

影像處理 LAB3

學號 姓名

4-1 Two-Dimensional Fast Fourier Transform

做法說明

1. 因為講義上 DFT 的公式，其 Time Complexity 是 $O(n^4)$ 為了避免運行過久，將圖片用 `imresize` 縮小成 128x128，來縮短處理時間。
2. 對原圖做 Zero Padding，把長、寬變成 2 倍，也就是 256x256。
3. 再把圖片上各 pixel 做 shift 處理，讓原點變成正中央，得到 `input_shift`，公式如下：

$$f_{shift}(x, y) = f(x, y) * (-1)^{x+y},$$
$$x = 0, 1, 2, 3, \dots, M-1; y = 0, 1, 2, 3, \dots, N-1$$

4. 將 `input_shift` 放入 `myDFT2` 做 DFT 處理，得到。
5. `myDFT2` 裡的實作：
 - (1) 把 `input` 從 2-D matrix 拉成長度為 $1 \times MN$ 的 row，稱做 `inputVector`
 - (2) 利用 4 層迴圈，來依序跑過每個 u, v, x, y
 - a. Loop 1 : $u = 0 : M-1$
 - b. Loop 2 : $v = 0 : N-1$
 - c. Loop 3 : $x = 0 : M-1$
 - d. Loop 4 : $y = 0 : N-1$
 - (3) 做一個長度為 $MN \times 1$ 的 column，稱做 `expMap`，用來儲存當前 $F(u, v)$ 所需的 exponential 的值

$$\text{expMap} = \begin{bmatrix} e^{z * (\frac{u*0}{M} + \frac{v*0}{N})} \\ e^{z * (\frac{u*0}{M} + \frac{v*1}{N})} \\ \vdots \\ e^{z * (\frac{u*0}{M} + \frac{v*(N-1)}{N})} \\ e^{z * (\frac{u*1}{M} + \frac{v*0}{N})} \\ \vdots \\ e^{z * (\frac{u*(M-1)}{M} + \frac{v*(N-1)}{N})} \end{bmatrix}, z = -j2\pi$$

在 Loop 2 時初始化，接著 Loop3、4 一個個計算並放入

- (4) 最後將 `inputVector` 和 `expMap` 相乘得到 $F(u, v)$ ，回到 Loop2 做 $F(u, v+1)$ ，或回到 Loop1(換下一行)做 $F(u+1, v)$

$$F(u, v) = \text{inputVector}_{1 \times MN} * \text{expMap}_{MN \times 1}$$

6. 對 DFT 後的圖做取絕對值得到 Spectrum 圖，但因為太過亮色，所以做 log 處理並進行縮放，讓範圍仍然維持在 0~1

7. 用 myGLPH 做出 GLPH filter，預設 $D_0=10$ 、 $(M,N)=\text{DFT 圖片的大小}$ ，做法如下：

(1) 取中心點

$$\text{midM} = \text{round}\left(\frac{M}{2}\right)$$

$$\text{midN} = \text{round}\left(\frac{N}{2}\right)$$

(2) 利用下面的公式，做出整個 filter

$$D(u, v) = e^{-\frac{(u-\text{midM})^2 + (v-\text{midN})^2}{2 \cdot D_0^2}}$$

8. 將 $F(u, v)$ 和 $\text{GLPH}(u, v)$ 做點對點的相乘，得到 G

$$G(u, v) = F(u, v) * \text{GLPH}(u, v)$$

$$u = 0, 1, 2 \dots M-1; v = 0, 1, 2 \dots N-1$$

9. 對 G 做 myIDFT2 並再次 shift，就可以得到經過 GLPH 的原圖(仍有 Zero padding)

10. myIDFT2 做法，基本上 myDFT2 一樣，只是反轉過來：

(1) 把 input 從 2-D matrix 拉成長度為 $1 \times MN$ 的 row，稱做 inputVector

(2) 利用 4 層迴圈，來依序跑過每個 x, y, u, v (和 myDFT2 順序不同!!)

a. Loop 1 : $x = 0 : M-1$

b. Loop 2 : $y = 0 : N-1$

c. Loop 3 : $u = 0 : M-1$

d. Loop 4 : $v = 0 : N-1$

(3) 做一個長度為 $MN \times 1$ 的 column，稱做 expMap，用來儲存當前 $f(x, y)$ 所需的 exponential 的值

$$\text{expMap} = \begin{bmatrix} e^{z * (\frac{x*0}{M} + \frac{y*0}{N})} \\ e^{z * (\frac{x*0}{M} + \frac{y*1}{N})} \\ \vdots \\ e^{z * (\frac{x*0}{M} + \frac{y*(N-1)}{N})} \\ e^{z * (\frac{x*1}{M} + \frac{y*0}{N})} \\ \vdots \\ e^{z * (\frac{x*(M-1)}{M} + \frac{y*(N-1)}{N})} \end{bmatrix}, z = j2\pi$$

在 Loop 2 時初始化，接著 Loop3、4 一個個計算並放入

(4) 最後將 inputVector 和 expMap 相乘、再除 MN 得到 $f(x, y)$ ，回到 Loop2 做 $f(x, y+1)$ ，或回到 Loop1(換下一行)做 $f(x+1, y)$

$$f(x, y) = \frac{1}{MN} (\text{inputVector}_{1 \times MN} * \text{expMap}_{MN \times 1})$$

11. 將 padding 部分去除，得到經過 GLPH 的原圖

結果圖片

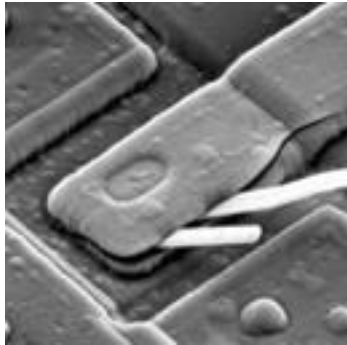
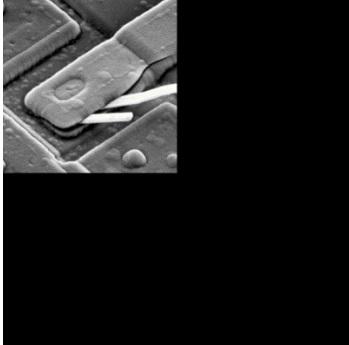
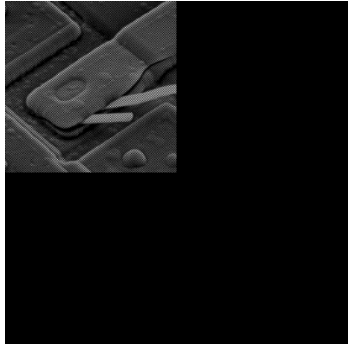
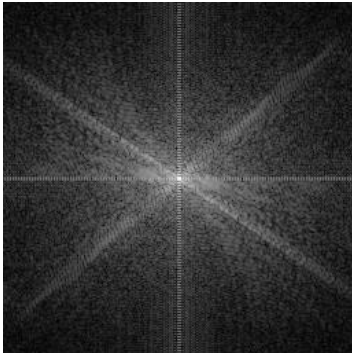
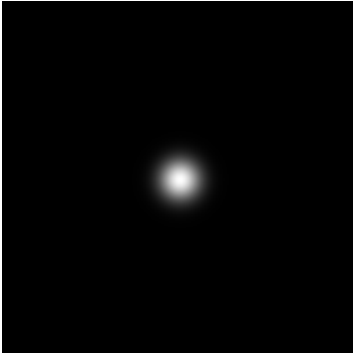
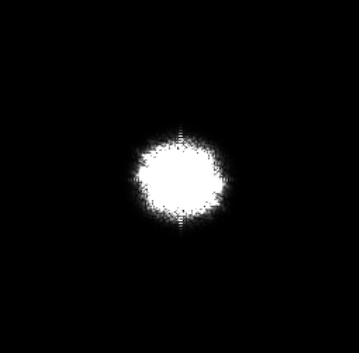


		
圖(a) 輸入的圖片	圖(b) Zero Padding	圖(c) Shift
		
圖(d) DFT Spectrum	圖(e) GLPH($D_0=10$)	圖(f) G 的 Spectrum
		
圖(g) G 經過 IDFT 和 shift	圖(h) 去除 Padding 後	

圖 1 Proj04_01

分析以及討論

1. DFT/FFT 運行時間

由講義上提供的式子，可以看出 DFT 是 $O(n^4)$ ，因此若對原本指定圖片(約 1000×1000)做處理，是非常耗時間。

而 FFT 可以把 Time complexity 降至 $O(n^2(\log n)^2)$ ，處理的速度會快上許多。

2. DFT 優化

先把本來是一個值乘一個值的算式，轉換成矩陣的模式，利用 matlab 的強大的矩陣運算來達到加速的效果。

3. 圖片解析度和 Spectrum 關係

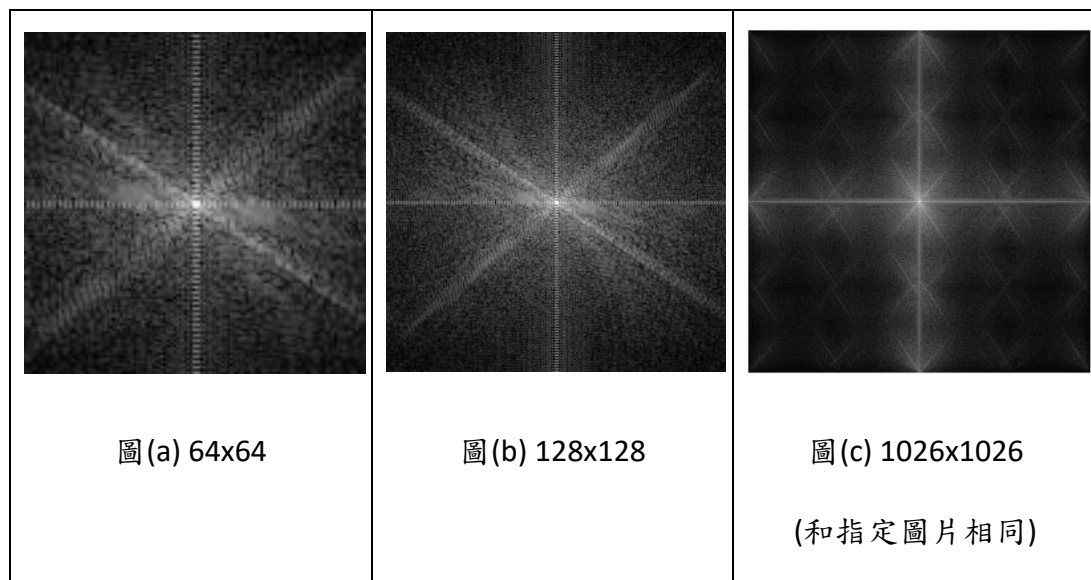


圖 2 圖片解析度和 Spectrum 關係

可以看出當解析度越低，Spectrum 圖的解析度也會隨著降低。

4. Padding 有無和 DFT 關係

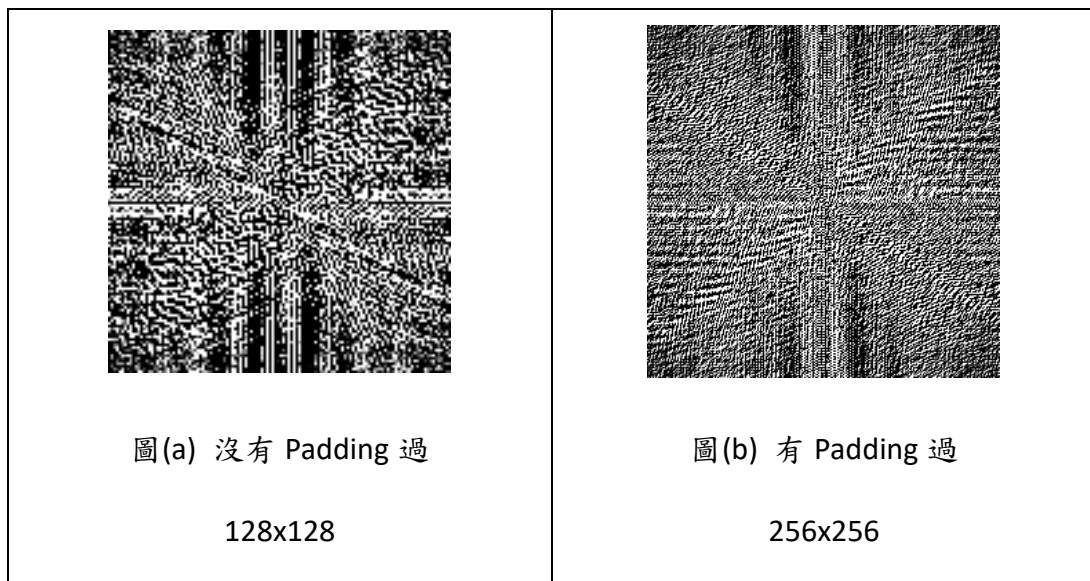


圖 3 Padding 有無和 DFT 關係

可以看出經過 Zero Padding 後得到的 DFT，會更加對稱而且密集，但不曉得有沒有其他意義。

4-2 Fourier Spectrum and Average Value

結果圖片



圖 4 Proj04_02

分析以及討論

1. 原圖算平均和 DFT 取中心點，兩者得到的值是否有差異？

沒有，兩邊算出來的值是一樣的，這也可以從公式裡推算出來。

2. 降低圖片解析度，是否影響平均?

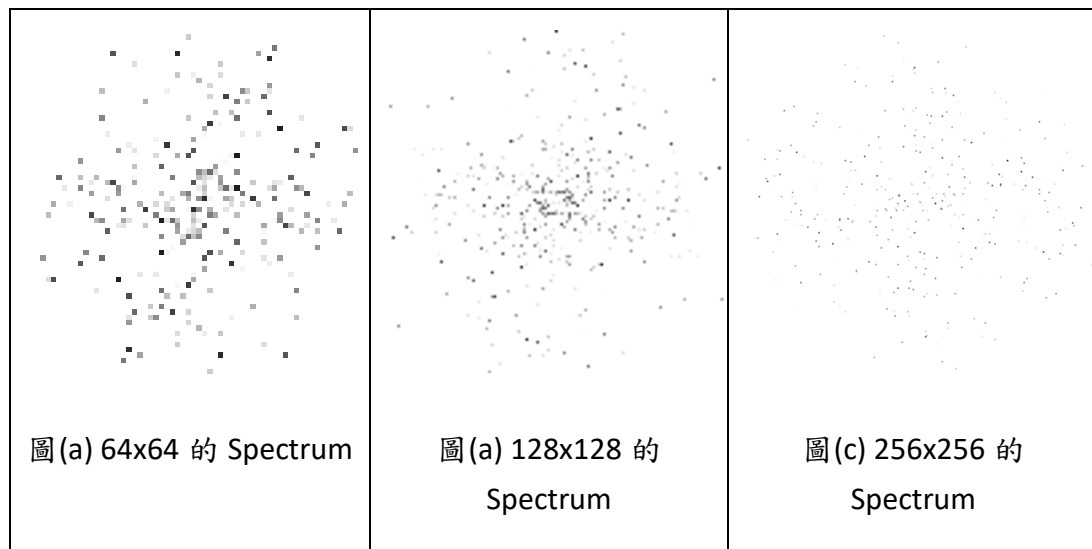


圖 5 圖片解析度和 Spectrum

上面三張圖是用不同解析度所得到的 Spectrum 圖，雖然每張圖都不相同，其中心的值都相同，也代表平均一樣。而用 DFT 前的圖算平均，也仍然相同，都是 0.813。

解析度並不會影響平均值

4-3 Lowpass Filtering

結果圖片

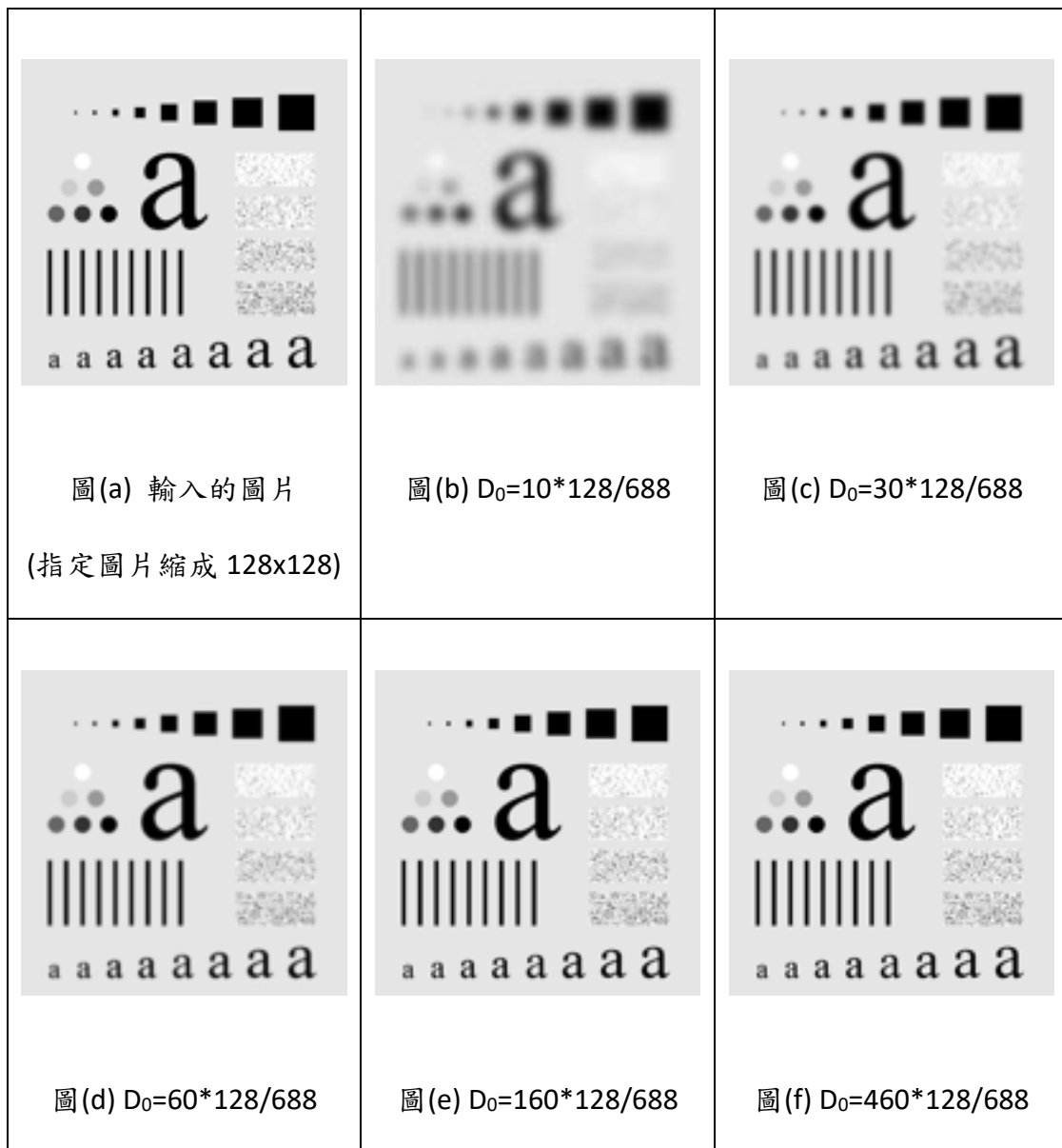


圖 6 Proj04_03

註. 因為圖片有縮小，所以 D_0 也相對應的縮小

分析以及討論

1. D_0 和 GLPH filter 的效果關係

若是 D_0 越大，代表能穿過的頻率越多，所以和原圖就越相似；反之， D_0 越小，代表能穿過的頻率越少，圖片會越模糊。

4-4 Highpass Filtering

結果圖片

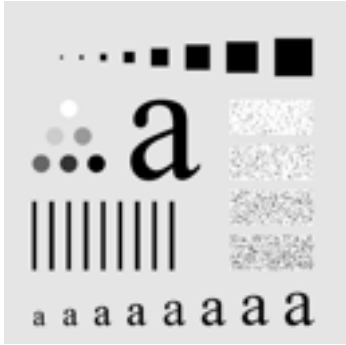

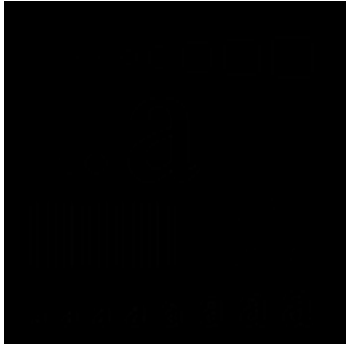
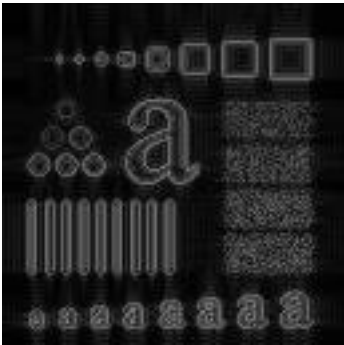

 <p>圖(a) 輸入的圖片 (指定圖片縮成 128x128)</p>	 <p>圖(b) $D_0=60*128/688$</p>	 <p>圖(c) $D_0=160*128/688$</p>
	 <p>圖(d) $D_0=60*128/688$ 經過亮化處理</p>	 <p>圖(e) $D_0=160*128/688$ 經過亮化處理</p>

圖 7 Proj04_04

註 1. 因為圖片有縮小，所以 D_0 也相對應的縮小

註 2. 因為原圖過暗，因此進行 powerlawTransform

分析以及討論

1. D_0 和 GHPH filter 的效果關係

和 GLPH 相反，若是 D_0 越大，代表能穿過的頻率越少，所以和邊界加強效果會變大；反之， D_0 越小，代表能穿過的頻率越多，圖片邊界會越模糊。

4-5 Highpass Filtering Combined with Thresholding

結果圖片


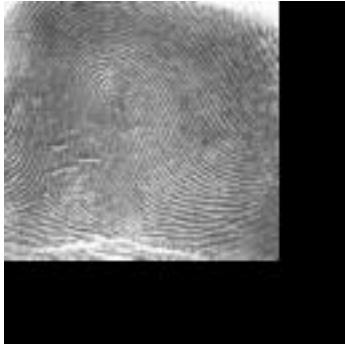


 <p>圖(a) 輸入的圖片 (指定圖片縮小 10 倍)</p>	 <p>圖(b) Zero Padding (Padding 成 128x128)</p>
 <p>圖(c) 經過 GHPH 後得到的 ($D_0=20$)</p>	 <p>圖(d) 經過 threshold 區分後 (threshold=0)</p>

圖 8 Proj04_05

分析以及討論

1. 有無 threshold 的差異

有了 threshold，就可以達成類日 ideal filter 的效果，但又不回產生類似波紋的副作用出來。

2. D_0 是否影響結果



	
$D_0=20$	$D_0=100$

圖 9 不同 D_0 的結果

差異並沒有很大，推測解析度才是讓效果變差的原因

3. 解析度和結果的關係

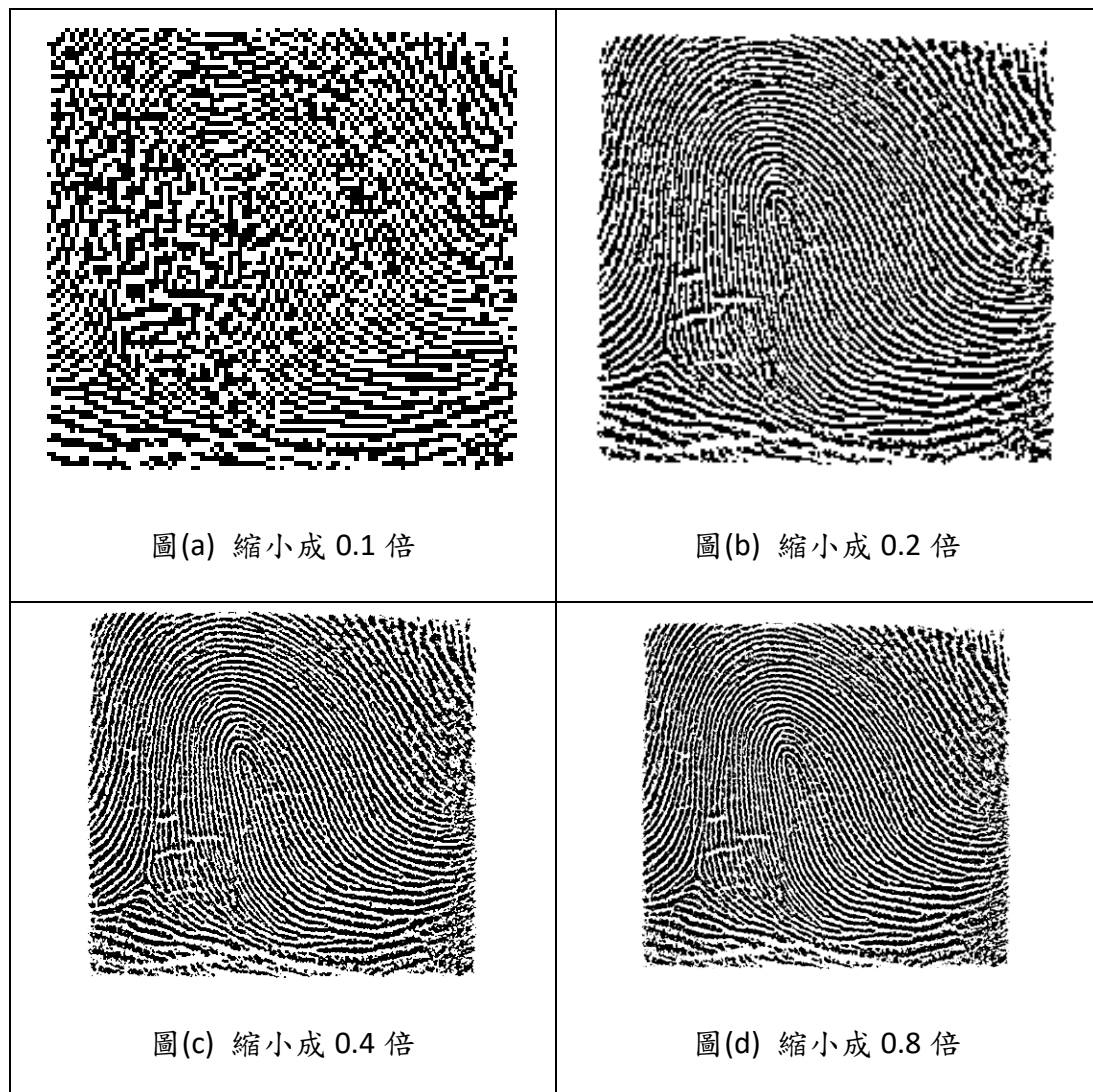


圖 10 不同解析度的結果

4. 放進 myDFT2/myIDFT2 之前，input 形狀影響

這次圖片為長方形，和前面幾個是正方形的情況不同。而我的 myDFT2/myIDFT2 裡的 input 必須是正方形才會運作正常，我認為是取中心點的方式所產生的錯誤。

因此，在使用 myDFT2/myIDFT2，我先將圖片 Zero Padding 成正方形來避免錯誤。