Name : Sarvagya Singh **SAPID : 60009200030**

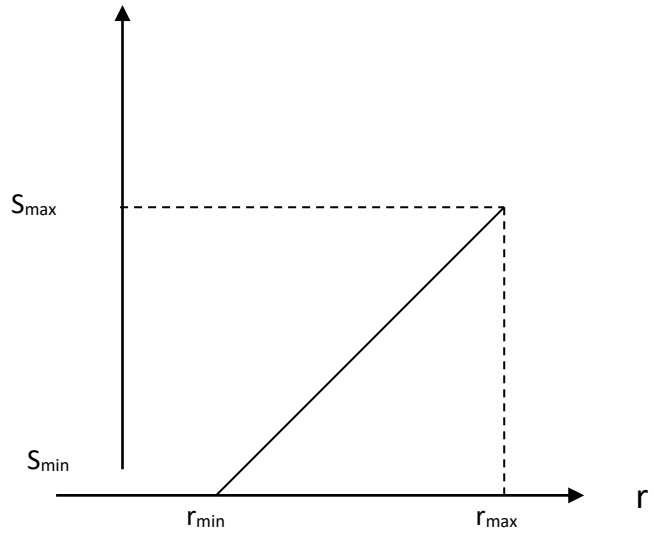
AIM: To Perform Histogram Stretching

THEORY:

Histogram Stretching

It is a method to increase the dynamic range of the image. Here we do not alter the basic shape of the histogram, but we spread it so as to cover the entire dynamic range. We do this by using a straight line equation having a slope

$(s_{max} - s_{min})/ (r_{max} - r_{min})$



s_{max} = Maximum grey level of output image

s_{min} = Minimum grey level of output image.

r_{max} = Maximum grey level of input image

r_{min} = Minimum grey level of input image.

$$S = T(r) = \frac{(smax - smin)}{(rmax - rmin)} (r - r_{min}) + smin$$

This transformation stretches and shifts the grey level range of input image to occupy the entire dynamic range (s_{max}, s_{min}).

RESULT:

So we have performed the image contrast stretching and thereby doing it the low contrast images start to have a larger range of image pixel values. This increases the range of intensity of the pixel values.The Histogram stretching increases the pixel range of the images but the distribution of the image remains the same thereby making sure not to create any anomaly.

Sarvayga Singh

60009200030 - K1

IPCV - lab6

Contrast Stretching

In [1]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow
```

In [2]:
```
img = cv2.imread('/content/imager.jpeg',0)
img
```

Out[2]:
```
array([[206, 206, 206, ..., 207, 207, 207],
       [206, 206, 206, ..., 207, 207, 207],
       [206, 206, 206, ..., 207, 207, 207],
       ...,
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255]], dtype=uint8)
```

In [3]:
```
np.shape(img)
```

Out[3]:
```
(280, 600)
```

In [4]:
```
final_img = [[0 for j in range(np.shape(img)[1])] for i in range(np.shape(img)[0])]
np.shape(final_img)
```

Out[4]:
```
(280, 600)
```

In [5]:
```
image = cv2_imshow(img)
```

In [6]:
```
rmax,rmin = np.max(img),np.min(img)
rX = (rmax , rmin)
rX
```

Out[6]:
```
(255, 31)
```

In [7]:
```
smax,smin = 255,0
sX = (smax , smin)
sX
```

Out[7]:
```
(255, 0)
```

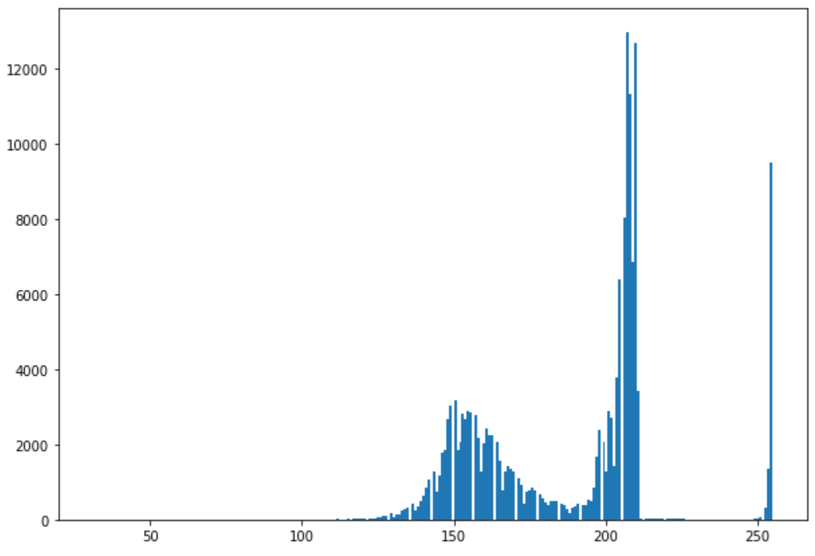
In [8]:
```
ratio = (sX[0]-sX[1])/(rX[0]-rX[1])
ratio
```

Out[8]:
```
1.1383928571428572
```

In [9]:
```
imgX = np.reshape(img,-1)
imgX
```

Out[9]:
```
array([206, 206, 206, ..., 255, 255, 255], dtype=uint8)
```

In [10]:
```
plt.figure(figsize=(10,7))
plt.hist(imgX, bins=256);
```



In [11]:
```
for i,lister in enumerate(img):
  for j,value in enumerate(lister):
    final_img[i][j] = ratio*(value-rmin) + smin

final_img = np.array(final_img)
```

In [12]:
```
np.shape(final_img)
```

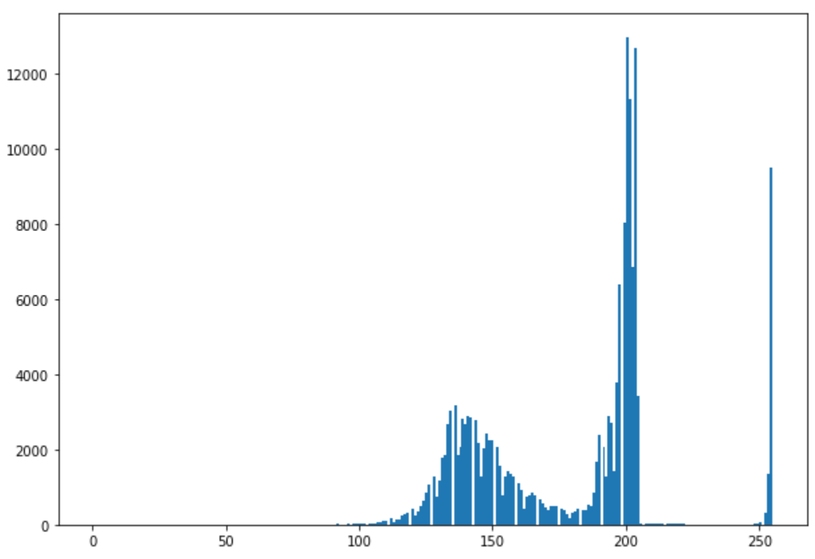
Out[12]:
```
(280, 600)
```

In [13]:
```
final_imgX = np.reshape(final_img,-1)
final_imgX
```

Out[13]:
```
array([199.21875, 199.21875, 199.21875, ..., 255.     , 255.     ,
       255.     ])
```

In [14]:
```
cv2_imshow(final_img)
```

In [16]:
```
plt.figure(figsize=(10,7))
plt.hist(final_imgX, bins=256);
```



In [17]:
```
np.max(final_imgX),np.min(final_imgX)
```

Out[17]:
```
(255.0, 0.0)
```