

(i) gray image

source code:

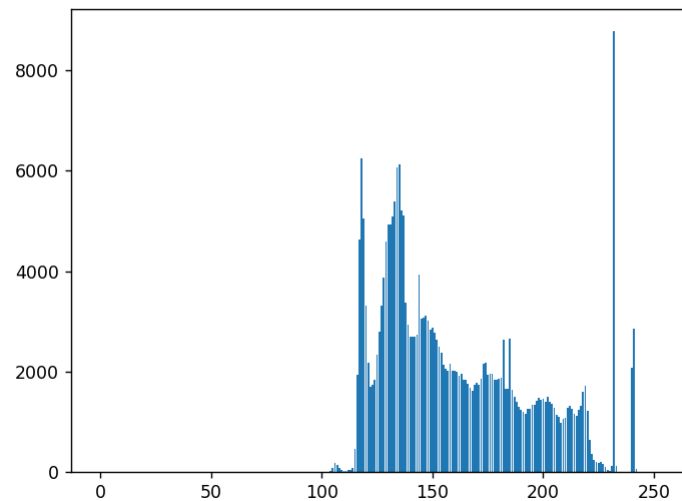
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib.image as mpimg
4 import cv2
5
6 # read the image
7 img = mpimg.imread("gray_input.jpg")
8
9 # count to draw diagram
10 color_cnt = [0] * 256
11 pixel_cnt = 0.0
12
13 for i in img:
14     for j in i:
15         color_cnt[j[0]] += 1
16         pixel_cnt += 1.0
17
18 # put data into plt
19 plt.bar([i for i in range(256)], color_cnt)
20 # show the diagram
21 plt.show()
22
23 # calculate the sum
24 sum_cnt = [0.0] * 256
25
26 for i in range(256):
27     if i != 0:
28         sum_cnt[i] += sum_cnt[i - 1] + float(color_cnt[i])
29
30 for i in range(256):
31     sum_cnt[i] /= pixel_cnt
32     sum_cnt[i] *= 255.0
33     sum_cnt[i] = round(sum_cnt[i])
34
35 # redraw the picture
36 y = len(img)
37 x = len(img[0])
38 for i in range(y):
39     for j in range(x):
40         for k in range(3):
41             img[i][j][k] = sum_cnt[img[i][j][k]]
42
43 # show after picture
44 plt.imshow(img)
45 # disable axis
46 plt.axis('off')
47 # save image's snapshot
48 plt.savefig('gray_output.jpg')
49 # show the image
50 plt.show()
```

```

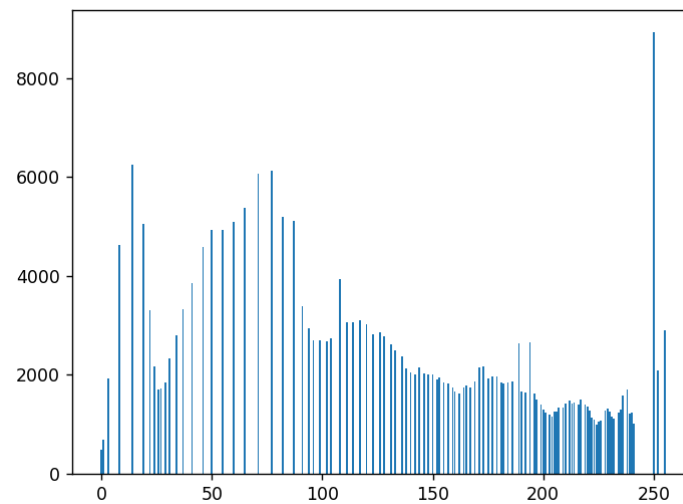
51
52
53 # after count to draw diagram
54 color_cnt = [0] * 256
55
56 for i in img:
57     for j in i:
58         color_cnt[j[0]] += 1
59
60 # put data into plt
61 plt.bar([i for i in range(256)], color_cnt)
62 # show the diagram
63 plt.show()

```

input:



output:



(ii) color image

source code:

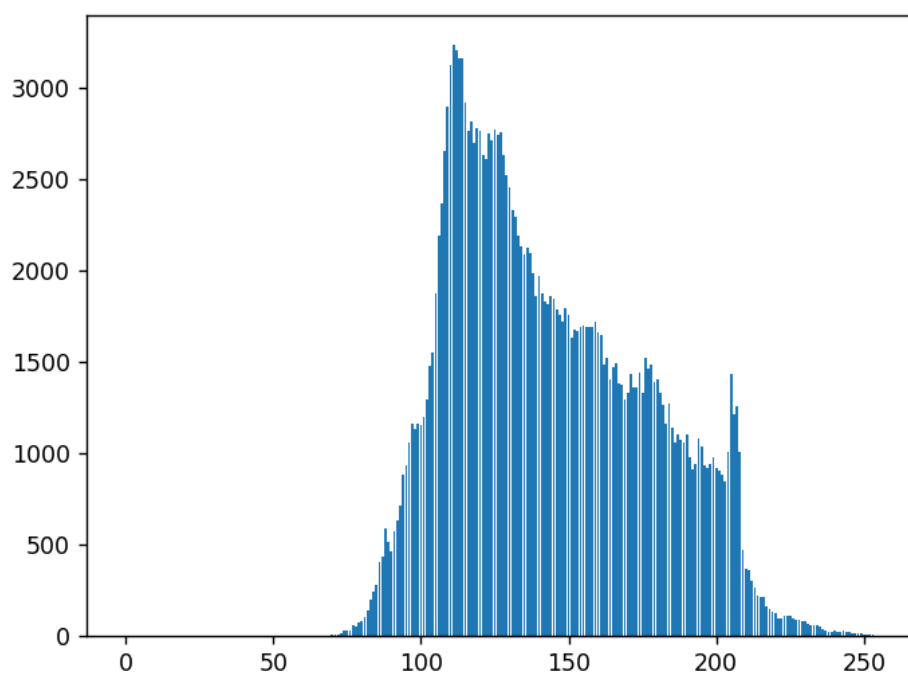
```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  import matplotlib.image as mpimg
4  import cv2
5
6  def rgb_to_gray(rgb):
7      return np.dot(rgb[...,:3], [0.2989, 0.5870, 0.1140])
8
9  # read the image
10 img = mpimg.imread("color_input.jpg")
11
12 # turn into gray image
13 gray_img = rgb_to_gray(img)
14
15 # round the picture
16 gray_img = np.around(gray_img)
17
18 # count to draw diagram
19 color_cnt = [0] * 256
```

```

20 pixel_cnt = 0.0
21
22 for i in gray_img:
23     for j in i:
24         color_cnt[int(j)] += 1
25         pixel_cnt += 1.0
26
27 # put data into plt
28 plt.bar([i for i in range(256)], color_cnt)
29 # show the diagram
30 plt.show()
31
32 # calculate the sum
33 sum_cnt = [0.0] * 256
34
35 for i in range(255):
36     if i != 0:
37         sum_cnt[i] += sum_cnt[i - 1] + float(color_cnt[i])
38
39 for i in range(255):
40     sum_cnt[i] /= pixel_cnt
41     sum_cnt[i] *= 255.0
42     sum_cnt[i] = round(sum_cnt[i])
43
44 # redraw the picture
45 y = len(img)
46 x = len(img[0])
47 for i in range(y):
48     for j in range(x):
49         for k in range(3):
50             if sum_cnt[int(gray_img[i][j])] != 0.:
51                 ration = sum_cnt[int(gray_img[i][j])] / gray_img[i][j]
52                 if int(img[i][j][k]) * ration > 255.0:
53                     img[i][j][k] = 255
54                 else:
55                     img[i][j][k] = round(img[i][j][k] * ration)
56
57 # show after picture
58 plt.imshow(img)
59 # disable axis
60 plt.axis('off')
61 # save image's snapshot
62 plt.savefig('color_output.jpg')
63 # show the image
64 plt.show()
65
66
67 # after count to draw diagram
68 color_cnt = [0] * 256
69 print(img)
70 for i in img:
71     for j in i:
72         color_cnt[round((int(j[0]) + int(j[1]) + int(j[2])) / 3)] += 1
73
74 # put data into plt
75 plt.bar([i for i in range(256)], color_cnt)
76 # show the diagram
77 plt.show()

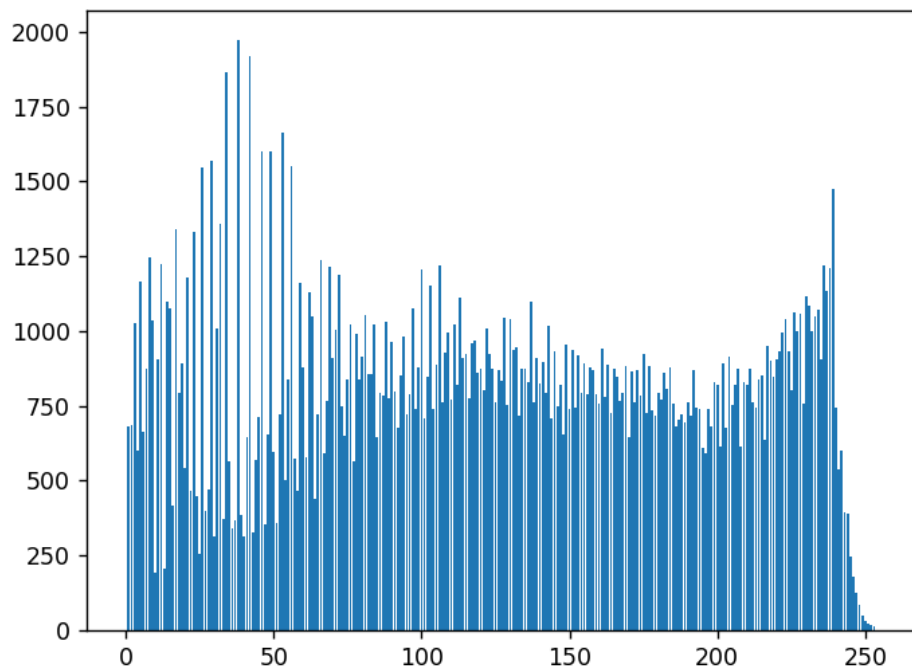
```

input:



output:





心得

這次作業蠻難的，特別是在處理彩色影像的時候。雖然只是灰階的加強版，但要考慮到有些顏色會有超過 255 的可能，所以需要去做額外判斷。還有在除 G' 的時候也要小心有些顏色是 0 而造成除法上的錯誤。