

第四章 實驗數據與效能比較

本章將呈現本論文所提出的架構實際效能量測的實驗數據與效能比較。首先要看的是一個不含 Nios 處理器的 k NN 分類器，當 $k=5$ 時 (k 為最接近鄰居的個數)，和我們提出的單模組系統與其他系統做比較的結果如表 4-1：

FPGA Implementation	FPGA Device	Number of codewords	Vector Dimension	Number of LEs	Maximum Clock Rate
Subspace k NN ($k=5, M=1$)	Altera Stratix EP1S40	1280	64	2415	50 MHz
Subspace PDS($M=1$)	Altera Cyclone EP1C20	256	64	3112	50 MHz
Hierarchical VQ[2]	Xilinx Vertex 400	256	64	6276	16MHz

表 4-1 k NN、NN 和 HVQ 的比較表

面積複雜度的量測是在 Altera 的 Stratix EP1S40F780C5 FPGA 發展板上合成硬體電路，發展板提供的邏輯單元數 (logic elements, LEs) 最大值為 41250 LEs [11]。至於選擇在 Stratix FPGA 上實現驗證硬體電路，是因為可規劃系統晶片設計 (System On a Programmable Chip, SOPC) 可以快速的將硬體設計實現驗證，具備快速上市與系統再修改等特性，非常適合做雛型設計。

硬體實現時，設計集內儲存預先設計好的碼字，碼字如圖 4-1 所示是由五張 128×128 的灰階圖經過計算所得來 ($t=1280$)，碼字的維度是 8×8 (i.e., $n=3$)，只儲存 $Y_{Lm}^1, \dots, Y_{Lm}^t$ ，其中 $m=2$ ，也就是說 Y_{Lm}^j 的維度是 4×4 。位元平面縮減的參數 $l=6$ ，VSDC 運算單元一次計算四個係數 ($\delta=4$)。



圖 4-1 設計集中五張 128×128 的圖樣

相同維度下，在我們的 kNN 系統中，設計集的碼字數量高達 1280 個，並且比 NN 單模組系統和 HVQ 系統 [2] 多出了複雜的排序電路以及類別決定電路，可是邏輯單元數卻只有 2415 LEs，面積複雜度只是 NN 單模組系統的 77%，更只有 HVQ 系統的 38% 而已。

接下來討論若一個 kNN 分類器採用我們提出內嵌於 Nios 處理器的部分距離搜尋專用硬體電路，在不同的模組參數 M 的情況下實現時所需花費的面積複雜度，量測結果如表 4-2 所示。

Design Set = 1280			
<i>M</i>	1	2	3
CPU time (μs)	137.77	69.13	46.13
LEs	6391	7441	8384

表 4-2 快速 kNN 分類器系統面積複雜度與計算時間關係表

雖然說實現子空間部分距離搜尋架構增加模組數量時，需要較多的 VSDC 運算單元以及存取 Design Set ROM，會有較高的繞線額外負擔，但從表 4-2 中可以觀察到，固定碼字數量 t 的情況下， M 個模組的系統執行時間會接近 1 個模組的系統執行時間的 $\frac{1}{M}$ 倍，也就是 M 個模組的系統執行速度會是 1 個模組的系統執行速度的 M 倍，可是要額外付出的硬體空間成本（含 Nios CPU）卻只有 16% 左右，可見本論文提出的架構對於 kNN 分類器部分距離搜尋演算法則的 VLSI 硬體實現有很大的幫助。

更進一步地來看本論文提出架構的效能，表 4-3 比較了部分距離搜尋在 Pentium4 處理器上以軟體實現和本論文提出的架構在 Nios 處理器上硬體實現所需的執行時間數據。

CPU type	PentiumIV 3.0 GHz		PentiumIV 1.8 GHz		Nios 50 MHz		
algorithm	kNN	Subspace PDS	kNN	Subspace PDS	Subspace PDS		
Implementation	Software	Software	Software	Software	Hardware/ Software Codesign (M=1)	Hardware/ Software Codesign (M=2)	Hardware/ Software Codesign (M=3)
CPU time (μs)	16079.2	262.5	24912.5	497.5	137.77	69.13	46.13

表 4-3 軟硬體 kNN 分類器效能比較

此時執行時間定義為對每個輸入向量完成 kNN 分類比對所需花費的平均 CPU 時間 (μs)。在表 4-3 中，Nios 處理器相關呈現的數據包含本論文提出的架構在一個模組到三個模組的執行時間量測；Pentium4 處理器相關呈現的數據包含了兩種純軟體搜尋的方式，一種是單純的 kNN 演算法則，沒有使用子空間搜尋或部分距離搜尋法則，另一種則是在小波域上做子空間的部分距離搜尋。在這個實驗中，kNN 分類器的設計集有 1280 個 8×8 的碼字 (i.e., $N=8$ 、 $n=3$)，影像來源同圖 4-1。子空間部分距離搜尋的各個參數分別為 $m=2$ 、 $l=6$ 以及 $\delta=4$ 。

Nios 處理器內嵌部分距離搜尋硬體電路可以透過增加模組個數的方式來達到進一步降低執行時間的目的，特別是模組數從一增加到二的時候降低的效果最為顯著，平均執行時間從 $137 \mu s$ 降到 $69 \mu s$ ，

下降了將近 50% 的時間。

另外，從表 4-3 中可以看到在 Nios 處理器內嵌部分距離搜尋硬體電路上執行效果也比在 Pentium4 處理器上以純軟體的方式實現相同演算法則來的好。一般基本的 kNN 演算法則在 Pentium4 3.0GHz 處理器上以純軟體的方式實現平均執行時間要 $16079\mu s$ 左右，使用子空間部分距離搜尋法則的實現平均執行時間只要 $262\mu s$ 左右，而我們以 Nios 50MHz 處理器實現子空間部分距離搜尋法則單模組架構卻僅需要 $137.77\mu s$ ，執行時間僅是 Pentium4 3.0GHz 處理器的 52% 左右；以 Nios 50MHz 處理器實現三模組的平均執行時間更是只有 $46\mu s$ ，比 Pentium4 3.0GHz 處理器快了 5 倍以上。由以上的各項實驗結果來看，本論文提出的架構對於硬體實現非常的有效益，不論是在效能或面積複雜度上都有優異的表現。