

# 利用 GPU 提升高光譜影像分類中 特徵萃取的計算效能

## Using GPU computing to improve the performance of feature extraction processing for hyperspectral image classification

張智昌\*

徐百輝\*\*

Chy-Chang Chang

Pai-Hui Hsu

### 摘要

在災害防救領域上，空間資訊對於災害的防救工作而言，一直扮演著重要的角色，然而在空間資訊的資料處理以及應用模式上，常見的問題就是資料量往往非常龐大，模式運算複雜且耗時。但是對於時間緊迫的災害應變期間，最缺乏的就是時間。因此尋求快速的資料處理與模式推估的結果，一直是努力的目標。高光譜影像含有細緻且豐富的光譜資訊，許多利用多光譜影像無法辨識的地物，可以利用高光譜影像來解決。近年來，國內外學者針對高光譜影像的特性，提出了一些光譜分析及影像分類的方法，並衍生出各種應用。但是高光譜影像具有高維度資料的特性，多達數百個波段的頻譜資料的這項優點，卻因為資料量過於龐大，反而造成資料處理以及分析與後續應用的困難。傳統上通常是以統計方法或是波段組合選取的方式來解決維度過高的問題，利用降維度的方式來進行，雖然是目前常見的解決方法，然而在降維度的過程中亦面臨許多問題，如以統計方法進行降維度，除了計算量龐大外，恐有漏失隱藏在內的資訊，而利用波段組合的方式進行降維度，若要將所有可能的組合都進行分析，那將是個天文數字。本團隊嘗試利用 GPU 可用來進行高速運算的特性，應用在高光譜影像的波段選取上，實驗結果，在計算效能上獲得了有效的改善，對於時間上有急迫需求的應用，如環境監測、災害應變等，能快速的提供有用的資訊。

關鍵字：高光譜影像、特徵萃取、GPU

---

\* 國家災害防救科技中心助研究員  
Senior Assistant Research Fellow, National Science and Technology Center for Disaster Reduction (Corresponding author)  
國立台灣大學土木工程學系博士班研究生  
Ph. D student, Department of Civil Engineering, National Taiwan University

\*\* 國立台灣大學土木工程學系助理教授  
Assistant Professor, Department of Civil Engineering, National Taiwan University

## Abstract

Because of the high spectral resolution of hyperspectral image, more detailed and subtle spectral signatures are provided and can be used to identify and quantify a large range of surface materials which cannot be identified by multispectral images. It seems that the high dimensionality of hyperspectral data should increase the abilities and effectiveness in classifying land use/cover types. However, the classification methods which have been successfully applied to multispectral data in the past are not as effective as to hyperspectral data. As the dimensionality increases with the number of bands, the number of training samples needed for training a specific classifier should be increased exponentially as well. A simpler, but sometimes very effective way of dealing with hyperspectral data is to reduce the number of dimensionality. This can be done by feature extraction that a small number of salient features are extracted from the hyperspectral data when confronted with a limited set of training samples. Bands selection is a kind of feature extraction methods that the optimal subset of spectral bands are chose for the discrimination of classes based on some assessing criteria. A selection criterion widely used in image classification is to determine the separability of different classes. In this study, statistical class separability is calculated based on the training data to estimate the expected error in the classification for various bands combinations. However, as the dimension of data increases, the combination of bands to be examined increases exponentially, resulting in unacceptable computational cost. Therefore, it is expected to reduce the calculation time by GPU computing. In this study, the bands selection algorithm was firstly paralleled based on the GPU programming model, and then an AVIRIS data set was used to test the efficiency of the GPU computing. The result showed that the GPU increases computing speed and the computation cost can be exactly reduced.

Keywords : Hyperspectral Images, Feature Extraction, GPU Computing

## 前 言

近年來遙測科技的進步，應用層面相當的廣，例如，氣象人員利用雲圖觀測與預測天氣、測量人員利用航空相片來製作地圖、環境學者依賴遙測收集環境資訊、農業人員辨識遙測影像以監測農作物生長的狀況等，災害研究相關人員用其作為災情的判釋與災因的分析上。加上遙測影像在空間解析度與光譜解析度都有相當大的提升，對於各類高精度的資料收集以及更準確的預測與分析都有很大的幫助。然而，更精確的資料意味著需要花更多的時間來進行分析，尤其對於具有時間急迫性的災防應用上，時間是最缺乏的資源，因此利用資訊科技的高效能運算技術來加速各類資料分析，將可有效改善效能上的問題。

高光譜影像資料包含了數十至數百個波段資訊，其資料量較傳統多光譜影像龐大許多，相對的，在萃取與分析這些光譜特徵上，也複雜許多。過去在進行影像資料的分析大多透過統計分析的方法，利用大量的訓練樣本來協助分類的進行，但由於高光譜影像過多的光譜資訊，造成進行統計分析的時間過於冗長，因此若能從這龐大的光譜資訊中，萃取出有效的光譜資訊，再利用這些資訊進行影像的分類，可以達到降低資料維度並提高分類精度，另外為解決計算時間過於冗長的問題，本研究嘗試利用 GPU 的高效能運算特性，透過資料分析與改善演算法，提升計算效能，降低運算所需時間。

## 高光譜影像的特徵萃取

### (一) 維度縮減

由於高光譜影像具有數十至數百個波段的資訊，因此透過維度縮減的作法，可以有效地來加速後續分析與避免過多的資訊造成影像判釋上的困難，一般而言，維度縮減有兩種做法，如圖 1，一是 feature selection，另一是 feature extraction (Tadjudin and Landgrebe, 1998)。feature selection 是從原始資料集合中，透過一些評估標準 (assessing criteria)，直接選取部分子集合組成具最佳可分離能力 (鑑別度) 的子集合；feature extraction 是將原始資料透過線性或非線性轉換的方式，得到一組具有可分離能力的新特徵。也有一些研究將兩種方法整合，先透過 feature extraction 得到具有可分離能力的一組新的特徵，再利用 feature selection 選出具有最佳可分離能力的子集合 (Jain et al., 2000)，本研究針對計算耗時的 feature selection 進行計算效能的提升，feature selection 的計算流程如後述。

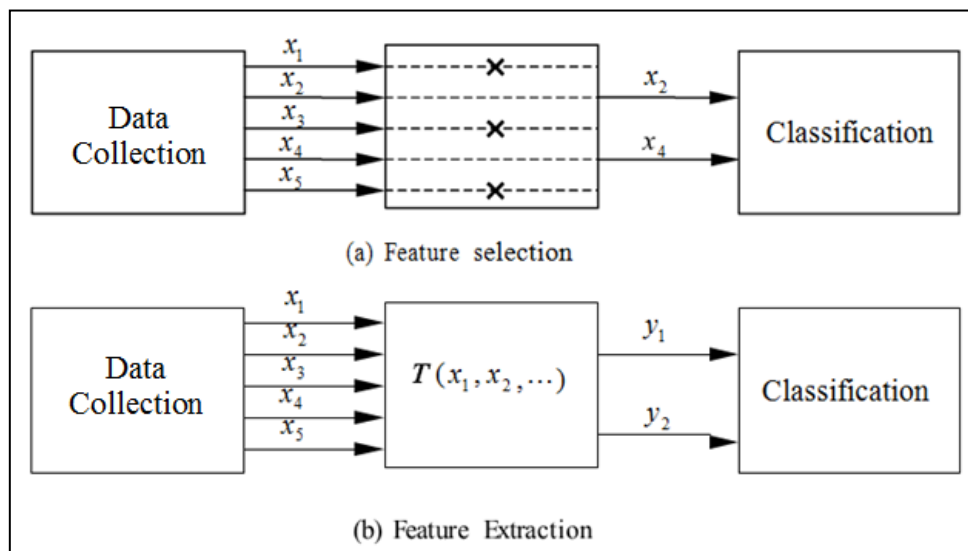


圖 1 The diagram of dimension reduction

### (二) 特徵選取 (feature selection)

「特徵選取」(feature selection)的目標是要從原有的特徵中挑選出最佳的部分特徵，使其辨識率能夠達到最高值。這些鑑別能力較好的特徵，不但能夠簡化高光譜影像分類的計算，而且也可以幫助我們瞭解各分類之間的因果關係。假設我們有一組特徵  $V = \{v_1, v_2, \dots, v_d\}$ ，我們希望透過某種分類方式來得到可分離能力  $J(\cdot)$ ，此可分離能力是所選取特徵的函數；而特徵選取的目標，就是要從  $V$  中找出一組可分離能力最佳的子集合  $S$ ，使得  $J(S) \geq J(T)$ ，其中  $T$  是任何由  $V$  所形成的子集合。

本研究中，選取子集合的準則是利用 Bhattacharyya distance 來測量不同特徵組合的可分離能力，以一個具有  $N$  個波段的高光譜影像來說，從  $N$  個波段中選取  $M$  個波段的作為子集合，其中  $M < N$ ，這樣的組合數有  $N! / (N - M)! M!$ ，然而，維度的增加使得組合數以指數的速度成長，例如要從 48 個波段中選取 12 個波段的子集合，其可能的組合數就高達 69,668,534,468 種，因此還需要利用最佳搜尋的判定程序，以避免必須計算所有可能的組合數。

### (三) Bhattacharyya distance

在統計學中，Bhattacharyya distance 用來測量兩個離散機率分布，在分類中常用測量類別之間的可分離能力，本研究用它來計算不同訓練樣本子集合其預期誤差的大小，以測量不同特徵組合的可分離能力，假設  $P_e$  是分類的誤差(classification error)， $J = 1 - P_e$ 。Bhattacharyya distance 的計算如式 1

$$J(\cdot) = \frac{1}{8} (\mu_2 - \mu_1)^T \left( \frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} (\mu_2 - \mu_1) + \frac{1}{2} \ln \left[ \frac{|\Sigma_1 + \Sigma_2|}{2 \sqrt{|\Sigma_1| |\Sigma_2|}} \right] \quad (1)$$

其中

$\mu_i$ :  $i$  類別子集合的平均值向量

$\Sigma_i$ :  $i$  類別子集合的協方差矩陣

### (四) 最佳搜尋程序(optimal search procedure)

一般來說，要選出一組可分離能力最好的子集合，並沒有什麼好方法，將所有可能的子集合逐一計算即可，然而，高光譜影像的維度高，各種波段所組合成的子集合數量龐大，一一計算太過耗時，效能太差，因此必須有一個搜尋程序來協助子集合的選取，常見的搜尋程序有 Sequential forward selection(SFS)以及 Sequential backward selection(SBS)。

SFS 的特徵組合選取程序如下：

1. 一次挑選一個特徵進行計算，得到可分離能力最高的一個特徵。
2. 下一次挑選的特徵與先前挑選的最佳特徵合併，計算出合併後的最佳特徵組合。

3. 重複前一步驟，直到所有特徵述選取完畢，即可得到具有最佳可分離能力的子集合。

SBS 的特徵組合選取程序如下：

1. 全部的特徵，一次移除一個特徵進行計算，得到可分離能力最高的一個特徵。
2. 下一次先前挑選的最佳特徵中再一次移除一個特徵，計算出移除後的最佳特徵組合。
3. 重複前一步驟，直到特徵數量選取完畢，即可得到具有最佳可分離能力的子集合。

SFS 與 SBS 的選取程序如圖 2 與圖 3。

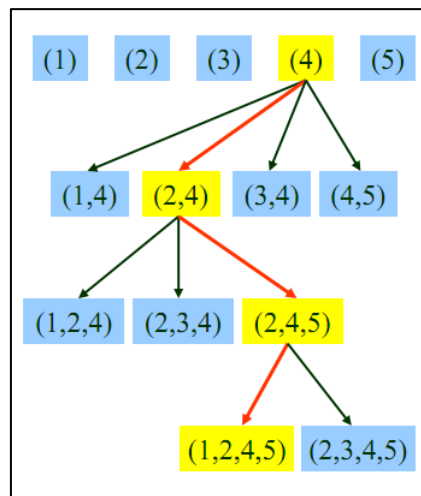


圖 2 Sequential forward selection( SFS)

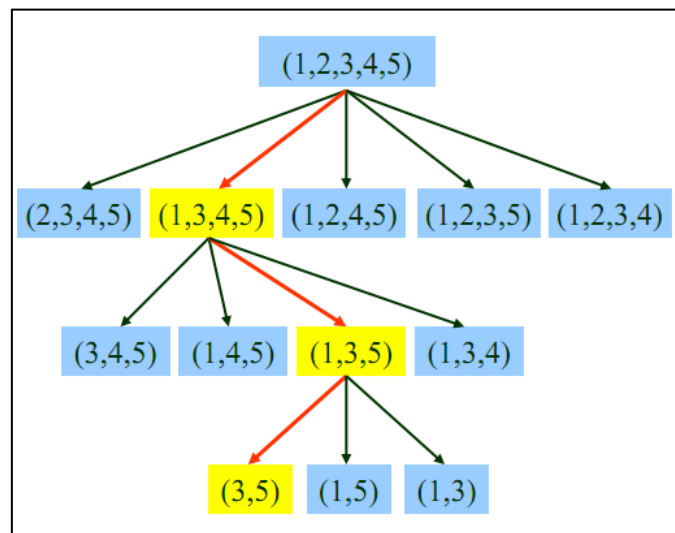


圖 3 Sequential backward selection( SBS)

## 運算效能的提升

### (一) 平行運算

由於整個特徵選取的過程，計算量龐大，雖然已可利用最佳搜尋程序來減少計算量，但最佳搜尋程序本身的計算量已是非常龐大，加上 Bhattacharyya distance 的計算亦非常耗時，因此必須尋求能加快運算速度的方法，以解決運算耗時的問題。目前科學界對於資料龐大且運算模式複雜的問題，常使用平行處理的方式，來加快運算的效能。本研究使用多核心 CPU 以及圖形處理器來進行平行運算以改進效能。

多核心 CPU 的平行運算模式是透過將原始程式依不同的硬體架構改寫成符合平行運算架構模式的演算法，再透過作業系統或是控制程式，將工作分散至不同的運算單元來執行（圖 4），可大大縮短運算時間。

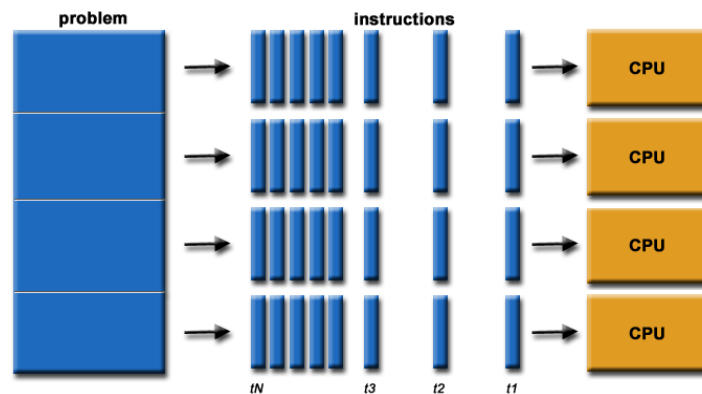


圖 4 多核心 CPU 平行運算(NVIDIA,2005)

平行運算可提升的效率如式 (2) 所示，其中 P 指程式中可平行化的比例，N 是指可用的運算處理單元數，S 是指可提升的速度。

$$S = \frac{1}{(1 - P) + \frac{P}{N}} \quad (2)$$

### (二) GPU with CUDA

CUDA (Compute Unified Device Architecture) 是一種可在圖形處理器(Graphic Processing Unit,GPU)進行平行運算的計算環境。CUDA 是一種以資料分配為基礎的一種平行處理架構，程式設計者可以利用支援 CUDA 的程式語言，例如 C 或 Matlab 寫程式，設計資料分配 (data decomposition) 及程式流程將運算工作分配到上千個執行緒(threads)及圖形處理器中數以百計的計算核心 (cores)來進行平行運算。由於以資料為基礎的平行運算，在演算法的設計上必須考慮資料的記憶體配置方式，將可以資料為基礎的平行化處理運算過程交給 GPU 進行運算，因此一些記憶體間資料交換頻繁的運算，必須將資料重新切割處理，或是交給 CPU 處理，否則計算上反而費時。圖 5 為 GPU 的硬體架構。

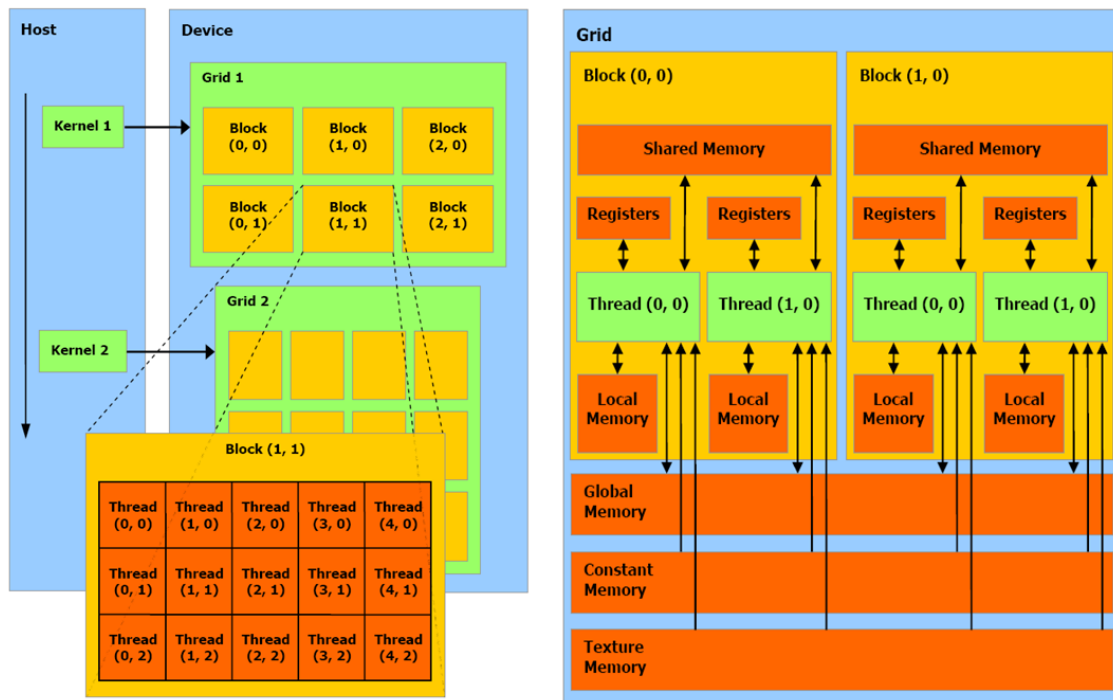


圖 5 GPU 的硬體架構(NVIDIA,2005)

## 實驗與成果

### (一)實驗設備與測試資料

實驗的設備如下表所示：

表 1 實驗設備

<ul style="list-style-type: none"> <li>● CPU           <ul style="list-style-type: none"> <li>- Intel Core 2 Quad Q6600 @ 2.4 GHz</li> <li>- Memory:4G</li> </ul> </li> <li>● GPU           <ul style="list-style-type: none"> <li>- Tesla C2050</li> <li>- ComputeCapability: '2.0'</li> <li>- DriverVersion: 3.2000</li> <li>- MaxThreadsPerBlock: 1024</li> <li>- MaxShmemPerBlock: 49152</li> <li>- MaxThreadBlockSize: [1024 1024 64]</li> <li>- MaxGridSize: [65535 65535]</li> <li>- Memory:3G</li> </ul> </li> <li>● Software           <ul style="list-style-type: none"> <li>- Visual Studio 2010</li> <li>- Matlab 2010b</li> <li>- Parallel Computing Toolbox 5.0</li> <li>- CUDA 3.2</li> </ul> </li> </ul>
--

測試資料是使用 AVIRIS (Airborne Visible/Infrared Imaging Spectrometer)1992 年產製的高光譜影像，拍攝範圍是 2 x 2 mile，地點是美國的 Indiana，影像大小 145px x 145px，具有 220 個波段的光譜資料。

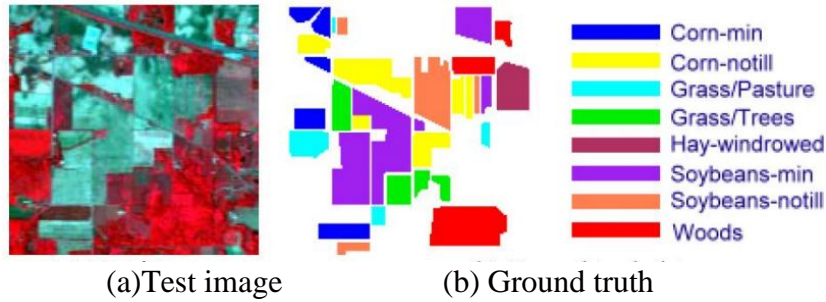


圖 6 測試資料：AVIRIS 高光譜影像

## (二) 成果

由於 Bhattacharyya distance 的計算過程中，必須用到反矩陣運算(式 1)，因此將需要反矩陣運算的矩陣，先做三角矩陣的處理再進行運算，可以大幅減少記憶體內資料交換的頻繁程度，以提高 GPU 運算效能。經改寫後的 Bhattacharyya distance 演算法如下。

```

m1=mean(X1);
C1=cov(X1);
m2=mean(X2);
C2=cov(X2);
C=(C1+C2)/2;
dm=(m1-m2)/chol(C);
try
d=0.125*dm*dm'+0.5*log(det(C/chol(C1*C2)));
catch
d=0.125*dm*dm'+0.5*log(abs(det(C/sqrtm(C1*C2))));
end

```

分別利用 CPU 與 GPU 進行 Bhattacharyya distance 的計算實驗結果如圖 7

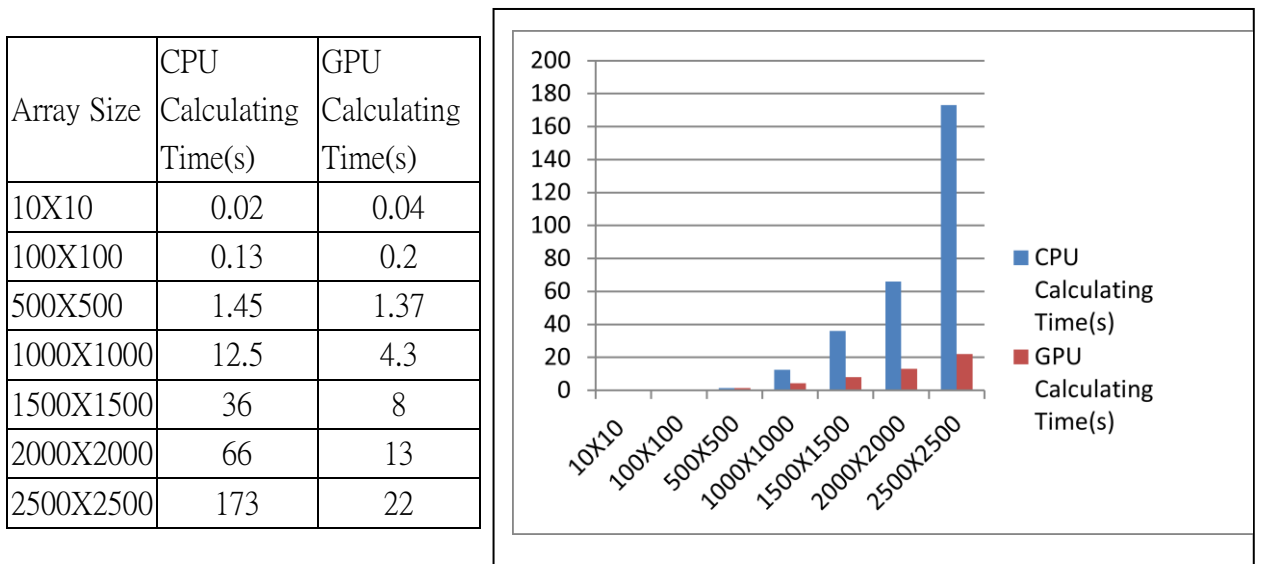


圖 7 Bhattacharyya distance 運算效能比較



Feature Number	Combinations	CPU Single-Core	CPU Quad-Core
1	98	0.25	0.44
2	4,753	11.77	3.43
3	152,096	450	110
4	3,612,280	11571	2774
5	67,910,864		45506

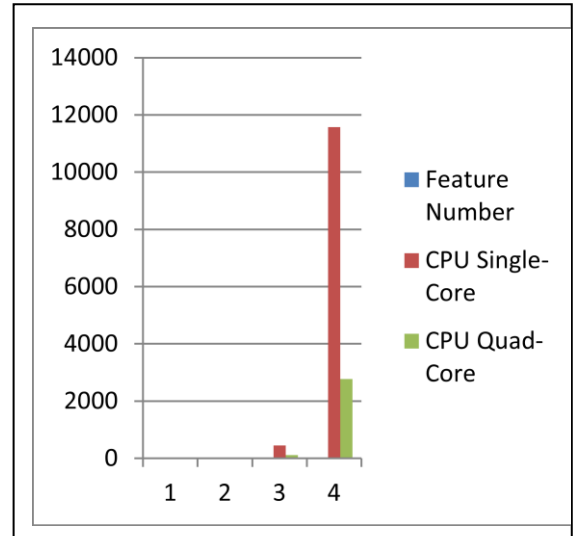


圖 8 Feature selection 運算效能比較

## 結 論

高光譜影像的分類過程中，由於光譜維度多，資料量龐大，必須進行維度縮減，以利分類的進行，利用特徵選取的方式可以降低高光譜影像的維度，並且維持影像分類的準確度。利用 GPU 的多運算核心優點，可以有效地縮短運算時間，且資料量愈大，與 CPU 運算的效能差距越大，當矩陣大小達 2500X2500 時，GPU 較 CPU 快約 8 倍以上。而多核心 CPU 也較單核心 CPU 的運算效能高，然而無論是 CPU 與 GPU 的運算，都必須針對演算法平行化以及資料的分割上進行處理。長期以來空間資訊的分析上，運算效能一直是空間研究人員亟待突破的瓶頸之一，經由本研究實驗的結果可以發現，未來可將平行運算技術用於空間資訊的分析上，應可大幅改善空間資訊分析上，運算效能不彰的問題。

## 參考文獻

- Jain, A. K., R. P. W. Duin, and J. Mao (2000) Statistical Pattern Recognition: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 4-37.
- J. Plaza, A. Plaza, R. P´erez, and P. Mart´inez (2008), Parallel Classification of Hyperspectral Images Using Neural Networks Computational Intelligence for Remote Sensing, Studies in Computational Intelligence, Vol. 133.
- Pai-Hui Hsu (2003) Spectral Feature Extraction of Hyperspectral Images using Wavelet Transform, Dissertation for Ph. D, National Cheng Kung University, Tainan, pp, 45-72.
- Roman Dudek , Carmelo Cuenca , Francisca Quintana (2007) Accelerating space variant Gaussian filtering on graphics processing unit, Proceedings of the 11th

international conference on Computer aided systems theory, February 12-16.

Tadjudin, S. and D. A. Landgrebe (1998) Classification of high dimensional data with limited training samples, in School of Electrical & Computer Engineering. West Lafayette: Purdue University, pp. 123.

CUDA C Programming Guide (2005) NVIDIA.