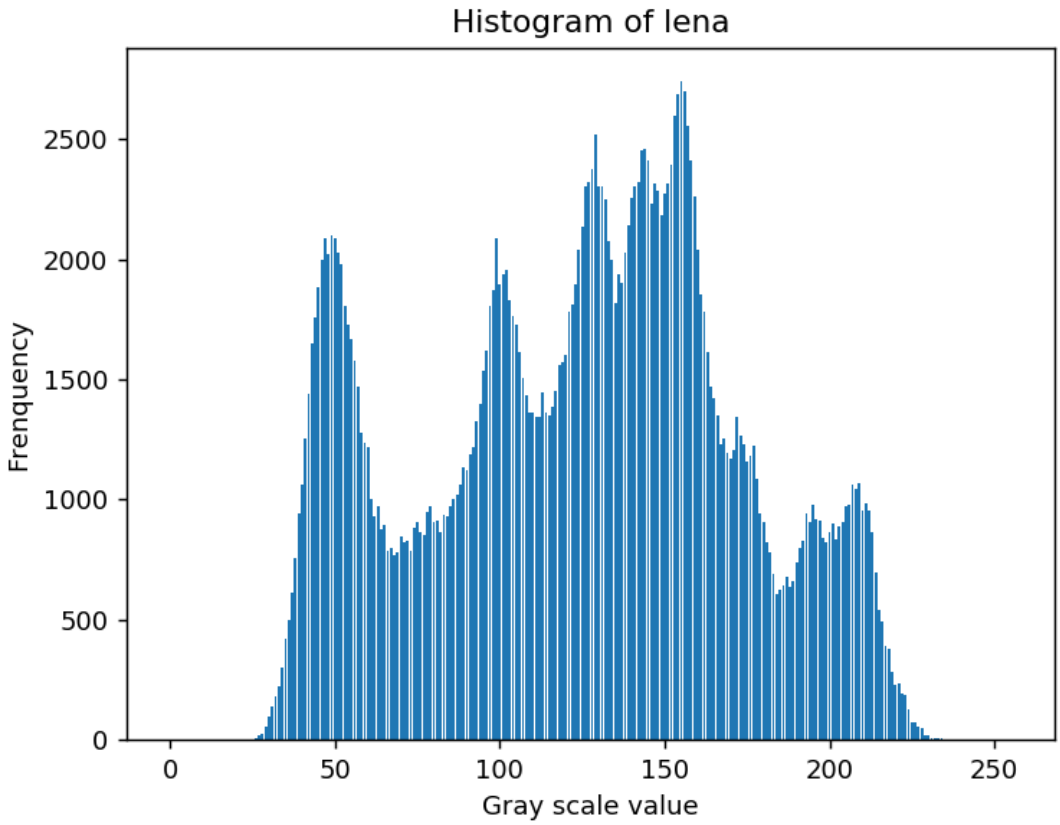
Computer Vision Homework #3

資工四 b05902115 陳建丞

1. Original image and histogram

- **Result**



- **Implementation**

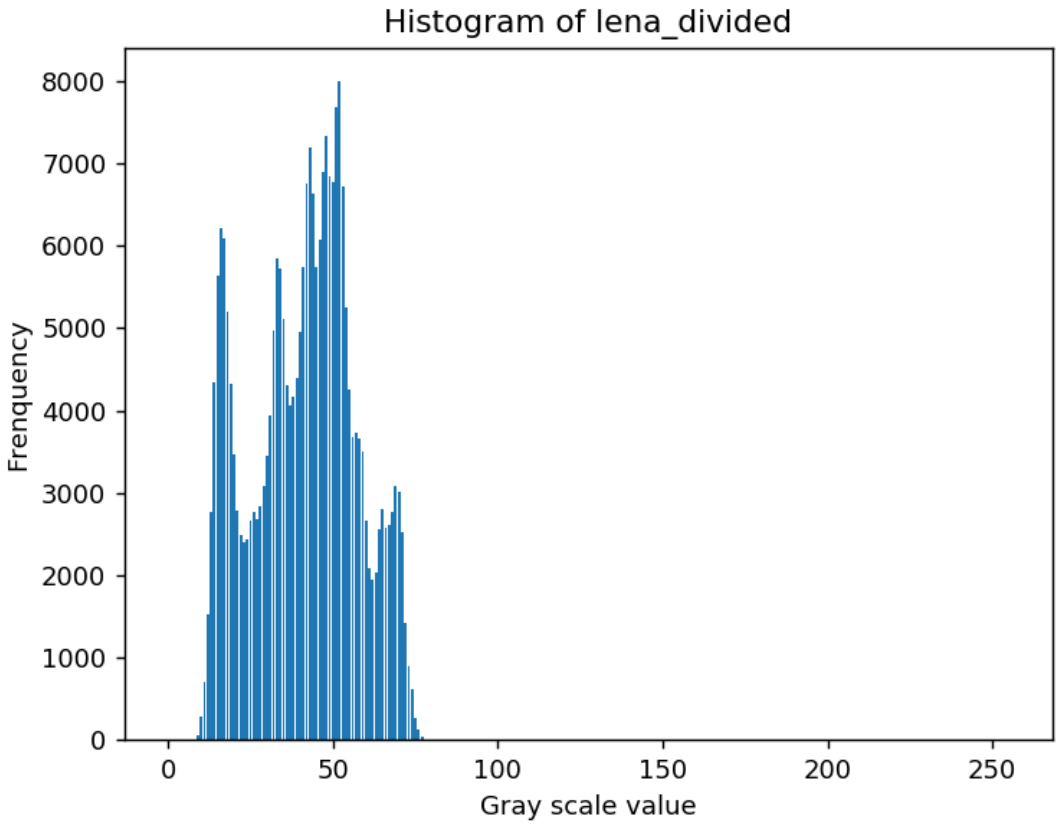
  Use the function(with `matplotlib`) from HW2 to draw the histogram of lena.bmp.

```
def Plot_histogram(img, name):
  pixels = np.zeros((256), dtype = int)

  for row in img:
    for i in row:
      pixels[i] += 1

  plt.bar(range(len(pixels)), pixels)
  plt.title('Histogram of ' + name)
  plt.xlabel('Gray scale value')
  plt.ylabel('Frenquency')
  plt.savefig(name + '_histogram.png', dpi=130)
  plt.clf()
  return pixels
```

2.

- **Result**

Apparently, the result image becomes very dark after the intensity divided by 3, which generates a histogram with most bars centralizing at the left side.



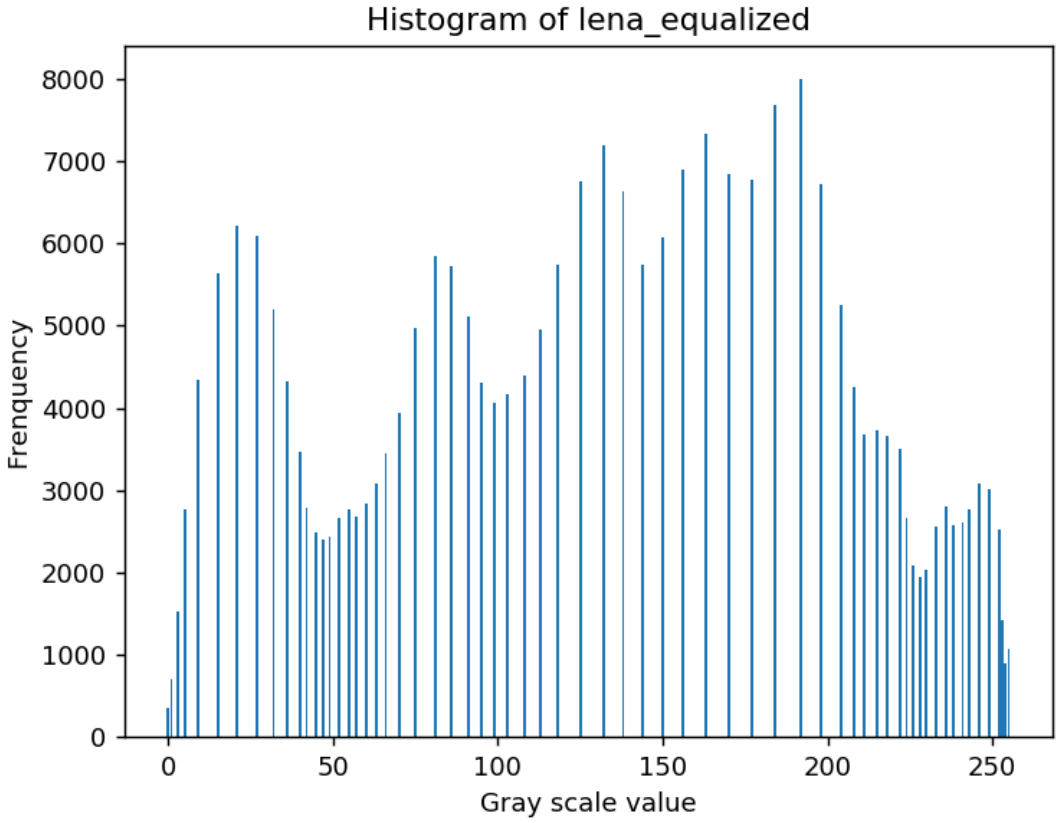
- **Implementation**

```
def Intensity_division(img, intensity):

  for i in range(len(img)):
    for j in range(len(img)):
      img[i][j] = round(img[i][j]/3)

  return img
```

3.

- **Result**

  After performing the histogram equalization, the contrast of the image becomes stronger. Same result can be observed clearly from the histogram, the bars are far more distributed than the original image.


Histogram of lena_equalized
Frenquency
Gray scale value
0
1000
2000
3000
4000
5000
6000
7000
8000
0
50
100
150
200
250

- **Implementation**

```
def Histogram_equalization(img, his):

  # Linearization
  cdf = np.zeros((256), dtype = int)
  sum = 0

  for i in range(len(his)):
    sum += his[i]
    cdf[i] = sum

  # Transformation
  img_equalized = img.copy()
  for i in range(len(img)):
    for j in range(len(img)):
      img_equalized[i][j] = round(255 * cdf[img[i][j]]/sum)

  return img_equalized
```

Use the array of every gray scale value's frequency from problem2 to calculate accumulated pixels values array (cdf). Then apply formula to compute the new pixel value of each pixel.

$$s_k = 255 \sum_{j=0}^{k} \frac{n_j}{n}$$