Histogram Matched 1106022 陳柏嘉

原圖 Original：

```
image = cv2.imread('img/GT86.jpg')
```

進行灰階轉換 Original_gray：

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

將 Original_gray 進行 Histogram Equalization 得到 washed-out 的圖片：

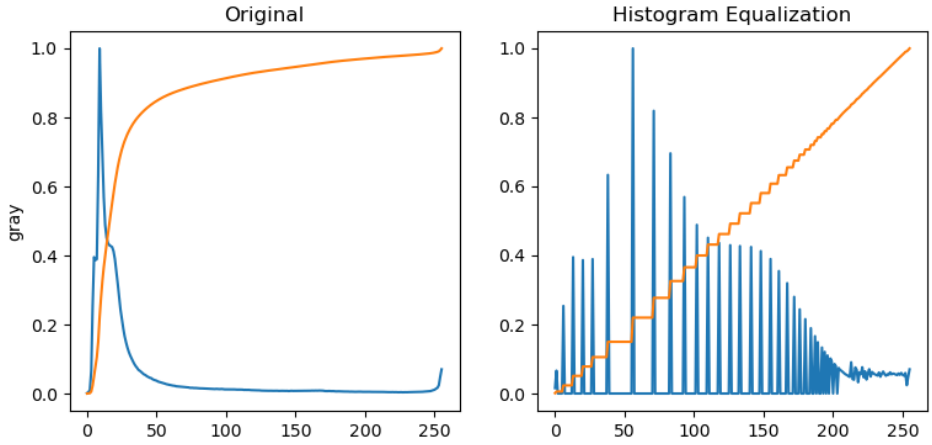
```
he = cv2.equalizeHist(gray)
```

比較 Original_gray 與 Histogram Equalization 的 histogram 與 cdf：

```
fig, he_axes = plt.subplots(nrows=1, ncols=2, figsize=(9, 4))
for i, img in enumerate((gray, he)):
    img_hist, bins = exposure.histogram(img[...], source_range='dtype')
    he_axes[i].plot(bins, img_hist / img_hist.max())
    img_cdf, bins = exposure.cumulative_distribution(img[...])
    he_axes[i].plot(bins, img_cdf)
    he_axes[0].set_ylabel('gray')

he_axes[0].set_title('Original')
he_axes[1].set_title('Histogram Equalization')
```

因為 Original_gray 本身為過暗的圖片，因此經過 Histogram Equalization 計算灰階值 0~225 的累積機率乘上 255 並四捨五入，得到映射的灰階值，其 cdf 為斜率 =1 的直線。因暗部的像素點被強制分配到較亮的區間，使圖片變得過亮 (washed-out)。



Histogram Matched 的 Reference 圖片：

```
reference = cv2.imread('img/S2000.jpg')
```

Reference_gray 圖片：

```
reference_gray = cv2.cvtColor(reference, cv2.COLOR_BGR2GRAY)
```

Original 經過 Histogram Matched 得到的圖片：

```
matched = match_histograms(image, reference, multichannel=multi)
```

在將 matched 轉為灰階圖片 matched_gray：

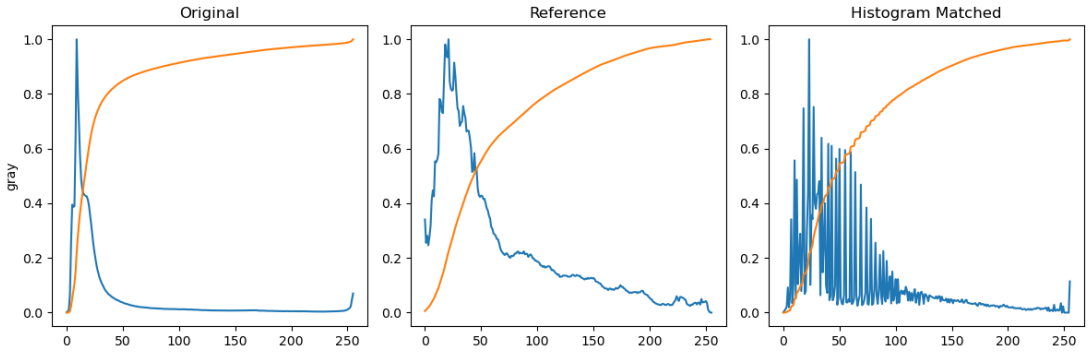
```
matched_gray = cv2.cvtColor(matched, cv2.COLOR_BGR2GRAY)
```

比較 Original_gray, Reference_gray 與 Histogram Matched 的 histogram 與 cdf：

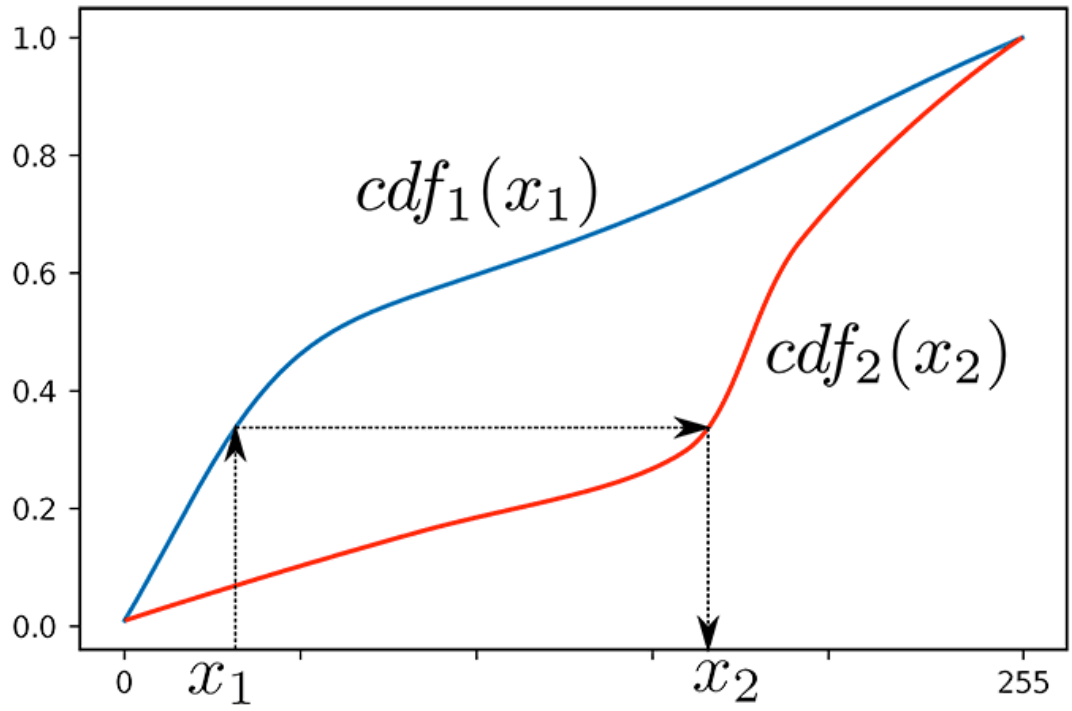
```
fig, hm_axes = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))
for i, img in enumerate((gray, reference_gray, matched_gray)):

    img_hist, bins = exposure.histogram(img[...], source_range='dtype')
    hm_axes[i].plot(bins, img_hist / img_hist.max())
    img_cdf, bins = exposure.cumulative_distribution(img[...])
    hm_axes[i].plot(bins, img_cdf)
    hm_axes[0].set_ylabel('gray')


hm_axes[0].set_title('Original')
hm_axes[1].set_title('Reference')
hm_axes[2].set_title('Histogram Matched')
```



Histogram Matched 原理：



來源

將原圖的灰階值經由cdf_1做 Histogram Equalization 後，再用cdf_2做逆 Histogram Equalization 映射到新的灰階值。能將 Original 的灰階分佈轉換成與 Reference 的灰階分佈一致，獲得與 Reference 色調相似的圖片。

結語：

在做 Histogram Matched 的時候，整個過程是非常有趣的，雖然在尋找適合的 Reference 素材時花了不少時間。在最後得出的結果還算滿意，整體動態範圍提升不少。

途中也試了不少有趣的組合~

Python Code：

```python
import cv2
import matplotlib.pyplot as plt

from skimage import exposure
from skimage.exposure import match_histograms

# 圖片讀取
image = cv2.imread('img/GT86.jpg')
multi = True if image.shape[2] > 1 else False

reference = cv2.imread('img/S2000.jpg')


# 將原圖轉為灰階
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("Gray", gray)
reference_gray = cv2.cvtColor(reference, cv2.COLOR_BGR2GRAY)

# histogram equalization
he = cv2.equalizeHist(gray)
cv2.imshow("Histogram Equalization", he)

# Histogram Matching
matched = match_histograms(image, reference, multichannel=multi)
matched_gray = cv2.cvtColor(matched, cv2.COLOR_BGR2GRAY)
cv2.imshow("matched", matched_gray)

# clahe
clahe = cv2.createCLAHE(clipLimit=5, tileGridSize=(8,8))
clahe_img = clahe.apply(gray)
cv2.imshow("clahe", clahe_img)

# 儲存
cv2.imwrite("1_gray.jpg", gray)
cv2.imwrite("2_he.jpg", he)
cv2.imwrite("3_clahe.jpg", clahe_img)
cv2.imwrite("4_matched.jpg", matched)
```

```
cv2.imwrite("5_matched_gray.jpg", matched_gray)
cv2.imwrite("6_reference_gray.jpg", reference_gray)

# Original HE 的比較
fig, he_axes = plt.subplots(nrows=1, ncols=2, figsize=(9, 4))
for i, img in enumerate((gray, he)):
    img_hist, bins = exposure.histogram(img[...], source_range='dtype')
    he_axes[i].plot(bins, img_hist / img_hist.max())
    img_cdf, bins = exposure.cumulative_distribution(img[...])
    he_axes[i].plot(bins, img_cdf)
    he_axes[0].set_ylabel('gray')

he_axes[0].set_title('Original')
he_axes[1].set_title('Histogram Equalization')

plt.savefig('Histogram Equalization.png')
plt.show()

# Original Clahe 的比較
fig, clahe_axes = plt.subplots(nrows=1, ncols=2, figsize=(9, 4))
for i, img in enumerate((gray, clahe_img)):
    img_hist, bins = exposure.histogram(img[...], source_range='dtype')
    clahe_axes[i].plot(bins, img_hist / img_hist.max())
    img_cdf, bins = exposure.cumulative_distribution(img[...])
    clahe_axes[i].plot(bins, img_cdf)
    clahe_axes[0].set_ylabel('gray')

clahe_axes[0].set_title('Original')
clahe_axes[1].set_title('Clahe')

plt.savefig('clahe.png')
plt.show()

# Original HM 的比較
fig, hm_axes = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))
for i, img in enumerate((gray, reference_gray, matched_gray)):

    img_hist, bins = exposure.histogram(img[...], source_range='dtype')
```

```
    hm_axes[i].plot(bins, img_hist / img_hist.max())
    img_cdf, bins = exposure.cumulative_distribution(img[...])
    hm_axes[i].plot(bins, img_cdf)
    hm_axes[0].set_ylabel('gray')


hm_axes[0].set_title('Original')
hm_axes[1].set_title('Reference')
hm_axes[2].set_title('Histogram Matched')

plt.tight_layout()
plt.savefig('comparison.png')
plt.show()


if (cv2.waitKey(0)==27):
    cv2.destroyAllwindows()
```