

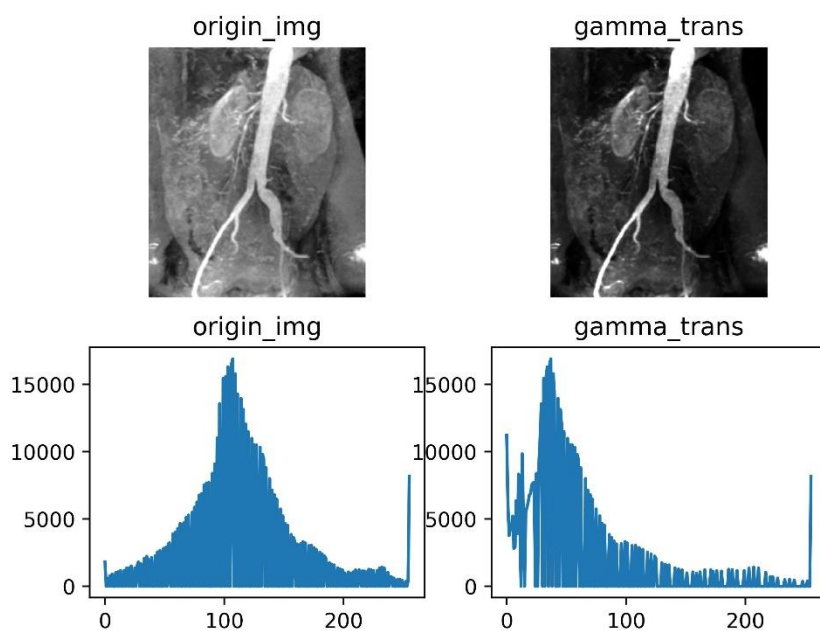
Problem1

In this assignment, you are asked to try Gamma transformation, histogram equalization, unsharp masking, and two additional methods of your choice to enhance the image shown on the right. For the two additional methods, you are encouraged to create your own based on the techniques you have learned from Chapter 3

Gamma transformation (課本 p125):

檔名: 1_gamma_transformation.jpg

使用 p125 的公式 $s = cr^{gamma}$ ，這裡取 $c=1$ 、 $gamma=2.2$ ，先將要對應的值畫成表格，然後依照 input image 的 intensity 去查表。可以看出來對比度增加，血管內的灰色細節也更明顯了。

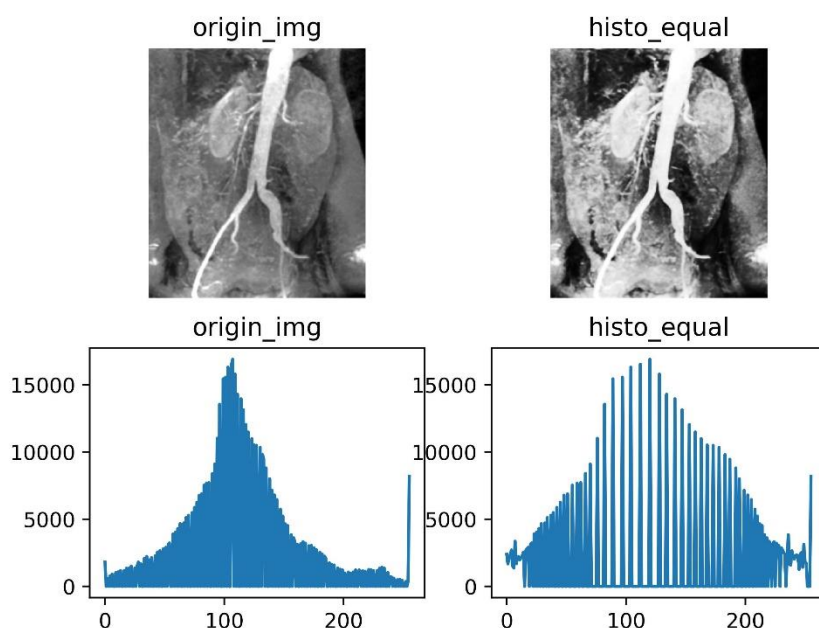


(這張比較表在./comparisons/
1_comparison_gamma_trans.jpg)

Histogram Equalization (課本 p134):

檔名: 1_histogram_equalization.jpg

直接 call `cv2.equalizeHist(origin_img)`。這張更能突現血管與腎本身的輪廓。



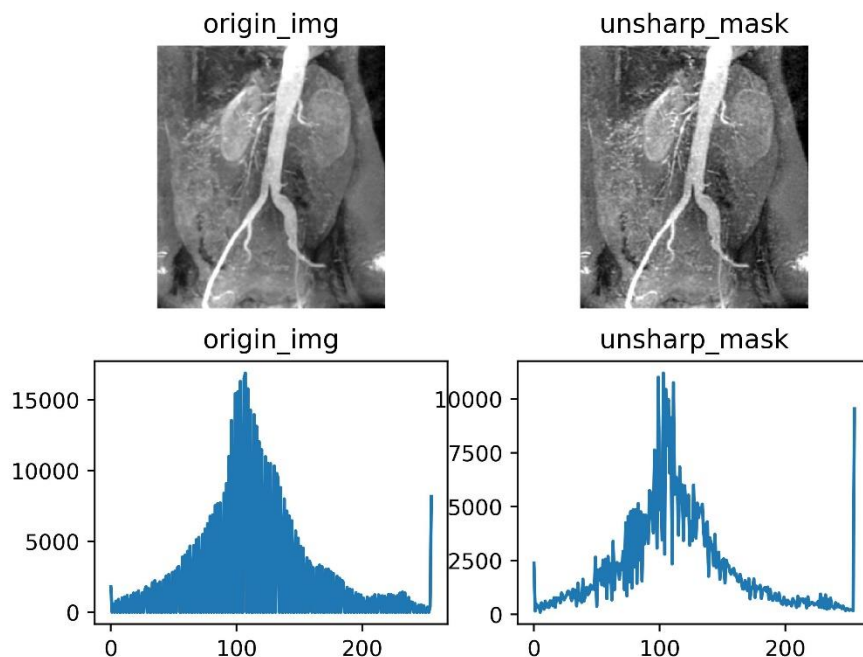
(這張比較表在./comparisons/
1_comparison_histo_equal.jpg)

Unsharp Masking (課本 p182):

檔名: 1_unsharp_masking.jpg

公式是採用 p182 的 equation 3-56 $g(x,y) = f(x,y) + kg_{mask}(x,y)$ ，這裡設 $k=1$ 。

$g_{mask}(x,y)$ 是用 kernel size=13 的 median blur。整體有被 sharpen。



(這張比較表在./comparisons/
1_comparison_unsharp_mask.jpg)

Additional Method 1 (課本 p193 改良):

檔名: 1_method1.jpg

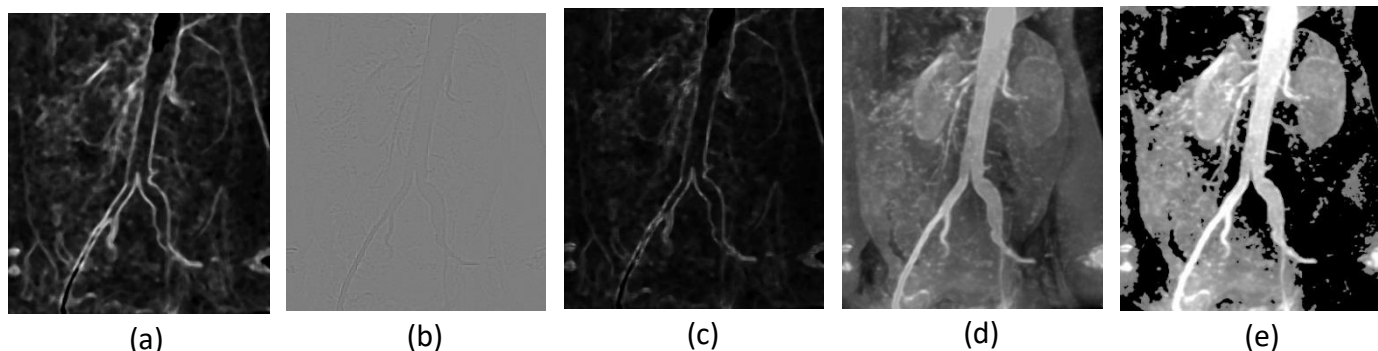
Step1: 做 Sobel (kernel size=3)後再做 median blur (kernel size=13)，如圖(a)。

Step2: 做 Laplacian (kernel size=3)，如圖(b)。

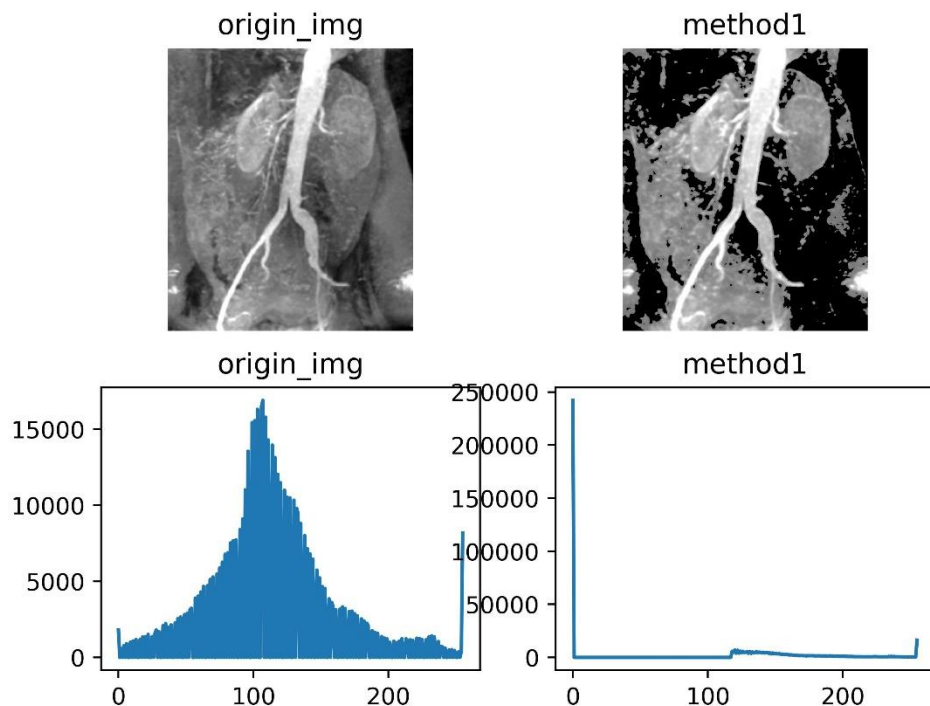
Step3: $enhance = step1 * step2$ ，如圖(c)。

Step4: $g(x,y) = origin_{img} + k * enhance$ 8，如圖(d)。

Step5: $g(x,y) = \begin{cases} 0, & \text{if } input(x,y) < 118 \\ 255, & \text{if } input(x,y) > 255 \\ input(x,y), & \text{otherwise} \end{cases}$ ，如圖(e)。



從結果(e)可看出，兩顆腎完整沒被遮擋，血管的輪廓有被加強，同時也完全保留了血管中的灰色 details，腎也有明顯被 enhanced。



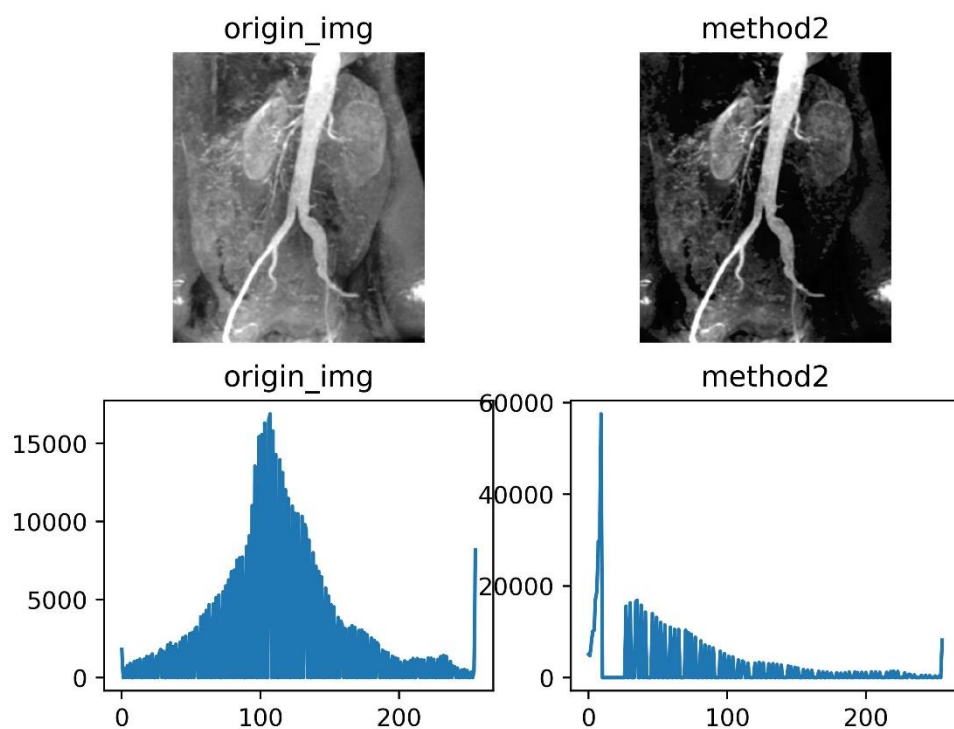
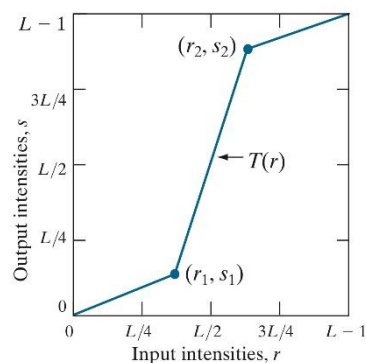
(這張比較表在./comparisons/
1_comparison_method1.jpg)

Additional Method 2 (課本 p129 改良):

檔名: 1_method2.jpg

採用課本 p129 的圖(如右)，因為我現在需要做 enhance，理論上要把 low intensity 的部分做 compression，high intensity 的部分作 stretching，所以我令 $(r_1, s_1) = (100, 10)$, $(r_2, s_2) = (130, 70)$ 。

輸出結果可以很明顯看出來原本暗的地方看起來更暗了，整體對比程度上升，血管內的灰色部分也更明顯，甚至連腎裡面的細節也更加清晰。我認為第 1 題五張 output 中，表現最好的是這張 method2。



(這張比較表
在./comparisons/
1_comparison_method2.jpg)

Problem2 Explain why the product of (b) and (e) in Fig. 3.58 serves as a mask image?

下圖取自上課 ppt 的圖，相乘可以取想要關注的區塊，以下面的例子來說，我只想看(a)裡的(b)範圍，把兩圖相乘就能得到(c)。

ROI (region of interest)的原理:

把(a)(b)做 pixel-wise 相乘，(b)裡 intensity=0 的地方和(a)相乘得到的結果也會是 0，可以把不想關注的地方變黑。



a b c

FIGURE 2.34 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

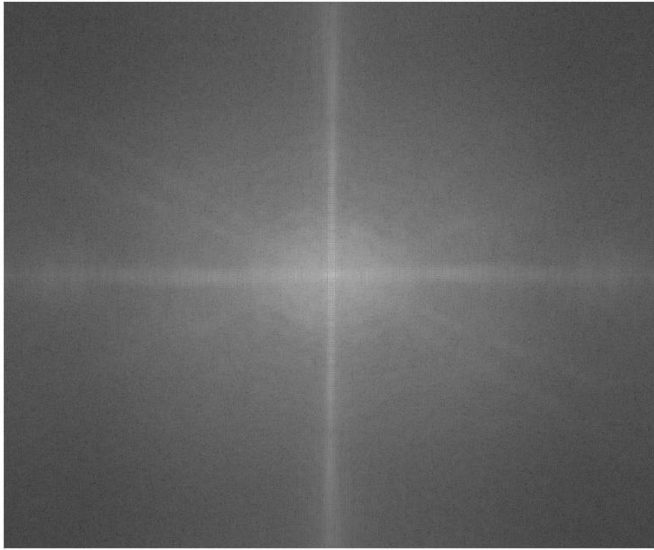
而有 edge 的地方會讓 sobel 有 high intensity，所以 sobel result 類似於上圖(b)，laplacian result 類似於上圖的(a)，最後做相乘就能得到關注於 edge 的 mask image。

Problem 3(a) Show the Fourier spectrum of the test image "keyboard."

檔名: 3a_fourier_spectrum.jpg

Output result:

(a) Fourier Spectrum



Problem 3(b) Enforce odd symmetry on the kernel. Show the kernel.

Enforce odd symmetry 後的 kernel 如下

$$\text{odd_symm_vertical_sobel_kernel} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 2 & 0 & -2 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \text{ 以}(2,2)\text{為 center (從 } 0 \text{ 開始數)}。$$

證明符合 odd symmetry:

根據課本 p244 的 equation 4-83 $w_o(x, y) = -w_o(M - x, N - y)$ ，這裡 $M=N=4$ 。

$$w_o(1,1) = -w_o(3,3)$$

$$w_o(1,2) = -w_o(3,2)$$

$$w_o(1,3) = -w_o(3,1).....$$

以此類推，可以發現每項皆符合前面定義的公式，所以證明它符合 odd symmetry。

補充 1: 沒有 enforce odd symmetry 前 kernel 如下

$$\text{vertical_sobel_kernel} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

補充 2: 這裡的 3*3 kernel 會和 `cv2.Sobel(origin_img, cv2.CV_64F, 1, 0, ksize=3)` 的 kernel 都差一個負號，因為 opencv 的 Sobel 實作是 correlation，而我們做的是 convolution。

(<https://stackoverflow.com/questions/42037075/does-opencv-sobel-operator-compute-correlation>)

Problem 3(c)(d) 我把 c d 兩小題 result 放在同一張圖

(c) Show the result of frequency-domain filtering of the test image using the vertical Sobel kernel.

(d) Compare your result in (c) with the result of space-domain filtering.

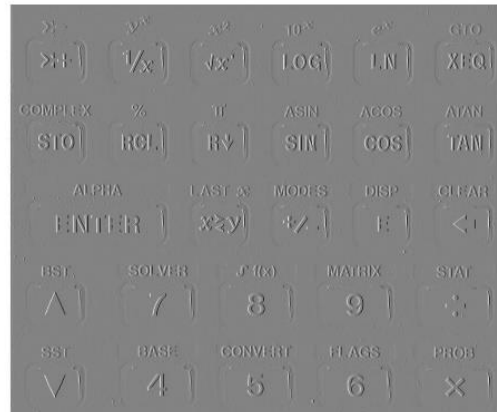
檔名: 3cd_odd_symm_result.jpg

Output result:

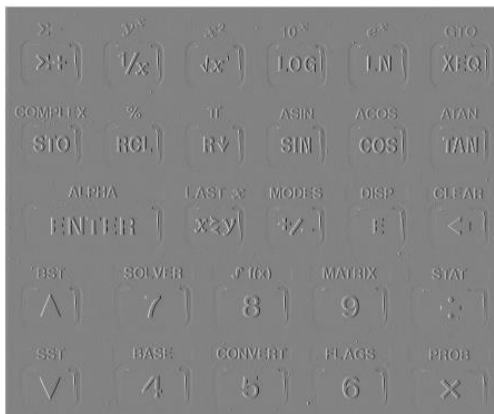
(c)(d) Enforce Odd Symmetry

Padded origin

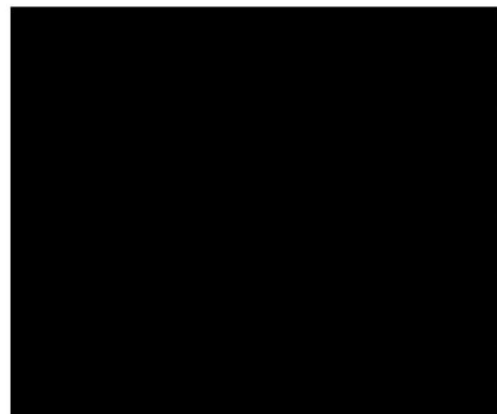
freq domain vertical sobel



spatial domain vertical sobel



Diff



1. 左上是經過 pad 後的原圖，padding 大小取自課本 p256 equation 4-100 & 4-101，這裡取

$$\begin{cases} P = A + C = 1360 + 4 = 1364 \\ Q = B + D = 1134 + 4 = 1138 \end{cases}。$$

2. 右上是在 frequency domain 做 Sobel，取的 kernel 為

$$\text{odd_symm_vertical_sobel_kernel} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 2 & 0 & -2 \\ 0 & 1 & 0 & -1 \end{bmatrix}，實作流程如課本 p272 所述。$$

3. 左下是在 spatial domain 做 Sobel，kernel 和上面一樣
4. 右下是 $\text{Diff} = \text{step2} - \text{step3}$ ，可以看到是全黑的圖，代表 $\text{diff}=0$ ，在兩個 domain 做出來的 result 會相同。

Problem 3(e)

Show the result of frequency-domain filtering without enforcing odd symmetry on the kernel.

檔名: 3e_wo_odd_symm_reasult.jpg

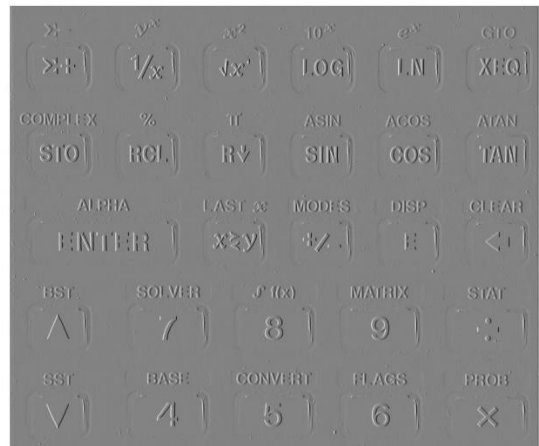
Output result:

(e) Without Odd Symmetry

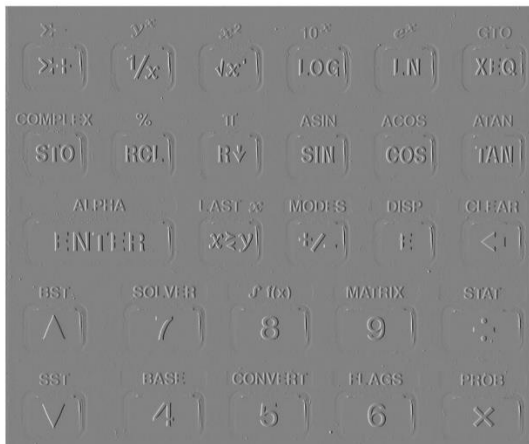
Padded origin



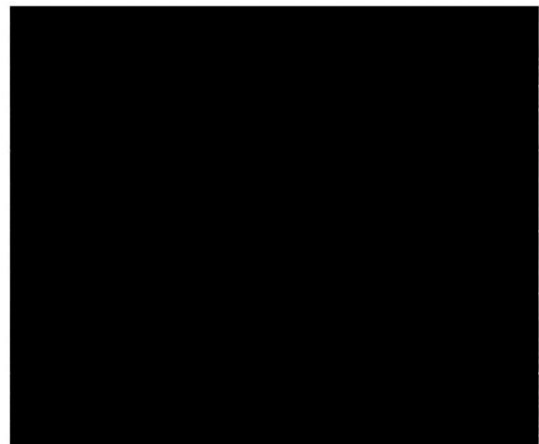
freq domain vertical sobel



spatial domain vertical sobel



Diff



1. 左上是經過 pad 後的原圖，padding 大小取自課本 p256 equation 4-100，這裡取

$$\begin{cases} P = A + C - 1 = 1360 + 3 - 1 = 1362 \\ Q = B + D - 1 = 1134 + 3 - 1 = 1136 \end{cases}.$$

2. 右上是在 frequency domain 做 Sobel，取的 kernel 為

$$vertical_sobel_kernel = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \text{ 實作流程如課本 p272 所述。}$$

3. 左下是在 spatial domain 做 Sobel，kernel 和上面一樣。
4. 右下是 $Diff = step2 - step3$ ，可以看到是全黑的圖，代表 $diff=0$ ，在兩個 domain 做出來的 result 會相同。

Problem 4

Similar to Problem 1 except that image enhancement is to be performed in the frequency domain, you are asked to try at least five methods to enhance the image shown on the right. Among the five methods, two can be your own creation.

Ideal High Pass Filtering (課本 p284):

檔名: 4_ideal_HPF_filtering.jpg

採用公式 $H(u, v) = \begin{cases} 1 + shift, & \text{if } D(u, v) > D_0 \\ 0 + shift, & \text{if } D(u, v) \leq D_0 \end{cases}$, 這裡設 $D_0=20$, $shift=1$ 。

Gaussian High Pass Filtering (課本 p287):

檔名: 4_gaussian_HPF_filtering.jpg

採用公式 $H(u, v) = \left(1 - e^{\frac{-D^2(u, v)}{2D_0^2}}\right) + shift$, 這裡設 $D_0=20$, $shift=1$ 。

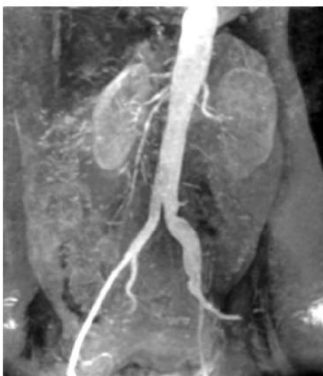
Butterworth High Pass Filtering (課本 p287):

檔名: 4_butterworth_HPF_filtering.jpg

採用公式 $H(u, v) = \left(\frac{1}{1 + [\frac{D_0}{D(u, v)}]^{2n}}\right) + shift$, 這裡設 $D_0=20$, $n=2$, $shift=1$ 。

從下面的 HPF 比較中可看出，三個 HPF 都有成功 enhance 圖片，其中 gaussian 和 butterworth 的 ringing 現象沒那麼嚴重。

Original img



ideal HPF



gaussian HPF

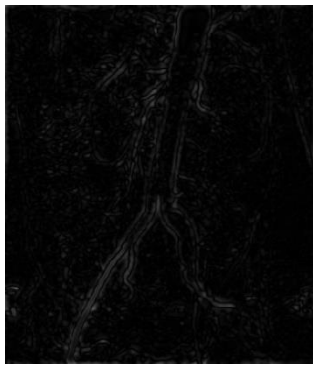


butterworth HPF



(這張比較表在./comparisons/
4_comparison_HPFs.jpg)

補充:

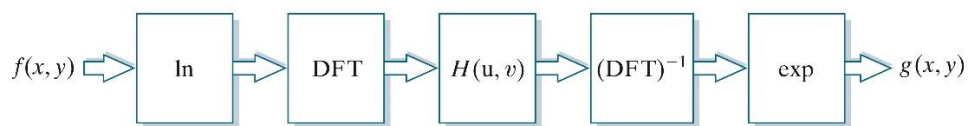


若是 HPF 沒有做 shift，則 output 結果會變得黑黑的。
左邊這張就是 butterworth HPF 最後沒加 shift 後的結果。

Homomorphic Filtering (課本 p293):

檔名: 4_homomorphic_filtering.jpg

根據 p295 圖 4.58 與 equation 4-147 實作



$$H(u, v) = (\gamma_H - \gamma_L) \left[1 - e^{\frac{-cD^2(u,v)}{D_0^2}} \right] + \gamma_L, \text{ 這裡設 } \gamma_L=0.5, \gamma_R=2, D_0=50, c=1。$$

整體的對比度變明顯，腎和主血管以外的地方變暗，同時也維持血管內的 detail，腎裡面的細節也更明顯(像是左腎的那個兩個暗點)。**我認為第 4 題的六張圖中，表現最好的是這張圖。**

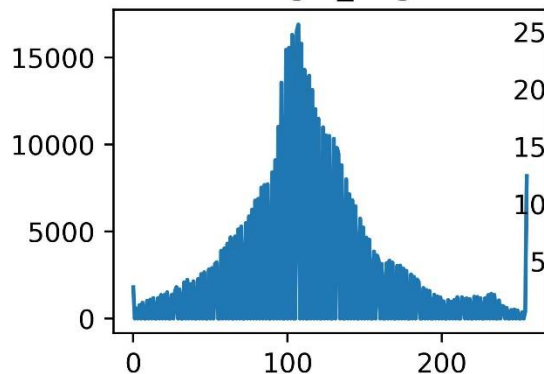
origin_img



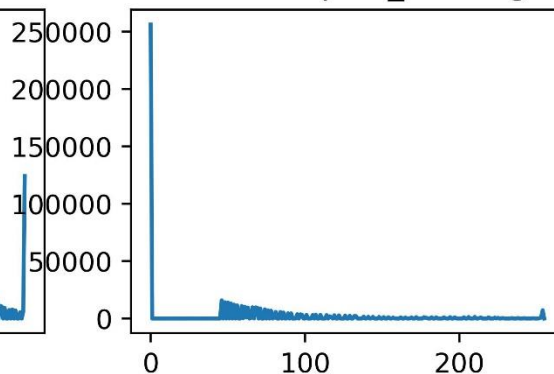
homomorphic_filtering



origin_img



homomorphic_filtering



(這張比較表
在./comparisons/
4_comparison_
homomorphic_
filtering.jpg)

Laplacian in Frequency domain (課本 p290):

檔名: 4_freq_laplacian_filtering.jpg

Step1: 先把 input image normalize 到 0~1 之間，因為 frequency domain 求出來的 $\nabla^2 f(x, y)$ 可能是 $\max f(x, y)$ 的好幾倍，最後做相加時會影響結果，所以要先 normalize 到可比較的範圍。

Step2: 做 DFT 得 $F(u, v)$ ，並根據課本 equation 4-124 算出 $H(u, v) = -4\pi^2 D^2(u, v)$ 。

Step3: $\nabla^2 f(x, y) = IDFT[H(u, v) * F(u, v)]$

Step4: $g(x, y) = f(x, y) + c\nabla^2 f(x, y)$ ，這裡取 $c = -0.45$

圖片整體有被 enhance，血管與腎的輪廓變很清晰，其他不重要的地方也有變暗。

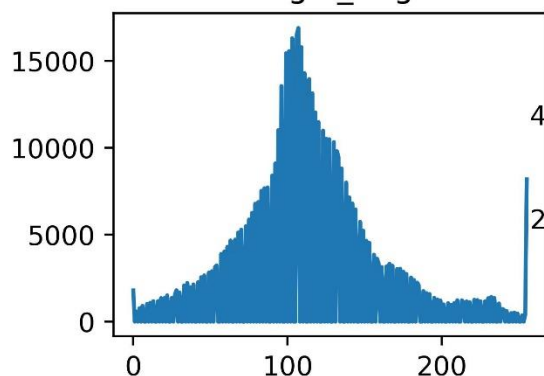
origin_img



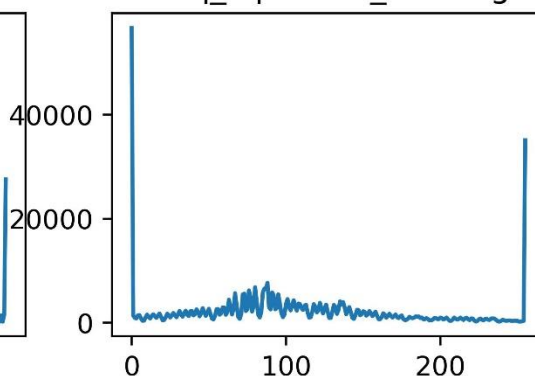
freq_laplacian_filtering



origin_img



freq_laplacian_filtering



(這張比較表
在./comparisons/
4_comparison_
freq_laplacian_
filtering.jpg)

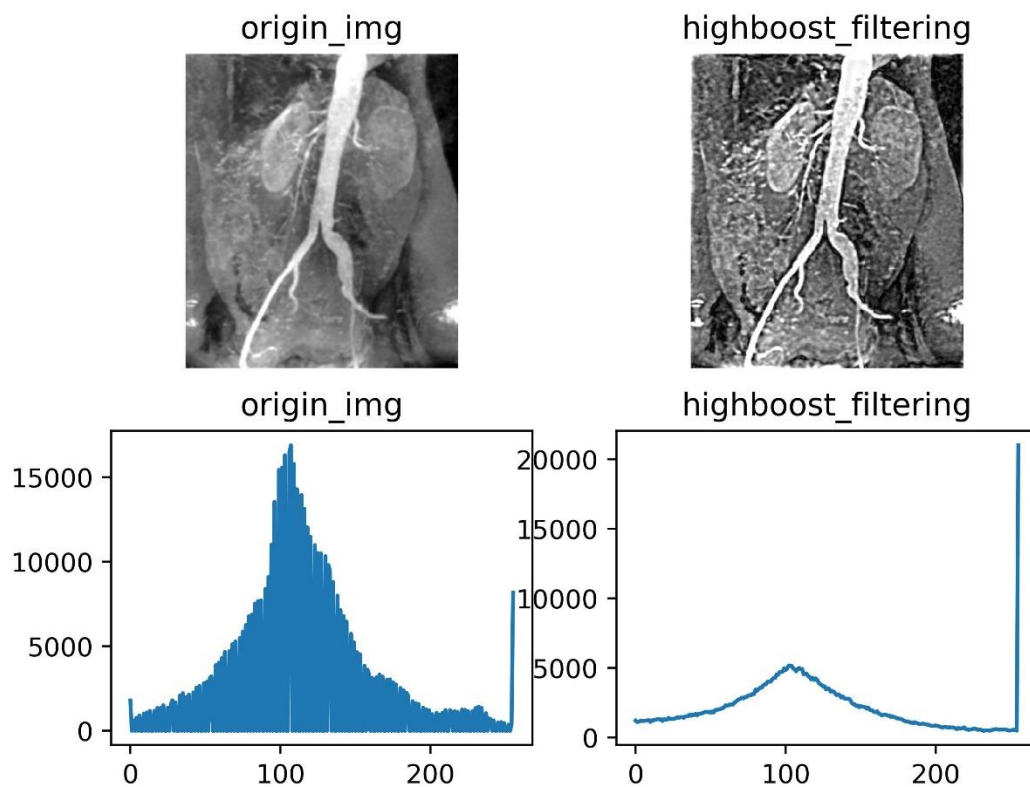
High Boost Filtering (課本 p291):

檔名: 4_highboost_filtering.jpg

使用課本 p291 equation 4-131 $g(x, y) = IDFT\{(1 + k[1 - H_{LP}(u, v)]) * F(u, v)\}$

這裡的 $H_{LP}(u, v)$ 是用 Butterworth Low Pass Filter 去實做 ($n=2, D_0=20$), $k=3$ 。

結果的 Enhancement 非常明顯，血管裡的灰色部分也都被 enhance 了，而且血管與腎的輪廓也變得更清楚。



(這張比較表
在./comparisons/
highboost_filtering.jpg)

Reference:

- [1] Digital Image Processing International Edition, 4th Edition by R. C. Gonzalez.
- [2] 老師上課的 ppt
- [3] Discussion with 楊冠彥同學

Problem1:

- [4] https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html
- [5] <https://www.idtools.com.au/unsharp-masking-with-python-and-opencv/>
- [6] <https://jason-chen-1992.weebly.com/home/-unsharp-masking>
- [7] <http://t.csdnimg.cn/PfwWh>
- [8] <http://t.csdnimg.cn/6yY4E>
- [9] <https://www.geeksforgeeks.org/opencv-python-program-analyze-image-using-histogram/>

Problem3:

- [10] <https://stackoverflow.com/questions/54990022/convolving-sobel-operator-in-x-direction-in-frequency-domain>
- [11] <https://stackoverflow.com/questions/77168452/how-to-select-an-appropriate-sobel-kernel-for-fft-2d>
- [12] <https://stackoverflow.com/questions/64743749/sobel-filter-giving-poor-results>
- [13] <https://stackoverflow.com/questions/25392314/manual-implementation-of-opencv-sobel-function>

Problem4:

- [14] <http://t.csdnimg.cn/8r5i7>
- [15] <http://t.csdnimg.cn/Z4hpe>
- [16] <https://stackoverflow.com/questions/30278229/high-pass-butterworth-filter-on-images-in-matlab>
- [17] <http://t.csdnimg.cn/601dl>
- [18] <http://t.csdnimg.cn/CVArx>
- [19] <http://t.csdnimg.cn/XWi3l>
- [20] <https://github.com/adenarayana/digital-image-processing/blob/main/Python%2009%20Frequency%20domain%20filter%20Laplacian%20Filter.py>
- [21] <http://t.csdnimg.cn/RUUEF>
- [22] <https://github.com/adenarayana/digital-image-processing/blob/main/Python%2012%20Unsharp%20Masking%20and%20Highboost%20Filtering%20in%20Frequency%20Domain.py>