Issue: 1

Firmin Kam

CLASSIFICATION TECHNIQUES FOR HYPERSPECTRAL REMOTE SENSING

DEFENCE COLLEGE OF MANAGEMENT AND TECHNOLOGY

MPhil THESIS

Dec 2009

CRANFIELD UNIVERSITY

DEFENCE COLLEGE OF MANAGEMENT AND TECHNOLOGY

DEPARTMENT OF INFOMATICS AND SENSORS

MPhil THESIS

Academic Year 2007-2009

Firmin Kam

CLASSIFICATION TECHNIQUES FOR HYPERSPECTRAL REMOTE SENSING

Supervisor: Dr P Yuen

Dec 2009

Abstract

This study concerns with classification techniques in high dimensional space such as that of Hyperspectral Imaging (HSI) data sets, with objectives of understanding the strength and weakness of various classifiers and at the same time to study how their performances can be assessed particularly when there is an absence of ground truth target map in the data set. The thesis summaries the work that carried out during the course of this study and it encompasses a brief survey of machine learning and classification theories, an outline of the HSI instrumentations, data sets that collected in the study and classification analysis.

It is found that the supervised classifiers such as the Maximum Likelihood (QD) and the Mahalanobis Distance (FD) classifiers, especially when they are coupled with techniques like Regularised Discriminant Analysis (RDA) or leave-one-out covariance estimations (LOOC), have demonstrated excellent performances comparable to that of the more complicated and computational costly classifiers like the Support Vector Machine (SVM). This work has also revealed that separability measures such as the Total Transformed Divergence (TTD) and Total Jeffries-Matusita Distance (TJM) can be an invaluable method for assessing the goodness of classification in principle. However, the present methods for the evaluation of the separability measures are insufficient for achieving this goal and further work in this area is needed. This study has also confirmed the effectiveness for using RDA and LOOC techniques for a better estimation of the covariance when the sample size is small, ie when the sample size per class to band ratio (β) is less than 100.

Through team work this study has contributed partially a number of publications in the area of hyperspectral imaging and machine visions.

List of Publications

1. Tong Chen, Peter Yuen, Kan Hong, Aristeidis Tsitiridis, Firmin Kam, James Jackman, David James, Mark Richardson, William Oxford, Jonathan Piper, Francis Thomas & Stafford Lightman,2009. Remote sensing of stress using Electro-optics imaging technique*. Proceedings of the SPIE,* 7486, pp. 0601-06-12.
2. Kan Hong, Peter Yuen, Tong Chen, Aristeidis Tsitiridis, Firmin Kam, Mark Richardson, David James, William Oxford, Jonathan Piper, Francis Thomas, Stafford Lightman,2009. Detection and classification of stress using thermal imaging technique. *Proceedings of the SPIE*, 7486, pp. 0I01-0I09.
3. Aristeidis Tsitiridis, Peter Yuen, Kan Hong, Tong Chen, Firmin Kam, James Jackman, David James & Mark Richardson, 2009. A biological cortex like target recognition and tracking in cluttered Background. *Proceedings of the SPIE,*7486, pp. 0G01-0G12.
4. Peter Yuen, Izzati Ibrahim, Kan Hong, Tong Chen, Aristeidis Tsitiridis, Firmin Kam, James Jackman, David James & Mark Richardson, 2009. Classification Enhancements in Hyperspectral Remote Sensing Using Atmospheric Correction Preprocessing Technique. *The Technical Defence S&T Bulletin (Buletin Teknikal S&T Pertahanan),* pp.91-99.
5. Peter Yuen, Tong Chen, Kan Hong, Aristeidis Tsitiridis, F Kam, James Jackman, David James, Mark Richardson, L Williams, William Oxford, Jonathan Piper, Francis Thomas& Stafford Lightman. Remote detection of stress using Hyperspectral imaging technique. *To be published in the proceeding of the 3rd International Conference on Crime Detection and Prevention ICDP-09*.
6. Peter Yuen, Kan Hong, Tong Chen, Aristeidis Tsitiridis, Firmin Kam, James Jackman, David James, Mark Richardson, L Williams, William Oxford, Jonathan Piper, Francis Thomas & Stafford Lightman. Emotional & physical stress detection and classification using thermal imaging technique. *To be published in the proceeding of the 3rd International Conference on Crime Detection and Prevention ICDP-09*.
7. Peter Yuen, Aristeidis Tsitiridis, Kan Hong, Tong Chen, Firmin Kam, James Jackman, David B.James & Mark A. Richardson. A cortex like neuromorphic target recognition & tracking in cluttered background. *To be published in the proceeding of the 3rd International Conference on Crime Detection and Prevention ICDP-09.*

Acknowledgements

I’d like to thank my Supervisor, Dr. Peter Yuen, for his support and encouragement to complete this thesis. During my years at Cranfield University, Dr. Yuen has taught me, not only about science and research but also how to work with people and life. I am greatly thankful for that.

Thanks also to the Sensor Group of the DOIS who provides scholarship for the financial support of this study and especial thanks to Dr. Mark Richardson who has given me comments and suggestions for improving my knowledge in electro-optics and sensing.

Thanks to the colleagues of our group who gave me inspiration on my works over the two years I have been at Cranfield University. I also thanks Kan Hong and Tong Chen, they both made my life at Cranfield so much easier by giving me helpful hands when I had trouble with the experiment. I am sure that the friendships I formed there will continue after I have left.

Last but not least, I thank all my loved ones who gave me moral supports for finishing up this thesis.

TABLE OF CONTENTS

List of Tables

List of Figures

1 Introduction

1.1 Research objectives

Classification technique has been a vital technology for the effective functioning of all surveillance system but the assessment of the performance of classifiers can be non-trivial particularly when there is an absence of ground truth target map.

This research exploits a range of classification techniques and to implement them for assessing the effectiveness of hyperspectral classifications using various statistical scoring methods without the need of ground truth target map.

1.2 Contributions of this research

Classification of hyperspectral image has been an intensive research within the remote sensing community in the last decade, and most of the research performed so far has been the development of sophisticated classification techniques such as graph based Bayesian network and other neural or genetic clustering techniques. Most of the work involves only one or two classification techniques, and furthermore relatively few concerns with how the performance of the classifier is assessed particularly when the target map is not available, such as those commonly found in the air-borne or space-borne hyperspectral imaging (HSI) data sets.

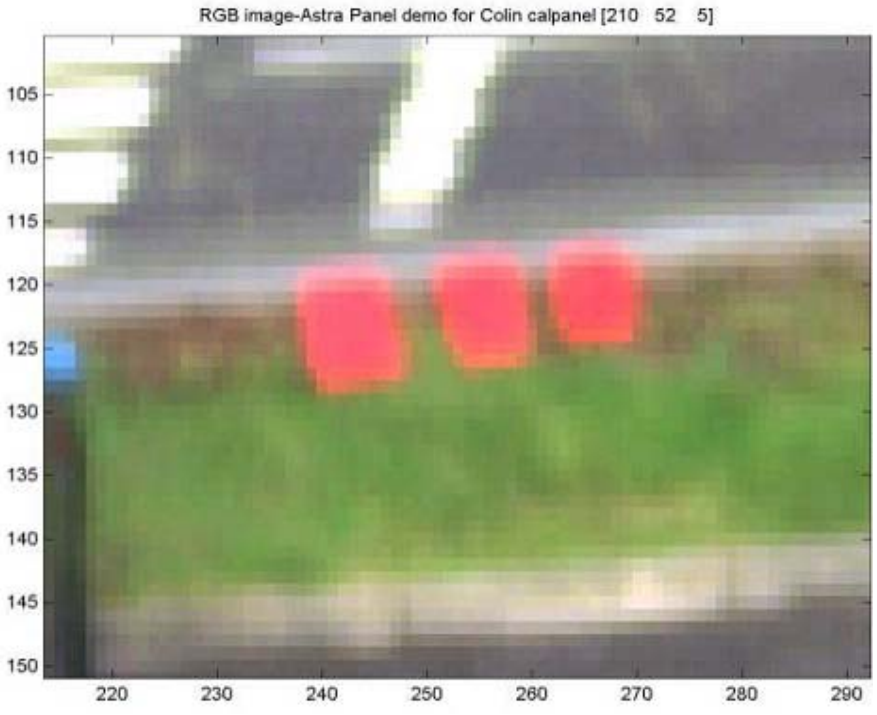
This study explores how the performance of classifiers can be better assessed and the contributions of this work have been:

1. An in-depth knowledge of machine learning theories.
2. A research which involves a range of classifiers for the classification of various hyperspectral image (HSI) data sets, which, have been collected and subsequently analysed during the course of this study.
3. The setting up of hyperspectral instruments involving both electro-optical hardware and camera control software developments.
4. A critical assessment of a range of statistical techniques to examine their usefulness as well as limitations for measuring the accuracy of classifiers with and without the use of target map.

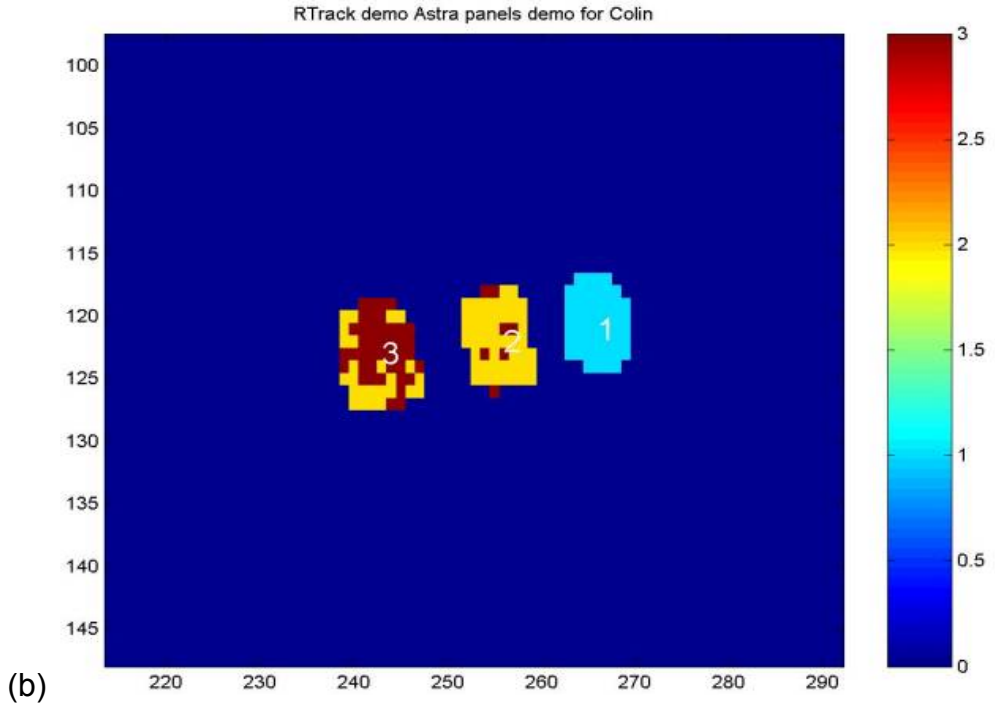
5. Through team work this study has contributed partially a number of publications in the area of hyperspectral imaging and machine visions.

1.3 Why hyperspectral imaging (HSI)?

Most machine vision research has involved 3-colour spectral bands (normally RGB) together with textural/temporal information for target classifications, but in many cases it has been found that the usefulness of this kind of technology is very limited. In scenarios such as targets in similar shape and colours in the RGB domain, such as the data that shown in Figure 1-1, conventional classification technique cannot distinguish visually identical objects like the 3 car panels of the same made (Astra) and colour (red) but with different ages. On the other hand, the use of a simple HSI classification technique can distinguish the two panels which are one year apart in ages (panels 1 & 2) and it even manages to separates the two panels (panels 2 & 3) which are only a few months different in ages.



(a)



(b)

Figure 1-1: Three look-alike Astra car panels which differ in ages. a) RGB image of the scene, b) False colour map of the classification result using simple HSI classification technique. (Yuen et al., 2005)

1.4 Future role of this study: Anti-terrorism and Homeland Security applications

Prior to the attacks of September 11, 2001, organised terrorist activities, such as Oklahoma City bombing, have always been the problems for many major cities and countries.. After the Sept 11 attack all government bodies have tried their best to tackle the problem by adding extra security measures, such as the implementation of additional more CCTVs around stations and airports. However, the effectiveness of these measures for anti-terrorism has remained to be a hot debate topic.

By increasing the number of CCTV not necessary improves the security effectiveness. In many cases, the police don't have the resources to cover the CCTV footage (Espiner, 2009) and therefore the efficiency of the surveillance through human operators on the CCTV system is highly questionable. Furthermore, it is a challenge to identify a subject from a crowd such as that shown in Figure 1-2. There is a real need to build an automatic surveillance system to improve counter-terrorism technology and it is hoped that this research will help to lead into a technique for realising a more robust surveillance system.

Figure 1-2: CCTV technology is not capable of identifying target from a crowd effectively.

2 Hyperspectral Imaging (HSI): an introduction

Hyperspectral imaging is a technique that generates data which consists of multiple spectral bands at each pixel location. Hyperspectral images can be thought of a collection of tens or hundreds of identical images but at different wavelength channels; these images are put together to form an image cube. Each pixel has its own spectral characteristic which can be viewed in the spectral space Figure 2-1.

The developments of pattern recognition and image processing techniques began to be seriously addressed since the advance in digital computer in 1960 (Landgrebe, 2002). The multi-spectral concept was originally proposed in earth observation remote sensing due to the cost of building high spatial resolution sensor in the space system. More advanced hyperspectral instruments with higher spectral resolution have been developed for remote sensing applications in the past decades. For example, the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) delivers data in 224 contiguous spectral channels spaced about 10nm apart from the spectral region from 0.4 to 2.45um.

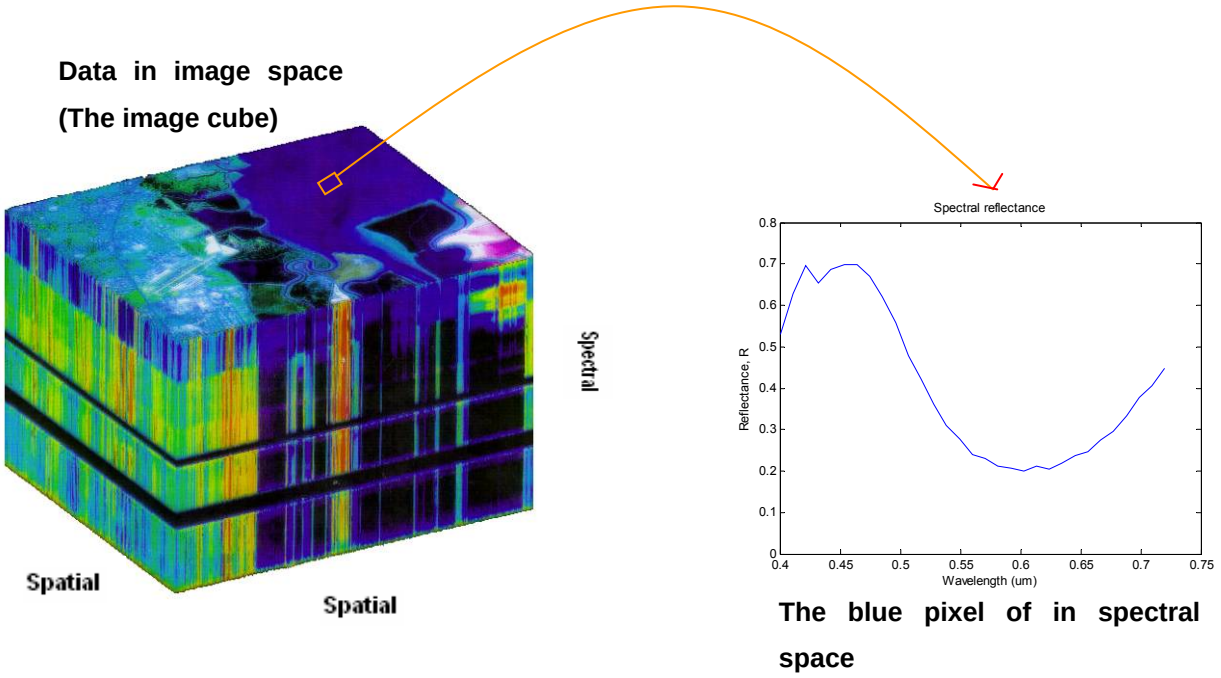


Figure 2-1: Sample of the hyperspectral image cube (Landgrebe, 2002)

2.1 Processing chain of hyperspectral Image (HSI) data

HSI classification begins with a raw digital image, normally with digital values, which passes through several processing steps before the classifier is applied. The general steps in HSI involve pre-processing (which includes sensor calibration and atmospheric compensation) and dimensional reductions to obtain usual data. Pre-processing plays a very important role in classification of hyperspectral image because of the large variants in the atmosphere condition and the sensor errors (Shaw and Burke, 2003; Richards and Jia, 2006). Although the spectrum of the solar radiation reaching the atmosphere is well characterised, the spectrum of the solar radiation reaching the ground is altered temporally and geographically dependent, because the solar radiation is propagating through the constantly changing atmosphere. Sensor errors, such as the focal-plane vibration, spatial and spectral aberrations, can further impede the recovery of the reflectance spectra by distorting and contaminating the raw imagery. Therefore the sensor must have low jitter and its geographical location must be recorded in real time. After calibration and correction to compensate the artefacts and gained variations in the sensor, low signal-to-noise ratio (SNR) band channels due to sensor noise are removed from the imagery.

2.2 Atmospheric Compensation

Atmospheric effects distort the image in a wavelength dependent fashion. **Absorption and scattering**: Before the light is reaching the ground, light is absorbed by gases, aerosols and water vapour. Further absorptions and scatterings occur after solar radiation is reflected by the target. **Upwelling Radiance**: Some solar radiation is scattered by the atmosphere into the field of view of the sensor without ever reaching the ground. **Secondary illumination**: Solar radiation is reflected by nearby objects before it illuminates the targets. **Adjacency effects**: Solar radiation is reflected by nearby objects and then scattered into the field of view of the sensor. Other environmental factors may also affect the images such as sun angle relative to zenith, sensor viewing angle, surface orientation of the target, cast shadows of clouds and ground cover.

The easiest correction method could be done by using information drawn from the image and averaging the relative reflection (Kruse et al., 1985), but the resulting images have not been very precise. The other method is done by creating radiative transfer (physical) models. Radiative transfer (physical) models, such as MODTRAN (Berk et al., 1998) or 6S (Vermote et al., 1997), are widely used and could be found in the public domain. Although these may be the preferred method, they require varying degrees of knowledge of the surface reflectance properties and the atmospheric conditions at the time the image was acquired . Those input parameters are often difficult to obtain.

2.2.1 Empirical line method (ELM)

Empirical Line Method (ELM) is a very popular alternative approach to radiative transfer models. The method assumes there is a linear relationship between raw digital values (or radiance) and reflectance spectra of the image. It also requires users to identify the reflectance of at least two homogeneous targets that are larger enough to be resolved (Karpouzli and Malthus, 2003). If both requirements are met, the conversion is simply done by calculating the gradients and offsets that convert digital values to reflectance for each spectral band. The reflectance conversions are considered valid only between the bright and dark target extremes and extrapolation outside this range is usually avoided (Baugh and Groeneveld, 2008).

The common way to implement the ELM is to deduce the slope and offset of the relationship between the radiance and the reflectance of several calibration panels in the scene. Once this relationship is established all other pixel values in the scene can then be ‘converted’ into reflectance and this is the method that has been adopted throughout in the data analysis of this work.

3 Overview of classification process

The procedure of hyperspectral image classification involves several basic steps: (1) Data acquisition, (2) Pre-processing, (3) Data presentation, (4) Decision making and (5) Performance evaluation. The issue dictates the choice of sensor, pre-processing technique, representation scheme, and the decision making process.

3.1 Data acquisition

Hyperspectral data consists of data gathered in more than one spectral band. The geometry of vector spaces changes continually as the dimensionality of the space increases. Nowadays, it is very normal that data to be analysed contains at least ten and perhaps as many as several hundred spectral bands. In hyperspectral images, both spatial and spectral resolutions contribute to the sample size, i.e. the data volume. It is desirable to gather information as much as possible but it is not feasible in practice; the main concerns are the cost and the rate of gathering data (Landgrebe, 2002; Shaw and Burke, 2003). High-resolution sensors would be very expensive and data transmission rate may be limited due to many factors. It is important to keep a correct balance between spatial and spectral resolutions. If the spatial resolution is too low, too many different materials may be mixed within a pixel. As a result, the image becomes meaningless even the spectral resolution is very high. On the other hand, for low spectral resolution, e.g. RGB image may not provide enough information for accurate classification, especially when the texture and shape of objects are similar to each other.

3.2 Pre-processing

Pre-processing plays a very important role in classification of hyperspectral image because of the large variants in the atmosphere condition and the sensor errors (Richards and Jia, 2006). Although the spectrum of the solar radiation reaching the atmosphere is well characterised, the spectrum of the solar radiation reaching the ground is altered temporally and geographically dependent because the solar radiation is propagating through the constantly changing atmosphere. Non-linear motion of the sensor can corrupt the spectral image by mixing the spectral together. Therefore the sensor must have low jitter and its geographical location must be recorded in real time.

After calibration and correction to compensate the artefacts and gained variations in the sensor, atmospheric correction is normally preformed.

Atmospheric effects distort the image by absorbing and scattering light in a wavelength dependent fashion. Before the light is reaching the ground, light is absorbed by gases, aerosols and water vapour. Some solar radiation is scattered by the atmosphere into the field of view of the sensor without ever reaching the ground. Further absorptions and scatterings occur after solar radiation is reflected by the target. Other environmental factors may also affect the images such as sun angle relative to zenith, sensor viewing angle, surface orientation of the target, cast shadows of clouds and ground cover and secondary illumination caused by nearby target. The correction could be done by using information drawn from the image and averaging the relative reflection; but the resulting images are not very precise. The other methods include creating empirical or physical models, but these require varying degrees of knowledge of the surface reflectance properties and the atmospheric conditions at the time the image was acquired (Beisl and Woodhouse, 2004).

3.3 Data presentation

Most of the distortions caused by the atmosphere should be corrected after the image is processed. Each pixel in a hyperspectral image contains a spectral profile which typically comprises hundreds of spectral bands. Hyperspectral imagery allows the detection and exploitation of narrow spectral features of target classes of interest, leading to an improved identification and discrimination of ground targets, and characterization of their related properties (Duda et al., 2000; Jain et al., 2000). However, the huge amount of data generated by hyperspectral systems may degrade the accuracy of classification result. There are no theoretical guidelines that suggest the appropriate patterns and features to use in specific situation (Marin et al., 1999). However, as Jain et al pointed out (Jain et al., 2000; Jain et al., 1999), a well defined feature extraction algorithm will lead to a compact pattern representation and yield significantly improved classification results.

3.4 Decision making process

The final step of the classification process is to organise patterns into groups. The choice of the decision making rule depends on the specific applications.

Provided there are enough training samples, supervised classification can normally outperform unsupervised algorithms. One of the most commonly used supervised classification techniques in hyperspectral imaging is probabilistic method. The probabilistic parametric techniques are based on Bayesian & maximum likelihood decision theory and require the estimation of its model parameters. The multivariate Gaussian density was the most popular density assumption (Duda et al., 2000); however it still plays a useful role in image classification (Chen and Peter Ho, 2008). Non-parametric classification such as K-nearest neighbour (KNN) (Duda et al., 2000) is popular methods. Unlike parametric classifiers, they do not require the estimation of its probability density function parameters.

Geometric techniques involve the use of decision boundaries to separate different classes. The use of Artificial Neural Networks (ANN) and Support Vector Machine (SVM) are very popular in the remote sensing community (Chen and Peter Ho, 2008). Although ANN was first invented by Frank Rosenblatt (Rosenblatt, 1958), it was not used in the remote sensing community since the first paper published in the early 1990's (Chen and Peter Ho, 2008). On the other hand, SVM is primarily a two-class classifier developed by Vapnik (Vapnik, 2000), which has drawn many attentions in the hyperspectral classification community because it achieves good performance in real world applications (Junping Zhang et al., 2001; Melgani and Bruzzone, 2004). The SVM method aims to find the optimal hyperplane, which is able to separate the input data into their respective classes. Melgani & Bruzzone (Melgani and Bruzzone, 2004) has compared SVM with two widely used classifiers, KNN and radial bias Functions (RBFs) neural network, and found that both linear and non-linear SVM out-performs KNN and RBF neural network in terms of classification accuracy.

If classification is done without the use of training sample sets, unsupervised algorithms are used. Unsupervised clustering is divided into hierarchical and partitional clustering. Hierarchical Clustering is a well-known unsupervised classification technique and its variant binary hierarchical classifier BHC (Kumar et al., 2002) has been found useful for

hyperspectral imaging in many literatures. The most classical partitional algorithm, k-mean clustering, has been proposed for many decades ago (MacQueen, 1966) but it is still widely used in many applications. Many variants of K-means have subsequently been proposed in recent years such as fuzzy c-mean (FCM) (Bezdek, 1981; Dunn, 1973) .

Different from the traditional clustering techniques, the Gaussian mixture modelling (GMM) approach provides a means of solving both simple and complex classification tasks as well as a way to substantiate results. Like supervised parametric technique, classification is done by estimating the density of each class but the class parameters are determined via the Expectation-Maximisation algorithm (EM), starting from the initial values selected systematically by the learning procedure.

There have been a lot of development to combine multiple classifiers for solving multi-class classification problem (Kittler, 1998; Ho et al., 1994).For example, SVM is a binary classifier. Therefore, in order to achieve multi-class classification, SVM type classifiers must be combined together.

3.5 Performance evaluation

Accuracy assessment is an important step to analyse and evaluate the quality and reliability of hyperspectral data. Assessments are divided into site & non-site specific type. We also propose a method when ground truth is not presented for accuracy assessment.

4 Classification methods: a literature survey

4.1 Supervised Classification

Supervised classification can be divided into probabilistic based and geometric based. Probabilistic approaches mainly involve finding density estimates of each class and classification is done based on those estimations. Density estimation can be sub-divided into parametric and non-parametric types of techniques. Geometrical methods are based on finding decision boundaries that can separated between different classes. Most pattern recognition methods are based on feature vectors and classifications are done by calculating similarity or distance within each category.

A training signature obtained from the parametric method can be critically dependent upon the parameters and entities of statistics underlying the data set, such as the covariance matrix and the mean of those coordinates of pixels that are contained in the array or bunch of the training sample. The following featured characteristics are also included in the signature of training that is obtained by the parametric method in addition to the standard featured characteristics of the training sets:

- Number of spectral bands in the image that need to be processed (as entertained by the program of training).
- The maximum and minimum values of data set in each and every spectral band for every bunch of training sample (maximum vector and the minimum vector).
- the mean value of data file in every spectral band for every cluster of training sample (called mean vector)
- For every group of training sets; the covariance matrix.
- Pixels quantity in the cluster of training sample.

The classification method of non parametric allots pixels to the class according to their location by the utilisation of the signatures that are obtained from non-parametric classifier, either outside the area or inside the area in the feature space image. The choice of the decision making rule depends on the specific applications.

Provided there are enough training samples, supervised classification can normally outperform unsupervised algorithms. One of the most commonly used supervised classification techniques in hyperspectral imaging is probabilistic method. The probabilistic parametric techniques using Bayesian & maximum likelihood decision theory require the estimation of model parameters, such as the multivariate Gaussian density function.

Non-parametric classification such as K-nearest neighbour (KNN) (Duda et al., 2000) and parallelepipeds are very common in the remote sensing community. K-NN categorises a sample which closest to the Kth nearest neighbour. Each class of the parallelepiped classifier is implemented finding the upper and lower bounds of each feature from the training data, pixel that is within such a parallelepiped are labelled to that class. Unlike parametric classifiers, they do not require the estimation of its probability density function parameters.

4.1.1 Parametric classification

Parametric classification has been one of the most commonly employed techniques for hyperspectral applications. This type of classifier is based on the statistical probability distributions for each class.

Let's assume that there are L classes, $w_i, i = 1,...,L$, in a multivariate mixture model. To determine the class in which a pixel x belong to, one must know the observation-conditional probabilities, $p(w_i \mid x)$, the probability of class w_i given by the observation x,. Classification is performed by finding the class with maximum conditional probability:

$$x \in w_i, \qquad p(w_i \mid x) > p(w_j \mid x) \forall j \neq i \qquad [4\text{-}1]$$

However, in practice these observation-conditional probability functions are often unknown.

Suppose the training data $x_1,...,x_n$ are sufficient enough for an accurate estimation, one can then estimate its probability distributions in each class. The probability of finding x for each class is given by $p(x \mid w_i)$. The probabilities can be derived by using the Bayes's theorem,

$$p(w_i \mid x) = p(x \mid w_i)p(w_i)/p(x) \quad [4\text{-}2]$$

and the data probability function is given by

$$p(x) = \sum_{i=1}^{K} p(x \mid w_i)p(w_i) \quad [4\text{-}3]$$

where $p(w_i \mid x)$ is now known as the posterior probability and $p(w_i)$ is known as the prior probability. The prior probability for each class occurring is $(0 < p(w_i) < 1)$ and for $i = 1,...,K$, the total prior probability is equal to $\sum_{i=1}^{K} p(w_i) = 1$.

The classification rule of equation [4-1] is now given by:

$x \in w_i$, $\quad p(x \mid w_i)p(w_i) > p(x \mid w_j)p(w_j) \forall j \neq i$ with the common factor $p(x)$ removed.

Since the logarithm is monotonically increasing, for mathematical convenience the probability terms can be changed to:

$$g_i(x) = \ln\{p(x \mid w_i)p(w_i)\}$$

$$= \ln p(x \mid w_i) + \ln p(w_i) \quad [4\text{-}4]$$

where $g_i(x)$ is sometimes known as the discriminant function and the classification rules of equation [4-1]becomes

$$x \in w_i, \quad g_i(x) > g_j(x) \forall j \neq i \quad [4\text{-}5]$$

4.1.1.1 Maximum likelihood classifier

In the case of Gaussian density with N bands, the parameter for each class θ_i denotes mean m_i and covariance matrix $\sum_i$, $\theta_i = (m_i, \sum_i)$. The likelihood probability is defined by:

$$p(x \mid w_i) = \frac{1}{(2\pi)^{N/2} |\sum_i|^{1/2}} \exp\left\{-\frac{1}{2}(x_i - m_i)^T \sum_i^{-1} (x_i - m_i)\right\} \quad [4\text{-}6]$$

The logarithmic form of the discriminant function becomes:

$$g_i(x) = \ln p(w_i) - \frac{N}{2}\ln(2\pi) - \frac{1}{2}|\sum_i| - \frac{1}{2}(x_i - m_i)^T \sum_i^{-1} (x_i - m_i) \quad [4\text{-}7]$$

Since the $-\frac{N}{2}\ln(2\pi)$ term is constant for all $g_i(x)$, it can be removed to simplify the calculation. Often, there is no useful information about the prior probability and equal prior probability is assumed. By removing all the unnecessary constant terms, the final discriminant function can be refined:

$g_i(x) = -|\sum_i| - (x_i - m_i)^T \sum_i^{-1}(x_i - m_i)$ or

$$g_i'(x) = |\sum_i| + (x_i - m_i)^T \sum_i^{-1}(x_i - m_i) \qquad [4\text{-}8]$$

where

$x \in w_i, \qquad g_i'(x) < g_j'(x) \forall j \neq i$

This is sometimes known as the maximum likelihood classifier, log-likelihood classifier or quadratic (Gaussian) classifier.

4.1.1.2 Mahalanobis Distance classifier

If we assume that the covariance $\sum_i$ for all classes are equal i.e $\sum_i = \sum$ for all i, the determined of the covariance is constant and can be ignored. The discriminant function becomes

$$fd_i(x) = (x_i - m_i)^T \sum^{-1}(x_i - m_i) \qquad [4\text{-}9]$$

This is known as the Mahalanobis Distance classifier or Fisher Linear Discriminant classifier. A pattern is classified by finding the minimum distance from the normalised mean.

4.1.1.3 Euclidean distance classifier

Consider the covariance matrices of all classes to be diagonal and equal, and the variances in each component to be identical, therefore $\sum_i = \sigma^2 I$. The logarithmic form of the original log-likelihood discriminant function becomes

$$g_i(x) = \ln p(w_i) - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\sigma^{2N} - \frac{1}{2}(x_i - m_i)^T \sigma^{-2}(x_i - m_i) \qquad [4\text{-}10]$$

Again we assume the prior probabilities are equal and remove all the constant terms, the discriminant function becomes

$g_i(x) = -(x_i - m_i)^T (x_i - m_i)$ or

$$d_i(x) = (x_i - m_i)^T (x_i - m_i) \quad [4\text{-}11]$$

Here, we are trying to find the minimum $d_i(x)$ which is called the Euclidean distance. Therefore this type of classifier is called the Euclidean distance classifier or minimum distance classifier.

4.1.1.4 Regularised discriminant analysis & leave-one-out covariance estimations

There are many methods previously employed for the estimation of sample covariance in the small sample size situations. RDA (Regularized Discriminant Analysis) has been one of the most commonly used techniques particularly in face recognition where the training sample is small compared to the large dimensions of features. Instead of simply estimating the covariance S from the training sample, RDA estimates $(S + \gamma I)$ where γ is the regularisation parameter and I is the identity matrix.

Consider a D dimension data set which contains L classes $\{X_i\}_{i=1}^{k}$ and each class is comprised of a number of samples $X_i = \{x_{ij}\}_{j=1}^{n_i}$ making up a total of $N = \sum_{i=1}^{L} n_i$ training samples. Thus, the estimated covariance $\hat{\Sigma}_i$ can be given by:

$$\hat{\Sigma}_i(\lambda, \gamma) = (1-\gamma)\hat{\Sigma}_i(\lambda) + \frac{\gamma}{D} tr[\hat{\Sigma}_i(\lambda)] I \quad [4\text{-}12]$$

where $\hat{\Sigma}_i(\lambda) = \frac{n_i}{W_i(\lambda)}[(1-\lambda)\Sigma_i + \lambda S]$

and

$$\hat{\Sigma}_i(\lambda) = \frac{n_i}{W_i(\lambda)}[(1-\lambda)\Sigma_i + \lambda S]$$

$$W_i(\lambda) = (1-\lambda)n_i + \lambda \sum_{i=1}^{k} n_i$$

and Σ_i is the covariance directly evaluated from the small training sample set.

The parameters $\lambda(0 \le \lambda \le 1)$ and $\gamma(0 \le \gamma \le 1)$ handle the contractions of the Σ_i in the directions of the class variance and the multiples of identity matrix respectively and both can be deduced from the eigen matrix of the data set.(Hayden and Twede, 2002)

In theory the minimum number of samples required for a fully characterised D-dimensional data set is D+1 samples, but the Leave-one-out covariance (LOOC) method can achieve this by using a minimum of as few as three (Hoffbeck and Landgrebe, 1996).

Instead of having the multiple identity matrix common covariances like that in the RDA, LOOC uses a mixing parameter for the selection of an appropriate mixture of the common covariance, sample covariance, diagonal sample covariance, and the diagonal common covariance:

$$\hat{\Sigma}_i(\alpha_i) = \begin{cases} (1-\alpha_i)diag(\Sigma_i) + \alpha_i\Sigma_i & 0 \le \alpha_i \le 1 \\ (2-\alpha_i)\Sigma_i + (\alpha_i - 1)S & 1 < \alpha_i \le 2 \\ (3-\alpha_i)S + (\alpha_i - 2)diag(S) & 2 < \alpha_i \le 3 \end{cases} \quad [4\text{-}13]$$

where S is known as the common covariance and it is evaluated from the weighted sum of Σ_i:

$$S = \frac{1}{N}\sum_{i=1}^{L}(n_i\Sigma_i) \quad [4\text{-}14]$$

where L is the total number of classes and ni is the number of pixel in class i and N is the total number of pixel.

The value of the mixing parameter α_i is selected so that a best fit to the training samples is achieved, in the sense that the average likelihood of the omitted samples is maximised. The average leave-one-out likelihood (LOOL) is given by:

$$LOOL_i(\alpha_i) = \frac{1}{n_i}\sum_{j=1}^{n_i}\ln\left[f\left(x_{i,j} \mid m_{i/k}, \hat{\Sigma}_{i/k}(\alpha_i)\right)\right] \quad [4\text{-}15]$$

where f() is the maximum likelihood function as given in equations [4-7]

The mean of class i without sample k is $m_{i/k} = \frac{1}{N_i - 1}\sum_{\substack{j=1 \\ j\neq k}}^{N_i} x_{i,j}$, where the notation i/k indicates the mean is computed without sample k from class i, and j is the pixel sample from class i. Similarly, the sample class covariance matrix of class i without sample k is:

$$\Sigma_{i/k} = \frac{1}{N_i - 2}\sum_{\substack{j=1 \\ j\neq k}}^{N_i} \left(x_{i,j} - m_{i/k}\right)^T \left(x_{i,j} - m_{i/k}\right)$$

Both the Σ_i and the covariance that estimated through the $\hat{\Sigma}_i(\lambda,\gamma)^{RDA}$ or $\hat{\Sigma}_i(\alpha_i)^{LooC}$ are used for the parametric classifiers. The LOOC is implemented firstly by removing a sample from the training set, and the mean of the remaining samples and their covariance matrices are then evaluated. Subsequently the likelihood of the remaining samples is calculated according to equation 4-13, producing the estimated covariance matrix $\hat{\Sigma}_i$ and the mean. This is then repeated until every sample is deleted. The mixing parameter is chosen when a maximum average likelihood is attained.

4.1.2 Non-parametric classification

The problem of using the parametric modelling techniques is that one must make an assumption of the parametric forms of the probability density function. For example, the Gaussian multivariate distribution is assumed before the parameters m_k , $\sum_k$ are estimated using the maximum likelihood method. In the case of unknown density function, non-parametric classifiers can be used to estimate the probability density function.

4.1.2.1 K-Nearest Neighbours classifier

In the K-NN rule, the class of the input pattern X is chosen as the class of the majority of its K nearest neighbours. The key idea of nearest neighbour algorithms is that any particular input data z and its neighbours are likely to share the same properties. The neighbours of z are defined by some distance metric. A distance metric is a scalar measurement of the distance between two points. In KNN, the neighbours z_j of z_i are the K data points with the smallest distance metric. The value of K is chosen to be big enough to ensure a meaningful estimate. There are many different methods of

computing the distances between two points [see Appendix for more information], the most common method is the Euclidean distance.

Suppose we have n labelled training samples in D dimensions, and it is seek to find the closest to a test point x (K = 1). The easiest approach is to inspect each training data point in turn, calculate its Euclidean distance to x. The test point x is then labelled to the class of that training sample that is currently closest to it.

The performance of K-NN classifier in finite design sample case significantly depends on the number K of nearest neighbours.

4.1.2.2 Parallelepiped Classification

The parallelepiped classifier is one of the simplest forms of supervised classifiers (Richards and Jia, 2006). The multidimensional box or parallelepiped for each class is found by inspecting the histogram of the training data for each class. The decision rule is form by finding the upper and lower limits of each class for all bands. A modified version of parallelepiped classifier is to find the mean and the variance of each class for all bands. This type of classifier is simple to train and use, but it suffers from two main drawbacks. If one pixel is in a region that no parallelepipeds cover, that the pixel is unclassified. Furthermore, parallelepipeds are often overlapping to each other if data is correlated, therefore some data will be assigned to more than one class. These factors are illustrated in Figure 4-1. The red and green rectangular boxes represent the parallelepipeds of class R (red) and class G (green). Any pixels that lie within both of the parallelepipeds are classified to the two classes; any pixels that lie beyond both of them are unclassified.

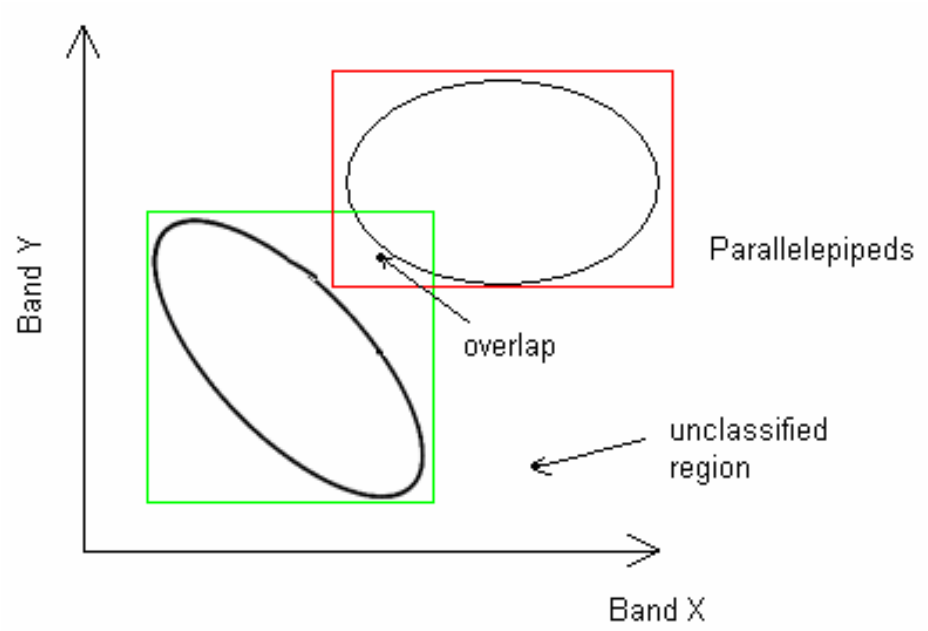


Figure 4-1: Example of 2-dimensional two classes' problem using parallelepiped method

4.1.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is based on one simple concept: it discriminates two classes by fitting an optimal separating hyperplane to the training samples of two classes in a multidimensional feature space (Waske and Benediktsson, 2007). Let us consider a supervised binary classification problem. Let us assume that the training set consists of N vectors from the d-dimensional feature space $x_i \in \Re^d \left(i = 1,2,\ldots,N\right)$. A target $y_i \in \{-1,+1\}$ is associated to each vector x_i. Let us assume that the two classes are linearly separable. This means that it is possible to find at least one hyperplane (linear surface) that can separate the two classes without errors. When the points x lies on the hyperplane, the hyperplane must satisfy

$$x \cdot w + b = 0 \qquad [4\text{-}16]$$

where the vector w is normal to the hyperplane, $|b|/\|w\|$ is the perpendicular distance from the hyperplane to the origin, and $\|w\|$ is the Euclidean norm of w. The SVM approach consists in finding the optimal hyperplane that maximises the distance d_+ (d_-) between the closest positive (negative) training sample and the separating hyperplane.

Let's define the margin of a separating hyperplane to be d_++d-. For the linearly separable case, the support vector algorithm simply looks for the separating hyperplane with largest margin. This can be formulated as follows: suppose that all the training data satisfy the following constraints:

$$x_i \cdot w + b \geq +1, \text{ for } y_i = +1 \quad [4\text{-}17]$$

$$x_i \cdot w + b \geq -1, \text{ for } y_i = -1 \quad [4\text{-}18]$$

These can be combined into one set of inequalities:

$$y_i\left(x_i \cdot w + b\right) - 1 \geq 0 \ \ \forall i \quad [4\text{-}19]$$

In order to find the find the optimal hyperplane, the margin of support vectors $\|w\|^{-1}$ needs to be maximised as shown in Figure 4-2. It is convenient to replace maximisation of $\|w\|^{-1}$ with minimisation $\frac{1}{2}\|w\|^2$ and the optimisation problem becomes:

Choose w,b to minimize $\frac{1}{2}\|w\|^2$ [4-20]

Subject to $y_i\left(x_i \cdot w + b\right) \geq 1 \ \ \forall i$

The above linearly constrained optimisation expression can be switched to the following dual problem representation using Lagrangian multipliers:

maximise: $\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \left(x_i^T x_j\right)$ [4-21]

subject to: $\alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

where the weight vector is terms of the training sets:

$$w = \sum_i \alpha_i y_i x_i \quad [4\text{-}22]$$

In the case where there exists no hyperplane that can separate between two classes, e.g. two overlapping distributed classes, soft margin method could choose the hyperplane that split the classes as clear as possible (Cortes and Vapnik, 1995). The solution of the optimisation problem becomes:

Choose w,b to minimize $\frac{1}{2}\|w\|^2 + C\sum_{i=`}^{N}\xi_i$ [4-23]

Subject to $y_i\left(x_i \cdot w + b\right) \geq 1 - \xi_i$, $\xi_i \geq 0. \ \forall i$

where ξ_i is the slack variables which measure the degree of misclassification and C is the cost parameter that determines the trade of between the maximisation of the margin and the minimisation of the degree of misclassification.

One of the advantages of SVM method is its ability to prevent over-fitting of the data by controlling the margin measures (Jain et al., 2000; Chen and Peter Ho, 2008). Furthermore, SVM algorithm can find the optimal separating hyperplane in a high dimensional space via the kernel trick (Boser et al., 1992). It is especially suitable to problems when classes are not linearly separable. The training vectors x_i are mapped into a higher dimensional space by replacing $\left(x_i^T x_j\right)$ with the kernel function $K\left(x_i^T x_j\right) \equiv \phi\left(x_i\right)^T \phi\left(x_j\right)$. The kernel functions include linear, polynomial, radial bias Function (RBF) and sigmoid:

$$\phi = \begin{Bmatrix} x_i * x_j & Linear \\ \left(x_i * x_j\right)^p & Polynomial \\ \exp\left(-n\left|x_i - x_j\right|^2\right) & RBF \\ \tanh\left(kx_i \cdot x_j + coefficient\right) & Sigmoid \end{Bmatrix} \quad [4\text{-}24]$$

The RBF has been the most popular choice of kernel types used in SVM models for hyperspectral application and many authors have employed SVM for the classification of hyperspectral images (Junping Zhang et al., 2001; Melgani and Bruzzone, 2004; Pal and Mather, 2004)

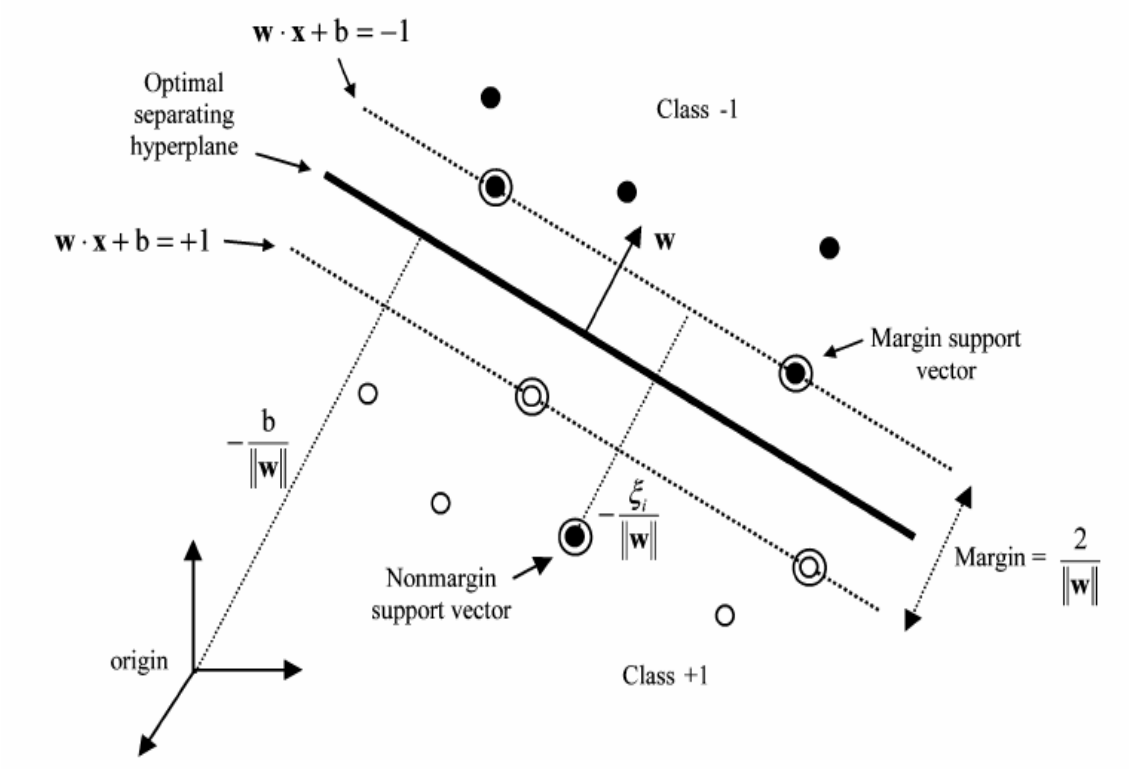


Figure 4-2: Support Vector Machine: The two classes of +1 and -1 are separated by the optimal hyperplane, and the support vectors are denoted with an extra circle. (Melgani and Bruzzone, 2004)

4.1.3.1 SVM Implementation

As outlined in previous section the support vector machine (SVM) belongs to a kind of binary classifiers that finds the best separation plane between two classes. For multi-class classifications, SVM can be deployed using multiple binary modules, commonly in a one against one or one-against-all manner (Melgani and Bruzzone, 2004). One against one involves the building up of one SVM for each pair of classes and the best classification is then chosen by voting. One against all classification method involves divide and conquer method in which one SVM is trained per class, with an objective to distinguish the pixels in a single class from the rest of the classes.

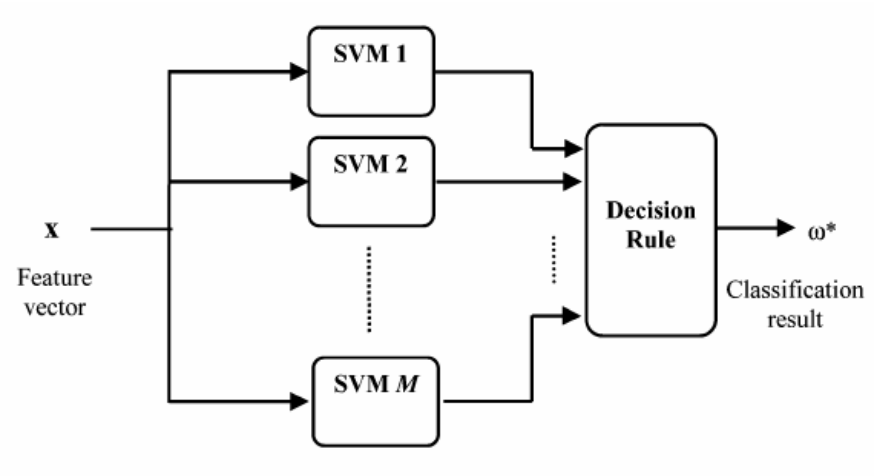


Figure 4-3: A typical parallel strategy for one vs one SVM implementation ((Melgani and Bruzzone, 2004))

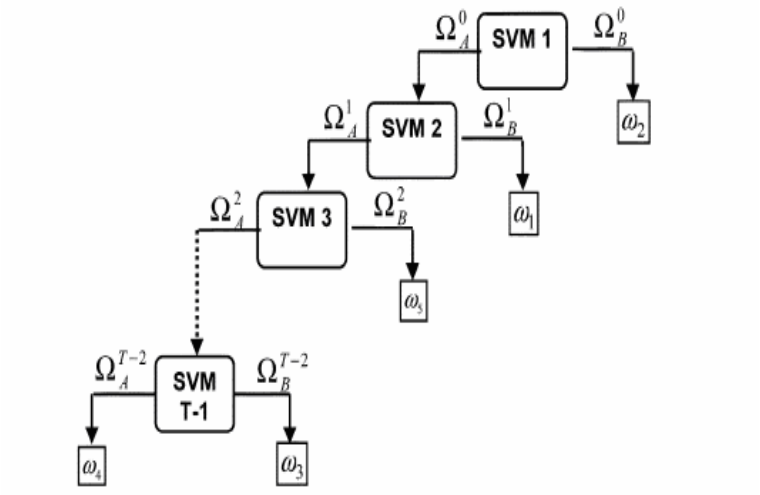


Figure 4-4: A typical cascading approach for one vs all SVM implementation ((Melgani and Bruzzone, 2004))

4.2 Unsupervised data clustering techniques

Patterns within a cluster are similar to each other and individuals from the same clusters should be dissimilar from those in other clusters. However, non-predictive clustering is a subjective process in nature and the result of classification depends on the methods for representing and grouping data. In the case of chemical mixture, one may group them by the colouring of the mixture but others may group them by the reactivity level of the mixtures. As (Xu and Wunsch, 2005) mentioned, most researchers in the literature describe a cluster by considering the internal homogeneity and the external separation. Both similarity and dissimilarity should be examinable in a clear and meaningful way.

Clustering is very useful in many pattern recognitions problems, such as remote sensing, one may not be able to obtain the ground truth information. Clustering techniques can be roughly divided into either hierarchical or partitional. In hierarchical clustering, data are partitioned in a series of steps from a cluster including all individuals into k clusters or vice versa, while partitional clustering separate data into k clusters in one step.

4.2.1 Hierarchical Clustering

Hierarchical clustering can be sub-divided into two main streams: Agglomerative or Divisive. Agglomerative method starts with n clusters and each cluster contains only one data, then a series of merge operations of clusters are performed based on the proximity (similarity) matrix until the desire amount of clusters are produced. Divisive method on the other hand works in an opposite way. The entire data set are treated as a single cluster at the beginning and the cluster are split in sequence into smaller clusters based on the dissimilarity until a criterion is met. The result of hierarchical clustering in a tree is known as dendrogram (Jain et al., 1999) which illustrates the processing of both agglomerative and divisive clustering.

Hierarchical agglomerative methods are more commonly used in practice because of the computation complexity of divisive algorithm (Jain et al., 1999). In general, most of the hierarchical algorithms are variants of simple linkage, complete linkage or minimum-variances method. They can be constructed by choosing appropriate coefficients in the formula. In simple linkage clustering, the minimum linkage distance of the samples data within clusters are measured and clusters are merged with the shortest distance. In complete linkage clustering, the maximum linkage distances of the samples data within clusters are measured and clusters are merged with the shortest distance.

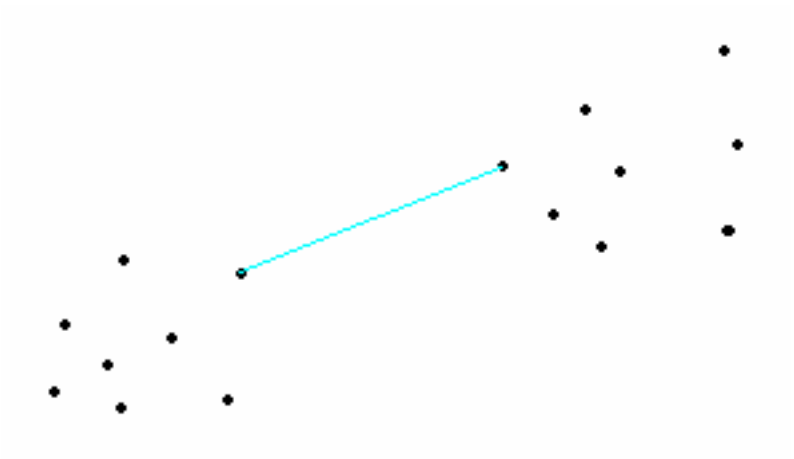


Figure 4-5: Simple Linkage Clustering

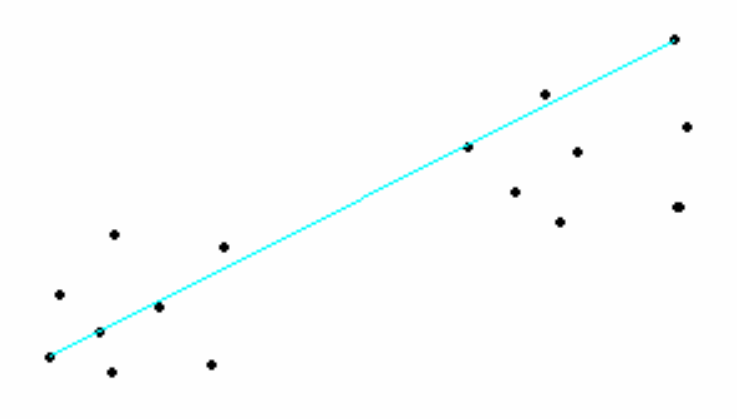


Figure 4-6: Complete Linkage Clustering

Typical hierarchical clustering algorithm is more flexible than partitional algorithms. For example, the simple linkage algorithm is able to detect elongated and irregular clusters during clustering whereas typical partitional algorithm only works well on isotropic clusters (Jain et al., 1999). One of the main disadvantages of hierarchical clustering algorithms is lack of robustness. For example, simple linkage ignored the tails of distribution whereas complete linkage can be strongly distorted by outliers such as noises (Jain et al., 1999). Once an object is assigned to a cluster, it will not be considered again which means the algorithms are not able to amend any previous misclassification. The time and space complexities are typically higher than partitional algorithms, therefore hierarchical clustering is rarely used for hyperspectral application because of the large data size and the high dimensionality. Nevertheless, the idea of hierarchical architectures has been incorporated in many other classification decision

rules such as SVM (Melgani and Bruzzone, 2004), and binary tree classifiers (Kittler, 1998; Ho et al., 1994).

4.2.2 Computation complexity for optimal partitional clustering

The basic methodology of partitional clustering is to assign a set of data into k clusters based on some criterions without hierarchical structure. In theory, the optimal partition results can be found by trying all the possible combinations, however, it is not practical due to the time complexity. In order to search all the possible combinations, the formula (Xu and Wunsch, 2005) is given

$$S(n,k)=\frac{1}{k!}\sum_{j=1}^{k}(-1)^{k-j}\binom{k}{j}j^{n} \qquad [4\text{-}25]$$

Suppose n=60 objects and k=3 clusters. It requires more than 10^{25} partitions to find all the possible results. Therefore many algorithms have been proposed for the past two decades in order to minimize the time cost but retain the accuracies.

4.2.3 Square-Error Clustering - K-means, ISODATA

The most commonly used criterion function in partitional clustering is the squared-error criterions. Suppose we have a set of n patterns in d-dimensional and we want to group them into K clusters $\{C_1, C_2, ..., C_k\}$. The sum of squared error criterion is defined as

$$E^2(D,M)=\sum_{i=1}^{K}\sum_{j=1}^{N}o_{ij}\left\|x_j-m_i\right\|^2 \qquad [4\text{-}26]$$

where

$$o_{ij}=\begin{cases}1 & \text{if } x_j \in \text{cluster j}\\ 0 & \text{otherwise}\end{cases} \quad \& \quad \sum_{i=1}^{K}oij=1 \ \forall \text{j}$$

$M=[m_1,...,m_k]$ is the mean or centroid vector of the cluster and m_i is the sample mean of the i^{th} cluster.

D = the partition matrix

The objective of the method is to partition the pattern set into K clusters such that the sum of square-error is as small as possible. The **K-means algorithm**, originally

proposed by McQueen (MacQueen, 1966), is the best-known square-error based algorithm. The meta code of the algorithm is as follows:

1. Choose the number of clusters K and then assign the mean vector M randomly or pick K patterns from the set randomly.
2. Assign each pattern in the data set to the nearest cluster based on the Euclidean distance between the pattern and the cluster centroid.
3. Recalculate the mean vector M from the current partition.
4. Repeat step 2-3 until convergence is achieved, i.e., all patterns do not change the cluster membership or minimal decrease in squared error.

Although K-means algorithm is simple to implement and the time complexity is low, there are several drawbacks. It can work very well for compact and hyperspherical clusters but not if the clusters are non-isotropic or hyperellipsoidal clusters (Jain et al., 1999). One of the major problems with K-mean algorithm is that it is sensitive to the selection of the initial partition and number of clusters K. Despite many authors had proposed different methods to select a good initial partition, there is no efficient and universal method to identify the initial partition and the number of cluster (Fraley and Raftery, 1998). The general technique is to run the algorithm many times with different K and initial centroids. Another problem is that it cannot guarantee convergence to the global minimum value.

There are various techniques to improve the K-means algorithm. The well-known iterative self-organising data analysis algorithm (**ISODATA**) (Ball and Hall, 1965) employs the ideal to split and merge clusters during each iteration. A cluster is split if the variance is above a pre-defined threshold T1 and two clusters are merged together if the distance between their centroids is below the threshold T2. Provided that T1 and T2 are carefully chosen, this technique is able to achieve optimum partition starting with an arbitrary initial centroid number. However the biggest problem with ISODATA (Appendix 13.4) is the introduction of more unknown parameters, such as the sample threshold, variance threshold and etc, which require the knowledge and experience of the user to choose the optimal parameters.

4.2.4 Fuzzy Clustering

The methods that are mentioned in the previous sections are all hard clustering method which means each pattern belongs to one and only one cluster. However in many data sets, there may not be clear boundaries between clusters, for example, there may even be several classes within the sub-pixels due to the spatial resolution of the image in hyperspectral data. Fuzzy clustering helps to relax the one pixel one class constraint by introducing the notation U_{ij} to represent the degree of membership for each class. The membership function U can be interpreted in this form :

For hard clustering

$$U_{ij} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

For soft clustering

$$U_{ij} = \begin{pmatrix} 0.6 & 0 & 0.4 & 0 \\ 0.1 & 0.8 & 0 & 0.1 \\ 0.99 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where each row =1 sample and each column =1 class and $\sum_{j=1}^{n} U_{ij} = 1, \forall i$. Using the membership matrix, one can find the optimal cluster by minimizing that value of a fuzzy criterion function. **Fuzzy c-mean algorithm** (FCM) (Bezdek, 1981) is the most popular fuzzy cluster algorithm, which attempts to find a partition for a set of data by minimizing the weighted squared error function

$$E^2(U, M) = \sum_{i=1}^{N} \sum_{j=1}^{K} (u_{ij})^p \left\| x_i - m_j \right\|^2 \qquad [4\text{-}27]$$

where

$M = [m_1, ..., m_k]$ is the mean or centroid matrix of the cluster and m_i is the sample mean of the i^{th} cluster.

$U=[u_{ij}]_{N*K}$ is the N*K fuzzy partition matrix

$p \in [1, \infty]$ is the fuzzy exponent and is usually set to 2

The fuzzy c-mean algorithm is

1. Randomly initialise the membership matrix U and select appropriate value for the stopping threshold e
2. Calculate the centroid matrix M using the formula

$$m_j = \frac{\sum_{i=1}^{N} \left((u_{ij})^p \cdot x_i \right)}{\sum_{i=1}^{N} (u_{ij})^p} \qquad [4\text{-}28]$$

3. Update the membership matrix U'

$$U' = \frac{1}{\sum_{y=1}^{K} \frac{\left\| x_i - m_j \right\|^{\frac{2}{p-1}}}{\left\| x_i - m_y \right\|}} \qquad [4\text{-}29]$$

4. Calculate $T = \left\| U' - U \right\| - e$ and set U=U'. If T<0 then STOP, else go back to step 2

Although fuzzy c-mean algorithm is better than hard k-mean algorithm at avoiding the local minima, it can still converge to local minima of the squared error. Also FCM suffers the same problems as encountered in k-mean such as the choice of the initial partition and sensitive to noise and outliers.

4.2.5 Neural Networks-Based Clustering

Artificial neural network (ANN) is built with the use of computer model and mathematics to mimic the actual biological nervous systems. Most of the ANNs need a 'teacher' to train the network, and therefore they are not useful for unsupervised classification. For clustering application, neural network-based algorithms are mainly based on **Self-organizing map** (SOM) (Kohonen, 1998).

The self-organizing map (SOM) model is based on the unsupervised learning of the neurons organized in a regular lattice structure. The topology of the lattice is triangular, rectangular or hexagonal. The objective of SOM is to allow visualization of high-dimensional patterns by representing them in a two-dimensional lattice structure. It can be achieved by grouping similar patterns and representing them by a neuron. The architecture of SOM is normally a simple single-layer network. Each input pattern is connected to all the output neurons and the weights between the input nodes and the

output nodes are changed during the learning process. The basic SOM training process is in the following steps.

1. Define the topology of SOM, e.g. hexagonal; Initial the reference vector mi randomly for each neuron i.
2. Select an input pattern x and compare it with the entire reference vector. Compute the distance using any types of metrics; Euclidean distance is normally used. Find the best matching unit (BMU) node c, i.e.

$$c = \arg_i \min\left\{\left\|x - m_i\right\|\right\} \quad \text{[4-30]}$$

3. Update the reference vector using

$$m_i(t+1) = m_i(t) + h_{ci}(t)\left[x - m_i(t)\right] \quad \text{[4-31]}$$

where the integer t=0,1,2… . The neighbour function $h_{ci}(t)$ is a smoothing kernel and $h_{ci}(t) \to 0$ when $t \to \infty$. The function can often be defined in two simple ways.

$$h_{ci}(t) = \begin{cases} \alpha(t) & \text{if } i \in N_c(t) \\ 0 & \text{if } i \notin N_c(t) \end{cases} \quad \text{[4-32]}$$

or in terms of Gaussian function

$$h_{ci}(t) = \alpha(t) \cdot \exp\left(-\frac{\left\|r_c - r_i\right\|^2}{2\sigma^2(t)}\right) \quad \text{[4-33]}$$

Nc(t) is the neighbourhood of the node c. The neighbourhood can be interpreted as concentric hexagons around the winner node c in case of a hexagonal neuron lattice. Both the learning-rate factor $\alpha(t)$ and the width of the kernel $\sigma(t)$ are monotonically decreasing function and the value $\alpha(t)$ must be bound between 0 and 1. r represents the location vectors of nodes c & i and $\left\|r_c - r_i\right\|$ increases as $h_{ci}(t) \to 0$.

4. Repeat step 2-3 until changes to weights fall below a pre-set threshold value.

SOM gives good approximation two-dimensional maps from multi-dimensional data and has been successfully use for many applications (Kohonen, 1998), but one major drawback is the quality of the result depends on the choice of parameters. Like k-mean

algorithm, SOM has to predefine the number of neurons, i.e. the number of clusters, for classification. The rate of convergence is depended on the learning rate and the neighbourhood function of the BMU. Once the SOM is trained, classification is done by labelling test samples to its closest neuron.

4.2.6 Mixture Model-Based Algorithm

Suppose data are generated by a mixture of several probability distributions and data in different cluster are extracted from different probability distributions, e.g., mixture of multivariate Gaussian. If the distributions are known, one can find the clusters by estimating the parameters of the underlying distributions.

Let's refer back the density function

$$p(x) = \sum_{i=1}^{K} p(x \mid w_i) p(w_i) \qquad [4\text{-}34]$$

The prior probability $p(w_i)$ is constant, α_i for each class and the likelihood $p(x \mid w_i)$ can be thought as a function that is dependent on a parameter θ_i and n observation $X = \{x_1, ..., x_n\}$. The mixture distribution probability function can be re-written as:

$$p(x) = \sum_{i=1}^{K} \alpha_i f(x, \theta_i) \qquad [4\text{-}35]$$

The next issue is to estimate the parameter θ_i of the model. One way to solve this problem is to apply the maximum likelihood estimation technique. This may be obtain by maximising $\prod_{j=1}^{n} p(x_j)$ with respect to θ_i and α_i under the constraint that $\sum_{i=1}^{K} \alpha_i = 1$. When complete label data is presented, the problem is simplified to the supervised classification estimation (see chapter 4.1.1.1). However, if there are many missing labels or even no label at all, the parameter cannot be estimated from the training data. In that case, the expectation-maximisation (EM) algorithm is often used to find this maximise likelihood parameters.

4.2.7 EM algorithm

EM algorithm (Dempster et al., 1977)is an efficient iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. In ML estimation, we wish to estimate the model parameter(s) for which the observed data are the most likely.

The EM algorithm consists of two processes: The E-step, and the M-step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology. In the M-step, the likelihood function is maximized under the assumption that the missing data are known(Borman, 2004). Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration (Dempster et al., 1977). Most of the probability density function p are built from the multivariate Gaussian and used successfully in a number of applications, although the model can be used with many different components such as, Wishart distribution (Chen and Peter Ho, 2008). The meta code of the EM algorithm is as follows:

1. The initial step: Guess the parameters for the mixture density, i.e we have to guess $\{\alpha_1 ... \alpha_K, \theta_1 ... \theta_K\}$. For Gaussian, $\theta = \{M, \sum\}$.
2. E-step: Estimation of the unobserved y's (which Gaussian is used), conditioned on the observation, using the values $\alpha_i^{(l)}, \theta_i^{(l)} = \{M_i^{(l)}, \sum_i^{(l)}\}$:

$$p\left(y_i \mid x_i, \alpha_i^{(l)}, \theta_i^{(l)}\right) = \frac{\alpha_i^{(l)}.p\left(x_i \mid \theta_i^{(l)}\right)}{p\left(x_i \mid \theta_i^{(l)}\right)} = \frac{\alpha_i^{(l)}.p\left(x_i \mid \theta_i^{(l)}\right)}{\sum_{k=1}^{K} \alpha_k^{(l)} p\left(x_k \mid \theta_k^{(l)}\right)} \qquad [4\text{-}36]$$

3. M-step: We now want to maximize the expected log-likelihood of the joint event: An EM algorithm iteratively improves an initial estimate $\alpha_i^{(l)}, \theta_i^{(l)}$ by constructing new estimates, $\alpha_i^{(l+1)}, \theta_i^{(l+1)}$.
4. If the new parameters have converged, i.e. no more change in the estimates, the process stops. Otherwise go back to 2.

4.3 Classifier Combination

There are many reasons for combining multiple classifiers to solve a problem. To increase efficiency one can adopt multistage combination rules whereby objects are classified by a simple classifier using a small set of cheap features in combination with a reject option (Atukorale and Suganthan, 1999). For the more difficult objects more complex procedures, possibly based on different features, are used. Some classifiers can only make binary decisions (Melgani and Bruzzone, 2004) in that case, combination of classifiers must be done in order to perform multi-class classification. Neural networks show different results with different initialisations due to the randomness inherent in the training procedure (Kohonen, 1998). Therefore instead of selecting the best network, one can combine various networks together and take the advantage of all the attempts to learn from the data.

The architecture of various classifiers can be divided three categories: Parallel, Cascading and Hierarchical (Jain et al., 2000). In the parallel architecture, all the individual classifiers are invoked independently, and their results are then combined by a combiner. Most combination schemes in the literature belong to this category. In the gated parallel variant, the outputs of individual classifiers are selected or weighted by a gating device before they are combined.

In the cascading architecture, individual classifiers are invoked in a linear sequence. The number of possible classes for a given pattern is gradually reduced as more classifiers in the sequence have been invoked. For the sake of efficiency, inaccurate but cheap classifiers (low computational and measurement demands) are considered first, followed by more accurate and expensive classifiers.

In the hierarchical architecture, individual classifiers are combined into a structure, which is similar to that of a decision tree classifier. The tree nodes, however, may now be associated with complex classifiers demanding a large number of features. The advantage of this architecture is the high efficiency and flexibility in exploiting the discriminant power of different types of features. Using these three basic architectures, we can build even more complicated classifier combination systems (Ho et al., 1994).

Consider a pattern recognition problem where pattern Z is to be assigned to one of the m possible classes $(w_i,...,w_m)$. Let us assume that we have R classifiers each representing the given pattern by a distinct measurement vector. Denote the measurement vector used by the i^{th} classifier by x_i. In the measurement space each class w_k is modelled by the probability density function $p(x_i \mid w_k)$ and its prior probability of occurrence is denoted $p(w_k)$. We shall consider the models to be mutually exclusive which means that only one model can be associated with each pattern (Baofeng Guo et al., 2006).

Now, according to the Bayesian theory, given measurements $x_i, i = 1,..., R$, the pattern, Z, should be assigned to class w_j provided the posterior probability of that interpretation is at maximum, i.e.

assign $Z \to w_j$ if

$$p(w_j \mid x_1,..., x_R) = \max_k p(w_k \mid x_1,..., x_R) \qquad [4\text{-}37]$$

4.3.1 Product rule

Let us assume that the representations used are conditionally statistically independent. We can use the product rule obtain the decision rule by

assign $Z \to w_j$ if

$$p^{-(R-1)}(w_j)\prod_{i=1}^{R} p(w_j \mid x_i) = \max_{k=1}^{m} p^{-(R-1)}(w_k)\prod_{i=1}^{R} p(w_k \mid x_i) \qquad [4\text{-}38]$$

The decision rule quantifies the likelihood of a hypothesis by combining the posterior probabilities generated by the individual classifiers by means of a product rule. It is effectively a severe rule of fusing the classifier outputs as it is sufficient for a single recognition engine to inhibit a particular interpretation by outputting a close to zero probability for it. As we shall see below, this has a rather undesirable implication on the decision rule combination as all the classifiers, in the worst case, will have to provide their respective opinions for a hypothesized class identity to be accepted or rejected.

4.3.2 Sum rule

In some applications it may be appropriate further to assume that the posterior probabilities computed by the respective classifiers will not deviate dramatically from the prior probabilities. In such a situation we obtain a sum decision

assign $Z \rightarrow w_j$ if

$$(1-R)p(w_j)+\sum_{i=1}^{R} p(w_j \mid x_i)=\max_{k=1}^{m}\left[(1-R)p(w_k)+\sum_{i=1}^{R} p(w_k \mid x_i)\right] \qquad [4\text{-}39]$$

As far as the sum rule is concerned, the assumption that the posterior class probabilities do not deviate greatly from the priors will be unrealistic in most applications.

4.3.3 Max Rule

Approximating the sum by the maximum of the posterior probabilities, we obtain

assign $Z \rightarrow w_j$ if

$$(1-R)p(w_j)+R\max_{i=1}^{R}(w_j \mid x_i)=\max_{k=1}^{m}\left[(1-R)p(w_k)+R\max_{i=1}^{R}(w_k \mid x_i)\right] \qquad [4\text{-}40]$$

which under the assumption of equal priors simplifies to

assign $Z \rightarrow w_j$ if

$$\max_{i=1}^{R}(w_j \mid x_i)=\max_{k=1}^{m}\max_{i=1}^{R}(w_k \mid x_i) \qquad [4\text{-}41]$$

4.3.4 Min Rule

Bounding the product of posterior probabilities from above we obtain

assign $Z \rightarrow w_j$ if

$$p^{-(R-1)}(w_j)\min_{i=1}^{R} p(w_j \mid x_i)=\max_{k=1}^{m} p^{-(R-1)}(w_k)\min_{i=1}^{R} p(w_k \mid x_i) \qquad [4\text{-}42]$$

which under the assumption of equal priors simplifies to assign

assign $Z \rightarrow w_j$ if

$$\min_{i=1}^{R}\left(w_j \mid x_i\right) = \max_{k=1}^{m} \min_{i=1}^{R}\left(w_k \mid x_i\right) \qquad [4\text{-}43]$$

4.3.5 Median Rule

Note that under the equal prior assumption, the sum rule can be viewed to be computing the average a posterior probability for each class over all the classifier outputs, i.e.,

assign $Z \rightarrow w_j$ if

$$\frac{1}{R}\sum_{i=1}^{R} p\left(w_j \mid x_i\right) = \max_{k=1}^{m} \frac{1}{R}\sum_{i=1}^{R} p\left(w_k \mid x_i\right) \qquad [4\text{-}44]$$

Thus, the rule assigns a pattern to that class the average a posterior probability of which is the maximum. If any of the classifiers outputs a posterior probability for some class which is an outlier, it will affect the average and this in turn could lead to an incorrect decision. It is well known that a robust estimate of the mean is the median. It could therefore be more appropriate to base the combined decision on the median of the posterior probabilities. This then leads to the following rule:

assign $Z \rightarrow w_j$ if

$$\underset{i=1}{\overset{R}{med}}\left(w_j \mid x_i\right) = \max_{k=1}^{m} \underset{i=1}{\overset{R}{med}}\left(w_k \mid x_i\right) \qquad [4\text{-}45]$$

4.3.6 Majority Vote Rule

assign $Z \rightarrow w_j$ if

$$\sum_{i=1}^{R} \Delta_{ji} = \max_{k=1}^{m} \sum_{i=1}^{R} \Delta_{ki} \qquad [4\text{-}46]$$

Note that for each class w_k the sum on the right hand side simply counts the votes received for this hypothesis from the individual classifiers. The class which receives the largest number of votes is then selected as the consensus (majority) decision.

5 Classifier complexity and dimensional reduction techniques

5.1 Introduction

Spectral information has the advantages of easily expandable dimensionality in feature space. In the early days when there were 7 to 10 bands in multi-spectral images, each band is treated as one feature and classification based on these features were not a problem. This is not true when nowadays each pixel profiles contain hundreds of bands, classification results may be degraded due to Hughes phenomenon and information redundancy.

Method of dimensionality reduction can be divided into two categories: feature extraction and feature selection. Feature extractions are used to extract the intrinsic properties of the data by transformations or combinations of the original data whereas feature selections are used to identify and discard features that may have low discriminability or may not contribute to the classification task (Jain et al., 2000). The choice between feature selection and feature extraction depends on the application domain.

5.2 Hughes phenomenon

If the class-conditional densities are completely known or the number of training is large and representative enough to estimate the underlying densities, then the classification error rate does not increases as the features size increases. However, when the number of training samples per class is considerably smaller than the feature dimension (Zeng and Trussell, 2004), the classifier accuracy may degrade with an increase in the number of features for a fixed and small sample size. This is often known as ‘Hughes phenomenon’ or ‘peaking phenomenon’ (Hughes, 1968). Some authors suggest that it is a good practice to keep the size of the training samples as least ten times as large as the dimensionality (Jain and Zongker, 1997), although the exact relationship between the probabilities of misclassification, the number of training samples and the number of features are very complicated. Nevertheless, the general guideline is to increase the ratio of sample size to dimensionality as the classifiers complexity increases.

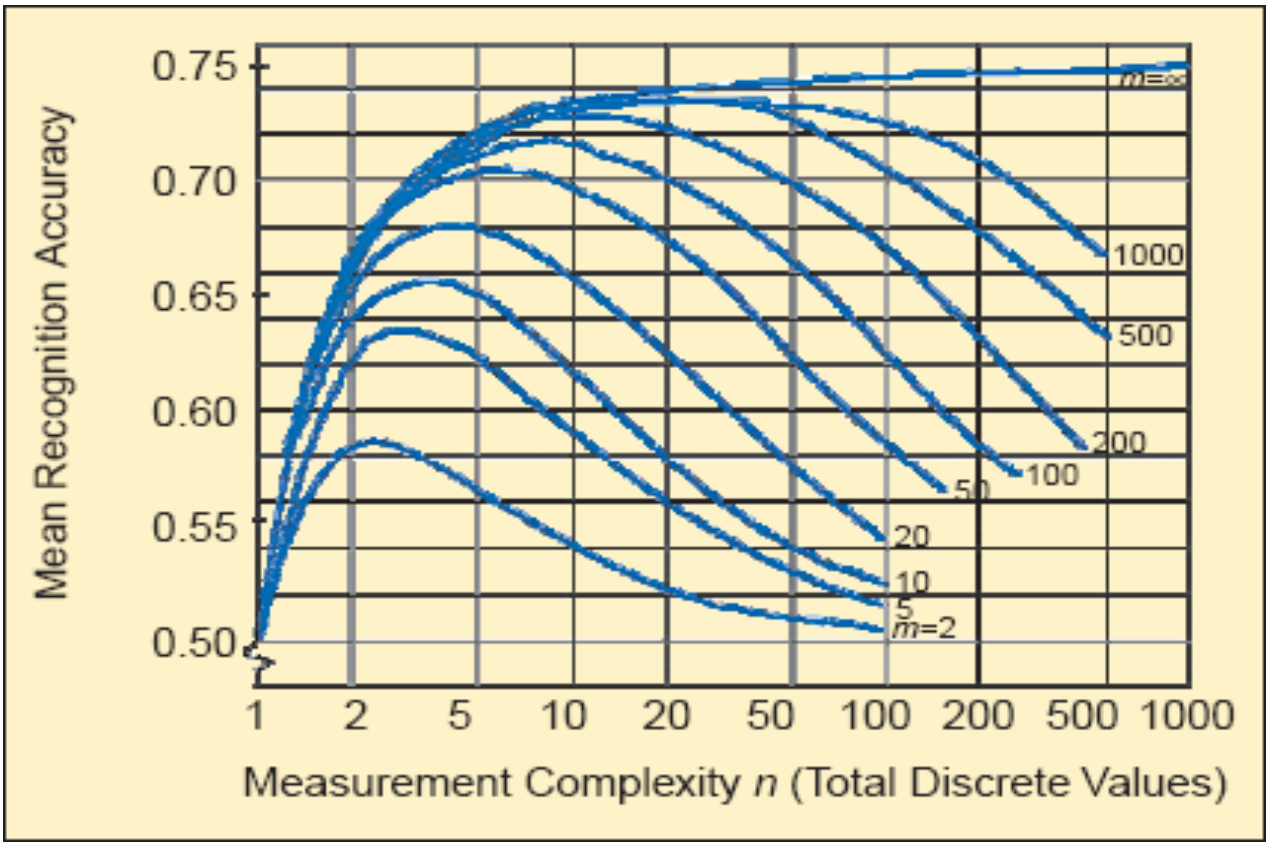


Figure 5-1: The Hughes phenomenon (Hughes, 1968). When the training sample size is small, the recognition accuracy decreases as the number of feature increases.

5.3 Information redundancy

High dimensional space is mostly empty in hyperspectral images, thus data can be projected to a lower dimensional subspace without losing significant information in terms of separability among the different statistical classes (Landgrebe, 2002). In many cases, it is unnecessary to process all the spectral bands of a hyperspectral image, since most materials have specific characteristics only at certain bands, which makes the remaining spectral bands somewhat redundant (Baofeng Guo et al., 2006). Watanabe's ugly duckling theorem (Watanabe, 1969) also supports the need of discarding redundant information, because It is possible to make two arbitrary patterns similar when large amounts of redundant features exist within both patterns.

Larger number of spectral bands may potentially make the discrimination between more detailed classes possible, but if there are many poor signal-to-noise ratio (SNR) band, the classification results will be degraded (Landgrebe, 2002), (Jia and Richards, 1999).

Therefore appropriate feature extraction/selection technique can remove those unwanted bands, thus improve the classification performance.

Spectral information has the advantage of easily expandable dimensionality in feature space without significant cost. In the early days when there were 7 to 10 bands in multi-spectral images, each band is treated as one feature and classification based on these features were not a problem. This is not true when nowadays each pixel profiles contain hundreds of bands.

If the class-conditional densities are completely known or the number of training is large and representative enough to estimate the underlying densities, then the classification error rate does not increases as the features size increases. However, when the number of training samples per class is considerably smaller than the feature dimension (Zeng and Trussell, 2004), the classifier accuracy may degrade with an increase in the number of features for a fixed and small sample size. This has been termed “the curse of dimensionality” by Bellman (Bellman, 1961), which leads to ‘Hughes phenomenon’ or ‘peaking phenomenon’ (Hughes, 1968). Larger number of spectral bands may potentially make the discrimination between more detailed classes possible, but if the training samples are insufficient for the classification requirement, the results will be degraded (Landgrebe, 2002), (Jia and Richards, 1999). For example, if quadratic classifier (see chapter 4.1.1.1) is applied to an N spectral band image, the size of training samples must be at least N+1, otherwise the sample covariance matrix will be singular. Some authors suggest that it is a good practice to keep the size of the training samples as least ten times as large as the dimensionality (Jain and Zongker, 1997), nevertheless, the general guideline is to increase the ratio of sample size to dimensionality as the classifiers complexity increases.

The goal of dimension reduction is to reduce the number of feature without sacrificing significant information. It is important to preserve the ‘useful’ information. Reducing too much features may lead to a loss in discrimination power, therefore lower the classification accuracy (Jain et al., 2000). On the other hand, appropriate reduction technique can remove those unwanted bands, thus improve the classification performance when training samples are limited. Method of dimensionality reduction can be divided into two categories: feature extraction and feature selection. Feature

extractions are used to extract the intrinsic properties of the data by transformations or combinations of the original data whereas feature selections are used to identify and discard features that may have low discrimination power or may not contribute to the classification task (Jain et al., 2000; Jain and Zongker, 1997). The choice between feature selection and feature extraction depends on the application domain and user preference.

5.4 Feature extraction

Feature extraction is the transformation of the original data (using all variables) to a data set with a reduced number of variables. One of the most commonly used techniques is Principal components analysis (PCA) (Duda et al., 2000). PCA is efficient and usually yields satisfactory outcomes in extracting useful features. Other interesting techniques like projection pursuit (Friedman and Tukey, 1988), and Maximum Noise Fraction transform (MNF) (Green et al., 1988) have also been used in hyperspectral imaging (Chang and Du, 1999; Jimenez and Landgrebe, 1999). Both techniques are quite similar to PCA in the way that they put the principal components from the most significant to the least significant. Projection pursuit involves finding the most "interesting" possible projections in multidimensional data whereas MNF orders the principal components according to the signal to noise ratio.

Supervised feature extractions algorithms are also used widely in hyperspectral applications. The most common example is Fisher linear discriminant analysis (or called Discriminate analysis feature extraction, DAFE) (Landgrebe, 2002; Duda et al., 2000). Training samples are required to find the best discriminant functions.

Neural network can be viewed as massively parallel computing systems consisting of a large number of simple processors with many interconnections. Neural networks provide a new suite of non-linear algorithms for feature extraction (using hidden layers). The popular networks, such as Self-Organizing Map (SOM) and multi-layer perceptions, can be used not only in classification and clustering, but also in non-linear feature extraction (Kohonen, 1998). Other non-linear feature extraction methods including Kernel PCA (KPCA), Isomap and Locally Linear Embedding (LLE) have also been attempted in

hyperspectral data (Scholkopf et al., 1997; Tenenbaum et al., 2000; Roweis and Saul, 2000).

5.4.1 Principal components analysis

The purpose of principal components analysis is to derive new variables that are linear combinations of the original variables and are uncorrelated. Geometrically, principal components analysis can be thought of as a rotation of the axes of the original coordinate system to a new set of orthogonal axes that are ordered in terms of the amount of variation of the original data they account for (Webb, 1999).

One of the reasons for performing a principal components analysis is to find a smaller group of underlying variables that describe the data. In order to do this, we hope that the first few components will account for most of the variation in the original data (Webb, 1999).

Principal components analysis is a variable-directed technique. It makes no assumptions about the existence or otherwise of groupings within the data and so is described as an unsupervised feature extraction technique (Webb, 1999).

The transformation is based on the covariance of the original data. Assume x represents the vector of a pixel in an N-dimensional image (Tsai et al., 2007). The image covariance matrix Σ, is an N*N matrix and can be constructed according to all pixels, xi, i=1,2, …,K and the mean vector m as below

$$\Sigma = E\left\{(x_i - m)(x_i - m)^T\right\} = \frac{1}{K}\sum_{i=1}^{K}(x_i - m)(x_i - m)^T \quad [5\text{-}1]$$

$$m = E\{x\} = \frac{1}{K}\sum_{i=1}^{K} x_i \quad [5\text{-}2]$$

There are two tasks in a PCA. The first is an egien-analysis to generate the transformation matrix A; and the second is the linear transformation for each pixel to project data onto the new orthogonal space, y (Tsai et al., 2007). The eigenvectors e_i of the scatter matrix are given by:

$$\Sigma e_i = \lambda e_i \quad [5\text{-}3]$$

and the transformation is defined as

$$y = A^T (x - m) \quad [5\text{-}4]$$

where A is a N* d matrix whose columns are the d eigenvectors with the largest d eigenvalues λ_i, sorted in decreasing order. Note that feature selection is performed within the formula because only the first d principle components are selected for classification.

Principal components analysis produces an orthogonal coordinate system. The axes are ordered in terms of the amount of variance in the original data. If the first few principal components account for most of the variation, then these may be used to describe the data, thus leading to a reduced-dimension representation. We might also like to know if the new components can be interpreted as something meaningful in terms of the original variables. However, in practice the new components will be difficult to interpret (Webb, 1999).

Directly applying PCA to the entire data set of a high dimensional hyperspectral image may not be good (Tsai et al., 2007), for example, hyperspectral remote sensing images usually exhibit higher variances in the short wavelengths; thus PCA will be dominated by those bands. Many authors in the literature have proposed many algorithms to improve PCA for better use with hyperspectral images. Segmented principal component transformation (SPCT), which is proposed by Jia and Richards (Richards and Jia, 2006), compares pairwise bands and then spectral bands are divided into groups according to the correlation matrix. The best principal components are extracted from each of the group. There is another PCA technique for non-linear feature extraction called kernel PCA (KPCA) (Scholkopf et al., 1998) and it has been in development in recent years. KPCA can efficiently compute PCs in high-dimensional feature spaces by means of integral operators and non-linear kernel functions. The basic idea of KPCA is to map the input space into a feature space via the kernel trick (Appendix 13.3) and then to compute the PCs in that feature space. Unlike PCA which only focus on second order statistics, KCPA can extract higher order statistics features (Mathieu Fauvel et al., 2006).

5.4.2 Maximum Noise Fraction transform (MNF)

The Maximum Noise Fraction transform (MNF) or noise-adjusted principal component transform (NAPCT) consists in projecting the original image in a space where the new components are sorted in order of signal to noise ratio (SNR) (Green et al., 1988; Chang and Du, 1999). While components in PCA maximise the variance in the data, MNF components maximise the signal-to-noise ratio. Finally, the inverse MNF allows the filtered image to be re-projected in the original space.

Our choice should then achieve the desired optimal ordering in terms of image quality. This transformation can be defined in several ways. It can be shown that the same set of eigenvectors is obtained by procedures that maximise either the signal-to-noise ratio or the noise fraction. We stress that all the results described can be obtained from either measure.

Let us consider a multivariate data set of p-bands with grey levels

$$\mathrm{Z}_i(x), i = 1,..., p \qquad [5\text{-}5]$$

where x gives the coordinates of the sample. We shall assume that

$$\mathrm{Z}(x) = \mathrm{S}(x) + N(x) \qquad [5\text{-}6]$$

Where $\mathrm{Z}^{\mathrm{T}}(x) = \{\mathrm{Z}_1(x),..., Z_P(x)\}$, and S(x) and N(x) are the uncorrelated signal and noise components of Z(x). Thus

$$Cov\{Z(x)\} = \Sigma = \Sigma_S + \Sigma_N \qquad [5\text{-}7]$$

where Σ_S and Σ_N are the covariance matrices of S(x) and N(x), respectively. The MNF transform can be expressed in the matrix form

$$\mathrm{Y}(x) = A^T Z(x), i = 1,..., p \qquad [5\text{-}8]$$

Where $\mathrm{Y}^{\mathrm{T}}(x) = (Y_1(x),..., Y_P(x))$ and $A = (a_1,..., a_P)$.

To obtain the MNF transform, we need to know the covariance matrices of the signal Σ_S and noise Σ_N, components and use the signal-to-noise ratio (SNR) to determine the ordering of the MNF components. In many practical situations, these covariance matrices are unknown and need to be estimated. Σ is usually estimated using the

sample covariance matrix of Z (x) and the noise components Σ_N could be extracted by some types of spatial filtering of each band. The selection of filters is determined by the estimated spatial characteristics of the noise and therefore no filters will extract noise completely. Another method is to use the minimum/maximum autocorrelation factors (MAF) (Switzer and Green, 1984) which estimates noise by exploiting the fact that, in most remotely sensed data, the signal correlation of neighbouring pixels are much stronger than the noise correlation at any point in the image.

5.4.3 Projection Pursuit

Projection Pursuit was first proposed by Friedman and Tukey (Friedman and Tukey, 1988) and was used as a technique for exploratory analysis of multivariate data. The idea is to project a high dimensional data set into a low dimensional data space while retaining the information of interest. It designs a projection index (PI) to explore projections of interestingness.

Let's assume that there are data N points with dimensionality K, $X = [x_1, x_2, ..., x_N]$ is a K*N data matrix, and a is a K-dimensional column vector, which serves as a desired projection.

Then $a^T X$ represents an N-dimensional row vector that is the orthogonal projections of all sample data points mapped onto the direction a, where T is the matrix transpose. Now if we let $H(.)$ be a function measuring the degree of the interestingness of the projection $a^T X$ for a fixed data matrix X, a projection index (PI) is a real-valued function of a, $I(a)$ defined by

$$I(a) = H(a^T X) \qquad [5\text{-}9]$$

The PI can be easily extended to multiple directions $\{a_1, ..., a_J\}$. In this case, $A = [a_1, a_2, ..., a_J]$ is a K*J projection direction matrix, and the corresponding projection index is also a real valued function $I(A): R^{K*J} \rightarrow R$ given by

$$I(A) = H(A^T X) \qquad [5\text{-}10]$$

In remote-sensing data analysis, the choice of the projection index is the most critical aspect of this technique. Jimenez (Jimenez and Landgrebe, 1999) suggests the use of Bhattacharyya distance as a measure of PI and the desired projection matrix A is constantly updated according to the value of PI.

5.4.4 Independent Component Analysis

Independent Component Analysis, ICA, has received considerable interest in recent years because of its versatile applications ranging from source separation, channel equalization to speech recognition and functional magnetic resonance imaging (Hyvärinen and Oja, 2000). The key idea of the ICA assumes that data are linearly mixed by a set of separate independent sources and de-mix these signal sources according to their statistical independency measured by mutual information (Ouyang et al., 2008). In order to validate its approach, an underlying assumption is that at most one source in the mixture model can be allowed to be a Gaussian source. This is due to the fact that a linear mixture of Gaussian sources is still a Gaussian source. More precisely, let be a mixed signal source vector expressed by

$$x = As \qquad [5\text{-}11]$$

where A is an L*N mixing matrix and s is a N-dimensional signal source vector with N signal sources needed to be separated. However the mixing matrix is normally unknown, therefore, the purpose of the ICA is to find W a de-mixing matrix that separates the signal source vector into a set of sources which are statistically independent. The independent component can be simply obtain by

$$s = Wx \qquad [5\text{-}12]$$

Pre-processing is performed before ICA is actually applied, which normally involve demeaning and whitening the data such that it has zero-mean and its components are uncorrelated with unity variances. The estimation of ICA is done measurement of non-gaussianity, minimization of mutual information and maximum likelihood estimation. The classical measure of non-gaussianity is kurtosis or the fourth-order statistics. The kurtosis of y is classically defined by

$$kurt(y) = E\{y^4\} - 3\left(E\{y^2\}\right)^2 \qquad [5\text{-}13]$$

A second very important measure of non-gaussianity is given by negentropy. Negentropy is based on the information-theoretic quantity of (differential) entropy. To obtain a measure of non-gaussianity that is zero for a Gaussian variable and always nonnegative, one often uses a slightly modified version of the definition of differential entropy, called negentropy. Negentropy J is defined as follows

$$J(y) = H(y_{gauss}) - H(y) \qquad [5\text{-}14]$$

where y_{gauss} is a Gaussian random variable of the same covariance matrix as y. Due to the above-mentioned properties, negentropy is always non-negative, and it is zero if and only if y has a Gaussian distribution. Negentropy has the additional interesting property that it is invariant for invertible linear transformations (Hyvärinen and Oja, 2000; Ouyang et al., 2008).

5.4.5 Fisher Linear Discriminant Analysis

To that purpose Fisher-LDA considers maximizing the following objective:

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \qquad [5\text{-}15]$$

where S_B is the "between classes scatter matrix" and S_W is the "within classes scatter matrix" (Welling, 2006). Note that due to the fact that scatter matrices are proportional to the covariance matrices we could have defined J using covariance matrices – the proportionality constant would have no effect on the solution. The definitions of the scatter matrices are:

$$S_B = \sum_c N_c (\mu_c - m)(\mu_c - m)^T \qquad [5\text{-}16]$$

$$S_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T \qquad [5\text{-}17]$$

where,

$$\mu_c = \frac{1}{N_c} \sum_{i \in c}^{N_c} x_i \qquad \& \qquad m = \frac{1}{N} \sum_i^N x_i = \frac{1}{N} \sum_c N_c \mu_c$$

and N_c is the number of cases in class c. Oftentimes you will see that for 2 classes S_B is defined as $S'_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$. This is the scatter of class 1 with respect to the scatter of class 2 and you can show that $S_B = \frac{N_1 N_2}{N} S'_B$, but since it boils down to multiplying the objective with a constant is makes no difference to the final solution. It is also interesting to observe that since the total scatter,

$$S_T = \sum_i (\mu_i - m)(\mu_i - m)^T \qquad [5\text{-}18]$$

is given by $S_T = S_W + S_B$ the objective can be rewritten as,

$$J(w) = \frac{w^T S_T w}{w^T S_w w} - 1 \qquad [5\text{-}19]$$

and hence can be interpreted as maximizing the total scatter of the data while minimizing the within scatter of the classes (Welling, 2006).

An important property to notice about the objective J is that its invariant w.r.t. rescaling of the vectors $w \to \alpha w$. Hence, we can always choose w such that the denominator is simply $w^T S_w w = 1$, since it is a scalar itself (Welling, 2006). For this reason we can transform the problem of maximizing J into a constrained optimisation problem. Using the Lagrangian to minimize $-\frac{1}{2} w^T S_B w$, the solution can be simplified to an eigenvalue equation as

$$S_B w = \lambda S_w w \qquad \Rightarrow \qquad S_W^{-1} S_B w = \lambda w \qquad [5\text{-}20]$$

5.4.6 Neural networks feature extractor

Neural networks can be used directly for feature extraction in an unsupervised fashion. A feed-forward network offers an integrated procedure for feature extraction; non-linear features can also be extracted by adding an extra hidden layer. The architecture of neural networks could also simulate other classical feature extraction technique such as PCA, show in Figure 5-2. The network has d input and d output where d is the given number of features. The hidden layer with three neurons captures the first three principal

components, and instead of using sigmoid, the neurons use linear transfer functions. Self Organising Map (SOM) (Landgrebe, 2002) is another type of neural networks, which can be used for non-linear feature extraction. The neurons in SOM are arranged in an m-dimensional grid, each neuron is connected to all the d-dimensional input features with different weights. After training is done, SOM offers an m-dimensional with spatial connectivity, which can be interpreted as feature extraction.

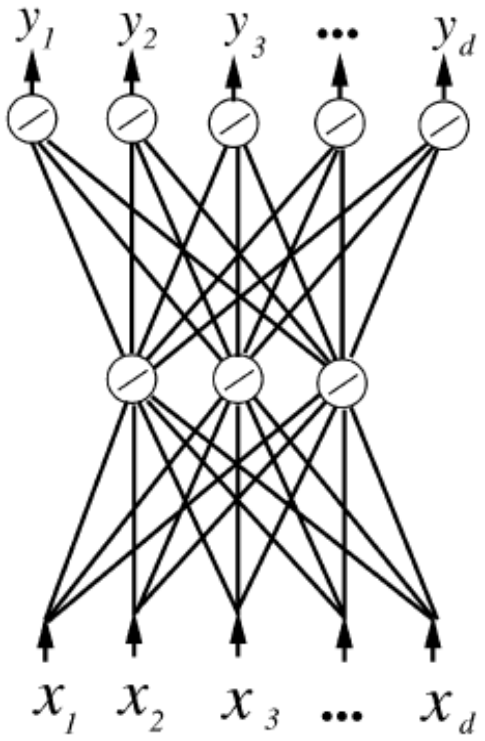


Figure 5-2: An example of linear neural network feature extractor (Jain et al., 2000).

5.5 Feature Selection

Although feature extraction algorithms provide good discrimination power, they may suffer from the fact that the transformed features do not have any physical meanings. On the other hand, feature selection discards some of the redundant features or bands may not be an optimal approach, but the images' properties remain. Feature selection is also optimal to lower the dimensionality for the data. Instead of projecting the features into another subspace, some features that are less relevant for classification are discarded. Feature selection techniques generally involve a search strategy, a selection evaluation function and a stopping criterion.

5.5.1 Search strategy

The next step is to choose the search strategy for feature selection. The most straightforward approach to the feature selection problem would require examining all possible combination and selecting the subset with the largest discriminant power. Although exhaustive search is an optimal approach, the computation complexity is large. In order to guarantee the optimality of a 12-dimensional feature subset out of 24 available features, approximately 2.7 million possible subsets must be evaluated.

The other optimal feature selection method which avoids the exhaustive search is based on the branch and bound algorithm (Narendra and Fukunaga, 1977). It is a top-down procedure, beginning with the set of p variables and constructing a tree by deleting variables successively. It relies on the monotonic property of the feature selection criterion J(). For two subsets of the variables, X and Y, $X \subset Y \Rightarrow J(X) < J(Y)$. The branch and bound algorithm may not be computationally feasible (Serpico and Bruzzone, 2001). The growth in the number of possibilities that must be examined is still an exponential function of the number of variables. Hence, in the case of feature selection for HSI classification, only suboptimal algorithms can be used.

There are many types of suboptimal feature selection found in the literature. The simplest search strategy is the best individual (BI) (Jain et al., 2000). This technique evaluates all features individually and ranks them according to the criterion function. The best feature subsets from the rank order. In general, BI method is not suitable for hyperspectral due to the fact that the best pair of features need not contain the best single feature(Jain et al., 2000; Landgrebe, 2005). are the sequential forward selection (SFS) and the sequential back selection (SBS). Although suboptimal algorithms such as the sequential forward floating selection (SFFS) method and the sequential backward floating selection (SBFS) methods are not capable of examining every feature combination, they will assess a set of potentially useful feature combinations.

5.5.2 Selection Criteria

The selection process is to identify bands which are a subset of the original spectral bands that contains most of the characteristics. Let the feature selection criterion for the

set X be represented by J(X). Let us assume that a higher value of J indicates a better feature. We can choose the feature set in essentially two ways (Webb, 1999).

1. Filter method: The first approach is to estimate the overlap between the distributions from which the data are drawn. Those feature sets with minimal overlap are chosen as final subsets. It has the advantage that it is often fairly easy to implement and computationally efficient. The final selection result is also independent of the final classifier employed, thus it does not inherit any bias of the classification algorithm. However, it has the disadvantage that the assumptions made in determining the overlap are often crude and may result in a poor estimate of the discriminant power (Webb, 1999).
2. Wrapper method: Wrapper method is very classifier on the reduced feature set can be and choose the feature sets for which the classifier performs well on a separate test/validation set. In this approach, the feature set is chosen to match the classifier. A different feature set may result with a different choice of classifier (Webb, 1999).

The choice of feature selection evaluation function is mainly depending on the method used. If the filter approach is used, then the evaluation function is based from the data intrinsic properties. The data intrinsic category includes distance (Keshava, 2004; Martinez-Uso et al., 2007), information entropy (Keshava, 2004), and dependence measures. If the wrapper method is chosen, then the feature selection criterion J(X)=(1-Pe), where Pe is the classification error rate.

6 Accuracy Assessment of image classification

6.1 Introduction

Accuracy assessment is an important step to analyze and evaluate the quality and reliability of hyperspectral data. Site-specific accuracy assessment has been commonly employed especially in the remote sensing community. The difference between site and non-site specific assessment is the use of spatial information of the map. In a non-site specific accuracy assessment, the total number of classified pixels for each category is compared regardless of the location of the pixels. In a site specific assessment, the classified results are compared with the same locations on the reference data. Therefore it avoids errors due to the wrongly classified pixels in the wrong locations. There are two types of criteria to measure the accuracy of the images: location accuracy and classification accuracy. Location accuracy is a measure of how precisely pixels of the image cubes are mapped to their true location on the ground. Classification accuracy assessment provides a comparison between classification results and known reference data.

6.2 Site-specific assessment

6.2.1 Confusion Matrix

The use of confusion matrix, error matrix or contingency matrix is currently the core method of the accuracy assessment in remote sensing literature (Foody, 2002). A confusion matrix is a square array of numbers which lists the reference/ ground-truth data in the columns and the classified results in the rows. The recommend (Foody, 2002; Congalton and Green, 1999) layout of a confusion matrix is present in Table 6-1.

Confusion matrix is very helpful in analysing the overall accuracy of the whole images as well as the accuracy of individual classes. The **overall accuracy** is the basic accuracy measure which is the sum of the correctly classified pixels (the diagonal of the matrix which is shaded grey) divided by the total number of pixels, n. It is normally sufficient to provide a good indication of the performance of a classification rule. However, presenting the overall accuracy alone may not be enough. The additional information

from the confusion matrix may become handy if further investigation about the reliability of the classification results is required.

The **producer's accuracy** indicates the percentage of each individual reference class was corrected identified in the classified map. The producer's accuracy of a class can be derived by dividing the number of correctly classified pixels by the total numbers of pixels of that particular reference class. It also shows the error of omission which refers to excluding an area (or some pixels) from the class in which it does truly belong.

The **user's accuracy** is directly related to the error of commission, the amount of area that is classified to a category which does not belong to that category. The user's accuracy is the number of correctly classified pixels divided by the total numbers of pixels that are classified as that particular class.

Under this method the ground truth data has been regarded as an accurate and reliable representation of the actual site. In fact, as Foody stated, "the ground data are just another classification which may contain error" (Foody, 2002). These may be errors from mislabelling of certain area and errors due to mis-location of the map. The reference data acquisition methods, sampling methods and class definitions are some factors that can influence the accuracy of the reference data itself. As long as the accuracy assessments are based on the reference data, there is a danger of falsely interpreting some classified results as errors which are in fact correct because of the inaccuracy of the reference. A thorough and precise ground truthing of the site may result in a more accurate map, but this is normally not feasible due to the cost and time of taking data. In the situations when actual ground truth data is absent, remote sensing data with finer spatial resolution is often used as the reference data. In this case the resulting confusion matrix and accuracy of the classified data are based on the derived reference map. This derived map is generated by photo interpreters and expert knowledge of the site which may notably distort fidelity of the accuracy report.

		Reference Class				No. of classified pixels
		1	2	3	K	
Classified Class	1	n_{11}	n_{12}	n_{13}	n_{1K}	n_{1+}
	2	n_{21}	n_{22}	n_{23}	n_{2K}	n_{2+}
	3	n_{31}	n_{32}	n_{33}	n_{3K}	n_{3+}
	K	n_{K1}	n_{K2}	n_{K3}	n_{KK}	n_{K+}
No. of ground truth pixels		n_{+1}	n_{+2}	n_{+3}	n_{+K}	n

Table 6-1: Standard format of a confusion matrix

Confusion matrix and the statistical measures that were mentioned above had been widely adopted in the remote sensing community. They are quite often recognised as the standard for accuracy assessment (Foody, 2002; Congalton and Green, 1999). However, in many situations when the ground reference data may not be an accurate and reliability source of information, the accuracy statements or report of the classified results are questionable. In the worst case when reference data is not presented at all, the use of confusion matrix and the statistical measures based on it may not be an option. Therefore there is a need to employ different accuracy assessment techniques.

6.2.2 Kappa Coefficient

The Kappa coefficient is a statistical measure to determine the agreement between two maps that was not occurring by chance. It is normally used to compare the agreement between reference data and classified result. Kappa coefficient or KHAT ($\hat{K}$) statistic has been used in sociology and psychology for many years since a seminal paper was published by Jacob Cohen (Cohen, 1960). However it was only widely promoted in the remote sensing community, since Congalton et al. introduced the method in 1983 (Congalton and Mead, 1994).

The KHAT is given by

$$\hat{K} = \frac{P_0 - P_C}{1 - P_C} \qquad [6\text{-}1]$$

where P_0 is the observed agreement and P_C is the chance agreement.

For computational purposes

$$\hat{K} = \frac{n\sum_{i=1}^{k} n_{ii} \quad - \sum_{i=1}^{k} n_{i+} n_{+i}}{n^2 - \sum_{i=1}^{k} n_{i+} n_{+i}}; \qquad [6\text{-}2]$$

where $n_{i+} = \sum_{j=1}^{k} n_{ij}$ & $n_{+j} = \sum_{i=1}^{k} n_{ij}$

The KHAT value is calculated for the matrix and is measured of how well the classification agrees with the reference data. If KHAT is equal to 1, both data sets are in perfect agreement and if KHAT is equal to 0, there are no agreements between both sets of data.

6.2.3 Drawbacks of site specific assessment methods

Confusion matrix and the statistical measures that were mentioned above had been widely adopted in the remote sensing community. They are quite often recognised as the standard for accuracy assessment (Foody, 2002; Congalton and Green, 1999). However, in many situations when the ground reference data may not be an accurate and reliability source of information, the accuracy statements or report of the classified results are questionable. In the worst case when reference data is not presented at all, the use of confusion matrix and the statistical measures based on it may not be an option. Therefore there is a need to employ different accuracy assessment techniques.

6.3 Non-site specific assessments

6.3.1 Cross Validation & the Leave One Out Method

The alternative methods which do not depend on ground truth map are the cross validation & Leave One Out (LOO) approach (Landgrebe, 2005). Both methods only use the training samples to assess the accuracy of the classification rules.

The cross validation starts by dividing the available labelled pixels into k subsets. Then one of those subsets is treated as the testing data and the rest of the pixels are used to train the classifier. Then the process repeats k times with each subset is used as the testing data once. The k assessment results can be averaged to produce a single estimation.

The leave one out method is a special form of the cross validation. It treats each individual labelled pixel as one subset and trains the classifier on the remainder subsets. The trained classifier is used label the pixel left out. That pixel is then replaced but another subset and the process repeated. This is done for all pixels in the training set and the average classification accuracy is calculated. This method can produce an unbiased estimate of classification accuracy if the samples are representative, (Landgrebe, 2005), but it is very computational expensive.

6.3.2 Bootstrapping

In the simplest form of bootstrapping methods are called the e0 bootstrap. For e0 bootstrap, the bootstrap training samples are chosen by randomly picking with replacement from the original training set. The testing data is drawn from original training set that was not chosen for bootstrap training. Another popular method is called the 0.632 bootstrap method, details of this method found in (Efron and Tibshirani, 1997).

6.4 Separability Measures

6.4.1 Overview

In the case where labelled samples are not presented or the labelled samples extracted from the image are not representative, all of the assessment methods like cross validation and bootstrapping are not suitable for accuracy purpose. There is a need to

use some alternative methods other than assessment methods based on the probability of error (correct) classification.

Consider a two-class problem with a dimensional space of two, our goal is to determine whether each sample belongs to class a or class b with minimum error. If the distributions of the two classes are well separated, then it is unlikely that the classifier would make a wrong decision. On the other hand, if there is a large degree of overlap between the two distributions, the classification error would expect to be large (Richards and Jia, 2006).

Consider now an attempt to quantify the separation between a pair of probability distributions as an indication of the degree of overlapping. It is not sufficient to use the distance between the two means of two distribution functions, the variances can also influence the tails of the distributions and hence the degree of overlapping between both of the distributions. Therefore in order to measure the separability one must use the mean distance and the covariance of the distributions (Richards and Jia, 2006).

6.4.2 Divergence

The calculation of divergence is related to the decision rules of maximum likelihood classification. Hence, in computing and estimating the signatures, divergence will be helpful in foretelling the results of the c classification obtained from maximum likelihood classifiers.

The separability can be calculated by three options. Covariance and the mean vectors of the signatures in the spectral bands that are being examined in order to find similarities and differences are taken into consideration by all the formulae.

The divergence (d_{ij}) can be calculated by the formula,

$$d_{ij} = \frac{1}{2} Tr\left\{\left(\sum_i - \sum_j\right)\left(\sum_j^{-1} - \sum_i^{-1}\right)\right\} + \frac{1}{2} Tr\left\{\left(\sum_j^{-1} + \sum_i^{-1}\right)\left(m_i - m_j\right)\left(m_i - m_j\right)^t\right\} \quad [6\text{-}3]$$
$$= \text{Term1} + \text{Term2}$$

Where,

i and j are the two class labels that are being examined for the similarities and the differences.

C_i is the covariance matrix for signature i

μ_i is the statistical mean for signature i

tr is the trace (algebra of matrix)

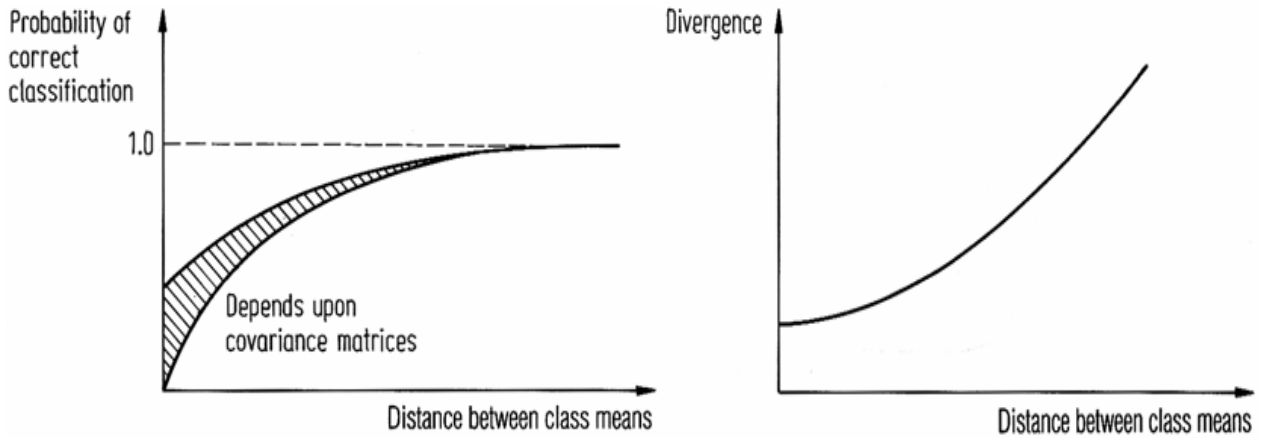
T is the matrix transposition.

Term 1 uses the covariance matrix and Term2 is the square distance between the means that is normalised by the covariance.

Note that the equation only measures the divergence between two distributions and both distributions must be normally distributed. In the case of more than two classes, it is important to check all pairwise divergences.

6.4.3 Problem with Divergence as a measure of classification performance

In theory, as the distributions of different classes become further away from each other classes in the multispectral space, the probability of correctly classifying a pattern is asymptotic to 1 as shown in Figure 6-1a. However, if divergence is used instead of probability of correct classification, the divergence increases quadratically towards infinity as the distances between class means increases as shown in Figure 6-1b. It implies that as the separations are already very large, further small increases can lead to huge increase in classification accuracy but it is not true in practice. There is only slight increase in classification accuracy as the probability gets closer to 1 as shown in Figure 6-1a. The Jeffries-Matusita and Transform divergence discuss in the next session do not suffer from this problem.



a) b)

Figure 6-1: a) probability correct classification as a function of spectral class separation (Richards and Jia, 2006) b) divergence as a function of spectral class separation (Richards and Jia, 2006)

6.4.4 Jeffries-Matusita Distance

The JM distance is derived by using the Bhattacharyya distance as a measure of separability assuming all classes are normally distributed. The Bhattacharyya distance is given by

$$B = \frac{1}{8}\left(m_i - m_j\right)^t \left\{\frac{\Sigma_i + \Sigma_j}{2}\right\}^{-1} \left(m_i - m_j\right) + \frac{1}{2}\ln\left\{\frac{\left|\left(\Sigma_i + \Sigma_j\right)/2\right|}{\left|\Sigma_i\right|^{\frac{1}{2}}\left|\Sigma_j\right|^{\frac{1}{2}}}\right\} \quad [6\text{-}4]$$

where the first term is the square of normalised distance between the class means.

The JM-distance is then given as follows in which B is referred to the Bhattacharyya distance:

$$j_{ij} = 2\left(1 - e^{-B}\right) \quad [6\text{-}5]$$

In a two class situation, the JM distance is asymptotic to 2.0 and the relationship between JM- distance and distance between class means can be shown in Figure 6-2. The shape of the curve is very similar to the plot in Fig a. with 100% classification accuracy when the JM-distance is equal to 2.

Although the JM-distance performs better than divergence as a measure of separability, the computationally complexity is high. In the case of divergence, most of the computational costs are largely on calculating the matrix inverse whereas JM-distance requires the matrix inverses and determinants. This implies the JM-distance is $\frac{1}{2}(M+1)$ times as expensive as divergences in time complexity. Due to the disadvantages with divergences and the computational cost of using JM-distance, Swain and Davis (Swain and Davis, 1978) has proposed the use of transformed divergence as a measure of separability.

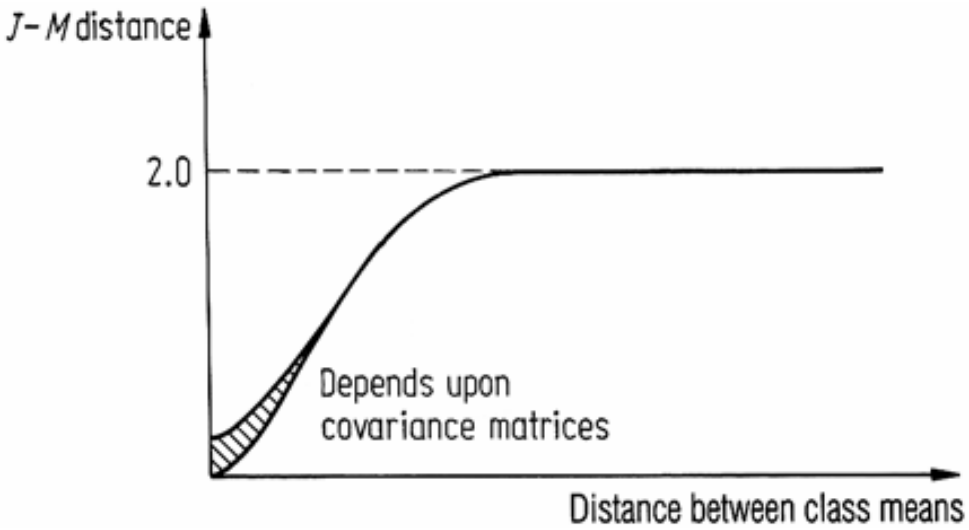


Figure 6-2: Jeffries-Matusita distance as a function of separations between the class means (Richards and Jia, 2006)

6.4.5 Transformed Divergence

By looking at and the equation of JM-distance, the parameter B is similar to the divergence. Both of them involve the use of covariance and the normalised distances between class means. Therefore it is possible to make use of the form of JM-distance which employs divergence as the parameter instead of the Bhattacharyya distance. The transformed divergence is given by:

$$td_{ij} = 2\left(1 - e^{-d_{ij}/8}\right) \qquad [6\text{-}6]$$

Transformed divergence describes the exponential decrement in the weight to the increment in the class distances. The range of the values of the transformed divergence scale is 0 to 2.0. The numerical value evaluates the separation between the two classes. If the obtained results are greater than 1.9 then the classes are able to separate. If the obtained results lie in between the values of 1.7 and 1.9 then the separation is considered fair enough. And if the value of the obtained results is below 1.7 then the separation is considered as poor.

Swain and King (Swain and King, 1973) have derived an empirical relationship between transformed divergence and classification accuracy (for two classes comparison) using 2790 sets of multidimensional, normally distributed data as shown in Figure 6-3. It

provides very useful information to validate the usefulness of transformed divergence as an alternative of the classification accuracy assessment.

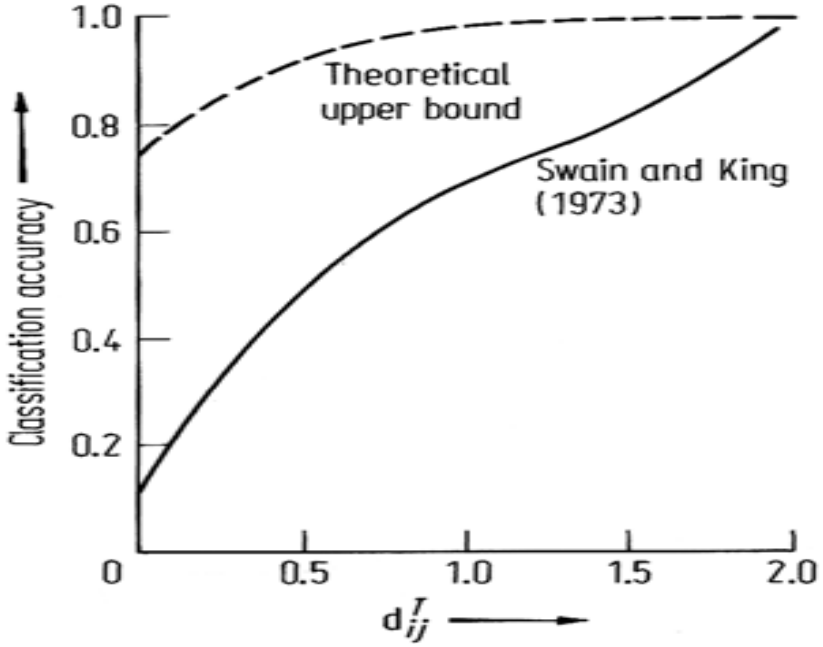


Figure 6-3: Probability of correction classification as a function of pairwise transformed divergence (Landgrebe, 2005)

The equations as shown in Equations 6-5 and 6-6 have been developed for measuring the dissimilarities between a pairwise of classes, and in this study (for details refer to chapter 12) we have derived an overall scoring for ALL the classes in the data sets for the Transformed Divergence (TD) and Jeffries-Matusita Distance (JM) as:

$$TTD = \frac{\sum_{i \neq j} \left(2 - TD_{ij}\right)}{2} \qquad [6\text{-}7]$$

$$TJM = \frac{\sum_{i \neq j} \left(2 - JM_{ij}\right)}{2} \qquad [6\text{-}8]$$

where TTD is the Total Transformed Divergence and TJM is the Total Jeffries-Matusita Distance

7 HSI instrumentations @ DCMT

There are several types of devices that can be used for measuring optical power level quantitatively. Domestic digital photo is more suitable for making images than quantitative measures of optical power. The two commonly used devices for hyperspectral analysis are thermal detectors and photoelectric detectors. Thermal detectors measure the heat generated by the absorption of radiant energy. Photoelectric detectors convert incident light to electrical signal. In this thesis, all data has been captured by photoelectric based instruments and therefore the fundamental property of photoelectric detectors will be briefly described here.

7.1 Photoelectric detectors

Photoelectric detectors are based upon quantum mechanics principle. Photon energy collected by the detector excites electrons from the valence band to the conduction band where they become the charge carriers and raise the conductance of the detectors. The photon energy is given by

$$E = \frac{hc}{\lambda}$$

$\mathrm{E} = \text{the photon energy}$
$\mathrm{h} = \text{Planck's constant}$
$\mathrm{c} = \text{speed of light}$
$\lambda = \text{wavelength of the light}$

[7-1]

7.2 Hyperspectral imaging camera

The advances in semiconductor technology in the last few decades have provided low cost and highly efficient devices such as the Charge Coupling Device (CCD), a type of photoelectric detectors, for hyperspectral applications. There are three main variants of cameras available from the commercial-off-the-shelf that are small and relatively low cost (<$100K) for hyperspectral imaging (HSI) application (Fisher et al., 1998). Most of them records hyperspectral data by dispersing the incoming light into its constitute wavelength, and then these wavelengths are capture by standard CCD camera. The

only difference between these three types of camera is the technique used to disperse the light.

The first type of camera captures hyperspectral data by passing the incoming light onto a transmission holographic grating such as that built by Kaiser Optical System Inc (Kaiser Optical Systems, 1994) as shown in Figure 7-1. Then the light is dispersed in a spectrum form which is then capture by a CDD camera. The second type of HSI camera uses a prism-grating-prism spectrograph which allows direct light dispersion. Figure 7-2 shows a design of such camera called ImSpector™ (Aikio, 2001) which is manufacture by Spectral Imaging Ltd. of Finland. The third type of camera is designed using standard reflective surface gratings without any proprietary hardware. Offner diffraction method (Davis et al., 2002; Bowles et al., 1998) was chosen due to its low distortion, high quality and its simplicity. In an Offner Imaging Spectrometer, as shown in Figure 7-3, incoming light that passes through the input slit is reflected by a mirror. Then reflective light is collect and focused onto the reflective grating by a collimating mirror. The grating disperses light into spectrum and it is then focused by a collimating mirror and to project the wavelength dispersed light onto the y-direction of the CCD sensor.

One of our visible to near infra-red (VNIR) cameras has been an Offner type camera which uses reflective type of grating providing a higher throughput that a transmission type of grating. The Offner hyperspectral camera is built and assembled by Headwall Photonics Inc. with a 0.040mm slit and it is then coupled with a standard CDD camera made by the PCO Germany.

To form a hyperspectral image cube as shown in Figure 2-1, a mirror scanner as shown in Figure 7-4 is normally placed in front of the camera lens because the CCD can only capture a line of image with multiple wavelengths for each scan. Typically, pixels on the x-direction of the CCD store the spatial information (the x-axis of the image cube) and pixels on the y-direction of the CCD store the spectral information (the z-axis of the image cube). The mirror is attached on a moving magnet motor and lines of images are then collected by rotating the angle of the mirror. Finally, the lines of images are put together on the y-axis to form the image cube.

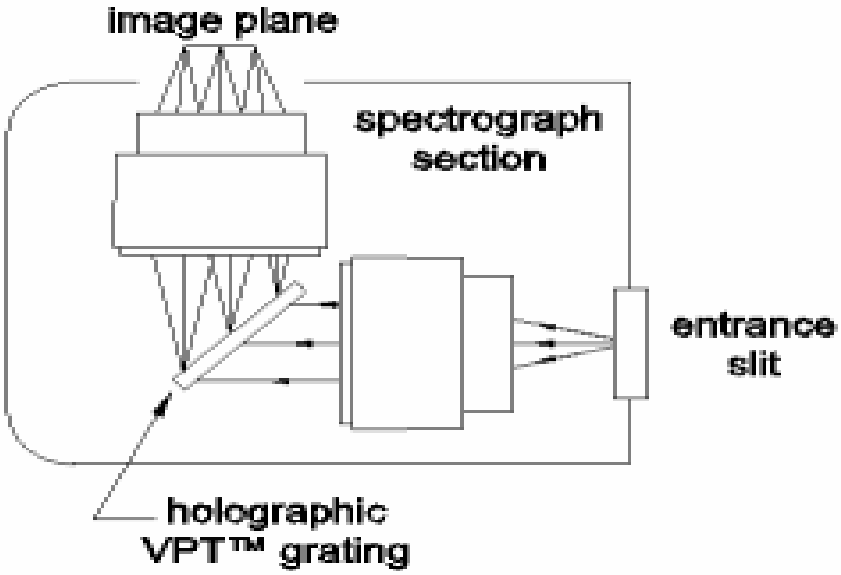


Figure 7-1: Holospec™ Spectrograph

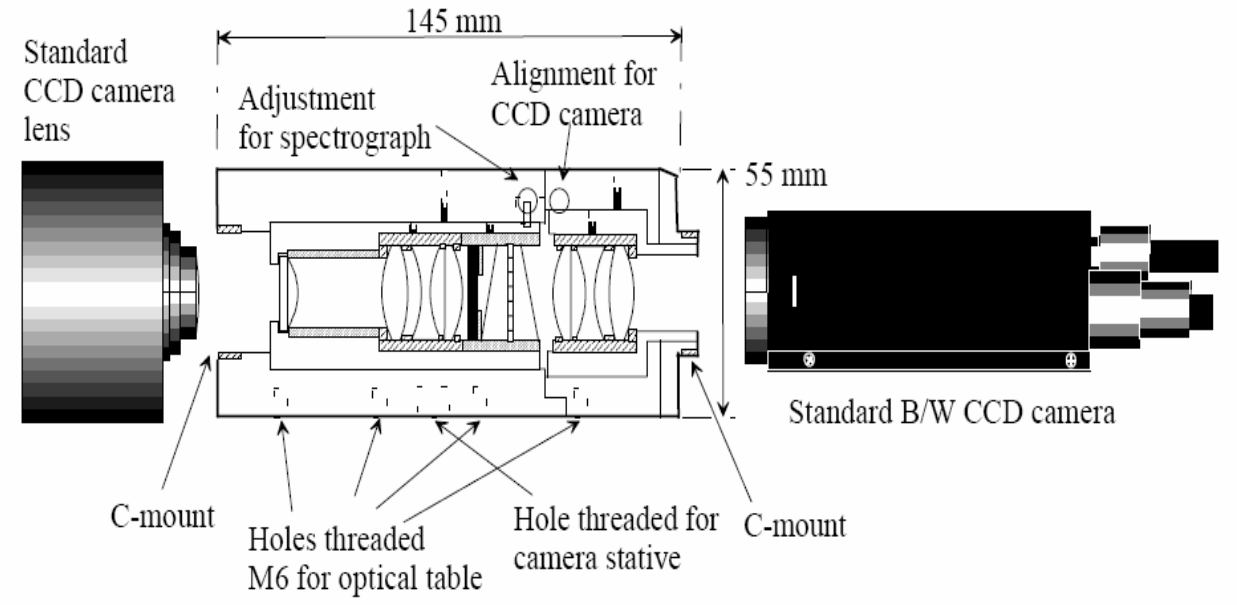


Figure 7-2: Diagram of the ImSpector™ camera

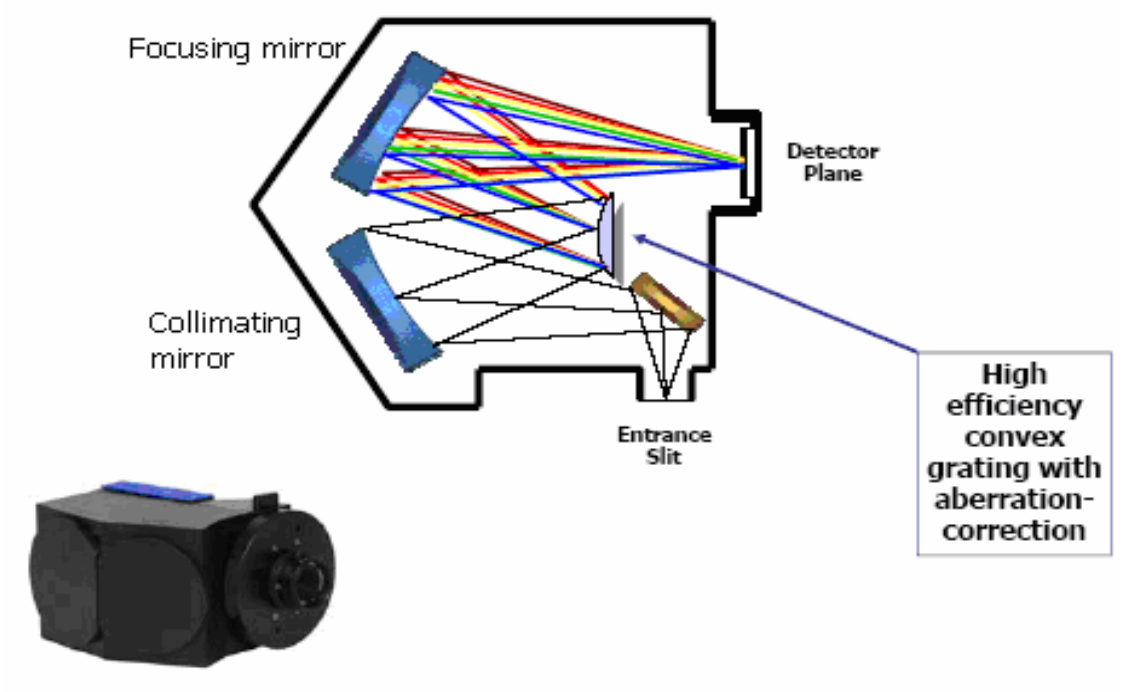


Figure 7-3: Diagram of an Offner Imaging Spectrometer and photo of the Headwall Photonics' built camera Hyperspec™

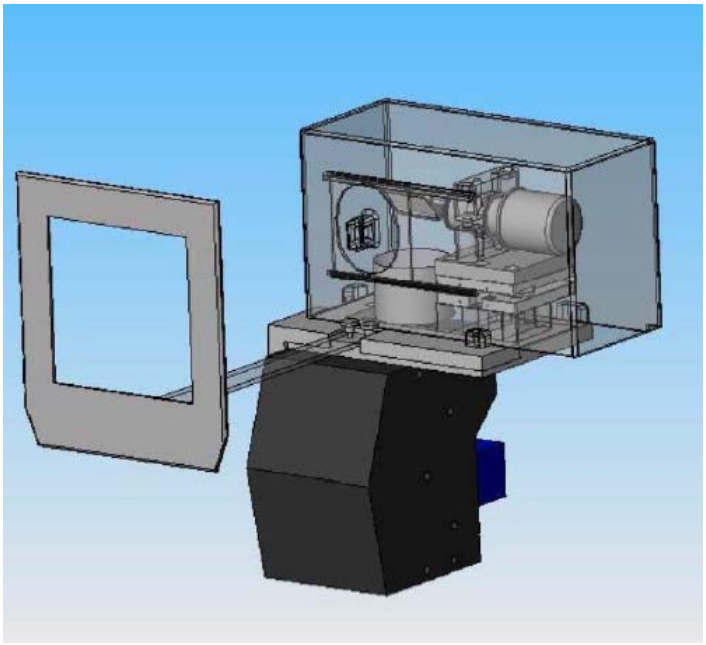


Figure 7-4: A mirror scanner design of the hyperspectral camera by Headwall Photonics

7.2.1 Calibrations of the hyperspectral camera

It is essentially important to check the spectral and radiometric calibration of the hyperspectral camera before it is deployed for field measurement. All calibration procedures was carried out in our laboratory using various optical instruments consisting of helium-neon (HeNe) laser, a sodium lamp, an Ocean Optics S200 spectrometer, a photometer and other optical equipments.

7.2.2 Spectral Calibration

The relationship between the wavelength and the registered pixel channel in the y-direction of the CCD array is expected to be linear according to the manufacture's calibration. The first experiment is to verify this relationship. One method is to use multispectral gas emission lamp or alternatively a broad band lamp together with a monochromator to output several known wavebands of light for spectral calibration. In this experiment we have used various light sources such as He-Ne laser, sodium gas discharge lamp and fluorescent light. The complete spectral characteristic of each light source is firstly measured by the spectrometer in the range of 400nm-900nm. The camera is set such that no spatial binning and all 1024 spectral channels have been used. Each experiment is repeated 50 times and they are then averaged to reduce the noise, subsequently the spectra recorded by the camera are then compared with that taken by the Ocean S200 spectrometer.

The laser experiment was performed by a 632.8nm (red) HeNe laser. The intensity of the laser was attenuated by filters and a beam splitter. Shown in Figure 7-5 is the spectra of the laser as recorded by the spectrometer and Figure 7-6 is the spectral response that recorded by the VNIR HSI camera. The spectral responses for both resembles to Gaussian like with full width at half maxima of about 4nm width. Results of the sodium lamp and background light from the spectrometer are shown in Figure 7-9 and Figure 7-12, respectively. These spectra exhibit several characteristic peaks in the 400-900nm range and they correspond well to that as recorded by the camera as shown in Figure 7-10 and Figure 7-13.

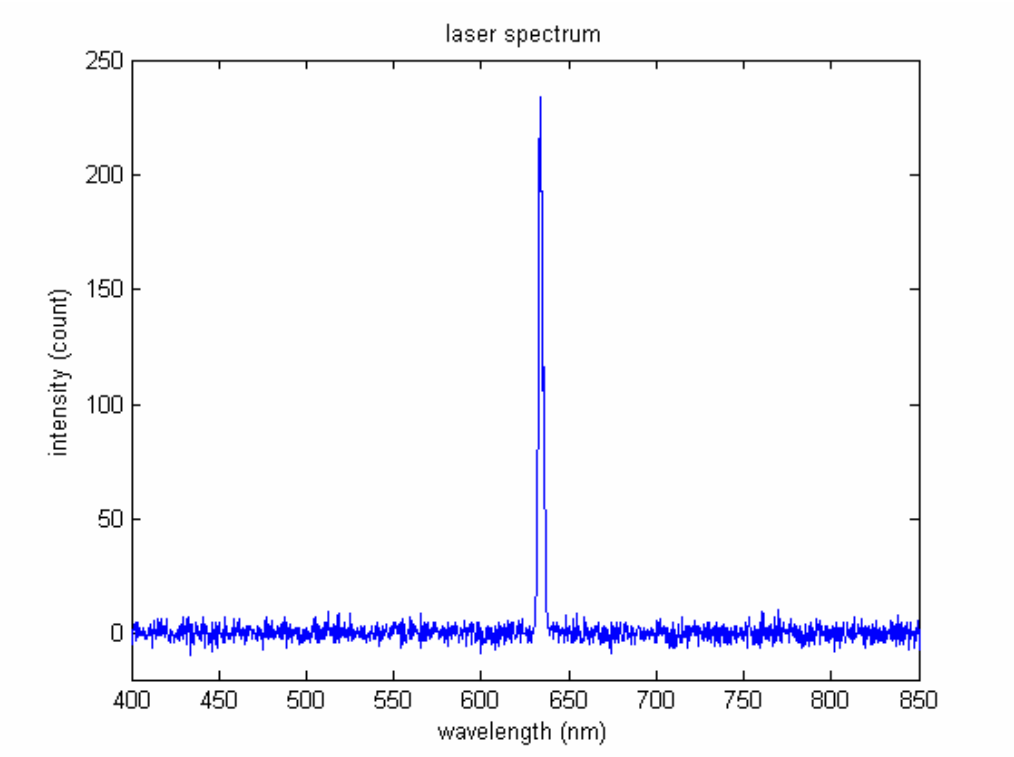


Figure 7-5: Spectral measurements of the He-Ne laser recorded by the spectrometer

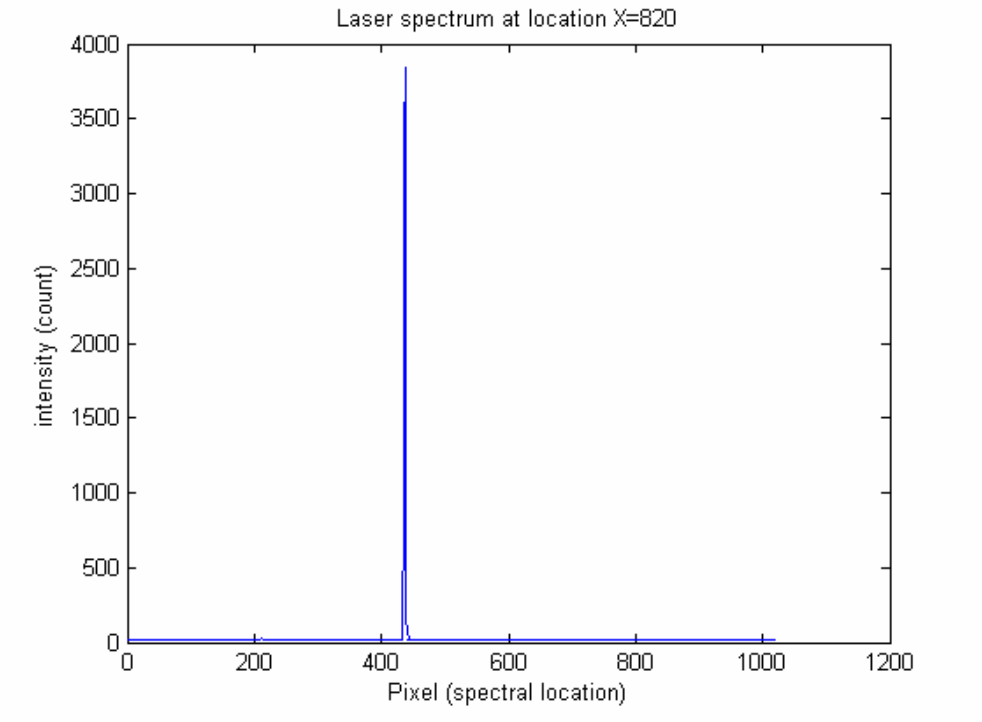


Figure 7-6: Spectral measurements of the He-Ne laser recorded by the camera

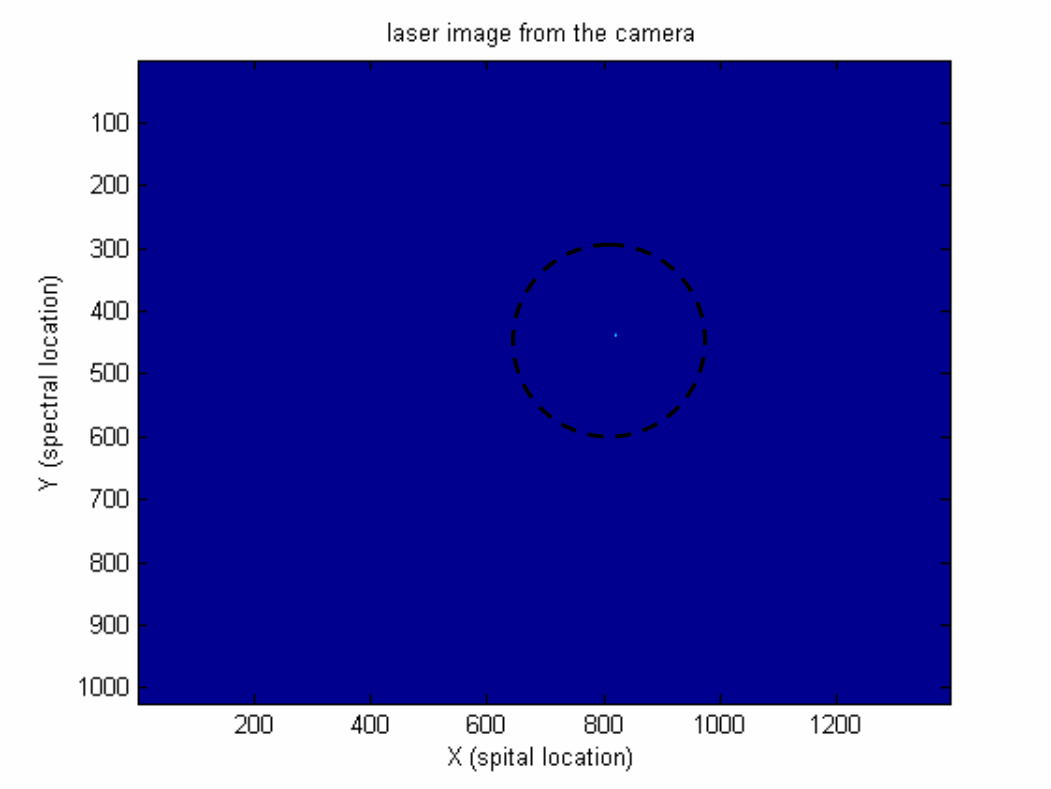


Figure 7-7: Spectral (y-axis) /spatial (x-axis) false colour image of a He-Ne laser dot (circled) as recorded by the VNIR HSI camera

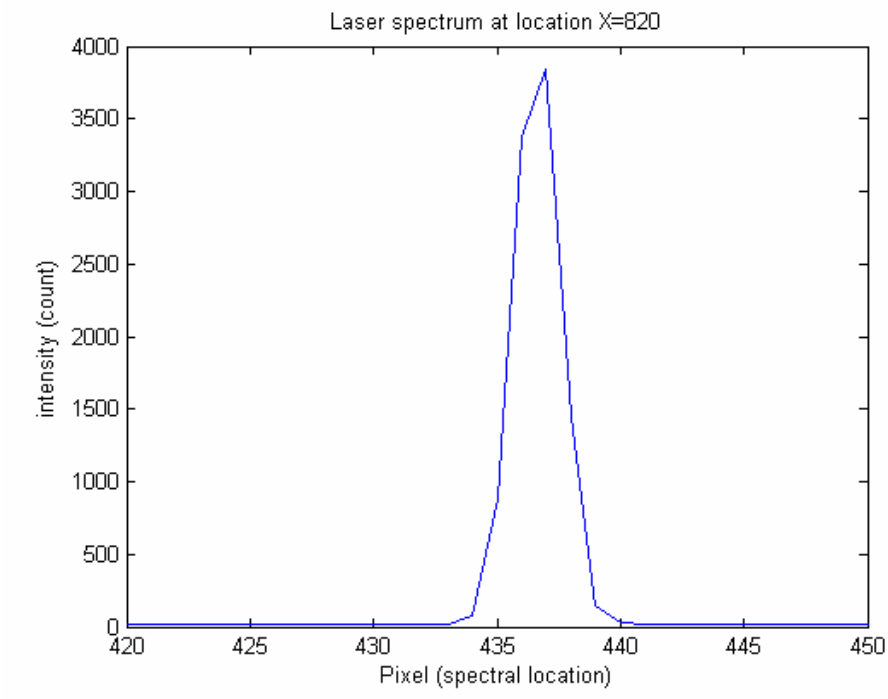


Figure 7-8: Spectral profile of the He-Ne laser dot as recorded by the VNIR HSI camera

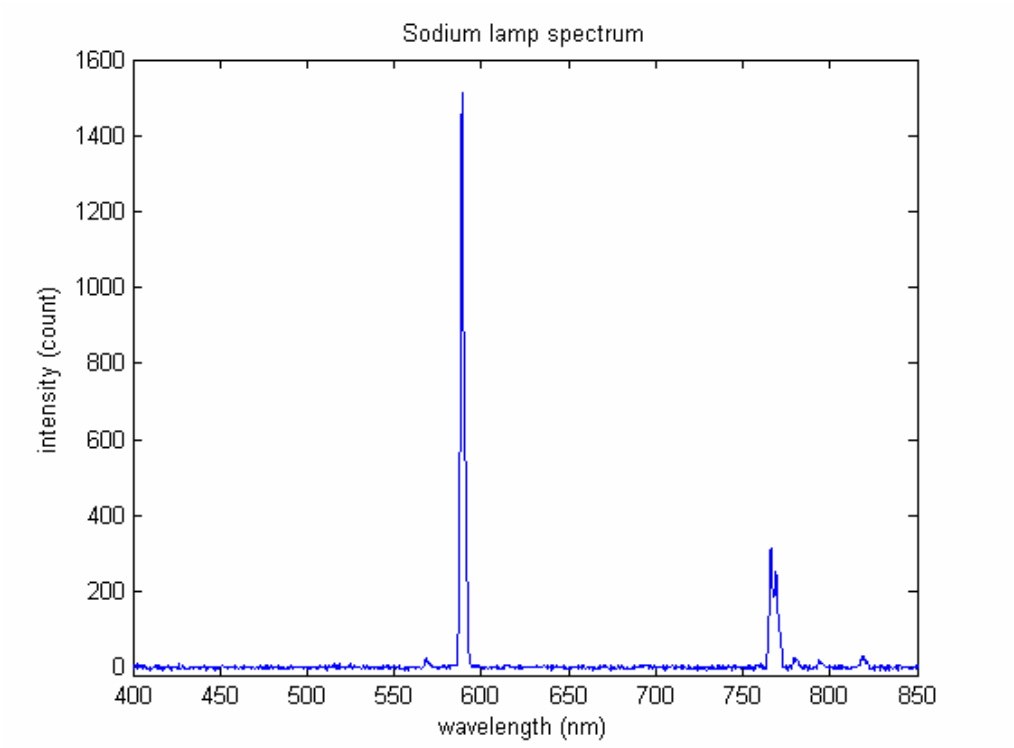


Figure 7-9: Spectral profile of the Sodium lamp that recorded by the S200 spectrometer

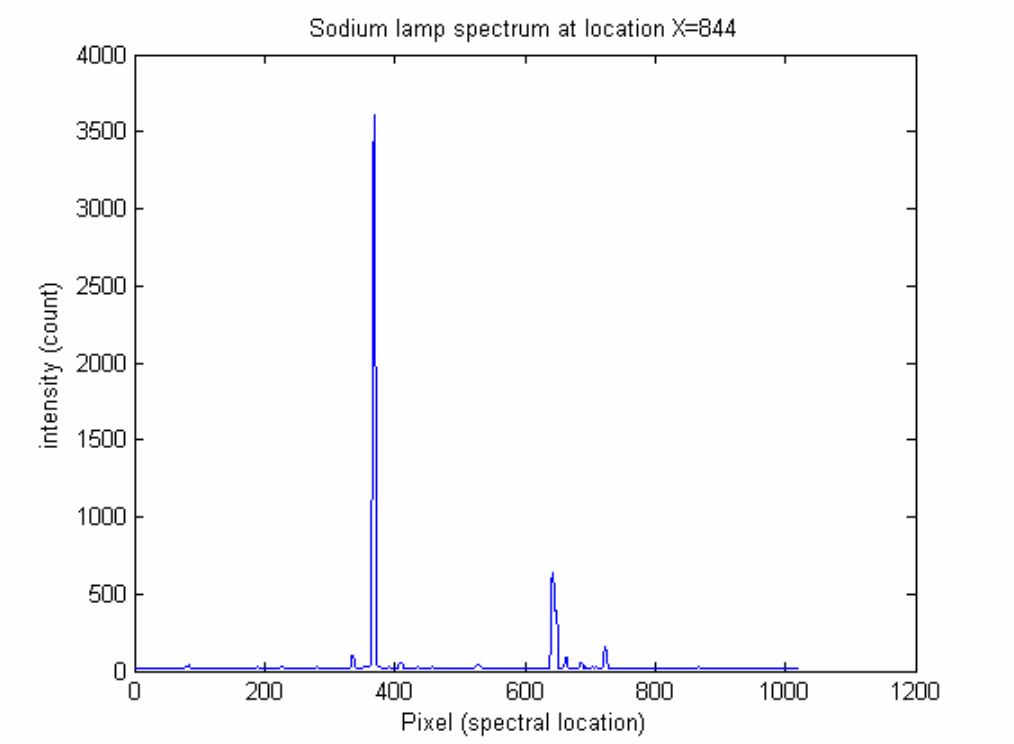


Figure 7-10: Spectral profile of the Sodium lamp as recorded by the camera

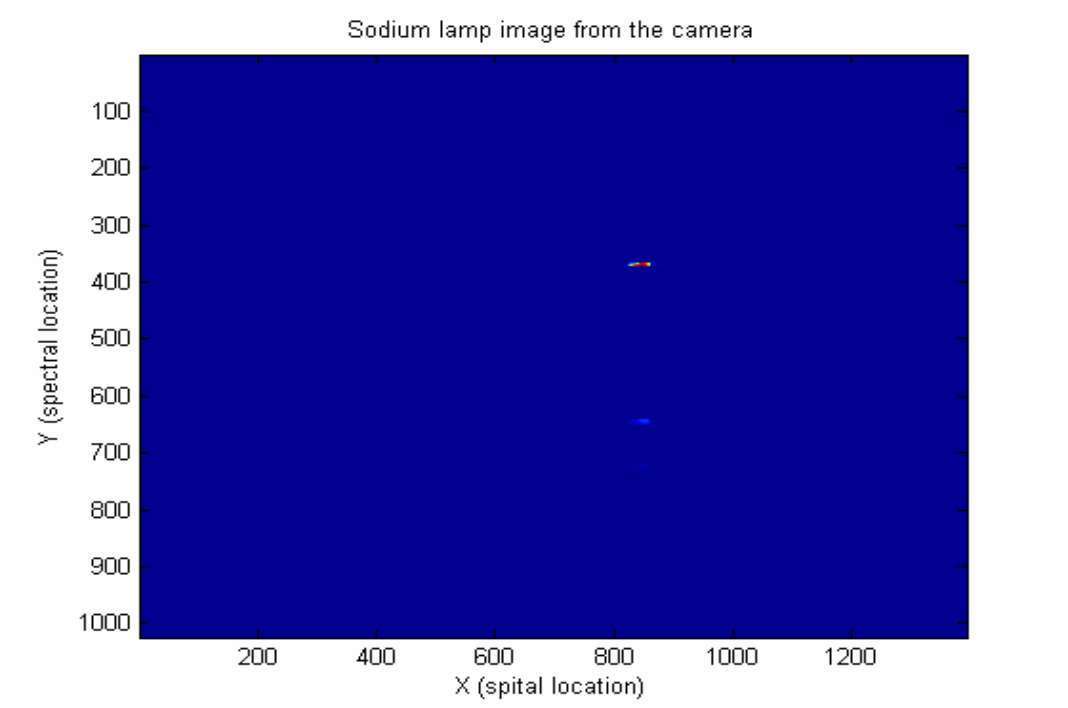


Figure 7-11: Spectral/spatial of a line of false colour image showing a spot of the Sodium lamp source as recorded by VNIR HSI camera

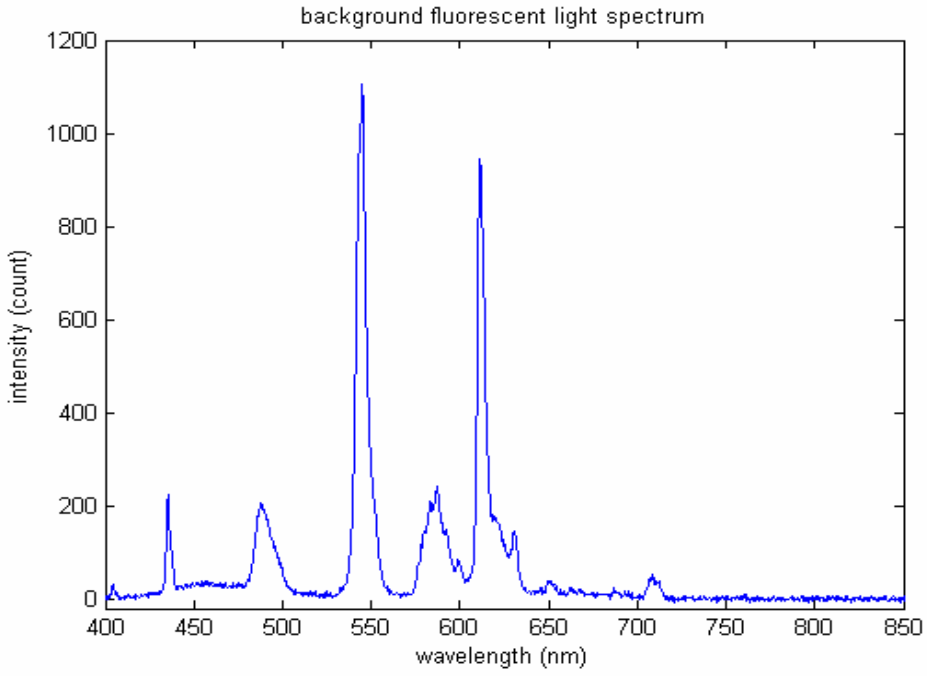


Figure 7-12: Spectral profile of the background fluorescent light as measured by the spectrometer

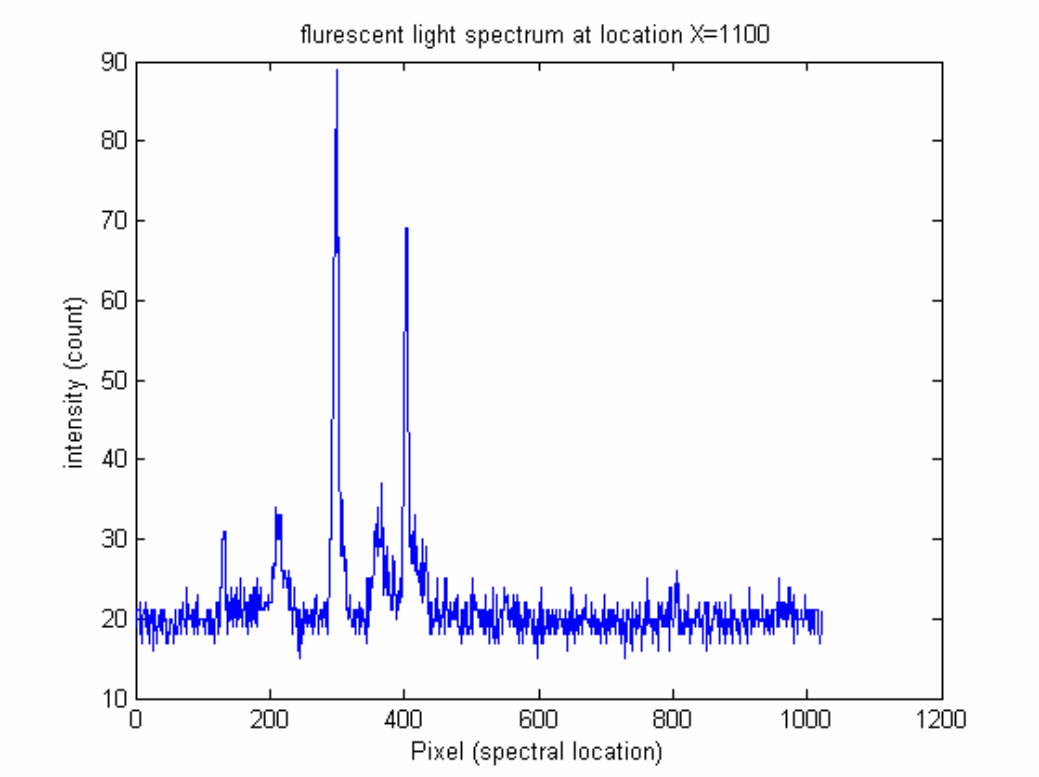


Figure 7-13: Spectral profile of the background fluorescent light as recorded by the VNIR HSI camera

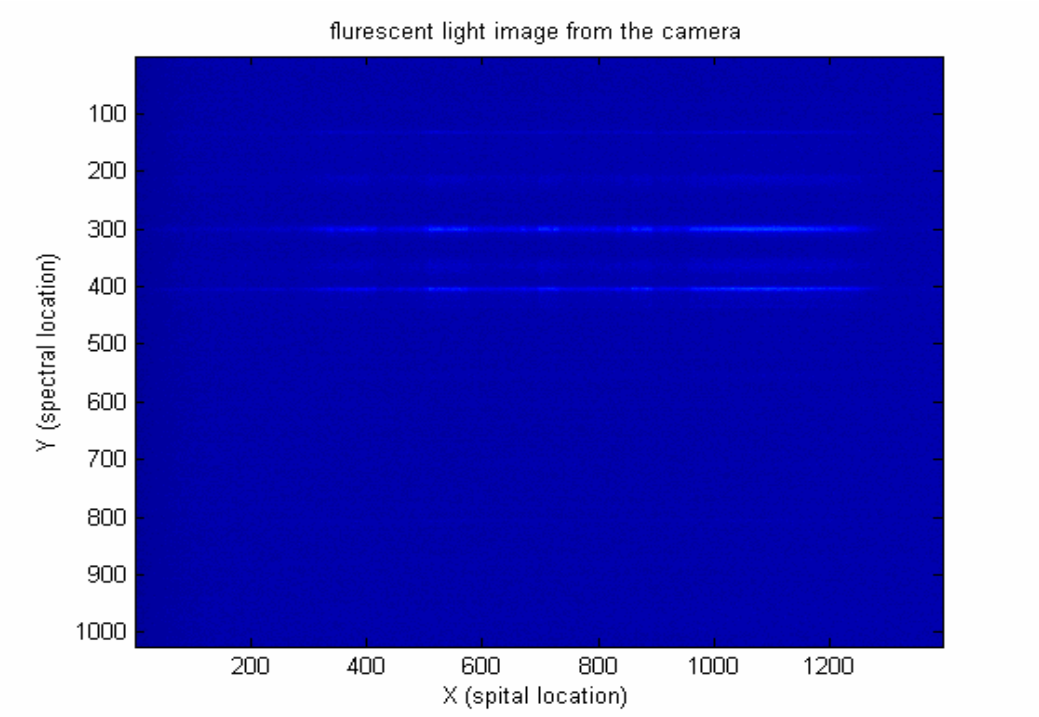
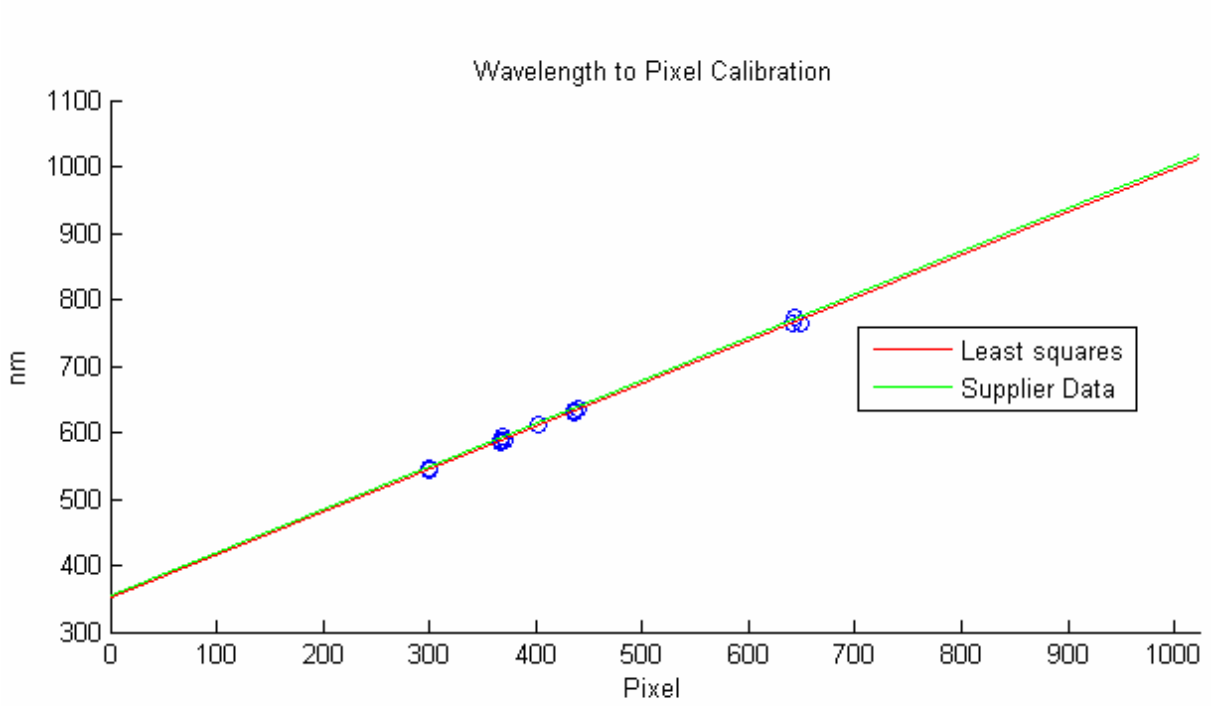


Figure 7-14: A line of spectral/spatial false colour image of the background fluorescent light as recorded by the VNIR HSI camera

The relationship between the pixel channels of the CCD with respected to the wavelength that measured by the spectrometer for the above light sources can be plotted out as shown in Figure 7-15. A linear relationship is found without any quadratic and higher term which agrees well with the manufacturer's calibration. The equation of the plot is derived using both least-square error and robust regression method. The gradient of the plot is the average pixel dispersion of camera and it is found to be at around 0.643nm/pixel. The gradient lies between 0.64nm/p-0.65nm/p with an average of 0.646nm/p being very close to that of the manufacture's calibration. The offset however has been found to be 352.575nm which is quite different from the supplier's value of 356.31627.



Least squares: Wavelength = 352.575 + 0.643125*Pixel RMS error = 2.0904

Supplier Data: Wavelength = 356.316 + 0.646*Pixel

Figure 7-15: Wavelength to Pixel calibration plot deduced in this work

7.2.3 Radiometric calibration

For applications such as target detection and classification, the exact radiometric calibration of the image pixels is not always necessary as long as the data are self-consistent. However other applications involving the use of physics quantities, such as the atmospheric correction by employing radiative-transfer models and the derivation of the abundances of elements or compounds within materials, will require that the data to be calibrated to standard units of measurement (Bowles et al., 1998).

Radiometric calibration has not been an easy task due to the artefacts such as non-uniform illuminations which are hard to estimate in practise. Furthermore, the sensitivity of individual pixels and their spectral sensitivity responses across the CCD sensor may vary as shown in Figure 7-16.

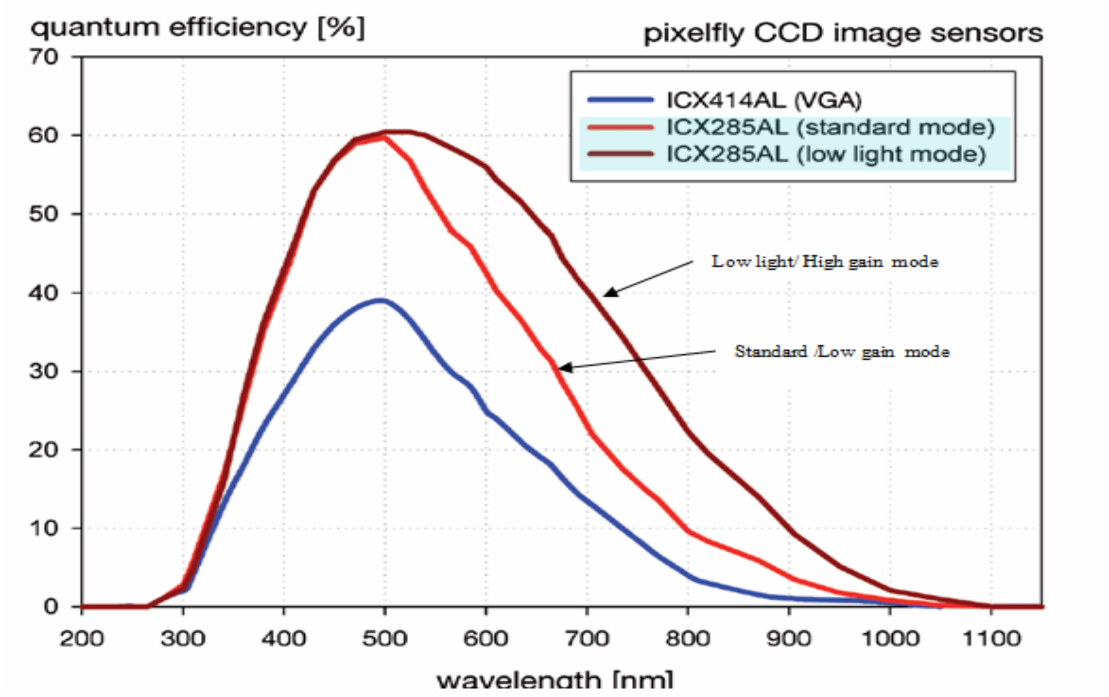


Figure 7-16: Spectral sensitivity of the VNIR HSI sensor (extracted from the COOKE Corporation PixelFly manual)

One method for the radiometric calibration of hyperspectral camera is to make use of an integrating sphere conforming to UK National Physical Laboratory (NPL) or US National Institute of Standards and Technology (NIST) standards (Davis et al., 2002)(Bowles et al., 1998). Typical light source is halogen lamp due to its broadband nature and the photon fluxes at each wavelength of this lamp have been well studied. The intensity of the output from the sphere at various light levels is determined by performing a transfer

calibration and data is taken with a range of intensities by using ND filters. The main drawback of this kind of setup is that the light sources may have to be recalibrated from time to time.

In this experiment the radiometric calibration is performed using a He-Ne (623.8nm) laser, a photometer, the hyperspectral camera together with some filters and optics as shown in Figure 7-17. The laser beam is passed through a series of filters before it enters the beam splitter, which is then directed to the camera at point A and the other is simultaneously detected by the S200 photometer at point B. The ratio of the two beams has been pre-calibrated using the S200 situated at the two points A & B for a range of beam intensities as shown in Figure 7-18. The linear relationship is calculated using both least-squares and robust regression methods.

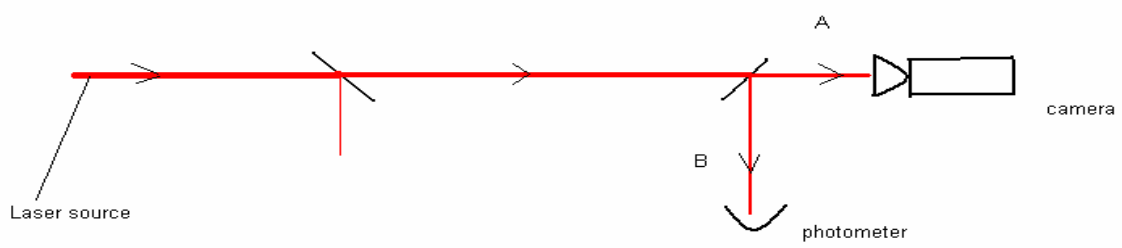
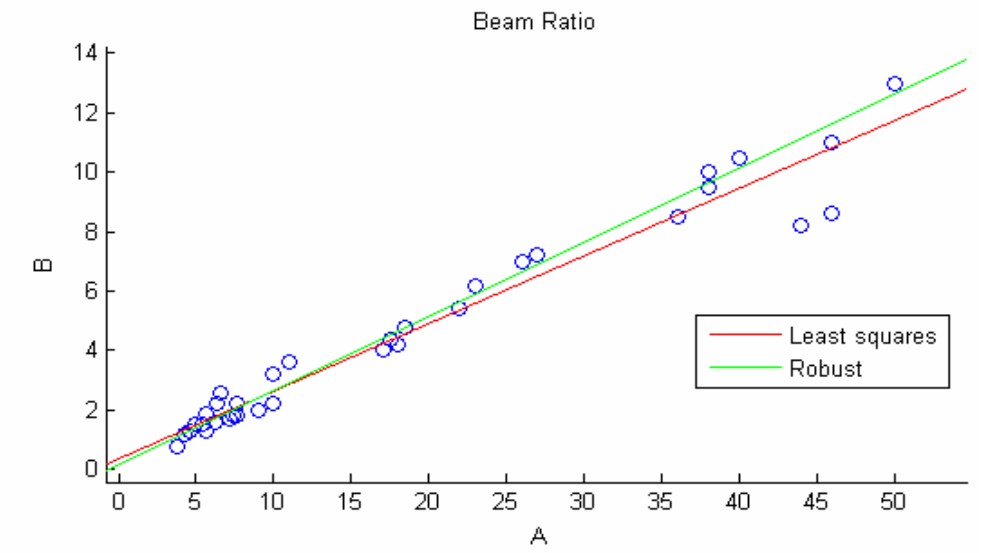


Figure 7-17: The experiment setup for the radiometric calibration in this work



Least squares:	A = 0.401581 + 0.227303*B	RMS error = 0.743634
Robust:	A = 0.160712 + 0.249608*B	RMS error = 0.465241

Figure 7-18: The intensity ratio of the beam splitter employed in this study

The S200 photometer measures in foot-lambert which can be converted into luminance using the following equation:

$$\text{Luminance}\left(\text{lm.sr}^{-1}.\text{m}^{-2}\right) = Footlambert * \frac{1}{0.3048^2 \pi} \qquad [7\text{-}2]$$

However, the conversion between photometry unit and radiometry unit is not trivial. According to the definition one watt of monochromatic green light (555 nm) equals to 683 lumens (lm) and the relationship between watt and lumen is wavelength dependent as shown in Figure 7-19. The equation for converting luminance to radiance is given by:

$$\text{Radiance}\left(\text{W.sr}^{-1}.m^{-2}\right) = \frac{\text{Luminance}}{\left(683 * \text{transfer ratio}(\lambda)\right)} \qquad [7\text{-}3]$$

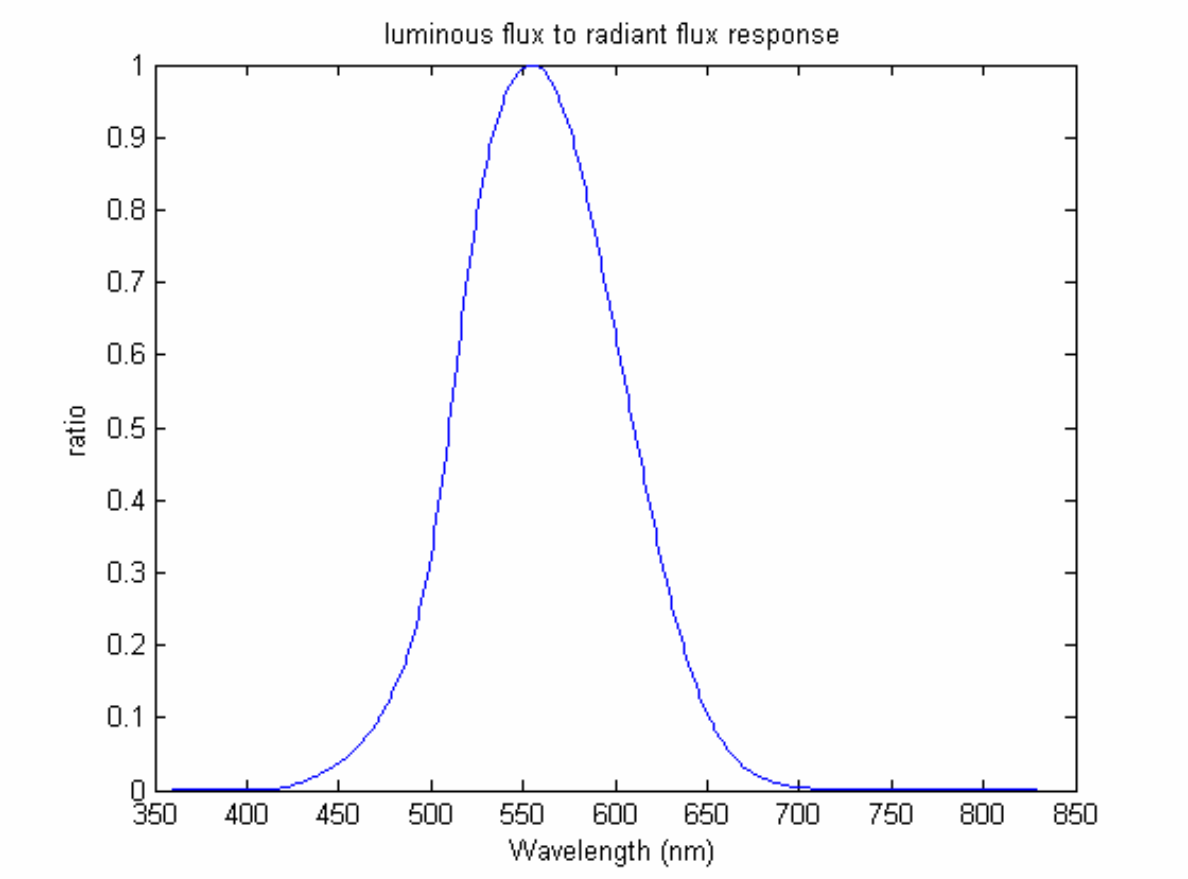


Figure 7-19: The transfer ratio between photometry and radiometry

To minimise the errors due to the sensor noise the HSI camera is kept at a constant temperature during the calibration and the dark current is taken after each measurement is made. 1024 frames of dark measurements have been averaged to represent the mean dark current, and the radiometric measurement is performed by averaging 50 frames of data less the averaged dark current frame. The experiment is repeated with different integration time of the camera until the intensity of the laser spot as measured by the camera reaches to its maximum count of 4095 (Figure 7-20). The relationships at 9 footlambert and 45 footlambert using linear least-squares fits are shown in Figure 7-21 and Figure 7-22 respectively.

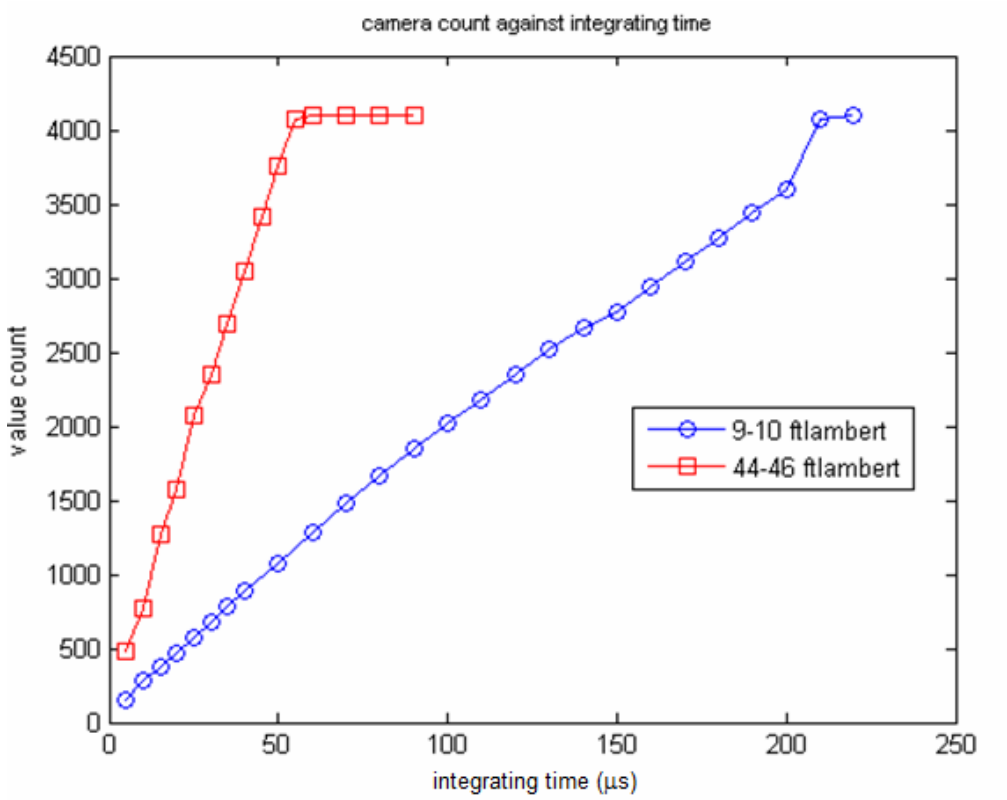


Figure 7-20: The HSI camera count against integrating time for two different beam intensities

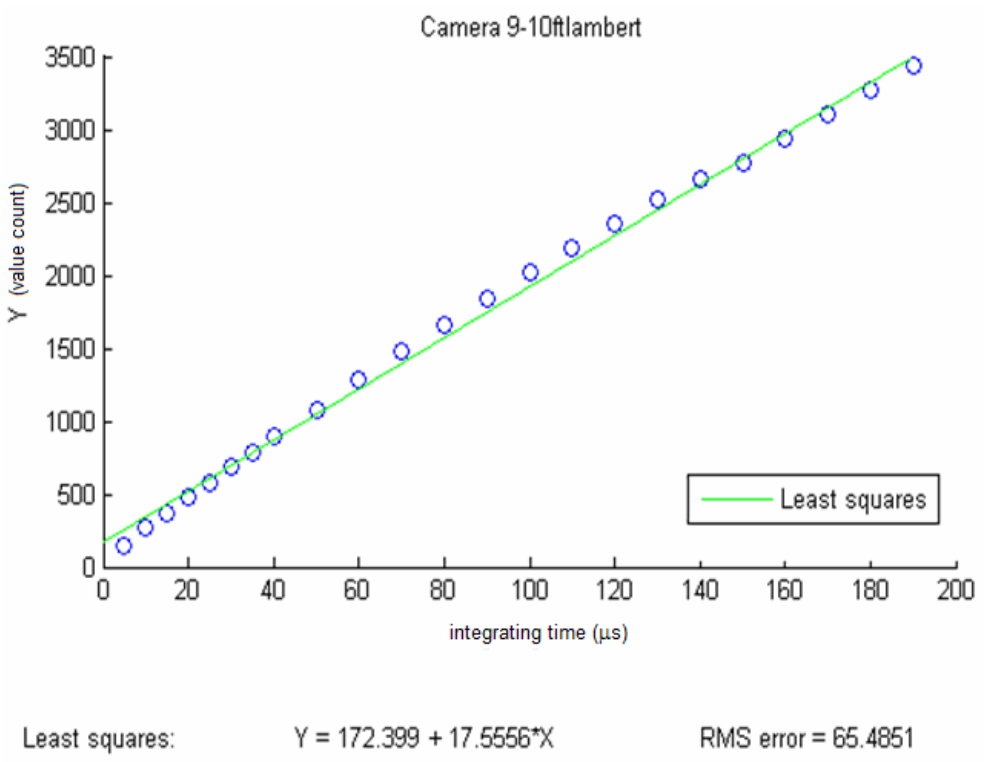


Figure 7-21: The relationship plot at around 9-10 footlambert

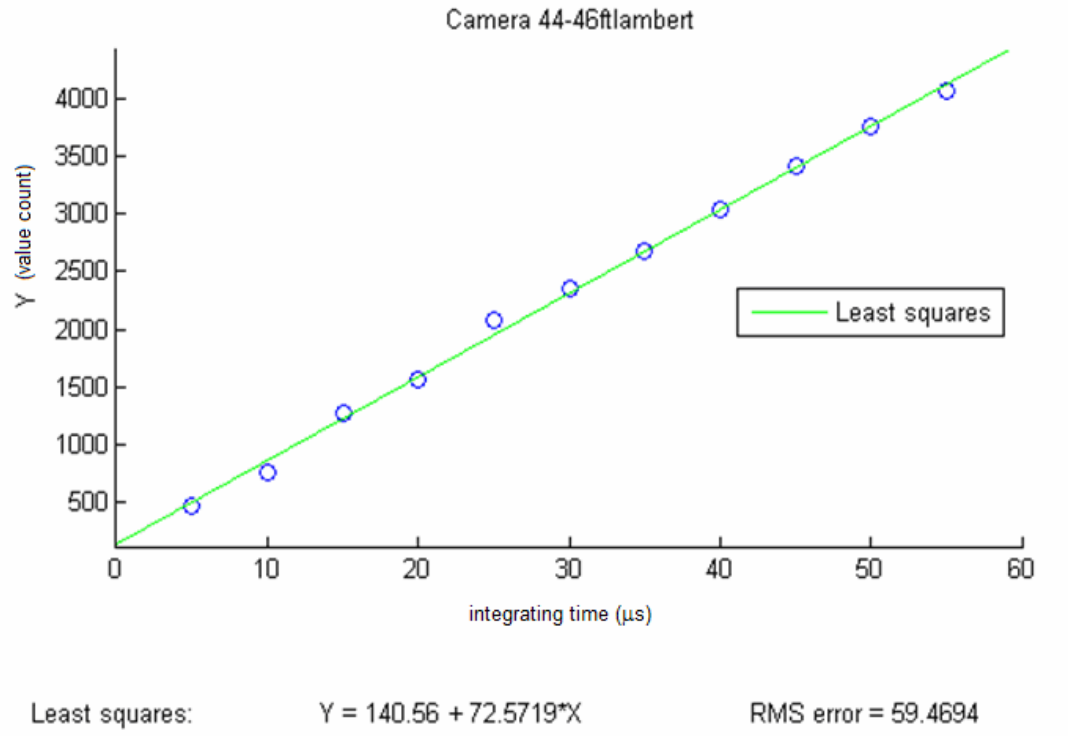


Figure 7-22: The relationship plot at around 44-46 footlambert

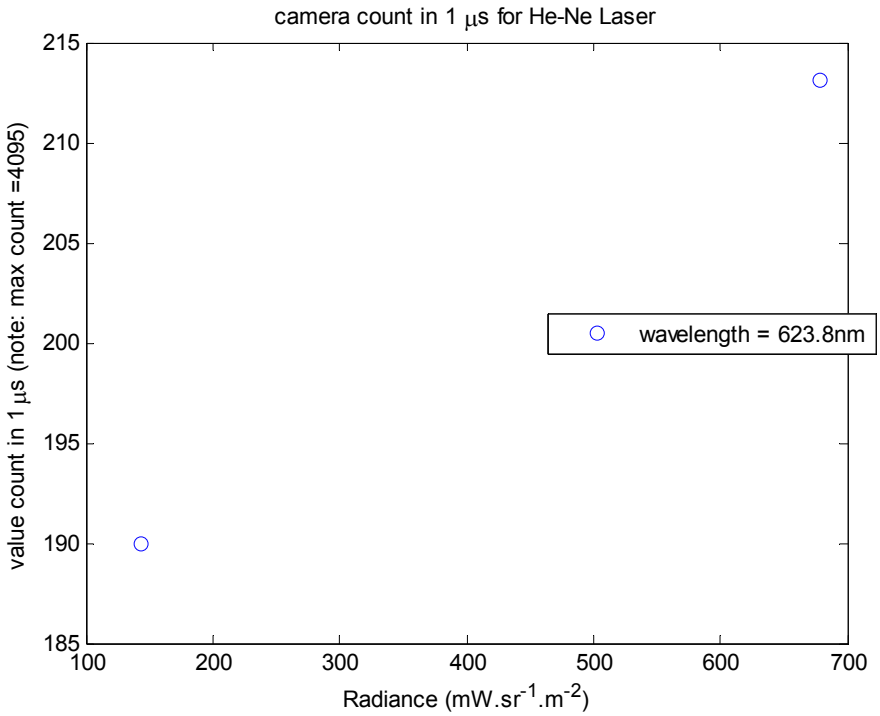


Figure 7-23: A graph showing the camera counts to radiance relationship

Assuming the light transmission efficiencies from the lens of the camera to the CCD sensor are constant at all wavelength, the digital value count to radiance relationship at other wavelength could be estimated from the quantum efficiency plot shown in Figure 7-16. However, in practice, the assumption is normally not true and transmission efficiencies are wavelength dependent. Due to the limited available equipments at the time of experiment, the radiometric calibration was performed at only the laser wavelength, and therefore the result was not sufficient to establish true relationships between the digital value count and radiance for all other wavelengths.

8 Hyperspectral data set

There have been two common approaches for the evaluation of classification performance in HSI research: one is the use of real data set which has been carefully ground-truthed and the other is the use of synthetic data (Landgrebe, 2005). However, most simulated HSI data sets have been far from 'realistic' because the estimated distribution may not always truly characterise the actual scene, and therefore, we have employed real data sets throughout this study.

8.1 Data set 1: Barrax set

The first experimental data set is taken by ESA/DRL in Barrax, Spain in 2000 at an altitude of 4km and it consists of 128 bands in the spectral range of 0.403um to 2.48 um. This data set is collected using the Hymap (Hyperspectral Mapping) instrument at around noon. The data is also geometric rectified to remove any artefacts due to the platform (plane) movement. Patches of the data set have been ground truthed and a target map has been drawn manually based on the land use map together with the partial ground truth data.

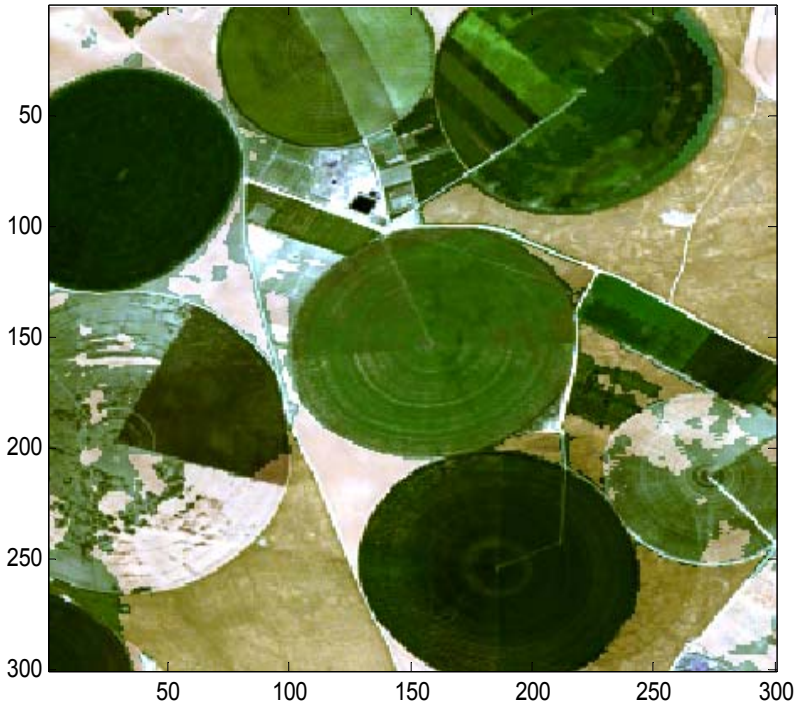


Figure 8-1: RGB image of the Barrax hyperspectral data taken at 4km range equivalent to a ground sampling distance of 3m per pixel.

8.2 Data set 2: Manchester data set

This data set was taken in Minho region of Portugal under daylight in the clear sky mid-morning of a summer in 2000 (Nascimento et al., 2002). The original data consists of 33 bands ranging from 400nm to 720nm and due to the low signal to noise ratio of the first and the last wavebands, they are discarded and leaves thirty one useable bands for analysis. The RGB image of the data set is depicted in Figure 8-2 and a typical classification using unsupervised K-means algorithm for 20 classes is shown in Figure 8-5. The test data and the training data sets are selected using similarity measures as depicted in chapter 6 and in this case the spectral angular mapper and Euclidean distance have been employed for assessing the pair-wise class similarities amongst the 20 classes. For a pair of classes with similarities below a preset threshold they are merged together, resulting in a 16-class data set with appreciable dissimilarities. The test data set and the training data are selected from homogeneous areas as shown in Table 8-2. The ground truth location map is shown in Figure 8-3. According to Equations 6-7 & 6-8 the total TTD and TJM scores of the training data selected from these 16 classes are shown in Table 8-1 and the pairwise scores are presented in Figure 8-6. These are the ‘best’ dissimilarity scores and all classification results will be equal or worse (higher) than these base line values.

TTD score	TJM score
0.0831	0.3059

Table 8-1: The TTD & TJM scores for the 16-class training data

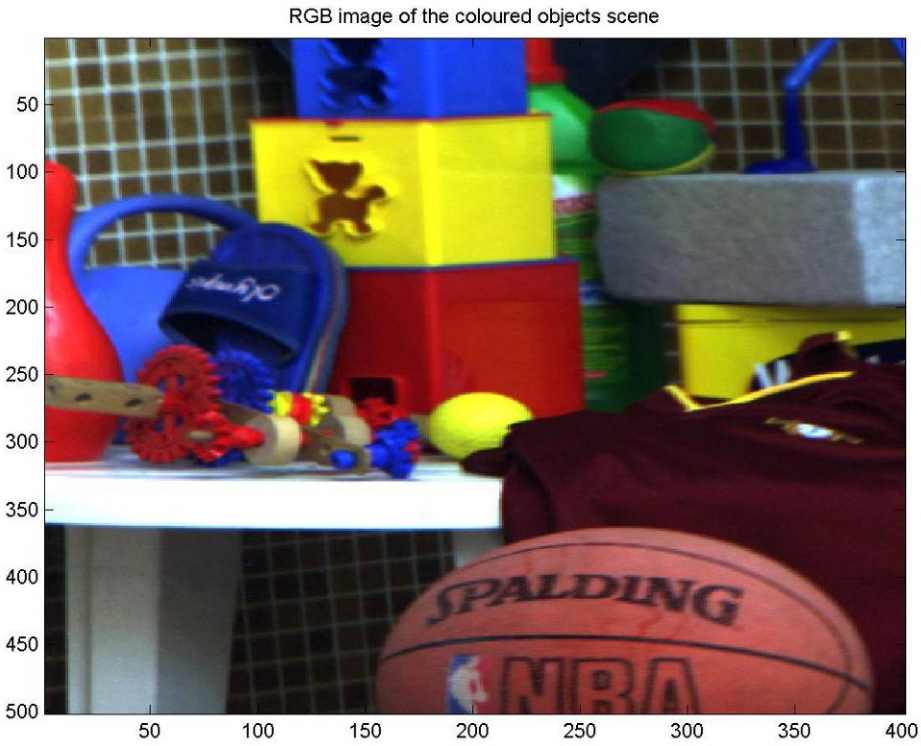


Figure 8-2: RGB image of the Manchester HSI data set

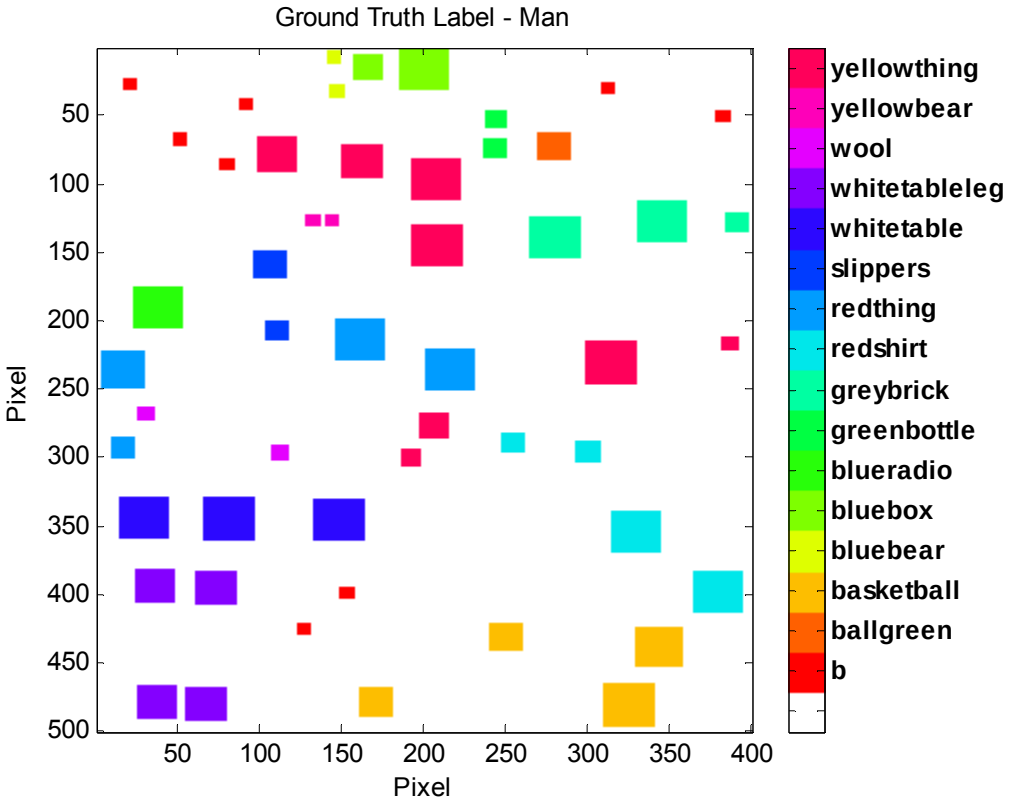


Figure 8-3: The ground-truthed map of the man data set.

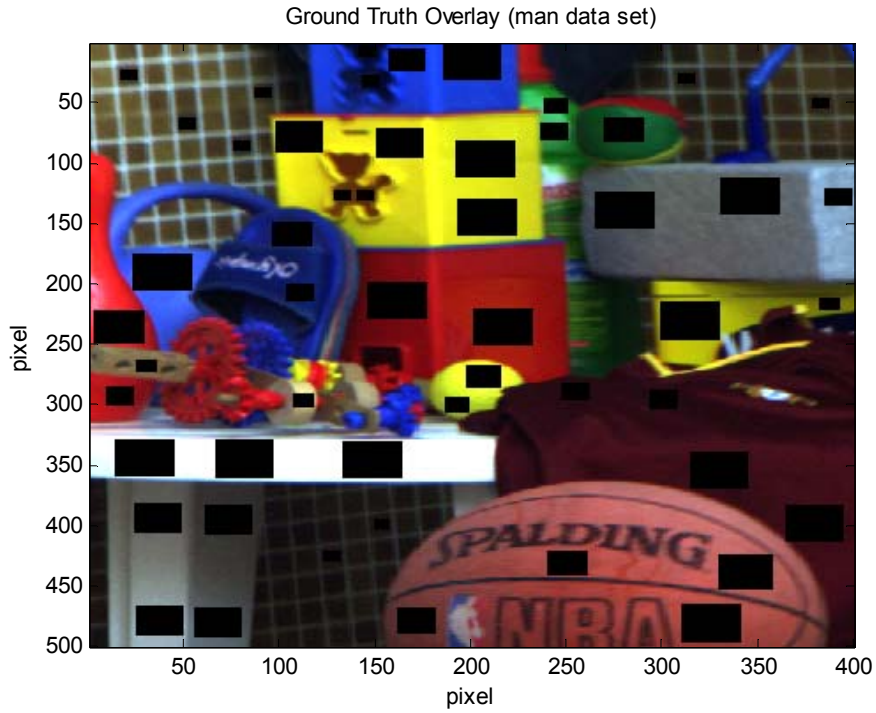


Figure 8-4: The ground-truthed overlay map of the man data set.

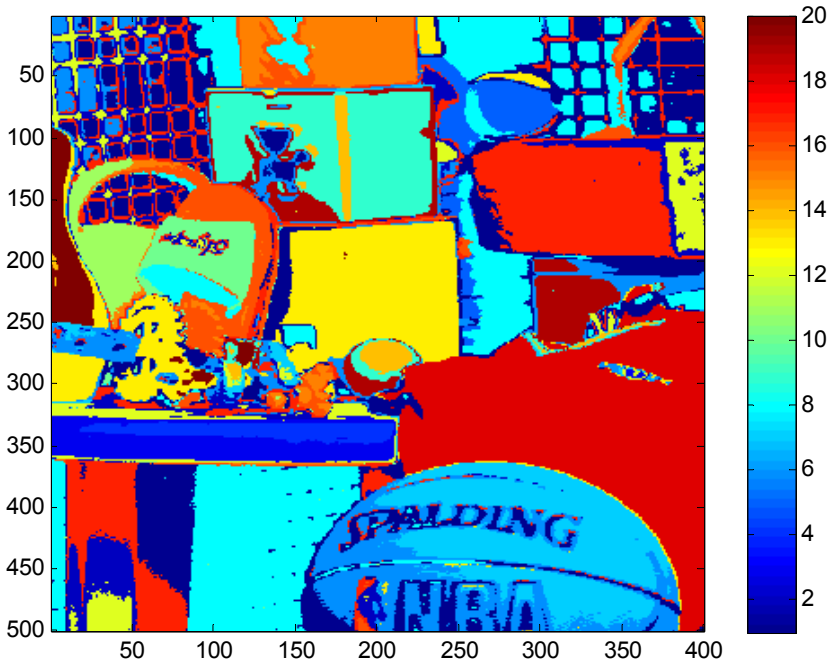


Figure 8-5: The 20-class clustering result by using k-means for the Manchester data set that presented in Figure 8-2.

Class	Class name	Description	Data Size	No. of Bands
1	'b'	Wall tiles	648	31
2	'ballgreen'	Green on the brick	441	31
3	'basketball'	Basketball	2684	31
4	'bluebear'	The bear pattern of the blue box	162	31
5	'bluebox'	The blue box (excluding the bear pattern)	1322	31
6	'blueradio'	The blue radio behind the slippers	961	31
7	'greenbottle'	The green parts of the bottle which is behind ‘ballgreen’ and all the boxes	394	31
8	'greybrick'	The grey brick	2147	31
9	'redshirt'	The red football jersey	2372	31
10	'redthing'	All reds (including the bowl, the red box and the cap of the green bottle	2876	31
11	'slippers'	The blue slippers	666	31
12	'whitetable'	The white table top	2883	31
13	'whitetableleg'	The legs of table	2500	31
14	'wood'	The wooden parts of the toy which is on the table	242	31
15	'yellowbear'	The bear pattern of the yellow box	162	31
16	'yellowthing’	All reds (including the yellow box (without the bear) and the cap of the green bottle	4784	31
		Total number of pixels	25244	

Table 8-2: The selection of 16-class ROI from the Manchester HSI image as the test and training data set

ground truth TD

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2
4	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2
5	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2
6	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2
7	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2
8	2	2	2	2	2	2	2	0	2	2	2	2	1.916893	2	2	2
9	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2	2
10	2	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2
11	2	2	2	2	2	2	2	2	2	2	0	2	2	2	2	2
12	2	2	2	2	2	2	2	2	2	2	2	0	1.999998	2	2	2
13	2	2	2	2	2	2	2	1.916893	2	2	2	1.999998	0	2	2	2
14	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2
15	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2
16	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0

ground truth JM

Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	2	1.999943	2	2	2	2	1.976844	2	2	2	2	1.928551	1.999798	2	2
2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1.999943	2	0	2	2	2	2	1.999997	2	2	2	2	1.999906	2	2	2
4	2	2	2	0	2	2	2	2	2	2	1.999986	2	2	2	2	2
5	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2
6	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2
7	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2
8	1.976844	2	1.999997	2	2	2	2	0	2	2	2	2	1.790707	1.999999	2	2
9	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2	2
10	2	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2
11	2	2	2	1.999986	2	2	2	2	2	2	0	2	2	2	2	2
12	2	2	2	2	2	2	2	2	2	2	2	0	1.998338	2	2	2
13	1.928551	2	1.999906	2	2	2	2	1.790707	2	2	2	1.998338	0	2	2	2
14	1.999798	2	2	2	2	2	2	1.999999	2	2	2	2	2	0	2	2
15	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2
16	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0

Figure 8-6: The pairwise JM and TD scores for the selected 16-class Manchester data set. (for more information about JM/TD please refer to section 6.4.5 & chapter 12)

8.3 Data set 3: Lab t-shirt

This data set is recorded in the laboratory using various bright colour t-shirts as targets. The HSI data as shown in Figure 8-7 was taken by the VNIR hyperspectral camera under the illumination of halogen lamps. There are ten different t-shirt colours and the RGB image and the ground-truthed map of this data set is presented in Figure 8-8 and Figure 8-9 respectively. The mean spectra of each class is presented in Figure 8-10, highlight the fact that some classes such as the two yellow ones are very similar to each other spectrally. Note that all of the t-shirt data have been converted into reflectance using the ELM techniques with the in-scene calibration panels (black, grey and white spectralons). To avoid complications only the centre part of the t-shirts have been selected for processing and all other pixels that are close to the boundaries between the t-shirts have been discarded (see Figure 8-9). The JM and the TD scores for this data set are found approaching to the theoretical limit of 2, suggesting a large dissimilarity between the classes (Figure 8-11). The TJM & TTD scores for this data set according to Equations 6-7 & 6-8 are zero. Note that the reflectance of the black material in Figure

8-10 dips below 0 at short wavelengths. This is likely due to uneven light illumination on scene which causes the black spectralon to be brighter than the black t-shirt, thus the black t-shirt negative values at short wavelength were attained when they were extrapolated from the ELM.

Figure 8-7: RGB Photograph of the t-shirt data set taken in the laboratory

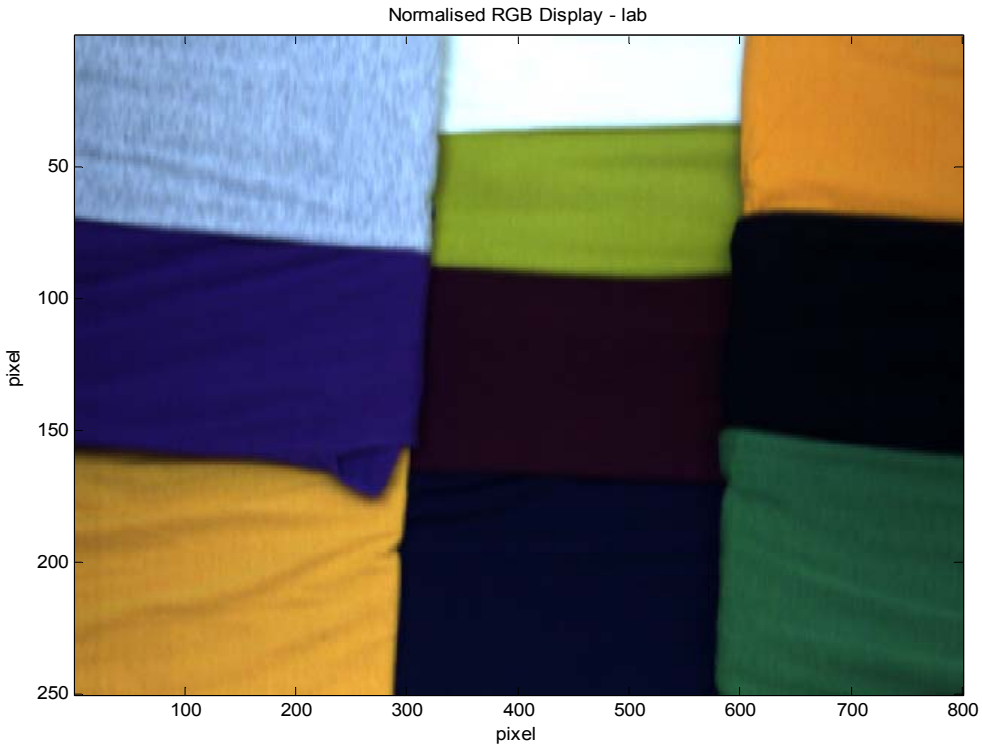


Figure 8-8: RGB model of the t-shirt HSI data

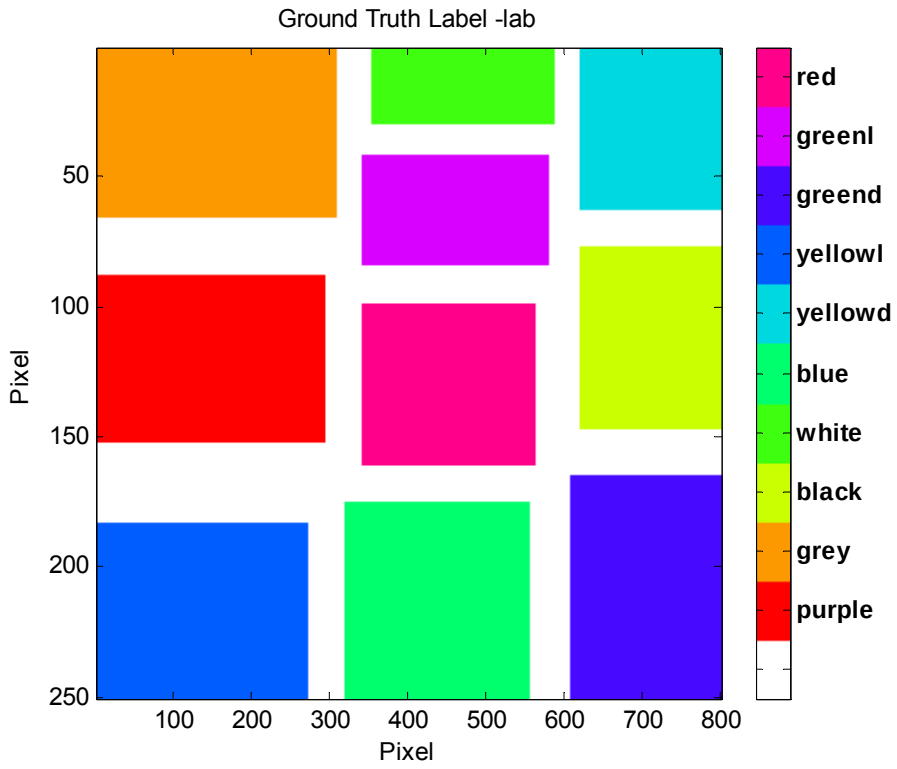


Figure 8-9: The ground-truthed map of the t-shirt data set. Note that the boundaries between the t-shirt have been removed due to the shadows.

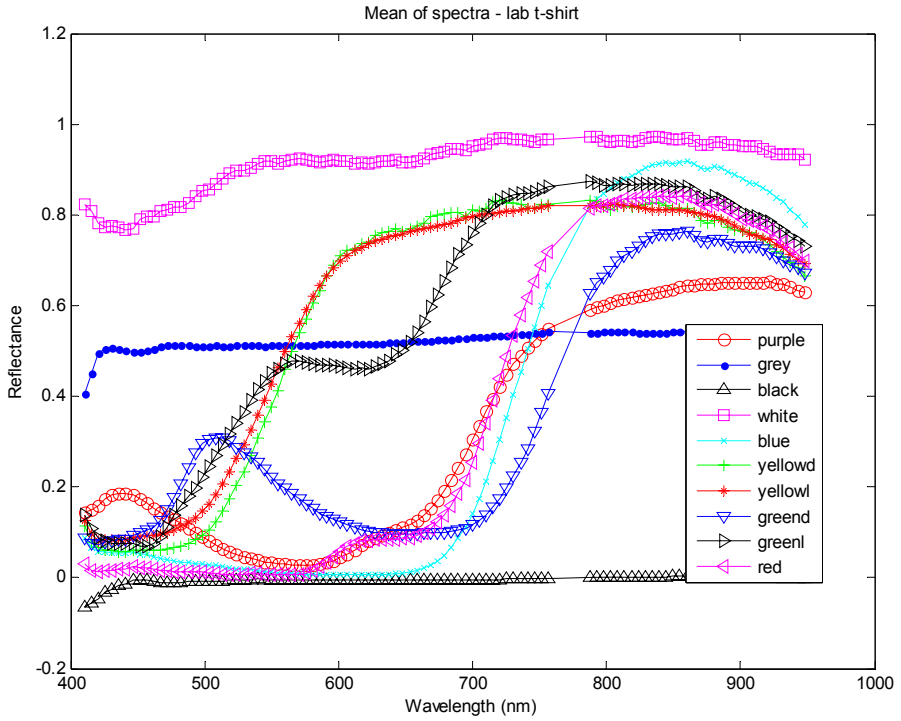


Figure 8-10: Mean spectra of the t-shirt data set

ground truth TD										
Class	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	2	2	2	2	2	2
2	2	0	2	2	2	2	2	2	2	2
3	2	2	0	2	2	2	2	2	2	2
4	2	2	2	0	2	2	2	2	2	2
5	2	2	2	2	0	2	2	2	2	2
6	2	2	2	2	2	0	2	2	2	2
7	2	2	2	2	2	2	0	2	2	2
8	2	2	2	2	2	2	2	0	2	2
9	2	2	2	2	2	2	2	2	0	2
10	2	2	2	2	2	2	2	2	2	0

ground truth JM										
Class	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	2	2	2	2	2	2
2	2	0	2	2	2	2	2	2	2	2
3	2	2	0	2	2	2	2	2	2	2
4	2	2	2	0	2	2	2	2	2	2
5	2	2	2	2	0	2	2	2	2	2
6	2	2	2	2	2	0	2	2	2	2
7	2	2	2	2	2	2	0	2	2	2
8	2	2	2	2	2	2	2	0	2	2
9	2	2	2	2	2	2	2	2	0	2
10	2	2	2	2	2	2	2	2	2	0

Figure 8-11: The pairwise JM and TD scores for the t-shirt data set highlight a large dissimilarity between the classes.

8.4 Data set 4: Shine t-shirt

This data set is similar to the one above but it was taken in an outdoor environment at about noon on the 27th of July, 2009. The background of the scene is the lawn of the campus and the data was taken under direct sun light as depicted in Figure 8-12. The RGB image of the hyperspectral data and the ground-truthed map are presented in Figure 8-13 and Figure 8-14 respectively and the mean spectra of each class is shown in Figure 8-15. Like the previous data set, the TD/JM scores for this scene also exhibit large dissimilarity with a zero score for both of the TTD & TJM.

Figure 8-12: RGB Photograph of the shine t-shirt data with the lawn as the background.

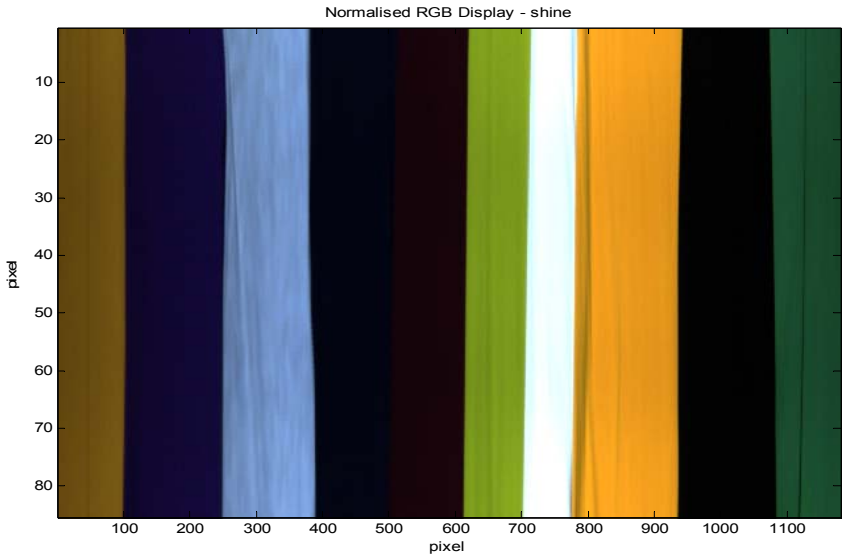


Figure 8-13: RGB image of the shine t-shirt data set

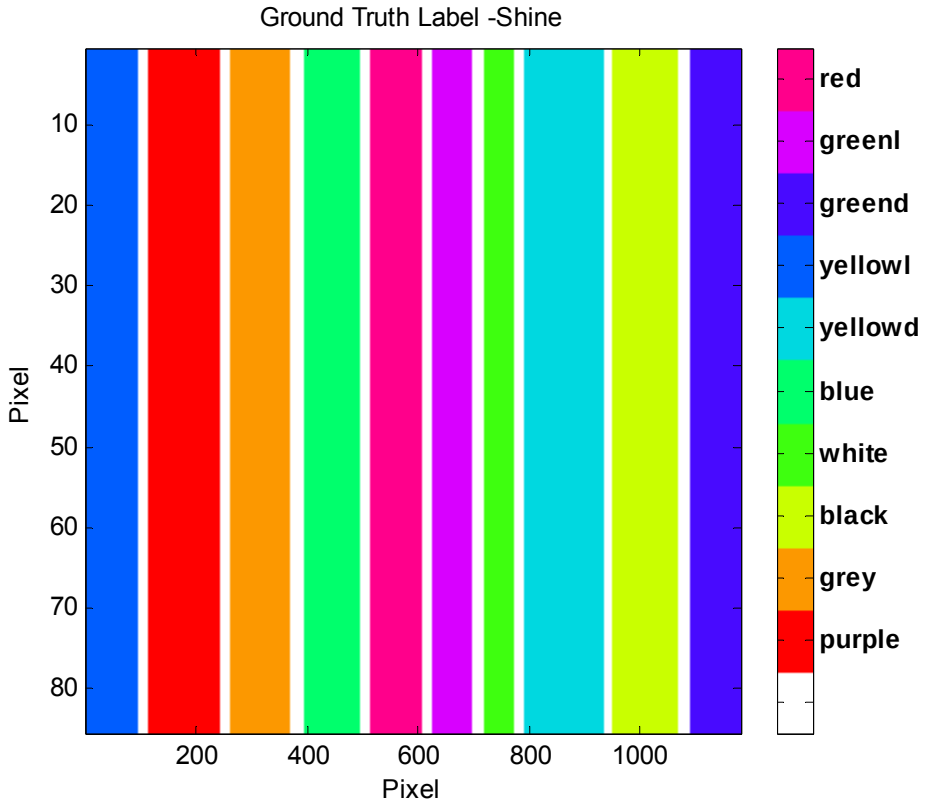


Figure 8-14: The ground-truthed map of the shine t-shirt data set with the boundaries between the t-shirt removed.

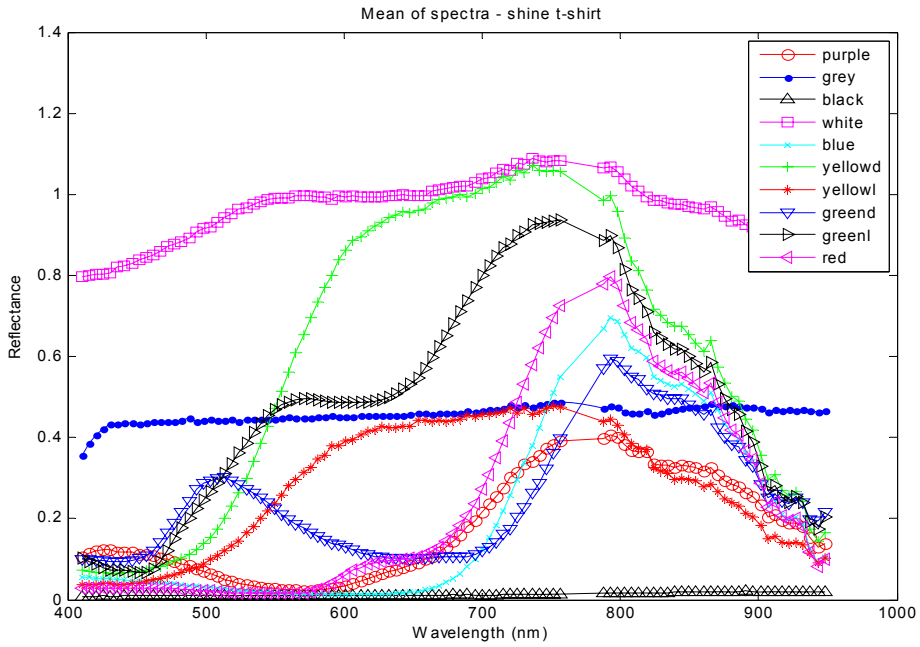


Figure 8-15: Mean spectra of the shine t-shirt data set

8.5 Data set 5: Cloud t-shirt

Again this data set is similar to the above but it was taken on a cloudy day at about noon on 27th of July, 2009. Figure 8-16, Figure 8-17 Figure 8-18 and Figure 8-19 respectively show the photograph, the RGB image, the target map and samples of the class signatures after ELM conversion of the scene. The separation measure for this data set is found to be the same as that presented in the last section.

Figure 8-16: RGB Photo taken in the lawn

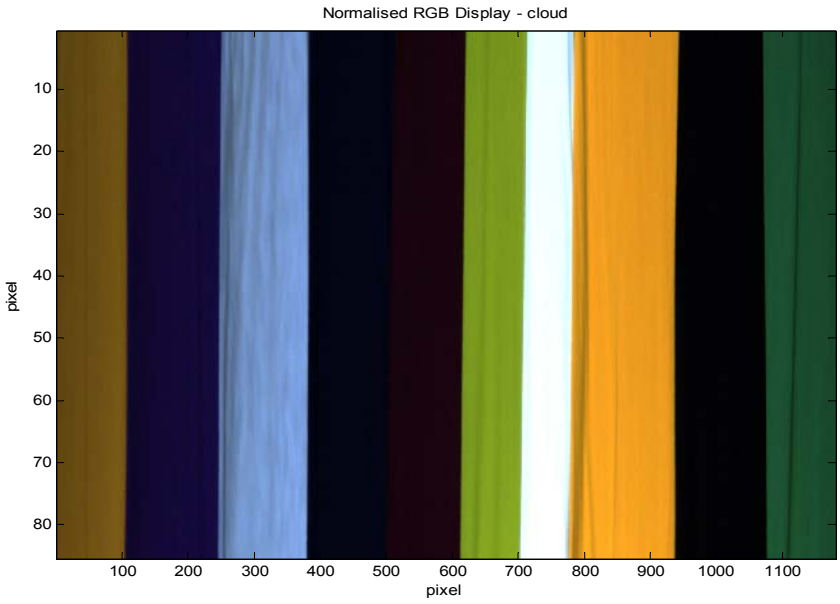


Figure 8-17: RGB model of the data of cloud t-shirt image

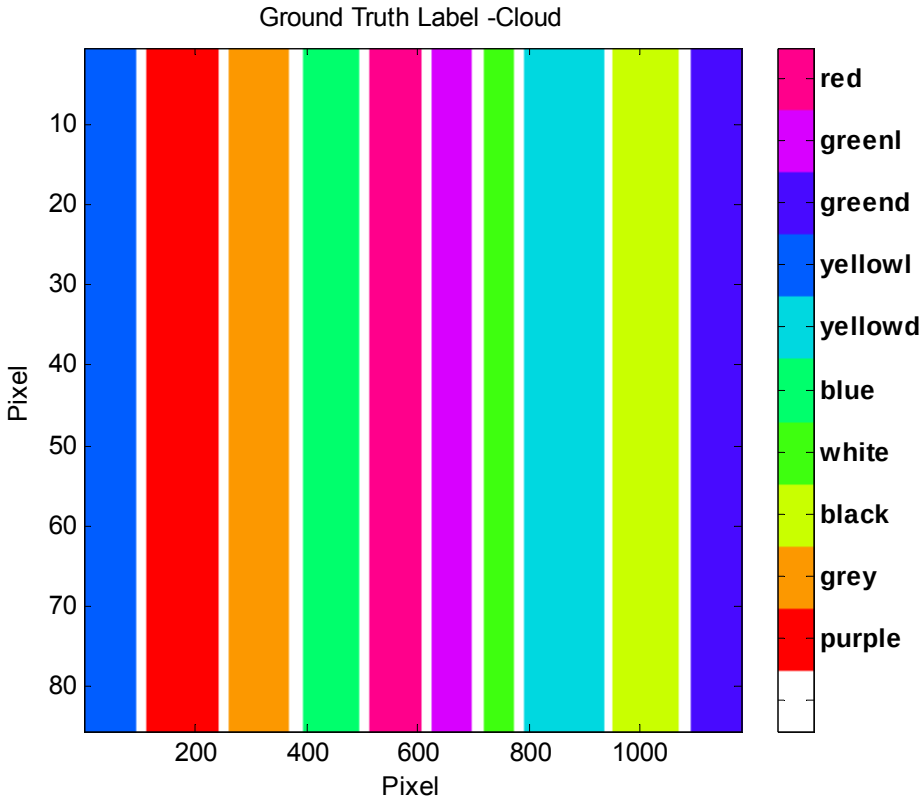


Figure 8-18: The ground-truthed map of the cloud t-shirt data set.

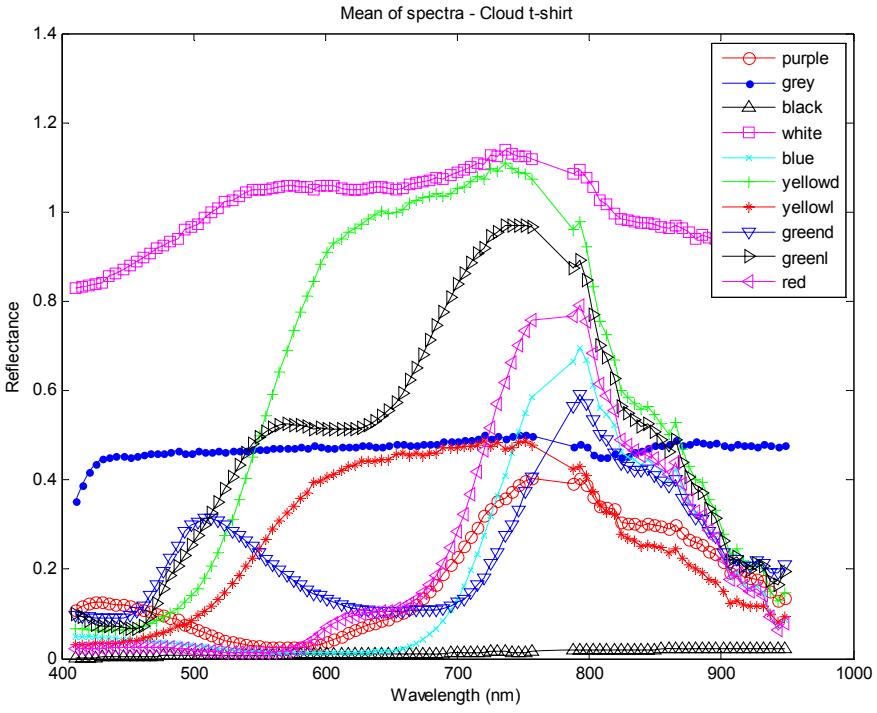


Figure 8-19: Mean spectra of the cloud t-shirt data set

8.6 Data set 6: Car t-shirt

This data set was collected in a car park background at 13.05 on the 8th of May, 2009 under the direct sun light illumination. The background of this scene has been low-reflectance tarmac, and the photograph, the RGB image, the target map and samples of the class signatures are shown in Figure 8-20, Figure 8-21, Figure 8-22 and Figure 8-23 respectively. The TTD & TJM scores for this data set are zero showing large dissimilarities amongst all the classes.

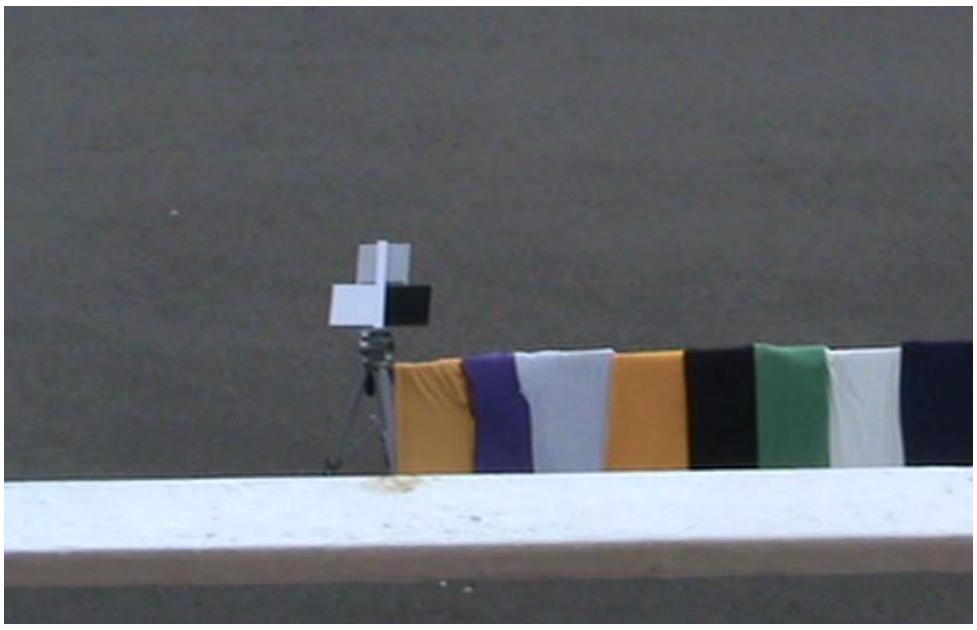


Figure 8-20: RGB Photograph of car park data set

Figure 8-21: RGB model of the car t-shirt data.

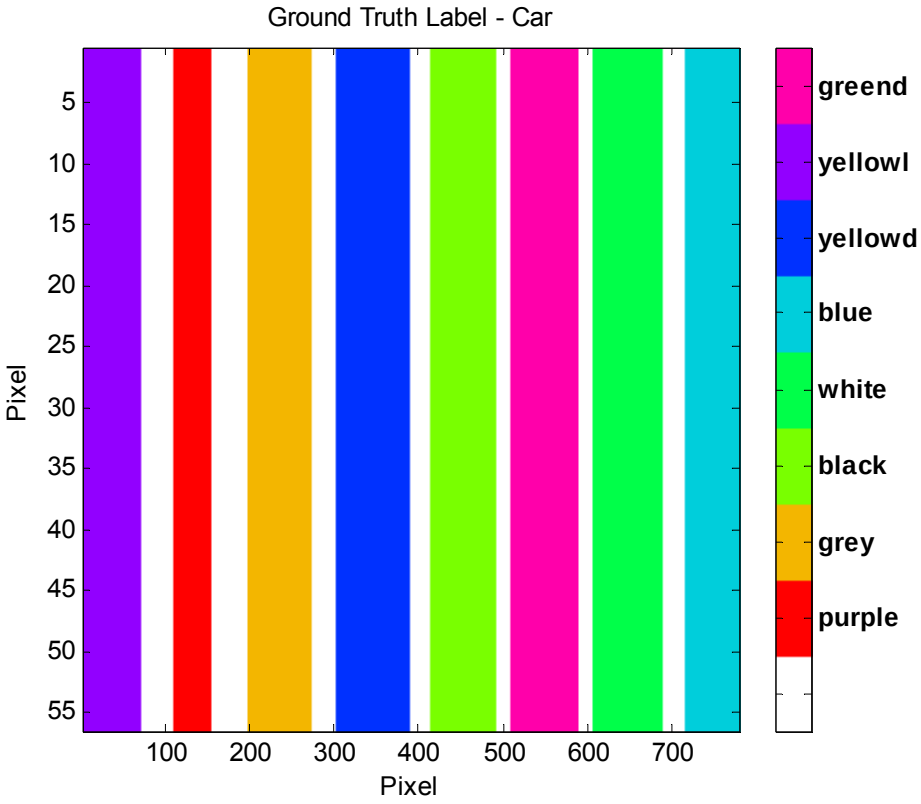


Figure 8-22: The ground-truthed map of the car t-shirt data with boundaries of the t-shirts removed.

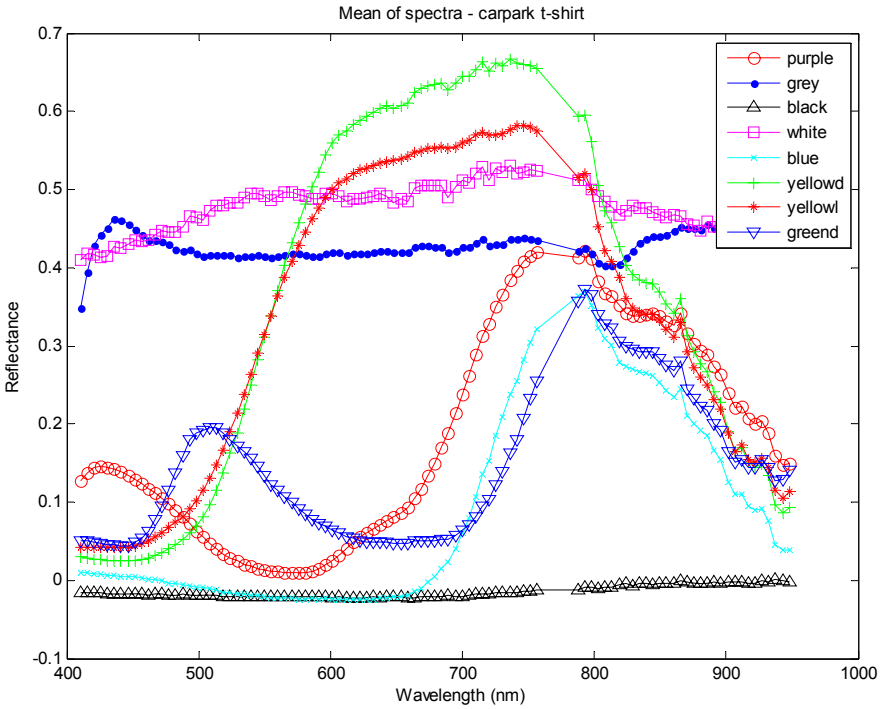
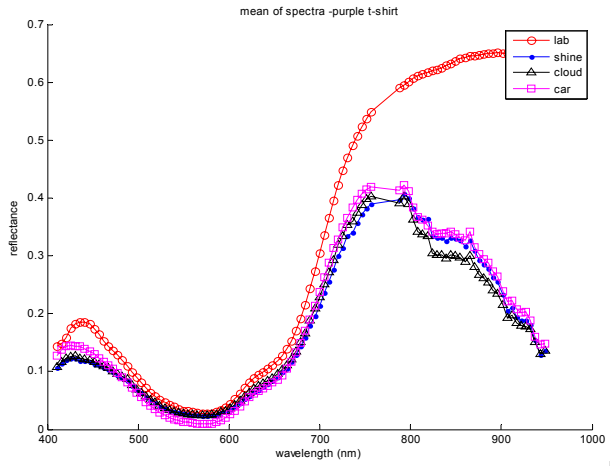


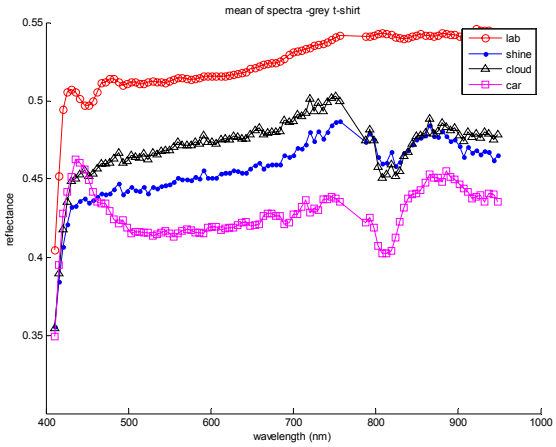
Figure 8-23: Mean spectra of the car t-shirt data set

8.7 Difference in apparent reflectance for data set 3-6

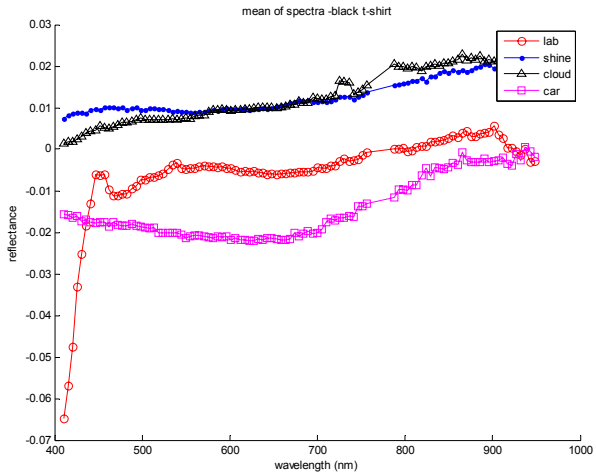
As shown in Figure 8-24, there are some observed differences in the apparent reflectance of the same materials under different conditions for data set 3-6. It can be seen that the angle of incident of the reference panels (black, grey, white Spectralon) is slightly difference from the angle of incident of the targets that are shown in Figure 8-7, Figure 8-12, Figure 8-16 and Figure 8-20. Therefore the ELM result of the same target can be seen quite different under different illumination conditions. This induces large errors in the classification if different data set is used for training and testing. For the rest of the study, only Barrax data (data set 1), Manchester data (data set 2) and lab t-shirt data (data set 3) are used for experiments.



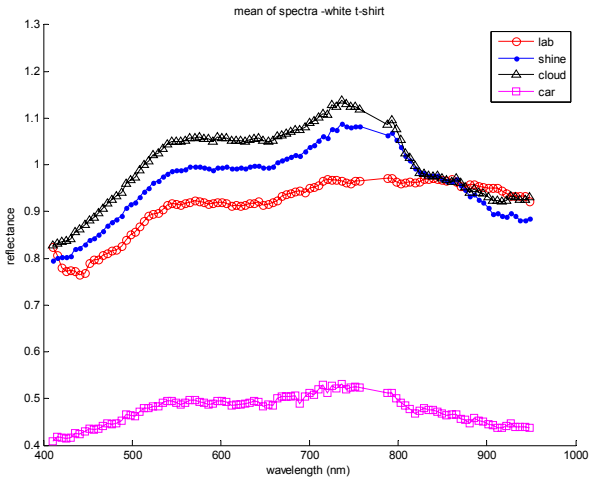
a)



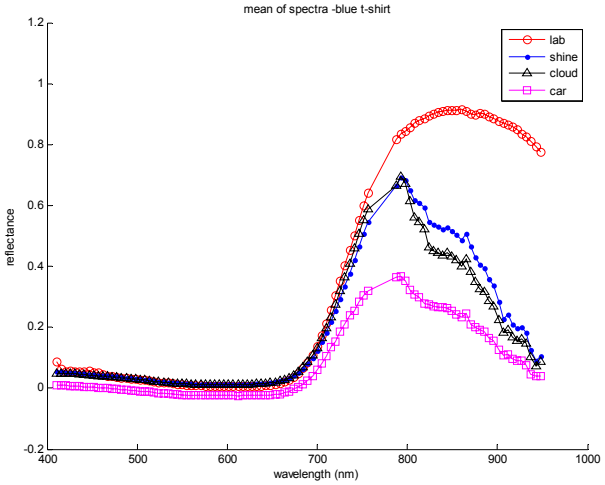
b)



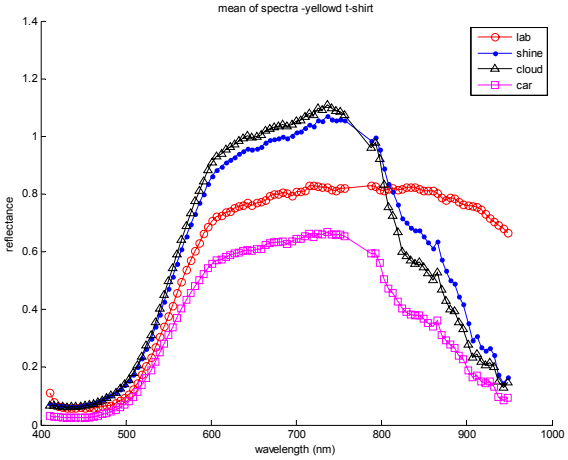
c)



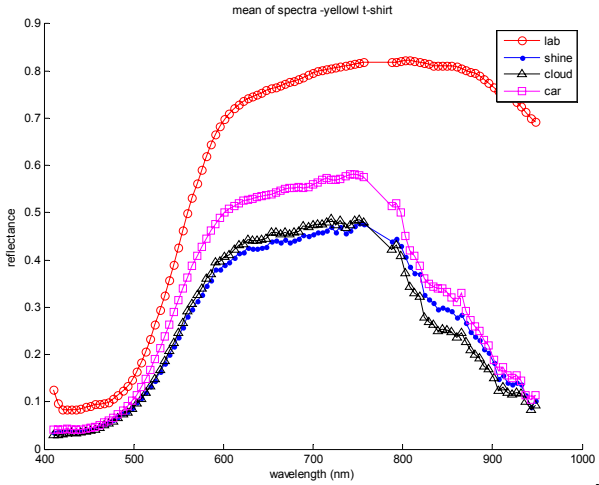
d)


mean of spectra -blue t-shirt
lab
shine
cloud
car
reflectance
wavelength (nm)

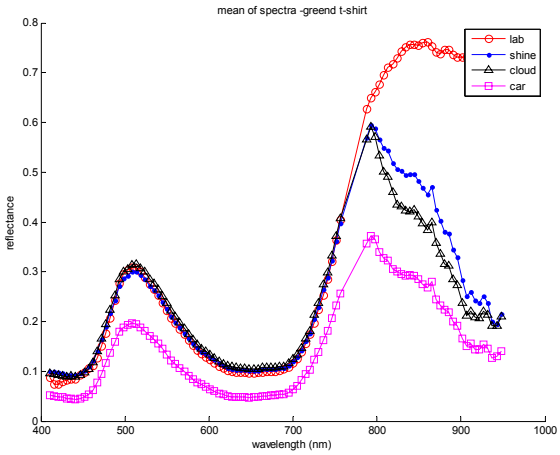
e)


mean of spectra -yellowd t-shirt
lab
shine
cloud
car
reflectance
wavelength (nm)

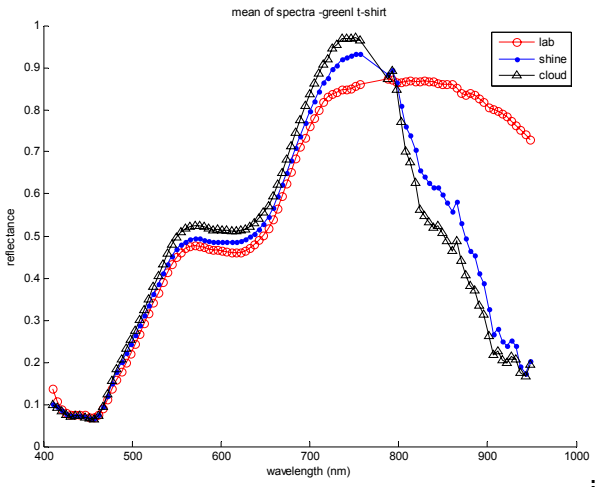
f)


mean of spectra -yellowl t-shirt
lab
shine
cloud
car
reflectance
wavelength (nm)

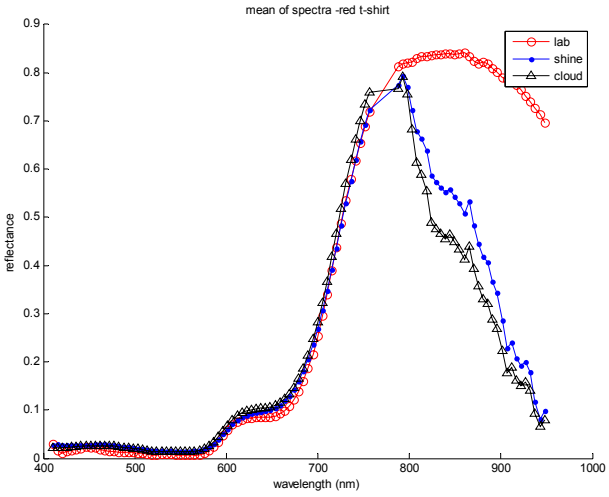
g)


mean of spectra -greend t-shirt
lab
shine
cloud
car
reflectance
wavelength (nm)

h)


mean of spectra -greenl t-shirt
lab
shine
cloud
reflectance
wavelength (nm)

i)


mean of spectra -red t-shirt
lab
shine
cloud
reflectance
wavelength (nm)

j)

Figure 8-24: shows the mean ELM reflectance spectra of the same t-shirts targets collected under various illumination conditions. a) purple t-shirt, b) grey t-shirt, c) black t-shirt, d) white t-shirt, e) blue t-shirt, f) dark yellow t-shirt, g) light yellow t-shirt, h) dark green t-shirt, i) light green t-shirt, j) red t-shirt

9 Hyperspectral image classification experiment

This chapter exploits a range of classification techniques using the TTD and TJM scoring methods. Throughout the chapter, all data points from the ground truth are used for training the supervised classifiers and the whole HSI scene (including the ground truth points) are classified and assessed using the TTD/TJM methods. It is hoped to establish a technique that could evaluate the classification performance without the need of ground truth target map.

9.1 Supervised classifications

9.1.1 Supervised Parametric Classification

Parametric classification method is based on statistical parameters established from the training samples, such as the mean and covariance matrix. In this experiment, three parametric classifiers have been employed and they are the minimum distance classifiers (ED), Mahalanobis distance (FD) classifiers and the Maximum-likelihood (QD) classifiers. The classifications were carried out using ALL of the 16-classes (selected) pixels as the training samples and the whole image as the test data set, and they are then classified by these classifiers which have been implemented in Matlab. The classification results of these three classifiers in false colour maps are presented in Figure 9-1, Figure 9-2 and Figure 9-3 respectively. The classification performances of these classifiers as measured by TTD and TJM have been tabulated in Table 9-1. Recall Equations 6-7 & 6-8 and Table 8-1 that the ideal TTD and TJM for this data set are 0.08316 & 0.3 respectively. It is clear that none of these classifiers perform anywhere close to this ideal condition, with the best score of ~0.6 attained by both the Mahalanobis (FD) and the maximum likelihood (QD) classifiers. In complex scene like the Manchester data set, it is expected that the QD classifier should have performed better than the FD because the QD models the probability density function for each class individually while the FD employs the common covariance for all the classes. However the TTD and TJM scores as shown in Table 9-1 indicates that the FD performs slightly better than that of the QD. This small difference in performance is likely due to the poor estimation of the covariance matrix in the QD for the two classes which are small in sizes (for more details refer to the next chapter).

Type of Classifier	TTD score	TJM score
Maximum-likelihood classifier (QD)	0.6279	2.29555
Mahalanobis distance classifier (FD)	0.6116	2.15585
Euclidean distance classifier (ED)	1.0635	3.00595

Table 9-1: The performance assessment for the classifications using 3 different parametric classifiers on the 16-class Manchester data set. Note that the training sample set consists of 100% of the test data.

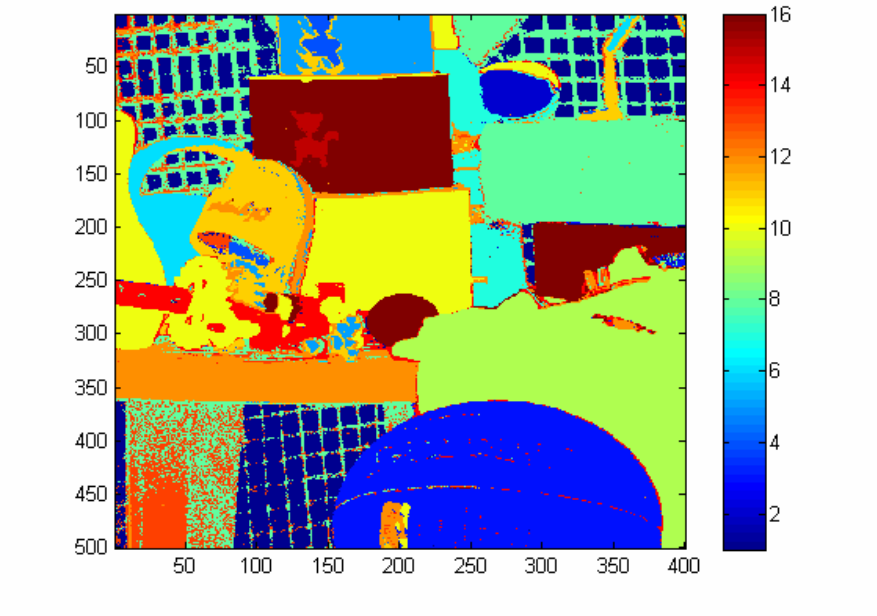


Figure 9-1: Typical classification result presented in false colour map by the Maximum-likelihood (QD) classifier using all ground truthed data as the training samples. The TTD is 0.627 which is far from ideal (base line TTD=0.08316)

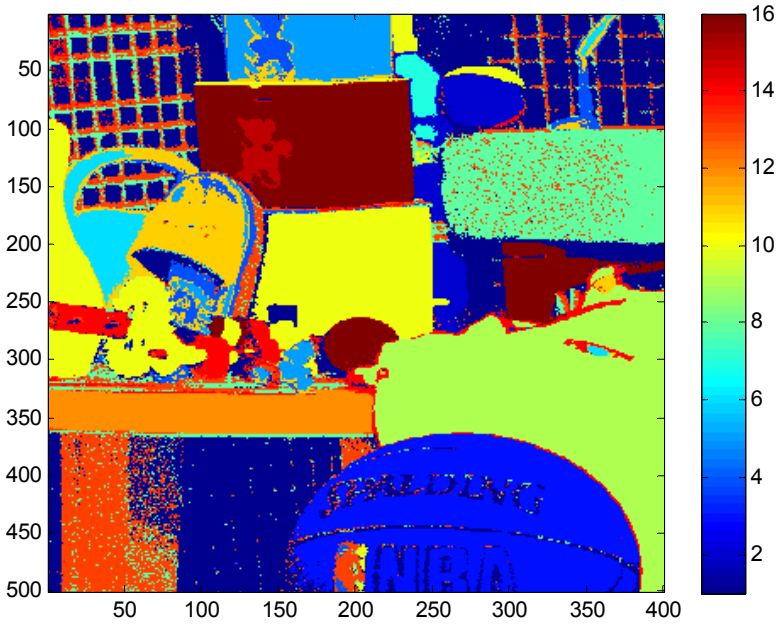


Figure 9-2: Typical classification result presented in false colour map by the Mahalanobis distance (MD) classifier using all ground truthed data as the training samples. The TTD is 0.61 which is far from ideal (base line TTD=0.08316)

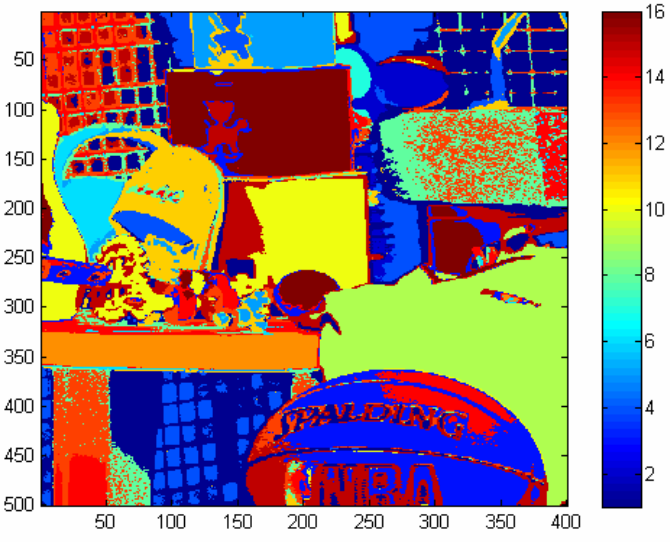


Figure 9-3: Typical classification result presented in false colour map by the Euclidean distance (ED) classifier using all ground truthed data as the training samples. The TTD is 1.06 which is far from ideal (base line TTD=0.08316)

9.1.2 Supervised Non-Parametric Classification

The objective of this experiment is to employ non parametric classifier such as KNN to compare the classifications by the parametric methods. One and eight neighbourhood conditions have been utilised here, and like the previous experiment all of the test data has been employed as the training data for this K-NN classifier. The classifier is implemented in Matlab and the classification performance as measured by TTD and TJM are shown in Table 9-2, showing no significant improvements by increasing the number of nearest-neighbourhoods in the KNN classifier. Typical results by the 1-KNN and 8-KNN are shown in Figure 9-4 which indicates very similar classification performances between them. By comparing this result with that of the classification by parametric methods presented in Table 9-2, it is clear that the KNN performs not as good as the parametric classifiers presented in the last section.

K-NN Classifier	TTD score	TJM score
1-NN	1.3935	3.70555
8-NN	1.3887	3.58075

Table 9-2: The performance assessment for the classifications by the KNN nonparametric classifiers on the 16-class Manchester data set. Note that the training sample set consists of 100% of the test data.

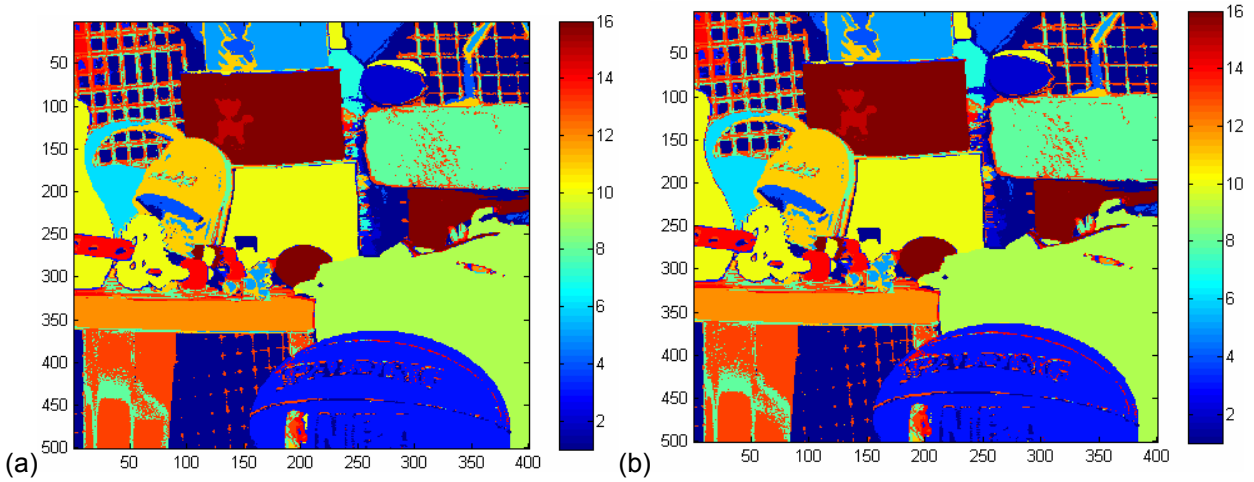


Figure 9-4: Typical classification result presented in false colour maps by (a) 1NN and (b) 8NN classifiers which utilise all ground truthed data as the training samples. TTD for both ~=1.4 which are worse than the parametric classifiers presented in the last section.

9.1.3 Parallelepiped Classifier

Two different decision rules have been adopted in the parallelepiped classifier:

1. The use of the maximum and minimum value of each band in the signature as the upper and lower limits of the parallelepipeds, and
2. The upper and lower bounds was determined by the mean of each bands, plus and minus 2*the standard deviations of each bands.

The classifications by using these two methods have been respectively presented in & Figure 9-5 & Figure 9-6, and for the first approach there are 27.24% and 18.76% of overlapped and unclassified pixel respectively, resulting in ~46% of the data remain either unclassified or undetermined. The second approach has shown an even worse result with a total of 57.4% undetermined region. This classifier is good in terms of speed but poor in terms of performance. Consequently, it is normally used in conjunction with other classifier such as maximum likelihood classifier (Richards and Jia, 2006). The TTD and TJM scoring measures have been avoided in this case due to the missing of large number of pixels in the classification.

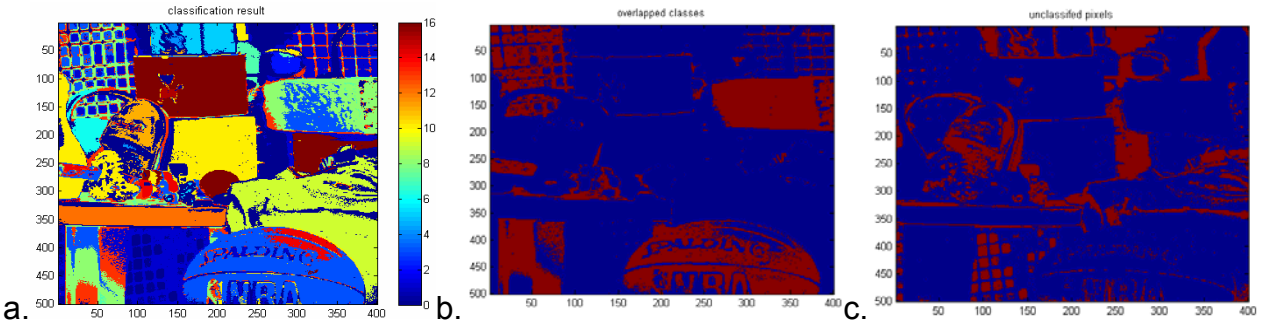


Figure 9-5: Parallelepiped classification result using the Max, Min of each band in the signature, a) the overall result, b) the amount of overlapped pixel (27.24%), c) the amount of unclassified pixel (18.76%)

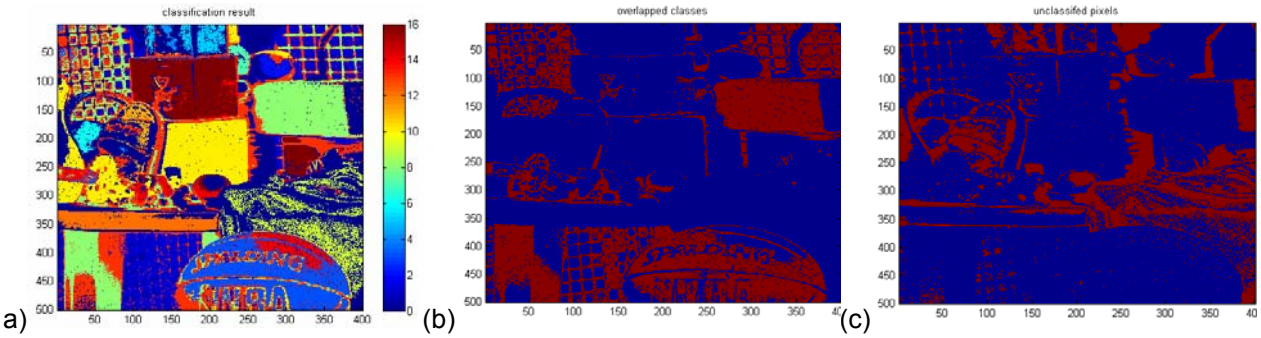


Figure 9-6: Parallelepiped classification result using the mean of each band, plus and minus 2*standard deviations a) the overall result, b) the amount of overlapped pixel (33.18%), c) the amount of unclassified pixel (24.22%)

9.2 Unsupervised classification

9.2.1 K-means clustering

K-means clustering has been a popular unsupervised classification technique due to its algorithmic simplicity. In this experiment the Manchester data set is examined by a K-means classifier that has been implemented in Matlab. The experiment is repeated for 50 runs and each begins with a random initialisation of 16 clusters. It is found that the classifications are quite sensitive to the initial conditions and typical results for a consecutive of two runs are presented in Figure 9-7. The overall averaged TTD and TJM for these 50 runs of classifications are shown in Table 9-3, which gives a TTD of ~0.91 being quite close to that of the best supervised parametric classifier given by the FD (0.61) for this data set (refer to section 9.1.1). Note that K-means has been an unsupervised classifier without any need of training, and its performance is seen better than some supervised techniques such as the ED (TTD~1) and the KNN (TTD~1.4) classifiers.

	K-means run	TTD score	TJM score
	1	0.6951	1.86595
	2	1.0671	2.08195
Average	1-50	0.9084	2.0735

Table 9-3: The performance assessment for the classifications by the K-means unsupervised classifiers on the 16-class Manchester data set. Note that the k-means classification according to the TTD is close to that of the best supervised parametric classifier.

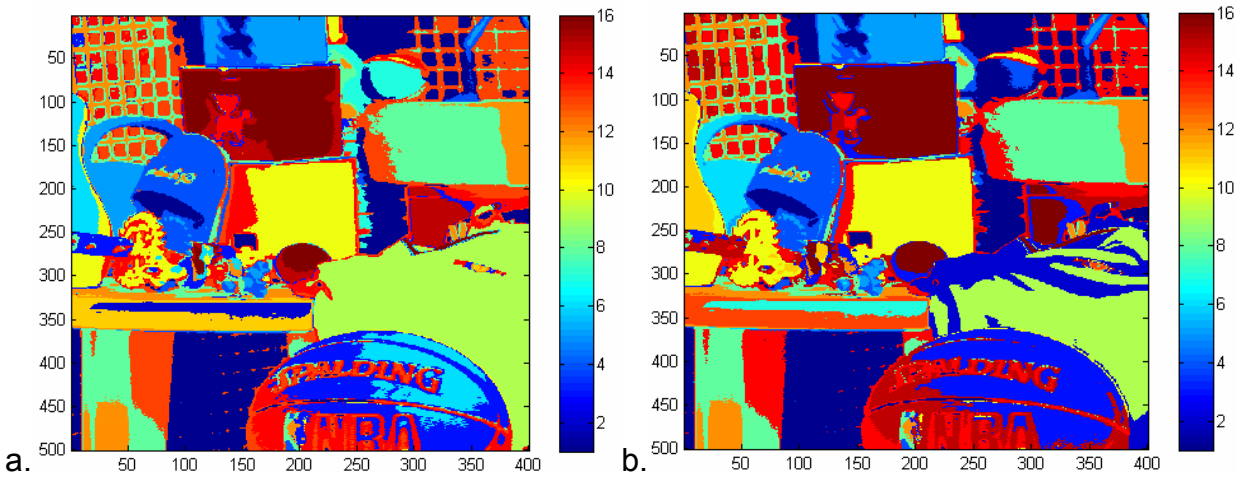


Figure 9-7: Typical consecutive runs of K-means classification with results presented in false colour maps (a) 1st run (b) 2nd run.

9.2.2 Fuzzy C-means

Fuzzy C-means can be regarded as a version of soft clustering K-means technique which uses the same cost function for minimising the mean squared errors of the cluster centriods (see section 4.5.4 above). In addition, the fuzzy C-means utilises a radial weighting function which is characterised by the exponential distance of the test pixel with respected to the cluster centriod. The settings of this exponential p (see equation 4-25) are data dependent. In this case two different values of p=(2,5) have been employed and the TTD and TJM scores over 50 runs are presented in Table 9-4, which highlights a really bad classification particularly when p=5 where the radial function becomes so peaky that some classified clusters have got only a few pixels inducing an ill-defined covariance and thus a very small TD/JM scores. This effect is exemplified in the classification results as depicted in Figure 9-8 for p equals to 2 and 5. Although fuzzy C-means belongs to a kind of unsupervised classification, the parameterisation of the 'correct' radial weighting function to suit for the data sets is found non-trivial.

(fuzzy-exponent) P	TTD score	TJM score
2	1.6416	3.2784
5	12.6624	13.9908

Table 9-4: shows the goodness of the fuzzy c-means classifications via the separability measures. Note that large errors are resulted particularly when the radial function is chosen to be very peaky (p=5).

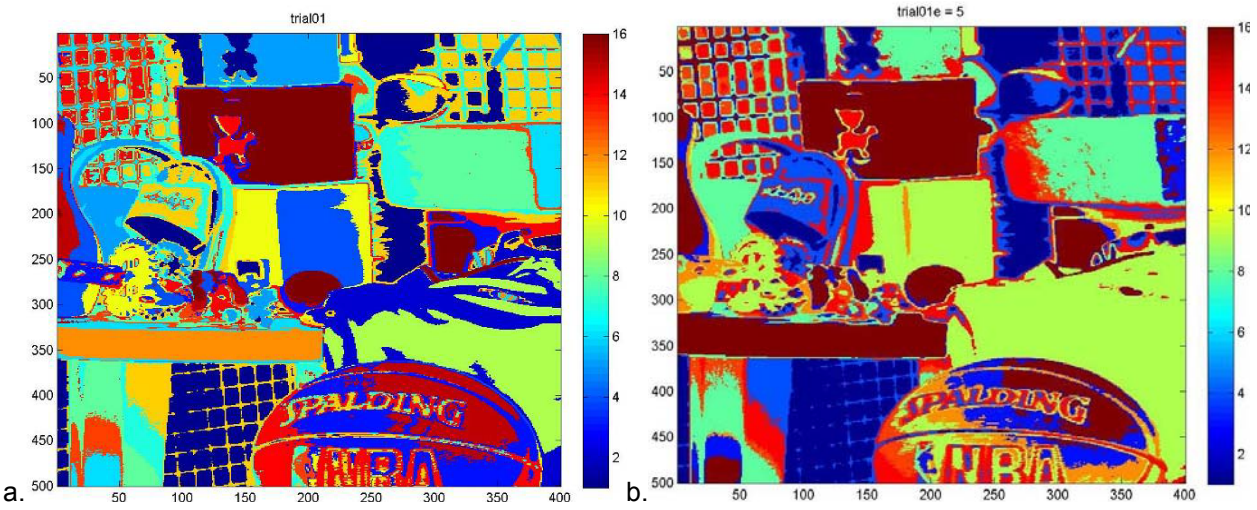


Figure 9-8: Typical classification result in false colour map by fuzzy c-means using a radial exponent (a)p=2 (b) p=5. Note that there are a lot of mis-classified pixels in (b) purely because of the wrongly choose of the radial weighting function.

9.2.3 Self-Organising Maps

Three different versions of SOM algorithm have been testified in this study: a) Matlab's neural network toolbox, b) Helsinki University's SOM toolbox and c) own algorithm developed in this work. One drawback for the MATLAB's SOM toolbox has been the limitation of only one learning rule (linear) and neighbourhood function is available. The Helsinki's SOM toolbox (version 2.0) has been a powerful and versatile algorithm but unfortunately there is a compatibility issue with the MATLAB version 7 and higher. Some of the SOM functions, e.g. the learning rules and the neighbourhood, have been developed during the course of this study and it is planned to piece this together with other SOM codes available from the public domain.

The basic idea of SOM is a self-evolving network which 'learns' when data is passed through the network in a sequential manner. Like other neural network (NN) based clustering algorithms, there are many parameters such as the topology of the network, the learning rules, the updating mechanism and the data input strategy which all can critically affect the performance of the classification.

Example of SOM clustering is illustrated in Figure 9-9 & Figure 9-10 where the 3 bands (band 3, 6, 22) of the Manchester data set have been passed through a 16-neuron SOM network using rectangular topology. The figures are plotted in the 3 dimensional

weighting of the net and the green dots represent the pixel vectors in the network's (weight) space. The red dots represent the centres of the 16 neurons. Figure 9-10 shows another view of the same plot which exhibits a planar structure, indicating that at least two of these dimensions are in a linear relationship.

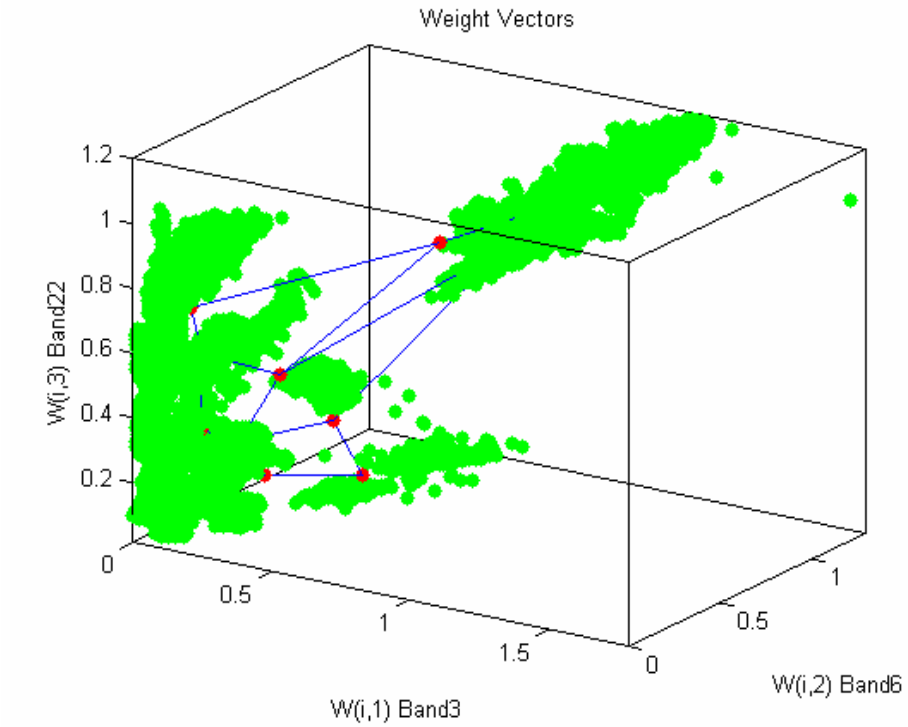


Figure 9-9: shows the clustering of 3-band Manchester data in a 16-neuron SOM network using rectangular topology. The plot is shown in the 3 weighting space of the net, with green dot represents the pixel vectors and red dot the centre of the 16 neurons.

Like many other classifiers the parameterisation of the SOM network requires a systematic investigation. In here the experiment involves a stepwise change of topology of linear, rectangular and hexagonal; various number of pixel vectors, epochs and experimental runs.

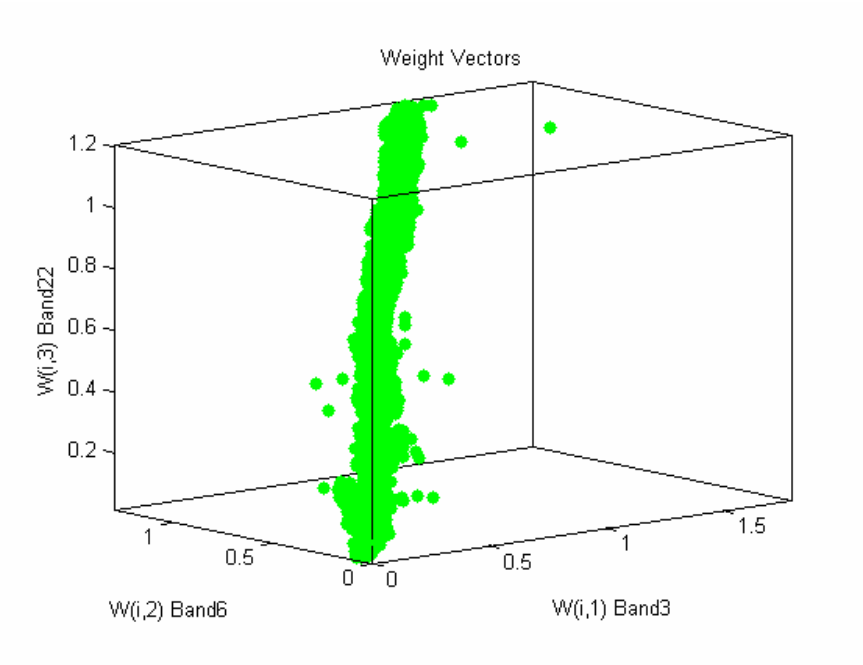


Figure 9-10: shows the same plot as the previous figure but in a different view, highlighting the planar structure of the pixel vector in the net space.

The classification results have shown that the performance is weakly dependent on the topology of the network (see Figure 9-11), and by using a power learning function together with a Gaussian neighbourhood and linear decreasing neighbourhood radius, the best TTD and TJM scores for the Manchester data set that have achieved are 0.95 and 2.49 respectively (see Table 9-5). This performance is very close to that of the K-Means and is comparable to the best supervised classifier (FD) for this data set.

SOM sample size	Topology	Fine tune (Y/N)	TTD score	TJM score
400*500(whole image)	Line	N	1.07856	2.181
Ran50	Rectangular	N	30.76824	31.85004
10000	Rectangular	N	0.7248	1.62168
whole image	Rectangular	N	0.94836	2.49168
whole image	Rectangular	Y	1.03452	2.27364
whole image	Hexagonal	N	0.94836	2.49168
whole image	Hexagonal	Y	1.03452	2.27364

Table 9-5: The performance of the classifications for the Manchester data set using the Helsinki SOM code.

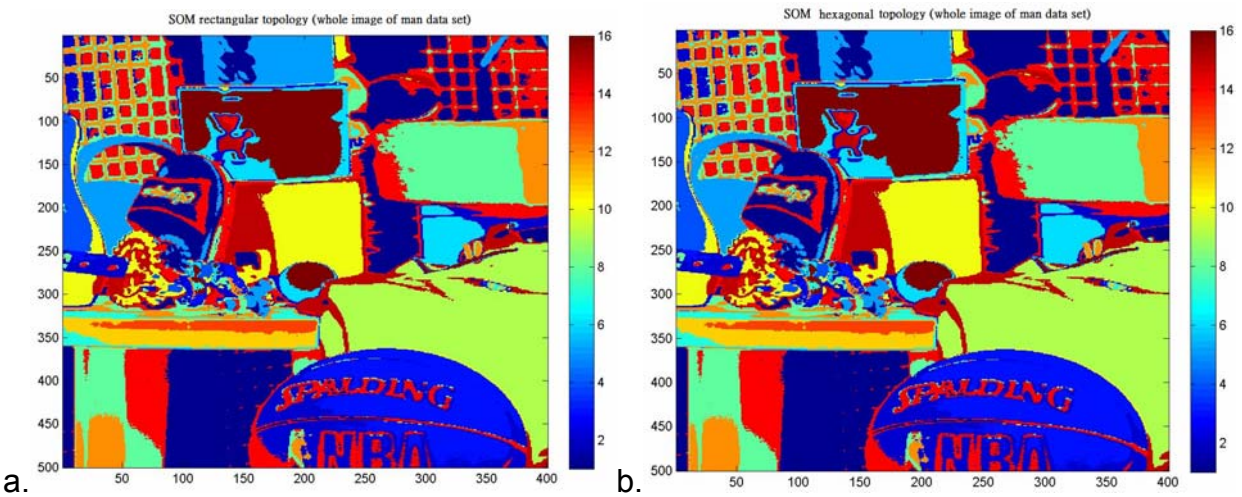


Figure 9-11: showing the classification results in false colour maps by the SOM using (a) rectangular and (b) hexagonal topology network. Both results exhibit a TTD of ~0.95, close to that of the K-Means and FD classifiers.

9.3 Effect of spectral range to classification accuracy

9.3.1 Spectral range experiment

It is mentioned in the previous section that the number of the spectral bands, the spectral range as well as the band resolutions are some of the factors that affect the classification performance. As a first step we have studied the GT accuracy (see Equation 10-1) of the Barrax data set (see section 8.1) as functions of these parameters. The classification accuracies have been evaluated with respected to the target map and the unsupervised K-means classifier has been employed for this study. As highlighted in the previous section 9.2.1 that the classifications by K-Means are prone to the initial conditions and hence experiment is repeated for 10 times to obtain an average. Figure 9-12 plots the accuracies versus the number of spectral bands (7,14,42,126,and 128) that have been employed for the classification of the Barrax data set. 126 bands were attained by discarding the two band extreme of both end of the spectral; 42 bands were achieved by aggregating three neighbouring bands into one band from the 126 bands dataset; 14 bands were achieved by aggregating three neighbouring bands into one band from the 42 bands dataset; and 7 bands were achieved by aggregating two neighbouring bands into one band from the 14 bands dataset. The dash lines represent the results of each run and the red solid line indicates the mean over all the runs. It is

seen from the figure that the performance reaches to a plateau at the range of 40th bands equivalent to a spectral range of 0.4-2.48um in this data set. This result may thus suggest that the classification performance will be further improved by using a proper band selection scheme for reducing the dimensionality of the data set conforming to the Hughes phenomenon.

The experiment is subsequently repeated by sub-sampling the spectral bands into every 20nm intervals and the classification result for this case is shown in Figure 9-12 which is remarkably similar to that presented in Figure 9-13.

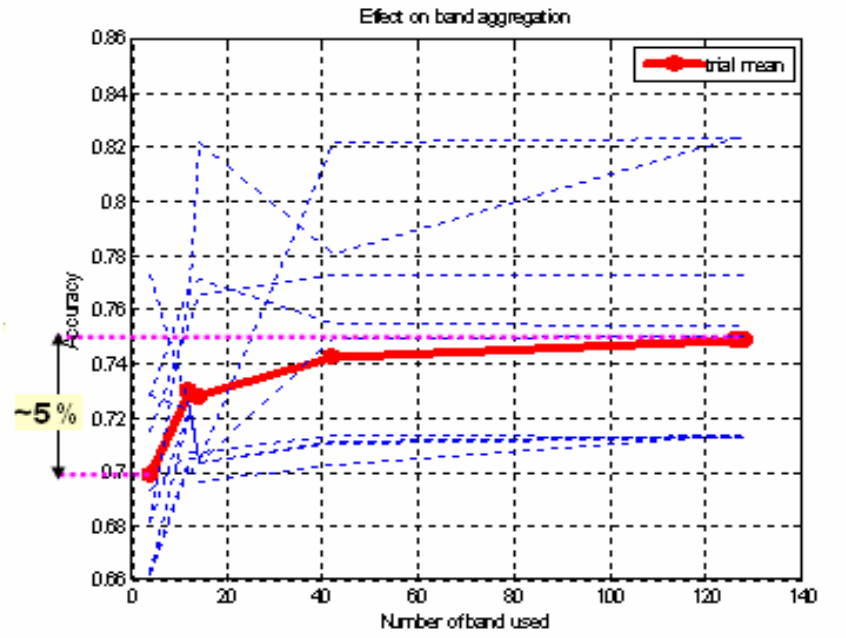


Figure 9-12: The accuracy of the K-Means classifier for the classification of the Barrax data set as a function of five input spectral ranges of 7,14,42,126 and128 bands.

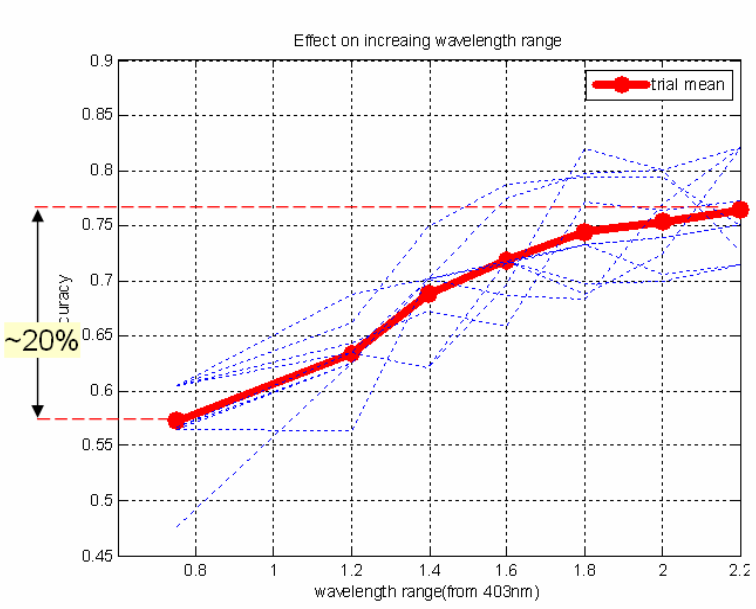


Figure 9-13: The accuracy of the K-Means for the classification of Barrax data after subsampling data in a step of 20nm intervals. Note that the dimensionalities as well as the spectral ranges are both increasing as the trace goes from left to the right.

10 Supervised classifications & performance assessments

10.1 SVM:- T-shirt and Manchester data sets: 40% training samples

SVM has been shown to be one of the most versatile supervised classifier ever invented in the machine learning research. The objective of this experiment is to test the performance of this classifier and to study how the model parameters affect its classification accuracy. A number of kernel functions with various model parameters implemented in a one-against-one as well as the one-against-all modes have been implemented in MATLAB, and their classification performances have been assessed through the class dissimilarity and site specific measures for the classification of the Manchester and the lab t-shirt data sets. The site specific measure directly compares the class labels of every pixel with respected to the ground truth:

$$GTaccuracy_C = \frac{\sum_{i=1}^{N_C} CL_i}{N_c} * 100\% \qquad [10\text{-}1]$$

where

C is the labelled classes

N_C is the total number of pixel in Class C

CL_i=1 for the i^{th} pixel in class C being correctly classified and is equal to 0 otherwise.

$$AverageGTaccuracy = \frac{\sum_{C} GTaccuracy_c}{\text{Total Number of Class}} \qquad [10\text{-}2]$$

Figure 10-1 and Figure 10-2 showing the classification results for the t-shirt and the Manchester data respectively using both SVM modes and kernel functions of linear, polynomial and radial bias Functions. All results have shown that the one-against-one (OAO) mode achieves a much better classification than the one-against-all (OAA) mode regardless of the kernel function employed. Thus in the rest of this section the SVM classification result will be presented for the one-against-one (OAO) mode only.

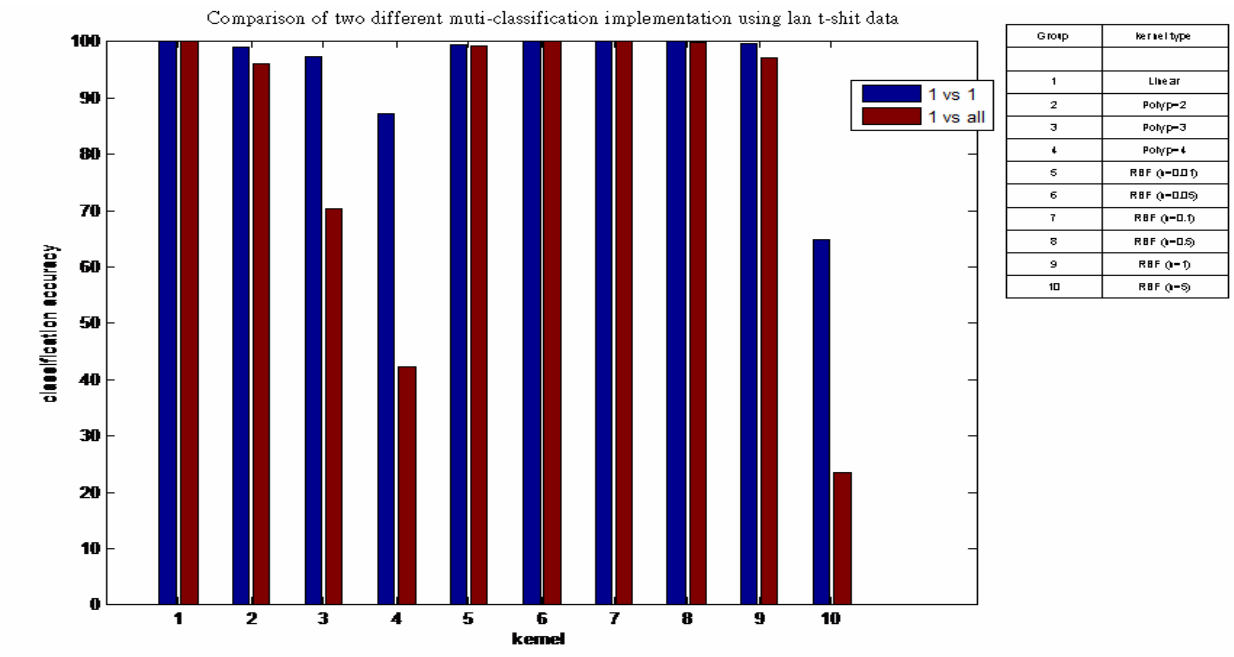


Figure 10-1: Classification results of SVM using various kernels in the OAO and OAA modes for the T-shirt data set. The accuracy is measured with respected to the ground truth (equ 10-1).

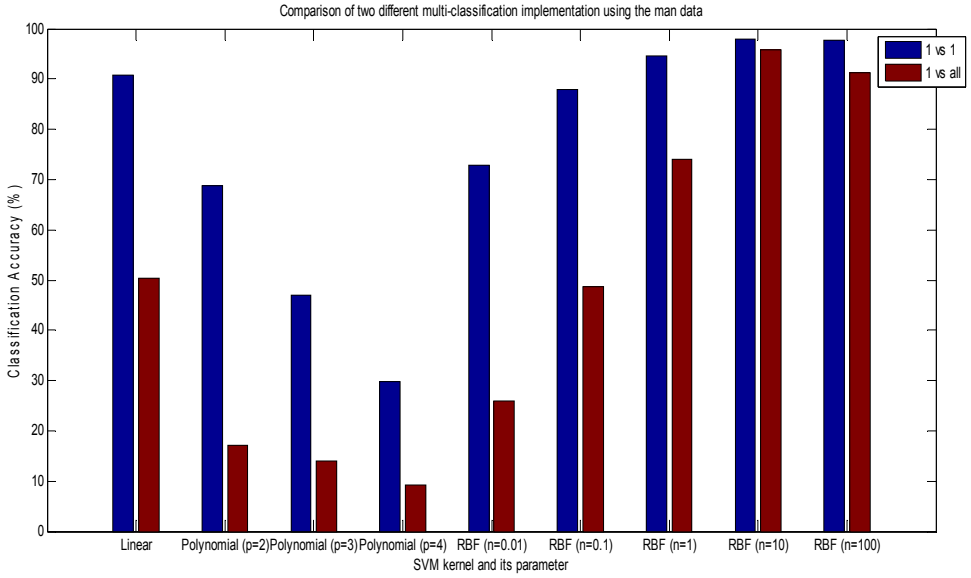


Figure 10-2: Classification results of SVM using various kernels in the OAO and OAA modes for the Manchester data set

10.2 SVM:- T-shirt and Manchester data sets: 100% training samples

It is seen from the previous experiment that the SVM that employs linear, polynomial (p=4) and the RBF (gamma=0.1) kernels exhibit classification accuracies of 100%, 87%

100% and 90%, 30% and 86% for the T-shirt and Manchester data sets respectively under the OAO mode. Note that this classification had been carried out using 40% of the training data set, and it is of interest to see how the accuracy is affected by extending the training data set to 100%.

10.2.1 Lab T-shirt data set

In this experiment the classification has been carried out the same way as that in section 10.1 but a 100% of the data pixels have been used for the training in this case. Table 10-1 shows a substantial performance improvements by the SVM polynomial (p=4) classifier going from an accuracy of 87% when 40% of training data is used, to almost 100% when the full data set is employed for the training. Figure 10-3 shows typical classification results to highlight almost 100% accuracies attained by all three SVM classifiers when the training is increased to 100% of the data set.

Type of SVM Classifier (cost=1)	Average GT accuracy	TTD score	TJM score
Linear	100%	0	0
Polynomial p=4	99.94%	1.18E-15	1.18E-14
RBF gamma=0.1	100%	0	0

Table 10-1: The performance assessment for the classifications using 3 different kernels for the SVM classifiers on the 10-class t-shirt data set. Note that TTD and TJM are calculated from the ground-truth region of interest only (see chapter 8), and it is not evaluated from the whole data set.

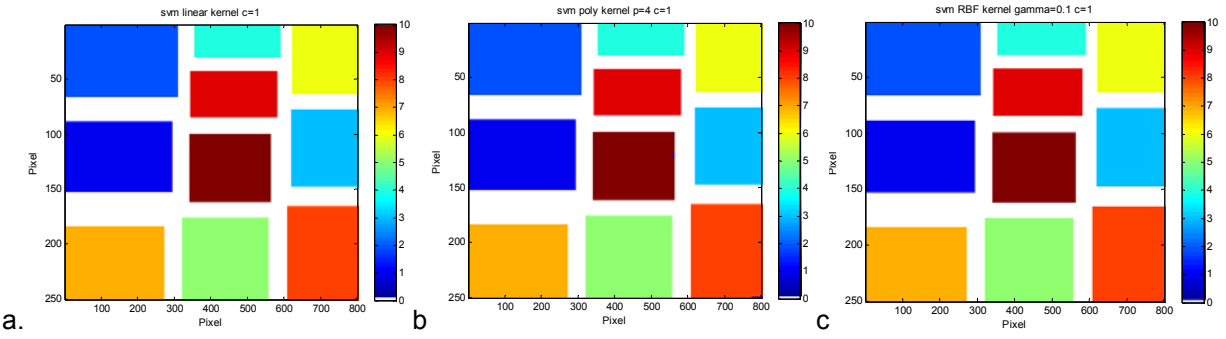


Figure 10-3: The classification results for the lab t-shirt data set using SVM with kernels of (a) linear, b) Polynomial (p=4) and c) RBF (gamma=0.1). The maps show the classifications of the ROI test areas in false colours and all results have shown almost 100% accuracy when ALL of the data have been used for the training (c.f. Figure 10-1 & Figure 10-2).

10.2.2 Manchester data set

The purpose of this section is to make a direct comparison between the classification performances by the SVM with respect to other supervised and unsupervised classifiers. Thus this experiment is run under exactly the same conditions as that of chapter 9, and three different kernels of linear, polynomial and RBF for the SVM have been employed here in this work. It is seen that the SVM classifier have performed excellently, achieving ~100% accuracy with almost ideal 0.088 TTD when the RBF kernel is employed (Table 10-2). Recall that the separativity measure TTD of the training data set is about 0.083 (Table 8-1). To make sure if the classification really achieves this high level of accuracy, the classification results in false colour maps obtained by the linear, polynomial and the RBF kernels are presented in Figure 10-4, Figure 10-5 and Figure 10-6 respectively. In Figure 10-6 it is verified that the number of mis-classified pixels in this classification result amounts to 126 pixels, which is exactly 0.5% of the overall 16-class data sets (25244 pixels). It is also noted from this experiment that the RBF has performed excellently over other kernels, and it is intuitive to study how the parameterisation of the RBF kernel can be achieved from the image data.

Type of SVM Classifier (cost=1)	Average GT accuracy	TTD score	TJM score
Linear	97.57%	0.1839	0.56995
Polynomial p=4	68.32%	183.9951	184.038
RBF gamma=10	99.50%	0.08886	0.31675

Table 10-2: The performance of 3 different SVM classifiers for the 16-class Manchester data set. Note that TTD and TJM are calculated from the ground-truth region of interest only, and it is not evaluated from the whole data set.

10.3 SVM:-The RBF and the cost parameter

To handle non-linearly separable classes, the RBF kernel are normally employed. The RBF is controlled by the parameters gamma (γ), which inversely scale to the variance of the cluster. To allow a soft margin for accommodating small amount of misclassifications, a cost parameter denoted by C can be implemented within the SVM to handle the exchange between the errors of the allowed training and stiffness of the separation plane. A larger C represents a greater capacity for the accommodation of misclassification errors. This cost parameter can be found by using either a pattern or a

grid search method. Grid search processes every value of the parameter in its total range with the help of the geometric shapes in the feature hyper-space. Pattern search method is commonly referred as the line search or compass search. It normally begins with the centre of its range and then processes every value parameters in all directions. The nucleus of the search, in this case the one with the highest accuracy, shifts towards a new point if the model appears to be better and the whole process repeats itself again. And in case of no improvement then the search decreases in step size, and the process will be terminated when the step size is reduced enough to a preset value.

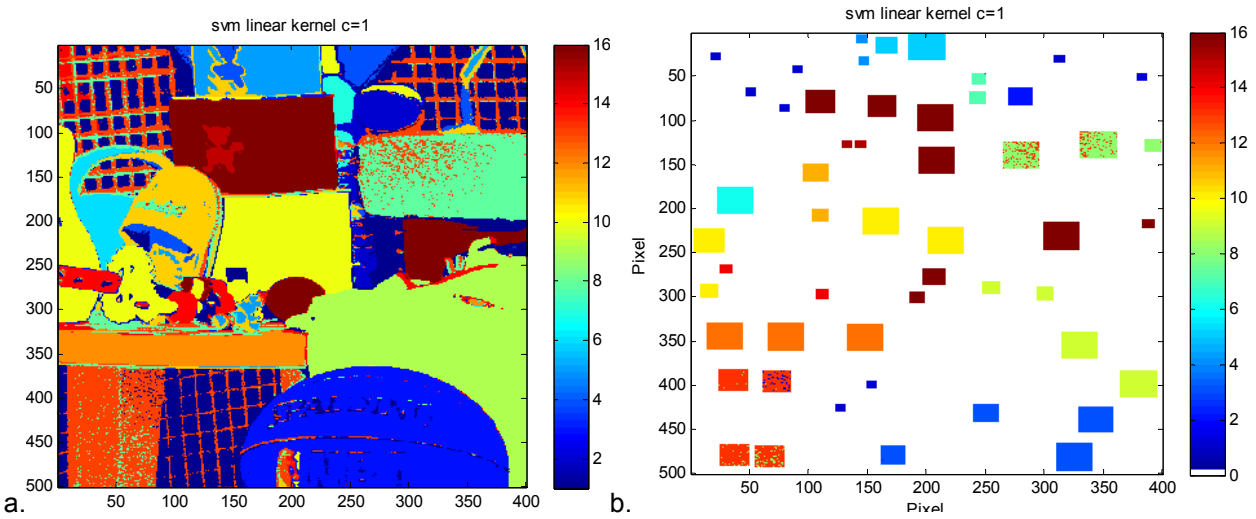


Figure 10-4: Shows the classification results in false colour map for the 16-class Manchester data by using the SVM linear kernel classifier, (a) the complete image (b) the selected ROI data set (25244 pixels). The accuracy of this classification is 97.6%.

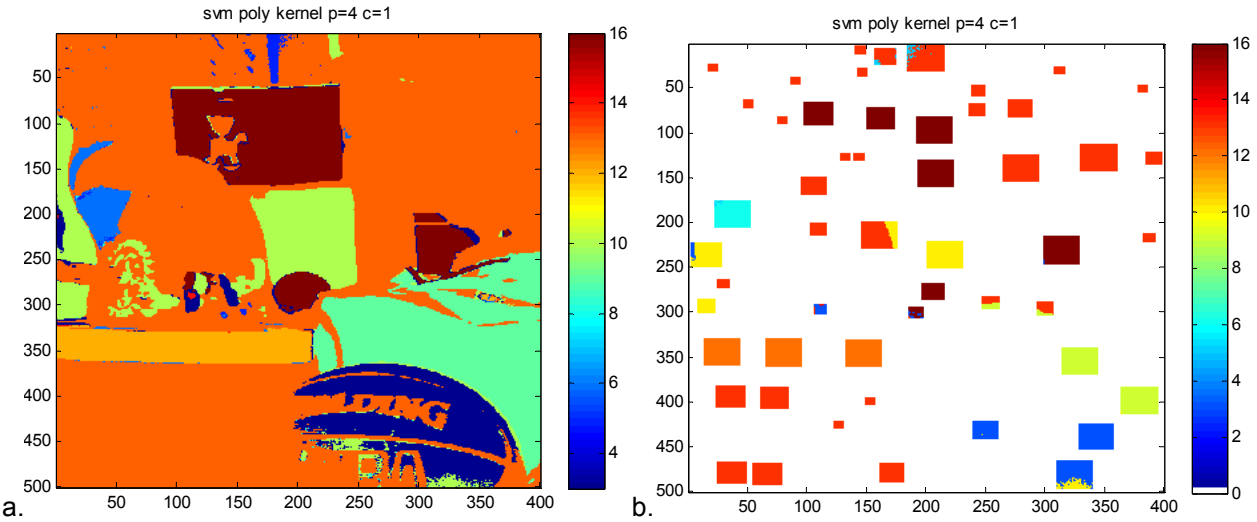


Figure 10-5: Shows the classification results (68% accuracy) in false colour map for the 16-class Manchester data by using the SVM polynomial kernel classifier, (a) the complete image (b) the selected ROI data set (25244 pixels).

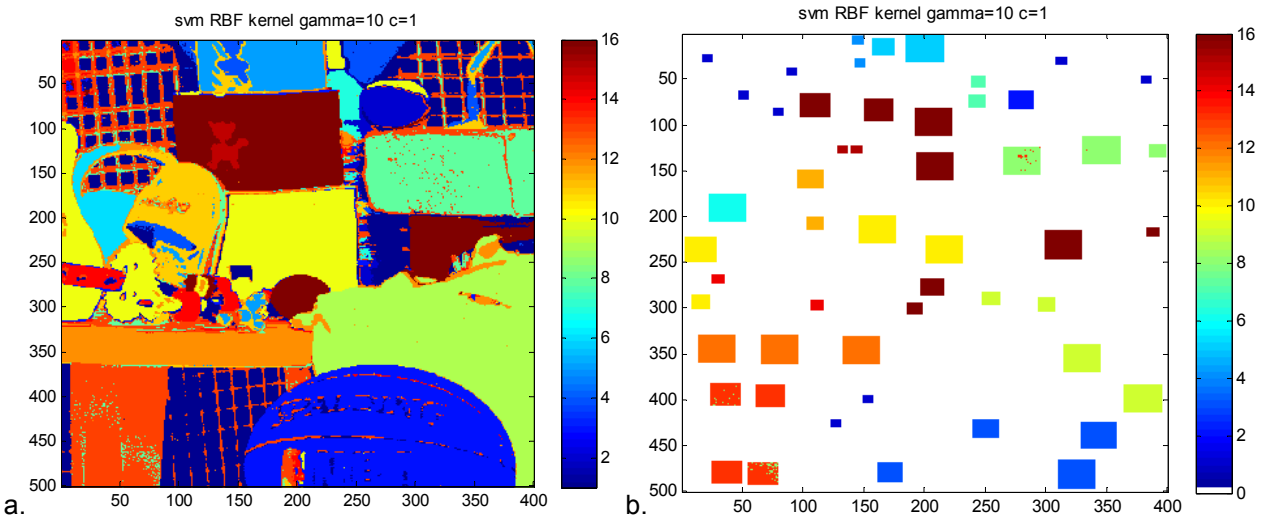


Figure 10-6: Shows the classification results (99.5% accuracy) in false colour map for the 16-class Manchester data by using the SVM RBF kernel classifier, (a) the complete image (b) the selected ROI data set. Note that the number of misclassified pixels in (b) amounts to 126 equivalent to 0.5% error.

Grid search has been a costly method as it involves calculations for many points within the search range for each of the parameters. For instance, if there are 10 intervals in the search for two parameters Gamma and C as in the RBF, then the model needs a 100 point-grid search.

The first objective of this experiment is to illustrate how the cost parameter of the SVM RBF can be found using a grid search method. Secondly, we'd like to make use of the grid search result to deduce the trustworthiness of the TTD and TJM as a means of performance assessment.

10.3.1 SVM RBF parameterisation: Grid search

The experiment is conducted in the same way as that presented in chapter 10.1 using randomly selected 40% of the data as the training set and 100% of the pixels in the ROI for the test data (see chapter 8 for details). In the RBF kernel there are two parameters γ, which inversely scale to the variance of the cluster; and C which controls the softness of the separation plane as mentioned above. The grid search tends to propagate in the directions of increasing variance and at the same time to minimise the γ.

Typical grid search result for the Manchester data set is shown in Figure 10-7, which plots the contour of classification accuracies as functions of the C and γ parameters. The optimum condition according to the search result is nucleated at (γ,c)=(16,128) giving almost 100% classification accuracy at the peak. To verify the results given by the search result as shown in Figure 10-7 , the classifications along three grid points of $C=2^7$ and to pick $\gamma=2^{(0,4,7)}$, the accuracies for using these parameters in the SVM RBF are presented in Figure 10-8 and they are then compared with that of the QD, FD and ED supervised classifiers.

It is evidenced from Figure 10-8 that the SVM RBF has indeed achieved accuracies of 99.5%, 99.7 and 98.6% when the parameters of (γ,C)=($2^{(0,4,7)}$, 2^7) are employed respectively. This classification performance exhibits remarkable accuracy compared to the well-known supervised classifiers such as QD & FD, provided that the optimal parameters of the SVM kernels are found. In contrast, the QD and the FD have achieved accuracies of ~99% and ~95% (see Table 10-3) without the need for elaborate parameterisation routines. Furthermore, the computational cost of these classifiers has been only a fraction of that required by the SVM.

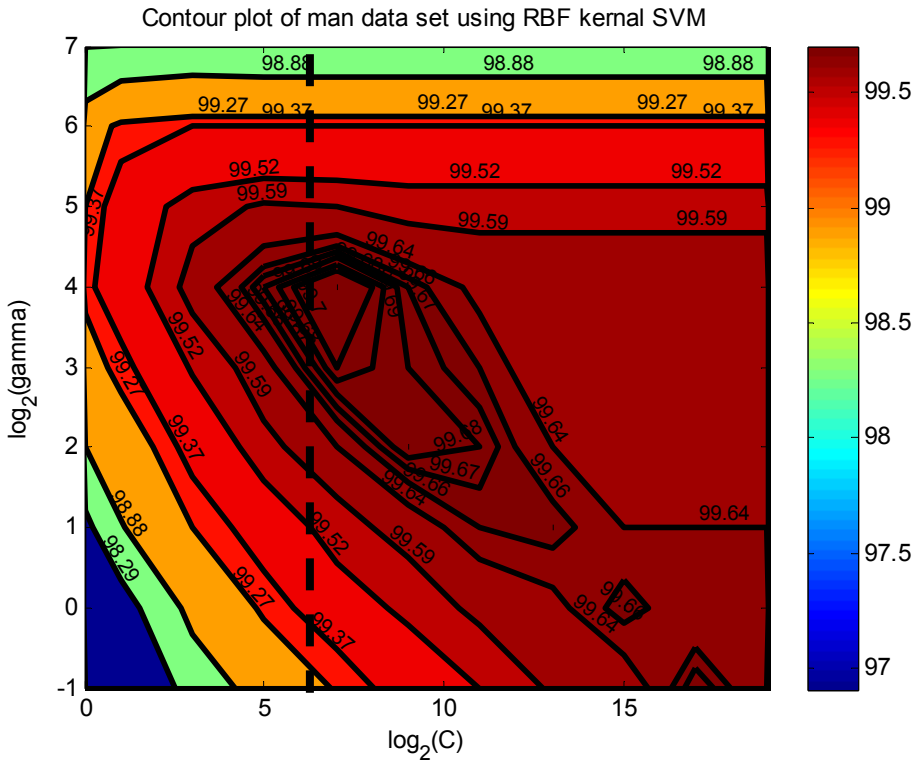


Figure 10-7: The grid search result for the parameterisation of the SVM RBF classifier plotting the contour relationships between the (γ,C) with respected to the classification accuracy. The employed image set is the Manchester data (40% training size) and the dotted line shows the grid points along $C=2^7$.

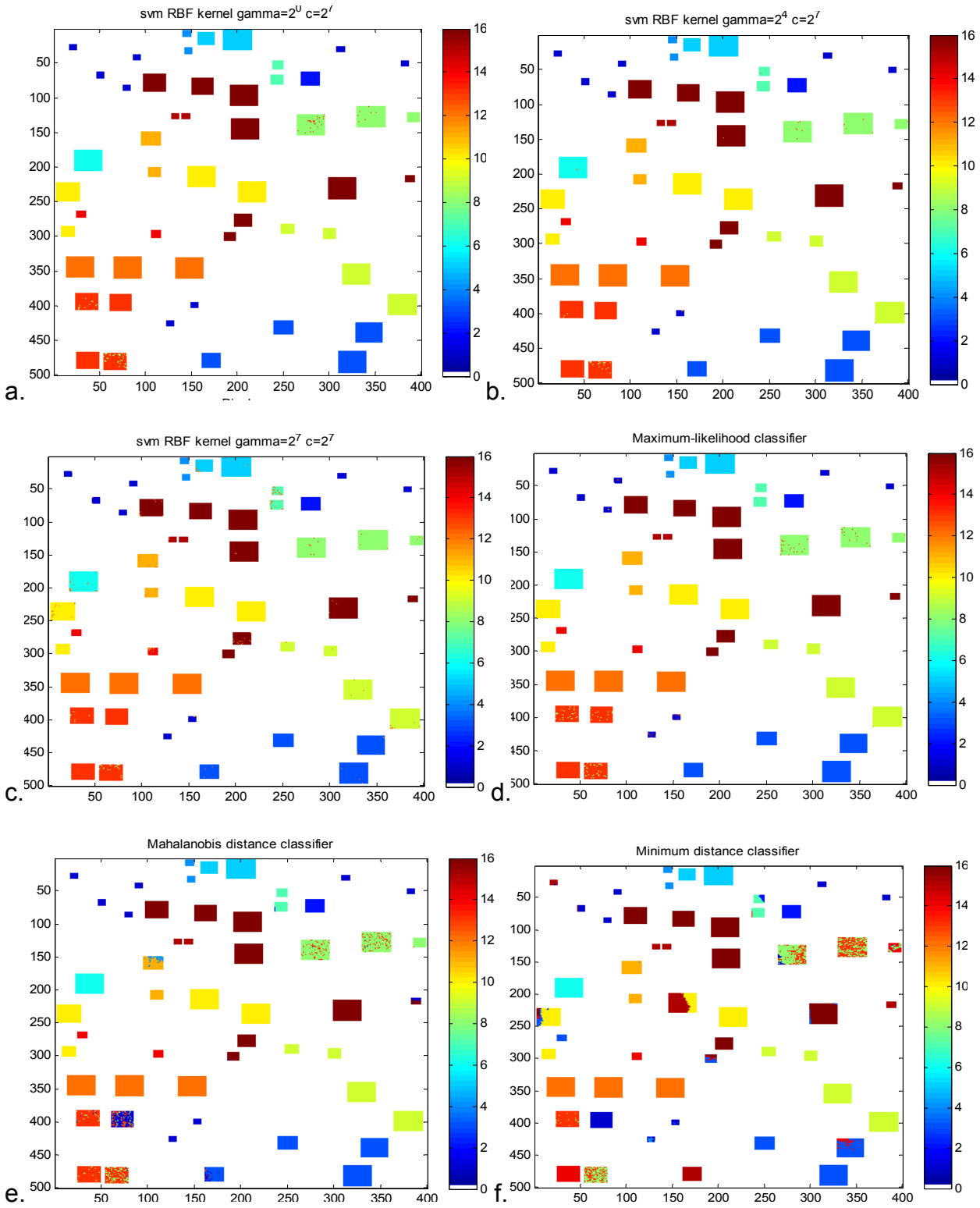
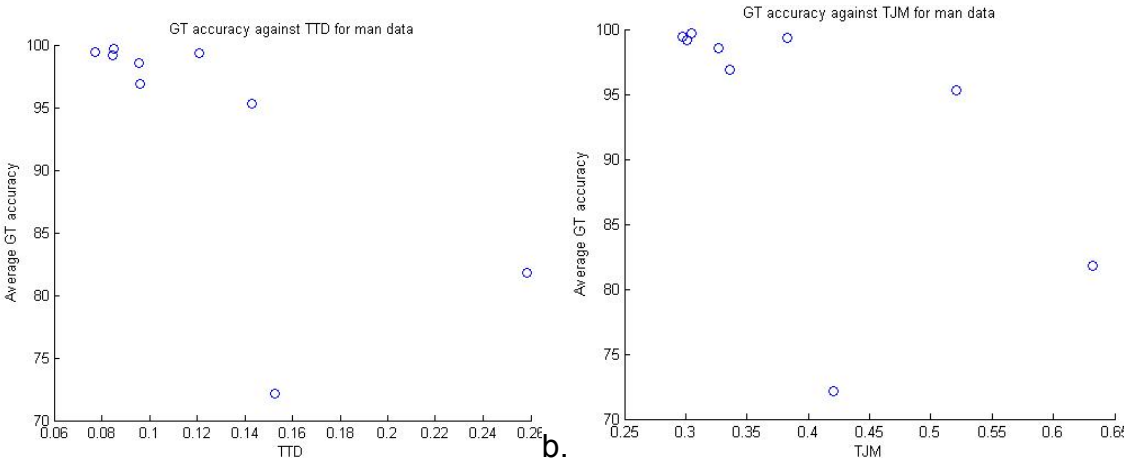


Figure 10-8: Shows the various SVM RBF classification results using parameters of C=2^7 and γ= a) 1, b) 2^4, c) 2^7 along the dashed line of the grid search as shown in Figure 10-7, and their classification accuracies are compared with d) QD, e) FD and f) ED classifiers. Note that the QD has achieved ~99% accuracy close to that of the optimised SVM RBF at (C,γ)=(2^7 , 2^4) with accuracy of ~99.5%.

Classifier	Gamma	C	GT accuracy	TTD score	TJM score
SVM (RBF)	2^0	128 (2^7)	99.47%	0.0773	0.29735
SVM (RBF)	2^4	128 (2^7)	99.73%	0.085	0.30505
SVM (RBF)	2^7	128 (2^7)	98.58%	0.0958	0.32695
SVM (RBF)	2^-1	1	96.91%	0.0961	0.3357
Kmeans	-	-	72.17%	0.1528	0.4207
KNN (k=1)	-	-	99.38%	0.1211	0.3830
Minimum distance classifier	-	-	81.78%	0.2585	0.63215
Mahalanobis distance classifier	-	-	95.34%	0.143	0.52065
Maximum likelihood classifier	-	-	99.13%	0.0848	0.30125

Table 10-3: shows the performances of the SVM and other classifiers for the classification of the Manchester data using 40% training sizes. Note that this experiment uses the ROI pixels of the data set while the experiment that presented in chapter 9 involves classification for the whole image.

It is noted that the classification accuracies that presented in Table 10-3 seemingly exhibited some degree of correlations with the separability measures such as the TTD and TJM. A scatter plot of the GTaccuracy against the TTD & TJM is shown in Figure 10-9, which shows an apparent polynomial-like relationship between the ground truth accuracy and the separability measures. Before the exact form of this polynomial relationship is established, it is essentially important to explore if the relationship is dependent on the data structure or statistical property of the data set.



a. b.

Figure 10-9: shows the scatter plot between the GTaccuracy and the separability measures a) TTD and b) TJM. It is not known if this relationship is dependent on the data characteristics (see next section).

To this end another data set, the lab-t-shirt image, which is characterised by almost uniform class size, is examined here using the same way as that performed in the last experiment. Due to the abundance of training data (minimum sample to band ratio β~68) and relatively uniform class sizes, the grid search shows quite a range of (γ,C) that gives the optimum classification performance of ~100% accuracy (Figure 10-10). Again, similar to the Manchester data set, this excellent performance has also been achieved by other supervised classifiers such as the QD and FD (100% accuracy) but they only need a fraction of the computational cost as that of the SVM (Table 10-4). One main difference between this data set with respect to the Manchester data is that, there are two classes (the yellow ones) which are quite similar to each other spectrally within this T-shirt data, although their class separabilities (such as TD) has shown an ideal value of 2 (see Figure 8-11). Hence the grid search result of Figure 10-10 has exhibited a very steep contour, falling off the accuracies very steeply from the peaked due to the misclassification of these two classes when the parameters are not optimum values.

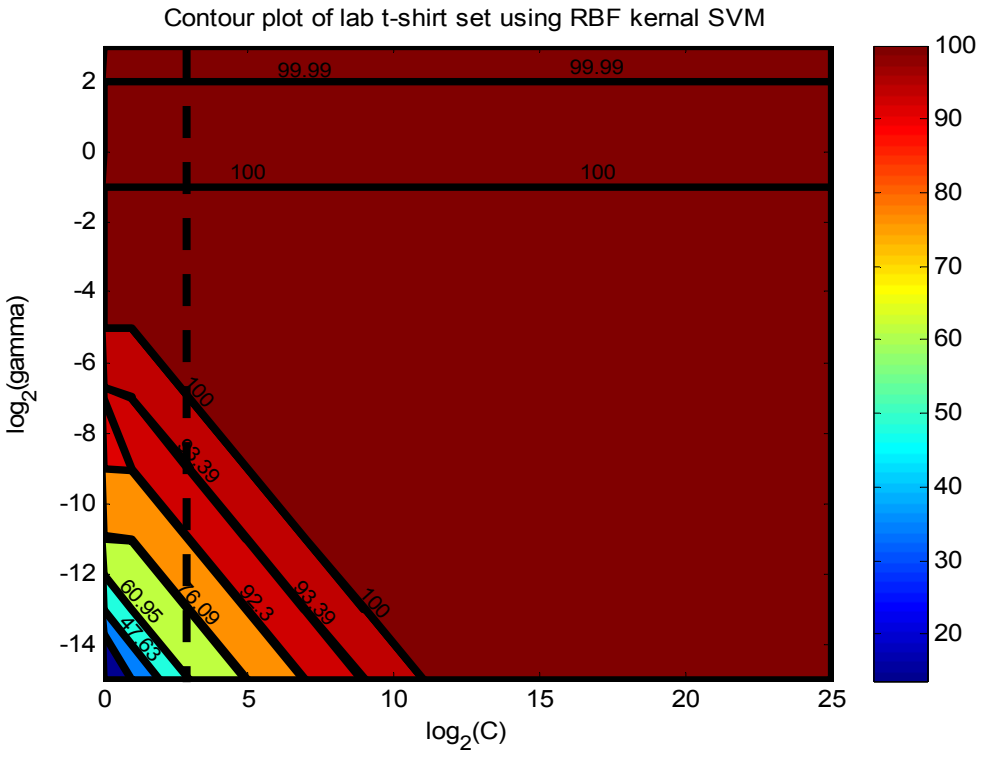


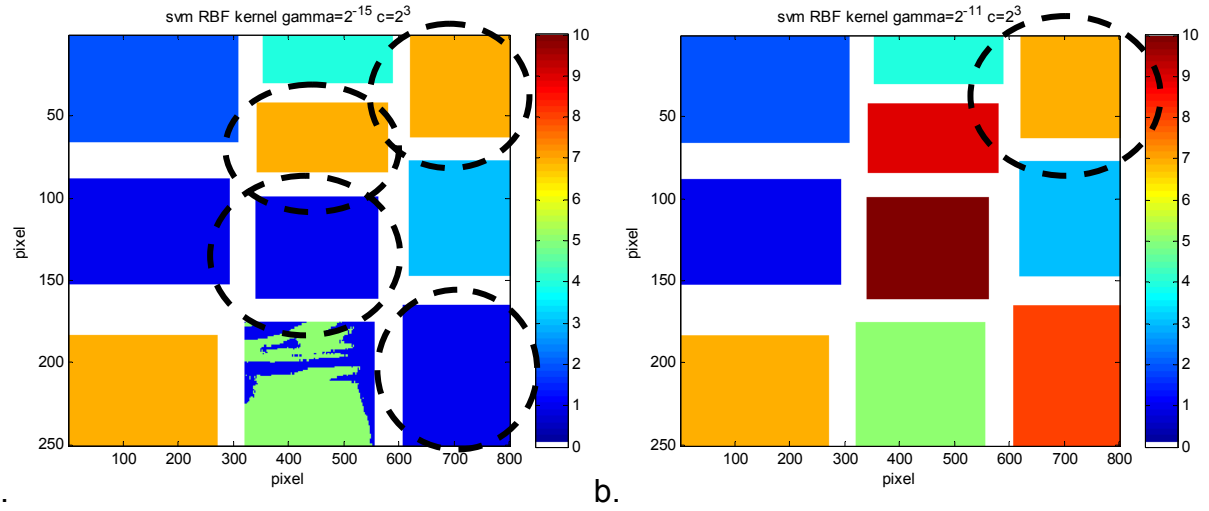
Figure 10-10: The grid search result for the parameterisation of the SVM RBF classifier plotting the contour relationships between the (γ,C) with respected to the classification accuracy. The data set employed is the lab T-shirt (40% training size) and the dotted line shows the grid points along $C=2^3$.

The presence of these two closely related classes in this data set has induced another issue for the calculation of the TTD and TJM. For example, when $(C,\gamma)=(2^3,2^{-15})$ the SVM RBF shows ~61% accuracy (see Table 10-4) and there are 4 classes completely missed. When $(C,\gamma)=(2^3,2^{-11})$ the SVM RBF shows ~92% accuracy with one class completely missed as shown in Figure 10-11. The TD values in these missed classes are zero (see Figure 10-12), and the TTD will be effectively increased by $N_m*(N_c-1)*2$, where N_m and N_c are the number of the missed class and the total number of classes respectively. This induces an artificial abrupt 'jump' on the TTD values.

Classifier	Gamma	C	GT accuracy	TTD score	TJM score
SVM (RBF)	2^--15	2^3	60.95%	60	60.0001
SVM (RBF)	2^--13	2^3	76.09%	48	48
SVM (RBF)	2^-11	2^3	92.30%	18	18
SVM (RBF)	2^-9	2^3	93.38%	0	0.1069
SVM (RBF)	2^-3	2^3	100%	0	0

kmeans	-	-	85.72%	18.2143	18.3106
Minimum distance classifier	-	-	99.83%	1.1759e-10	6.6150e-11
Mahalanobis distance classifier	-	-	100%	0	0
Maximum likelihood classifier	-	-	100%	0	0

Table 10-4: shows the performances of the SVM and other classifiers for the classification of the lab T-shirt data using 40% training sizes.



a. b.

Figure 10-11: highlights the classification results in false colours when classes are missed (circled) using RBF parameters of : a) (C,γ)=(2^3,2^{-15}) with 61% accuracy, b) (C,γ)=(2^3,2^{-11}) with 92% accuracy.

TTD for SVM RBF gamma =2^15 c=2^3 (lab t-shirt data)

Class	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	2	0	2	0	0	0
2	2	0	2	2	2	0	2	0	0	0
3	2	2	0	2	2	0	2	0	0	0
4	2	2	2	0	2	0	2	0	0	0
5	2	2	2	2	0	0	2	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	2	2	2	2	2	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

TJM for SVM RBF gamma =2^15 c=2^3 (lab t-shirt data)

Class	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	1.99988	0	2	0	0	0
2	2	0	2	2	2	0	2	0	0	0
3	2	2	0	2	2	0	2	0	0	0
4	2	2	2	0	2	0	2	0	0	0
5	1.99988	2	2	2	0	0	2	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	2	2	2	2	2	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

a.

TTD for SVM RBF gamma =2^11 c=2^3 (lab t-shirt data)

Class	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	2	0	2	2	2	2
2	2	0	2	2	2	0	2	2	2	2
3	2	2	0	2	2	0	2	2	2	2
4	2	2	2	0	2	0	2	2	2	2
5	2	2	2	2	0	0	2	2	2	2
6	0	0	0	0	0	0	0	0	0	0
7	2	2	2	2	2	0	0	2	2	2
8	2	2	2	2	2	0	2	0	2	2
9	2	2	2	2	2	0	2	2	0	2
10	2	2	2	2	2	0	2	2	2	0

TJM for SVM RBF gamma =2^11 c=2^3 (lab t-shirt data)

Class	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	2	0	2	2	2	2
2	2	0	2	2	2	0	2	2	2	2
3	2	2	0	2	2	0	2	2	2	2
4	2	2	2	0	2	0	2	2	2	2
5	2	2	2	2	0	0	2	2	2	2
6	0	0	0	0	0	0	0	0	0	0
7	2	2	2	2	2	0	0	2	2	2
8	2	2	2	2	2	0	2	0	2	2
9	2	2	2	2	2	0	2	2	0	2
10	2	2	2	2	2	0	2	2	2	0

b.

Figure 10-12: highlights the issue for the calculation of the TTD and TJM when some classes are completely missed in the classification result. The figure shows the TTD and TJM for a) (C,γ)=(2^3,2^{-15}) with TTD of 60, b) (C,γ)=(2^3,2^{-11}) with TTD=18. The very high values of the TTD in these cases are caused by the zero TD in the missed classes (highlighted in yellow).

When all classes are present such as in the case of $(C,\gamma)=(2^3,2^{-9})$ and $(2^3,2^{-3})$, the classifications have shown accuracies of 93% and 100% corresponding to the TJM values of 0.1 and zero respectively (see Figure 10-13). Note that the TTD for these two cases both show zero values and this will be investigated in the next section.

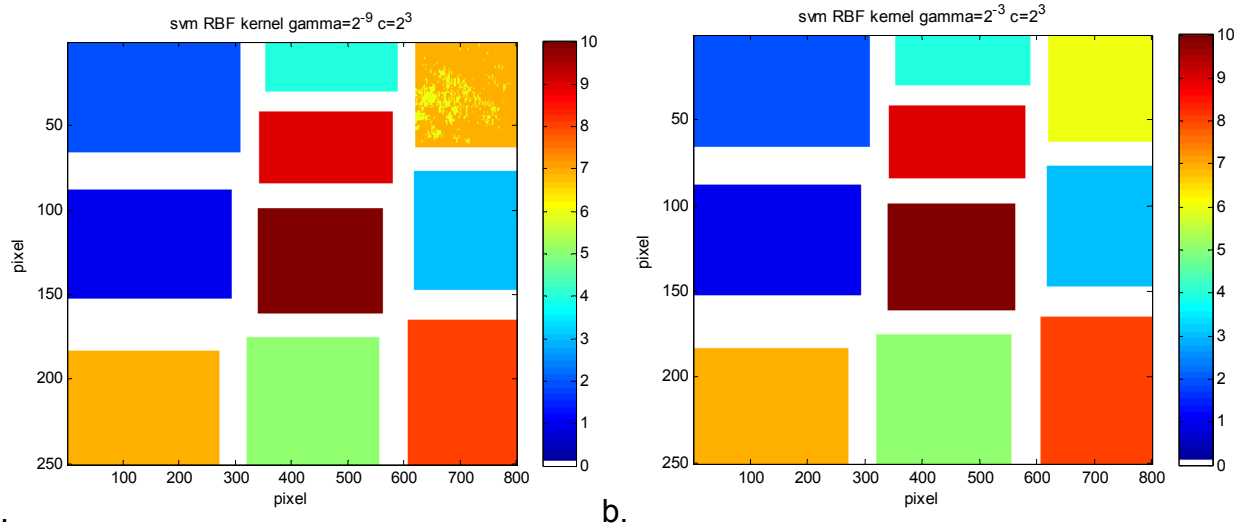


Figure 10-13: shows the classification results in false colours when using slightly non-optimal RBF parameters: a) $(C,\gamma)=(2^3,2^{-9})$ with 93% accuracy, b) $(C,\gamma)=(2^3,2^{-3})$ with 100% GTaccuracy.

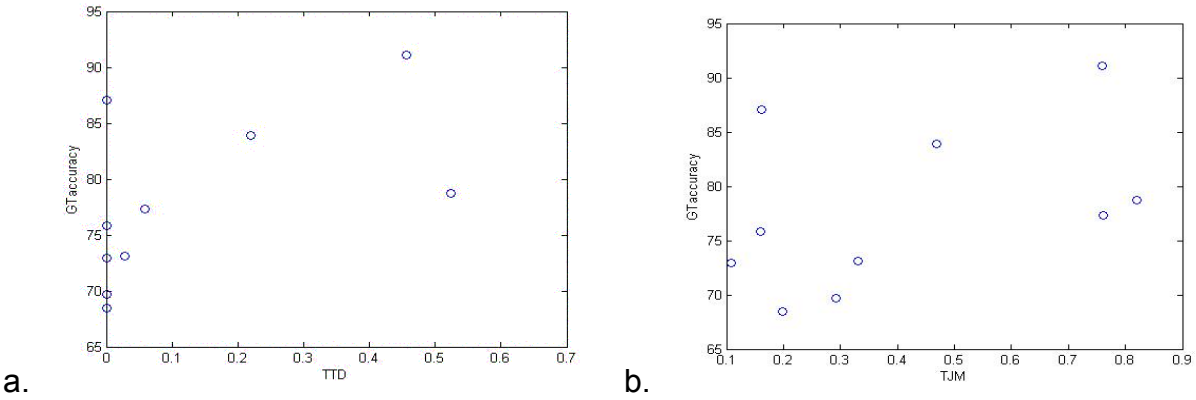
10.4 Separability measures vs ground truth: relationships and issues

It is of interest to study if the GT accuracy can be correlated with the TTD and TJM values according to the results presented in the last few sections. It is noted from the previous section that the current method for the evaluation of the TTD/TJM values according to Equation 6-7 and 6-8 are not valid if there are classes completely missed in the classification result. Henceforth all data presented in this section will be restricted to the classification results that do not miss any classes, i.e. $N_m=0$.

10.4.1 T-shirt data sets and β+ issues

This data set is characterised by having almost uniform class sizes which is advantageous for the proper evaluation of the most important ingredients of the TD and JM: the class covariance. Table 10-5 shows the results obtained from a range of classifiers on various data sets collected during the course of this study, and a selection

of those are then plotted in Figure 10-14 ,which, hardly shows any correlation of the GT accuracy with respected to both the TTD and TJM at all.



a. b.

Figure 10-14: shows the scatter plot between the GTaccuracy and the separability measures for the lab t-shirt data a) TTD and b) TJM. Please refer to Table 10-5 for the complete set of the results.

Recall the TD and JM equations in 6-3 and 6-4 that both techniques require the estimation of the class covariance Σ_i and hence it is important to make sure that the ratios of the number of the samples in each class of the classification result, with respected to the band ratio, denoted in henceforth as β+, are appreciable and in large values typically >60. Otherwise, the covariance will be badly estimated due to the small sample size (for more details please refer to the next chapter).

Classifier	Training data	Test data set	Training sample size	Min class size to band ratio β+	GT accuracy	TTD score	TJM score
kmeans	lab t-shirt	lab t-shirt	40%	0.68	85.720	18.214	18.311
QD	car t-shirt	car t-shirt	101	1.58	77.766	0.049	0.215
QD	car t-shirt	car t-shirt	101	1.72	71.386	0.000	0.230
QD	car t-shirt	car t-shirt	101	3.34	74.080	0.000	0.179
QD	car t-shirt	car t-shirt	101	4.26	61.311	0.000	0.106
QD	cloud t-shirt	cloud t-shirt	101	4.46	71.531	0.000	0.002
QD	Man	Man	40%	5.16	99.130	0.085	0.301
KNN (k=1)	Man	Man	40%	5.22	99.380	0.121	0.383
SVM (RBF)	Man	Man	40%	5.23	99.470	0.077	0.297
SVM (RBF)	Man	Man	40%	5.23	99.730	0.085	0.305
SVM (RBF)	Man	Man	40%	5.23	98.580	0.096	0.327
SVM (RBF)	Man	Man	40%	5.23	96.910	0.096	0.336
QD	car t-shirt	car t-shirt	101	5.34	56.831	0.000	0.046
FD	Man	Man	40%	5.39	95.340	0.143	0.521
QD	car t-shirt	car t-shirt	101	5.48	62.500	0.000	0.012
ED	Man	Man	40%	6.77	81.780	0.259	0.632
QD	cloud t-shirt	cloud t-shirt	101	6.91	80.999	0.000	0.004
QD	shine t-shirt	shine t-shirt	101	7.85	70.817	0.000	0.138
QD	car t-shirt	car t-shirt	101	8.35	68.232	0.405	0.801
QD	car t-shirt	car t-shirt	101	8.49	58.033	0.010	0.264
QD	cloud t-shirt	cloud t-shirt	101	9.76	77.559	0.000	0.055
Kmeans	Man	Man	40%	10.19	72.170	0.153	0.421
QD	lab t-shirt	lab t-shirt	101	11.56	72.975	0.000	0.107
QD	car t-shirt	car t-shirt	101	14.06	78.278	0.000	0.189
QD	lab t-shirt	lab t-shirt	101	15.23	77.344	0.058	0.762
QD	lab t-shirt	lab t-shirt	101	15.59	68.455	0.000	0.198
SVM (RBF)	lab t-shirt	lab t-shirt	40%	15.86	93.380	0.000	0.107
QD	cloud t-shirt	cloud t-shirt	101	16.93	65.912	0.000	0.011
QD	cloud t-shirt	cloud t-shirt	101	17.02	85.477	0.000	0.000
QD	cloud t-shirt	cloud t-shirt	101	17.68	79.834	0.000	0.045
QD	shine t-shirt	shine t-shirt	101	18.24	78.313	0.007	0.024
QD	shine t-shirt	shine t-shirt	101	18.68	90.066	0.000	0.016
QD	car t-shirt	car t-shirt	101	18.74	80.717	0.000	0.345
QD	lab t-shirt	lab t-shirt	101	21.29	78.734	0.524	0.821
QD	shine t-shirt	shine t-shirt	101	21.95	80.194	0.000	0.072
QD	cloud t-shirt	cloud t-shirt	101	23.48	85.951	0.000	0.000
QD	shine t-shirt	shine t-shirt	101	26.73	84.304	0.000	0.083
QD	cloud t-shirt	cloud t-shirt	101	30.57	74.481	0.000	0.035
QD	lab t-shirt	lab t-shirt	101	37.72	75.828	0.000	0.160
QD	lab t-shirt	lab t-shirt	101	37.84	69.664	0.000	0.292
QD	shine t-shirt	shine t-shirt	101	41.05	83.625	0.000	0.017
QD	lab t-shirt	lab t-shirt	101	41.68	83.942	0.219	0.469
QD	shine t-shirt	shine t-shirt	101	47.69	86.933	0.000	0.043
QD	lab t-shirt	lab t-shirt	101	50.4	87.088	0.000	0.162
QD	cloud t-shirt	cloud t-shirt	101	52.45	91.411	0.000	0.007
QD	shine t-shirt	shine t-shirt	101	52.61	91.043	0.053	0.109
QD	cloud t-shirt	cloud t-shirt	101	56.97	91.474	0.427	0.524
QD	shine t-shirt	shine t-shirt	101	61.75	85.125	0.000	0.026
QD	shine t-shirt	shine t-shirt	101	61.97	87.472	0.000	0.003
SVM (RBF)	lab t-shirt	lab t-shirt	40%	67.86	100.000	0.000	0.000
ED	lab t-shirt	lab t-shirt	40%	67.86	99.830	0.000	0.000
FD	lab t-shirt	lab t-shirt	40%	67.86	100.000	0.000	0.000
QD	lab t-shirt	lab t-shirt	40%	67.86	100.000	0.000	0.000
QD	shine t-shirt	shine t-shirt	150	73.43	99.970	0.000	0.000
QD	lab t-shirt	lab t-shirt	101	82.06	73.089	0.028	0.331
QD	lab t-shirt	lab t-shirt	101	99.96	91.095	0.456	0.758

Table 10-5: shows all the classification results performed in this work using a range of classifiers, with a hope to establish the relationship between the GT accuracy with respected to the TTD and TJM scores.

The results in Table 10-5 have been sorted in ascending order of the β+. It clearly shows that the GTaccuracy vs TTD/TJM plots presented in Figure 10-9 and Figure 10-14 for the Manchester and the T-shirt data sets, respectively, contain substantial errors due to the small values of the β+. It is the worst for the Manchester data which typically exhibits very small β+ of ~5. A quick glance at the Table 10-5 shows that there are only a handful of 9 runs that listed at the bottom of the table may be suitable for establishing the relationship between the GTaccuracy and the separability measures (TTD/TJM). Unfortunately, most of these 9 results have got very similar GT accuracy of ~100% and there is only one at ~70%, giving us only two data points which is too few to establish a proper relationship.

It is also noted from the bottom of Table 10-5 that an outliner of TTD/TJM value is seen having a very high separability score even though the GT accuracy is in fact at ~91%.

10.4.2 GTaccuracy simulation results

To understand more about the puzzles raised in the previous sections, a series of experiment is designed hoping to shed some light into the problem and also with a hope to establish a true relationship between the GT accuracy and the TTD/TJM scores. To this end, a simulation experiment is conducted such that the class labels of a control number of pixels in the ground truth maps are artificially altered, creating simulated 'misclassification' situations. Two sets of simulations have been performed using the Manchester and the T-shirt data as the templates, and the 'misclassifications' have been controlled under the following three scenarios:

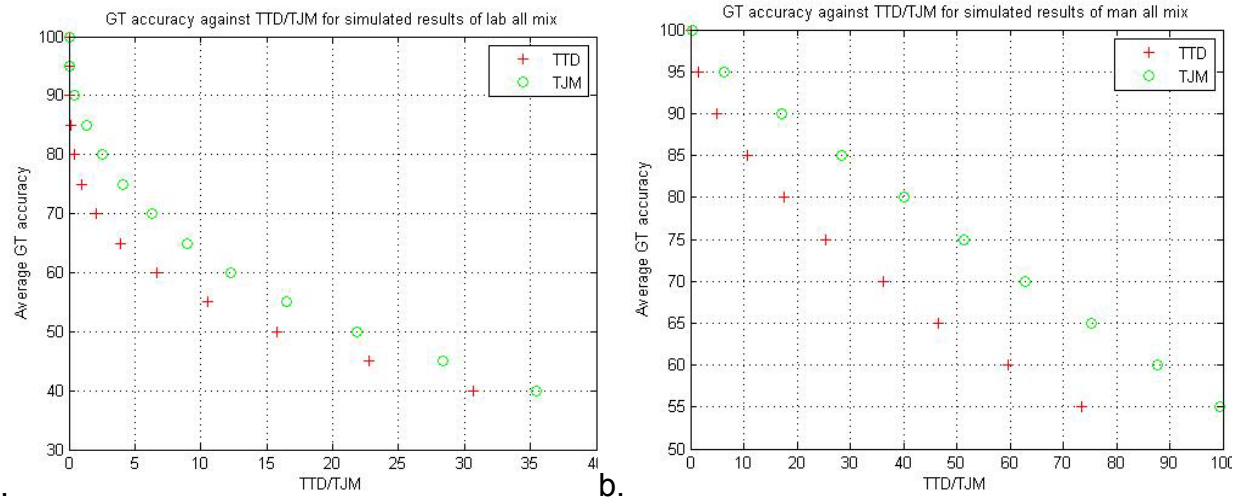
1. All misclassified pixels are randomly selected from ALL classes and this is designated as 'all mix' in Table 10-6
2. All misclassified pixels are randomly selected from 5 classes and this is designated as '5 class mix' in Table 10-6
3. All misclassified pixels are randomly selected from 2 classes and this is designated as '2 class mix' in Table 10-6

Simulation	Data set	Min class size to band ratio β+	GT accuracy	TTD score	TJM score
All mix Sim	lab t-shirt	119.37	39.999	30.643	35.457
All mix Sim	lab t-shirt	116.99	45.000	22.748	28.411
All mix Sim	lab t-shirt	113.9	49.999	15.794	21.880
All mix Sim	lab t-shirt	108.53	55.000	10.524	16.501
All mix Sim	lab t-shirt	102.4	59.999	6.622	12.296
All mix Sim	lab t-shirt	97.81	65.000	3.926	8.954
All mix Sim	lab t-shirt	93.61	69.999	2.106	6.313
All mix Sim	lab t-shirt	89.34	75.000	0.949	4.112
All mix Sim	lab t-shirt	85.16	79.999	0.370	2.526
All mix Sim	lab t-shirt	81.64	85.000	0.108	1.292
All mix Sim	lab t-shirt	76.74	89.999	0.024	0.448
All mix Sim	lab t-shirt	72.13	95.000	0.003	0.055
All mix Sim	lab t-shirt	67.87	99.999	0.000	0.000
5 Class mix	lab t-shirt	99.96	55.000	16.289	16.329
5 Class mix	lab t-shirt	99.96	59.999	17.443	17.458
5 Class mix	lab t-shirt	99.96	65.000	13.805	13.916
5 Class mix	lab t-shirt	99.96	69.999	8.841	9.176
5 Class mix	lab t-shirt	99.96	75.000	4.516	5.054
5 Class mix	lab t-shirt	99.96	79.999	2.155	2.694
5 Class mix	lab t-shirt	98.52	85.000	0.842	1.313
5 Class mix	lab t-shirt	89.79	89.999	0.160	0.479
5 Class mix	lab t-shirt	78.09	95.000	0.005	0.086
5 Class mix	lab t-shirt	67.86	99.999	0.000	0.000
2 Class mix	lab t-shirt	67.86	65.000	1.030	1.068
2 Class mix	lab t-shirt	67.86	69.999	2.029	2.070
2 Class mix	lab t-shirt	67.86	74.999	3.101	3.113
2 Class mix	lab t-shirt	67.86	79.999	3.132	3.142
2 Class mix	lab t-shirt	67.86	84.998	1.623	1.672
2 Class mix	lab t-shirt	67.86	89.998	0.286	0.356
2 Class mix	lab t-shirt	67.86	94.998	0.007	0.019
2 Class mix	lab t-shirt	67.86	99.997	0.000	0.000
All mix Sim	man data	8.55	54.999	73.535	99.485
All mix Sim	man data	7.5	59.998	59.575	87.616
All mix Sim	man data	6.7	64.998	46.614	75.376
All mix Sim	man data	6.16	69.997	36.264	62.958
All mix Sim	man data	5.24	74.996	25.434	51.311
All mix Sim	man data	4.74	79.999	17.589	40.105
All mix Sim	man data	3.95	84.998	10.801	28.286
All mix Sim	man data	3.08	89.998	5.025	17.168
All mix Sim	man data	2.32	94.997	1.520	6.391
All mix Sim	man data	1.62	99.996	0.077	0.294
5 Class mix	man data	7.11	64.998	30.963	61.216
5 Class mix	man data	6.36	69.997	20.903	49.389
5 Class mix	man data	5.79	74.996	13.595	38.262
5 Class mix	man data	5.03	79.999	7.782	27.852
5 Class mix	man data	3.99	84.998	4.629	19.870
5 Class mix	man data	3.27	89.998	2.387	11.015
5 Class mix	man data	2.43	94.997	1.362	4.687
5 Class mix	man data	1.62	99.996	0.083	0.306
2 Class mix	man data	1.62	57.978	0.509	3.328
2 Class mix	man data	1.62	63.991	0.404	3.264
2 Class mix	man data	1.62	69.981	0.327	3.111
2 Class mix	man data	1.62	75.994	0.231	2.867
2 Class mix	man data	1.62	81.984	0.157	2.593
2 Class mix	man data	1.62	87.997	0.102	2.183
2 Class mix	man data	1.62	93.987	0.082	1.507
2 Class mix	man data	1.62	99.976	0.083	0.284

Table 10-6: shows simulated classification results for the T-shirt and Manchester data sets in a controlled manner. Please refer to the text for the full details of the experiment.

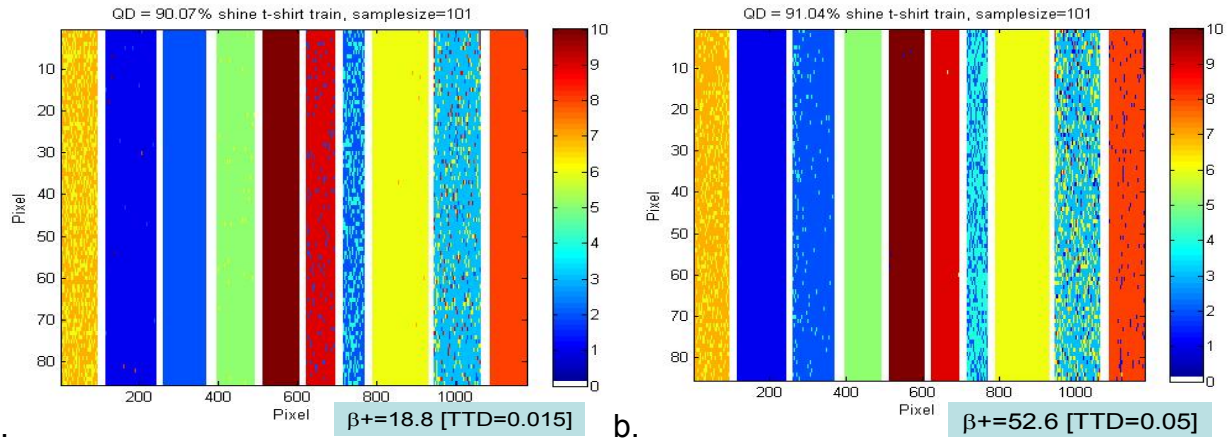
10.4.2.1 Minimum sample to band ratio (β+) issues

The significance of the β+ to the separability scores can be examined according to the simulation results presented in Table 10-6. Figure 10-15 a & b shows the GT accuracy relationship of the all-mixed simulation data with the TTD and TJM using the T-shirt and Manchester classification results, respectively. Noted that the TTD/TJM scores of the T-shirt data that presented in Figure 10-15a has been evaluated from classes with very large β+ values, nominally ~90, whereas the Manchester data (Figure 10-15b) has been evaluated with a maximum β+ values of ~8. Figure 10-15 shows quite clearly that the GT accuracies do indeed scale very well with the separability measures such as the TTD and the TJM, and this relationship is very dependent on the β+ values of the data set. When the β+ values of the classes over ~90 the GT accuracies scale non-linearly with the TTD and TJM (see Figure 10-15a)., and the relationship becomes linear when the β+ values are small (see Figure 10-15b). Thus, this result has shown the very important role of the β+ values for the proper evaluation of the TD/JM scores and thus their total TTD/TJM values. Figure 10-16 demonstrates how the β+ value misleads the TD/JM assessment. Figure 10-16a and Figure 10-16b both have the same GT accuracies but the one with much larger β+ value, Figure 10-16b, shows a much higher TTD scores than the one in Figure 10-16a.



a. b.

Figure 10-15: shows the relationship between the GTaccuracy & the TTD/TJM using the simulated data of the 'all-mixed' classification results: a) the T-shirt data with nominal β+ values of ~90, b) the Manchester data with nominal β+ values of ~5. The plot shows the significance of the β+ values to the TTD evaluation.

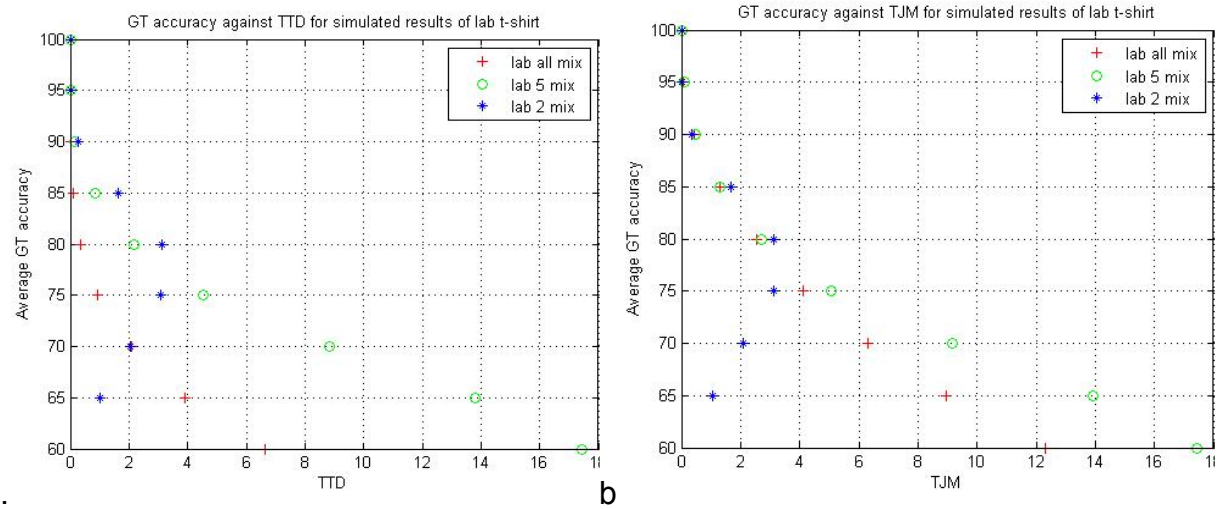


a. b.

Figure 10-16: demonstrates how the β+ value indeed poses an important factor for the evaluation of the TD/JM values: a) β+ values =18.8, TTD=0.015 and b) β+ value = 52.6, TTD=0.05. In both cases the GT accuracy are ~90% but the TTD of (a) is ~4 times less than (b) simply because of the different β+ values.

10.4.2.2 TTD/TJM evaluations issues

Having identified the importance of the β+ values to the separability assessments, it is doubtful if the method for the evaluation of the TTD/TJM using the equations 6-7 & 6-8 are sufficient. Figure 10-17a & b plots TTD/TJM values using the simulation classification results under all-mixed, 5 class mixed and 2 class mixed conditions. It is clear from the figure that the TTD/TJM values are sensitive to the distributions of the misclassified pixels.



a. b

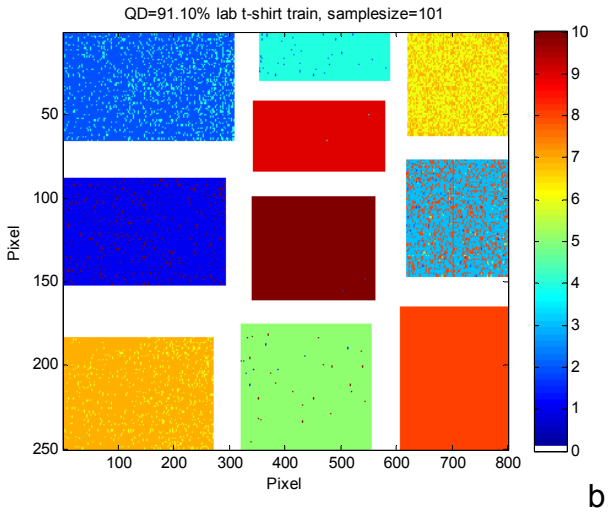
Figure 10-17: casts the doubt if the evaluation methods for the a) TTD and b) TJM are correct. Data presented is the simulation classification results under all-mixed, 5 class mixed and 2 class mixed conditions. It is clear that the TTD values are sensitive to the distributions of the misclassified pixels.

10.4.2.3 One single imperfect class in the classification

It is noted from Table 10-5 that there is one odd result which gives apparently very high TTD and TJM values even though the GT accuracy is in fact achieves 91%. The β+ value of this run has been very high at ~100, and the classification result as shown in Figure 10-18a has indicated the presence of misclassified pixels variably distributed in 5 classes, but most of them are in fact cumulated in a single class. Figure 10-18b shows the pairwised TD and JM values, and there is only one class which exhibits particularly low value, and this is directly translated into the TTD giving an apparently high inaccuracy.

According to the results presented in the last few sections, it is confirmed that separability measures can be an invaluable method for assessing the goodness of classification in principle. However, the present ways for the evaluation of the separability measures are insufficient for achieving this goal and further work in this area is greatly needed.

a.



b.

TTD for lab 90.095

Class	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	2	2	2	2	2	2
2	2	0	2	1.999935	2	2	2	2	2	2
3	2	2	0	2	2	2	2	2	2	2
4	2	1.999935	2	0	2	2	2	2	2	2
5	2	2	2	2	0	2	2	2	2	2
6	2	2	2	2	2	0	1.54403	2	2	2
7	2	2	2	2	2	1.54403	0	2	2	2
8	2	2	2	2	2	2	2	0	2	2
9	2	2	2	2	2	2	2	2	0	2
10	2	2	2	2	2	2	2	2	2	0

TJM

Class	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	2	2	2	2	2	1.997911
2	2	0	2	1.906689	2	2	2	2	2	2
3	2	2	0	2	2	2	2	2	2	2
4	2	1.906689	2	0	2	2	2	2	2	2
5	2	2	2	2	0	2	2	2	2	2
6	2	2	2	2	2	0	1.336929	2	2	2
7	2	2	2	2	2	1.336929	0	2	2	2
8	2	2	2	2	2	2	2	0	2	2
9	2	2	2	2	2	2	2	2	0	2
10	1.997911	2	2	2	2	2	2	2	2	0

Figure 10-18: to investigate the odd result seen in Table 10-5 which gives 'abnormally' high TTD value of 0.45 but the GTaccuracy is in fact 91%. See text for more information.

10.5 Summary

In the first part of this chapter, HSI classification using both SVM modes and kernel functions of linear, polynomial and radial bias Functions have been experimented. It can be seen that one against one (OAO) implementation was the preferred choice compare to one against all (OAA) implementation in all cases. Furthermore, assuming that all parameters were fully optimised, SVM classification could give the best GT accuracy when radial bias function kernel was used.

In the second of this chapter, the first objective of the experiment was to illustrate how the cost parameter of the SVM RBF can be found using a grid search method. The second objective was to deduce the trustworthiness of the TTD and TJM as a means of performance assessment by using of the grid search result. It can be seen that grid search method was suitable for finding optimal parameters of SVM with kernel and cost functions. On the other hand, it had been confirmed that separability measures can be an invaluable method for assessing the goodness of classification in principle. However, the present ways for the evaluation of the separability measures are insufficient for achieving this goal and further work in this area is greatly needed

11 Small sample classifications

11.1 Introduction

Most classifiers such as maximum likelihood require the estimation of the class covariance and mean but in general they are not known and one common way is to estimate them from the training data set. If the sizes of the training data are comparable to the number of features, the covariance and mean are generally good representation of the complete data set. Unfortunately, in many real cases the availability of labelled data is very limited and this leads to a bad classification as the result of badly estimated classifier parameters. In HSI it is well-known that the accuracy of the parameter estimation could achieve within ~1% error provided a sample size to band ratio β of ~100 is available. One way to increase β is via feature extraction or feature selection, however, the discriminate power of classification decreases as useful information is discarded. This chapter investigates how the performances of classifiers are affected as a function of various sizes of training samples. The effectiveness of a couple methods proposed for solving this small sample size problem have been testified in this study.

11.2 Experimental conditions

Two data sets in which one consists of classes in appreciable sizes (lab t-shirt data set with minimum β ~75), and the other having classes in various different sizes with a minimum β~5 (the Manchester data set), have been employed in this experiment. The lab t-shirt data set which consists of 100 bands and training samples of 1000, 500, 200, 150, 130, 110, 50, 20, 10 samples corresponding to sample to band ratio β of 100, 5, 2, 1.5, 1.3, 1.1, 0.5, 0.2 and 0.1 have been utilised in this work. The Manchester data set has 31 bands and training sample sizes of 60, 50, 45, 40, 32, 30, 25, 20, 15, 10, 5, corresponding to the max and min of the β of 2 and 0.17 respectively have been utilised. All experiments are repeated ten times and the averaged GTaccuracy according to Equation 10-1 and 10-2 have been used for assessing the performances of the classifiers.

To highlight the effect, some supervised classifiers that do not involve covariance have been included for a direct comparison. Three different kinds of classifiers have been included in this experiment:

A. Classifiers NOT using covariance:

1. Euclidean distance classifier (ED)
2. SVM with RBF kernel (SVMRBF)

B. Classifiers that use covariance:

1. Mahalanobis distance Classifier (FD)
2. Maximum Likelihood classifier (QD)

C. Classifiers that use the covariance estimated for small sample problems:

1. Maximum Likelihood Classifier with RDA (QD+RDA)
2. Maximum Likelihood Classifier with LOOC (QD+LOOC)

The parameters λ, γ in RDA and α_i in LOOC (Equation 4-12 to 4-15) are found by performing a grid search on the training data to obtain the contour map as functions of maximum likelihood using the leave-one-out method (see section 4.1.1.4). (Landgrebe, 2005; Hoffbeck and Landgrebe, 1996). Then the parameters are chosen which correspond to the highest maximum likelihood for the test data set.

The pseudo-code for running the experiments using QD classifier as an example:

1. Calculate the class mean $m_i = \frac{1}{N}\sum_{j=1}^{N_i} x_{i,j}$; and class sample covariance $\Sigma_i = \frac{1}{N_i - 1}\sum_{j=1}^{N_i}\left(x_{i,j} - m_i\right)^T\left(x_{i,j} - m_i\right)$.

2. Calculate the logarithm of maximum likelihood of all sample x, $g_i(x) = -\left|\Sigma_i\right| - \left(x_i - m_i\right)^T \Sigma_i^{-1}\left(x_i - m_i\right)$.

3. Repeat step 1-2 for L number of times for all other classes

4. Use $g_i(x)$ to determine the class of each test sample.

The pseudo-code for running the experiments using LOOC classifier as example:

1. Choose a value for parameter α_i

2. Remove 1 sample, k, from the training data set of class i

3. Calculate the class mean without k, $m_{i/k} = \frac{1}{N_i - 1}\sum_{\substack{j=1 \\ j\neq k}}^{N_i} x_{i,j}$; and class sample covariance without k, $\Sigma_{i/k} = \frac{1}{N_i - 2}\sum_{\substack{j=1 \\ j\neq k}}^{N_i}\left(x_{i,j} - m_{i/k}\right)^T\left(x_{i,j} - m_{i/k}\right)$

4. Repeat step 1-2 for L number of times for all other classes

5. Calculate the common covariance without k, $S_{i/k} = \frac{1}{N-1}\left(\sum_{\substack{j=1 \\ j\neq i}}^{L}\left(n_j\Sigma_j\right) + n_{i/k}\Sigma_{i/k}\right)$

6. Calculate the LOO covariance $\hat{\Sigma}_{i/k}(\alpha_i)$ from Equation [11-2]

7. Calculate the logarithm of maximum likelihood for a give α_i for all sample x. $g_{i/k}(x,\alpha_i) = -\left|\hat{\Sigma}_{i/k}(\alpha_i)\right| - \left(x_{i/k} - m_{i/k}\right)^T\hat{\Sigma}_{i/k}(\alpha_i)^{-1}\left(x_{i/k} - m_{i/k}\right)$.

8. Repeat step 2 to step 7 until all samples have been removed once

9. Calculated the $LOOL_i(\alpha_i) = \frac{1}{n_i}\sum_{j=1}^{n_i}\ln\left[f\left(x_{i,j} \mid m_{i/k}, \hat{\Sigma}_{i/k}(\alpha_i)\right)\right]$ =>

$LOOL_i(\alpha_i) = \frac{1}{n_i}\sum_{j=1}^{n_i} g_{i/k}(x,\alpha_i)$ for all class L

10. Repeat step 1-10 for a range of α_i

11. Choose the parameter α_i for each class i with the maximum LOOL.

12. Run the QD classifier but replace the sample covariance Σ_i with the new covariance $\hat{\Sigma}_i(\alpha_i)$.

11.3 Results

The classifications by using the above mentioned classifiers for the T-shirt lab and the Manchester data sets are shown in Figure 11-1 and Figure 11-3 respectively. The figures plot the GT accuracy of the classifiers against the sample to band ratio β. It is seen from Figure 11-1 that the classifiers, such as the ED, achieves relatively constant performance independent of sample sizes, than that of the QD which makes use of the training sample for the estimation of the class covariance. Note that although SVM does not need to estimate class covariance for the RBF classification, the small sample size induces larger error in the parameterisation of the kernel and which in turn affects the overall performance.

The FD, which uses the common covariance of the data set via the mean of the overall class covariance estimated from the training samples, shows excellent performances independent of training sample sizes. This is partly because of the more or less uniform class sizes in this data set, and partly due to the highly homogeneous of the scene: all samples have uniform colours (spectra) over the entire t-shirt. These will help a better estimation of the common covariance and thus enhancing the classification efficiency, even when β approaches to as small as ~0.2.

It is seen from Figure 11-2 that both the RDA and the LOOC method have indeed improved the characterisation of the covariance very effectively as evidenced by the much improved classification accuracy at very small β of ~0.5. When using the same classifier (QD) alone without RDA or LOOC, the classification performance is seen to stabilise not until β>3.

The high effectiveness for both of the RDA and the LOOC to help solve small sample size problem is further reinforced by observing similar behaviour from another set of results using the Manchester data under the conditions similar to the experiment as described above (see Figure 11-3).

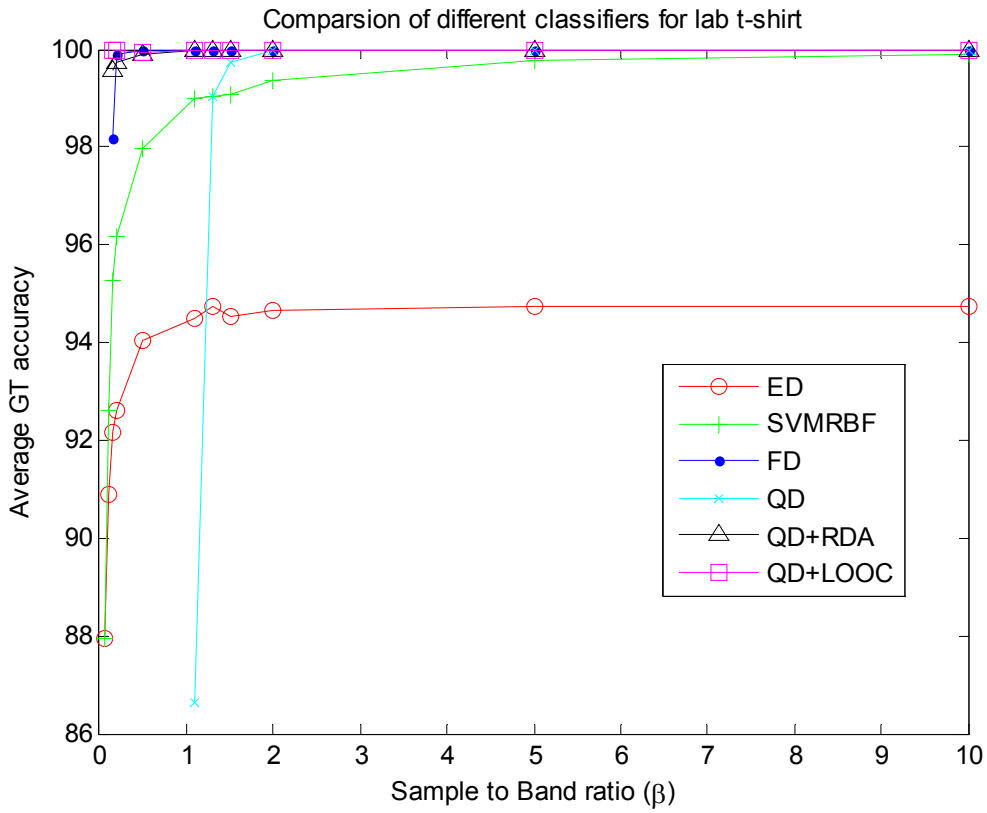


Figure 11-1: Classification results of the lab t-shirt data as function of sample to band ratio β.

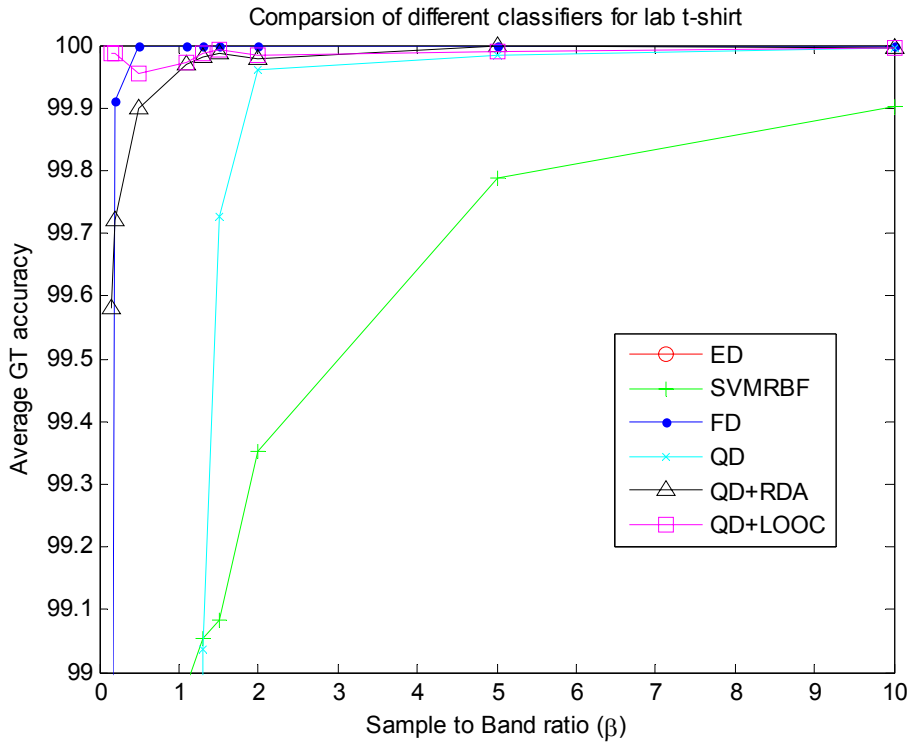


Figure 11-2: A close up view of Figure 11-1, highlighting the effects of the RDA and LOOC for the better characterisation of the covariance of small sample size.

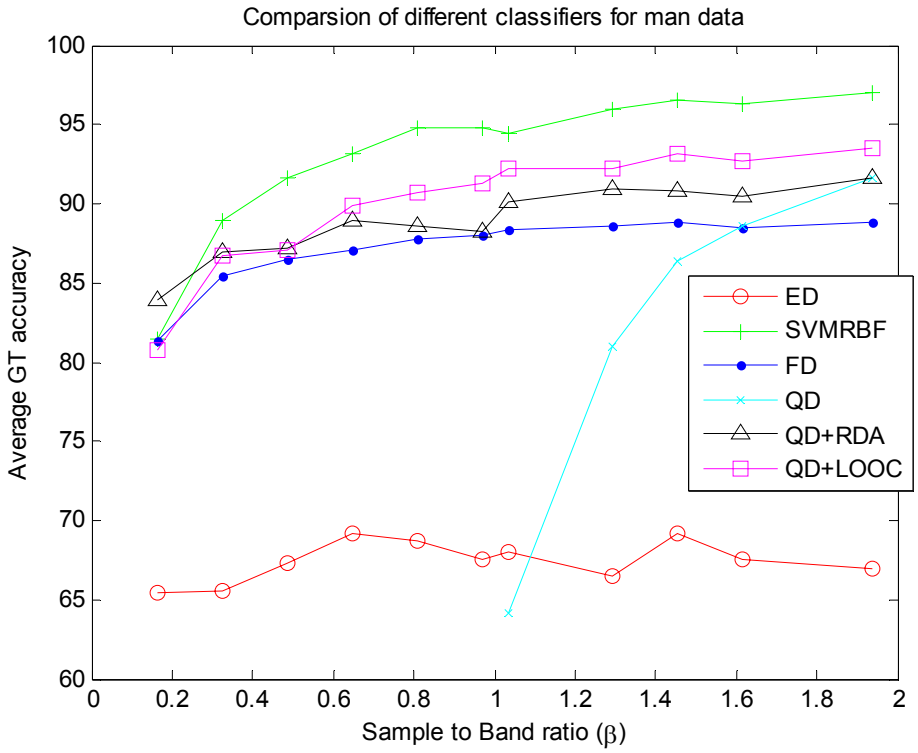


Figure 11-3: Classification results of the Manchester data as function of sample to band ratio β.

12 Conclusions & outlook

The development of classification techniques has been one of the most active research within the machine learning and remote sensing community. The real driving force behind it is the real need of a robust technology for material differentiations. Despite of over half of a century of developments in classification technology, the advantages and disadvantages of each techniques and in particular, the assessment of classification performance for the very high dimensional of hyperspectral imaging (HSI) data sets were hardly documented.

This study exploits a range of classification techniques and the implementation for assessing the effectiveness of hyperspectral classifications using various statistical scoring methods. It is hoped to establish a technique that could evaluate the classification performance without the need of ground truth target map.

Throughout this study the author has conducted an in-depth survey of machine learning and classification theories, and subsequently to implement them for assessing their performances. In this work the author has also helped to establish the HSI instrumentations such as camera calibrations and machine computer interfacing, experimental trials for data collections and instrument maintenances.

This research has involved a range of supervised and unsupervised classifiers for the classification of a number of HSI data sets, and in general the supervised ones such as the Maximum Likelihood (QD) and the Mahalanobis Distance (FD) classifier, especially when they are coupled with techniques like Regularised Discriminant Analysis (RDA) or leave-one-out covariance estimations (LOOC), have shown excellent performances comparable to that of the more complicated and computational costly classifiers like the Support Vector Machine (SVM). It is also found that separability measures such as the Total Transformed Divergence (TTD) and Total Jeffries-Matusita Distance (TJM), can be an invaluable method for assessing the goodness of classification in principle. However, the present ways for the evaluation of the separability measures are insufficient for achieving this goal and further work in this area is greatly needed. This study has also confirmed the effectiveness for using RDA and LOOC techniques for a better estimation

of the covariance when the sample size is small, ie when the sample size per class to band ratio (β) is less than 100.

Through team work this study has contributed partially a number of publications in the area of hyperspectral imaging and machine visions.

12.1 Outlook

During the course of this work it is found that the future research related to the objectives of this work can be pursued in the following directions:

1. Further feature selection and extraction technique for computation cost reduction and at the same time to improve classification efficiency. For example, the dimension of t-shirt data could be reduced to 32 when PCA is utilised which will immediately solve the β ratio problem in the experiments. Therefore future research could be pursued in this direction.
2. Other forms of separability measures such as the entropy. The shortfall of TD and JM could also be verified by collecting data that are not normally distributed. In the future, other types of distributed could be utilised such as the Wishart distribution or mixture of Gaussian. distribution
3. A more robust method for the computation of the TTD and TJM. One way of improvement could be done by weighting the TD/JM score according to the size of pixels for each class. Another way to improve the current method could be done by coupling TD/JM with RDA and LOOC.
4. A more effective scene calibration method: as shown in Figure 8-24, the ELM reflectance of the same target can be seen quite different under different illumination conditions. Simple ELM conversion cannot handle non-linear effects and this induces large errors in the classification. One solution is to capture the scene under controlled environments and therefore all non-linear effects should be eliminated. This may not be practical in capturing nature scene and large area of scene. Another way to solve the problem could done by using some equipments that can calculated the Bidirectional Reflection Distribution Factor (BRDF) for each target. The shortfall of this solution is the excess cost and man power to operate.

13 Appendix

13.1 Distance Measures

Measures	Formula
Minkowski distance	$D_{ij} = \left(\sum_{l=1}^{d} \left| y_{il} - y_{jl} \right|^{\frac{1}{m}} \right)^{m}$
Euclidean distance	$D_{ij} = \left(\sum_{l=1}^{d} \left| y_{il} - y_{jl} \right|^{\frac{1}{2}} \right)^{2}$
City-block distance	$D_{ij} = \sum_{l=1}^{d} \left| y_{il} - y_{jl} \right|$
Mahalanobis distance	$D(x) = \sqrt{(x-\mu)^T C^{-1} (x-\mu)}$ where C is the within group covariance matrix
Pearson correlation distance	$D_{ij} = 1 - r_{ij}$, where $r_{ij} = \frac{\sum_{l=1}^{d} (x_{il} - \overline{x_i})(x_{jl} - \overline{x_j})}{\sqrt{\sum_{l=1}^{d} (x_{il} - \overline{x_i})^2 \sum_{l=1}^{d} (x_{jl} - \overline{x_j})^2}}$
Kullback-Leibler divergence	$D_{KL}(X_{i,} X_j) = \sum_{x \in \Re} p_i(x) \log \frac{p_i(x)}{p_j(x)} + \sum_{x \in \Re} p_j(x) \log \frac{p_j(x)}{p_i(x)}$
Bhattacharyya distance	$B(P\|Q) = \sum_{i=1}^{n} \sqrt{p_i q_i}$
Information Entropy	$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$
Mutual information	$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x) \cdot p(y)}$
Cosine similarity	$S(x,y) = ar \cos \frac{x \cdot y}{\|x\|_2 \|y\|_2}$

13.2 Tables of Search Algorithms (Jain et al., 2000)

Method	Property	Comments
Exhaustive Search	Evaluate all $\binom{d}{m}$ possible subsets.	Guaranteed to find the optimal subset; not feasible for even moderately large values of m and d.
Branch-and-Bound Search	Uses the well-known branch-and-bound search method; only a fraction of all possible feature subsets need to be enumerated to find the optimal subset.	Guaranteed to find the optimal subset provided the criterion function satisfies the monotonicity property; the worst-case complexity of this algorithm is exponential.
Best Individual Features	Evaluate all the m features individually; select the best m individual features.	Computationally simple; not likely to lead to an optimal subset.
Sequential Forward Selection (SFS)	Select the best single feature and then add one feature at a time which in combination with the selected features maximizes the criterion function.	Once a feature is retained, it cannot be discarded; computationally attractive since to select a subset of size 2, it examines only $(d-1)$ possible subsets.
Sequential Backward Selection (SBS)	Start with all the d features and successively delete one feature at a time.	Once a feature is deleted, it cannot be brought back into the optimal subset; requires more computation than sequential forward selection.
"Plus l-take away r" Selection	First enlarge the feature subset by l features using forward selection and then delete r features using backward selection.	Avoids the problem of feature subset "nesting" encountered in SFS and SBS methods; need to select values of l and $r(l > r)$.
Sequential Forward Floating Search (SFFS) and Sequential Backward Floating Search (SBFS)	A generalization of "plus-l take away-r" method; the values of l and r are determined automatically and updated dynamically.	Provides close to optimal solution at an affordable computational cost.

13.3 Kernel trick

Kernel trick is method of using a linear algorithm to solve non-linear problem by mapping the data into a higher dimensional space. The kernel trick is based on Mercer's theorem, given that the function satisfies

1. $K(x,x')$ is continuous
2. $K(x,x')=K(x',x)$. Symmetric
3. $K(x,x')$ is positive-definite, i.e. $\sum_{i.j} a_i a_j K(x_i,x_j)>0$ for any finite subset {x…x_n} of X and real numbers $\{a_i\}_{i=1}^{N}$

If the function satisfies the three criteria, then it can be express as the inner product

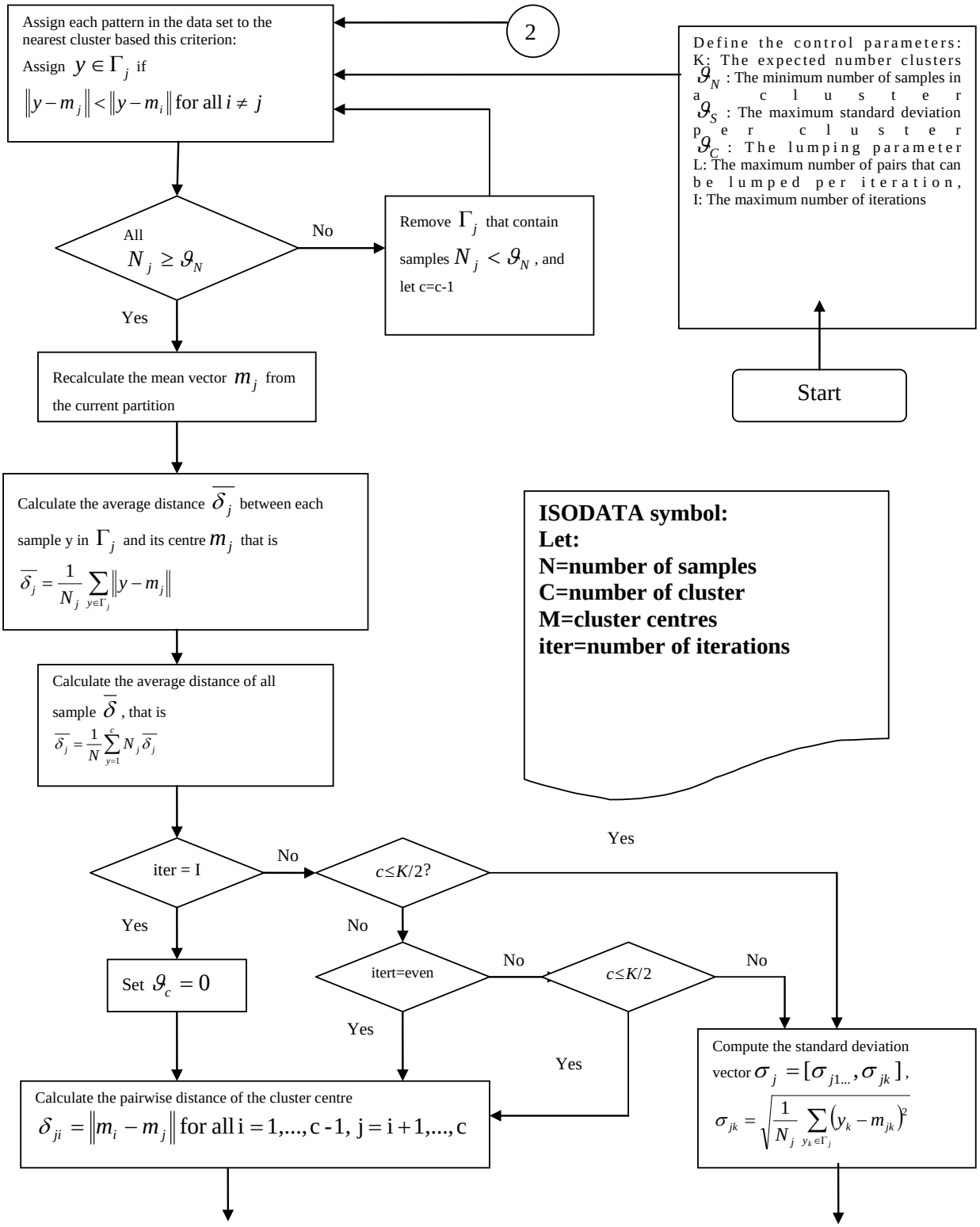
$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

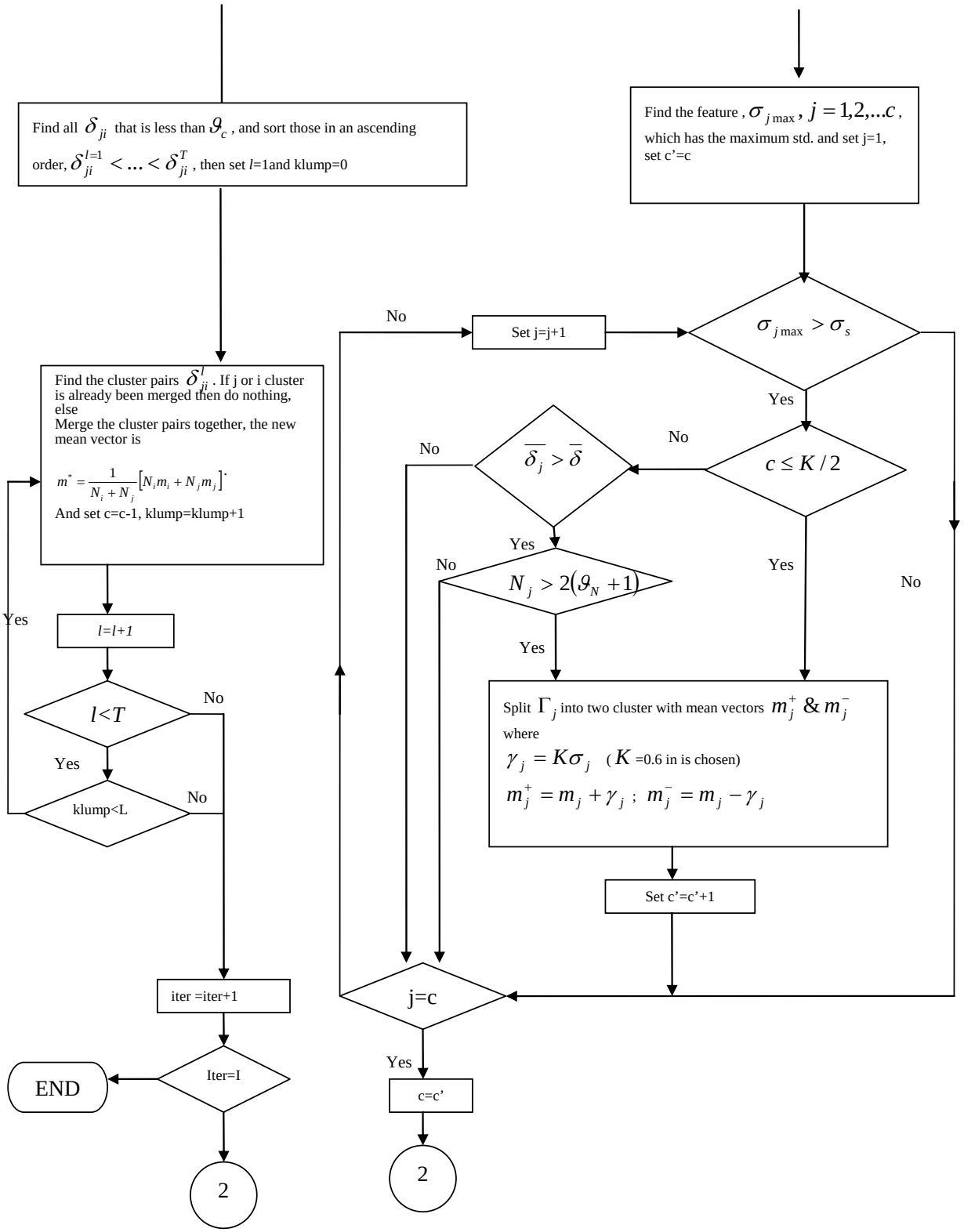
Some common kernels are:

Polynomial: $K(x, x') = (x \cdot x' + t)^d$

Radial Bias Function: $K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2} \right)$

13.4 ISODATA flow chart




Find all δ_{ji} that is less than ϑ_c, and sort those in an ascending order, $\delta_{ji}^{l=1} < \ldots < \delta_{ji}^{T}$, then set l=1and klump=0
Find the cluster pairs δ_{ji}^{l}. If j or i cluster is already been merged then do nothing, else
Merge the cluster pairs together, the new mean vector is
$m^* = \frac{1}{N_i + N_j}\left[N_i m_i + N_j m_j\right]$.
And set c=c-1, klump=klump+1
l=l+1
l<T
No
Yes
klump<L
No
Yes
iter =iter+1
Iter=I
END
2
Find the feature , $\sigma_{j\max}$, $j = 1,2,\ldots c$, which has the maximum std. and set j=1, set c'=c
$\sigma_{j\max} > \sigma_s$
No
Set j=j+1
Yes
$c \le K/2$
No
$\overline{\delta_j} > \overline{\delta}$
No
Yes
$N_j > 2(\vartheta_N + 1)$
No
Yes
Split Γ_j into two cluster with mean vectors m_j^+ & m_j^-
where
$\gamma_j = K\sigma_j$ (K =0.6 in is chosen)
$m_j^+ = m_j + \gamma_j$; $m_j^- = m_j - \gamma_j$
Set c'=c'+1
j=c
Yes
c=c'
2

14 Reference

Aikio, M. (2001), "Hyperspectral prism-grating-prism imaging spectrograph", *VTT Publications,* .

Atukorale, A. S. and Suganthan, P. N. (1999), "Combining multiple HONG networks for recognizing unconstrainedhandwritten numerals", *Neural Networks, 1999. IJCNN'99. International Joint Conference on,* Vol. 4, .

Ball, G. H. and Hall, D. J. (1965), "ISODATA, a novel method of data analysis and pattern classification", .

Baofeng Guo, Gunn, S. R., Damper, R. I. and Nelson, J. D. B. (2006), "Band Selection for Hyperspectral Image Classification Using Mutual Information", *Geoscience and Remote Sensing Letters, IEEE,* vol. 3, no. 4, pp. 522-526.

Baugh, W. M. and Groeneveld, D. P. (2008), "Empirical proof of the empirical line", *International Journal of Remote Sensing,* vol. 29, no. 3, pp. 665-672.

Beisl, U. and Woodhouse, N. (2004), "Correction of atmospheric and bidirectional effects in multispectral ADS40 images for mapping purposes", *Int.Arch.Photogramm.Remote Sens,* vol. 35.

Bellman, R. (1961), *Adaptive Control Processes: A Guided Tour,* First Edition ed, Princeton University Press.

Berk, A., Bernstein, L. S., Anderson, G. P., Acharya, P. K., Robertson, D. C., Chetwynd, J. H. and Adler-Golden, S. M. (1998), "MODTRAN Cloud and Multiple Scattering Upgrades with Application to AVIRIS", *Remote Sensing of Environment,* vol. 65, no. 3, pp. 367-375.

Bezdek, J. C. (1981), "Pattern recognition with fuzzy objective function algorithms", .

Borman, S. (2004), "The Expectation Maximization Algorithm A short tutorial", *Unpublished paper available at* http://www.seanborman.com/publications, .

Boser, B. E., Guyon, I. M. and Vapnik, V. N. (1992), "A training algorithm for optimal margin classifiers", *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory,* Pittsburgh, Pennsylvania, United States, ACM, New York, NY, USA, pp. 144.

Bowles, J., Kappus, M., Antoniades, J., Baumback, M., Czarnaski, M., Davis, C. and Grossmann, J. (1998), "Calibration of inexpensive pushbroom imaging spectrometers", *Metrologia,* vol. 35, pp. 657-661.

Chang, C. I. and Du, Q. (1999), "Interference and noise-adjusted principal components analysis", *IEEE Transactions on Geoscience and Remote Sensing,* vol. 37, no. 5, pp. 2387-2396.

Chen, C. H. and Peter Ho, P. -. (2008), "Statistical pattern recognition in remote sensing", *Pattern Recognition,* vol. 41, no. 9, pp. 2731-2741.

Cohen, J. (1960), "A coefficient of agreement for nominal scales", *Educational and psychological measurement,* vol. 20, no. 1, pp. 37-46.

Congalton, R. G. and Green, K. (1999), *Assessing the accuracy of remotely sensed data: principles and practices,* CRC Press.

Congalton, R. G. and Mead, R. A. (1994), "A quantitative method to test for consistence and correctness in photointerpretation", *Remote sensing thematic accuracy assessment: a compendium,* , pp. 260.

Cortes, C. and Vapnik, V. (1995), "Support-Vector Networks", *Machine Learning,* vol. 20, no. 3, pp. 273-297.

Davis, C., Bowles, J., Leathers, R., Korwan, D., Downes, T. V., Snyder, W., Rhea, W., Chen, W., Fisher, J. and Bissett, P. (2002), "Ocean PHILLS hyperspectral imager: design, characterization, and calibration", *Optics Express,* vol. 10, no. 4, pp. 210-221.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977), "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society.Series B (Methodological),* vol. 39, no. 1, pp. 1-38.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000), *Pattern Classification 2nd Edition,* John Wiley & Sons.

Dunn, J. C. (1973), "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters", *Cybernetics and Systems,* vol. 3, no. 3, pp. 32-57.

Efron, B. and Tibshirani, R. (1997), "Improvements on cross-validation: The. 632 bootstrap method", *Journal of the American Statistical Association,* , pp. 548-560.

Espiner, T. (2009)*, Acpo: Police swamped by CCTV data*, available at: http://news.zdnet.co.uk/security/0,1000000189,39652586,00.htm.

Fisher, J., Baumback, M., Bowles, J., Grossmann, J. and Antoniades, J. (1998), "Comparison of low-cost hyperspectral sensors", *Proc. SPIE,* Vol. 3438, pp. 23.

Foody, G. M. (2002), "Status of land cover classification accuracy assessment", *Remote Sensing of Environment,* vol. 80, no. 1, pp. 185-201.

Fraley, C. and Raftery, A. E. (1998), "How many clusters? Which clustering method? Answers via model-based cluster analysis", *The Computer Journal,* vol. 41, no. 8, pp. 578.

Friedman, J. H. and Tukey, J. W. (1988), "A projection pursuit algorithm for exploratory data analysis", *The Collected Works of John W.Tukey: Graphics 1965-1985, Volume V,* , pp. 149.

Green, A., Berman, M., Switzer, P. and Craig, M. D. (1988), "A transformation for ordering multispectral data in terms of image quality with implications for noise removal", *IEEE Transactions on Geoscience and Remote Sensing,* vol. 26, no. 1, pp. 65-74.

Hayden, A. F. and Twede, D. R. (2002), "Observations on the relationship between eigenvalues, instrument noise, and detection performance", *Proceedings of SPIE,* Vol. 4816, pp. 355.

Ho, T. K., Hull, J. J. and Srihari, S. N. (1994), "Decision combination in multiple classifier systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 16, no. 1, pp. 66-75.

Hoffbeck, J. P. and Landgrebe, D. A. (1996), "Covariance matrix estimation and classification with limited training data", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 18, no. 7, pp. 763-767.

Hughes, G. (1968), "On the mean accuracy of statistical pattern recognizers", *Information Theory, IEEE Transactions on,* vol. 14, no. 1, pp. 55-63.

Hyvärinen, A. and Oja, E. (2000), "Independent component analysis: algorithms and applications", *Neural Networks,* vol. 13, no. 4-5, pp. 411-430.

Jain, A. K., Murty, M. and Flynn, P. (1999), "Data clustering: a review", *ACM computing surveys (CSUR),* vol. 31, no. 3, pp. 264-323.

Jain, A. and Zongker, D. (1997), "Feature selection: evaluation, application, and small sample performance", *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, no. 2, pp. 153-158.

Jain, A. K., Duin, R. P. W. and Mao, J. (2000), "Statistical pattern recognition: A review", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 1, pp. 4-37.

Jia, X. and Richards, J. A. (1999), "Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification", *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 37, no. 1, pp. 538-542.

Jimenez, L. O. and Landgrebe, D. A. (1999), "Hyperspectral data analysis and supervised feature reduction viaprojection pursuit", *IEEE Transactions on Geoscience and Remote Sensing,* vol. 37, no. 6, pp. 2653-2667.

Junping Zhang, Ye Zhang and Tingxian Zhou (2001), "Classification of hyperspectral data using support vector machine", *Image Processing, 2001. Proceedings. 2001 International Conference on,* Vol. 1, pp. 882.

Kaiser Optical Systems, I. (1994), *HoloSpec VPT System Operations Manual* , .

Karpouzli, E. and Malthus, T. (2003), "The empirical line method for the atmospheric correction of IKONOS imagery", *International Journal of Remote Sensing,* vol. 24, no. 5, pp. 1143-1150.

Keshava, N. (2004), "Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries", *IEEE Transactions on Geoscience and Remote Sensing,* vol. 42, no. 7, pp. 1552-1565.

Kittler, J. (1998), "Combining classifiers: A theoretical framework", *Pattern Analysis & Applications,* vol. 1, no. 1, pp. 18-27.

Kohonen, T. (1998), "The self-organizing map", *Neurocomputing,* vol. 21, no. 1-3, pp. 1-6.

Kruse, F. A., Raines, G. L. and Watson, K. (1985), "Analytical techniques for extracting geologic information from multichannel airborne spectroradiometer and airborne imaging spectrometer data", *International Symposium on Remote Sensing of Environment, Fourth Thematic Conference," Remote Sensing for Exploration Geology", San Francisco, California,* pp. 1.

Kumar, S., Ghosh, J. and Crawford, M. M. (2002), "Hierarchical fusion of multiple classifiers for hyperspectral data analysis", *Pattern Analysis & Applications,* vol. 5, no. 2, pp. 210-220.

Landgrebe, D. A. (2005), *Signal theory methods in multispectral remote sensing,* Wiley-Interscience.

Landgrebe, D. (2002), "Hyperspectral image data analysis", *Signal Processing Magazine, IEEE,* vol. 19, no. 1, pp. 17-28.

MacQueen, J. B. (1966), "Some methods for classification and analysis of multivariate observations", .

Marin, J. A., Brockhaus, J., Rolf, J., Shine, J., Schafer, J. and Balthazor, A. (1999), "Assessing band selection and image classification techniques onHYDICE hyperspectral data", *1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings,* Vol. 1, .

Martinez-Uso, A., Pla, F., Sotoca, J. M. and Garcia-Sevilla, P. (2007), "Clustering-based hyperspectral band selection using information measures", *IEEE Transactions on Geoscience and Remote Sensing,* vol. 45, no. 12, pp. 4158-4171.

Mathieu Fauvel, Jocelyn Chanussot and Jon Atli Benediktsson (2006), "Kernel Principal Component Analysis for Feature Reduction in Hyperspectrale Images Analysis", *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic,* pp. 238.

Melgani, F. and Bruzzone, L. (2004), "Classification of hyperspectral remote sensing images with support vector machines", *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 42, no. 8, pp. 1778-1790.

Narendra, P. M. and Fukunaga, K. (1977), "A Branch and Bound Algorithm for Feature Subset Selection", *Computers, IEEE Transactions on,* vol. C-26; C-26, no. 9, pp. 917-922.

Nascimento, S. M. C., Ferreira, F. P. and Foster, D. H. (2002), "Statistics of spatial cone-excitation ratios in natural scenes", *Journal of the Optical Society of America A,* vol. 19, no. 8, pp. 1484-1490.

Ouyang, Y. C., Chen, H. M., Chai, J. W., Chen, C. C., Chen, C. C. C., Poon, S. K., Yang, C. W. and Lee, S. K. (2008), "Independent component analysis for magnetic resonance image analysis", *EURASIP Journal on Advances in Signal Processing,* vol. 2008.

Pal, M. and Mather, P. M. (2004), "Assessment of the effectiveness of support vector machines for hyperspectral data", *Future Generation Computer Systems,* vol. 20, no. 7, pp. 1215-1225.

Richards, J. A. and Jia, X. (2006), *Remote sensing digital image analysis: an introduction,* Springer.

Rosenblatt, F. (1958), "The perceptron: A probabilistic model for information storage and organization in the brain, 1958", *Psychological review,* vol. 65, pp. 386-408.

Roweis, S. T. and Saul, L. K. (2000), "Nonlinear dimensionality reduction by locally linear embedding", *Science,* vol. 290, no. 5500, pp. 2323.

Scholkopf, B., Smola, A. and Muller, K. R. (1998), "Nonlinear component analysis as a kernel eigenvalue problem", *Neural computation,* vol. 10, no. 5, pp. 1299-1319.

Scholkopf, B., Smola, A. J. and Muller, K. R. (1997), "Kernel principal component analysis", *Lecture notes in computer science,* vol. 1327, pp. 583-588.

Serpico, S. B. and Bruzzone, L. (2001), "A new search algorithm for feature selection in hyperspectral remote sensing images", *Geoscience and Remote Sensing, IEEE Transactions on,* vol. 39, no. 7, pp. 1360-1367.

Shaw, G. and Burke, H. (2003), "Spectral imaging for remote sensing", *Lincoln Laboratory Journal,* , no. 14, pp. 3-28.

Swain, P. H. and Davis, S. M. (1978), *Remote sensing: the quantitative approach,* McGraw-Hill New York.

Swain, P. H. and King, R. C. (1973), "Two effective feature selection criteria for multispectral remote sensing", *Proceedings of the 1st International Joint Conference on Pattern Recognition, IEEE,* Vol. 73, pp. 536.

Switzer, P. and Green, A. (1984), "Min/Max autocorrelation factors for multivariate spatial imagery: Technical Report No. 6, Department of Statistics", .

Tenenbaum, J. B., Silva, V. and Langford, J. C. (2000), "A global geometric framework for nonlinear dimensionality reduction", *Science,* vol. 290, no. 5500, pp. 2319.

Tsai, F., Lin, E. K. and Yoshino, K. (2007), "Spectrally segmented principal component analysis of hyperspectral imagery for mapping invasive plant species", *International Journal of Remote Sensing,* vol. 28, no. 5-6, pp. 1023-1040.

Vapnik, V. N. (2000), *The nature of statistical learning theory,* Springer Verlag.

Vermote, E. F., Tanré, D., Deuzé, J. L., Herman, M. and Morcrette, J. -. (1997), "Second simulation of the satellite signal in the solar spectrum, 6s: an overview", *IEEE Transactions on Geoscience and Remote Sensing,* vol. 35, no. 3, pp. 675-686.

Waske, B. and Benediktsson, J. A. (2007), "Fusion of support vector machines for classification of multisensor data", *IEEE Transactions on Geoscience and Remote Sensing,* vol. 45, no. 12, pp. 3858.

Watanabe, S. (1969), "Theorem of the ugly duckling", in *Knowing and guessing: A quantitative study of inference and information,* John Wiley & Sons, , pp. 376-377.

Webb, A. (1999), *Statistical pattern recognition,* A Hodder Arnold Publication.

Welling, M. (2006), "Fisher linear discriminant analysis", *Department of Computer Science, University of Toronto,* .

Xu, R. and Wunsch, D. (2005), "Survey of clustering algorithms", *IEEE Transactions on Neural Networks,* vol. 16, no. 3, pp. 645-678.

Yuen, P., Bishop, G. and Blagg, A. (2005), "Geometric and statistical spectral unmixing for subpixel target detection", EMRS DTC annual conference, .

Zeng, H. and Trussell, H. J. (2004), "Dimensionality reduction in hyperspectral image classification", *Image Processing, 2004. ICIP '04. 2004 International Conference on,* Vol. 2, pp. 913.