

# Computer Vision Homework 3

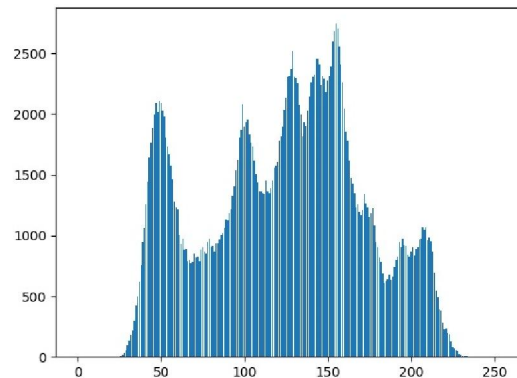
R08922079 資工所一 洪浩翔

## Part 1

(a) Original image and its histogram



Original Lena image



Corresponding histogram

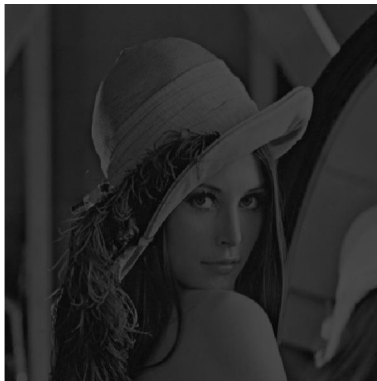
```
def histogram(image):  
    histo = np.zeros(256)  
    for i in range(256):  
        histo[i] = np.where(image==i)[0].shape[0]  
    histo = list(histo)  
    return histo  
  
def saveBar(barList, name):  
    plt.bar(range(0, 256), barList)  
    plt.savefig(name)  
    plt.clf()
```

Code for histogram calculating and plotting

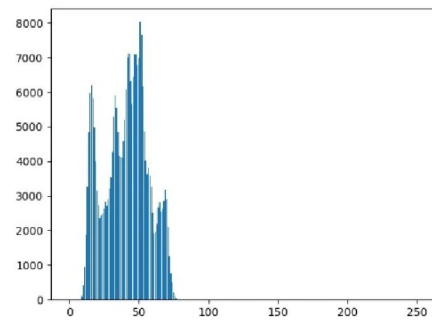
```
#a. original image and its histogram  
histo = histogram(image)  
saveBar(histo, "histogramOriginal.jpg")
```

Code for part (a)

(b) Image with intensity divided by 3 and its histogram



Lena divided by 3



Corresponding histogram

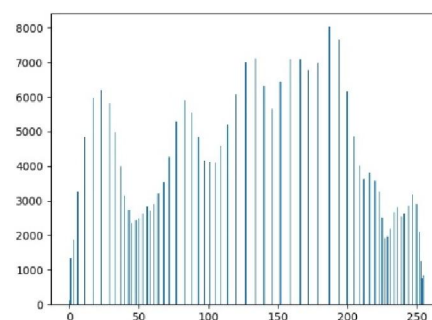
```
#b. image with intensity divided by 3 and its histogram
image = image // 3
histo = histogram(image)
saveBar(histo, "histogramReduce.jpg")
cv2.imwrite('reduceLena.jpg', image)
```

Code for part (b)

(c) Image after applying histogram equalization to (b) and its histogram



Histogram equalized Lena



Corresponding histogram

```
#c. image after applying histogram equalization to (b) and its histogram
counting = np.array(histo)
index = 0
w,h = image.shape
cdfMin = counting[int(min(image.flatten()))]
output = np.zeros(image.shape)
for i,j in enumerate(counting):
    index += j
    value = int(round(((index-cdfMin)/(w*h-cdfMin))*255))
    PixelIndex = np.where(image == i)
    output[PixelIndex] = value
cv2.imwrite('HistoEqualLena.jpg', output)
histo = histogram(output)
saveBar(histo, "histogramHistoEqual.jpg")
```

Code for part (c)