

# Introduction to Image Processing HW4

408420083 電機三 B 陳昀顯

## HW4.1 Image interpolation through spatial filtering

### · 實驗目的

自行定義出 `imenlagre(image,filt)` 來完成圖片放大兩倍的效果，其過程需使用 `zero interleaving` 放大圖片並使用兩種 `spatial filtering` – Bilinear、Bicubic，最後使用 `imresize` 縮小成原尺寸形狀，和原圖比較計算出 PSNR。

### · 實驗原理

這次使用到 `zero interleaving` 的技術將圖片放大兩倍，如圖 1 的結果：

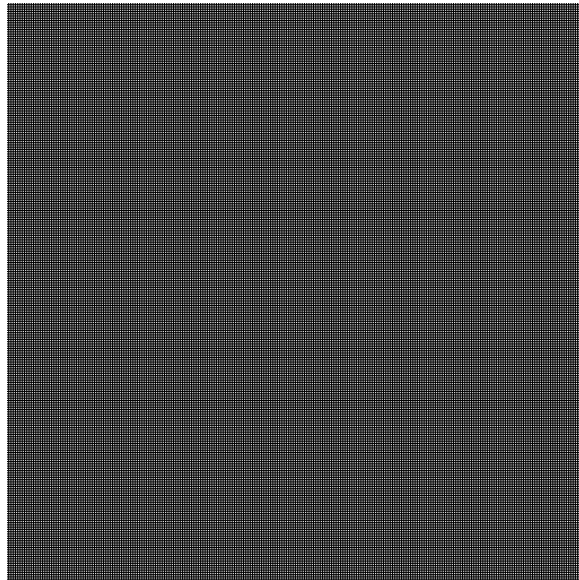


圖 1、經過 zero interleaving 處理後的圖片

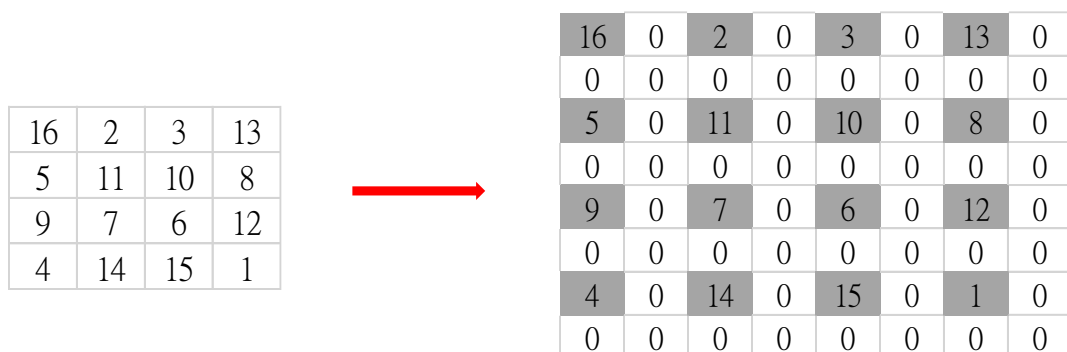


圖 2、zero interleaving 內部結果(放大兩倍後)

若我們細看內部的結構可以發現在奇數點上的 pixel 才有值其餘為 0，套用 `imfilter`(預設為 `Padding 0` 補邊緣處)之功能帶入 `bilinear`、`bicubic spatial filtering`：

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \frac{1}{64} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

bilinear                      bicubic

圖 3、濾波器內部設計

以 Bilinear filter 的效果來說，它的結果呈現如下圖 4:

0	0	0	0	0	0	0	0	0	0	16	9	2	2.5	3	8	13	6.5
0	16	0	2	0	3	0	13	0	0	10.5	8.5	6.5	6.5	6.5	8.5	10.5	5.25
0	0	0	0	0	0	0	0	0	0	5	8	11	10.5	10	9	8	4
0	5	0	11	0	10	0	8	0	0	7	8	9	8.5	0	9	10	5
0	0	0	0	0	0	0	0	0	0	9	8	7	6.5	6	9	12	6
0	9	0	7	0	6	0	12	0	0	6.5	8.5	10.5	10.5	10.5	8.5	6.5	3.25
0	0	0	0	0	0	0	0	0	0	4	9	14	14.5	15	8	1	0.5
0	4	0	14	0	15	0	1	0	0	2	4	7	7.25	0	4	0.5	0.25
0	0	0	0	0	0	0	0	0	0								

圖 4、濾波器前 vs 濾波後之結果

首先觀察原座標點可得知其值沒有改變，由於鄰近的 8 個座標點都是 0，經過計算權重後 x4、/4 仍為 1，其餘點也是如此。以綠色框框的點來說，其權重值為 2 跟 3 皆 x2 後再/4，符合能量守恆之原則。

## HW4.2 Image interpolation in frequency domain

### · 實驗目的

自行定義出 `imenlagre_fre(image, filt)` 來完成圖片放大兩倍的效果，其過程需使用 zero interleaving 放大圖片並使用兩種 spatial filtering – Bilinear、Bicubic，整個過程需在頻域轉換後用反傅立葉求出結果，最後使用 `imresize` 縮小成原尺寸形狀，和原圖比較計算出 PSNR。

### · 實驗原理

這次實驗使用了 Convolution 理論作為頻域轉換的工具，將影像 M 與空間濾波器 S 做捲積運算，其方法是將 S 的 Mask 逐次放在 M 的像素上，計算對應的 M 灰階值與 S 元素的乘積，兩者結果相加後得到新像素值。我們簡化式子後可得：

$$F(M * S) = F(M)F(S')$$

其中 S' 為補 0 後與 M 同樣大小，使用到 Matlab 中的 FFT 快速傅立葉轉換能使每一列相乘次數比直接運算還要來的快。

· 成果展示

表一、使用 Bilinear 差值法縮放 cameraman.png

圖 1、imenlarge x2 cameraman.png(左:Time Domain 右:Freq. Domain)

圖 2、imresize x0.5 cameraman.png(左:Time Domain 右:Freq. Domain)



	PSNR_1	28.4104	1×1	double
	PSNR_2	28.2845	1×1	double

圖 3、比較 PSNR between the original image and the rescaled back image  
(PSNR1:Time Domain PSNR2:Freq. Domain)

由上圖結果來看，使用 Bilinear filter 去插值空洞對於在時域/頻域中之圖片，兩者的表現是一致的。

表二、使用 Bicubic 差值法縮放 cameraman.png



圖 1、imenlarge x2 cameraman.png(左:Time Domain 右:Freq. Domain)



圖 2、imresize x0.5 cameraman.png(左:Time Domain 右:Freq. Domain)



	PSNR_1	28.0385	1×1	double
	PSNR_2	22.2668	1×1	double

圖 3、比較 PSNR between the original image and the rescaled back image  
(PSNR1:Time Domain PSNR2:Freq. Domain)

觀察經頻域轉換後的圖片，可看到上面多了一條黑色的雜訊，且顏色與最底層區域吻合，不太確定是  $F(S') \times F(M)$  相乘時把整個 pixel 往下位移 x 格，多餘出來的部分承接到最上層部分，使 PSNR 與時域轉換相比低了不少。



表三、使用 Bilinear 差值法縮放 Galaxy.png

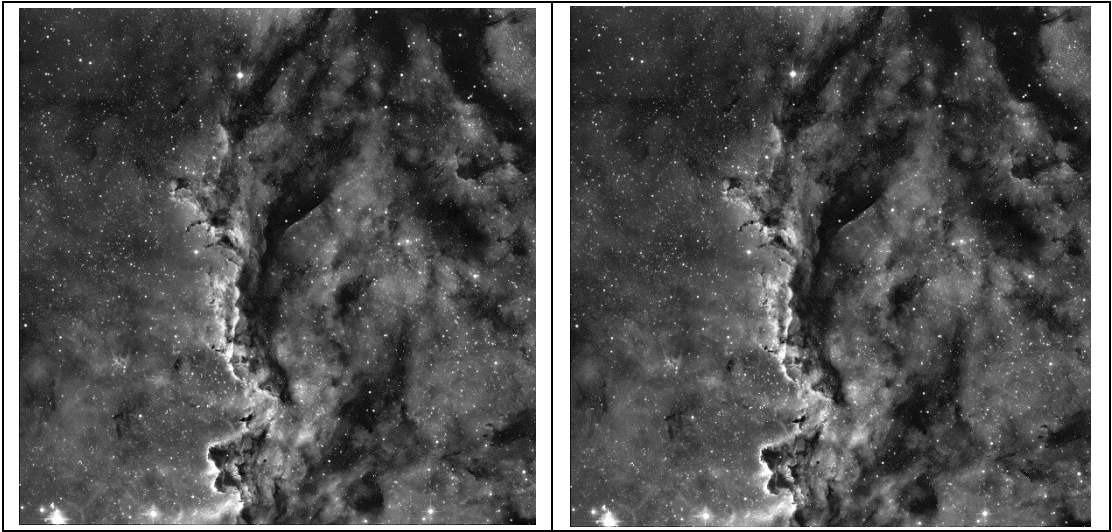


圖 1、imenlarge x2 Galaxy.png (左:Time Domain 右:Freq. Domain)

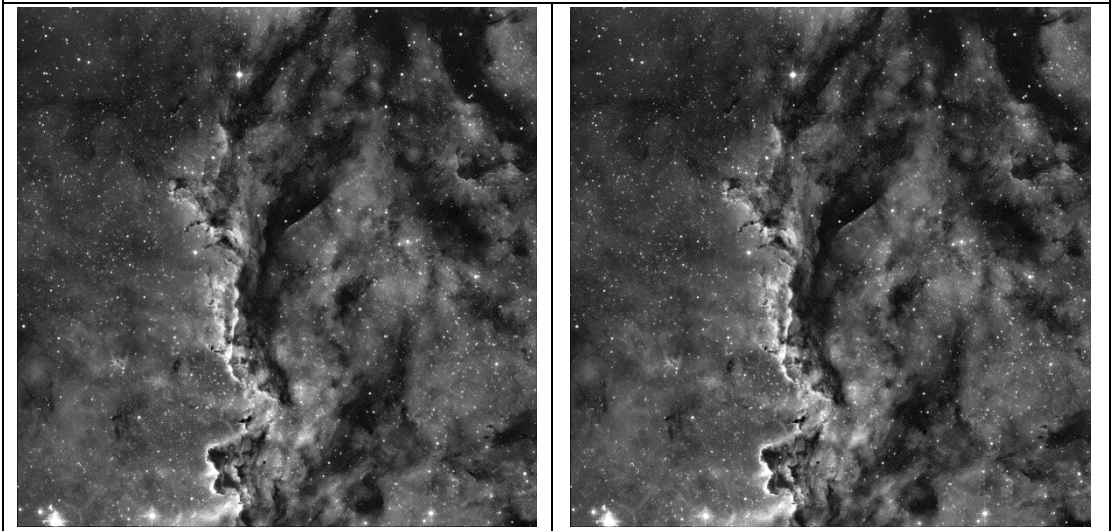


圖 2、imresize x0.5 Galaxy.png (左:Time Domain 右:Freq. Domain)



	PSNR_1	31.6373	1×1	double
	PSNR_2	31.6484	1×1	double

圖 3、比較 PSNR between the original image and the rescaled back image  
(PSNR1:Time Domain PSNR2:Freq. Domain)

由上圖結果來看，使用 Bilinear filter 去插值空洞對於在時域/頻域中之圖片，兩者的表現是一致的。

表四、使用 Bicubic 差值法縮放 Galaxy.png

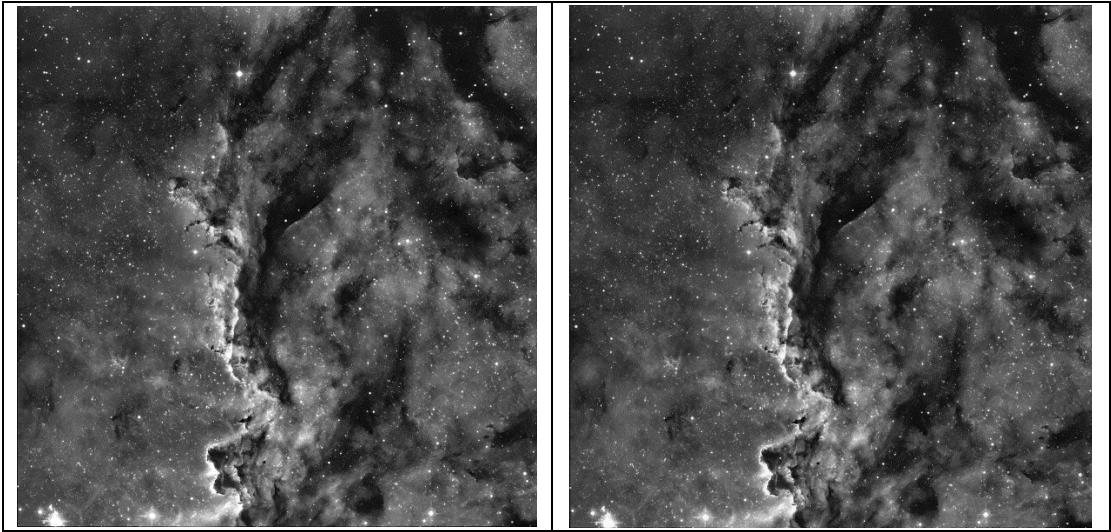


圖 1、imEnlarge x2 Galaxy.png (左:Time Domain 右:Freq. Domain)

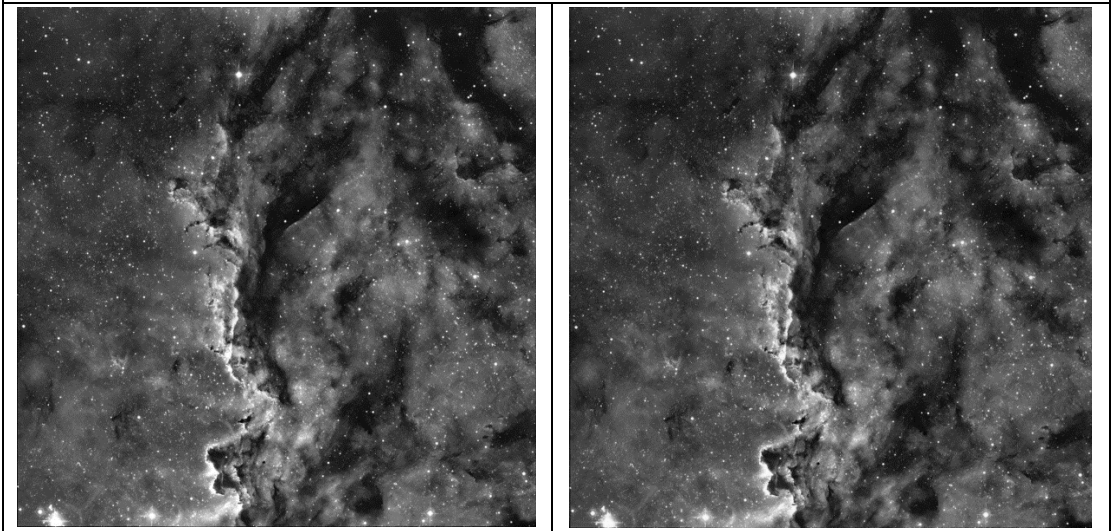


圖 2、imresize x0.5 Galaxy.png (左:Time Domain 右:Freq. Domain)



	PSNR_1	31.5436	1×1	double
	PSNR_2	25.8449	1×1	double

圖 3、比較 PSNR between the original image and the rescaled back image  
(PSNR1:Time Domain PSNR2:Freq. Domain)

觀察經頻域轉換後的圖片，可看到上面多了一條黑色的雜訊，且顏色與最底層區域吻合，不太確定是  $F(S') \cdot F(M)$  相乘時把整個 pixel 往下位移 x 格，多餘出來的部分承接到最上層部分，使 PSNR 與時域轉換相比低了不少。

### complex\_texture 細節比較 – 放大兩倍時



圖 1、complex\_texture (Bilinear)



圖 2、complex\_texture (Bicubic)

比較上圖的中間線條處，Bicubic 在圖片放大時的效果遠比 Bilinear 的還要好一些，尤其是銀河邊緣處，線條的呈現感覺很不錯。故當圖片要進行放大動作時，Bicubic filter 是唯一首選的濾波器。在運算中，Bilinear 的平滑作用可能會使圖像的細節產生退化，在進行放大處理時，這種影響更為明顯。在其他應用中，Bilinear 的斜率不連續性會產生不希望的結果。Bicubic 不僅考慮到周圍四個直接相鄰像素點灰階值的影響，也考慮到灰階值的變化率，產生比 Bilinear 更為平滑的邊緣，計算精度很高，處理後的圖像像質損失最少。



complex\_texture 細節比較 – 經放大再縮小 0.5 倍時



圖 1、complex\_texture (Bilinear)



圖 2、complex\_texture (Bicubic)

經過比對後可發現兩者似乎沒什麼太大的相異處，考慮到運算的效率，在圖片縮小時會選擇 **Bilinear filter** 作為首選。故在進行圖像縮放處理時，應根據實際情況做出選擇，既要考慮時間方面的可行性，又要對變換後圖像質量進行考慮，才能達到我們想要的結果。