

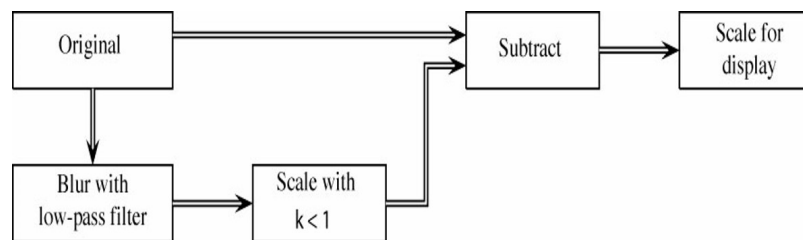
NTNU 影像處理 HW5

廖家緯

2020.4.15

- **Outline:**

1. CalSelect an experimental image
2. Apply a 3 by 3 (a) average filter and (b) median filter to the image
3. Unsharp masking



- **Code(Python):**

```
1 import numpy as np
2 import cv2
3
4 # 讀取灰階影像
5 I = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
6
7 # 周圍補0
8 I1 = np.pad(I, ((1,1),(1,1)), 'constant', constant_values = (0,0))
9 N = np.shape(I1)[0]
10 M = np.shape(I1)[1]
11
12 # Average filter 和 Median filter
13 I2 = np.zeros((N, M), int).astype('double')
14 I3 = np.zeros((N, M), int).astype('uint8')
15
16 for i in range(1, N-1):
17     for j in range(1, M-1):
18         I2[i,j] = I1[i-1:i+2, j-1:j+2].mean()
```

```

19         I3[i,j] = sorted(I1[i-1:i+2, j-1:j+2].reshape(9))[4]
20
21 I2 = I2.astype('uint8') # 經過 Average filter 掃過的影像
22 I3 = I3.astype('uint8') # 經過 Median filter 掃過的影像
23
24 # Unsharp masking
25 k = 0.4
26 s = 1/(1-k)
27 I4 = ((I1 - k*I2)*s).astype('uint8')
28 I5 = ((I1 - k*I3)*s).astype('uint8')

```

- Input image:

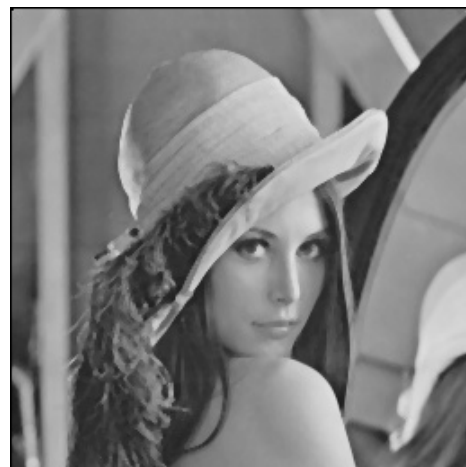


Original

- Result:



Average filter



Median filter



Unsharp Masking by average filter Unsharp Masking by median filter

- **Experience:**

在這次作業中，average filter 和 median filter 的製作原本可能需要 4 層迴圈，我使用 numpy 直接對矩陣逐項計算，因此減少到只需 2 層迴圈，大幅降低了時間複雜度。而處理 Unsharp masking，我嘗試了不同的 k ，最後選用看起來比較完美的 0.4，我們可以發現，經過 Unsharp masking 後，頭髮與眼睛的輪廓變得比較明顯。另外，為了擔心 Unsharp masking 後的影像會變得太暗，我根據課本提供的方法，乘上 s 值來調整明暗程度（課本 k 剛好與我的成倒數，因此我將公式重新調整過）。