

SVD 壓縮照片技術結合超解析度插值方法的探討

林春峰

國立勤益科技大學資訊工程系

411030 臺中市太平區中山路二段 57 號

4a917016@gm.student.ncut.edu.tw

摘要

壓縮照片技術可以幫助照片輕量化，超解析度方法則能夠提昇照片的解析度大小，本方法希望將這二者的領域結合起來，也就是說既想要壓縮照片的內容，也想要同時放大照片的解析度品質，在壓縮照片時，以不要太過破壞照片內容細節的前提下，也能保留原始照片中的良好輪廓，使照片輕量化，並提升照片的解析度，本方法在超解析度方面，將使用插值領域，並列舉出最近鄰插值、雙線性插值、雙三次插值這三種插值方法來依序對照片進行解析度提昇，並評測其效果，而壓縮照片的方法將選用 SVD 壓縮技術，來與超解析度插值方法結合，且使用 PSNR 為評價指標。

關鍵詞：SVD 壓縮照片技術，插值方法，超解析度

SVD Compression Technique Combined with the Super-Resolution Interpolation Method

CHUN FENG LIN

Department of Computer Science and Information Engineering, NATIONAL CHIN YI UNIVERSITY OF TECHNOLOGY

No.57, Sec. 2, Jhongshan Rd., Taiping Dist., Taichung City 411030, Taiwan, R.O.C.

4a917016@gm.student.ncut.edu.tw

ABSTRACT

The image compression technique and super-resolution method can enhance light-weight images and scale the image resolution, respectively. In this research, these two methods were combined to compress the image content and up sample the image resolution quality. When the image is compressed, the original image contour is retained under the “restrict” condition to prevent the image content from being excessively damaged. This process allows for enhancing of the image light weight and scaling of the image resolution. In this study, nearest-neighbor interpolation, bilinear interpolation, and bicubic interpolation were applied in the super-resolution domain to scale image resolution. An SVD technique was employed to compress the images and combine the three interpolation methods. This study used the PSNR to measure the processed image quality.

Key Words: SVD, interpolation method, super resolution



一、前言

壓縮照片技術在於幫助照片所攜帶的訊息輕量化，以利於照片在傳輸與存儲上的快速化，照片可以說是由一連串的像素值所組成的矩陣，壓縮照片技術就是在規範這些像素值矩陣可以按照某個方法對像素值做處理，輸出的像素矩陣具有一定的規律，以簡化像素的內容，但從整體上來看，處理後輸出的結果與原始照片毫無差異，用戶拍出的照片內容在處理後沒有被破壞，這在保存照片的目的上是至關重要的，而在壓縮的手法上，有些會將照片整體分區域地進行處理 [4]，在這些各自區域裡的像素值，則按照高低臨限二分化，有些將照片整體的顏色限定在某幾個顏色種類之內 [19]，照片中所有的顏色按照限定的這幾個顏色聚集，如此做可以壓縮照片的內容，而本方法要探討的是奇異值分解 (Singular Value Decomposition, SVD) 壓縮技術，以及 SVD 這項技術結合超解析度插值方法的探討，也就是說既想要對照片做解析度的提昇，也想要同時壓縮照片的內容，在不太過破壞照片品質的前提下，也能保留照片中良好的細節，使照片輕量化，以方便於儲存與傳輸，而超解析度的方法將會運用到插值領域，本方法列舉出最近鄰插值，雙線性插值，雙三次插值，這三種插值方法，來各自與 SVD 壓縮技術做結合，探討這三種方法對照片放大解析度後，再進行 SVD 壓縮流程，是否會對照片造成什麼影響，以及如此做之後，是否還能保留原始照片中良好的細節，接著再使用峰值訊噪比 (Peak Signal-to-Noise Ratio, PSNR) 來觀測輸出照片的品質數據，究竟會有何起伏變化，是本方法在做完 SVD 與超解析度插值的初步調查之後，將專注探討的。

而解析度技術已經發展到 4k 以上的水準，可是有些普通設備拍攝出來的解析度其實才 4k 不到，且 4k 設備成本高昂，已知影響照片品質的因素是設備硬體裡的感光器尺寸大小，4k 設備的感光器尺寸就是如此大，能拍出細節豐富的照片，普通設備的感光器尺寸就是如此地小，拍攝出的照片品質有限，如何藉由數位處理的方法，將普通設備拍出來的照片也可以變地像 4k 設備拍出來的照片那麼清晰，照片細節更豐富理想，解析度也更大，而現行的超解析度方法可能只非常專注於照片處理後的品質，追求更好的照片效果，使照片細節看起來更合理，或者在超解析度的同時也銳化照片，以用來免除照片中模糊的區域，更或是為了避免處理後的照片紋理可能會出現幻影，以及解決處理後的照片太過銳

利化，導致人眼觀看不舒適等等的問題，但這些超解析度技術可能沒有考慮到一個問題——處理後的照片容量，雖然容量這個問題看似無關緊要，但如果出現需要大量處理的批次照片，那可能輸出入就會動用高達幾 gb 以上的容量，本方法設想的插值結合 SVD 壓縮的方案從結果顯示，假設拿來實驗的某張照片，如圖 1 所示，也許他的原始容量是 265 kb，經由選擇高度保留照片細節的參數，來先進行單純的 SVD 壓縮，發現輸出的照片容量能被壓縮至 155 kb，輸出照片的容量與原始照片相比，容量減小了將近一半，而輸出照片的質量也能相當不錯，達到處理後的輸出照片與原始照片毫無差異的目標，輸出結果仍能極大地保留原始照片中細緻的紋理，這樣單純 SVD 壓縮就能有如此的效果，才有後續執行超解析度插值的探討。

究竟先對照片做 SVD 壓縮，再插值放大解析度，還是先超解析度插值後，再進行 SVD 壓縮，這個問題攸關於步驟的先後順序，是先壓縮再放大，還是先放大再壓縮，似乎也是值得思考的問題，會對結果的 PSNR 值等等的判別標準將有什麼影響，是必須要慎重考慮的。

而經由實驗發現，若先做 SVD 壓縮照片，再插值放大解析度的方法，得到了如下敘述的效果，假設對某張照片按照圖 2 的流程去執行，會得到 43.59 db (使用雙線性插值) 的 PSNR 值，通常來說，PSNR 可以代表照片的品質，尤其是針對照片接受某種處理之後，得到的輸出照片與原始照片



圖 1. 原始照片經由 SVD 壓縮後，處理後輸出的照片容量比原來更小

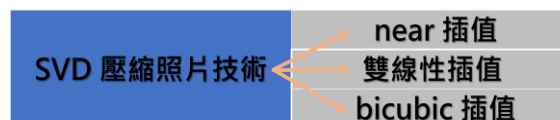


圖 2. 先對照片做 SVD 壓縮，再插值放大解析度



互相做像素之間的比較運算，輸出入這二張照片之間各自對映的區域像素先做相減，再平方運算，並倒數與 255 的平方相除，再做 log 的處理，

$$MSE = \frac{1}{w \cdot h} \sum_{i=1}^{w \cdot h} (original\ pixel - process\ pixel)^2 \quad (1)$$

$$PSNR = 10 \cdot \log_{10} \left[\frac{(255)^2}{MSE} \right] \quad (2)$$

算式(1)表示如果要得到 PSNR，請先求得 MSE (Mean Square Error)，MSE 意味著二張照片之間的像素值差距，original pixel 代表原始照片像素，process pixel 代表原始照片經過某種處理後的輸出照片像素，二者像素執行先相減再平方的運算，並從二張照片互相對映的第一個像素開始運算，並除以照片的長乘寬解析度，所以二張照片的解析度必須要一樣，否則二張照片的長與寬是無法對映起來的，尤其本方法是在做超解析度相關任務，原始輸入照片被壓縮處理，並放大成輸出照片，需要再額外備分一張原始照片純放大解析度的副本，來與處理後的輸出照片做 PSNR 計算，如果輸出照片直接與原始輸入照片進行 PSNR 處理，會發生輸出照片長寬超出輸入照片的原始長寬錯誤，二者的像素位置無法對映，來進行相減的運算，因為此時的輸入照片仍只是小尺寸，原本的像素位置已經無法與輸出照片進行像素之間的 MSE 對映計算，除非只是對照片單純做壓縮處理，再與原始照片做 PSNR 運算，不然就要對輸入照片做昇採樣來保留另外的副本，再與提昇解析度並結合壓縮處理的照片進行 PSNR 計算，算式(2)所指示的 PSNR 公式代表得到 MSE 後，進行基底為十的 log 處理，再乘以十就可以得到二張照片互相運算的 PSNR 值，通常 PSNR 值越高，代表照片品質越好，也就是原始輸入照片經過某種處理計算後，其輸出照片的品質仍能保持著與原始照片同樣的水準，如果 PSNR 值落在三十幾左右，代表是人眼觀察可以接受的結果，假設 PSNR 太低，代表失真太多，意味著輸入照片經過處理得到的輸出，其結果已經喪失原本照片中很多的細節。

接著回到圖 3 的內容，回顧前面敘述的課題，關於究竟要先對照片做 SVD 壓縮，再插值放大解析度，還是先超解析度插值後，再對照片進行 SVD 壓縮處理，這二者攸關於處理順序的問題，也已有實驗顯示如果選擇先插值放大照片解析度，再進行 SVD 的壓縮處理方案，若如此做能得到更高的 48.46 db PSNR 值（使用雙線性插值）。



圖 3. 先對照片插值，提昇照片解析度，再進行 SVD 壓縮流程

本方法將敘述的順序如下，第一部分是前言，第二部分是本方法進行的相關研究，第三部分是發想出來的架構方法，第四部分是實驗結果，第五部分是結論。

二、相關研究

超解析度技術發展至今，約略可以被分為二大種類，第一種是傳統插值方法，而第一種傳統方法經歷過稀疏編碼方法，以及建立資料庫學習方法，後來也有運用隨機森林方法，支援向量回歸等等，這些類似線性回歸觀念的方法來對照片進行超解析度，直到卷積運算執行超解析度應用 [5]，就已形成第二種大類別的深度學習方法，到後來也有生成對抗網路這個架構來處理超解析度的任務 [11]，而生成對抗網路運用生成器與判別器這二個網路，生成器負責產生高清照片，而另一個判別器的任務，則是將生成器製造出來的高清照片與備好的高清照片互相比較，判斷哪個照片更真實更清晰，二者互相較量來產出最佳的結果，所以生成對抗網路即是用上述這個觀念來處理超解析度的應用，上述所說的方法本質都是卷積運算，但上述方法在訓練運行時，高度複雜的計算對記憶體有著苛刻的要求，處理器的負荷效能幾乎無法承擔費時的計算量，必須要有價格昂貴，效能頂級的顯示卡才有辦法完成超解析度的任務 [12]，本方法採用的超解析度方法是傳統的插值領域，雖然本插值方法在處理效果上不如前述的方法，但簡潔可行，即是對照片的像素做昇採樣，再對像素矩陣進行運算，來得到超解析度後的照片，這樣的操作有明確的計算流程得到解析度放大的輸出入照片，不像前述所說的神經網路，藉由黑盒子內部的複雜運算，得到表面說得通，但背後運算不知所以的結果。

而要與超解析度插值方法結合探討的壓縮照片技術，則已形成好幾種壓縮方法的領域，約略有 kmeans [19]分群方法來對照片整體的顏色限定在某 k 個顏色種類之內，照片中所有的顏色按照限定的這 k 個顏色聚集，如此做可以壓縮照片的內容進行分群，達到壓縮照片的效果，以及 BTC [1]方



塊編碼技術對照片內容中所有的像素執行分區域的二分化像素，來簡化像素值的構成，從而輕量化照片，而本方法要專注探討的是 SVD 壓縮照片技術。

(一) SVD 壓縮照片

Compton [2]等學者針對 SVD 壓縮照片技術做了詳盡的說明，對於一步接著一步的推導步驟有著明確的指示，可以說是很好地闡述了 SVD 壓縮照片技術的內涵，Pandey [14]則在文章裡介紹了使用 matlab 軟體將 SVD 的壓縮照片過程，通過實驗再現，Swathi [6]嘗試各種不同的參數，來觀測 SVD 壓縮照片的效果，Sadek [16]則把著重的焦點放在運用圖表來繪製輸入照片在被 SVD 壓縮時，各種條件對壓縮照片造成的影響，一目了然地顯示在圖表上，Dasgupta [3]將 jpeg 引入 SVD 方法，運用 SVD 壓縮照片技術來處理 jpeg 格式的照片檔，Kahu [10]引入了所謂的特徵值(eigen value)的概念，探討 eigen value 與 SVD 壓縮照片技術的關係，並繪製成圖表顯示，Sandhu [18]使用了矩陣的型式，來解釋 SVD 的運作流程，Prasanth [15]運用像素直方圖的概念，將原始輸入照片與 SVD 壓縮處理後輸出的照片轉化成直方圖，觀察輸出入照片直方圖在 SVD 壓縮前後的變化，千葉大學的學者 Kotera Hiroaki [8]則將 SVD 壓縮處理的照片轉換成三維空間，以觀察壓縮照片的顏色在三維空間的分布，Ahmed [17]演示了選擇不同的奇異值，會產生各不相同的 SVD 壓縮照片效果，Huang [9]使用簡單的數字點矩陣照片來示範 SVD 壓縮照片的方法，Mulcahy [13]從旋轉二維平面照片的觀點來解釋 SVD 壓縮照片的概念，Gowri [7]在 SVD 壓縮處理照片的實驗中，額外地附上了照片經過 SVD 壓縮後的容量變化，假設如圖 4 所示，原始輸入照片的容量是 25.5 kb，經過 SVD 壓縮後，容量縮小至 23.4 kb，如圖 5 所示，可以見得 SVD 壓縮照片技術對於幫助照片的容量輕量化，與縮小容量方面是非常有效的。

而本方法的 SVD 壓縮照片技術也會朝著這樣的方向行進，但本方法是要結合插值來放大解析度，所以勢必要探討解析度提昇且經過 SVD 壓縮的照片，與原始小尺寸的輸入照片，這二者的照片容量究竟有什麼變化影響，通常提昇解析度的照片容量會比原始照片容量更大，而本方法也希望能與網路上的放大照片服務比較，究竟網路服務與本方法所架構的提昇照片解析度並結合 SVD 壓縮的流程，哪一種方案在輕量化照片容量的方面更可行，且保留了照片中良好的細節，以及可行的紋理品質。

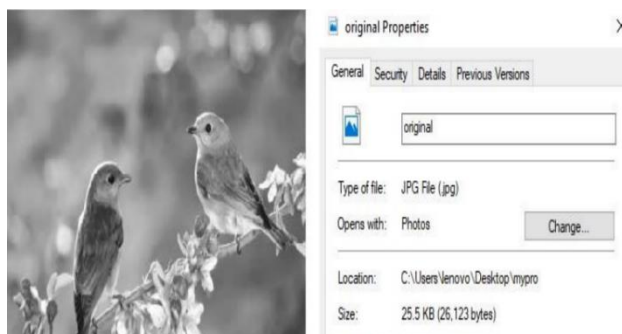


圖 4. 在對照片做 SVD 壓縮之前，容量是 25.5 kb



圖 5. 在對照片做 SVD 壓縮之後，容量縮小至 23.4 kb，且畫質不變，並額外附上是使用 190 個奇異值能擁有這樣的效果

(二) 超解析度插值

在插值方法提昇解析度領域中，本方法將列舉出最近鄰插值、雙線性插值、以及雙三次插值，他們的觀念都是先將原始輸入照片中的既有像素拓展至放大照片裡的新位置，這些既有像素搬移至放大照片後，他們之間會有空白無值的間隔，如何計算出新的像素值填充至這些空白的位置，再經由這樣的迭代計算，產生出結果照片，讓這樣插值放大流程最終呈現出來的照片，在觀看品質上是可行的。這即是此三種插值方法在規範上如何設計，運用何規則去計算新像素值，採用什麼樣的過程將新像素值填補至輸出照片中的新位置，這三種插值方法各有不同。

1. 最近鄰插值

最近鄰插值的概念是在將輸入照片插值放大時，將照片內容中的像素向外填充成此像素一模一樣的值，也就是說輸出照片中的新位置，由這個新位置映射到原始照片位置的像素來填充，如圖 6 所示。



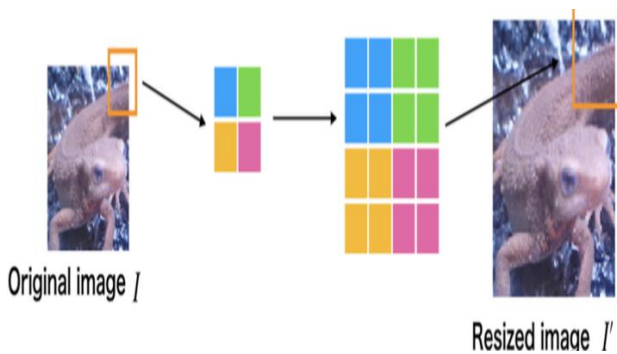


圖 6. 假設有一張原始輸入照片 original image，左方輸入照片的右上橙框所標示的像素內容，經由向外擴充同樣的像素值，得到右方的輸出照片 resized image，左右二個橙框所各自註記的輸入像素與輸出像素內容，經過中間的像素演示所轉換得到。

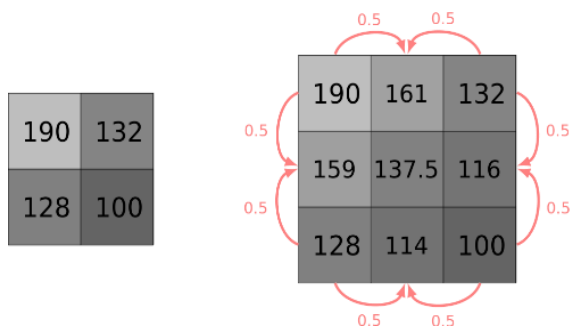


圖 7. 雙線性插值的例子

2. 雙線性插值

雙線性插值的觀念可以用一個例子來比擬，假想有一個二乘二的像素區塊，這二乘二的像素區塊要拓展成三乘三像素，如圖 7 所示，

原始的區塊像素被各自拓展至新區塊的最外角，中間的空白值則由彼此之間的平均值來填補，假設是三乘三像素拓展至五乘五等等以上的案例，則按照此想法以此類推，使用原物件等比例縮放至新物件的觀念，來建立線性延伸的映射方法，來拉伸整個像素，填滿既定的新像素。

3. 雙三次插值

在使用 bicubic 插值方法前，需要先決定權重，所謂的權重，是指雙三次插值法在對照片進行處理的時候，是對照片中整體像素矩陣分區域地進行迭代處理，藉由想要對原始照片放大多少尺寸，與原始照片的尺寸相除，得到了此縮放比例，如下述算式 (3) 右半部分的 scale，由這個 scale 縮放比例，來相乘照片長寬掃描中尋訪到的迭代子 i ，能取得映

射至照片被放大的對應座標 $mapx$ ，另外再向對應座標往下取整，這二者相減得到偏差 u 。

$$mapx = i(scale), scale = \left(\frac{inwidth}{outwidth}\right) \text{ or } \left(\frac{inheight}{outheight}\right) \quad (3)$$

$$u = mapx - floor(mapx) \quad (4)$$

$$b(ii - u) = \begin{cases} 1 - 2|x|^2 + |x|^3, & |x| \leq 1 \\ 4 - 8|x|^1 + 5|x|^2 - |x|^3, & 1 < |x| < 2 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

算式 (3)、(4)、(5) 演示了雙三次插值計算需要處理的部分，首先算式 (3) 中的 scale 縮放比例，是藉由想要對原始照片放大多少尺寸，與原始照片的尺寸相除所得到， i 意味著在掃描照片長寬中尋訪到的迭代子，透過迭代子來逐步掃描照片長寬的座標，以此對照片做插值提昇解析度處理，映射到的座標 $mapx$ 由迭代子尋訪到的座標乘以縮放比例可以得到，算式 (4) 的偏差 u ，由映射座標 $mapx$ 向下取整與原本的映射座標 $mapx$ 相減得到。

再深入地迭代掃描時，更進一步的迭代子 ii 與偏差 u 相減，代入 $b(ii - u)$ 函式處理得到權重，由此權重相乘對映座標，如此反復迭代處理，即可換算出所有座標代表的所有像素，得到整體的雙三次提昇解析度照片。

三、架構方法

本架構方法要先釐清究竟在實驗流程的順序上，是先對照片進行 SVD 壓縮，再插值放大照片的解析度，還是先藉由插值方法來提昇原始輸入照片的解析度，再進行照片的 SVD 壓縮，到底這二種選擇會對實驗結果有什麼影響之類的參數變化，而先行做這樣的試驗會用到二張實驗照片，這些照片來自西班牙 Granada 大學的資料集，其性質具有代表性意義，而且公開的測試照片集，這二張照片分別代表二大風格種類，第一個種類是風景照片（解析度為 450*310），第二個則是建築照片（解析度為 640*512），在對照片執行超解析度任務時，統一對照片的比例放大二倍，第一張風景類照片取材自西班牙 Granada 大學的公開資料集，如圖 8 所示，這張照片描繪的是一座座的岩柱，而第二張實驗的照片也同樣取材自西班牙 Granada 大學的公開資料集，如圖 9 所示，這是一張建築照，照片中左方取到的日光面積，實際上注意觀察，像是從左方打光而來。





圖 8. 第一張實驗照片——處理風景



圖 9. 第二張實驗照片——處理建築

表 1. 先壓縮再放大實驗 PSNR 表 (運算速度單位為秒)

先 SVD 壓縮照片再放大	處理風景	處理風景運算速度	處理建築	處理建築運算速度
最近鄰插值	32.36 db	≈ 2	46.97 db	≈ 5
雙線性插值	32.31 db	≈ 9	47.09 db	≈ 23
雙三次插值	31.96 db	≈ 82	46.67 db	≈ 198

表 2. 先放大再壓縮實驗 PSNR 表 (運算速度單位為秒)

先放大再 SVD 壓縮照片	處理風景	處理風景運算速度	處理建築	處理人像運算速度
最近鄰插值	39.24 db	≈ 17	42.34 db	≈ 6
雙線性插值	48.46 db	≈ 56	55.70 db	≈ 23
雙三次插值	35.16 db	≈ 467	40.89 db	≈ 203

由表 1 可以得知：如果先對照片做 SVD 壓縮，再進行表中所列舉的，任意一個皆可的插值方法，平均都可以得到 40 幾以上的 PSNR 值，若 PSNR 值在 30 左右或以上，代表是人眼觀看可以接受的品質，而此結果的 PSNR 已經超過 40 幾以上，平均在 43 左右，這樣的數據代表輸入照片經由處理，產出的結果就視覺品質而論，其效能是極其不錯的，

尤其是指原始輸入照片經由這樣的流程設計，得到的處理後照片再與原始照片進行比較，輸出入照片這二者的紋理品質並沒有相差多少。

而表 2 是改變實驗順序得到的 PSNR 結果，也就是將前面所敘述的先壓縮再放大這個流程，由「壓縮」與「放大」這二個步驟調換，變成「先」對照片做插值放大照片解析度後，「再」進行 SVD 壓縮，而這樣的實驗結果與前述的表 1 有著截然不同的差異，首先，表 1 的方案對輸入做處理都得到近似統一的 PSNR 值結果，平均值都落在 43.37，但表 2 所顯示的第二種流程方案卻得到了起伏不定的 PSNR 值，雖然在這樣的情境下，先對照片做雙線性插值，再進行 SVD 壓縮能夠得到接近 50 幾左右的 PSNR 值，最高有 55 左右，（最低 48 左右，接近 50，這樣的高數據，是指先用雙線性插值對圖 8 所示的風景照片提昇解析度，再進行 SVD 壓縮），但其他另外二個插值在 PSNR 效果上來看就沒有這麼優秀，首先探討到最近鄰插值先提昇照片解析度，在進行 SVD 壓縮照片的方案，先對照片做最近鄰插值也就意味著最終插值放大出來的照片，其內容像素有很多都是由原始輸入照片中的舊有像素移動過來的。

由圖 10 之後這些額外增廣的實驗照片測試中，可以看到右方小橙框放大至左方橙框的註記，顯示了右方人物的臉龐輪廓出現了鋸齒狀的像素區塊，沿著面龐的線條逐步蔓延，順著人物的腮幫往下，也可以看到人物的衣服也有這樣的情況，直至左方人物的衣服，他的領角也是同樣的鋸齒區塊，之所以會發生這樣的情境，可以用放大圖中的黑色衣服與背景日光做比較，黑衣服意味著較暗，較黑的像素，日光意味著較白，較亮的像素，在黑衣服代表的暗像素，與日光代表的亮像素之交界處，二者平均中和了整體的顏色，形成一個梯形向上的鋸齒線條。



圖 10. 第二張實驗照片——鋸齒線條





圖 11. 第二張實驗照片——方狀顆粒



圖 12. 第二張實驗照片— SVD 壓縮處理前後

而雙三次插值在對照片放大解析度時，發現在處理後的照片中會產生綿連不斷的方狀顆粒充斥著整個像素區域，如圖 11 所示。

在 SVD 壓縮處理的前後，如圖 12 所示，如果以人眼觀察的標準來看，在還沒 SVD 壓縮處理前的照片中，肉眼可以勉強觀察到左方人物的面色看起來稍微有點紅潤，比較偏向暖色，而右方經過 SVD 壓縮處理後的照片，人物的臉龐有一點點淡冷色系的色調。

為了先理解 SVD 的運作原理，而不是空口評價 SVD 壓縮處理照片後的效果如何，接下來探討 SVD 的理論，首先假設有某個四乘四的像素值區塊，這樣的像素區塊可以被表達成一個列向量乘以一個行向量，再乘以權重，這樣的乘法表達式有四個，通過逐步地近似累加這四個乘法表達式，最終會近似變成原本所說的那某個四乘四的像素值區塊，這樣的用意，是指像素區塊可以被分成列向量乘以行向量，藉此來簡化所攜帶的訊息，就如前述所說的四乘四像素區塊，總共十六個像素，可以被簡約成一列共四元素的列向量，與一行共四元素的行向量，是為這二者的乘積，這樣的觀念可以應用來執行壓縮照片的任務，如圖 13 所示。

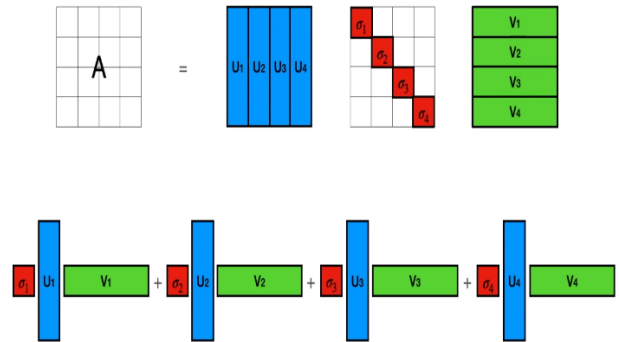


圖 13. SVD 壓縮技術示意圖

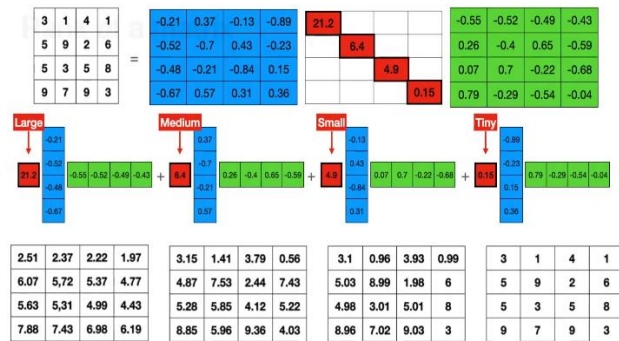


圖 14. SVD 壓縮技術範例圖

圖 13 中設想有某像素區塊 A，這個像素區塊 A 可以被分割成四個單元的模組，一個模組單位的內容，是一個共有四元素的列向量 V（綠色示意），與一個共有四元素的行向量 U（藍色示意），再一個權重 sigma（紅色示意），這三者的乘積組成一個單位，累計共有四個單元，近似相加變回原始的像素區塊 A，SVD 壓縮技術即是用這個觀念來簡化照片內容。

若有 SVD 壓縮照片的範例如圖 14 所顯示，假想有某四乘四的像素值區塊 A，這個區塊 A 的序列內容如同範例所描述地被填為 3、1、4、1、5、9、2、6、5、3、5、8、9、7、9、3，這個序列區塊可以被分成四個模組單元，也就是列向量乘以行向量乘以權重，共有四個單位近似相加，最終變回原始所描述的像素區塊，即可做到對像素內容的壓縮，推廣至照片的概念，即是面對像素全體，乃至整體圖像的內容本身，進行訊息的壓縮，SVD 技術即是用這樣的想法對照片展開壓縮，簡約化內容的描述，在傳輸上極有幫助。

接下來延續探討實驗流程順序的問題，關於究竟先對照片做 SVD 壓縮，再插值提昇解析度，還是先進行插值放大



照片解析度，再使用 SVD 技術壓縮照片，針對這個問題，前述所演示的表 1 與表 2 已經很清楚，雖然表 2 的方案——先放大再壓縮的處理有個不錯的數值，在第二項先雙線性插值的流程可以得到接近 50 幾左右的 PSNR 值，最高有 52.47，(最低 48.46 左右，接近 50)雖有如此優異的數據，但另外二項插值方法的表現與雙線性插值的表現落差太大，有的 PSNR 值甚至落到 30 幾左右，也與表 1 的先壓縮再放大方案之穩定效能相差太大，而另二種插值方法也有鋸齒線條與方狀顆粒的視覺問題，似乎也是表 1 與表 2 這兩類方法的通病，也就是說，不管先放大再壓縮，還是先壓縮再放大，這二種插值方法可能就是會產生鋸齒與顆粒這樣的視覺缺點，所以本方法的實驗流程即定為先壓縮再放大，底定了本架構的實作基調，因為表 1 穩定的 PSNR，平均都落在 40 幾左右，本方法希望一個穩健的順序下進行實驗，也就是先對照片進行 SVD 壓縮，再插值提昇照片的解析度，而給定的流程圖如圖 15 所示。

四、實驗結果

實驗的平臺使用 python 的程式語言，以及自行用程式撰寫的應用 GUI，並輔以 anaconda 的集成平臺，方法將從 PSNR 值的指標定向轉為 SVD 壓縮前後的容量變化，這樣的步驟受到 Gowri [7]的啟發，Gowri 在他的研究中，展示了照片被 SVD 壓縮處理時的實驗，並額外附上了原始照片經過 SVD 壓縮後的容量變化，本方法採納 Gowri 附上照片容量經過處理前後的呈現方式，但本方法所想要的不僅僅是只有原始輸入照片經過 SVD 壓縮前後的容量變化而已，還加入網路服務(<https://bigjpg.com/>)，前述網站使用的是 SRCNN [5]技術來放大照片，SRCNN 運用三層卷積層，就最終第三層的結果要取得高解析度的照片來看，第一層進行擷取特徵，接著第二層是非線性映射，最後得到解析度最高的那張照片，如圖 16 所示。

除此之外還有插值方法等等的試驗變量，實驗最終的目的就是要探討在照片容量方面，網路放大照片服務，與本方法架構的插值放大解析度結合 SVD 方法，哪一個能夠更有效的輕量化照片，接下來的操作如圖 17 所示，上半部分藍框區域是原始照片，下半部分橙框區域是網路服務放大的照片，給定第二張實驗照片的原始容量是 28.7 kb (藍框示意)，經由網路服務對照片尺寸放大二倍，放大後的照片容量變為 109 kb (橙框示意)。



圖 15. 流程圖

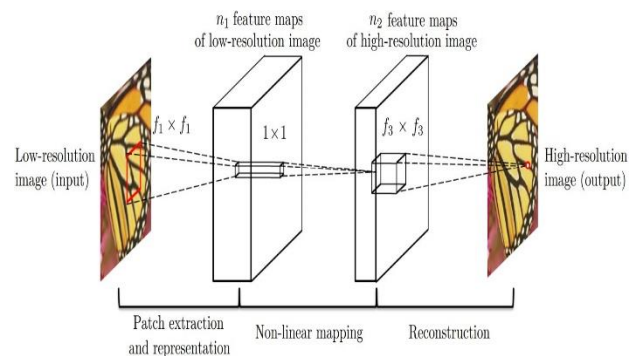


圖 16. 網路放大照片服務所用的框架



圖 17. 網路服務放大照片與原始照片比較

在圖 17 的實驗結果中，可以看到原始照片容量是 28.7 kb (藍框示意)，處理後輸出的照片容量是 109 kb (橙框示意)，容量變地龐大了足足接近五倍。

在圖 18 中，雙線性插值提昇照片解析度結合 SVD 壓縮的實驗結果中，原始輸入照片的容量是 28.7 kb (藍框示意)，壓縮處理後輸出的照片容量是 68 kb (橙框示意)，容量只是多了僅僅一倍，總共的容量頂多只是變成兩倍多一點左右，





圖 18. 雙線性插值放大照片與原始照片比較

而且此視覺品質與網路服務放大出來的處理後輸出照片不相上下，比前面所描述的另外二個插值方法更加優秀，前述的缺點有鋸齒線條與方狀顆粒，而雙線性插值在放大照片的同時，也能保留照片裡的物件中極細緻的線條，以實驗照片二的人像來舉例，右方人物臉角線條被雙線性插值放大後，仍能看清楚人物臉龐的輪廓，而不像前述的插值方法會產生方狀顆粒，即使雙三次插值照片處理後，可能會讓輸出照片整體變地明亮，但放大後隱約可見的綿密顆粒，不見得被人眼評觀所允許。

接下來的表 3，全面地揭示了實驗照片一的風景照，與實驗照片二的人像照在接受本實驗的所有方法處理後的照片容量，從實驗結果來看，雙線性插值的表現最好，風景照的原始容量是 265 kb，經由雙線性插值放大，容量約略變為二倍的 525 kb，比其他龐大了三四倍的方法來說，雙線性插值擁有良好的輕量化效果，也在插值放大照片的同時，也能妥善地保留照片中原有的細節輪廓，在人像照的例子也是如同此理。

雙線性插值的表現極為優秀，就如同表 3 所演示的，其他插值方法處理同樣的照片所產出的容量，如果由雙線性插值來執行的話，可能雙線性插值產出的容量比其他插值方法的將近一半都不到，可見雙線性插值方法在輕量化照片的前提下，也能良好地提昇照片的解析度。

表 3. 照片容量經由處理前後的實驗表

方案	處理風景	處理人像
原始容量	265 kb	28.7 kb
網路服務	815 kb	109 kb
最近鄰插值	525 kb	86.3 kb
雙線性插值	422 kb	64.9 kb
雙三次插值	626 kb	134 kb

五、結論

究竟先壓縮再放大，還是先放大再壓縮的探討，可以從表 2 的穩定 PSNR 值得到定論，在圍繞著先對照片進行 SVD 壓縮，再插值提昇解析度的應用情境下，尤以雙線性插值的表現最為可行，不論是在保留原始照片中的良好細節，亦或是輕量化輸出照片的容量方面，雙線性插值均做到了二者皆能兼顧，也希望針對另外二個插值方法的鋸齒問題、方狀顆粒進行深入探討，而插值後照片產生鋸齒狀態的情形，在於插值方法只是單純地對照片中的待處理區域向外擴散補值，並不能良好地表達原照片想要傳達的輪廓與線條之訊息，方狀顆粒是因為雙三次插值技術本身在公式計算的缺陷，雖然雙三次插值生成的照片從視覺上感覺起來很明亮，但細看卻有著微小矩形顆粒，由人眼感受起來不太美觀，本方法希望提出改良的方案，也嘗試能夠對 SVD 壓縮技術進行改善，讓 SVD 壓縮照片技術的表現更加優秀。

參考文獻

1. Chang, C. C. and J. C. Chuang (2006) Using a simple and fast image compression algorithm to hide secret information. *International Journal of Computers and Applications*, 28(4), 329-333.
2. Compton, A. E. and S. L. Ernstberger (2020) Singular value decomposition: applications to image processing. *Journal of Undergraduate Research*, 17, 99-105.
3. Dasgupta, A. and Rehna V. J. (2011) JPEG image compression using singular value decomposition. 20th International Conference on Advanced Computing, Communication and Networks, Maui, Hawaii.
4. Delp, E. J. and O. R. Mitchell (1979) Image compression using block truncation coding. *IEEE Transactions on Communications*, 27, 1335-1341.
5. Dong, C., K. He, C. C. Loy and X. Tang (2014) Learning a deep convolutional network for image super-resolution. Proceedings of 2014 European Conference on Computer Vision, Switzerland.
6. Gopichand, G., H. R. Swathi, S. Sohini and Surbhi (2017) Image compression using singular value decomposition. *IOP Conference Series Materials Science and Engineering*, 263(4), 1-8.
7. Gowri, P., K. Senbaga Priya, R. K. Hari Prasath and S.



- Pavithra (2019) Image compression using singular value decomposition. *International Journal of Mathematics Trends and Technology*, 65(8), 74-81.
8. Hiroki, K. (2003) RGB to spectral image conversion using spectral pallette and compression by SVD. Proceedings of 2003 International Conference on Image Processing, Barcelona, Spain.
9. Huang, T. S. and P. M. Narendra (1975) Image restoration by singular value decomposition. *Applied Optics*, 14(9), 2213-2216.
10. Kahu, S. and R. Rahate (2013) Image compression using singular value decomposition. *International Journal of Computers and Applications*, 2(8), 244-248.
11. Ledig, C., L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang and W. Shi (2017) Photo-realistic single image super-resolution using a generative adversarial network. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii.
12. Luu, T. M., T. Vu and C. D. Yoo (2018) Perception-enhanced image super-resolution via relativistic generative adversarial networks. Proceedings of 2018 European Conference on Computer Vision, Munich, Germany.
13. Mulcahy, C. and J. Rossi (1998) A fresh approach to the singular value decomposition. *The College Mathematics Journal*, 29(3), 199-207.
14. Pandey, J. P. and L. S. Umrao (2019) Digital image processing using singular value decomposition. Proceedings of 2nd International Conference on Advanced Computing and Software Engineering, Alanya, Turkey.
15. Prasantha, H. S., H. L. Shashidhara and K. N. B. Murthy (2007) Image compression using SVD. 2007 International Conference on Computational Intelligence and Multimedia Applications, Sivakasi, Tamil Nadu.
16. Sadek, R. A. (2012) SVD based image processing applications: state of the art, contributions and research challenges. *International Journal of Advanced Computer Science and Applications*, 3(7), 26-34.
17. Sahar Khalid Ahmed (2018) A comparison of the methods used for selecting singular values in image compression using SVD. *International Journal of Computer Applications*, 181(1), 10-15.
18. Sandhu, K. and M. Singh (2018) Image compression using singular value decomposition. *International Journal of Latest Research in Science and Technology*, 7(5), 5-8.
19. Wan, X. (2019) Application of k-means algorithm in image compression. *IOP Conference Series Materials Science and Engineering*, 563(5), 1-4.

收件：110.09.07 修正：110.10.27 接受：111.01.27

