Name: Jay Goyal

Roll number: C017

Program /Branch: Btech EXTC

Semester: Sem 6

Dateofperformance: 23 /01 / 2021

DateofSubmission: 24 /01 /2020

EXPERIMENT-3

Aim:

a) To write a program in PYTHON to plot histogram of an image

b) To plot histogram of different images and classify them as low contrast, high contrast, dark and bright images.

c) To write a program in PYTHON to perform histogram stretching on an image

d) To write a program in PYTHON to perform histogram equalization

PART A

```
#importing the required libraries
import cv2
import matplotlib.pyplot as plt
import numpy as np
```

```
# defining a function to count the number of times a pixel occurs
#the function returns the pixel values and their count as lists
def histogram_image(img):
  [m,n]=img.shape
  count=[]
  pixels=[]
  for k in range(0,256):
    pixels.append(k)
    countt=0
    for i in range(m):
      for j in range(n):
        if img[i,j]==k:
          countt+=1
    count.append(countt)
  return (pixels,count)
```

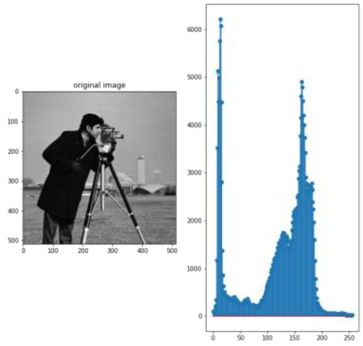
```
#reading the image
img1=cv2.imread('/content/cameraman.tif',0)
type(img1)
```

```
     numpy.ndarray
```

```
#calling the function defined above
pixel1,count1=histogram_image(img1)
```

```
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.title('original image')
plt.imshow(img1,cmap='gray')
plt.subplot(1,2,2)
plt.stem(pixel1,count1)
```

```
     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:6: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set t

     <StemContainer object of 3 artists>
```



PART B

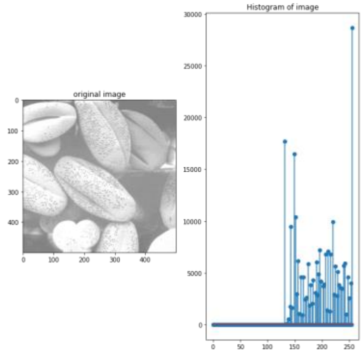
```
#reading the image
img2=cv2.imread('/content/Fig0320(1)(top_left).tif',0)
type(img2)
```

```
     numpy.ndarray
```

```
pixel2,count2 = histogram_image(img2)
```

```
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.title('original image')
plt.imshow(img2,cmap='gray',vmin=0,vmax=255)
plt.subplot(1,2,2)
plt.title('Histogram of image')
plt.stem(pixel2,count2)
```

```
     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set t
       import sys
     <StemContainer object of 3 artists>
```

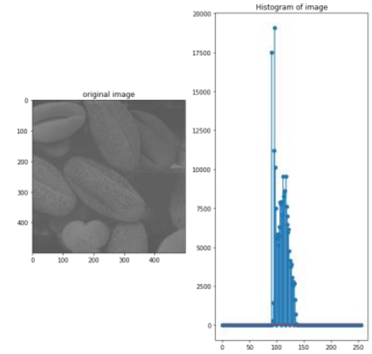


Inference : Here since the accumulation of frequencies of the pixels responsible for brightness in an image are higher in the histogram, this is a bright image.

```
#reading the image
img3=cv2.imread('/content/Fig0320(2)(2nd_from_top).tif',0)
type(img3)

     numpy.ndarray
```

```
pixel3,count3 =histogram_image(img3)
```

```
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.title('original image')
plt.imshow(img3,cmap='gray',vmin=0,vmax=255)
plt.subplot(1,2,2)
plt.title('Histogram of image')
plt.stem(pixel3,count3)

     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performanc e of a stem plot. To remove this warning and switch to the new behaviour, set t
       import sys
     <StemContainer object of 3 artists>
```

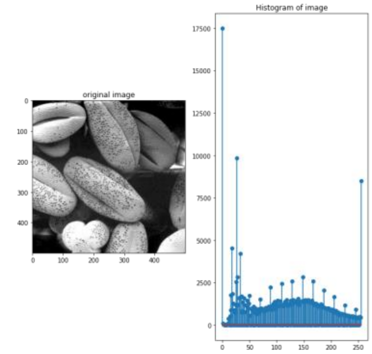


Inference : Here since the frequencies of the pixels are accumulated in the center of the histogram, this is a low contrast image.

```
#reading the image
img4=cv2.imread('/content/Fig0320(3)(third_from_top).tif',0)
type(img4)

     numpy.ndarray
```

```
pixel4,count4 =histogram_image(img4)
```

```
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.title('original image')
plt.imshow(img4,cmap='gray',vmin=0,vmax=255)
plt.subplot(1,2,2)
plt.title('Histogram of image')
plt.stem(pixel4,count4)

     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineColle ction instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set t
       import sys
     <StemContainer object of 3 artists>
```

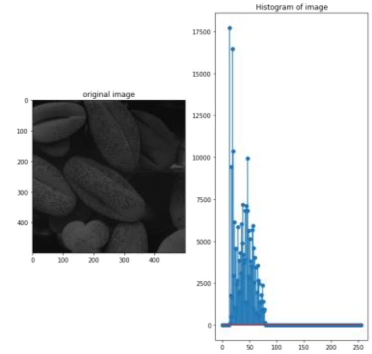


Inference : Here since the frequencies of the pixels are spread out through a range in the histogram, this is a high contrast image.

```
#reading the image
img5=cv2.imread('/content/Fig0320(4)(bottom_left).tif',0)
type(img5)

     numpy.ndarray
```

```
 pixel5,count5 =histogram_image(img5)
```

```
plt.figure(figsize=(10,10))
plt.subplot(1,2,1)
plt.title('original image')
plt.imshow(img5,cmap='gray',vmin=0,vmax=255)
plt.subplot(1,2,2)
plt.title('Histogram of image')
plt.stem(pixel5,count5)

     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineColle ction instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set t
       import sys
     <StemContainer object of 3 artists>
```



Inference : Here since accumulation of the frequencies of the pixels responsible for darkness in an image are higher in the histogram, this is a dark image.

PART C- Histogram Streching

Formula for Histogram Streching:

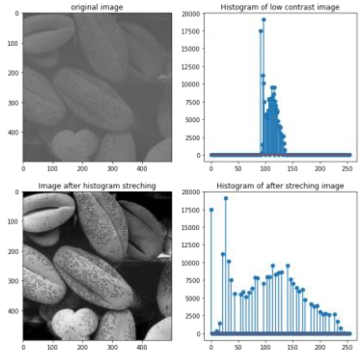
S = T(r) = ((Smax - Smin)/(rmax - rmin))(r - rmin) + Smin

```
rmax=np.max(img3) # max pixel value in image
rmin=np.min(img3) # min pixel value in image
smax=255 # max pixel value in new image
smin=0  #  min  pixel  value  in  image

ratio=(smax-smin)/(rmax-rmin)

m1,n1=img3.shape


#Creating an array of zeros that will be the new image
img6=np.zeros((m1,n1),dtype=np.int)

for i in range(m1):
  for j in range(n1):
     img6[i,j]=ratio*(img3[i,j]-rmin)+smin
cv2.imwrite('HISTOGRAMSTRECHEDIMAGE.png',img6)
```

```
     True
```

```
plt.figure(figsize=(10,10))
plt.subplot(2,2,1)
plt.title('original image')
plt.imshow(img3,cmap='gray',vmin=0,vmax=255)
plt.subplot(2,2,2)
plt.title('Histogram of low contrast image')
plt.stem(pixel3,count3)
plt.subplot(2,2,3)
plt.title('Image  after  histogram  streching')
plt.imshow(img6,cmap='gray',vmin=0,vmax=255)
plt.subplot(2,2,4)
plt.title('Histogram of after streching image')
pixel6,count6 =histogram_image(img6)
plt.stem(pixel6,count6)
```

```
     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set t
       import sys
     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:14: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set

     <StemContainer object of 3 artists>
```



Inference: Here we have performed histogram streching on a low contrast image. We infer that the process of histogram streching has not altered the shape of the original image but has only increased the dynamic range of the image thus we obtain a histogram of an image that has pixels spread throughout that dynamic range. This image so obtained is a high contrast image and thus provides more information.

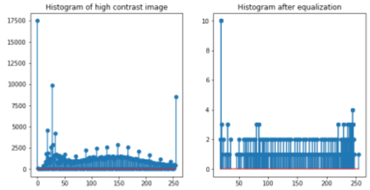
PART D - Histogram Equalization

```
# nk is the sum of all elements in list count4
nk=0
c=len(count4)
for i in range(0,c):
    nk = nk + count4[i]
nk
```

```
     250000
```

```
#Calculating the pdf of the taken list
pdf=[]
for i in range(0,c):
  pdf.append(count4[i]/nk)


#Calculating the cdf of the given list
cdf=[pdf[0]]
for i in range(1,c):
  update=cdf[i-1]+pdf[i]
  cdf.append(update)


#Multiplying the cdf by(L-1)
mul=[]
for i in range(0,c):
  mul.append(cdf[i]*(255))


#Rounding off the values in the mul list to whole values
import math
rounded=[]
for i in range(0,c):
  rounded.append(math.ceil(mul[i]))


#Counting the number of occurance of rouunded pixels
count0=[]
for k in rounded:
    countt1=0
    for i in range(c):
        if rounded[i]==k:
          countt1+=1
    count0.append(countt1)



plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
plt.title('Histogram of high contrast image')
plt.stem(pixel4,count4)
plt.subplot(1,2,2)
plt.title('Histogram after equalization')
plt.stem(rounded,count0)
```

```
     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:4: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set t
       after removing the cwd from sys.path.
     /usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:7: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This significantly improves the performance of a stem plot. To remove this warning and switch to the new behaviour, set t
       import sys
     <StemContainer object of 3 artists>
```



Inference: Here we can see that the overall shape of the histogram has changed and has become equalized while in histogram steching the shape of the histogram remains the same only its dynamic range changes.