

---本次題目因為環環相扣，故小題之間程式碼就不做分開撰寫---

code link:

<https://colab.research.google.com/drive/1L5XuxbpgwSap7UzPJkeRakphqRjNGlSp?usp=sharing>

Q8 程式碼如下

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 # 計算圖片的直方圖(a)小題
5 def compute_histogram(image):
6
7     histogram = cv2.calcHist([image], [0], None, [256], [0, 256])
8     return histogram.flatten()
9
10 # 對圖片做histogram equalization
11 def histogram_equalization(image):
12     #計算其直方圖
13     histogram = compute_histogram(image)
14
15     # 算他的CDF
16     cdf = histogram.cumsum()
17     cdf_normalized = cdf * histogram.max() / cdf.max()
18
19     # pixel value mapping到cdf
20     equalized_image = np.interp(image.flatten(), np.arange(256), cdf_normalized).reshape(image.shape)
21
22     return equalized_image
```

```
23
24 image_path = "~/content/Fig0308(a) (fractured_spine).tif"
25 image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
26
27 image = np.uint8(image)
28
29 # 原圖的直方圖
30 histogram_original = compute_histogram(image)
31 # 並且對圖做histogram equalization
32
33 equalized_image = histogram_equalization(image)
34 equalized_image = np.uint8(equalized_image)
35
36 # equalized image的直方圖
37 histogram_equalized = compute_histogram(equalized_image)
38
39 #以下繪圖
40 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 4))
41
42
43 ax1.imshow(image, cmap='gray')
44 ax1.set_title('Original Image')
45 ax1.axis('off')
46
47
48 ax2.plot(histogram_original)
49 ax2.set_title('Histogram of Original Image')
50 ax2.set_xlabel('Pixel Value')
51 ax2.set_ylabel('Frequency')
```

```

54 fig2, (ax3, ax4) = plt.subplots(1, 2, figsize=(10, 4))
55
56 |
57 ax3.imshow(equalized_image, cmap='gray')
58 ax3.set_title('Enhanced Image')
59 ax3.axis('off')
60
61
62 ax4.plot(histogram_equalized)
63 ax4.set_title('Histogram of Enhanced Image')
64 ax4.set_xlabel('Pixel Value')
65 ax4.set_ylabel('Frequency')
66
67 plt.tight_layout()
68 plt.show()

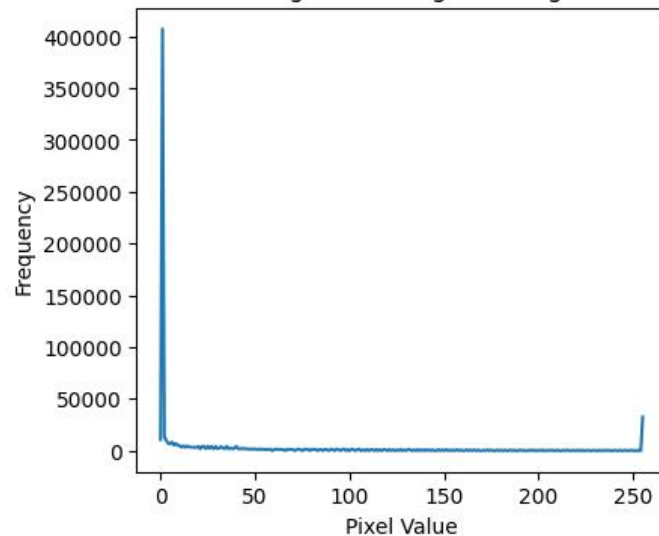
```

Result:

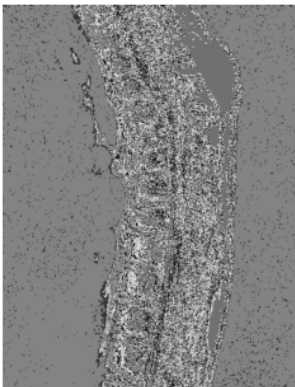
Original Image



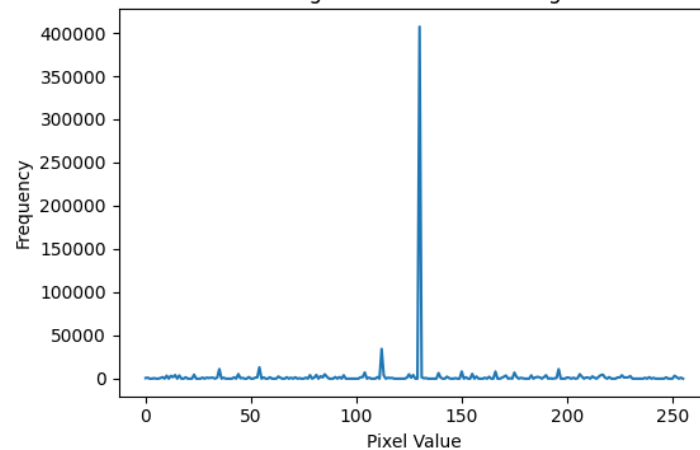
Histogram of Original Image



Enhanced Image



Histogram of Enhanced Image



Q9 程式碼如下

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5
6 image_path = "/content/Fig0338(a) (blurry_moon).tif"
7 image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
8
9 def spatial_filtering(image, mask):
10     filtered_image = cv2.filter2D(image, -1, mask)
11     return filtered_image
12
13 # 建立一個3*3 mask, 並且讓user自行輸入遮罩數值
14 laplacian_mask = np.zeros((3, 3), dtype=np.float32)
15 for i in range(3):
16     for j in range(3):
17         coefficient = float(input(f"Enter the coefficient at position ({i+1}, {j+1}): "))
18         laplacian_mask[i, j] = coefficient
19
20 # 這裡負責做 Laplacian enhancement
21 def laplacian_enhancement(image, laplacian_mask):
22     # Apply spatial filtering using the Laplacian mask
23     enhanced_image = spatial_filtering(image, laplacian_mask)
24
25     # Add the original image to the enhanced image to obtain the final result
26     final_image = cv2.add(image, enhanced_image)
27
28     return final_image
29
30
31 # 拿上面寫好的Laplacian enhancement function對圖片做增強
32 enhanced_image = laplacian_enhancement(image, laplacian_mask)
33
34 #把原圖和增強後的圖都秀出來
35 plt.subplot(1, 2, 1)
36 plt.imshow(image, cmap='gray')
37 plt.title('Original Image')
38 plt.axis('off')
39
40 plt.subplot(1, 2, 2)
41 plt.imshow(enhanced_image, cmap='gray')
42 plt.title('Enhanced Image')
43 plt.axis('off')
44
45 plt.show()
```

使用 3.37(a)的 mask(中心點-4)

```
Enter the coefficient at position (1, 1): 0
Enter the coefficient at position (1, 2): 1
Enter the coefficient at position (1, 3): 0
Enter the coefficient at position (2, 1): 1
Enter the coefficient at position (2, 2): -4
Enter the coefficient at position (2, 3): 1
Enter the coefficient at position (3, 1): 0
Enter the coefficient at position (3, 2): 1
Enter the coefficient at position (3, 3): 0
```

Original Image



Enhanced Image



使用 3.37(b)的 mask(中心點-8)

```
Enter the coefficient at position (1, 1): 1
Enter the coefficient at position (1, 2): 1
Enter the coefficient at position (1, 3): 1
Enter the coefficient at position (2, 1): 1
Enter the coefficient at position (2, 2): -8
Enter the coefficient at position (2, 3): 1
Enter the coefficient at position (3, 1): 1
Enter the coefficient at position (3, 2): 1
Enter the coefficient at position (3, 3): 1
```

Original Image



Enhanced Image



Q10 程式碼如下

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 image_path = "/content/Fig0340(a)(dipxe_text).tif"
6 image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
7
8 def spatial_filtering(image, mask):
9     # 使用 filter2D 函數進行空間濾波
10    filtered_image = cv2.filter2D(image, -1, mask)
11    return filtered_image
12
13 # 建立一個 3x3 的遮罩，並由使用者輸入遮罩數值
14 laplacian_mask = np.zeros_like(image, dtype=np.float32)
15 mask_sum = 0.0 # 儲存遮罩係數的總和
16 for i in range(3):
17     for j in range(3):
18         coefficient = float(input(f"Enter the coefficient at position ({i+1}, {j+1}): "))
19         laplacian_mask[i, j] = coefficient
20         mask_sum += coefficient
21
22 # 將遮罩進行正規化
23 laplacian_mask /= mask_sum
24
25 # 計算平均化處理後的影像
26 averaged_image = spatial_filtering(image, np.ones((3, 3), dtype=np.float32) / float(mask_sum))
27
28 # 執行高增幅濾波
29 k = float(input("Enter the value of k: "))
30 boosted_image = image + k * (image - averaged_image)
31
32 # 顯示原始影像
33 plt.subplot(1, 2, 1)
34 plt.imshow(image, cmap='gray')
35 plt.title('Original Image')
36 plt.axis('off')
37
38 # 顯示高增幅濾波後的影像
39 plt.subplot(1, 2, 2)
40 plt.imshow(boosted_image, cmap='gray')
41 plt.title('High-Boost Filtered Image')
42 plt.axis('off')
43
44 plt.tight_layout()
45 plt.show()
```

Result:課本使用  $k = 4.5$  作為 3.40(e)的示範，那經過比較原圖後可知，本例子使用高增幅濾波並將  $k = 4.5$  代入之後字體邊緣輪廓變得較明顯，但也有一點點粗糙的裂痕。

```
Enter the coefficient at position (1, 1): 1
Enter the coefficient at position (1, 2): 1
Enter the coefficient at position (1, 3): 1
Enter the coefficient at position (2, 1): 1
Enter the coefficient at position (2, 2): 1
Enter the coefficient at position (2, 3): 1
Enter the coefficient at position (3, 1): 1
Enter the coefficient at position (3, 2): 1
Enter the coefficient at position (3, 3): 1
Enter the value of k: 4.5
```

Original Image



High-Boost Filtered Image

