Computer Vision HW3

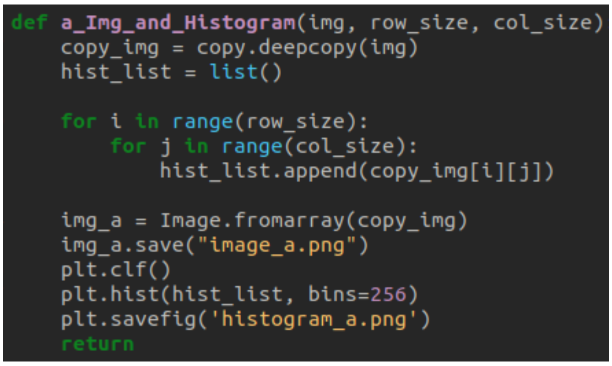
R12922054 資工所 邱信瑋

Description

1. Original image and its histogram

How to implement :

這題好像跟作業二出的一模一樣...總之就是把所有的點遍歷一遍然後用list存起來，最後將其作圖



```
def a_Img_and_Histogram(img, row_size, col_size)
    copy_img = copy.deepcopy(img)
    hist_list = list()

    for i in range(row_size):
        for j in range(col_size):
            hist_list.append(copy_img[i][j])

    img_a = Image.fromarray(copy_img)
    img_a.save("image_a.png")
    plt.clf()
    plt.hist(hist_list, bins=256)
    plt.savefig('histogram_a.png')
    return
```

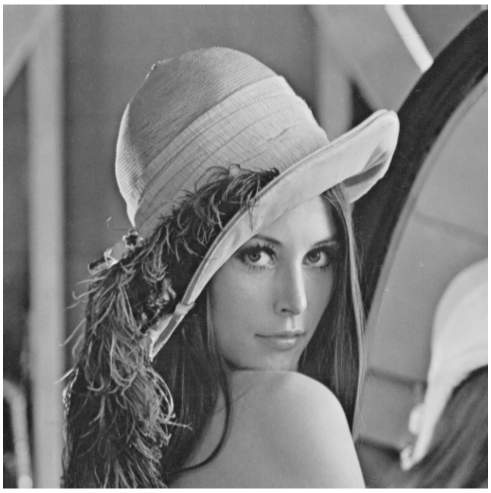


Figure 1: image for part(a).

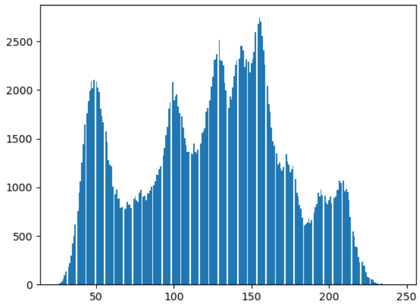
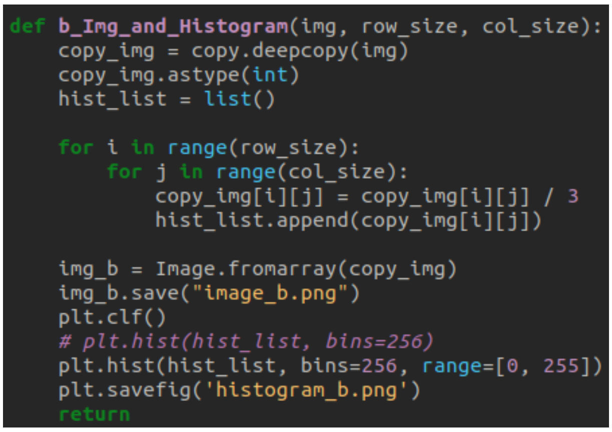


Figure 2: histogram image for part(a).

2. Image with intensity divided by 3 and its histogram

How to implement :

跟第一題一樣的手法，只是多了一個將所有Pixel除以3的動作



```
def b_Img_and_Histogram(img, row_size, col_size):
    copy_img = copy.deepcopy(img)
    copy_img.astype(int)
    hist_list = list()

    for i in range(row_size):
        for j in range(col_size):
            copy_img[i][j] = copy_img[i][j] / 3
            hist_list.append(copy_img[i][j])

    img_b = Image.fromarray(copy_img)
    img_b.save("image_b.png")
    plt.clf()
    # plt.hist(hist_list, bins=256)
    plt.hist(hist_list, bins=256, range=[0, 255])
    plt.savefig('histogram_b.png')
    return
```

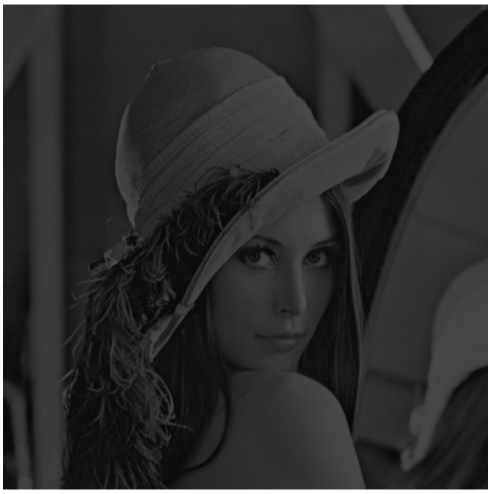


Figure 3: image for part(b).

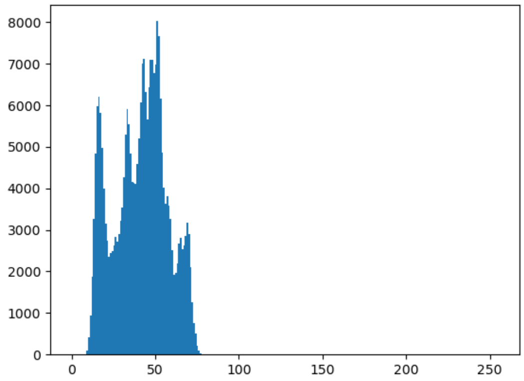
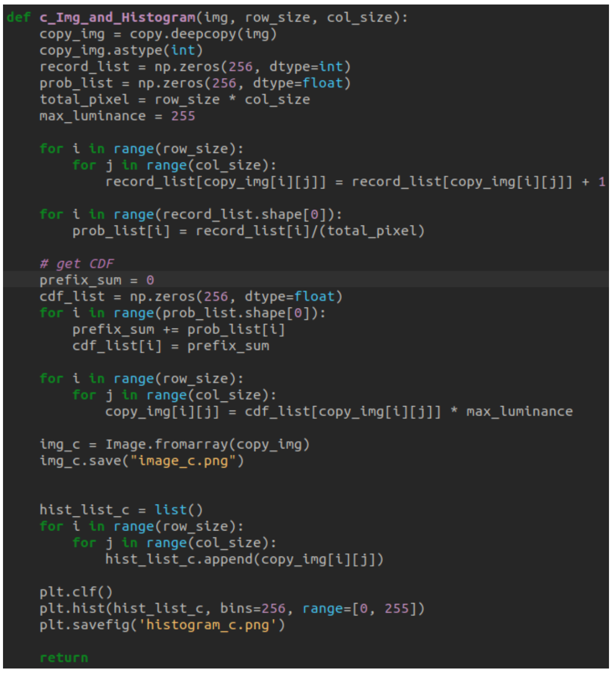


Figure 4: histogram image for part(b).

3. Image after applying histogram equalization to (b) and its histogram

How to implement :

使用b小題的結果當作輸入，用list紀錄每個Pixel的Intensity，再透過Prefix sum更新cdf list，最後再乘上最大亮度255



```python
def c_Img_and_Histogram(img, row_size, col_size):
    copy_img = copy.deepcopy(img)
    copy_img.astype(int)
    record_list = np.zeros(256, dtype=int)
    prob_list = np.zeros(256, dtype=float)
    total_pixel = row_size * col_size
    max_luminance = 255

    for i in range(row_size):
        for j in range(col_size):
            record_list[copy_img[i][j]] = record_list[copy_img[i][j]] + 1

    for i in range(record_list.shape[0]):
        prob_list[i] = record_list[i]/(total_pixel)

    # get CDF
    prefix_sum = 0
    cdf_list = np.zeros(256, dtype=float)
    for i in range(prob_list.shape[0]):
        prefix_sum += prob_list[i]
        cdf_list[i] = prefix_sum

    for i in range(row_size):
        for j in range(col_size):
            copy_img[i][j] = cdf_list[copy_img[i][j]] * max_luminance

    img_c = Image.fromarray(copy_img)
    img_c.save("image_c.png")


    hist_list_c = list()
    for i in range(row_size):
        for j in range(col_size):
            hist_list_c.append(copy_img[i][j])

    plt.clf()
    plt.hist(hist_list_c, bins=256, range=[0, 255])
    plt.savefig('histogram_c.png')

    return
```

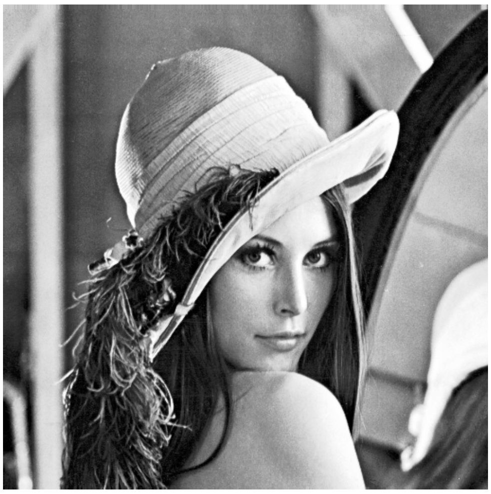


Figure 5: image for part(c).

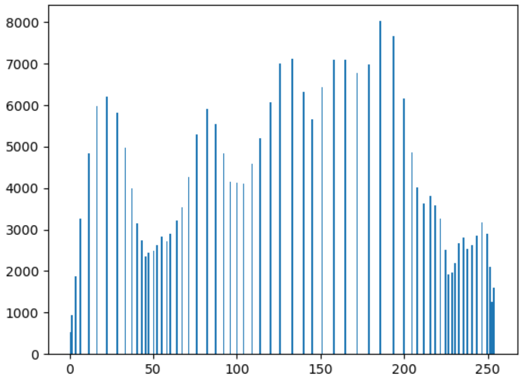


Figure 6: histogram image for part(c).