I. Problem Statement

- Write a program to generate images and histograms:
  (a) original image and its histogram
  (b) image with intensity divided by 3 and its histogram
  (c) image after applying histogram equalization to (b) and its histogram

II. Programming Tools

- Programming language: Python 3.8.5
- Library: Numpy 1.19.1, OpenCV 4.0.1, matplotlib 3.3.1

III. Problem-Solving Process

a. original image and its histogram

宣告一個 256 維的 numpy array，用來統計圖片 pixel value 的個數，在使用 matplotlib 中的 bar()畫出 histogram，並用 cv2.imwrite()輸出圖片。

```
if choice == 0: #  original image and its histogram
    statistic = np.zeros(256)
    r, c = img.shape
    for i in range(r):
        for j in range(c):
            statistic[img[i,j]] += 1
    cv2.imwrite('./output/1.jpg',img)
    plt.style.use('seaborn-white')
    plt.bar(range(256) ,statistic)
    plt.xlabel('pixel value')
    plt.ylabel('number')
    plt.savefig('./output/1_histogram.jpg')
    plt.clf()
```

b. image with intensity divided by 3 and its histogram

將圖片的每個 pixel value 除以三後無條件捨去，再藉由 a 小題的方式畫出 histogram，並用 cv2.imwrite()輸出圖片。

```
elif choice == 1: # image with intensity divided by 3 and its histogram
    ret = img//3
    statistic = np.zeros(256)
    r, c = ret.shape
    for i in range(r):
        for j in range(c):
            statistic[ret[i,j]] += 1
    cv2.imwrite('./output/2.jpg',ret)
    plt.style.use('seaborn-white')
    plt.bar(range(256) ,statistic)
    plt.xlabel('pixel value')
    plt.ylabel('number')
    plt.savefig('./output/2_histogram.jpg')
    plt.clf()
```

c. image after applying histogram equalization to (b) and its histogram

使用 b 小題的結果，並使 numpy 中的 cumsum()對統計資料進行累加，取出 cdf_min 與 cdf_max 後，再對圖片的每個 pixel 做 histogram equalization，最後 cv2.imwrite()輸出圖片。

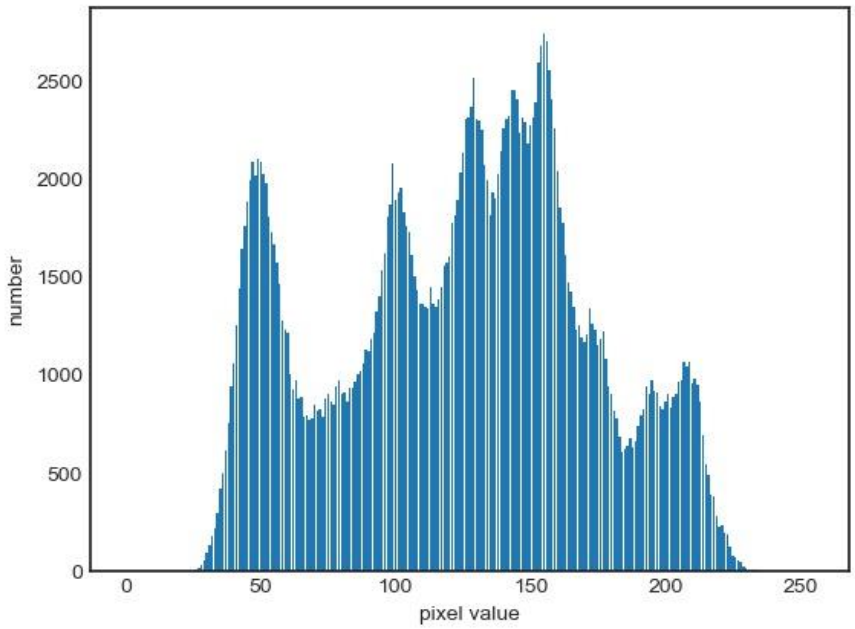
```
elif choice == 2: # image after applying histogram equalization to (b) and its histogram
    img = img//3
    ret = np.zeros_like(img)
    statistic = np.zeros(256)
    statistic_ret = np.zeros(256)
    r, c = ret.shape
    for i in range(r):
        for j in range(c):
            statistic[img[i,j]] += 1
    statistic = np.cumsum(statistic)
    cdf_min = min(statistic)
    cdf_max = max(statistic)

    for i in range(r):
        for j in range(c):
            ret[i,j] = round((statistic[img[i,j]] - cdf_min)/(cdf_max-cdf_min)*255)
            statistic_ret[ret[i,j]] += 1
    cv2.imwrite('./output/3.jpg',ret)
    plt.style.use('seaborn-white')
    plt.bar(range(256) ,statistic_ret)
    plt.xlabel('pixel value')
    plt.ylabel('number')
    plt.savefig('./output/3_histogram.jpg')
```

IV. Results

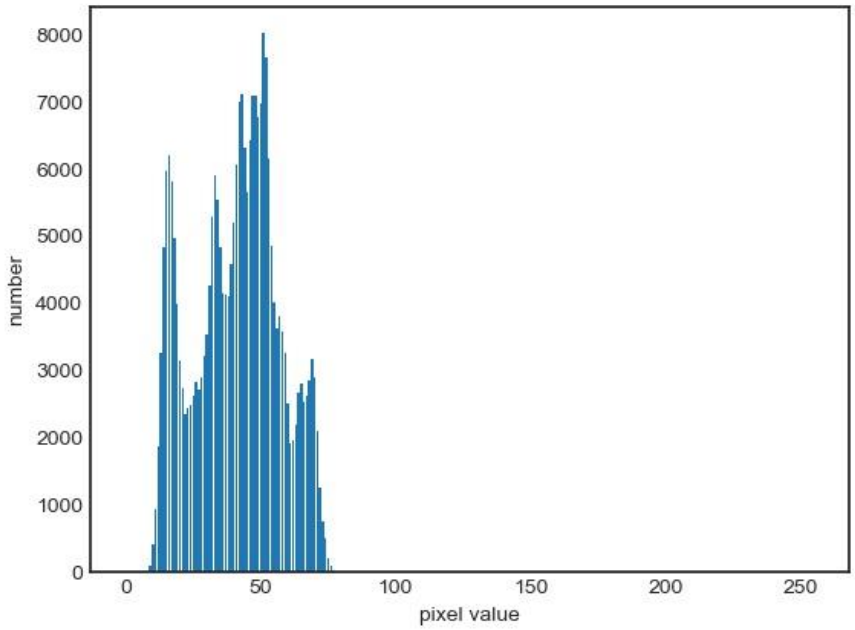
a. original image and its histogram

資工所 r08922123
Author: 王韻豪



b. image with intensity divided by 3 and its histogram

資工所 r08922123
Author: 王韻豪



c. image after applying histogram equalization to (b) and its histogram

資工所 r08922123
Author: 王韻豪