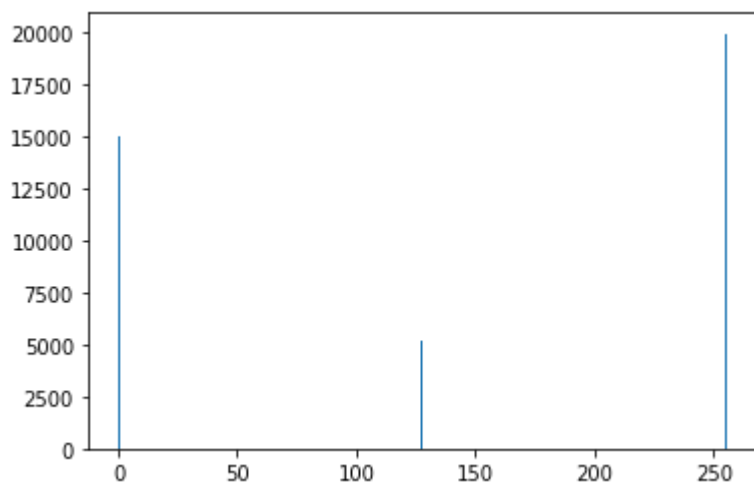
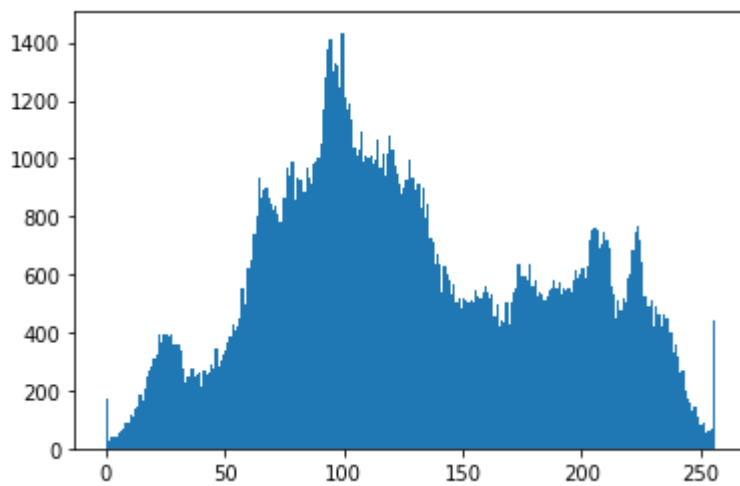


Experiment 2: Histogram Equalization

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
im=np.zeros((200,200),np.uint8)
cv2.rectangle(im,(0,100),(200,200),(255),-1)
cv2.rectangle(im,(0,50),(100,100),(127),-1)
cv2_imshow(im)
plt.hist(im.ravel(),256,[0,256])
plt.show()
```

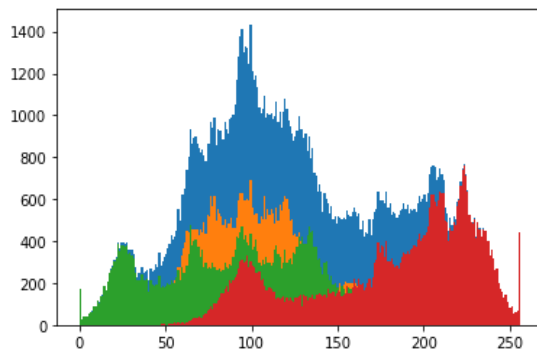


```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
im=cv2.imread('/content/Lena.jfif')
cv2_imshow(im)
plt.hist(im.ravel(),256,[0,256])
plt.show()
```



```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
im=cv2.imread('/content/Lena.jfif')
b,g,r=cv2.split(im)
cv2_imshow(im)
cv2_imshow(b)
cv2_imshow(g)
cv2_imshow(r)
plt.hist(im.ravel(),256,[0,256])
plt.hist(b.ravel(),256,[0,256])
plt.hist(g.ravel(),256,[0,256])
plt.hist(r.ravel(),256,[0,256])
plt.show()
```

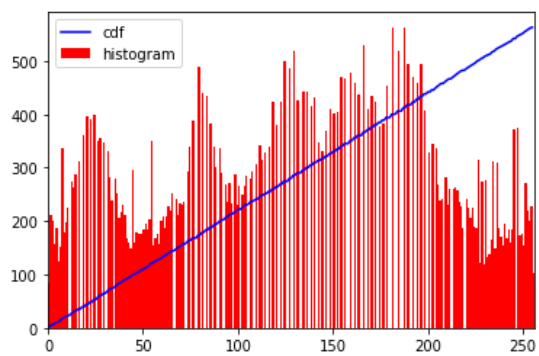
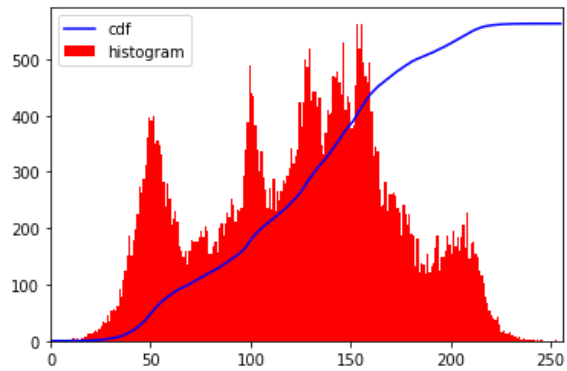




```
import cv2
import numpy as np
from matplotlib import pyplot as plt
from google.colab.patches import cv2_imshow

#path = "/content/drive/MyDrive/content/lena.jpg"
img = cv2.imread('/content/Lena.jfif',0)
cv2_imshow(img)
equ=np.zeros((img.shape[1],img.shape[0]),np.uint8)
equ = cv2.equalizeHist(img)
cv2_imshow(equ)
hist,bins = np.histogram(img.flatten(),256,[0,256])
cdf = hist.cumsum()
cdf_normalized = cdf * float(hist.max()) / cdf.max()
plt.plot(cdf_normalized, color = 'b')
plt.hist(img.flatten(),256,[0,256], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.show()
hist_1,bins_1 = np.histogram(equ.flatten(),256,[0,256])
cdf_1 = hist_1.cumsum()
#print (cdf)
cdf_normalized_1 = cdf_1 * float(hist_1.max()) / cdf_1.max()
#print(cdf_normalized)
plt.plot(cdf_normalized_1, color = 'b')
plt.hist(equ.flatten(),256,[0,256], color = 'r')
plt.xlim([0,256])
plt.legend(('cdf','histogram'), loc = 'upper left')
plt.show()
```





```
# import Opencv
import cv2

# import Numpy
import numpy as np

# read a image using imread
img = cv2.imread('/content/Lena.jfif', 0)

# creating a Histograms Equalization
# of a image using cv2.equalizeHist()
equ = cv2.equalizeHist(img)

# stacking images side-by-side
res = np.hstack((img, equ))

# show image input vs output
cv2.imshow(res)
```

