

實習四

影像強化處理



課程大綱

- 實習00: Colab 環境
- 實習04: 影像強化處理





實習 00

Colab 環境

Colab Env.

Before we start...

```
1 | #mount drive
2 | from google.colab import drive
3 | drive.mount('/content/drive')

4 | # import libraries
5 | import sys
6 | import os
7 | import cv2
8 | import numpy as np
9 | from matplotlib import pyplot as plt
10 | from google.colab.patches import cv2_imshow
```



實習 04

影像強化處理

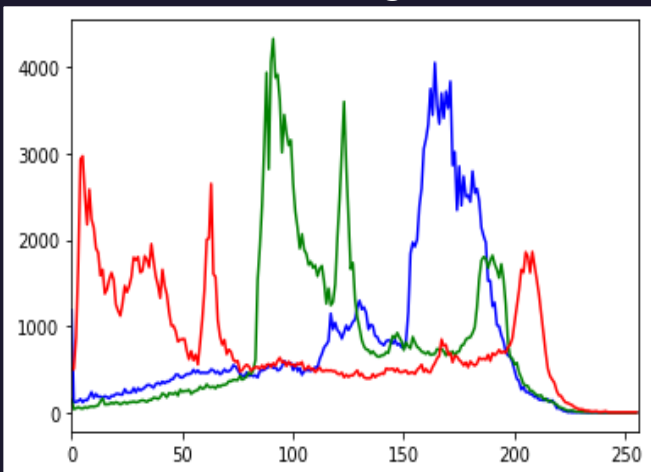
TASK: 彩色/灰階影像的直方圖繪製

- 使用 `cv2.imread()` 讀取一張影像。
- 使用 `cv2.calcHist(影像, 通道, 遮罩, 區間數量, 數值範圍)` 取得

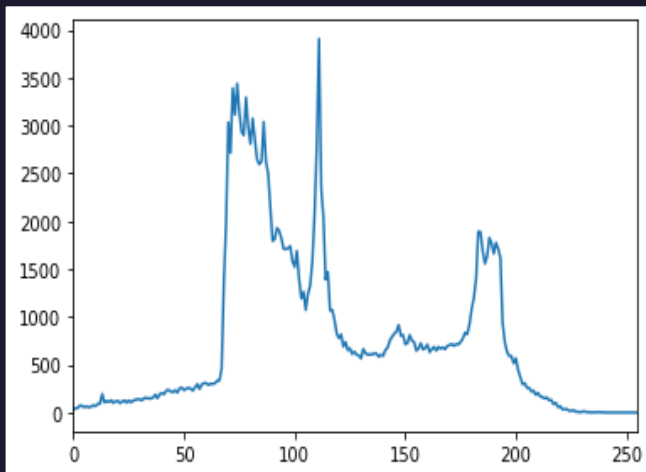
實習影像下載: <https://reurl.cc/gQ2gYR>



RGB image



Gray-scale Image



```
1 # import libraries
2 import sys
3 import os
4 import cv2
5 import numpy as np
6 from matplotlib import pyplot as plt
7 from google.colab.patches import cv2_imshow
```

```
7 # read an image by cv2.imread()
8 folder = r'/content/drive/MyDrive/images'
9 path_img = os.path.join(folder, 'home.jpg')
10 img = cv2.imread(path_img)
11 # Afterwards, a check is executed, if the image was loaded correctly.
12 if img is None:
13     sys.exit("Could not read the image.")
14 cv2_imshow(img)
```

```
14 # Calculate it
15 color = ('b', 'g', 'r')
16 for i, col in enumerate(color):
17     hist = cv2.calcHist([img], [i], None, [256], [0, 255])
18     plt.plot(hist, color=col)
19     plt.xlim([0, 255])
20 plt.show()
```

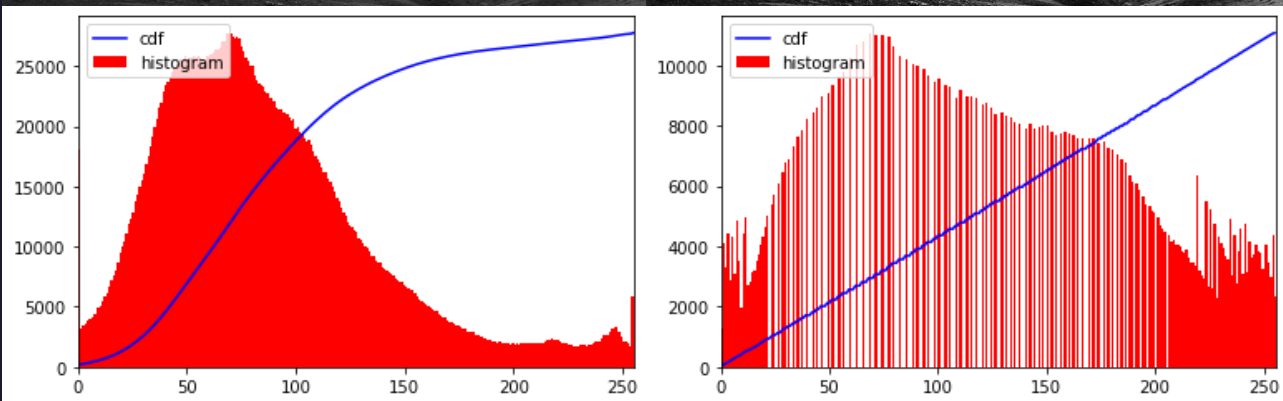
```
21 # Calculate it in gray-scale image
22 img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
23 cv2_imshow(img_gray)
24 hist = cv2.calcHist([img_gray], [0], None, [256], [0, 255])
25 plt.plot(hist)
26 plt.xlim([0, 255])
27 plt.show()
```

TASK: 影像的直方圖均等化

Histogram Equalization (HE)

To equalize histograms of images by using the OpenCV function `cv::equalizeHist`

- More information:
 - Opencv document: `equalizeHist()` [#link](#)
 - Opencv document: Histogram Equalization [#link](#)



原始灰階影像

HE 影像

實習影像下載: <https://reurl.cc/Ay7Ebj>

```
1 # read an image
2 folder = r'/content/drive/MyDrive/images'
3 path_img = os.path.join(folder, 'Hawkes_Bay_NZ.jpg')
4 img = cv2.imread(path_img)
5 # Afterwards, a check is executed, if the image was loaded correctly.
6 if img is None:
7     sys.exit("Could not read the image.")
8 cv2_imshow(img)
```

```
9 # transfer to gray-scale image
10 img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
11 # Histogram Calculation
12 hist = cv2.calcHist([img_gray], [0], None, [256], [0, 255])
13 plt.plot(hist)
14 plt.xlim([0, 255])
15 plt.show()
```

```
16 # histogram equalization by cv2.equalizeHist()
17 equ = cv2.equalizeHist(img_gray)
18 res = np.hstack((img_gray, equ))
19 cv2_imshow(res)
```

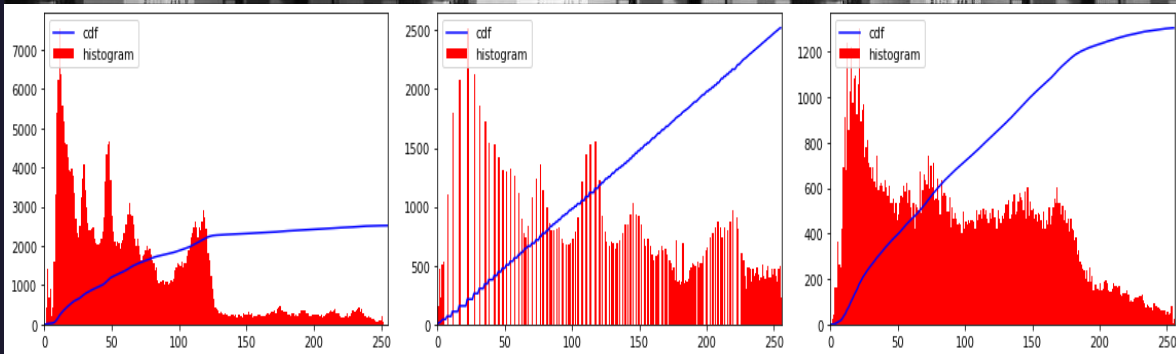

TASK: 影像的限制對比度自適應直方圖等化(CLAHE)

CLAHE (Contrast Limited Adaptive Histogram Equalization)

```
cv.createCLAHE( [ , clipLimit[, tileGridSize]] ) -> retval
```

- Parameters:
 - `clipLimit`: Threshold for contrast limiting.
 - `tileGridSize`: Size of grid for histogram equalization. Input image will be divided into equally sized rectangular tiles. `tileGridSize` defines the number of tiles in row and column.

實習影像下載: <https://reurl.cc/bG2ZNy>



```
1 # read an image
2 folder = r'/content/drive/MyDrive/images'
3 path_img = os.path.join(folder, 'tsukuba_1.png')
4 img = cv2.imread(path_img)
5 # Afterwards, a check is executed, if the image was loaded correctly.
6 if img is None:
7     sys.exit("Could not read the image.")
8 cv2.imshow(img)
9 # transfer to gray-scale image
10 img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
1 # HE
2 equ = cv2.equalizeHist(img_gray)
3 # create a CLAHE object (Arguments are optional).
4 clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
5 cla = clahe.apply(img_gray)
6 # display
7 res = np.hstack((img_gray, equ, cla))
8 cv2.imshow(res)
```

```
1 # cumulative distribution function (cdf)
2 hist, bins = np.histogram(img_gray.flatten(), 256, [0, 255])
3 cdf = hist.cumsum()
4 cdf_normalized = cdf * float(hist.max()) / cdf.max()
5 plt.plot(cdf_normalized, color = 'b')
6 plt.hist(img_gray.flatten(), 256, [0, 255], color = 'r')
7 plt.xlim([0, 255])
8 plt.legend(('cdf', 'histogram'), loc = 'upper left')
9 plt.show()
10 # cumulative distribution function (cdf)
11 hist, bins = np.histogram(equ.flatten(), 256, [0, 255])
12 cdf = hist.cumsum()
13 cdf_normalized = cdf * float(hist.max()) / cdf.max()
14 plt.plot(cdf_normalized, color = 'b')
15 plt.hist(equ.flatten(), 256, [0, 255], color = 'r')
16 plt.xlim([0, 256])
17 plt.legend(('cdf', 'histogram'), loc = 'upper left')
18 plt.show()
19 # cumulative distribution function (cdf)
20 hist, bins = np.histogram(cla.flatten(), 256, [0, 255])
21 cdf = hist.cumsum()
22 cdf_normalized = cdf * float(hist.max()) / cdf.max()
23 plt.plot(cdf_normalized, color = 'b')
24 plt.hist(cla.flatten(), 256, [0, 255], color = 'r')
25 plt.xlim([0, 256])
26 plt.legend(('cdf', 'histogram'), loc = 'upper left')
27 plt.show()
```




Thanks for listening

Thank You

詹宏澤

Email: chanhts323@gmail.com

<https://sites.google.com/view/peter-chan>

