Name: Jay Goyal

Roll no.: C017

Semester: VI

Program: B.Tech

Branch: EXTC

Date of performance: 22nd January

Date of Submission: 29th January

Experiment Number: 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

Conclusion: Outcome:

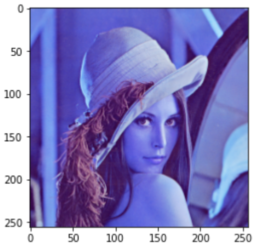
from the experiment we learnt how to plot histogram of an image,plot histogram of different images and classify them as low contrast, high contrast, dark and bright images,perform histogram stretching and histogram equalization.

Collab Link: https://colab.research.google.com/drive/1hHqL7WXNgrIfeM-yJBjb3cq4zD7tU3d8?usp=sharing

```
#Importing all the required Libraries
import cv2
import matplotlib.pyplot as plt
import numpy as np
from skimage import io
```

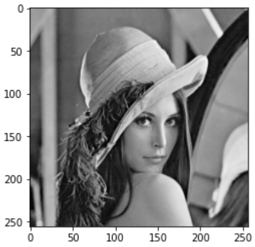
```
#Read the image
import cv2
img1 = cv2.imread("/content/lena_color_256.tif",1)
cv2.imwrite("lena_color_256.tif",img1)
plt.imshow(img1)
```

```
<matplotlib.image.AxesImage at 0x7fead5ff0470>
```



```
#Turn the image into grey
im1=cv2.cvtColor(img1,cv2.COLOR_BGR2GRAY)
plt.imshow(im1,cmap='gray')
```

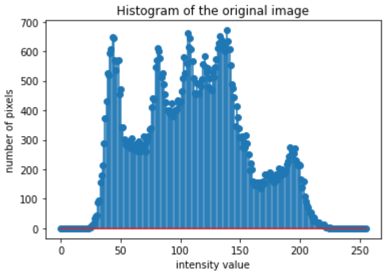
```
<matplotlib.image.AxesImage at 0x7fead42c53c8>
```



```
# import cv2, numpy, matplotlib
import cv2
import numpy as np
import matplotlib.pyplot as plt


# function to obtain histogram of an image
def hist_plot(img):

    # empty list to store the count
    # of each intensity value
    count =[]

    # empty list to store intensity
    # value
    r = []

    # loop to traverse each intensity
    # value
    for k in range(0, 256):
        r.append(k)
        count1 = 0

        # loops to traverse each pixel in
        # the image
        for i in range(m):
            for j in range(n):
                if img[i,j]== k:
                    count1+= 1
        count.append(count1)

    return (r, count)


img = cv2.imread('/content/lena_color_256.tif', 0)

# To ascertain total numbers of rows and
# columns of the image, size of the image
m, n = img.shape
r1, count1 = hist_plot(img)
```

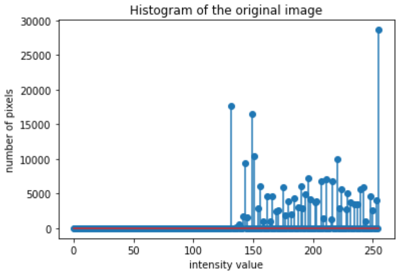
```
# plotting the histogram
plt.stem(r1, count1)
plt.xlabel('intensity value')
plt.ylabel('number of pixels')
plt.title('Histogram of the original image')
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:43: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This s
Text(0.5, 1.0, 'Histogram of the original image')
```



```
# import cv2, numpy, matplotlib
import cv2
import numpy as np
import matplotlib.pyplot as plt


# function to obtain histogram of an image
def hist_plot(img):

    # empty list to store the count
    # of each intensity value
    count =[]

    # empty list to store intensity
    # value
    r = []

    # loop to traverse each intensity
    # value
    for k in range(0, 256):
        r.append(k)
        count1 = 0

        # loops to traverse each pixel in
        # the image
        for i in range(m):
            for j in range(n):
                if img[i,j]== k:
                    count1+= 1
        count.append(count1)

    return (r, count)


img = cv2.imread('/content/Fig0320(1)(top_left).tif', 0)

# To ascertain total numbers of rows and
# columns of the image, size of the image
m, n = img.shape
r1, count1 = hist_plot(img)

# plotting the histogram
plt.stem(r1, count1)
plt.xlabel('intensity value')
plt.ylabel('number of pixels')
plt.title('Histogram of the original image')
```

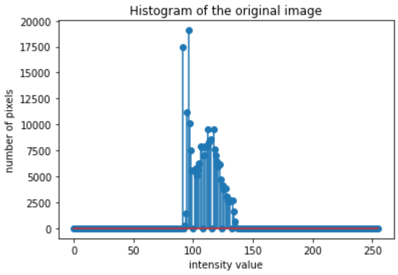
```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:43: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This s
Text(0.5, 1.0, 'Histogram of the original image')
```



Thus we obtained a Histogram of bright intensity

```
# import cv2, numpy, matplotlib
import cv2
import numpy as np
import matplotlib.pyplot as plt


# function to obtain histogram of an image
def hist_plot(img):

    # empty list to store the count
    # of each intensity value
    count =[]

    # empty list to store intensity
    # value
    r = []

    # loop to traverse each intensity
    # value
    for k in range(0, 256):
        r.append(k)
        count1 = 0

        # loops to traverse each pixel in
        # the image
        for i in range(m):
            for j in range(n):
                if img[i,j]== k:
                    count1+= 1
        count.append(count1)

    return (r, count)


img = cv2.imread('/content/Fig0320(2)(2nd_from_top).tif', 0)

# To ascertain total numbers of rows and
# columns of the image, size of the image
m, n = img.shape
r1, count1 = hist_plot(img)

# plotting the histogram
plt.stem(r1, count1)
plt.xlabel('intensity value')
plt.ylabel('number of pixels')
plt.title('Histogram of the original image')
```

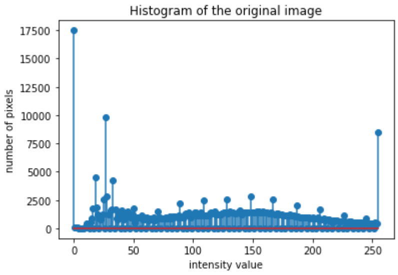
```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:43: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This s
Text(0.5, 1.0, 'Histogram of the original image')
```



Thus we obtained low contrast image

```
# import cv2, numpy, matplotlib
import cv2
import numpy as np
import matplotlib.pyplot as plt


# function to obtain histogram of an image
def hist_plot(img):

    # empty list to store the count
    # of each intensity value
    count =[]

    # empty list to store intensity
    # value
    r = []

    # loop to traverse each intensity
    # value
    for k in range(0, 256):
        r.append(k)
        count1 = 0

        # loops to traverse each pixel in
        # the image
        for i in range(m):
            for j in range(n):
                if img[i,j]== k:
                    count1+= 1
        count.append(count1)

    return (r, count)


img = cv2.imread('/content/Fig0320(3)(third_from_top).tif', 0)

# To ascertain total numbers of rows and
# columns of the image, size of the image
m, n = img.shape
r1, count1 = hist_plot(img)

# plotting the histogram
plt.stem(r1, count1)
plt.xlabel('intensity value')
plt.ylabel('number of pixels')
plt.title('Histogram of the original image')
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:43: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This s
Text(0.5, 1.0, 'Histogram of the original image')
```

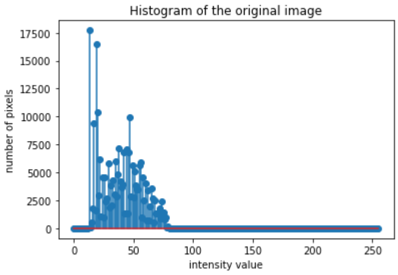


Thus we obtain high contrast image

```
# import cv2, numpy, matplotlib
```

```
import cv2
import numpy as np
import matplotlib.pyplot as plt


# function to obtain histogram of an image
def hist_plot(img):

    # empty list to store the count
    # of each intensity value
    count =[]

    # empty list to store intensity
    # value
    r = []

    # loop to traverse each intensity
    # value
    for k in range(0, 256):
        r.append(k)
        count1 = 0

        # loops to traverse each pixel in
        # the image
        for i in range(m):
            for j in range(n):
                if img[i,j]== k:
                    count1+= 1
        count.append(count1)

    return (r, count)


img = cv2.imread('/content/Fig0320(4)(bottom_left).tif', 0)

# To ascertain total numbers of rows and
# columns of the image, size of the image
m, n = img.shape
r1, count1 = hist_plot(img)

# plotting the histogram
plt.stem(r1, count1)
plt.xlabel('intensity value')
plt.ylabel('number of pixels')
plt.title('Histogram of the original image')
```

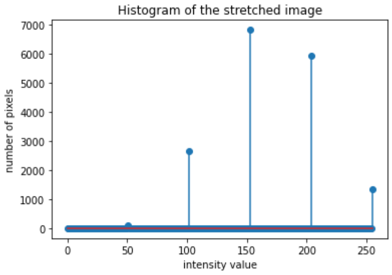
```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:43: UserWarning: In Matplotlib 3.3 individual lines on a stem plot will be added as a LineCollection instead of individual lines. This s
Text(0.5, 1.0, 'Histogram of the original image')
```



Thus we obtained dark contrast image

```
# Transformation to obtain stretching
constant = (255-0)/(img.max()-img.min())
img_stretch = img * constant
r, count = hist_plot(img_stretch)

# plotting the histogram
plt.stem(r, count)
plt.xlabel('intensity value')
plt.ylabel('number of pixels')
plt.title('Histogram of the stretched image')

# Storing stretched Image
cv2.imwrite('Stretched Image 4.png', img_stretch)
```

```
/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 si
  import sys
True
```



Thus we obtained a stretched image.