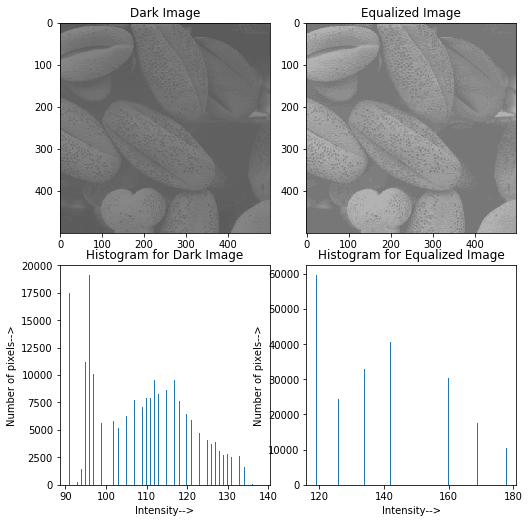
plt.ylabel ("Number of pixels-->")

plt.title ("Histogram for Equalized Image")

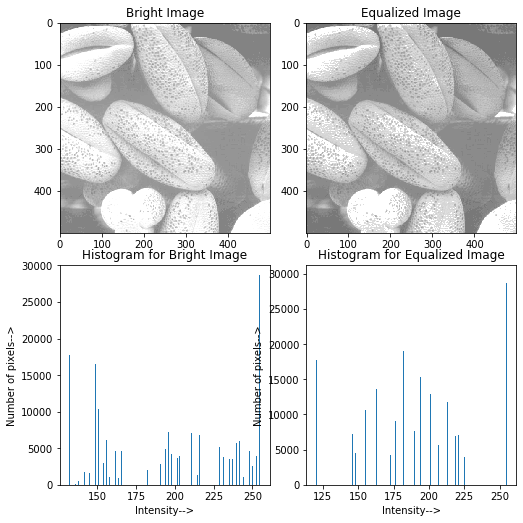
**Output for very dark image:**



**Output for dark image:**



**Output for bright image:**

****

**Conclusion:**

* Histogram of very dark image shows that most of the pixels are in the range 0 to 80.
* Histogram of dark image shows that most of the pixels are in the range 0 to 140.
* Histogram of bright image shows that most of the pixels are in the range 150 to 250.
* After equalizing these images, histogram is redistributed in the range 0 to 255 for all. Therefore, all the equalized images have pixel intensities in the range 0 to 255.

**Experiment 5B**

**Aim:** Histogram Stretching

**Code for histogram stretching:**

#Histogram stretching

#stretches minimum and maximum pixel intensity in original image to the range 0 to 255

image = io.imread ('C:/Users/dhruv/Desktop/College/NOTES/IVP/Labs/pollen\_very\_dark.tif')

sh=image.shape

row = sh[0]

col = sh[1]

s\_min = 0

s\_max = 255

r\_min= np.min(np.min(image))

r\_max = np.max(np.max(image))

im\_stretch = image.copy()

for r in range (0,row):

for c in range (0,col):

temp1= image[r][c]

temp2 = (((s\_max-s\_min)/(r\_max-r\_min))\*(temp1-r\_min))+ s\_min

im\_stretch [r][c] = temp2

plt.figure (figsize=(7,7))

plt.subplot (1,2,1)

io.imshow (image)

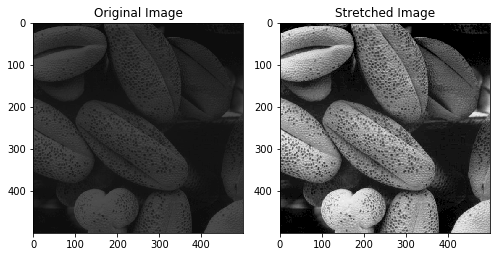
plt.title("Original Image")

plt.subplot (1,2,2)

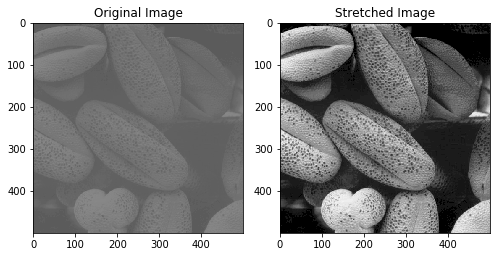
io.imshow (im\_stretch)

plt.title("Stretched Image")

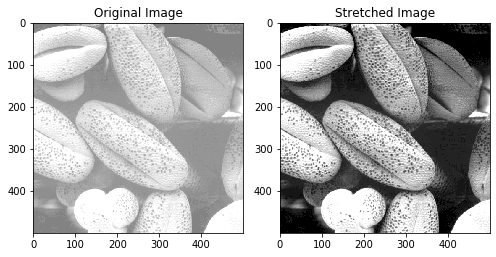
**Output for very dark image:**



**Output for dark image:**



**Output for bright image:**



**Conclusion:**

* Histogram stretching reduces minimum pixel intensity of original image to 0 and increases maximum pixel intensity to 255.
* Intermediate pixel intensities of the original image are increased proportionately.
* In histogram equalization, number of pixels with various intensity values are redistributed whereas in histogram stretching, pixel intensities are transformed to new values to cover the entire range. They are not redistributed.