

HW1

Histogram

Equalization

405420083 劉承祐

Due date: April 21, 2020

Date handed in: April 20, 2020

■ Technical description

File: main.py

```
13 import matplotlib.pyplot as plt
14 import numpy as np
15
16
17 if input('請選擇圖片 0: Lena.bmp 1. Peppers.bmp : ')=='0':
18     imName='Lena.bmp'
19 else:
20     imName='Peppers.bmp'
21
22 if input('請選擇方法 0: 全域 1. 局部 : ')=='0':
23     isGlobal=True
24 else:
25     isGlobal=False
26
27
28 img = plt.imread(imName, format='bmp')
29 # img.shape
30 # (M, N, 4) for RGBA images.
```

Line 13~28: import，選擇，讀檔

```
33 def getCDF(chunk, chunk_size, RGBA):
34     # count PDF
35     list_CDF = [0 for _ in range(256)] # 這裡的 range(256) 是 image 位元深度 2**8=256
36     for i in range(chunk_size):
37         for j in range(chunk_size):
38             list_CDF[ chunk[i][j][RGBA] ] +=1
39     list_PDF = list_CDF.copy()
40     # PDF to CDF
41     for i in range(chunk_size -1):
42         list_CDF[i+1] += list_CDF[i]
43     return list_PDF, list_CDF
```

Line 33~43: function getCDF(二維圖片陣列，陣列邊長)

Line 36: for 遍歷圖片找對應值

Line 41: for 遍歷陣列加前值到後值，轉換成 CDF

```

45 def getHV(chunk, chunk_size, RGBA=0):
46     pdf, data = getCDF(chunk, chunk_size, RGBA)
47
48     plt.title('color: {0}'.format('R' if RGBA==0 else 'G' if RGBA==1 else 'B')) # 寫個title而已讓我偷懶一下
49     plt.ylabel('times for X') # y label
50     plt.xlabel('0 - 255') # x label
51     line1, = plt.plot(range(256), pdf[:,], color = 'red', label = 'PDF')
52     line2, = plt.plot(range(256), data[:,], color = 'blue', label = 'CDF')
53     plt.legend(handles = [line1, line2], loc='upper right')
54     plt.show()
55
56     hv = []
57     for i in range(chunk_size):
58         inner = []
59         for j in range(chunk_size):
60             c = data[ chunk[i][j][RGBA] ]
61             inner.append( round( ( c -min(data)) /(max(data) -min(data)) *255) )
62         hv.append(inner)
63     return hv

```

Line 33~43: function getHV(二維圖片陣列, 陣列邊長)

Line 48~54: 顯示 PDF、CDF 對照表

Line 56~62: H(v)公式

```

65 # [R-2Dtable, G-2Dtable, B-2Dtable] to [[R,G,B], ]-2Dtable
66 def RtGtBt2RGBt(table, leng):
67     output = []
68     for i in range(leng):
69         inner = []
70         for j in range(leng):
71             inner.append([table[0][i][j], table[1][i][j], table[2][i][j]])
72         output.append(inner)
73     return output

```

[[R, R, R...]],
[[G, G, G...]],
[[B, B, B...]]

➡

[[R, G, B],
[[R, G, B],
[[R, G, B]...

Line 66~73: 陣列轉換，如圖右方

```

89 if isGlobal:
90     R_G_B = [ getHV(img, img.shape[0], 0),
91               getHV(img, img.shape[0], 1),
92               getHV(img, img.shape[0], 2) ]
93
94     x = RtGtBt2RGBt(R_G_B, img.shape[0])
95
96     x = np.array(x, np.uint8)
97     plt.imsave('{0}_GlobalTransform.png'.format(imName), x, format='png')

```



Line 89~97: 選擇全域轉換，RGB 依次執行 Histogram Equalization、陣列轉格式、轉換後圖片存檔

```

98 else:
99     # img_eqSize = input('請輸入要把 image 切成幾等份(邊長切): ')
100    img_eqSize = 4
101    coeff = int(img.shape[0] / img_eqSize)
102
103    ## 二維陣列等份拆解 - 存入一維陣列中
104    def insert_image(img, output, posY, posX, coeff): ## Y在X前面對應
105        outer = []
106        for i in range(coeff):
107            tmp = []
108            for j in range(coeff):
109                tmp.append(img[i + posY * coeff][j + posX * coeff])
110            outer.append(tmp)
111        output.append(outer) # 使 output[0][1]...都是二維
112
113    ## 二維陣列等份拆解 - 存入一維陣列中
114    img_eqed = []
115    for i in range(img_eqSize):
116        for j in range(img_eqSize):
117            insert_image(img, img_eqed, i, j, coeff)

```

11	12	13	14	15	16
21	22	23	24	25	26
31	32	33	34	35	36
41	42	43	44	45	46
51	52	53	54	55	56
61	62	63	64	65	66

arr[0~9]
arr[0] 11 12
21 22
arr[1] 13 14
23 24

Line 100~117: 陣列等份

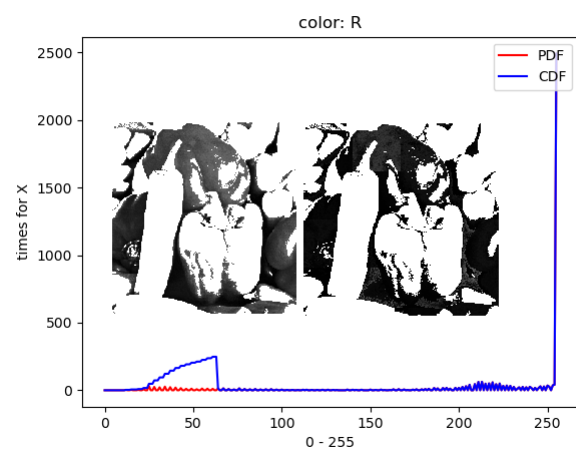
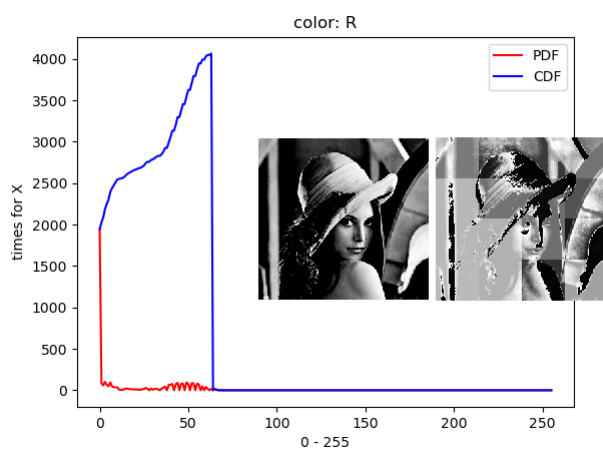
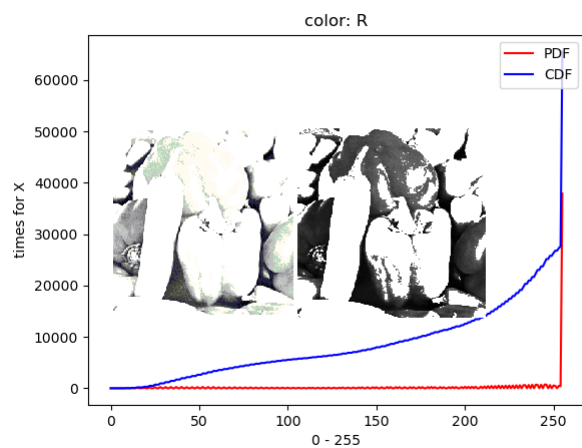
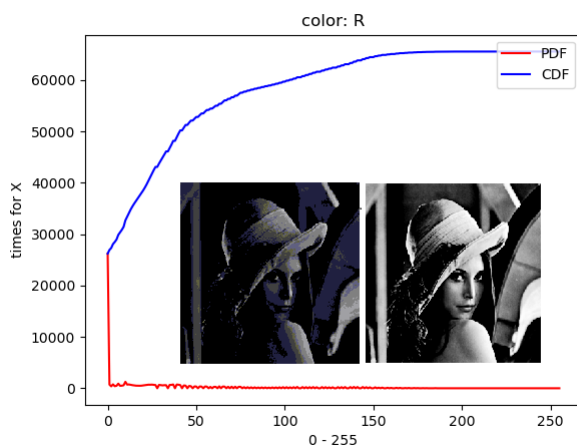
```

125     # 拆解完後的二維陣列，就跟 isGlobal=1 一樣處理就可以
126     for count in range(img_eqSize ** 2):
127         R_G_B = [ getHV(img_eqed[count], coeff, 0),
128                 getHV(img_eqed[count], coeff, 1),
129                 getHV(img_eqed[count], coeff, 2) ]
130
131         x = RtGtBt2RGBt(R_G_B, coeff)
132
133         x = np.array(x, np.uint8)
134         plt.imsave('{0}_BlockTransform{1}.png'.format(imName, count), x, format='png')

```

Line 127~134: 跟 Global 一樣

■ Experimental results



■ Discussions

Block Histogram Equalization 不知道對不對，看起來感覺比做之前還醜

■ References and Appendix

實作參考網址(無抄襲)

- <http://iris123321.blogspot.com/2017/05/histogram-equalization.html>
- <https://zh.wikipedia.org/wiki/直方圖均衡化>

requirements.txt

- matplotlib
- pillow
- numpy