

一個以視窗為界面之影像壓縮系統(I)

林灶生

國立勤益技術學院 電子工程系 副教授

摘要

本研究目的在建立一個綜合各種演算法的影像壓縮系統。系統中使用的演算法包括向量量化(Vector Quantization)、方塊截短編碼(Block Truncation Coding, BTC)、霍夫曼編碼(Huffman Coding), 其中向量量化包括了LBG、模糊C-均值(Fuzzy C-Mean, FCM)、處罰模糊C-均值(Penalized Fuzzy C-Mean, PFCM)、補償模糊C-均值(Compensated Fuzzy C-Means, CFCM)、處罰模糊競爭學習網路(Penalized Fuzzy Competitive Learning Network, PFCLN)、補償模糊競爭學習網路(Compensated Fuzzy Competitive Learning Network, CFCLN)、及退火霍普神經網路(Annealed Hopfield Neural Network, AHNN); 而方塊截短編碼有傳統方塊截短編碼及衝量絕對值方塊截短編碼(Absolute Moment Block Truncation Coding, AMBTC)。我們的目的在將這些理論實現並完成一個適合視窗作業系統且具親和性之人機介面, 提供一有效且快速之研究環境。

一、前言

在今日科技越來越發達之趨勢之下, 人們需要的資訊量也日益增加, 因此如何以最少的空間儲存大量的資料便是人們所一直追求的目標。而資訊則包含了大量的文字及影像資料, 其中影像資料若未經壓縮往往達數百KB甚至GB的大小, 也就是說一張磁片通常只能存幾張圖片。因此在醫學界的斷層掃描或工商業界的視訊會議, 均要求影像作非常有效的應用, 所以藉由指導同學專題研究設計影像壓縮系統, 找尋新的影像壓縮演算法。我們以Borland C++ Builder為工具, 將傳統演算法及作者所發展之新方法, 融合而成一個影像壓縮系統; 希望提供給影像壓縮影像壓縮領域研究者一個有效且快速之工具。

二、影像壓縮原理

影像壓縮之目的在於減少儲存空間與縮短傳輸所需的時間。一般而言, 影像壓縮可概分為無失真影像壓縮法(Lossless Image Compression)與失真影像壓縮法(Lossy Image Compression)二類。影像資料可視為資訊與累贅(Information and Redundancy)之組合。無失真影像壓縮法只刪除累贅部份, 毫不損傷資訊部份; 反之, 失真影像壓縮法不僅刪除累贅部份, 且亦除去部份資訊。由於失真影像壓縮法可得到較高之資料壓縮比, 因此, 在不影響原來影像品質的情況下, 成為壓縮技術之研究主流。本節僅就本系統相關之影像壓縮演算法做概念性之說明。

2.1 向量量化

向量量化為一種簡單且壓縮比相當高之演算法，在此一領域中包括了編碼簿之設計與編碼簿之搜尋二個研究方向，本系統只探討編碼簿之設計。編碼簿的產生是將要壓縮的影像集中選出幾張具有代表性的影像做為訓練集(Training Set)，然後以這個訓練集訓練出編碼簿。最簡化的編碼簿應該就是使用這張影像本身為訓練集所產生的。這種編碼簿我們稱之為區域的編碼簿(Local Codebook)。使用區域碼簿通常都可以有相當好的壓縮效率。這是因為該影像所特有之特徵都會被自己的區域編碼簿所充份地考慮進去，因而可找得到足以代表的碼向量。就編碼簿之設計，在此系統中總共包括了 LBG [1, 2]、模糊 C-均值(Fuzzy C-Mean, FCM) [3-5]、處罰模糊 C-均值(Penalized Fuzzy C-Mean, PFCM) [6, 7]、補償模糊 C-均值(Compensated Fuzzy C-Means, CFCM) [8]、處罰模糊競爭學習網路(Penalized Fuzzy Competitive Learning Network, PFCLN) [9]、補償模糊競爭學習網路(Compensated Fuzzy Competitive Learning Network, CFCLN) [8]、及退火霍普神經網路(Annealed Hopfield Neural Network, AHNN) [10]，本節僅簡略說明各個演算法之動作，欲詳知其理論根據，請查閱相關參考論文。

2.1.1 Linde-Buzo-Gray(LBG)演算法

LBG 演算法是由 Linde, Buzo, 及 Gray [1]三位先生所發展出來，基本上與 C-均值演算完全一樣，茲就其應用於編碼簿設計說明如下 [2]:

1. 從一向量訓練集開始，設定起始編碼簿 $\{X_i\}$ ， $i=1,2,\dots,N_c$ ，選擇一個失真估算函數 d 以及設定一個失真改善臨界值 ϵ (實數)。設定回合計數器 l 為 1，並設定所有訓練向量之平均失真起始值， $D(0)$ ，為一個相當大的值。
2. 對所有的訓練向量， X ，計算它與各個碼向量， $\{X_i\}$ ， $i=1,2,\dots,N_c$ ，之距離， $d(X, X_i)$ 。若 $d(X, X_i) \leq d(X, X_j)$ ， $j=1,2,\dots,N_c$ ，則將 X 將對應到 X_i (歸到 X_i 這一群)。如此下來，我們會將整個訓練集分成 N_c 個群。計算由分群的過程中所產生的平均失真值 D 。如果從前一個回到這個回合的平均失真改善比例小於等於臨界值 ϵ ，則認定已經到達收斂的情形，演算法停止。否則繼續執行第三個步驟。

$$\frac{D(n-1) - D(n)}{D(n-1)} \leq \epsilon$$

3. 更新編碼簿，以 $\{X_{i(n+1)}\}$ 取代 $\{X_{i(n)}\}$ ，其中 $\{X_{i(n+1)}\}$ 為第 i 群的訓練向量經過誤差最小化之條件計算而得。若是 MSE 誤差估算，則 $\{X_{i(n+1)}\}$ 正是第 i 群向量之正均值(中心)， $\{X_{i(n+1)}\}$ 的第 k 個元素為所有第 i 群內的訓練向量之第 k 個元素的平均值， $k=1,2,\dots,n$ 。設定 $l \leftarrow l+1$ 並回到第 2 步驟。雖然每一個回合都會使平均失真值減少，但是如果 ϵ 的值設得太小也可能得花上好幾個回合才能達到收斂，因此一般我們都會設定一定次數的上限，若是 l 大於這個上限，演算法也予以中止。

2.1.2 模糊 C-均值演算法 (FCM)

模糊理論是以一數學架構定義人類辨識處理之不確定性，自 1965 年 Zadeh[11]發表模糊集合理論以來，模糊集合理論已被應用於不同之領域，諸如醫學影像分析、影像壓縮、圖訊識別

等。與一般群集技術一樣，模糊群集技術亦使用最小平方差準則，模糊目的函數為

$$J_{FCM} = \frac{1}{2} \sum_{j=1}^c \sum_{i=1}^n (u_{i,j})^m |z_x - w_j|^2$$

其中“|·|”為歐基里得(Euclidean)距離。在群集分析上，模糊 C-均值為一相當知名且功能甚強的技術。

2.1.3 懲罰模糊 C-均值演算法 (PFCM)

另外一種具有額外懲罰項，經由 Yang[6, 7]所證明比 FCM 方法更有效率的模糊群集策略，被稱為懲罰模糊 C-均值(Penalized Fuzzy C-means, PFCM)演算法,其目的函數被定義如下:

$$\begin{aligned} J_{PFCM} &= \frac{1}{2} \sum_{j=1}^c \sum_{i=1}^n u_{i,j}^m |x_i - w_j|^2 - \frac{1}{2} v \sum_{j=1}^c \sum_{i=1}^n u_{i,j}^m \ln \alpha_j \\ &= J_{FCM} - \frac{1}{2} v \sum_{j=1}^c \sum_{i=1}^n u_{i,j}^m \ln \alpha_j, \end{aligned}$$

其中， α_j 是群集 j 的比例常數， $v(\geq 0)$ 是一個常數值。當 $v=0$ ，則 J_{PFCM} 等於 J_{FCM} ，並且同群集強度 α_j ，群集中心 w_j ，以及歸屬程序 $u_{i,j}$ 分別被定義成

$$\alpha_j = \frac{\sum_{i=1}^n u_{i,j}^m}{\sum_{j=1}^c \sum_{i=1}^n u_{i,j}^m}; j = 1, 2, \dots, c \quad (1)$$

$$w_j = \frac{\sum_{i=1}^n u_{i,j}^m x_i}{\sum_{i=1}^n u_{i,j}^m}, \text{ 及} \quad (2)$$

$$u_{i,j} = \left(\frac{\left(|x_i - w_j|^2 - v \ln \alpha_j \right)^{1/(m-1)}}{\sum_{l=1}^c \left(|x_i - w_l|^2 - v \ln \alpha_l \right)^{1/(m-1)}} \right)^{-1}; i = 1, 2, \dots, c. \quad (3)$$

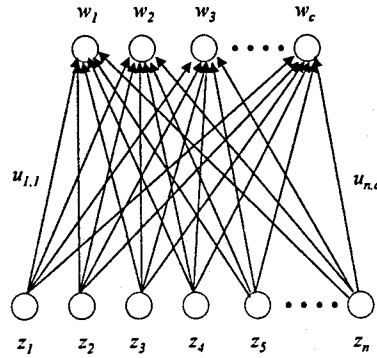
2.1.4 處罰模糊競爭學習網路(PFCLN)

處罰模糊競爭學習網路主要是結合競爭學習神經網路(Competitive Learning Neural Network)或稱全贏神經網路(Winner-Take-All Neural Network)與懲罰模糊C-均值而成。什麼是競爭式學習法？正如名稱所形容的，當輸入呈現時，網路上的神經元會彼此競爭，以便獲得被活化的機會，而這個機會只賦予對輸入最有反應的類神經元，這個被稱作優勝者的類神經元，其加權值會被調整，以便增加其與此時輸入之間的相似性。有一點值得注意的是，通常每個類神經元的加權值會被「正規化」(Normalized)成長度為1的單位向量。競爭學習神經網路架構如(圖一)所示。其目的函數與懲罰模糊C-均值之目的函數完全一樣，並以一學習速率 η 依下式修正其輸出值

$$W_j(n+1) = W_j(n) + \Delta W_j(n) \quad (4)$$

其中

$$\Delta W_j = -\eta \frac{\partial E}{\partial w} = \eta \sum_{i=1}^n \mu_{i,j}^m (X_i - W_j) \left[1 - \left[\frac{m}{m-1} (1 - \mu_{i,j}) \right] \right] \quad (5)$$



(圖一)競爭學習網路架構圖

2.1.5 補償模糊 C-均值演算法 (CFCM)

為了改進懲罰模糊 C-均值演算法之執行效率及增加重建品質，補償模糊 C-均值演算法將懲罰模糊 C-均值演算法之懲罰項 $-\ln(\alpha_j)$ 改成補償項 $\tanh(\alpha_j)$ 。其目的函數則修改成：

$$\begin{aligned} J_{CFCM} &= \frac{1}{2} \sum_{j=1}^c \sum_{i=1}^n u_{i,j}^m |x_i - w_j|^2 + \frac{1}{2} v \sum_{j=1}^c \sum_{i=1}^n u_{i,j}^m \tanh(\alpha_j) \\ &= J_{FCM} + \frac{1}{2} v \sum_{j=1}^c \sum_{i=1}^n u_{i,j}^m \tanh(\alpha_j), \end{aligned}$$

且其歸屬程度 $u_{i,j}$ 被更改成

$$u_{i,j} = \left(\frac{\left(|x_i - w_j|^2 + v \tanh \alpha_j \right)^{1/(m-1)}}{\sum_{l=1}^c \left(|x_i - w_l|^2 + v \tanh \alpha_l \right)^{1/(m-1)}} \right)^{-1}; i=1,2,\dots,c. \quad (6)$$

其餘之定義均同於懲罰模糊 C-均值演算法。

2.1.6 補償模糊競爭學習網路(PFCLN)

補償模糊競爭學習網路架構與處罰模糊競爭學習網路完全一樣，不同點在於歸屬程度係以第(6)式修正之。

2.1.7 退火霍普神經網路(AHNN)

退火神經網路(Annealed Neural Network, ANN)是一種最適化應用網路模式，它是由 D.E Van den Bout 與 T.K. Miller 於 1988 年所提出[12]。雖然用霍普菲爾-坦克網路解決組合最佳化問題有許多優點，然而，因為此方法受限於它是一種最陡坡降(steepest descent)程序，所以通常收斂於一個最鄰近起始值的局部最小值，而非總體最小值。因而只能用許多不同的起始值去搜

尋總體最小值，但此一方法仍不能保證收斂值至總體最小值。此外，在應用霍普菲爾-坦克網路解決組合最佳化問題時，有許多參數需決定，而且解答的優劣對參數值的設定十分敏感，然而參數的設定卻缺乏系統化的方法，只能已嘗試錯誤的方式為之。退火神經網路(Annealed Neural Network)是一種相當新的類神經網路模式，它融合了模擬退火演算法的高解答品質，與霍普菲爾-坦克神經網路快速收斂的優點。退火神經網路已被用以解決許多組合最佳化問題，例如旅行推銷員問題。

AHNN 結合了退火技術(Annealing Algorithm)、一種新的冷卻程序(Cooling Schedule)、以及能將能量快速集中於一點的霍普菲爾網路(Hopfield Neural Network)，是一種用來解決將能量最佳化問題的新技術。對一個二維的 n 個訓練向量、取樣大小為 c 的影像來講，AHNN 由 n 乘 c 個神經元組成一個二維陣列。而這樣的分類問題可以利用映對著退火式霍普菲爾神經網路以達到能量函數的保留。在經過反覆的更新及神經元之間的連結之後，當在能量最小化時將可將整個網路的能量集中到一個穩定的狀態。

而 AHNN 也利用了波芝曼(Boltzmann)的可能性分布來排除在能量函數中一些影響因子，每一個神經元被定義為一個坐標 (x,i) (其中 i 介於 1 到 c ， x 參考每一個神經元組介於 1 到 n) 因此它的神經元被定義為 $V_{x,i}$ ，而取樣方面的資料則被定義為 $W_{x,i,y,i}$ ，而外部偏差值的取樣對於神經元來講可被定義為 $I_{x,i}$ ；根據這樣的方式能量函數可被定義為

$$E = \frac{1}{2} \sum_{x=1}^n \sum_{i=1}^c V_{x,i} \left| z_x - \sum_{y=1}^n W_{x,i,y,i} V_{y,i} \right|^2 - \sum_{x=1}^n \sum_{i=1}^c I_{x,i} V_{x,i} \quad (7)$$

為了強制地產生一個適應性的定義，目的函數將被定義為

$$E = \frac{A}{2} \sum_{x=1}^n \sum_{i=1}^c V_{x,i} \left| z_x - \frac{1}{\sum_{h=1}^n V_{h,i}} z_y V_{y,i} \right|^2 + \frac{B}{2} \sum_{x=1}^n \sum_{i=1}^c \sum_{j=1}^c V_{x,i} V_{x,j} + \frac{C}{2} \left[\left(\sum_{x=1}^n \sum_{i=1}^c V_{x,i} \right) - n \right]^2 \quad (8)$$

E 是所有分布在訓練向量 Z_x 及 Z_y 歐基里得距離所定義的能量總和，相同地 V 是屬於 0 到 1 之間的神經元狀態。分布在此函數中的第二個項目嘗試著讓任何的訓練向量不會在最後的解時隸屬於聚類 i 和 j 。第三個項目保證這些 n 個訓練向量只分布在 c 個聚類，很明確地最後兩個處罰項可以負起強迫的功用，而第一個最小項讓能量最小化。而這三個項目由設計者用三個餘因子結合起來成一個項目進而簡化成

$$E = \frac{1}{2} \sum_{x=1}^n \sum_{i=1}^c V_{x,i} \left| z_x - \frac{1}{\sum_{h=1}^n V_{h,i}} z_y V_{y,i} \right|^2 \quad (9)$$

為了讓合理的時間內找到最佳化的值，而新的神經元狀態依據能量及溫度的變化可依下式組成一個波芝曼分布：

$$V_{x,i} \propto e^{-\Delta E_{x,i}/T} \quad (10)$$

且中值場能量 $E_{x,i}$ 可以由上式目的函數計算求得為：

$$E_{x,i} = \left| z_x - \sum_{y=1}^n \frac{1}{\sum_{h=1}^n V_{h,i}} z_y V_{y,i} \right|^2 \quad (11)$$

而神經元狀態可依下式被正規化

$$V_{x,i} = \frac{e^{-E_{x,i}/T}}{\sum_{j=1}^c e^{-E_{x,j}/T}} \quad (12)$$

退火神經網路最重要的是需要一退火排程(Cooling Schedule)來掌控整個網路之收斂速度與效能。作者[10]針對退火神經網路發展出快速且能將能量逼近整體最小值之冷卻排程，此冷卻排程定義如下：

$$T_k = \frac{1}{\beta + 1} [\beta + \tanh(\alpha)^k] T_{k-1}, \quad k = 1, 2, \dots \quad (13)$$

其中 T_k 為第 k 次之冷卻溫度，且 α ， β 均為常數。

2.2 方塊截短編碼

方塊截短編碼(Block Truncation Coding, BTC) [2, 13] 在影像壓縮上使用非常廣泛，近年雖然也有人試著將 BTC 用在一維訊號上，總之都不如使用在二維訊號來的自然。在一個 BTC 系統中，影像首先被分割成許多不重疊的 $n \times n$ (一般是 4×4) 的方塊，每一個方塊上面的像素可能有不同之灰階值，但是在經過 BTC 編碼後，只剩下兩個值。對於任何一個方塊，決定於方塊上像素值的統計情況，我們以一個位元圖(Bit Map)及兩個重建階(Reconstruction Levels)， a 和 b ，來表示該方塊。首先我們計算原方塊裡所有像素的量化臨界值(通常為像素值之平均)，原來之像素值若大於此臨界值，則位元圖上該位置的位元設為 1;否則設為 0。所有位元圖上 1 的位值，重建之像素值都為 b ;所有位元上 0 的位置，重建之像素值都是 a 。而 a 、 b 的選擇一般則以保留原方塊的統計特性，使得重建方塊也具此特性為標準。

由於臨界值的區隔使得重建方塊上產生一個梯狀邊。原始的 BTC 設計也會在原來變化很緩慢的方塊上在編碼後產生輪廓線(Contouring Artifacts)，這是因為重建方塊中重建值的突然改變所造成。因為每個方塊各別處理之故，還有其他不良的效應產生，如邊(Edge)之不連續等。往好的方面講，BTC 裡的臨界值處理可以使影像中明顯的邊忠實地重建，其他壓縮法通常都會使明顯的邊變得模糊。

另一種可以得到更佳效果的方法，我們稱之為動量絕對值 BTC(Absolute Moment BTC, 簡稱 AMBTC) [2]。AMBTC 的量化器被設計成保留住動量的絕對值不變。其平均值 \tilde{X} 與第一絕對主要動量之定義分別下：

$$\tilde{X} = \frac{1}{m} \sum_{i=1}^m X_i \quad (14)$$

$$\alpha = \frac{1}{m} \sum_{i=1}^m |X_i - \tilde{X}| \quad (15)$$

如果我們還是設定量化器的臨界值為 \tilde{X} ，能保持 X 與 a ，不變的重建階為

$$a = \tilde{X} - \frac{m\alpha}{2(m-q)} \quad (16)$$

$$b = \tilde{X} - \frac{m\alpha}{2q} \quad (17)$$

其中 q 為方塊中像素值大於 \tilde{X} 的數目。和前面一樣，我們可以直接將 a 和 b 編碼送出，也可以聯合量化 \tilde{X} 和 α 以降低位元率。

2.3 霍夫曼編碼

Huffman 編碼與 Shannon-Fano 編碼有許多的共同特性，它也為每個符號產生不等位元數的碼。同時，機率比較高的符號使用比較少的位元數來編碼，它也是具有唯一解碼性質的即時解碼，因此它可以建立一個解碼樹。建立 Huffman 解碼樹與建立 Shannon-Fano 解碼樹的程序完全不同，Shannon-Fano 解碼樹是從上往下建，先找出每個碼的 MSB(Most Significant Bit), 最具意義位元)位元，然後往下做，直到樹葉為止。Huffman 解碼樹則反過來，從樹葉開始工作直到樹根為止。建立 Huffman 樹的過程很簡單，首先每個符號先攤開，成為 Huffman 樹的樹葉，這些樹葉將經由下列的演算法接成一棵樹，附在每一個樹葉的是每個符號出現的頻率或機率，並當做該樹葉的加權比重(Weight)。一般來講，Shannon-Fano 編碼的效率和 Huffman 編碼差不多，但是，Huffman 編碼效率至少一定不會比 Shannon-Fano 差，因此它很快地便成為這類編碼法的主流。由於兩種編碼法所需要之處理時間差不多，Huffman 編碼可以很容易證明它是所有即時編碼中最好的。Huffman 編碼在許料壓縮處理標準中被採用，例如由 CCITT 所提出之傳真壓標準及 JPEG 等，由此可見其重要性。今天，我們仍然可以在許多學術性期刊上找到關於 Huffman 編碼的論文。

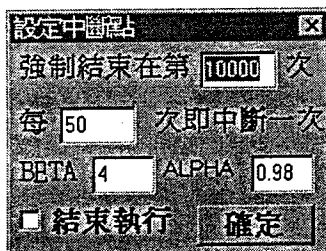
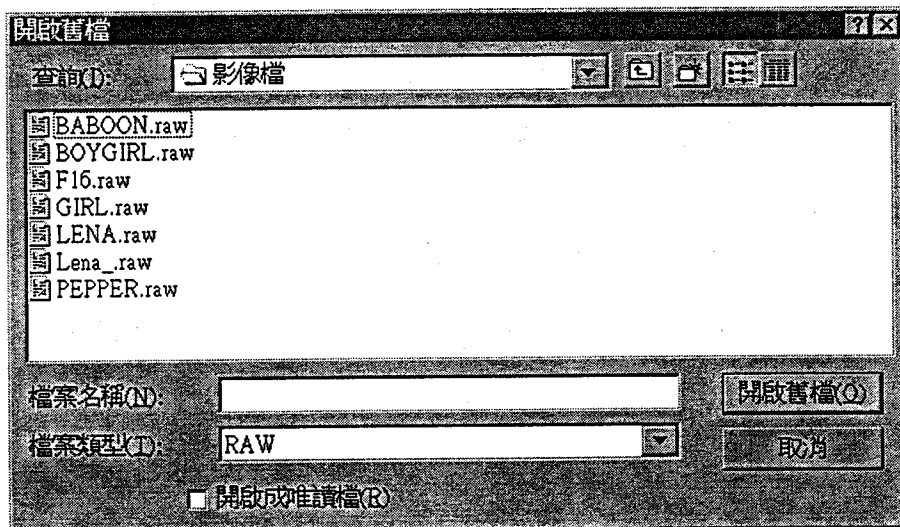
三、影像壓縮系統操作說明

如(圖二)所示，影像壓縮系統包括了檔案開啓、向量量化、方塊截短編碼、霍夫曼編碼、轉換編碼、分頻邊碼、中斷點設定、及關於本系統說明等八種選單。除上面幾節所列之演算法外，餘者尚未完成，有待以後繼續將其他方法加入本系統中。

再如(圖二)，選擇檔案選項後，再依(圖三)之瀏覽器選擇適當之影像檔案置於系統畫面左側，接著再選擇適當壓縮方法加以執行。解壓縮重建結果則置於系統畫面右側，重建品質則以峰值訊號雜訊比(Peak Signal Noise Ratio, PSNR)與均方差根值(Root Mean Square Error, RMSE)計量。計量結果與相關參數則置於系統畫面下方。程式會根據(圖四)設定中斷點內的次數來做強制結束，執行 AHNN 法時，可先在此調整 α 及 β 的值。



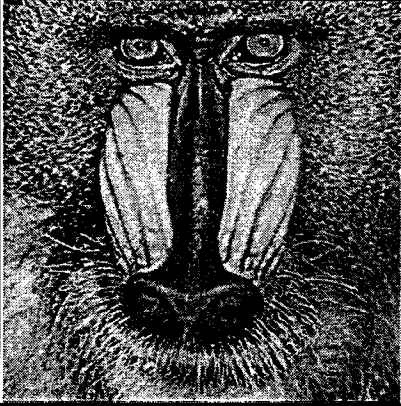

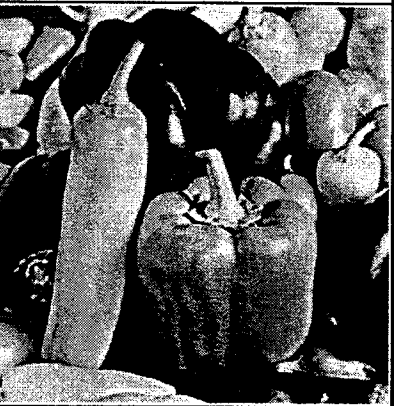






(圖二) 影像壓縮系統畫面



(圖四) 中斷點及相關參數設定畫面

四、重建影像品質與執行效能

由於篇幅所限，本節僅將 CFCM 演算法並設定編碼簿為 64 及 256 做向量量化重建所得之結果顯示於(圖五)中，(表一)至(表六) 則列出各種向量量化方法與 BTC 法之重建品質與執行效率。由表中可知 BTC 演算法比向量量化演算法有較佳之重建品質與執行效率，然其壓縮比較低。在所列之向量量化方法中所有模糊方法(包括結合競爭神經網路)之重建品質均較 LBG 方法好，

| CFCM 測試報告 | | | | 取樣方式 4 X 4 | | | |
|------------|---|----------|--|------------|---|----------|----------|
| 圖形名稱 | | BABOON | | F16 | | PEPPER | |
| 原始圖案 |  | |  | |  | | |
| |  | |  | |  | | |
| 編碼簿方式為 64 | | | | | | | |
| PSNR | RMSE | 23.10824 | 17.82905 | 25.67858 | 13.26206 | 26.09372 | 12.64312 |
| 編碼簿方式為 256 |  | |  | |  | | |
| | | | | | | | |
| PSNR | RMSE | 24.62126 | 14.97881 | 28.38578 | 9.71072 | 29.23105 | 8.8115 |

(圖五)以 CFCM 演算法設定編碼簿為 64 及 256 做向量量化重建所得之結果

惟執行效率較差。而退火霍普式神經網路演算法重建影像之 PSNR 較所有模糊方法差且不超過 0.5dB，此差距並不足以表示其重建品質較差。由於每一種方法之起始設定均為隨機，因此可見論文中所提向量量化之新方法所得重建品質均較傳統 LBG 佳。

(表一) 取樣方式為 4 X 4 且編碼簿大小為 64，各種向量量化方法與 BTC 之 PSNR

| 取樣方式 4 X 4 | | 編碼簿大小 64 | | | 各方法之 PSNR 差異 | | | |
|------------|----------|----------|----------|----------|--------------|----------|---------|--|
| 圖形名稱 方法 | BABOON | F16 | PEPPER | LENA | GIRL | BOYGIRL | 壓縮 比 | |
| LBG | 23.22211 | 25.32375 | 25.94514 | 26.52908 | 28.60057 | 29.30120 | 12.8 | |
| FCM | 23.09378 | 25.68064 | 26.16400 | 26.56181 | 29.40039 | 30.31229 | 12.8 | |
| PFCM | 23.12645 | 25.64344 | 26.18491 | 26.61610 | 29.39751 | 30.20069 | 12.8 | |
| CFCM | 23.10824 | 25.67858 | 26.09372 | 26.60396 | 29.48468 | 30.28129 | 12.8 | |
| PFCLN | 23.11246 | 25.63218 | 26.20037 | 26.48478 | 29.40769 | 30.32350 | 12.8 | |
| AHNN | 23.00456 | 25.47659 | 26.00987 | 26.29342 | 29.27896 | 30.21906 | 12.8 | |
| BTC | 26.99811 | 28.48102 | 28.82573 | 29.67366 | 32.95513 | 33.64334 | 4 | |

(表二) 取樣方式為 4 X 4 且編碼簿大小為 128，各種向量量化方法與 BTC 之 PSNR

| 取樣方式 4 X 4 | | 編碼簿大小 128 | | | 各方法之 PSNR 差異 | | | |
|------------|----------|-----------|----------|----------|--------------|----------|---------|--|
| 圖形名稱 方法 | BABOON | F16 | PEPPER | LENA | GIRL | BOYGIRL | 壓縮 比 | |
| LBG | 23.90884 | 26.28867 | 27.09784 | 27.59416 | 29.51619 | 30.04648 | 10.667 | |
| FCM | 23.81772 | 26.89954 | 27.59851 | 27.79109 | 30.70711 | 31.49320 | 10.667 | |
| PFCM | 23.85029 | 26.88466 | 27.57771 | 27.76586 | 30.66213 | 31.64758 | 10.667 | |
| CFCM | 23.81798 | 26.80819 | 27.64430 | 27.83161 | 30.56454 | 31.56454 | 10.667 | |
| AHNN | 23.58234 | 26.69012 | 27.40176 | 27.48831 | 30.39265 | 31.48912 | 10.667 | |
| BTC | 26.99811 | 28.48102 | 28.82573 | 29.67366 | 32.95513 | 33.64334 | 4 | |

(表三) 取樣方式為 4 X 4 且編碼簿大小為 256，各種向量量化方法與 BTC 之 PSNR

| 取樣方式 4 X 4 | | 編碼簿大小 256 | | | 各方法之 PSNR 差異 | | | |
|------------|----------|-----------|----------|----------|--------------|----------|---------|--|
| 圖形名稱 方法 | BABOON | F16 | PEPPER | LENA | GIRL | BOYGIRL | 壓縮 比 | |
| LBG | 24.69498 | 27.15751 | 28.17485 | 28.89154 | 30.43074 | 31.05354 | 8 | |
| FCM | 24.66791 | 28.31605 | 29.30182 | 29.18167 | 31.84471 | 33.06210 | 8 | |
| PFCM | 24.66106 | 28.35360 | 29.26603 | 29.26696 | 31.80319 | 33.17432 | 8 | |
| CFCM | 24.62126 | 28.38578 | 29.23105 | 29.20225 | 31.67905 | 33.13349 | 8 | |
| PFCLN | 24.59734 | 28.37654 | 29.11892 | 29.13320 | 31.71623 | 33.04028 | 8 | |
| AHNN | 24.39123 | 28.05673 | 29.03781 | 28.91294 | 31.52968 | 32.89723 | 8 | |
| BTC | 26.99811 | 28.48102 | 28.82573 | 29.67366 | 32.95513 | 33.64334 | 4 | |

(表四) 取樣方式為 4 X 4 且編碼簿大小為 64，各種向量量化方法與 BTC 之執行效率

| 取樣方式 4 X 4 | | 編碼簿大小 64 | | | 各方法之執行時間比例 | | | |
|------------|--------|----------|--------|------|------------|---------|------|--|
| 圖形名稱 方法 | BABOON | F16 | PEPPER | LENA | GIRL | BOYGIRL | 壓縮比 | |
| LBG | 28 | 88 | 98 | 142 | 302 | 102 | 12.8 | |
| FCM | 589 | 1024 | 840 | 630 | 638 | 522 | 12.8 | |
| PFCM | 1140 | 885 | 484 | 1347 | 1399 | 987 | 12.8 | |
| CFCM | 657 | 859 | 443 | 582 | 1107 | 2355 | 12.8 | |
| PFCLN | 367 | 703 | 792 | 489 | 658 | 674 | 12.8 | |
| AHNN | 269 | 583 | 629 | 315 | 410 | 481 | 12.8 | |
| BTC | 30 | 30 | 30 | 30 | 30 | 30 | 4 | |

(表五) 取樣方式為 4 X 4 且編碼簿大小為 128，各種向量量化方法與 BTC 之執行效率

| 取樣方式 4 X 4 | | 編碼簿大小 128 | | | 各方法之執行時間比例 | | | |
|------------|--------|-----------|--------|------|------------|---------|--------|--|
| 圖形名稱 方法 | BABOON | F16 | PEPPER | LENA | GIRL | BOYGIRL | 壓縮比 | |
| LBG | 105 | 296 | 346 | 688 | 1100 | 688 | 10.667 | |
| FCM | 1122 | 1233 | 1152 | 1098 | 1272 | 765 | 10.667 | |
| PFCM | 1152 | 924 | 603 | 780 | 1008 | 1110 | 10.667 | |
| CFCM | 834 | 1089 | 1041 | 849 | 1119 | 522 | 10.667 | |
| AHNN | 218 | 338 | 129 | 141 | 254 | 321 | 10.667 | |
| BTC | 30 | 30 | 30 | 30 | 30 | 30 | 4 | |

(表六) 取樣方式為 4 X 4 且編碼簿大小為 256，各種向量量化方法與 BTC 之執行效率

| 取樣方式 4 X 4 | | 編碼簿大小 256 | | | 各方法之執行時間比例 | | | |
|------------|--------|-----------|--------|------|------------|---------|-----|--|
| 圖形名稱 方法 | BABOON | F16 | PEPPER | LENA | GIRL | BOYGIRL | 壓縮比 | |
| LBG | 498 | 1362 | 1764 | 4060 | 5138 | 4844 | 8 | |
| FCM | 4560 | 10170 | 5745 | 4875 | 9750 | 3585 | 8 | |
| PFCM | 6225 | 6810 | 3345 | 5730 | 7560 | 5715 | 8 | |
| CFCM | 3420 | 9270 | 2895 | 5520 | 5715 | 7095 | 8 | |
| PFCLN | 1287 | 2672 | 2435 | 4950 | 6345 | 5010 | 8 | |
| AHNN | 786 | 821 | 943 | 1470 | 1734 | 1452 | 8 | |
| BTC | 30 | 30 | 30 | 30 | 30 | 30 | 4 | |

五、結果與討論

本文提出一結合各種不同影像壓縮演算法之發展系統，做為未來發展其它影像壓縮演算法之建構工具。未來希望加入其他諸如轉換編碼、分類編碼、預測編碼、碎形編碼及加入基因演

算法(Genetic Algorithm)與配合其他神經網路之影像編碼法，使本系統更趨完善。

六、誌謝

本文承蒙 林基源老師及其專題同學建立系統雛型，呂永茂等同學之實驗協助，以及相關參考文獻的幫忙，方得以完成現在之規模，在此致上最大之謝意。

七、參考文獻

- [1] Y. Linde, A. Buzo, and R. M. Gray, 1980, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 85-94.
- [2] 戴顯權, 1996, 資料壓縮, 松崗圖書, 台北.
- [3] Jzau-Sheng Lin, Shao-Han Liu, and Chi-Yuan Lin, 1998, "The application of fuzzy Hopfield neural network to design better codebook for image vector quantization," *IEICE Trans. on Fundamentals*, vol. E81-A, no. 8, pp. 1645-1651.
- [4] Jzau-Sheng Lin, K. S. Cheng, and C. W. Mao, 1996, "A fuzzy Hopfield neural network for medical image segmentation," *IEEE Trans. Nucl. Sci.*, vol.43, pp.2389-2398.
- [5] Jzau-sheng Lin, K. S. Cheng, and C. W. Mao, 1996, "Multispectral magnetic resonance images segmentation using fuzzy Hopfield neural network," *J. Biomed. Comput.*, vol.42, pp.205-214.
- [6] M. S. Yang, 1993, "On a class of fuzzy classification maximum likelihood procedures," *Fuzzy Sets and Systems*, vol. 57 pp. 365-375.
- [7] M. S. Yang, and C. F. Su, 1994, "On parameter estimation for normal mixtures based on fuzzy clustering algorithms," *Fuzzy Sets and Systems*, vol. 68 pp. 13-28.
- [8] Jzau-Sheng Lin, "Fuzzy clustering using a compensated fuzzy Hopfield network," to be appeared in *Neural Processing Letters*.
- [9] C. Y. Lin and Jzau-Sheng Lin, 1998, "The Application of Penalized Fuzzy Competitive Learning Network to Image Compression," 第四屆電腦與通訊研討會, pp. 199-204.
- [10] Jzau-Sheng Lin, 1999, "An annealed Hopfield neural network to vector quantization for image compression," *Optical Engineering*, vol. 38, pp. 599-605.
- [11] L. A. Zadeh, 1965, "Fuzzy sets," *Inform. Control*, vol. 8, pp. 338-353.
- [12] D. E. Van Den Bout, and T. K. Miller III, 1988, "A traveling salesman objective function that works," *International Conference on Neural Network*, vol.II, pp 299-303.
- [13] E. J. Delp, O. R. Mitchell, 1979, "Image compression using block truncation coding," *IEEE Trans. Commun.* Vol. COM-27, pp. 1335-1342.