ABSTRACT

Title of thesis: A PERFORMANCE CHARACTERIZATION OF KERNEL-BASED ALGORITHMS FOR ANOMALY DETECTION IN HYPERSPECTRAL IMAGERY

Hirsh Goldberg
Master of Science, 2007

Thesis directed by: Professor Rama Chellappa
Department of Electrical Engineering

This thesis provides a performance comparison of linear and nonlinear subspace-based anomaly detection algorithms. Using a dual-window technique to separate the local background into inner- and outer-window regions, pixel spectra from each region are projected onto subspaces defined by projection vectors that are generated using three common pattern classification techniques; the detection performances of these algorithms are then compared with the Reed-Xiaoli anomaly detector. Nonlinear methods are derived from each of the linear methods using a kernelization process that involves nonlinearly mapping the data into a high-dimensional feature space and replacing all dot products with a kernel function using the *kernel-trick*. A projection separation statistic determines how anomalous each pixel is. These algorithms are implemented on five hyperspectral images and performance comparisons are made using receiver operating characteristic (ROC) curves. Results indicate that detection performance is data dependent but that the nonlinear methods generally outperform their corresponding linear algorithms.

A PERFORMANCE CHARACTERIZATION OF KERNEL-BASED ALGORITHMS FOR ANOMALY DETECTION IN HYPERSPECTRAL IMAGERY

by

Hirsh R. Goldberg

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2007

Advisory Committee:
Professor Rama Chellappa, Chair/Co-Advisor
Dr. Nasser Nasrabadi, Co-Advisor
Professor Steven Tretter

Dedication

To Debbie.

Thank you for your never-ending support and patience throughout this entire process.

Acknowledgements

I would first like to thank my advisory committee, Dr. Steven Tretter, Dr. Rama Chellappa, and Dr. Nasser Nasrabadi, for their suggestions and input for this thesis. In addition, I want to thank Volkan Cevher, Amit Banerjee, Heesung Kwon, and Joshua Broadwater for their willingness to help me with some of the technical details along the way. A very sincere thanks goes to Dr. Rama Chellappa, whose generosity made this research possible and was very much appreciated. Finally, a special thank you to Dr. Nasser Nasrabadi for his time, input, and overall support.

This research was partially funded by the Army Research Laboratory during the summer of 2006. All other times from August 2005 until May 2007 were funded by Grant # G41Z93G4 as part of the Multiple University Research Initiative (MURI) from the Army Research Office.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Hyperspectral image analysis, otherwise known as image spectroscopy, involves the use of data acquired from a special type of electro-optical sensor to perform some desired remote sensing application. Hyperspectral imaging (HSI) sensors are essentially the same as multispectral sensors, which have been widely used since the 1970s [1], with the simple distinction that HSI sensors use a greater number of spectral bands and the sequence of bands used is contiguous [2].

For many years, researchers have developed schemes to exploit the highly useful spectral information often provided by a hyperspectral sensor. Whether for classification, detection, or surveillance and tracking purposes, many of these methods have shown promise in terms of being able to effectively analyze a hyperspectral scene [2–16]. For target detection applications, both spectral match detectors and anomaly detectors have garnered a great deal of attention. These two classes of detectors have one key difference. Spectral match detectors require the spectra of the materials in question to be known *a priori*; anomaly detectors do not have this requirement. Instead, they search for pixels in-scene which have spectra that deviate significantly (in a statistical sense) from those spectra which are part of a globally assumed background or their respective local backgrounds.

The main goal of an anomaly detector is to distinguish target pixels from

background clutter pixels using no information about what it is you are searching for; this is usually accomplished by separating the local area surrounding a pixel into two distinct regions using a dual-window approach. One way of designing an anomaly detector is by projecting the input test pixel spectrum onto a subspace whose bases are defined by some projection vectors which can be generated in some way using data from the dual window regions. In [3] researchers compared linear subspace-based anomaly detection algorithms using projection vectors which were generated using three common subspace projection techniques - Principal Component Analysis (PCA), Fisher Linear Discriminant (FLD) Analysis, and the Eigenspace Separation Transform (EST). In addition, the performance of the benchmark Reed-Xiaoli (RX) anomaly detector is also compared with the performances of the three proposed linear methods.

In many situations, however, a linear classifier is not always sufficient; that is, most real-world data are not linearly separable. By using a nonlinear mapping, the data are transformed into a higher- (possibly infinite-) dimensional feature space where separating the data using a linear hyperplane is now often possible. This linear hyperplane in the feature space corresponds to an exact nonlinear boundary in the original input space. Similarly, the nonlinear boundary theoretically enhances detection and classification results.

Unfortunately, it is computationally infeasible to carry out any algorithms in this high-dimensional space. However, using kernelization techniques - a concept which comes from machine learning - this problem can be circumvented. To kernelize an algorithm, all dot products between mapped vectors in the feature

space are instead computed using a predetermined kernel function on the input data which correspond to the mapped data. This significantly simplifies the mathematical computation. This thesis examines the performance of the kernel versions of each of the four methods described above and applies each one to the subspace-based anomaly detection problem. More specifically, Kernel Principal Component Analysis (KPCA), Kernel Fisher Discriminant (KFD), and Kernel Eigenspace Separation Transform (KEST), are all used to generate the projection vectors in the feature space. In addition to these three subspace-based methods, the Kernel-RX algorithm is also implemented and the performances of each of these four algorithms are compared against each other as well as against their linear counterparts.

With so many different variations of hyperspectral detection algorithms available in the literature, it is often a difficult challenge to determine which method is superior. Even more, the most optimal method could depend on the hyperspectral image under consideration; therefore, it may be a nearly impossible task to classify any single algorithm as 'globally optimal'. This thesis examines multiple variations (both linear and nonlinear) of one specific type of detection algorithm - a subspace-based anomaly detection algorithm - and characterizes the detection performance and average relative computational time of each method. Performance comparisons are based on receiver operating curve (ROC) and area under the curve (AUC) analysis.

This thesis is designed to provide a performance comparison of linear and nonlinear subspace-based anomaly detection algorithms and is structured in the following manner. Chapter 2 provides a background discussion on hyperspectral imaging

including the theory and applicability of this type of remote sensing. In Chapter 3, a more detailed explanation of target detection applications as well as an introduction to subspace-based anomaly detection can be found. Brief descriptions of the three linear methods that are used to generate the projection vectors are found in Chapter 4. Chapter 5 contains an introduction to kernel-based learning techniques while Chapter 6 shows how each of the three linear methods from Chapter 4 can be extended into their respective nonlinear kernelized methods. Chapter 7 provides a detailed discussion of the well-known RX algorithm as well as its nonlinear kernel version. In Chapter 8, some of the issues involved with kernel methods are discussed. Results and analysis of all eight methods as applied to simulated data as well as multiple hyperspectral data sets can be found in Chapter 9. The hyperspectral images come from the Hyperspectral Digital Imagery Collection Experiment (HYDICE) and the Airborne Hyperspectral Imager (AHI) data sets. Finally, concluding remarks are made in Chapter 10.

Chapter 2

An Introduction to Hyperspectral Imaging

This chapter provides a very brief introduction into hyperspectral imaging technology and theory. It is by no means intended to be a complete discussion as the in depth theory on the subject lies outside the scope of this thesis. For more information on this topic, please see one of the many references cited in this chapter.

2.1 Hyperspectral Sensors

A hyperspectral sensor is a specific type of electro-optical remote sensing system; it gathers spectral information about specific objects without actually physically contacting them. Typically mounted on an aerial craft (i.e. an airplane or a satellite), hyperspectral sensors collect radiance data from a spatial area on the surface of the earth below. There are four main components of a hyperspectral remote sensing process: the illumination source (often the sun), the path which the energy takes through the atmosphere, the ground scene, and the sensor itself [8]. An example of the interaction of all parts of a hyperspectral remote sensing system can be found in Figure 2.1. All four components interact in a very specific way to allow this technology to provide detailed remotely sensed spectral data. A detailed description of the overall sensor process is explained very well in [8]. The sun's energy as a function of wavelength over the electromagnetic spectrum is referred

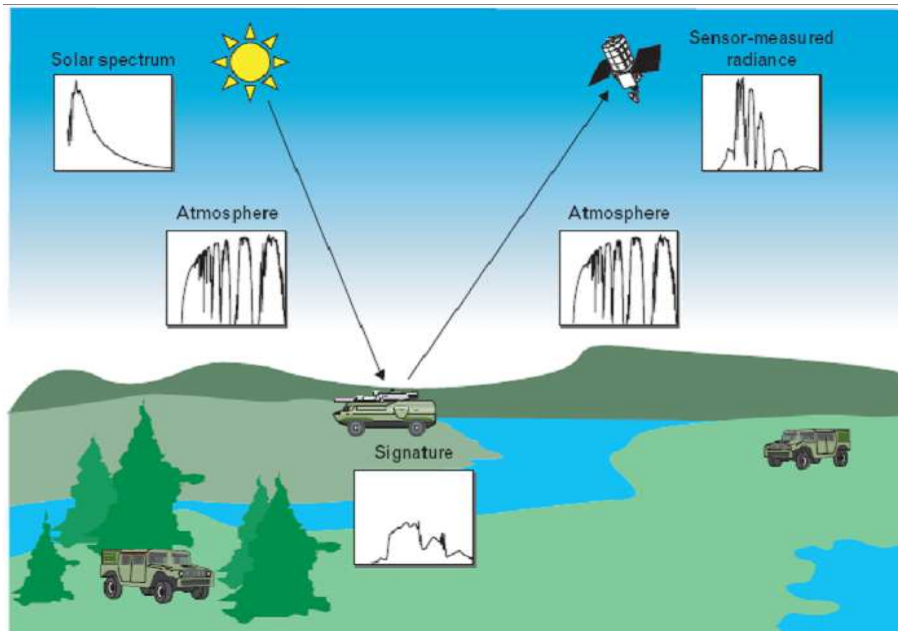


Figure 2.1: This figure illustrates the process involved in a hyperspectral remote sensing system. The spectrum eventually collected by the sensor is a combination of the solar spectrum, atmospheric attenuation, and the material spectral signature. [8]

to as the solar spectrum. As this energy travels from the sun to the surface of the earth, the spectral distribution of intensity is altered by certain atmospheric conditions. When the solar energy comes in contact with materials on the ground, it goes through another spectral transformation, a change which depends on how much that particular material reflects, transmits, or absorbs energy at various electromagnetic wavelengths. After propagating back up through the atmosphere and going through more intensity alterations, the sensor collects the data and stores it for later processing.

At each pixel in the ground scene, the sensor measures spectral radiance in numerous closely-spaced frequency bands. For example, a typical sensor might

collect radiance data in each one of a series of 10-nm wide spectral bands over a spectral range of 0.4 μm through 2.5 μm for a total of 210 frequency bands. This corresponds to the spectral region encompassing the visible to near infrared (VNIR) (0.4 μm - 1 μm) and the short wave infrared (SWIR) (1 μm - 2.5 μm) frequencies. Other sensors are defined in the long wave infrared (LWIR) region (7 μm - 15 μm). The exact spectral range varies with each sensor.

There are many factors which affect the quality of data acquired by the sensors - the most significant of which is solar energy. The sensors are in fact measuring solar reflectance off of objects on the ground. Without enough solar energy, most hyperspectral sensors will not work. LWIR sensors, which for the most part corresponds to the thermal infrared region, avoid this problem because in this frequency range, materials do not reflect as much solar radiation as they emit. Other things that affect data quality and accuracy are air temperature, the presence of water vapor or other mixed gases in the atmospheric, the solar angle, and shadowing. All of these things can cause the collected data to be unusable for processing and detection purposes because of low signal-to-noise ratios.

2.2 Hyperspectral Imagery

The data collected from the sensor are presented in the form of a hyperspectral cube. Each cube has three dimensions, the first two being spatial dimensions while the third is the spectral dimension. This third dimension spans the entire frequency range used by the sensor. The depth of the cube is equal to the number of spectral

bands being used in the sensor. The cube can be viewed in two distinct, yet equally informative ways. The first is as a set of spectra, one at each pixel location. Each spectrum contains the radiance (or associated reflectance) data as a function of wavelength; in addition, each spectrum has a length equal to the number of spectral bands. Every pixel contains the spectral signature data collected from a physical area on the ground scene below; this area is known as the pixel surface. More specifically, each reflectance spectrum is the average of all of the spectra present within a pixel surface region. (This concept will be further explained shortly). The second way to view the cube is as a sequence of radiance (or reflectance) images, one at each spectral band. This viewpoint is analogous to how video data is presented - as a sequence of images, one at each time sample. The first method shows spectral variance at a given spatial location while the second method displays the distribution of radiance (or reflectance) in the spatial region being measured at a given spectral value. These concepts are illustrated in Figure 2.2. While all HSI applications attempt to exploit either the spectral information or the spatial information provided by the sensor, a large number of applications exploit both simultaneously.

2.3 Hyperspectral Theory

The underlying principle of hyperspectral imaging lies in the fact that any given material will emit a certain amount of energy - this amount varies with, and is a function of, wavelength [8]. In particular, "all materials reflect, absorb, and emit electromagnetic energy at specific wavelengths in distinctive patterns related

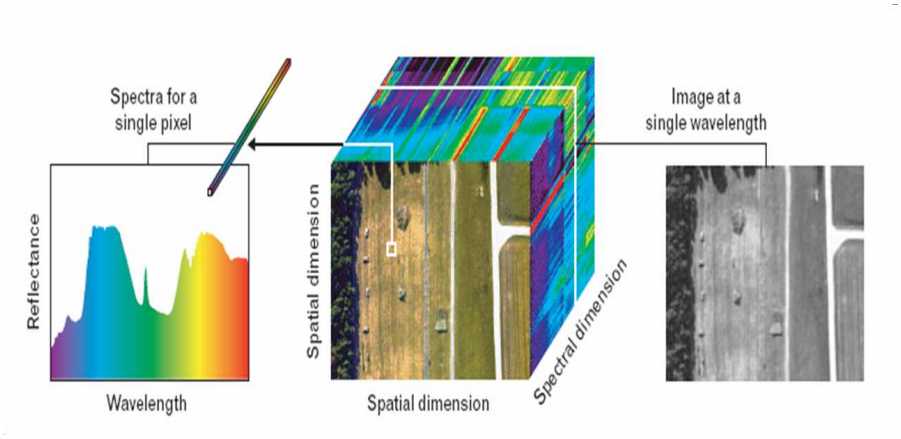


Figure 2.2: This figure shows the basic layout of a hyperspectral cube (center). The cube can be viewed in two ways. The first (left) is as a set of material spectra, one at each pixel location. The second (right) is as a spatial distribution of radiance (reflectance) data at one specific wavelength. Both ways are equally informative. [8]

to their molecular composition" [6]. The goal of some detection algorithms is to be able to identify a material in a given scene based on the reflectance spectrum it emits. Since every pure material emits a unique spectrum, this task is theoretically feasible. One can simply match the measured spectrum with those in a known database such as in [17] to detect and/or classify all desired materials in an image. However, due to complications such as spectral variability, spatial resolution, and sensor compensation techniques, this concept is only valid theoretically.

There are two very important issues that arise when dealing with HSI data - spectral variability and spatial resolution. Both must be considered when designing a target detection system. These issues prevent hyperspectral remote sensing systems from simply and easily extracting a pixel spectrum and matching it precisely with a spectrum of a known material. As mentioned before, there theoretically exists a fixed deterministic spectrum for any given material. However, in practice, this

is not the case. Due to certain uncontrollable and immeasurable conditions even spectral data taken from the same sample of a given material will not all be exactly the same over the entire surface. This effect becomes even more apparent in remote sensing applications. Variations in purity of the material surface, "atmospheric conditions, sensor noise, material composition, location, surrounding materials, and other factors" are the leading causes of this phenomenon [2]. Figure 2.3, which shows 36 measured spectra of a theoretically homogeneous region of a hyperspectral image, illustrates the spectral variability effect. Even within a very confined spatial region, the spectra are very similar yet do not perfectly match. Thus, "an inherent spectral variability prevents the characterization of homogeneous surface materials by unique spectral signatures" [8].

Another issue which needs to be taken into account is spatial resolution. This dilemma evolves from the differences in pixel area sizes captured by a single pixel in the image in relation to the distribution of materials in the ground scene below. This problem, which is very much dependent upon the actual settings of the sensor as well as the height at which the sensor collects its data, gives rise to two pixel types - pure pixels and mixed pixels. A pure pixel is one whose spectrum is that of only one specific material. On the other hand, a mixed pixel is one whose spectrum is a mixture of, or a combination of, spectra of multiple materials on the ground. In this case, detection becomes much more difficult as spectra of non-target materials interfere with the detection process.

These are two key issues which must be dealt with in order to develop a quality hyperspectral target detection algorithm. In the algorithms described below, it will

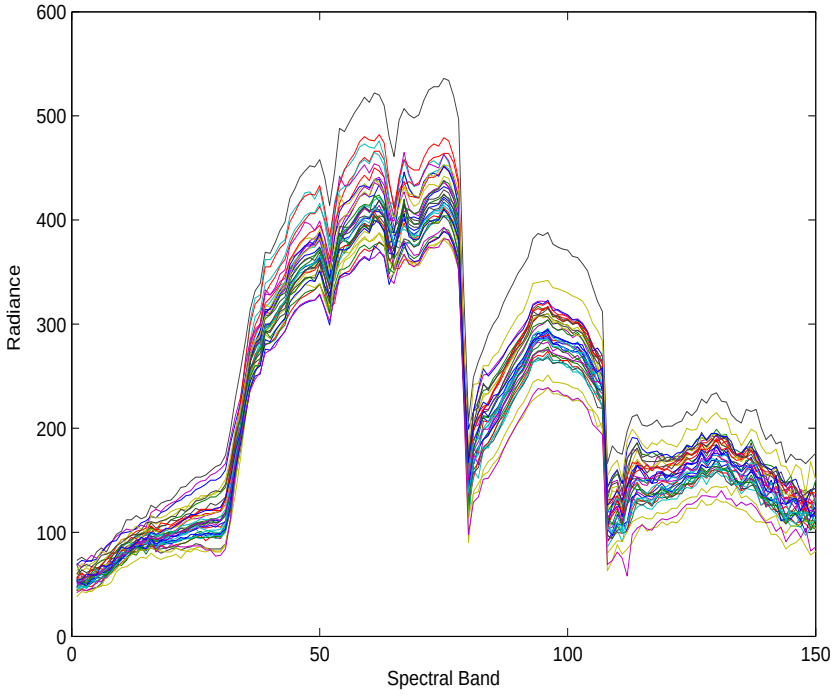


Figure 2.3: This image illustrates the spectral variability that is inherent in hyperspectral imagery. Shown here are 36 contiguous spectral signatures from a hypothetically homogenous background clutter region. All spectral vectors have the same basic shape yet none has an identical spectral signature.

be seen how these issues are taken into consideration.

2.4 HSI Applications

Hyperspectral imaging (HSI) has been used in a variety of civilian and military applications. For instance, HSI can be used for terrain classification as well as for environmental and agricultural monitoring. Geologists are able to use hyperspectral data to determine the composition of the ground at a specific location and search for locations of certain materials without having to do any physical tests on the soil. More commonly, however, HSI is used for military applications such as automatic

target recognition (ATR) [8, 13] and surveillance [18]. The fact that most military targets are man-made objects leads directly to the desirability of performing target detection using hyperspectral imagery. This is due to the fact that most man-made objects are composed of materials that are often significantly different from the surrounding background in which they are placed (usually natural objects). Using spectral information rather than (or in addition to) spatial information to detect objects is often desirable since most military targets are hidden from plain view. With recent significant advances in hyperspectral sensor technology, military applications for hyperspectral imagery continues to grow in number, and with this comes an increasing need to improve target detection and classification algorithms. Equally as important is a need to determine which of the many detection algorithms being proposed are actually capable of quality detection performance. As mentioned above, the goal of this thesis is to provide a comparison between linear and nonlinear anomaly detectors for hyperspectral imagery.

Chapter 3

Anomaly Detection in Hyperspectral Imaging

3.1 Types of Target Detectors

The number of available hyperspectral target detection applications has exploded in the last decade largely due to advances in hyperspectral sensor technology. Sensors developed more recently are able to provide both a higher spatial resolution as well as a higher spectral resolution; both of these are essential for high performance detection algorithms. The overwhelming majority of hyperspectral target detection algorithms that have been studied in recent years can be placed into one of two major categories: spectral match detectors and anomaly detectors.

In the first case, the spectra of 'materials of interest' are assumed to be known *a priori* and can be obtained either from one of many preexisting spectral libraries [17] or can be manually extracted from the hyperspectral data cube itself. This information can then be used in many ways in order to locate the desired targets [7, 13]. Spectral matching algorithms, however, intrinsically pose many problems. Researchers in [7] showed that matched subspace detectors are naturally reliant on how closely the spectra in the hyperspectral image match the spectra for an already known substance in the spectral library [4]. This creates many significant issues. First, spectral libraries are limited in size and may not contain the appropriate material spectra needed for one particular hyperspectral image. In other words,

any given spectral library most likely does not contain the spectra of all materials; moreover, many man-made materials are not pure material but instead are a composite of materials. It is nearly impossible that there exists spectral information for all possible combinations of materials. Secondly, as mentioned in Section 2, the data that are collected by the sensor need to be properly converted from a radiance domain to reflectance. This can be problematic if the sensor is not properly calibrated to compensate for atmospheric conditions or if this information is not accurately available [4].

On the other hand, anomaly detectors do not use or assume any information about the spectra of 'materials of interest'. These types of algorithms are useful when the target spectra are not readily available or are found to be unreliable [2]. These detectors involve finding pixels in the image which are sufficiently distinct (in a statistical sense) from either a globally assumed background or the immediate neighboring pixels. Some anomaly detection algorithms attempt to model the background in some way and locate those pixels which cannot be described well by that model [2, 4, 6]. The background can be modeled either by sampling pixels from the entire image (globally) or by sampling pixels only from a neighboring region immediately surrounding the current test pixel (locally). Generally, when globally sampling background spectra, the detector has difficulty locating "isolated targets in the open if the signature is similar to that of previously classified background material" [2]. Using a local method provides a much more adaptive approach; however, these detectors will sometimes generate a higher false alarm rate (FAR) corresponding to isolated spectral anomalies [2]. For example, a small bush in an otherwise dusty

terrain will most likely be classified as an anomaly because the material spectral signature of the bush will likely be different than that of its immediate surrounding. Nevertheless, most man-made objects have spectra which typically have a lower correlation with natural objects than the correlation between two natural objects [3]. This fact should help to reduce the number of these types of false alarms. One disadvantage of some anomaly detectors is that many of them require the assumption of a distribution for the background clutter. Since background clutter can contain *many* different materials, the assumption of one specific distribution is often problematic and will increase the FAR. A mixture of Gaussian distributions is often used because it greatly simplifies the results [4]. The three linear subspace-based methods used in this thesis require no such assumption of a background model; instead, they use data from the image in order to generate a subspace describing that data. In addition, anomaly detectors will almost certainly yield higher false alarm rates than matched spectral detectors because no knowledge about the desired target spectra is known. Nonetheless, this thesis examines the performance of anomaly detectors using a local background approach.

3.2 The Dual-Window Approach

One very common method used in many anomaly detector algorithms is known as the dual-window approach [3–5, 9]. Using this technique is a way of attempting to exploit both spatial variability in the image as well as spectral variability among different materials. At each pixel location concentric rectangular windows centered

at the test pixel are opened creating two disjoint regions - an inner window region (IWR) and an outer window region (OWR). Hence, the local pixel neighborhood is separated into two smaller regions. The size of the inner window is generally set so that it can fully enclose a target. In most anomaly detectors another concentric rectangle centered at the test pixel known as the 'guard band' is utilized as well. An example of a dual-window with guard band is seen in Figure 3.1. The guard band is slightly larger in size than the IWR yet still smaller than the OWR. The main purpose of the guard band is to reduce the probability that some target spectra will inhabit the OWR and hence affect the background clutter pixels [5]. Use of a guard band becomes very critical in situations when the test pixel lies at the edge of a target region as shown in Figure 3.2. Without the use of a guard band as in Figure 3.2(a), some of the target pixels leak into the OWR. Figure 3.2(b) shows how use of a guard band prevents this from occurring. In the latter case, there are no target pixels in the OWR.

The fundamental basis and applicability of the dual window approach lies in the fact that the statistical properties of the two regions will greatly depend on where in the image the dual window is centered. For example, if both the inner and outer windows lie in a relatively homogeneous area (e.g. a grassy field), the spectra have a high probability of being statistically similar. On the other hand, if the two windows contain materials which are generally different from one another, (e.g. if the window is centered around a target), then the statistical properties of the regions will most likely be dissimilar. The greater this dissimilarity, the greater the projection differences of the two regions onto the corresponding subspace will be, and hence,

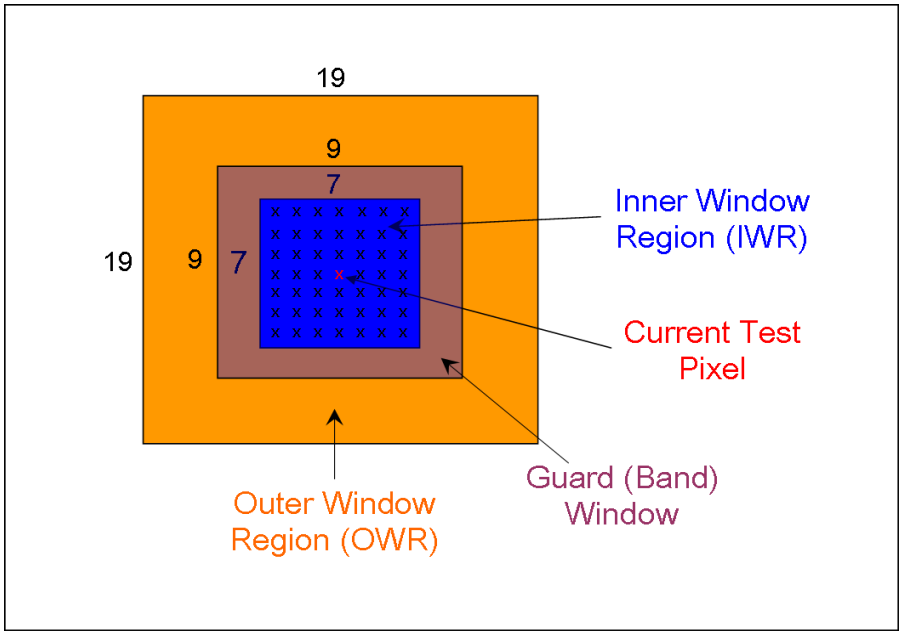
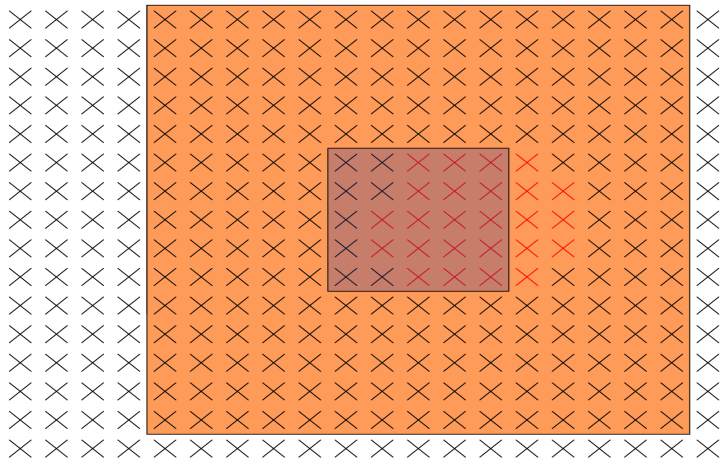
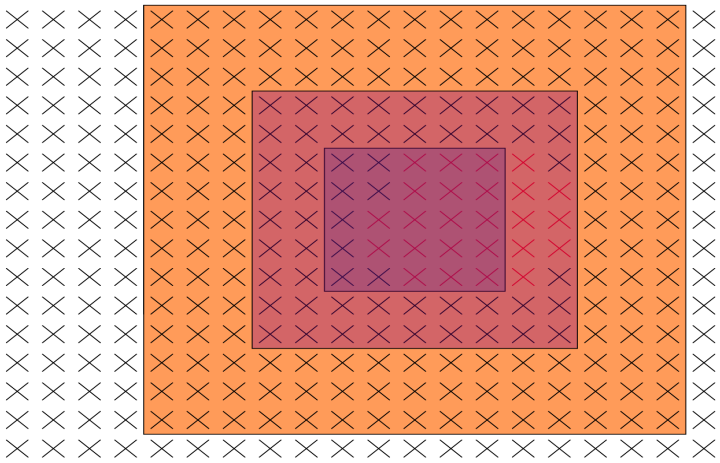


Figure 3.1: An example of a dual window with guard band. The numbers represent the length in pixels making up the side of each window. Each 'x' in the IWR represents one pixel. The pixel in red represents the current test pixel. The figure is not necessarily drawn to scale

the more likely an anomaly will be detected at that particular test pixel. In loose terms, a pixel will be marked as an anomaly if the statistical properties of its own spectral vector and those of the region immediately surrounding it are 'sufficiently' different from the statistical properties of the spectra in the outer window region. The term 'sufficiently' is determined by an appropriate threshold which will often depend on the application.



(a)



(b)

Figure 3.2: This figure illustrates the importance of the use of a dual window with a guard band. Each 'x' represents one pixel. The red ones are the target pixels and the black ones are the background pixels. The dual window is centered at a pixel on the edge of the target. (a) Without a guard band, some of the target pixels leak into the OWR. (b) With a guard band, this problem no longer occurs.

3.3 Subspace-Based Anomaly Detection

The spectra in both the IWR and OWR can be characterized by their respective first and second moments - i.e. their mean vectors and covariance (or correlation) matrices. In this thesis these statistical characteristics are used to generate basis vectors for a subspace onto which test spectra from the IWR and OWR are projected. Projection vectors are generated using both eigen- and non-eigen based discrimination methods. Data with Gaussian distributions are optimally represented in the mean-squared error sense using eigen analysis. Thus, eigen-based methods can usually effectively distinguish differences between spectra based on their statistical properties. However, non-eigen based methods can also be used as long as they are able to exploit second order data statistics [3]. The projection vectors form the bases for a subspace onto which the mean spectra of the OWR and the current test pixel spectrum (TPS) are projected. The TPS will hereby be denoted as $\mathbf{r}$ throughout the remainder of this thesis. Ideally, the projection vectors should be generated in such a way so as to maximize the separation between the respective projections of the TPS and the mean vector of the OWR spectra if these two sets are statistically dissimilar.

In [3], the projection separation statistic is calculated using

$$s = \left| \sum_i \mathbf{w}_i^T (\hat{\boldsymbol{\mu}}_{\mathbf{X}} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}) \right| \tag{3.1}$$

where $\hat{\boldsymbol{\mu}}_{\mathbf{X}}$ and $\hat{\boldsymbol{\mu}}_{\mathbf{Y}}$ represent the means of the IWR and OWR spectra respectively and are defined in Equations (4.3)-(4.4). The vectors $\mathbf{w}_i$ are projection vectors, and i is some variable which is usually much less than the number of spectral bands.

This equation can be adapted in very subtle ways that do not significantly alter the results - only the physical meaning of the metric. The projection separation in this thesis is calculated using

$$s' = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T \mathbf{W}\mathbf{W}^T (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{3.2}$$

where $\mathbf{W} = [\mathbf{w}_1\, \mathbf{w}_2\, \ldots\, \mathbf{w}_m]$ is a matrix whose columns are m projection vectors. Equation (3.2) is referred to as the projection separation statistic (PSS). This is notably equivalent to the norm of Equation (3.1) with one minor exception; the TPS is projected onto the subspace rather than the mean of the IWR spectra ($\hat{\boldsymbol{\mu}}_{\mathbf{X}}$). The product $\mathbf{W}\mathbf{W}^T$ is known as a projection operator and represents a subspace characterizing the spectra that were used to generate the projection vectors $\mathbf{w}_i$. An anomaly is detected if the projection separation, s', is greater than some threshold, η. This threshold can be manually or adaptively chosen depending on the application.

It is also possible to project the difference $(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})$ onto the complement subspace $(\mathbf{I} - \mathbf{W}\mathbf{W}^T)$. This subspace represents all that is not represented by $\mathbf{W}\mathbf{W}^T$. For example, suppose the sample spectra from the OWR of a dual window are used in some way to generate the projection vectors $\mathbf{w}_i$. In addition, assume that the OWR contains background clutter. The subspace $\mathbf{W}\mathbf{W}^T$ in some way represents the background clutter contained within the OWR. The complement subspace $(\mathbf{I} - \mathbf{W}\mathbf{W}^T)$ will possibly include the target subspace, a subspace representing any noise present, as well as any other background clutter not present in the current OWR. Thus, the projection separation can also be calculated using

$$s' = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T (\mathbf{I} - \mathbf{W}\mathbf{W}^T)(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{3.3}$$

Equation (3.3), known as the complement projection separation statistic (CPSS), is only used in some algorithms (e.g. - PCA and EST). In the experimental results section, only the best results between Equation (3.2) and Equation (3.3) are reported and mention will be made regarding which equation was used.

Chapter 4

Linear Methods

This chapter provides an introduction to the three linear methods which are compared in this thesis. Each method is used to generate projection vectors which serve as a basis for a subspace onto which input data are projected. All three methods have been widely used in various pattern classification applications. Here, of course, they are being used in a hyperspectral anomaly detection setting. The exact equation that is used in each case is provided at the end of each subsection. In particular, each method is used to generate the matrix $\mathbf{W}$. Each expression is then plugged back into Equation (3.2) or Equation (3.3) for the final PSS equation for each algorithm.

Notation

To help provide notational consistency throughout this thesis, some terminology and notation is detailed here. First, denote a spectral vector which is contained within the IWR of a dual-window centered at the current test pixel by $\mathbf{x}_k = (x_k(1), x_k(2), \ldots, x_k(J))^T$ where J refers to the number of spectral bands and $k = 1, \ldots, N_{in}$. Assuming that there are a total of N_{in} pixels in the IWR, the matrix $\mathbf{X}$ is of size $J \times N_{in}$ and contains the spectra of each one of these samples as one of its N_{in} columns. Thus,

$$\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_{N_{in}}] . \tag{4.1}$$

Similarly, let a spectral vector which is contained within the OWR of the same dual window be denoted by $\mathbf{y}_l$ where $l = 1, \ldots, N_{out}$. Given that there are N_{out} pixels in the OWR, the $J \times N_{out}$ matrix $\mathbf{Y}$ is one whose columns are the spectral vectors of the pixels in the OWR. So,

$$\mathbf{Y} = \left[\mathbf{y}_1 \ \mathbf{y}_2 \ \cdots \ \mathbf{y}_{N_{out}}\right] \tag{4.2}$$

represents the matrix of local background clutter samples. That is, the background clutter is assumed to contain the spectra of the pixels in the OWR of a dual window centered at the current test pixel.

Furthermore, let

$$\hat{\boldsymbol{\mu}}_X = \frac{1}{N_{in}} \sum_{i=1}^{N_{in}} \mathbf{x}_i \tag{4.3}$$

$$\hat{\boldsymbol{\mu}}_Y = \frac{1}{N_{out}} \sum_{j=1}^{N_{out}} \mathbf{y}_j \tag{4.4}$$

be defined as the statistical means of the IWR and OWR spectra, respectively. The vector $\hat{\boldsymbol{\mu}}_Y$ represents the estimate of the mean of the background clutter. The covariance matrices of the IWR and OWR spectra are given by

$$\mathbf{C}_X = \frac{1}{N_{in} - 1} \left(\mathbf{X} - \hat{\boldsymbol{\mu}}_X\right) \left(\mathbf{X} - \hat{\boldsymbol{\mu}}_X\right)^T \tag{4.5}$$

$$\mathbf{C}_Y = \frac{1}{N_{out} - 1} \left(\mathbf{Y} - \hat{\boldsymbol{\mu}}_Y\right) \left(\mathbf{Y} - \hat{\boldsymbol{\mu}}_Y\right)^T . \tag{4.6}$$

These matrices contain the second order structure of the samples in the IWR and

the OWR, respectively. Finally, as defined in Section 3.3, the spectral vector of the current test pixel is denoted by the vector $\mathbf{r}$.

4.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most commonly used method for feature extraction and dimensionality reduction and can be found in countless pattern classification algorithms. The underlying goal is to find a projection which best represents theinput data in the least-squared sense [19]. In order to generate the projection vectors, $\mathbf{w}_i$, the conventional PCA algorithm found in numerous texts [19, 20] is used. Here, the method is elaborated as it applies to subspace-based hyperspectral anomaly detection.

First, background clutter samples are collected from the OWR; these samples are the columns of the matrix $\mathbf{Y}$ defined in Equation (4.2). Next, the covariance matrix, $\mathbf{C_Y}$, of these sample vectors is calculated using this dataset in Equation (4.6). $\mathbf{C_Y}$ is then eigen-decomposed into its eigenvectors $\mathbf{V}$ and their corresponding eigenvalues $\mathbf{\Lambda}$ as

$$\mathbf{C_Y} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T. \tag{4.7}$$

Equation (4.7) is an equivalent form of the solution to the eigenvalue equation

$$\mathbf{V}\mathbf{\Lambda} = \mathbf{C_Y}\mathbf{V}. \tag{4.8}$$

Next, the projection vectors are taken as the first m eigenvectors with the highest corresponding eigenvalues. Thus,

$$\mathbf{W}_{PCA} = \tilde{\mathbf{V}} = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_m] \tag{4.9}$$

where m is a configurable constant. As explained in Section 8, altering the value of m (i.e. - changing the number of eigenvectors used) could change the performance of the anomaly detector. Using the PCA eigenvectors as the projection vectors allows for the extraction of the principal second-order statistics (components) of the input data.

Using Equation (4.9) as the projection vectors and substituting this result into Equation (3.2) gives

$$\mathbf{PCA}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T \left(\mathbf{W}_{PCA}\mathbf{W}_{PCA}^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{4.10}$$

As mentioned in Section 3.3, it is also possible to substitute Equation (4.9) into Equation (3.3) giving

$$\mathbf{PCA}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T \left(\mathbf{I} - \mathbf{W}_{PCA}\mathbf{W}_{PCA}^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{4.11}$$

The idea behind using the PCA eigenvectors lies in the fact that since these eigenvectors are optimal (i.e. - they minimize the mean-square error) in terms of their representation of the spectral vectors of the OWR, the projection of the difference between the test pixel, $\mathbf{r}$, and the outer window mean, $\hat{\boldsymbol{\mu}}_{\mathbf{Y}}$, will be large if the dual-window is centered on an anomalous target. Pixels which have low second-order correlation - corresponding to a high $\mathbf{PCA(r)}$ value - will likely be labeled as an anomaly.

The algorithm outlined above can also be developed using samples collected form the IWR. In this case, Equations (4.10) and (4.11) remain the same with the exception that the projection vectors, $\mathbf{w}_i$, are generated using the spectral samples contained in the IWR rather than the OWR. In this thesis, Equations (4.10) and

(4.11) are referred to as the 'PCA Algorithm' or simply 'PCA'. In Chapter 9, only the best result among the four possible choices for PCA is given.

4.2 Fisher Linear Discriminant Analysis

Although PCA has proven to be very useful for representing data in a more efficient way, Fisher Linear Discriminant (FLD) Analysis attempts to seek an optimal direction for discriminating between classes. FLD has been one of the most widely used discrimination tools in pattern recognition applications [19]. In this setting, the objective is to discriminate between a target class and a background class.

First, it is assumed that the IWR pixel spectra are one class of data while the OWR pixel spectra are a second class. The between-class scatter matrix is defined as

$$\mathbf{S_B} = (\hat{\boldsymbol{\mu}}_X - \hat{\boldsymbol{\mu}}_Y)(\hat{\boldsymbol{\mu}}_X - \hat{\boldsymbol{\mu}}_Y)^T \tag{4.12}$$

while the within-class scatter matrix can be written as

$$\mathbf{S_W} = \mathbf{C}_X + \mathbf{C}_Y \tag{4.13}$$

where $\mathbf{C}_X$ and $\mathbf{C}_Y$ are the covariance matrices of the samples in the IWR and OWR defined by Equations (4.5) and (4.6), respectively. The matrix $\mathbf{S_B}$ is a measure of how well the means of the two classes are separated while the matrix $\mathbf{S_W}$ is a measure of the compactness of each class cluster [3].

In order to calculate the optimal discrimination direction, $\mathbf{w}^*$, the criterion function needs to be maximized over all possible $\mathbf{w}$ [19]. The criterion function is

given by

$$\mathbf{w}^* = \max_{\mathbf{w}} \mathrm{J}(\mathbf{w}) = \frac{|\mathbf{w}^T \mathbf{S_B} \mathbf{w}|}{|\mathbf{w}^T \mathbf{S_W} \mathbf{w}|}. \tag{4.14}$$

Solving this is accomplished by solving the eigenvalue problem

$$\mathbf{S_B} \mathbf{w} = \lambda \mathbf{S_W} \mathbf{w}. \tag{4.15}$$

and taking the most significant (leading) eigenvector. Further analysis shown in [19, 21] leads to the solution

$$\mathbf{w_F} = \mathbf{w}^* = \mathbf{S_W^{-1}} (\hat{\boldsymbol{\mu}}_\mathbf{X} - \hat{\boldsymbol{\mu}}_\mathbf{Y}) \tag{4.16}$$

Using Equation (4.16) as a single projection vector, $\mathbf{W}$, and substituting this result into Equation (3.2) gives

$$\mathbf{FLD}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)^T \left(\mathbf{w}_F \mathbf{w}_F^T \right) (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y). \tag{4.17}$$

The idea behind using FLD is that it will produce a large projection separation if the spectral means of the IWR and OWR are sufficiently dissimilar while the spectral vectors in each region are tightly clustered. In this thesis, Equation (4.17) is referred to as the 'FLD Algorithm' or simply 'FLD'.

4.3 Eigenspace Separation Transform

The Eigenspace Separation Transform (EST) was developed by Torrieri as a preprocessing technique to extract features for neural network classifiers [22] and has been used successfully by researchers for automatic clutter rejection [23]. Like PCA, EST aims to extract features from a training set by projecting the input patterns

onto a lower-dimensional orthogonal subspace. In this thesis it is used to generate projection vectors in order to separate target pixels from background clutter.

The EST algorithm is laid out as follows:

1. Compute the $J \times J$ Difference Correlation (DCOR) matrix

$$\begin{aligned} \hat{\mathbf{M}} &= \mathbf{R}_X - \mathbf{R}_Y \\ &= \frac{1}{N_{in}}\mathbf{X}\mathbf{X}^T - \frac{1}{N_{out}}\mathbf{Y}\mathbf{Y}^T \end{aligned} \tag{4.18}$$

   where $\hat{\mathbf{M}}$ is simply the difference between the correlation matrices of the IWR ($\mathbf{R}_X$) and OWR ($\mathbf{R}_Y$) and represents the second order statistic differences between the two regions [3] .

2. Calculate the eigenvalues of $\hat{\mathbf{M}}$. Since this matrix is symmetric yet not necessarily non-negative definite, there will be at least one negative eigenvalue.

3. Calculate E_p and E_n by

$$E_p = \sum_{i=1}^{N} \lambda_i \qquad \text{if } \lambda_i \geq 0 \tag{4.19}$$

$$E_n = \sum_{i=1}^{N} |\lambda_i| \qquad \text{if } \lambda_i < 0 \tag{4.20}$$

   These represent the sums of the absolute values of the positive and negative eigenvalues, respectively.

   a. If $E_p > E_n$ then use some number, m, of eigenvectors of $\hat{\mathbf{M}}$ associated with the m largest positive eigenvalues to form the N x m matrix $\mathbf{S}$.

   b. If $E_n > E_p$ then use some number, m, of eigenvectors of $\hat{\mathbf{M}}$ associated with the m largest negative eigenvalues to form the N x m matrix $\mathbf{S}$.

c. If $E_n = E_p$ then use some number, m, of either set of eigenvectors of $\hat{\mathbf{M}}$ associated with the largest m positive or negative eigenvalues to form the N x m matrix $\mathbf{S}$, with a bias towards the smaller set.

The matrix $\mathbf{S}$ then becomes the set of EST projection vectors

$$\mathbf{W}_{\text{EST}} = \mathbf{S} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_m] \tag{4.21}$$

where $\mathbf{v}_i$ $(i = 1, \ldots, m)$ are the m most significant (either positive or negative) eigenvectors or $\hat{\mathbf{M}}$. The effects of the value of m on the performance of EST is shown in Section 8.

Using Equation (4.21) as the projection vectors and substituting this result into Equation (3.2) gives

$$\mathbf{EST}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)^T \left(\mathbf{W}_{EST}\mathbf{W}_{EST}^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_Y). \tag{4.22}$$

As in Section 4.1 for PCA, it is also possible to project onto the complement subspace, $\left(\mathbf{I} - \mathbf{W}_{EST}\mathbf{W}_{EST}^T\right)$. Thus, substituting Equation (4.21) into Equation (3.3) yields

$$\mathbf{EST}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_\mathbf{Y})^T \left(\mathbf{I} - \mathbf{W}_{EST}\mathbf{W}_{EST}^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_\mathbf{Y}). \tag{4.23}$$

Since it is possible to use either the positive eigenvectors or the negative eigenvectors, there are four possible equations that can possibly be used. In this thesis, Equations (4.22) and (4.23) are referred to as the 'EST Algorithm' or simply 'EST'. In the experimental results in Chapter 9, only the best results among the four possible choices of EST are shown. Once again, mention is made as to which of the algorithms is used.

Chapter 5

Kernel Learning Theory

This chapter explains the theoretical foundation on which we are able to extend the linear methods to their corresponding nonlinear (kernelized) versions by using an arbitrary nonlinear mapping associated with a special class of kernel function k. This kernel function can be informally thought of as a similarity measure. Using such nonlinear mappings and certain kernel functions allows for the implicit transformation of the data into a high- (possibly infinite-) dimensional kernel feature space without actually ever having to map the data into that space.

5.1 An Introduction to Kernel Learning Theory

The main concept of kernel-based theory is illustrated by the 2-D simulated data set in Figure 5.1. Readers familiar with Support Vector Machine theory will find very similar concepts here. In the figure on the left, it is clear that there exists no linear boundary that will perfectly discriminate the two classes in the input space. By using a nonlinear mapping Φ, the data is projected into a higher-dimensional feature space $\mathcal{F}$. The mapped data in $\mathcal{F}$ is shown in the figure on the right. It is clear that the data has been positioned in $\mathcal{F}$ in such a way that a linear boundary can perfectly discriminate the two data sets. To be precise, the figure on the right is *not* a two-dimensional space, though it appears that way on paper.

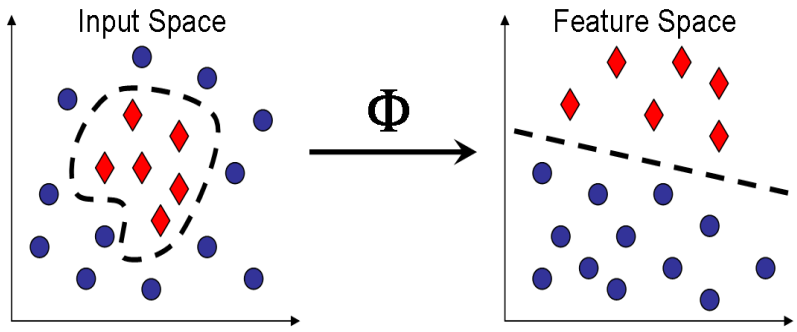


Figure 5.1: This figure displays the basic idea behind the use of kernels. In the input space, only a highly nonlinear decision boundary can be found to separate the data. However, if the data is mapped into a higher-dimensional feature space using Φ, a linear separating hyperplane can (often) be found. By using kernel functions, it is possible to find this separating hyperplane without actually mapping the data into the feature space.

Furthermore, the dashed line that separates the data is in fact meant to be a linear separating *hyperplane* with dimension one less than the dimension of $\mathcal{F}$. This linear separating hyperplane corresponds to an exact nonlinear discriminating boundary in the original input space.

Suppose the input data set lies in the data space ($\chi \in \Re^J$) and let $\mathcal{F}$ be a feature space (also known as a Hilbert space) associated with χ by some nonlinear mapping function Φ. In particular,

$$\Phi : \chi \rightarrow \mathcal{F}$$

$$\mathbf{x} \mapsto \Phi(\mathbf{x}). \tag{5.1}$$

where $\mathbf{x}$ is an input vector ($\mathbf{x} \in \chi$) which is mapped into a much higher dimensional feature space. Mapping the data using Φ into $\mathcal{F}$ is useful in many ways. The most significant benefit is that it is possible to define a similarity measure using the dot

product in $\mathcal{F}$ in terms of a function of the corresponding data in the input space. Thus, it is possible to write

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) &= \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \\ &= \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \end{aligned} \tag{5.2}$$

Equation (5.2) is commonly referred to in machine learning literature as the *kernel trick* [24] and was first used in [25]. The nonlinear mapping Φ could potentially generate a feature space in which it is not computationally feasible to directly implement any algorithm (or realistically perform any computations at all); fortunately, the kernel trick helps circumvent this problem.

Equation (5.2) states that all dot products in $\mathcal{F}$ (a task which is otherwise computationally infeasible) can be implicitly computed by simply using kernel functions defined on the input data. Moreover, all of this can be accomplished without actually mapping the input vectors into $\mathcal{F}$. Hence, conveniently, the mapping Φ does not even need to be identified nor defined. In other words, Equation (5.2) illustrates that all dot products in $\mathcal{F}$ can be replaced by an appropriately chosen kernel function k.

It is then reasonable to ask what types of kernel functions can be deemed 'appropriate'. The answer to this question is that $k(\cdot, \cdot)$ must be continuous, symmetric, and positive definite. That is, the following conditions must be satisfied:

1. $k(\mathbf{x}_i, \mathbf{x}_j)$ must be a continuous function

2. $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$

3. $k(\mathbf{x}_i, \mathbf{x}_j) > 0$ for all x_i, x_j.

Any function that satisfies the above three conditions is known as a *Mercer Kernel* [24]. Equivalently, Mercer's Theorem states that if $k(\cdot, \cdot)$ is a Mercer kernel, then there exists a feature (Hilbert) space $\mathcal{F}$ and an associated nonlinear mapping Φ such that Equation (5.2) holds true [26].

For a more comprehensive discussion about the properties of various types of kernels and for more information on kernel-based learning in general, see [24, 27].

5.2 Choosing the Kernel Function

Table 5.1 shows a few of the many types of kernel functions which are useful in kernel-based learning algorithms. As mentioned above, in order to be able to use the kernel trick, one must choose a Mercer kernel. It is common practice in hyperspectral imaging applications to choose the Gaussian radial basis function (RBF) kernel, which is indeed a Mercer kernel. This kernel is often chosen for mathematical convenience as well as to match the Gaussianity assumption explained in Chapter 3. Another reason why the RBF kernel is useful is due to its *translation-invariance* property [9]. This means that the value of the kernel at any two points $\mathbf{x}$ and $\mathbf{y}$ only depends on the relative difference between the two spectra; it does not depend on the absolute position of either vector in the input space. Since we are comparing spectral differences between a test pixel and its local background, this kernel property is highly desirable. The RBF kernel parameter $\sigma > 0$ is the standard deviation defining the width of the Gaussian distribution. Naturally, the

Kernel	**Function** $k(\mathbf{x}, \mathbf{y})$	**Kernel Parameters**
Gaussian (RBF)	$e^{\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$	$\sigma > 0$
Polynomial	$\langle \mathbf{x}, \mathbf{y} \rangle^d$	$d \in \mathbb{N}$
Sigmoid	$\tanh(\kappa \langle \mathbf{x}, \mathbf{y} \rangle + \gamma)$	$\kappa > 0, \gamma < 0$
Inhomogeneous Polynomial	$(\langle \mathbf{x}, \mathbf{y} \rangle + c)^d$	$d \in \mathbb{N}, c \geq 0$

Table 5.1: Examples of common kernel functions. For this thesis, the Gaussian RBF kernel is chosen.

choice of σ affects the performance of any algorithm using the RBF kernel. This parameter should be chosen so that the kernel can fully exploit the variations among the data [9]. This issue is explored in greater detail in Chapter 8.

Chapter 6

Kernel Method Extensions

In this chapter, each of the three methods in Sections 4.1-4.3 is extended into the feature space $\mathcal{F}$ and then kernelized. As described in Chapter 5, this involves using a nonlinear mapping function $\mathbf{\Phi}$ as defined in Equation (5.1). Since the dimensionality of $\mathcal{F}$ is very high (possibly infinite), it is impractical to actually implement any algorithm in that space. In order to circumvent this problem, dot products in the feature space are replaced by kernel functions using the kernel trick (Equation 5.2). The following three sections present a derivation of each of the kernelized algorithms.

Notation

As was done at the beginning of Chapter 4, some terminology associated with the nonlinear kernel methods is defined here to help provide notational consistency throughout this thesis. First, using a nonlinear mapping $\mathbf{\Phi}$ all of the data in the input space is mapped into $\mathcal{F}$. The original data sets in the input space defined by $\mathbf{X}$ and $\mathbf{Y}$ are mapped into the feature space and denoted by

$$\mathbf{X}_\Phi = \Phi(\mathbf{X}) = [\Phi(\mathbf{x}_1)\, \Phi(\mathbf{x}_2)\, \ldots\, \Phi(\mathbf{x}_{N_{in}})] \tag{6.1}$$

$$\mathbf{Y}_\Phi = \Phi(\mathbf{Y}) = \left[\Phi(\mathbf{y}_1)\, \Phi(\mathbf{y}_2)\, \ldots\, \Phi(\mathbf{y}_{N_{out}})\right] \tag{6.2}$$

This means that $\mathbf{X}_\Phi$ represent the mapped IWR spectra and $\mathbf{Y}_\Phi$ represents the mapped OWR spectra. The statistical means of the mapped data in $\mathcal{F}$ are given by

$$\hat{\boldsymbol{\mu}}_{X_\Phi} = \frac{1}{N_{in}} \sum_{i=1}^{N_{in}} \Phi(\mathbf{x}_i) \tag{6.3}$$

$$\hat{\boldsymbol{\mu}}_{Y_\Phi} = \frac{1}{N_{out}} \sum_{j=1}^{N_{out}} \Phi(\mathbf{y}_j). \tag{6.4}$$

Note that due to the nonlinearity of the mapping Φ, $\hat{\boldsymbol{\mu}}_{X_\Phi} \neq \Phi(\hat{\boldsymbol{\mu}}_X)$. In other words, the mapped mean vector is in general not equal to the mean of the mapped vectors in the feature space.

For many of the following methods, it is assumed that the mapped data is centered in $\mathcal{F}$. Thus, denote a centered vector in $\mathcal{F}$ as $\Phi_c(\mathbf{x}_i) = \Phi(\mathbf{x}_i) - \hat{\boldsymbol{\mu}}_{X_\Phi}$, $i = 1, \ldots, N_{in}$. The same holds true for all data in the OWR (i.e. - $\Phi_c(\mathbf{y}_j) = \Phi(\mathbf{y}_j) - \hat{\boldsymbol{\mu}}_{Y_\Phi}$, $j = 1, \ldots, N_{out}$). Then, let

$$\mathbf{X}_{c_\Phi} = [\Phi_c(\mathbf{x}_1)\, \Phi_c(\mathbf{x}_2)\, \ldots\, \Phi_c(\mathbf{x}_{N_{in}})] \tag{6.5}$$

$$\mathbf{Y}_{c_\Phi} = \left[\Phi_c(\mathbf{y}_1)\, \Phi_c(\mathbf{y}_2)\, \ldots\, \Phi_c(\mathbf{y}_{N_{out}})\right] \tag{6.6}$$

where the columns of $\mathbf{X}_{c_\Phi}$ and $\mathbf{Y}_{c_\Phi}$ represent the centered IWR and OWR spectra in the feature space, respectively. The covariance matrices of the centered spectra in the feature space are given by

$$\mathbf{C}_{X_\Phi} = \frac{1}{N_{in}} \mathbf{X}_{c_\Phi} \mathbf{X}_{c_\Phi}^T \tag{6.7}$$

$$\mathbf{C}_{Y_\Phi} = \frac{1}{N_{out}} \mathbf{Y}_{c_\Phi} \mathbf{Y}_{c_\Phi}^T. \tag{6.8}$$

Finally, the mapped test pixel spectra is given by $\Phi(\mathbf{r})$.

Using $\boldsymbol{\Phi}$, Equation (3.2) is projected into $\mathcal{F}$ yielding a nonlinear projection

separation statistic (NPSS) equation of

$$s' = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \mathbf{W}_\Phi \mathbf{W}_\Phi^T (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}) \tag{6.9}$$

where $\mathbf{W}_\Phi = [\mathbf{w}_\Phi^1 \, \mathbf{w}_\Phi^1 \, \ldots \, \mathbf{w}_\Phi^m]$ is a matrix whose columns are the set of m projection vectors in $\mathcal{F}$. As mentioned in Section 3.3, it is also possible in some cases to project onto the complement subspace. Using $\boldsymbol{\Phi}$, Equation (3.3) becomes

$$s' = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \left(\mathbf{I}_\Phi - \mathbf{W}_\Phi \mathbf{W}_\Phi^T\right) (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}) \tag{6.10}$$

Once again, an anomaly is detected if the projection separation, s', is greater than some threshold, η.

The following methods outline a few of the ways in which the projection vectors can be calculated. These methods are simply nonlinear versions of each of the four algorithms detailed in Chapter 4.

6.1 Kernel Principal Component Analysis

In this section, the Principal Component Analysis method is mapped into the feature space $\mathcal{F}$ and then reformulated solely in terms of dot products. The kernel trick is then utilized to help make the problem computationally feasible. As in the linear PCA algorithm, the Kernel Principal Component Algorithm (KPCA) can be formulated using either the IWR or OWR spectra to formulate the KPCA projection vectors in the feature space. For the purposes of the development, the OWR spectra will be used.

6.1.1 PCA in the Feature Space

Let $\hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}$ be the mean of the background samples in the feature space as defined in Equation (6.4) and let $\mathbf{C}_{\mathbf{Y}_\Phi}$ be the background clutter covariance matrix in the feature space as defined in Equation (6.8). The PCA eigenvectors in the feature space are found by solving the mapped version of Equation (4.8); more specifically, the eigenvalues $\boldsymbol{\Lambda}_\Phi$ and the eigenvectors $\mathbf{V}_\Phi$ of $\mathbf{C}_{\mathbf{Y}_\Phi}$ in $\mathcal{F}$ can be obtained by solving

$$\mathbf{V}_\Phi \boldsymbol{\Lambda}_\Phi = \mathbf{C}_{\mathbf{Y}_\Phi} \mathbf{V}_\Phi. \tag{6.11}$$

In fact, $\boldsymbol{\Lambda}_\Phi = diag\,(\lambda_\Phi^1\, \lambda_\Phi^2\, \ldots\, \lambda_\Phi^p)$ and $\mathbf{V}_\Phi = [\mathbf{v}_\Phi^1\, \mathbf{v}_\Phi^2\, \ldots\, \mathbf{v}_\Phi^p]$ contain only the p nonzero eigenvalues and corresponding eigenvectors of $\mathbf{C}_{\mathbf{Y}_\Phi}$. Equation (6.11) represents all solutions of the associated eigen problem. For notational convenience in the discussion to follow, one single solution of this problem is

$$\lambda_\Phi^k \mathbf{v}_\Phi^k = \mathbf{C}_{\mathbf{Y}_\Phi} \mathbf{v}_\Phi^k \tag{6.12}$$

for $k = 1, \ldots, p$.

6.1.2 Kernelization of PCA in the Feature Space

Since Equation (6.11) cannot be solved explicitly due to the extreme high-dimensionality of $\mathcal{F}$, we must find some way to make use of the kernel trick. Using theoretical analysis found in [28], all eigenvector solutions, $\mathbf{v}_\Phi^k$, of Equation (6.11) lie in the span of the vectors in $\mathbf{Y}_{c_\Phi}$. In other words,

$$\begin{aligned} \mathbf{v}_\Phi^k &= \sum_{j=1}^{N_{out}} \alpha_{jk} \Phi_c(\mathbf{y}_j) \\ &= \mathbf{Y}_{c_\Phi} \boldsymbol{\alpha}_k \end{aligned} \tag{6.13}$$

where $\boldsymbol{\alpha}_k = [\alpha_{1k}, \alpha_{2k}, \ldots, \alpha_{N_{out}k}]^T$ are expansion coefficients. At this point, it is not yet clear how to calculate these expansion coefficients. However, continuing with the derivation will provide more insight.

Substituting Equations (6.8) and (6.13) into Equation (6.12) produces

$$\begin{aligned} N_{out}\lambda_\Phi^k \sum_{j=1}^{N_{out}} \alpha_{jk}\Phi_c(\mathbf{y}_j) &= \sum_{i=1}^{N_{out}}\sum_{j=1}^{N_{out}} \alpha_{jk}\Phi_c(\mathbf{y}_i)\Phi_c(\mathbf{y}_i)^T\Phi_c(\mathbf{y}_j) \\ &= \sum_{i=1}^{N_{out}}\sum_{j=1}^{N_{out}} \alpha_{jk}\Phi_c(\mathbf{y}_i)k(\mathbf{y}_i,\mathbf{y}_j) \end{aligned} \tag{6.14}$$

where the second equality utilizes the kernel trick. Pre-multiplying both sides of Equation (6.14) by $\Phi_c(\mathbf{y}_i)^T$ and simplifying yields a new generalized eigen problem of

$$N_{out}\lambda_\Phi^k \mathbf{K}_\mathbf{Y}\boldsymbol{\alpha}_k = \mathbf{K}_\mathbf{Y}^2\boldsymbol{\alpha}_k \tag{6.15}$$

where $\mathbf{K}_\mathbf{Y}$ is the kernel (Gram) matrix whose elements are $(\mathbf{K}_\mathbf{Y})_{ij} = k(\mathbf{y}_i, \mathbf{y}_j)$ with $i, j = 1, \ldots, N_{out}$. After using the centering procedure outlined in Appendix A to achieve the centered Gram matrix $\hat{\mathbf{K}}_\mathbf{Y}$, it is possible to write all solutions of Equation (6.15) as

$$N_{out}\boldsymbol{\Lambda}_\Phi\hat{\mathbf{K}}_\mathbf{Y}\mathbf{A} = \hat{\mathbf{K}}_\mathbf{Y}^2\mathbf{A} \tag{6.16}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2\ \cdots\ \boldsymbol{\alpha}_p]$ is a matrix whose columns are the nonzero eigenvectors of the centered Gram matrix $\hat{\mathbf{K}}_\mathbf{Y}$ and $\boldsymbol{\Lambda}_\Phi$ contains the associated nonzero eigenvalues. Removing the N_{out} term is easily accomplished by absorbing it into $\hat{\mathbf{K}}_\mathbf{Y}$. It is shown in [28] that the solution to the generalized eigenvalue problem in Equation (6.16) can be found by equivalently solving

$$\breve{\boldsymbol{\Lambda}}_\Phi\breve{\mathbf{A}} = \hat{\mathbf{K}}_\mathbf{Y}\breve{\mathbf{A}} \tag{6.17}$$

and setting

$$\boldsymbol{\alpha}_i = \frac{\breve{\boldsymbol{\alpha}}_i}{\sqrt{\breve{\lambda}_i}} \tag{6.18}$$

for $i = 1, \ldots, p$. These $\boldsymbol{\alpha}_i$ are exactly the expansion coefficients referenced in Equation (6.13). In addition, it turns out that the set of eigenvalues for the two problems are equal - i.e. $\boldsymbol{\Lambda}_\Phi = \breve{\boldsymbol{\Lambda}}_\Phi$.

Using Equation (6.13), the truncated set of eigenvectors in the features space can be written as

$$\tilde{\mathbf{V}}_\Phi = \mathbf{Y}_{c_\Phi} \tilde{\mathbf{A}}_{KPCA} \tag{6.19}$$

where $\tilde{\mathbf{A}} = [\boldsymbol{\alpha}_1 \, \boldsymbol{\alpha}_2 \cdots \boldsymbol{\alpha}_m]$ is a matrix containing the m most significant eigenvectors of $\hat{\mathbf{K}}_\mathbf{Y}$ divided by the square root of their respective eigenvalues and m is a configurable constant. The vectors $\tilde{\mathbf{V}}_\Phi$ are now the projection vectors in the feature space and are used as the vectors of $\mathbf{W}_\Phi$. Substituting Equation (6.19) into Equation (6.9) gives

$$\begin{aligned} \mathbf{KPCA}(\mathbf{r}) &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}))^T \left(\tilde{\mathbf{V}}_\Phi \tilde{\mathbf{V}}_\Phi^T \right) (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \\ &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \mathbf{Y}_{c_\Phi} \tilde{\mathbf{A}}_{KPCA} \tilde{\mathbf{A}}_{KPCA}^T \mathbf{Y}_{c_\Phi}^T (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \end{aligned} \tag{6.20}$$

The above equation can be simplified by noting that

$$\begin{aligned} \Phi(\mathbf{r})^T \mathbf{Y}_{c_\Phi} &= \Phi(\mathbf{r})^T \left(\left[\Phi(\mathbf{y}_1), \Phi(\mathbf{y}_2), \ldots, \Phi(\mathbf{y}_{N_{out}}) \right] - \frac{1}{N_{out}} \sum_{i=1}^{N_{out}} \Phi(\mathbf{y}_i) \right) \\ &= \mathbf{k}(\mathbf{Y}, \mathbf{r})^T - \frac{1}{N_{out}} \sum_{i=1}^{N_{out}} k(\mathbf{y}_i, \mathbf{r}) \\ &\triangleq \mathbf{K}_{\mathbf{Yr}} \end{aligned} \tag{6.21}$$

where the entries in $\mathbf{k}(\mathbf{Y},\mathbf{r})^T$ are the kernels $k(\mathbf{y}_i,\mathbf{r})$ with $i = 1,\ldots,N_{out}$ and $(1/N_{out})\sum_{i=1}^{N_{out}} k(\mathbf{y}_i,\mathbf{r})$ is simply the mean value of the vector $\mathbf{k}(\mathbf{Y},\mathbf{r})^T$. Using similar mathematical analysis,

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{Y_\Phi}^T \mathbf{Y}_{c_\Phi} &= \frac{1}{N_{out}}\sum_{i=1}^{N_{out}} \mathbf{K}(\mathbf{y}_i,\mathbf{Y}) - \frac{1}{N_{out}^2}\sum_{i=1}^{N_{out}}\sum_{j=1}^{N_{out}} k(\mathbf{y}_i,\mathbf{y}_j) \\ &\triangleq K_{\hat{\boldsymbol{\mu}}_\mathbf{Y}} \end{aligned} \tag{6.22}$$

where the latter term in the first equation above is simply the mean of the kernel matrix $\hat{\mathbf{K}}_\mathbf{Y}$. In machine learning, $\mathbf{K}_{\mathbf{Yr}}$ and $\mathbf{K}_{\hat{\boldsymbol{\mu}}_Y}$ are commonly referred to as empirical kernel maps [24].

Substituting these results into Equation (6.20) results in

$$\mathbf{KPCA}(\mathbf{r}) = \left(\mathbf{K}_{\mathbf{Yr}}^T - \mathbf{K}_{\mathbf{Y}\hat{\boldsymbol{\mu}}}^T\right)^T \tilde{\mathbf{A}}_{KPCA}\tilde{\mathbf{A}}_{KPCA}^T \left(\mathbf{K}_{\mathbf{Yr}}^T - \mathbf{K}_{\mathbf{Y}\hat{\boldsymbol{\mu}}}^T\right). \tag{6.23}$$

In addition, it is also possible to use the complement subspace. Using Equation (6.13) as the projection vectors in feature space, $\mathbf{W}_\Phi$, and substituting into Equation (6.10) yields

$$\mathbf{KPCA}(\mathbf{r}) = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \left(\mathbf{I}_\Phi - \mathbf{Y}_{c_\Phi}\tilde{\mathbf{A}}_{KPCA}\tilde{\mathbf{A}}_{KPCA}^T\mathbf{Y}_{c_\Phi}^T\right)(\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \tag{6.24}$$

As mentioned above, the KPCA algorithm can be formulated using the IWR spectra rather than the OWR spectra to generate the projection vectors $\mathbf{w}_\Phi^i$ in the feature space. In this case, Equations (6.23) and (6.24) are still used; however, $\tilde{\mathbf{A}}_{KPCA}$ will be the m most significant eigenvectors of the centered Gram matrix $(\hat{\mathbf{K}}_\mathbf{X})_{ij} = k(\mathbf{x}_i,\mathbf{x}_j)$ with $i,j = 1,\ldots,N_{in}$. Equations (6.23) and (6.24) are the equations used for the Kernel Principal Component Analysis algorithm in this thesis

and are referred to as 'KPCA'. In the experimental results in Chapter 9, only the best results out of the four possible KPCA equations are reported for each image.

6.2 Kernel Fisher Discriminant Analysis

In [21, 24, 29] researchers used the kernel trick to compute the Fisher discriminant in the feature space $\mathcal{F}$ and showed that it is a powerful discrimination tool. A description of the extension of FLD into the feature space and subsequent kernelization is shown below.

6.2.1 FLD in the Feature Space

In this section, the FLD algorithm is extended to the feature space whereupon the problem is reformulated in terms of kernel functions. Using the nonlinear mapping $\mathbf{\Phi}$ to map the input data into the feature space $\mathcal{F}$ it is now necessary to maximize the cost function

$$\mathrm{J}(\mathbf{w}_\Phi) = \frac{|\mathbf{w}_\Phi^T \mathbf{S}_\mathbf{B}^\Phi \mathbf{w}_\Phi|}{|\mathbf{w}_\Phi^T \mathbf{S}_\mathbf{W}^\Phi \mathbf{w}_\Phi|}. \tag{6.25}$$

Here, $\mathbf{w} \in \mathcal{F}$ and $\mathbf{S}_\mathbf{W}^\Phi$ and $\mathbf{S}_\mathbf{B}^\Phi$ are the within-class and between-class scatter matrices, respectively, in $\mathcal{F}$ given by

$$\mathbf{S}_\mathbf{W}^\Phi = \mathbf{X}_{c_\Phi} + \mathbf{Y}_{c_\Phi} \tag{6.26}$$

$$\mathbf{S}_\mathbf{B}^\Phi = (\hat{\boldsymbol{\mu}}_{X_\Phi} - \hat{\boldsymbol{\mu}}_{Y_\Phi})(\hat{\boldsymbol{\mu}}_{X_\Phi} - \hat{\boldsymbol{\mu}}_{Y_\Phi})^T. \tag{6.27}$$

The mean vectors in the features space $\hat{\boldsymbol{\mu}}_{X_\Phi}$ and $\hat{\boldsymbol{\mu}}_{Y_\Phi}$ are defined in Equations (6.3) and (6.4) while the covariance matrices $\mathbf{X}_{c_\Phi}$ and $\mathbf{Y}_{c_\Phi}$ are defined in Equations (6.7)

and (6.8).

Finding an optimal $\mathbf{w}_\Phi$ by maximizing Equation (6.25) is not mathematically tractable considering the simple fact that the feature space is of high- (possibly infinite-) dimensionality.

6.2.2 Kernelization of FLD in the Feature Space

Fortunately, we can reformulate this problem in terms of dot products in the feature space and then utilize the kernel trick. Doing so makes the problem at hand much more computationally feasible.

Based on reproducing kernel theory any solution $\mathbf{w}_\Phi$ to Equation (6.25) can be expanded as

$$\begin{aligned} \mathbf{w}_\Phi &= \sum_{i=1}^{N_{TOT}} \alpha_i \Phi(\mathbf{z}_i) \\ &= \mathbf{Z}_\Phi \boldsymbol{\alpha} \end{aligned} \tag{6.28}$$

where $N_{TOT} = N_{in} + N_{out}$ and

$$\begin{aligned} \mathbf{Z}_\Phi &= [\mathbf{z}_1\ \mathbf{z}_2\ \ldots\ \mathbf{z}_{N_{TOT}}] \\ &= \left[\Phi_c(\mathbf{x}_1)\ \ldots\ \Phi_c(\mathbf{x}_{N_{in}})\ \Phi_c(\mathbf{y}_1)\ \ldots\ \Phi(\mathbf{y}_{N_{out}})\right] \end{aligned} \tag{6.29}$$

is a matrix whose columns are the mapped vectors in $\mathcal{F}$ of the corresponding spectra in both the IWR and OWR. Combining the definition of $\hat{\boldsymbol{\mu}}_{\mathbf{X}_\Phi}$ and Equation (6.28)

yields

$$\begin{aligned}
\mathbf{w}_\Phi^T \hat{\boldsymbol{\mu}}_{\mathbf{X}_\Phi} &= \frac{1}{N_{in}} \sum_{j=1}^{N_{TOT}} \sum_{l=1}^{N_{in}} \alpha_j \Phi(\mathbf{x}_j)^T \Phi(\mathbf{x}_l) \\
&= \frac{1}{N_{in}} \sum_{j=1}^{N_{TOT}} \sum_{l=1}^{N_{in}} \alpha_j k(\mathbf{x}_j, \mathbf{x}_l) \\
&= \boldsymbol{\alpha}^T \mathbf{M}_{in} \qquad (6.30)
\end{aligned}$$

where $(\mathbf{M}_{in})_j \triangleq \frac{1}{N_{in}} \sum_{l=1}^{N_{in}} k(\mathbf{x}_j, \mathbf{x}_l)$. Similarly, using the definition of $\hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}$ and Equation (6.28) gives

$$\begin{aligned}
\mathbf{w}_\Phi^T \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi} &= \frac{1}{N_{out}} \sum_{j=1}^{N_{TOT}} \sum_{l=1}^{N_{out}} \alpha_j \Phi(\mathbf{y}_j)^T \Phi(\mathbf{y}_l) \\
&= \frac{1}{N_{out}} \sum_{j=1}^{N_{TOT}} \sum_{l=1}^{N_{out}} \alpha_j k(\mathbf{y}_j, \mathbf{y}_l) \\
&= \boldsymbol{\alpha}^T \mathbf{M}_{out} \qquad (6.31)
\end{aligned}$$

where $(\mathbf{M}_{out})_j \triangleq \frac{1}{N_{out}} \sum_{l=1}^{N_{out}} k(\mathbf{y}_j, \mathbf{y}_l)$. Notice that the second equations in both expansions above are the direct result of using the kernel trick.

By using Equations (6.27), (6.30) and (6.31) the numerator of Equation (6.25) can be written as

$$\mathbf{w}_\Phi^T \mathbf{S}_\mathbf{B}^\Phi \mathbf{w}_\Phi = \boldsymbol{\alpha}^T \mathbf{A} \boldsymbol{\alpha} \qquad (6.32)$$

where $\mathbf{A} = (\mathbf{M}_{in} - \mathbf{M}_{out})(\mathbf{M}_{in} - \mathbf{M}_{out})^T$. Similarly, using a similar argument, the denominator of Equation (6.25) can be rewritten as

$$\mathbf{w}_\Phi^T \mathbf{S}_\mathbf{W}^\Phi \mathbf{w}_\Phi = \boldsymbol{\alpha}^T \mathbf{B} \boldsymbol{\alpha} \qquad (6.33)$$

where $\mathbf{B} = \mathbf{K}_{in}(\mathbf{I} - \mathbf{1}_{N_{in}})\mathbf{K}_{in}^T + \mathbf{K}_{out}(\mathbf{I} - \mathbf{1}_{N_{out}})\mathbf{K}_{out}^T$, $\mathbf{I}$ is the identity matrix, $\mathbf{K}_{in}$ is an $N_{TOT} \times N_{in}$ Gram matrix, $\mathbf{K}_{out}$ are $N_{TOT} \times N_{out}$ Gram matrix, and $\mathbf{1}_{N_{in}}$ and $\mathbf{1}_{N_{out}}$

are matrices with each entry equal to $1/N_{in}$ and $1/N_{out}$, respectively. For example, each element of $\mathbf{K}_{in}$ and $\mathbf{K}_{out}$ is given by

$$\begin{aligned}(\mathbf{K}_{in})_{mn} &= k(\mathbf{x}_n, \mathbf{x}_m) \\ (\mathbf{K}_{out})_{mn} &= k(\mathbf{y}_n, \mathbf{y}_m).\end{aligned}$$

A combination of Equations (6.32) and (6.33) means that $\boldsymbol{\alpha}$ can now be found by maximizing

$$\mathrm{J}(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \mathbf{A} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{B} \boldsymbol{\alpha}}. \tag{6.34}$$

As in the solution to the analogous problem in the input space (Equation (4.14)), Equation (6.34) can be solved simply by finding the leading eigenvector, $\boldsymbol{\alpha}_{KFD}$, of $\mathbf{B}^{-1}\mathbf{A}$. Thus, the Fisher discrimination vector in $\mathcal{F}$ in Equation (6.28) becomes $\mathbf{w}_\Phi = \mathbf{w}_{F_\Phi} = \mathbf{Z}_\Phi \boldsymbol{\alpha}_{KFD}$.

Substituting this result as the projection vector in $\mathbf{W}_\Phi$ into Equation (6.9), gives

$$\begin{aligned}\mathbf{KFD}(\mathbf{r}) &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \left(\mathbf{w}_{F_\Phi} \mathbf{w}_{F_\Phi}^T\right) (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \\ &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \mathbf{Z}_\Phi \boldsymbol{\alpha}_{KFD} \boldsymbol{\alpha}_{KFD}^T \mathbf{Z}_\Phi^T (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \end{aligned} \tag{6.35}$$

To simplify this equation, let

$$\begin{aligned}\Phi(\mathbf{r})^T \mathbf{Z}_\Phi &= \Phi(\mathbf{r})^T \left[\Phi(\mathbf{x}_1)\ \Phi(\mathbf{x}_2)\ \ldots\ \Phi(\mathbf{x}_{N_{TOT}})\right] \\ &= \mathbf{k}\left(\mathbf{Z}, \mathbf{r}\right)^T \\ &= \mathbf{K}_{\mathbf{Zr}}\end{aligned} \tag{6.36}$$

and

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}^T \mathbf{Z}_\Phi &= \frac{1}{N_{out}} \sum_{\mathbf{y} \in OWR} \Phi(\mathbf{y})^T \left[\Phi(\mathbf{z}_1)\, \Phi(\mathbf{z}_2)\, \ldots\, \Phi(\mathbf{z}_{N_{TOT}})\right] \\ &= \frac{1}{N_{out}} \sum_{\mathbf{y} \in OWR} \mathbf{k}\left(\mathbf{Z}, \mathbf{y}\right)^T \\ &= \mathbf{K}_{\mathbf{Z}\boldsymbol{\mu}}. \end{aligned} \tag{6.37}$$

Using Equations (6.36) and (6.37), Equation (6.35) becomes

$$\mathbf{KFD}(\mathbf{r}) = \left(\mathbf{K}_{\mathbf{Zr}}^T - \mathbf{K}_{\mathbf{Z}\boldsymbol{\mu}}^T\right)^T \boldsymbol{\alpha}_{KFD} \boldsymbol{\alpha}_{KFD}^T \left(\mathbf{K}_{\mathbf{Zr}}^T - \mathbf{K}_{\mathbf{Z}\boldsymbol{\mu}}^T\right) \tag{6.38}$$

Equation (6.38) is the equation used for the Kernel Fisher Discriminant algorithm in this thesis.

Numerical Issues

Unfortunately, as explained in [21] this problem is ill-posed as it is formulated. More specifically, numerical issues result in the matrix $\mathbf{B}$ not being positive definite - a potentially serious problem since the algorithm calls for its inversion. There are a few ways to circumvent this issue. One of the most common way is to use a technique known as regularization. In this method, a multiple (γ - known as the regularization constant) of the identity matrix is added to the matrix $\mathbf{B}$. Thus, the matrix

$$\mathbf{B}_\gamma = \mathbf{B} + \gamma \mathbf{I}. \tag{6.39}$$

If γ is chosen large enough, the problem is no longer numerically unstable as $\mathbf{B}_\gamma$ will be positive definite. Another regularization technique is to add a multiple of the

kernel matrix $\mathbf{K_Z}$ where each element $(\mathbf{K_Z})_{ij} = k(\mathbf{z}_i, \mathbf{z}_j)$ where $i, j = 1, \ldots, N_{TOT}$. Then,

$$\mathbf{B}_\gamma = \mathbf{B} + \gamma \mathbf{K_Z}. \tag{6.40}$$

Clearly, the choice of an appropriate regularization constant will be data dependent. In this research, the value of γ was chosen experimentally and the regularization method shown in Equation (6.40) was used as it produced better results.

6.3 Kernel Eigenspace Separation Transform

In this section, the Eigenspace Separation Transform is mapped into the feature space $\mathcal{F}$ and then reformulated solely in terms of dot products. Once again, the kernel trick is utilized to help make the problem computationally feasible.

6.3.1 EST in the Feature Space

Recall that, like Fisher Discriminant Analysis, the Eigenspace Separation Transform is defined for two classes and uses information about both classes to determine a discrimination boundary. The difference correlation matrix (DCOR) $\mathbf{R}_\Phi$ for the input data in the feature space can be written as

$$\begin{aligned}\mathbf{R}_\Phi &= \mathbf{R}_{\mathbf{X}_\Phi} - \mathbf{R}_{\mathbf{Y}_\Phi} = \frac{1}{N_{in}}\Phi(\mathbf{X})\Phi(\mathbf{X})^T - \frac{1}{N_{out}}\Phi(\mathbf{Y})\Phi(\mathbf{Y})^T \\ &= \begin{bmatrix} \Phi(\mathbf{X}) & -\Phi(\mathbf{Y}) \end{bmatrix} \begin{bmatrix} \Phi(\mathbf{X})^T/N_{in} \\ \\ \Phi(\mathbf{Y})^T/N_{out} \end{bmatrix}.\end{aligned} \tag{6.41}$$

Here, the correlation matrix in the feature space of the IWR spectra set is $\mathbf{R}_{\mathbf{X}_\Phi} = \Phi(\mathbf{X})\Phi(\mathbf{X})^T/N_{in}$ and likewise, the correlation matrix in the feature space of the

OWR spectra set is $\mathbf{R}_{\mathbf{Y}_\Phi} = \Phi(\mathbf{Y})\Phi(\mathbf{Y})^T/N_{out}$. The eigen decomposition of DCOR in the feature space can be rewritten in block-matrix form in terms of its positive and negative eigenvalues and eigenvectors as

$$\mathbf{R}_\Phi = \begin{bmatrix} \mathbf{V}_{+_\Phi} & \mathbf{V}_{-_\Phi} \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_{+_\Phi} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_{-_\Phi} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{+_\Phi}^T \\ \mathbf{V}_{-_\Phi}^T \end{bmatrix} \tag{6.42}$$

where the columns of $\mathbf{V}_{+_\Phi}$ and $\mathbf{V}_{-_\Phi}$ are the eigenvectors in the feature space with their corresponding non-zero positive ($\mathbf{\Lambda}_{+_\Phi}$) and negative ($\mathbf{\Lambda}_{-_\Phi}$) eigenvalues, respectively. In order to diagonalize the DCOR matrix $\mathbf{R}_\Phi$ we must find all eigenvectors (both positive and negative) $\mathbf{V}_\Phi$ and all nonzero eigenvalues $\mathbf{\Lambda}_\Phi$ which satisfy the equation

$$\mathbf{V}_\Phi \mathbf{\Lambda}_\Phi = \mathbf{R}_\Phi \mathbf{V}_\Phi. \tag{6.43}$$

6.3.2 Kernelization of EST in the Feature Space

Due to the (possibly) extreme high-dimensionality of the feature space, (6.43) cannot be explicitly solved. In order to circumvent this problem, the equation can be kernelized by writing it in terms of kernel functions. Doing so allows us to implement the equation in the original input domain in terms of kernel functions.

Each eigenvector $\mathbf{v}_\Phi^k$ in the feature space can be written as a linear combination of the centered input data as

$$\begin{aligned} \mathbf{v}_\Phi^k &= \frac{1}{\sqrt{N_{in}}} \sum_{i=1}^{N_{in}} \alpha_i^k \Phi(\mathbf{x}_i) \lambda_i^{-\frac{1}{2}} - \frac{1}{\sqrt{N_{out}}} \sum_{j=1}^{N_{out}} \beta_j^k \Phi(\mathbf{y}_j) \lambda_j^{-\frac{1}{2}} \\ &= \frac{1}{\sqrt{N_{in}}} \Phi(\mathbf{X}) \boldsymbol{\alpha}^k \mathbf{\Lambda}_+^{-\frac{1}{2}} - \frac{1}{\sqrt{N_{out}}} \Phi(\mathbf{Y}) \boldsymbol{\beta}^k \mathbf{\Lambda}_-^{-\frac{1}{2}} \end{aligned} \tag{6.44}$$

where the expansion coefficients, $\boldsymbol{\alpha}^k$ and $\boldsymbol{\beta}^k$, are defined as $\boldsymbol{\alpha}^k = (\alpha_1^k, \alpha_2^k, \ldots, \alpha_{N_{in}}^k)^T$

and $\boldsymbol{\beta}^k = (\beta_1^k, \beta_2^k, \ldots, \beta_{N_{out}}^k)^T$ for $k = 1, \ldots, N_t$ where $N_t = N_{in} + N_{out}$. Equation (6.44) can be used to write all eigenvectors with non-zero eigenvalues as

$$\begin{aligned}
\mathbf{V}_\Phi &= \begin{bmatrix} \mathbf{v}_\Phi^1 & \mathbf{v}_\Phi^2 & \cdots & \mathbf{v}_\Phi^{N_t} \end{bmatrix} \\
&= \Phi(\mathbf{X})\mathbf{A}\boldsymbol{\Lambda}_+^{-\frac{1}{2}} - \Phi(\mathbf{Y})\mathbf{B}\boldsymbol{\Lambda}_-^{-\frac{1}{2}} \\
&= \begin{bmatrix} \Phi(\mathbf{X}) & -\Phi(\mathbf{Y}) \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_+^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_-^{-\frac{1}{2}} \end{bmatrix} \\
&= \begin{bmatrix} \Phi(\mathbf{X}) & -\Phi(\mathbf{Y}) \end{bmatrix} \mathbf{D} \qquad (6.45)
\end{aligned}$$

where

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} \frac{\boldsymbol{\alpha}^1}{\sqrt{N_{in}}} & \frac{\boldsymbol{\alpha}^2}{\sqrt{N_{in}}} & \cdots & \frac{\boldsymbol{\alpha}^{N_t}}{\sqrt{N_{in}}} \\ \frac{\boldsymbol{\beta}^1}{\sqrt{N_{out}}} & \frac{\boldsymbol{\beta}^2}{\sqrt{N_{out}}} & \cdots & \frac{\boldsymbol{\beta}^{N_t}}{\sqrt{N_{out}}} \end{bmatrix}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_+ & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_- \end{bmatrix}$$

and the columns of

$$\begin{aligned}
\mathbf{D} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_+^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_-^{-\frac{1}{2}} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\boldsymbol{\alpha}^1}{\sqrt{N_{in}}} & \frac{\boldsymbol{\alpha}^2}{\sqrt{N_{in}}} & \cdots & \frac{\boldsymbol{\alpha}^{N_t}}{\sqrt{N_{in}}} \\ \frac{\boldsymbol{\beta}^1}{\sqrt{N_{out}}} & \frac{\boldsymbol{\beta}^2}{\sqrt{N_{out}}} & \cdots & \frac{\boldsymbol{\beta}^{N_t}}{\sqrt{N_{out}}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda} \end{bmatrix}^{-\frac{1}{2}} \qquad (6.46)
\end{aligned}$$

represent the eigenvectors of a kernel matrix associated with the kernelized version of EST (shown below). By substituting equations (6.41) and (6.45) into (6.43) and using the kernel trick from Equation (5.2) to simplify we obtain

$$\boldsymbol{\Lambda}\Big[\Phi(\mathbf{X}) \; -\Phi(\mathbf{Y})\Big]\mathbf{D} = \Big[\Phi(\mathbf{X}) \; -\Phi(\mathbf{Y})\Big]\begin{bmatrix} \frac{\mathbf{K_{XX}}}{N_{in}} & -\frac{\mathbf{K_{XY}}}{N_{in}} \\ \frac{\mathbf{K_{YX}}}{N_{out}} & -\frac{\mathbf{K_{YY}}}{N_{out}} \end{bmatrix}\mathbf{D} \qquad (6.47)$$

where $\mathbf{K_{XX}} = \Phi(\mathbf{X})^T\Phi(\mathbf{X})$ is an $N_{in}\times N_{in}$ kernel (Gram) matrix, $\mathbf{K_{YY}} = \Phi(\mathbf{Y})^T\Phi(\mathbf{Y})$ is an $N_{out} \times N_{out}$ kernel matrix, $\mathbf{K_{XY}} = \Phi(\mathbf{X})^T\Phi(\mathbf{Y})$ is an $N_{in} \times N_{out}$ kernel matrix, and $\mathbf{K_{YX}} = \Phi(\mathbf{Y})^T\Phi(\mathbf{X})$ is an $N_{out} \times N_{in}$ kernel matrix. Each of the entries in all four matrices is obtained in terms of the kernel function k.

Multiplying both sides of (6.47) by $[\Phi(\mathbf{X}) \quad \Phi(\mathbf{Y})]^T$ and again using (5.2) to simplify produces

$$\mathbf{\Lambda}\begin{bmatrix} \frac{\mathbf{K_{XX}}}{N_{in}} & -\frac{\mathbf{K_{XY}}}{N_{in}} \\ \frac{\mathbf{K_{YX}}}{N_{out}} & -\frac{\mathbf{K_{YY}}}{N_{out}} \end{bmatrix}\mathbf{D} = \begin{bmatrix} \frac{\mathbf{K_{XX}}}{N_{in}} & -\frac{\mathbf{K_{XY}}}{N_{in}} \\ \frac{\mathbf{K_{YX}}}{N_{out}} & -\frac{\mathbf{K_{YY}}}{N_{out}} \end{bmatrix}^2\mathbf{D}. \tag{6.48}$$

As in Section 6.1, solving equation (6.48) is tantamount to finding the eigenvectors and eigenvalues of the kernel matrix

$$\mathbf{K}_{KEST} = \begin{bmatrix} \frac{\mathbf{K_{XX}}}{N_{in}} & -\frac{\mathbf{K_{XY}}}{N_{in}} \\ \frac{\mathbf{K_{YX}}}{N_{out}} & -\frac{\mathbf{K_{YY}}}{N_{out}} \end{bmatrix} = \tilde{\mathbf{D}}\mathbf{\Lambda}\tilde{\mathbf{D}}^T. \tag{6.49}$$

and normalizing each of the eigenvectors by the square root of its associated eigenvalue. Here, the columns of the matrix $\tilde{\mathbf{D}} = \left[\tilde{\mathbf{d}}_1\,\tilde{\mathbf{d}}_2\,\ldots\,\tilde{\mathbf{d}}_{N_t}\right]$ represent the positive and negative eigenvectors of the KEST kernel matrix, $\mathbf{K}_{KEST}$. Then, $\mathbf{D} = \left[\frac{\tilde{\mathbf{d}}_1}{\sqrt{\lambda_1}}\,\frac{\tilde{\mathbf{d}}_2}{\sqrt{\lambda_2}}\,\ldots\,\frac{\tilde{\mathbf{d}}_{N_t}}{\sqrt{\lambda_{N_t}}}\right]$ where $\mathbf{\Lambda} = diag\left(\lambda_1, \lambda_2, \ldots, \lambda_{N_t}\right)$.Equivalently, they are the expansion coefficients in (6.44). The corresponding positive and negative eigenvalues are contained in the diagonal matrix $\mathbf{\Lambda}$. For simplicity, the eigenvalues and corresponding eigenvectors should be ordered from most positive significant to most negative significant.

Let the KEST projection vectors, $\mathbf{W}_{KEST}$ vectors be either the first m positive

or negative eigenvectors of $\mathbf{D}$. Thus, either

$$\begin{aligned}\mathbf{W}_{KEST} = \mathbf{W}^{+}_{KEST} &= [\mathbf{d}_1\, \mathbf{d}_2\, \cdots\, \mathbf{d}_m] \\ \mathbf{W}_{KEST} = \mathbf{W}^{-}_{KEST} &= [\mathbf{d}_{N_t}\, \mathbf{d}_{N_t-1}\, \cdots\, \mathbf{d}_{N_t-m+1}] \end{aligned} \tag{6.50}$$

where, as with KPCA, m is a configurable constant. More on this topic can be found in Chapter 8. The choice of using most positive significant or most negative significant is a data dependent choice and is determined using the procedure outlined for the linear EST method in Section 4.3.

Substituting Equation (6.50) for $\mathbf{D}$ in Equation (6.45) (i.e. - only using the first m positive or negative eigenvectors) and using this result as the projection vectors, $\mathbf{W}_{\Phi}$, in Equation (6.9) yields

$$\begin{aligned}\mathbf{KEST}(\mathbf{r}) &= \left(\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}\right)^T \left(\mathbf{V}_\Phi \mathbf{V}_\Phi^T\right) \left(\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}\right). \\ &= \left(\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}\right)^T \Phi(\bar{\mathbf{Z}}) \mathbf{W}_{KEST} \mathbf{W}_{KEST}^T \Phi(\bar{\mathbf{Z}})^T \left(\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}\right) \end{aligned} \tag{6.51}$$

where $\Phi(\bar{\mathbf{Z}}) = [\Phi(\mathbf{X}) \ -\Phi(\mathbf{Y})]$. For notational convenience, let

$$\begin{aligned}\Phi(\mathbf{r})^T \Phi(\mathbf{Z}) &= \Phi(\mathbf{r})^T \left[\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_{N_{in}}), -\Phi(\mathbf{y}_1), \ldots, -\Phi(\mathbf{y}_{N_{out}})\right] \\ &= \left[k(\mathbf{x}_1, \mathbf{r}), \ldots, k(\mathbf{x}_{N_{in}}, \mathbf{r}), -k(\mathbf{y}_1, \mathbf{r}), \ldots, -k(\mathbf{y}_{N_{out}}, \mathbf{r})\right] \\ &= \mathbf{k}(\bar{\mathbf{Z}}, \mathbf{r})^T \\ &= \mathbf{K}_{\bar{\mathbf{Z}}\mathbf{r}} \end{aligned} \tag{6.52}$$

where the second equal sign is as a direct result of using the kernel trick in Equation (5.2). The vector $\mathbf{k}(\bar{\mathbf{Z}}, \mathbf{r})^T$ is commonly referred to as the empirical kernel map of

an input vector $\mathbf{r}$ [24]. Similarly, define

$$\begin{aligned}\hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}^T \Phi(\bar{\mathbf{Z}}) &= \frac{1}{N_{out}} \sum_{\mathbf{y} \in OWR} \Phi(\mathbf{y})^T \left[\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_{N_{in}}), -\Phi(\mathbf{y}_1), \ldots, -\Phi(\mathbf{y}_{N_{out}})\right] \\ &= \frac{1}{N_{out}} \sum_{\mathbf{y} \in OWR} \mathbf{k}\left(\bar{\mathbf{Z}}, \mathbf{y}\right)^T \\ &= \mathbf{K}_{\bar{\mathbf{Z}}\hat{\boldsymbol{\mu}}}. \end{aligned} \tag{6.53}$$

Using Equations (6.52) and (6.53), Equation (6.51) becomes

$$\mathbf{KEST}(\mathbf{r}) = \left(\mathbf{K}_{\bar{\mathbf{Z}}\mathbf{r}}^T - \mathbf{K}_{\bar{\mathbf{Z}}\hat{\boldsymbol{\mu}}}^T\right)^T \mathbf{W}_{KEST} \mathbf{W}_{KEST}^T \left(\mathbf{K}_{\bar{\mathbf{Z}}\mathbf{r}}^T - \mathbf{K}_{\bar{\mathbf{Z}}\hat{\boldsymbol{\mu}}}^T\right) \tag{6.54}$$

As in the case of linear EST in Section 4.3, it is also possible to project onto the complement subspace. Again using Equation (6.45) as the projection vectors, $\mathbf{W}_\Phi$, in Equation (6.10) yields

$$\mathbf{KEST}(\mathbf{r}) = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \left(\mathbf{I}_\Phi - \Phi(\bar{\mathbf{Z}}) \mathbf{W}_{KEST} \mathbf{W}_{KEST}^T \Phi(\bar{\mathbf{Z}})^T\right) (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \tag{6.55}$$

Equations (6.54) and (6.55) are the equations used for the Kernel Eigenspace Separation Transform algorithm in this thesis and are referred to as 'KEST'. Of the four possible choices here, only the best results are presented in Chapter 9.

Chapter 7

RX and Kernel-RX Anomaly Detectors

7.1 RX Anomaly Detector

Since its introduction by Reed and Yu, the Reed-Xiaoli (RX)-anomaly detector [30] has become the benchmark for hyperspectral anomaly detection methods. Using the generalized-likelihood ratio test (GLRT) as its detection basis, the algorithm is a constant false-alarm rate (CFAR) anomaly detector. The RX-detector has become so popular in HSI anomaly detection applications because of its natural assumption that neither the target spectrum nor the covariance matrix of the background clutter need to be known [9].

The foundation of the RX-algorithm is a binary hypothesis testing problem with hypotheses $\mathbf{H}_0$ and $\mathbf{H}_1$. The two competing hypotheses are formulated as

$$\begin{aligned} \mathbf{H}_0 : \mathbf{x} &= \mathbf{n} \qquad \text{(No target present)} \\ \mathbf{H}_1 : \mathbf{x} &= a\mathbf{s} + \mathbf{n} \qquad \text{(Target Present)} \end{aligned} \tag{7.1}$$

where a is a scale factor which is equal to zero for $\mathbf{H}_0$ and $a > 0$ for $\mathbf{H}_1$. The vector $\mathbf{n}$ is for the noise process of the background samples while $\mathbf{s} = (s_1, s_2, \ldots, s_J)^T$ is the target spectrum. This algorithm assumes that for hypothesis $\mathbf{H}_0$, the data can be modeled as a zero-mean normal density with covariance matrix $\mathbf{C_Y}$; for hypothesis $\mathbf{H}_1$, the data can be modeled as a normal density with mean $\mathbf{s}$ and covariance matrix

$\mathbf{C_Y}$. Hence,

$$\begin{aligned} \mathbf{H}_0 &\sim \mathcal{N}(0, \mathbf{C_Y}) \\ \mathbf{H}_1 &\sim \mathcal{N}(\mathbf{s}, \mathbf{C_Y}). \end{aligned} \tag{7.2}$$

Both $\mathbf{s}$ and the true structure of the background clutter covariance matrix, $\mathbf{C}_b$, are unknown. Instead, the covariance matrix of the OWR samples, $\hat{\mathbf{C}}_\mathbf{Y}$, is assumed to be an estimate of $\mathbf{C_Y}$. When a specific target spectrum is sought, the assumption that both distributions have the same covariance matrix $\mathbf{C}_Y$ is invalid. However, since the specific statistical form of the anomaly is not known, it is not possible to estimate the covariance matrix in this case (under hypothesis $\mathbf{H}_1$). In order to simplify matters, the aforementioned assumption is made.

In [30], the final form of the RX-algorithm is given as

$$\mathbf{RX}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)^T \left(\frac{N_{out}}{N_{out}+1} \hat{\mathbf{C}}_Y + \frac{1}{N_{out}+1} (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)(\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)^T \right)^{-1} (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y) \underset{H_0}{\overset{H_1}{\gtrless}} \eta \tag{7.3}$$

where η is an alterable detection threshold. As the number of background clutter samples approaches infinity ($N_{out} \to \infty$), the above equation becomes

$$\mathbf{RX}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)^T \hat{\mathbf{C}}_Y^{-1} (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y). \tag{7.4}$$

The RX-algorithm (which compares the difference between the test spectrum and the mean spectrum of the immediate background) is similar to the Mahaloanobis distance measure. In this thesis, Equation (7.4) was used for all implementation purposes as the conventional RX-algorithm.

7.2 Kernel-RX Algorithm

7.2.1 RX in the Feature Space

In this section, the RX algorithm is extended into the feature space. Using similar Gaussianity assumptions as were used in Section 7.1, the two hypotheses in the feature space become

$$\begin{aligned} \mathbf{H}_{0_\Phi} : \Phi(\mathbf{x}) &= \mathbf{n}_\Phi \qquad \text{(No target present)} \\ \mathbf{H}_{1_\Phi} : \Phi(\mathbf{x}) &= b\Phi(\mathbf{s}) + \mathbf{n}_\Phi \qquad \text{(Target Present)} \end{aligned} \tag{7.5}$$

where, again, b is a scale factor which is equal to zero for $\mathbf{H}_{0_\Phi}$ and $b > 0$ for $\mathbf{H}_{1_\Phi}$. Using the GLRT in the feature space allows the RX algorithm in the feature space to be defined as

$$\mathbf{RX}(\Phi(\mathbf{r})) = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{Y_\Phi})^T \hat{\mathbf{C}}_{\mathbf{Y}_\Phi}^{-1} (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{Y_\Phi}) \tag{7.6}$$

where $\hat{\mathbf{C}}_{Y_\Phi}$ is the estimated covariance matrix of the background clutter in the feature space and $\hat{\boldsymbol{\mu}}_{Y_\Phi}$ is the mean of the background samples in the feature space. Again, similar to the linear case, two Gaussian distributions are assumed with different means but equal covariance structures. In the feature space Equation (7.6) corresponds to a linear detector; however, it corresponds to a nonlinear detector in the original input space.

7.2.2 Kernelization of the RX algorithm in the Feature Space

Unfortunately, Equation (7.6) cannot be directly implemented because of the high dimensionality of the feature space $\mathcal{F}$ into which Φ maps the data. Once again,

circumventing this problem is easily accomplished using kernel trick to kernelize the equation.

The background covariance estimate matrix, $\hat{\mathbf{C}}_{Y_\Phi}$, can be decomposed as

$$\hat{\mathbf{C}}_{Y_\Phi} = \mathbf{V}_\Phi \mathbf{\Lambda}_Y \mathbf{V}_\Phi^T \tag{7.7}$$

where $\mathbf{\Lambda}_Y$ is a diagonal matrix who diagonal elements are the eigenvalues of $\hat{\mathbf{C}}_{Y_\Phi}$. The matrix $\mathbf{V}_\Phi$ contains the corresponding eigenvectors of $\hat{\mathbf{C}}_{Y_\Phi}$ in the feature space. That is,

$$\mathbf{V}_\Phi = \left[\mathbf{v}_\Phi^1, \mathbf{v}_\Phi^2, \ldots, \mathbf{v}_\Phi^p\right] \tag{7.8}$$

where p corresponds to the number of eigenvectors whose eigenvalues are nonzero. The decomposition in Equation (7.7) is also known as a spectral decomposition [20].

Inverting a matrix whose eigenvalues contain values which are very close to zero (or are equal to zero) can potentially result in numerically unstable results. For this reason, the pseudo-inverse is used instead. The pseudoinverse of $\hat{\mathbf{C}}_{Y_\Phi}$ can be expressed as

$$\hat{\mathbf{C}}_{Y_\Phi}^{\#} = \mathbf{V}_\Phi \mathbf{\Lambda}_Y^{-1} \mathbf{V}_\Phi^T. \tag{7.9}$$

Furthermore, the nonzero eigenvectors of the centered Gram matrix $\left(\mathbf{K}_Y \triangleq \hat{\mathbf{K}}(\mathbf{Y}, \mathbf{Y})\right)$ normalized by the square root of the corresponding eigenvalues are denoted as

$$\boldsymbol{\beta}^j = \left(\beta_1^j, \beta_2^j, \ldots, \beta_p^j\right)^T \qquad\qquad j = 1, \ldots, N_{out} \tag{7.10}$$

Analysis in [9] shows that the feature space eigenvectors $\mathbf{v}_\Phi^j$ can also be written as

$$\mathbf{v}_\Phi^j = \sum_{i=1}^{N_{out}} \beta_i^j \Phi_c(\mathbf{y}(i)) \lambda_j^{-\frac{1}{2}}. = \mathbf{Y}_{c_\Phi} \boldsymbol{\beta}^j \lambda_j^{-\frac{1}{2}}. \tag{7.11}$$

where λ_j is the eigenvalue corresponding to the j^{th} eigenvector. That is, each eigenvector in the features space can be expressed in terms of the centered input vectors and the normalized eigenvectors of the centered Gram matrix. If only the eigenvectors with corresponding nonzero eigenvalues are considered, the eigenvector ensemble can be written as

$$\mathbf{V}_\Phi = \mathbf{Y}_{c_\Phi} \mathcal{B} \Lambda^{-\frac{1}{2}} \tag{7.12}$$

where $\mathcal{B} = \left(\boldsymbol{\beta}^1, \boldsymbol{\beta}^2, \ldots, \boldsymbol{\beta}^{N_{out}}\right)^T$ and Λ is a diagonal matrix containing the corresponding eigenvalues. Inserting Equation (7.12) into Equation (7.9) and then subsequently substituting that result into Equation (7.6) gives

$$\mathbf{RX}(\Phi(\mathbf{r})) = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{Y_\Phi})^T \mathbf{Y}_{c_\Phi} \mathcal{B} \boldsymbol{\Lambda}_{\mathbf{Y}}^{-2} \mathcal{B}^T \mathbf{Y}_{c_\Phi}^T (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{Y_\Phi}). \tag{7.13}$$

For notational simplicity, let $\Phi(\mathbf{r})^T \mathbf{Y}_{c_\Phi}$ and $\hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}^T \mathbf{Y}_{c_\Phi}$ be as defined in Equations (6.21) and (6.22), respectively. Finally, it is possible to express

$$\hat{\mathbf{K}}_{\mathbf{Y}} = \mathcal{B} \boldsymbol{\Lambda}_Y \mathcal{B}^T. \tag{7.14}$$

It is possible to only use m vectors of $\mathcal{B}$ in Equation (7.14); doing so helps with the regularization issue. Using the results from Equations (6.21, 6.22, and 7.14) and inserting them into Equation (7.13) yields

$$\mathbf{KRX}(\mathbf{r}) = \left(\mathbf{K}_{\mathbf{Yr}}^T - \mathbf{K}_{\hat{\boldsymbol{\mu}}_Y}^T\right)^T \hat{\mathbf{K}}_{\mathbf{Y}}^{-2} \left(\mathbf{K}_{\mathbf{Yr}}^T - \mathbf{K}_{\hat{\boldsymbol{\mu}}_Y}^T\right). \tag{7.15}$$

Equation (7.15) is the kernel-RX equation used in this thesis.

Chapter 8

Kernel Parameters

8.1 RBF Kernel Parameter

Recall that the Gaussian radial basis function (RBF) kernel has the form

$$k(\mathbf{x}, \mathbf{y}) = e^{\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \tag{8.1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are input data vectors and $\sigma > 0$ is the RBF kernel parameter which represents the standard deviation and defines the width of the Gaussian distribution. When used in an anomaly detection setting, the choice of the kernel parameter is crucial; it has an undeniable effect on the performance of many kernel-based algorithms including the ones explored in this paper. Figure 8.1 shows the effect sigma can have on the detection performance of a subspace-based anomaly detector. The example shown consists of five values of σ ranging from 0.002 to 20 for a detector using KPCA. It can easily be seen that the performance of the detectors at different values of the kernel parameter are quite different. This example helps underscore the importance of choosing σ properly.

Unfortunately, the choice of an optimal σ is a highly nontrivial problem and at this time, there exist no analytical solutions for optimizing this parameter. This is mainly due to the fact that the optimal σ is highly data dependent; in addition, the best choice also changes with each algorithm.

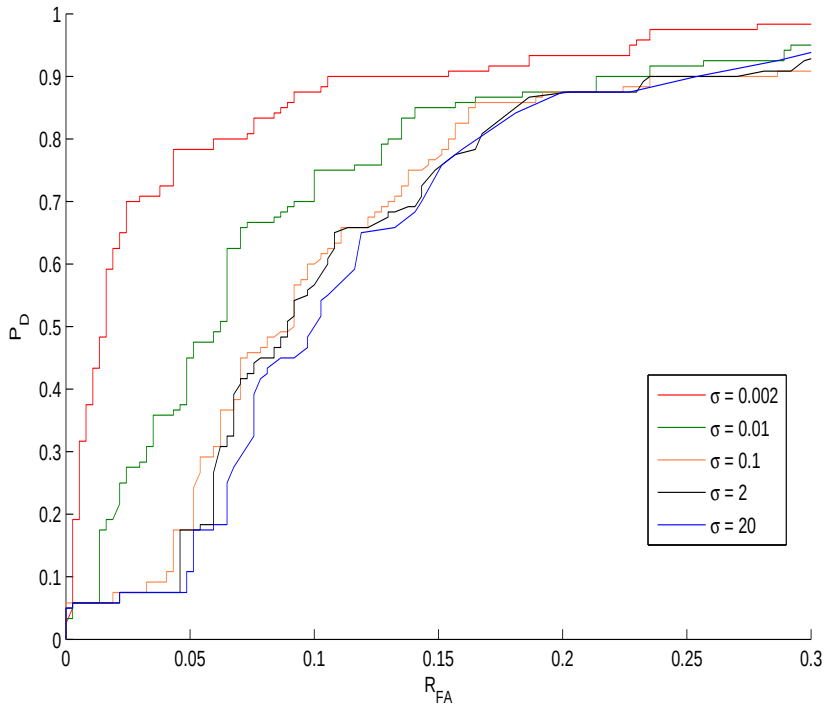


Figure 8.1: This figure illustrates the importance of choosing a good value of the kernel parameter, σ. Using σ values of [0.002, 0.01, 0.1, 2, 20], the detector performance changes drastically. If a bad value of σ was chosen, the detector would not perform up to its potential.

However, in order to compare the performance of these algorithms against the others it is important to be able to do so when each algorithm is exhibiting its best performance (or at least close to it). Therefore, an attempt was made in this thesis to experimentally optimize the value of σ by using a kind of cross-validation technique.

The algorithm for choosing a nearly optimal $\hat{\sigma}$ is as follows:

1. Vary the value of σ over an appropriate range.

2. Randomly sample β background pixels and τ target pixels. It is important here to choose β and τ such that there are enough pixels to have this method

be statistically worthwhile. For this paper, $\beta = 250$ and $\tau = 120$.

3. For each value of sigma, and at each pixel, open up a dual window obtain an output PSS value using one particular method.

4. Perform ROC analysis on the sampled results as a function of σ.

5. Determine which value of σ generated the best performance by calculating the highest area under the ROC curve.

Figure 8.2 shows an example of the algorithm outlined above. This example comes from analysis using a KEST-based anomaly detector on the DR-II HYDICE image. (See Chapter 9 for more on this image). From this graph, the value of the kernel parameter is chosen to be the one at which the value of the area under the ROC curve is highest - here, that value is $\sigma = 38$.

This method is advantageous for two reasons: first, it significantly reduces the computation time and secondly, it provides a relatively good approximation concerning the overall performance of a given method at each value of σ.

8.2 Choosing the Number of Eigenvectors

In the linear and nonlinear versions of both the PCA and EST algorithms, choosing the number of significant eigenvectors to use, m, becomes important. Fortunately, while varying the number of eigenvectors used affects the performance results, experimental results show that the difference in performance ability is not drastically altered - in fact, the difference is minimal at best. Figure 8.3 shows

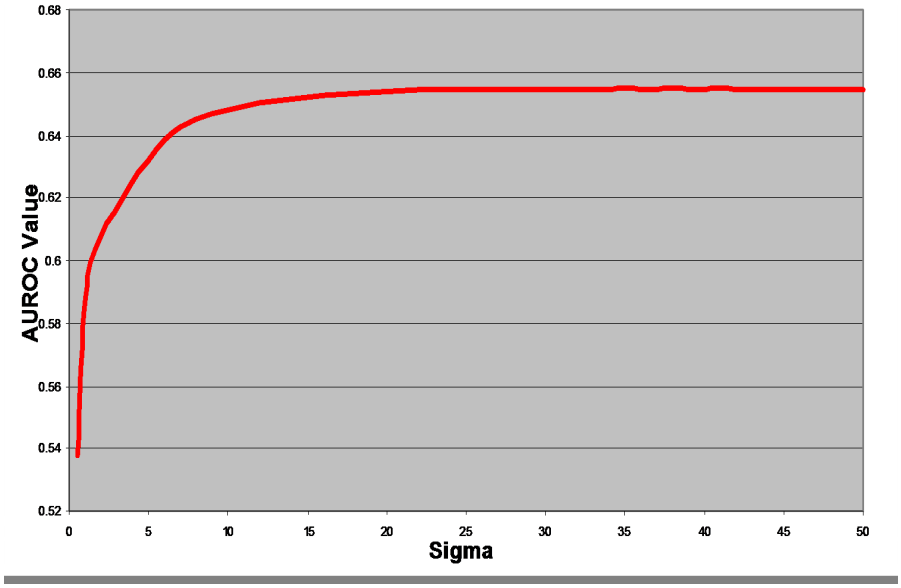


Figure 8.2: This figure shows an example of how σ was chosen in this thesis. The graph plots the area under the ROC (AUROC) curve as a function of σ. The value of σ corresponding to the greatest AUROC value was chosen as the kernel parameter. This example comes from a KEST analysis on the DR-II HYDICE image. Here, the optimal value for the kernel parameter occurs at around $\sigma = 38$.

the performance results of an EST-based subspace-anomaly detector when $m = 3, 5, 8, 10, 20,$ and 35. From this figure, it can be seen that altering the number of eigenvectors used as the bases for the subspace often does not drastically change the performance of the detector in question - if at all. Here, there is a slight yet insignificant change in detector performance. However, for the most part, it is clear that increasing the number of eigenvectors used will not significantly alter the performance. There are some cases, however, where the performance changes more than what is shown in the figure. It is for this reason that the following method is used to determine the number of eigenvectors to choose.

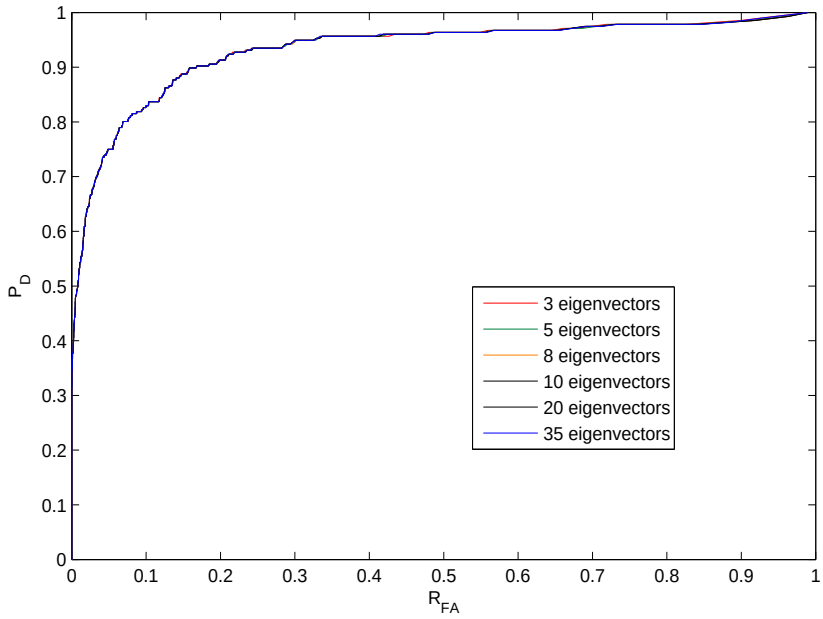


Figure 8.3: This figure shows an example of how the detector performance does not change significantly if the number of eigenvectors used as the bases vectors to generate the subspace is altered.

It is very difficult to systematically determine an optimal value for m. A graph of the average eigenvalues arranged from most significant to least significant can be found in Figure 8.4 and it can be seen that the eigenvalues decay very rapidly. Results indicate that choosing m so that the eigenvalues chosen represent a large portion of the energy yield acceptable outcomes. This technique was used in choosing m for PCA, EST, KPCA, and KEST. While it is certainly not a method which will optimize performance, this method does provide results which in most cases are relatively close to the optimal performance bound.

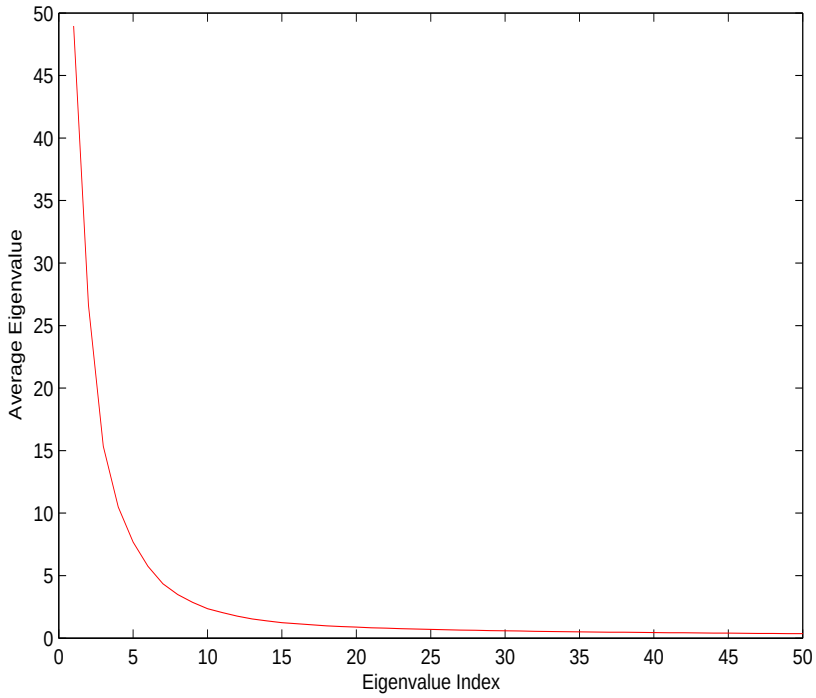


Figure 8.4: This plot shows the average eigenvalues for an anomaly detector using KPCA. It is easy to see that the eigenvalues die out very quickly. Results indicate that using the m most significant eigenvectors (where the corresponding m eigenvalues represent a large portion of the total energy in the eigenvalues) yield acceptable performance levels.

Chapter 9

Experimental Results

In this chapter, each of the eight algorithms is implemented using both simulated illustrative toy data as well as real hyperspectral imagery from the Hyperspectral Digital Imagery Collection Experiment (HYDICE) and Airborne Hyperspectral Imager (AHI) data sets. Performance results using ROC analysis, area under the ROC curve (AUC) analysis, and implementation time statistics for each of the eight methods are provided and compared.

9.1 Simulated Data

Each of the eight equations are implemented here as discrimination methods on an illustrative toy data set. The data set, shown in Figure 9.1, consists of two nonlinear Gaussian mixtures. Class 1 is represented by the red (*) points; Class 2 is represented by the blue (o) points. It is clear from this figure that no linear separating hyperplane can be placed that perfectly separates the two data classes.

In order to implement the algorithms, Class 1 and Class 2 were defined as the two sets corresponding to the data in the IWR and OWR of a fictional dual window. Extending the problem to an anomaly detection setting, Class 1 represents the target data and Class 2 represents the background data. The results using each of the methods on the simulated data set are shown in Figures 9.2(a)-9.2(h). To

improve visual quality, the points in Class 2 are now yellow (o). For the nonlinear algorithms, the kernel parameter σ was experimentally determined and set equal to a value which provided a decent looking result. The green lines in Figures 9.2(c), 9.2(e), and 9.2(g) are the projection vectors used in each case. The blue contour lines are decision boundaries at different thresholds. The shading defines relative projection separation values; lighter shading means a larger projection separation statistic value which in turn implies a higher likelihood that point will be classified as an anomaly. Similarly, darker areas correspond to points which are more likely to be classified as background clutter.

It is strikingly clear that all four of the nonlinear methods have significantly better discrimination abilities than their linear counterparts. Each of the four nonlinear methods generates decision boundaries which very nicely conform to the overall shape of the distribution. While it is difficult to actually compare the performance among the four nonlinear algorithms, it is nonetheless easy to see that the nonlinear methods perform better detection than the linear methods.

9.2 Hyperspectral Imagery

A total of five real hyperspectral images from two different HSI sensor data sets were use two compare the performances of the eight algorithms outlined above. Two of the images are from the Hyperspectral Digital Imagery Collection Experiment (HYDICE) data set and the other three are from the University of Hawaii's Airborne Hyperspectral Imager (AHI) sensor.

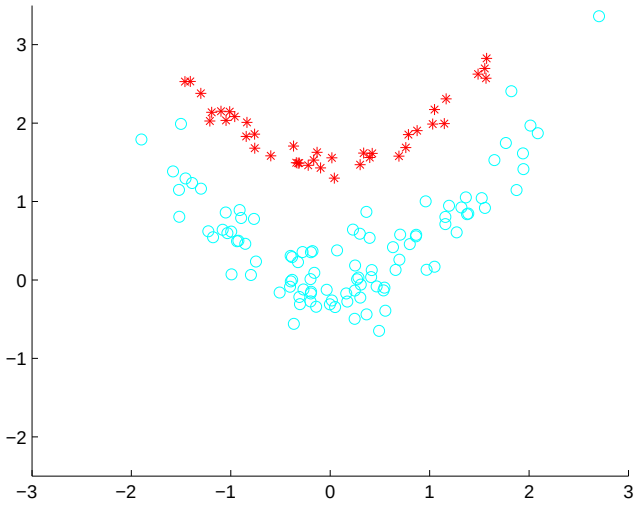
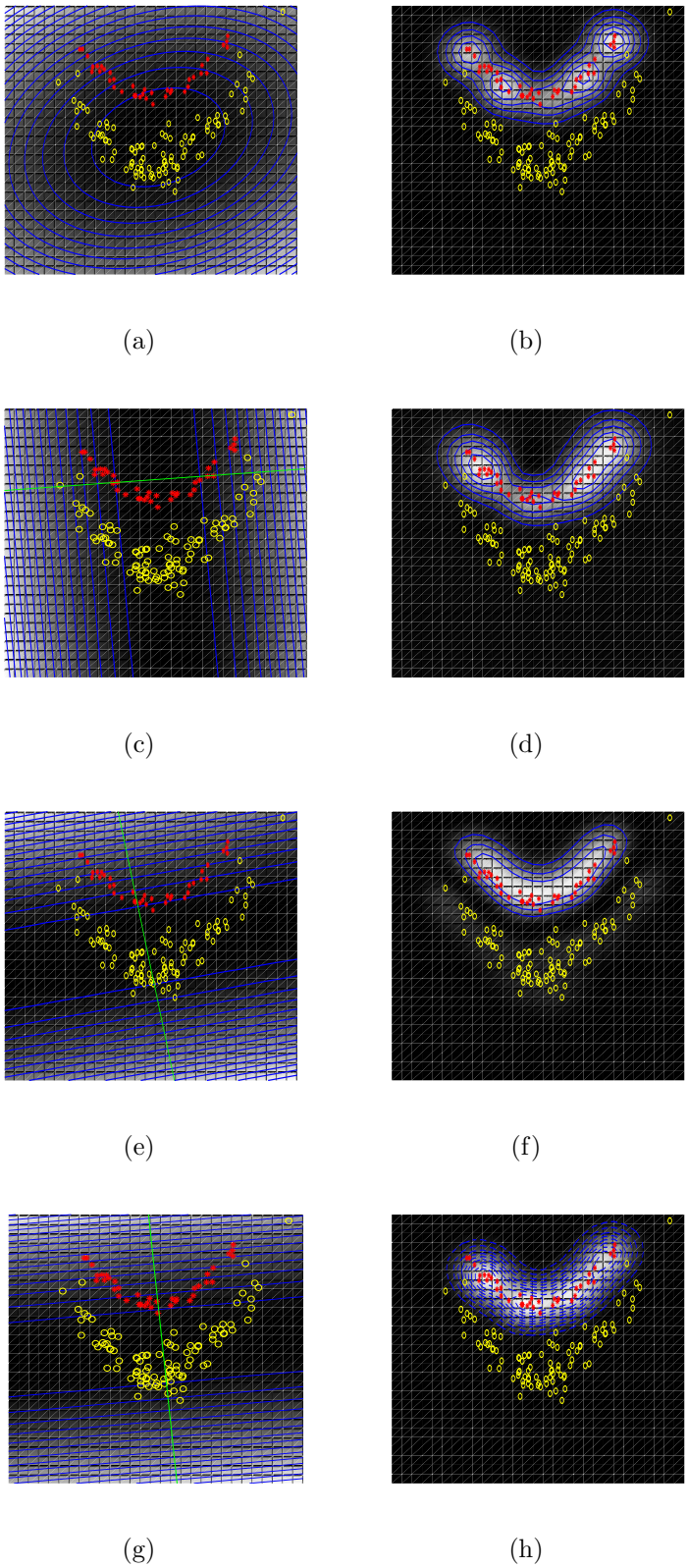


Figure 9.1: Original Simulated 2-D Data Set. A mixture of two nonlinear Gaussian distributions. The red points (*) represent the data in Class 1 and the blue (o) represent the data in Class 2.

Before any processing, all spectra in each image are normalized so that all values the data cube lie between zero and one. The normalization factor is calculated as the largest value among all spectral components in each hyperspectral image. This normalization helps to effectively use the dynamic range of the RBF kernel [9]. In all algorithms a dual window was used to collect data as described in Chapter 3. To keep things consistent, an IWR of 7x7 pixels, a guard band of 9x9 pixels, and an OWR of 19x19 pixels were used for all algorithms and for all images. It was stated that the IWR size should be about as large as the biggest target in the image. This is more or less the case for all images. The size of the OWR was chosen such that there are a sufficient number of background samples available for further processing.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 9.2: Contour and surface plots for the 2-D simulated data set using (a) RX, (b) KRX, (c) PCA, (d) KPCA, (e) FLD, (f) KFD, (g) EST, and (h) KEST.

9.2.1 Methods of Comparison

In order to compare the performances of each of the methods, receiver operating characteristic (ROC) curves were generated based on ground truth information obtained from each image. The ROC curves provide a visual quantitative comparison by plotting the probability of correct detection, P_D, versus the false alarm rate (FAR), R_{FA}. For each hyperspectral image, ground truth was obtained by determining the locations of all pixels in the image which correspond to a target to be detected. To generate the ROC curves, the resulting output image from one of the detectors must first be normalized to values between 0 and 1; this can be easily accomplished by dividing each pixel in the image by the maximum pixel value. Next, a threshold (T) is varied from 1 decreasing incrementally to 0. Usually, the increment is kept small (on the order of 10^{-3} so that the resulting ROC curve is relatively smooth. At each threshold, all pixels with values greater than T are classified as a target while all pixels with values less than T are classified as background. Then, P_D and R_{FA} are calculated by

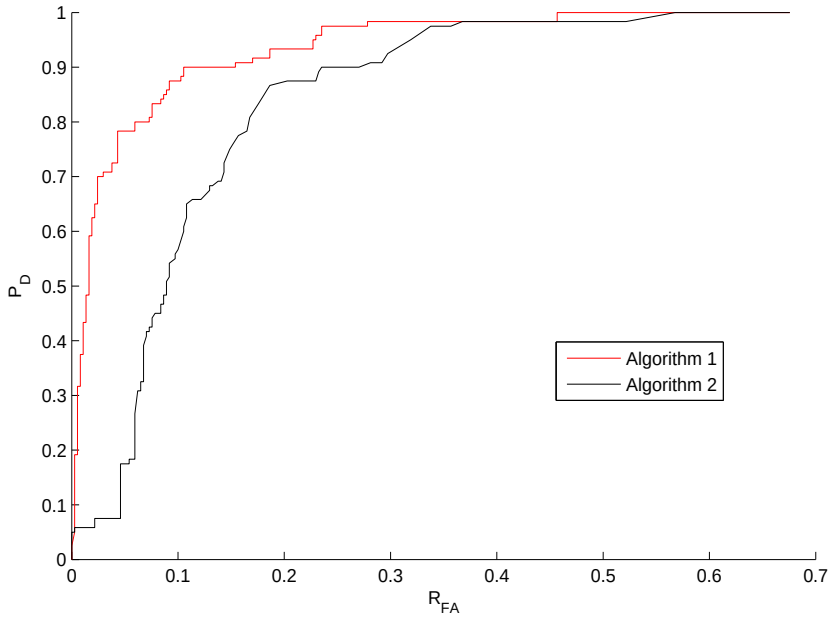
$$\begin{aligned} P_D &= \frac{N_{hit}}{N_T} \\ R_{FA} &= \frac{N_{miss}}{N_{TP}} \end{aligned} \tag{9.1}$$

where, at each threshold T, N_{hit} is the number of pixels correctly identified as target, N_T is the total number of target pixels in the ground truth for that image, N_{miss} is the number of pixels incorrectly labeled as targets, and N_{TP} is the total number of pixels in the image. The value of P_D is only equal to 1 when every single one of the target pixels are correctly detected. Since some of the target pixels may correspond

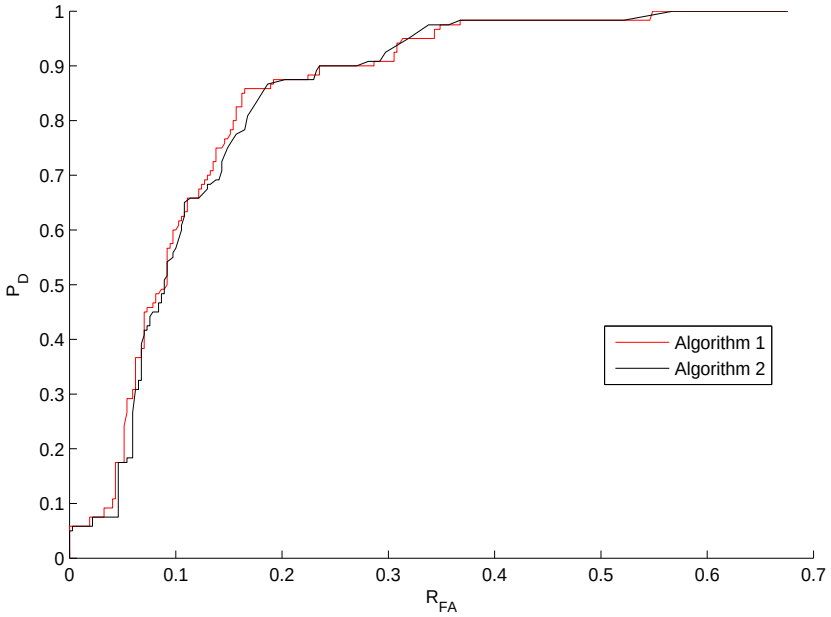
to mixed pixels, achieving a value of $P_D = 1$ at low FAR is often difficult to achieve. P_D is often less than 1 until higher FAR rates. Naturally, a 'good' ROC curve is one that approaches $P_D = 1$ very rapidly; in other words, for very good detection performance, the ROC curve should climb rapidly before it moves towards the right and levels out. Theoretically, the best possible detection result is one that detects all of the target pixels before it detects even one background pixel; this corresponds to an ROC curve that goes straight up until $P_D = 1$ and then remains at this value for all R_{FA}. In a geometric sense, the more concave the ROC curve, the better the performance of the anomaly detection method. For an anomaly detector, achieving this theoretical limit is extremely difficult.

Making a direct comparison of the ROC curves of different algorithms is occasionally a difficult task. In some instances, the ROC curve analysis for one method will exhibit better performance over another method over all values of R_{FA} as shown in Figure 9.3(a). In these instances it is easy to claim with confidence that Algorithm 1 performs better than Algorithm 2. In other cases, however, the ROC curves for two different algorithms can cross one or more times as in Figure 9.3(b). It cannot therefore be claimed that Algorithm 1 performs better than Algorithm 2 by simply looking at which curve is higher because at some point in the range of R_{FA}, each curve is higher than the other.

One way to address this issue is to compare the detection probabilities at a single operating point - this method will be referred to as single operating point (SOP) comparison. That is, given a false alarm rate, R_{FA}, determine the values of P_D for each of the methods at that FAR. Then simply use these detection prob-



(a)



(b)

Figure 9.3: This figure illustrates the occasional difficulty in analyzing ROC curves simply by visual observation. In (a), it is quite clear that Algorithm 1 performs better than Algorithm 2. However, (b) illustrates a case where it is not so obvious which algorithm is best. The ROC curves here are extremely close and cross each other many times over the whole range of R_{FA}. This issue illustrates the need for the use of AUROC analysis.

abilities to compare the performance of each algorithm. Unfortunately, there are situations where this method will fail to accurately characterize the performances of the methods. For example, consider a situation in which the ROC curve of Algorithm 1 is significantly higher than that of Algorithm 2 except for one very small FAR region around $R_{FA} = 0.1$. By visual inspection, it could be concluded with a high degree of confidence that Algorithm 1 performs better than Algorithm 2. However, if the SOP comparison is used to compare the performances of the two algorithms, and the operating point is blindly chosen at $R_{FA} = 0.1$, then the result will be that Algorithm 2 is incorrectly declared as the stronger performing method.

Another, more statistically significant way of addressing this issue is by using a singular scalar index which characterizes the performance of an ROC curve. In [31, 32], researchers claim that calculating the area under the ROC curve (AUC) provides such a measure. Geometrically, the AUC provides a sense of how concave the ROC curve is. As explained above, a greater concavity generally indicates a better performing detection algorithm. The AUC can also be calculated over different ranges of R_{FA}. Calculating it over the entire range (0 - 1) measures an overall performance of the algorithm. Calculating over a smaller range (varying from 0 to 0.3 for example) provides a measure of the performance of the algorithm over that FAR range.

In this thesis, a general algorithm comparison is qualitatively provided by visual subjective inspection of the output images followed by a more quantitative comparison using ROC curves and the AUC metric. The ranges over which the AUC is calculated is stated where appropriate.

Figure 9.4: A portion of the DR-II hyperspectral image from the HYDICE data set.

9.2.2 HYDICE Imagery

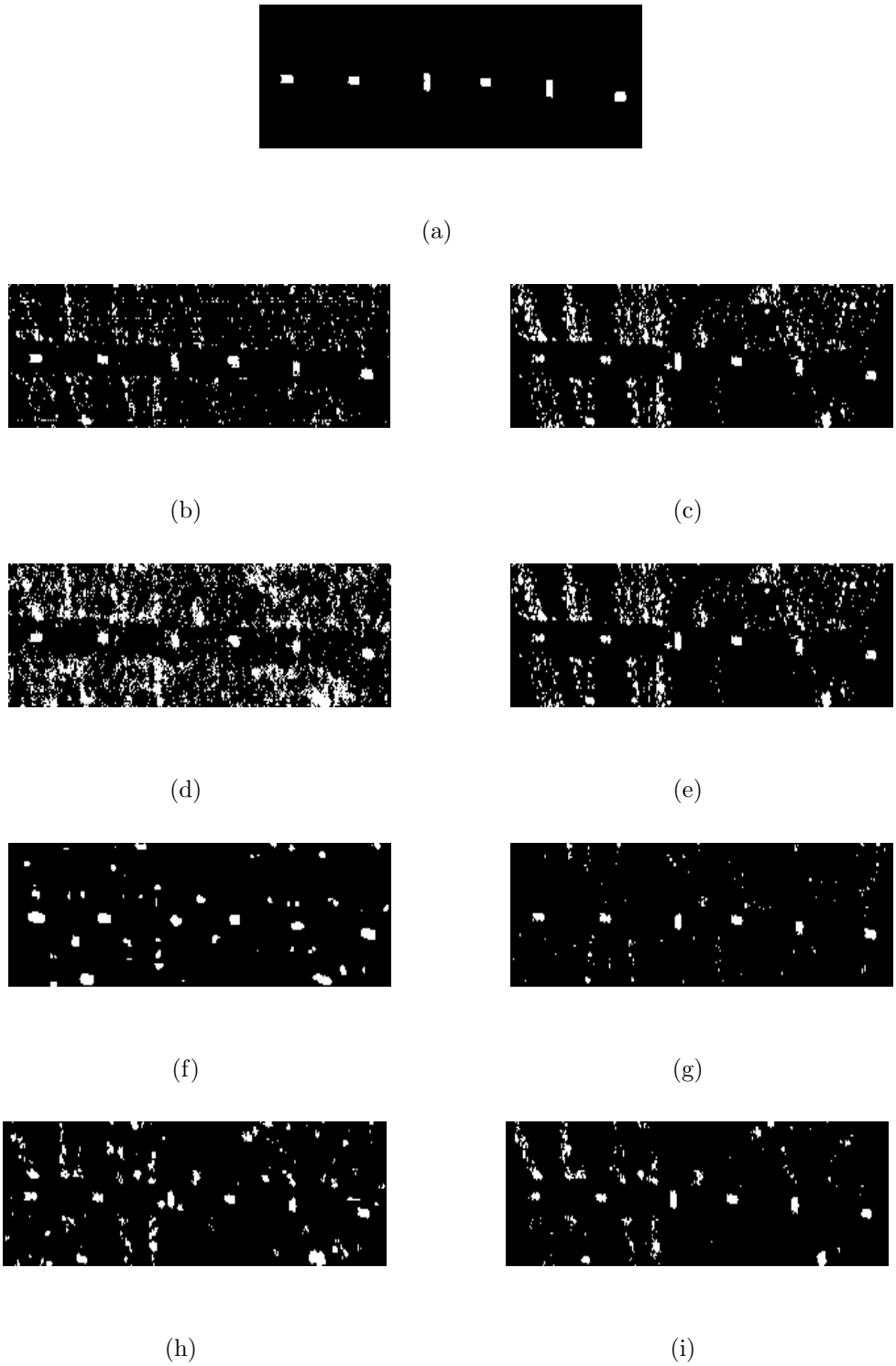
The HYDICE sensor collects radiance information over a spectral range spanning the VNIR and SWIR frequency ranges (0.4 - 2.5 μm). Each band is approximately 10 nm wide generating a spectral resolution consisting of 210 spectral bands. Due to water absorption and low signal-to-noise ratio (SNR), only 150 of those bands are actually used; bands 1-22, 102-108, 137-151, and 195-210 had been previously removed. The ground sample distance for a HYDICE sensor, which refers to the actual length on the ground corresponding to one pixel, depends on the height of the sensor during data collection and can range anywhere from 0.75 m - 4 m. The two HYDICE images used in this thesis are the Desert Radiance (DR-II) and Forest Radiance (FR-I) data sets. The DR-II image consists of 6 'targets of interest' on a dirt road running through a dusty terrain with light vegetation. The FR-I image has 14 'targets of interest' in a grassy field situated near a dense forrest. The DR-II and FR-I images are shown in Figures 9.4 and 9.5, respectively.

Figure 9.5: A portion of the FR-I hyperspectral image from the HYDICE data set.

9.2.2.1 DR-II Results and Analysis

The ground truth for the DR-II image is shown in Figure 9.6(a). It clearly shows the location of the six 'targets of interest'. All eight algorithms were implemented for this image and the best outputs for each can be seen in Figures 9.6(b) - 9.6(i). From the output images of the algorithms alone, it is often difficult to visually determine how well the anomaly detector is performing. Therefore, for visual purposes as well as for easier comparison, a binary threshold was used on each image where the threshold in each image corresponds to the level which achieves an 80% correct detection rate for that image. Note that he threshold changes for each output.

The results shown are the best results obtained using the detectors outlined in Chapters 4, 6, and 7. The optimal values for the kernel parameter σ and the number of eigenvectors to use were calculated using the algorithms described in Chapter 8. For PCA, the first 6 eigenvectors using the OWR spectra were used in Equation (4.11). For KPCA, the first 6 eigenvectors using the OWR spectra were used in Equation (6.24). For EST and KEST, the first 3 positive eigenvectors were



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 9.6: (a) Ground truth for the DR-II HYDICE image. Output results at 80% detection rate using (b) RX, (c) PCA, (d) FLD, (e) EST, (f) KRX, (g) KPCA, (h) KFD, and (i) KEST.

used in Equations (4.22) and (6.54), respectively. For RX and KRX, the highest 50 eigenvectors were kept when implementing the inverse covariance matrix.

The ROC curves over the entire range of FAR for each of the eight methods can be seen in Figure 9.7. To gain a better perspective on the performance of the algorithms at low FAR, Figure 9.8 shows the same results over a FAR range of 0 to 0.1. From the graphs alone, it appears that each of the four nonlinear methods perform better than their respective linear counterparts. In addition, all four nonlinear detectors aggregately exhibit better results than all four linear detectors. At low FAR, KPCA in this situation performs best among all methods followed by KRX, KEST and KFD. Among the linear methods, PCA, EST, and RX all perform about the same with FLD clearly performing the worst out of all the detectors.

The areas under the curves calculated over the full range of FAR (0-1) and over the subinterval (0-0.1) are shown in Tables 9.1 and 9.2. Table 9.1 indicates that on the average over all FAR, KEST actually performs the best among all detectors. The ROC curve for KEST crosses that of KPCA at a FAR of about 0.09. From then on, its performance remains at a higher level than any of the other detectors. Over all FAR, KEST, KPCA, and KRX all perform at about 95% average detection rate. KFD and RX are slightly lower at around 93% average detection rate. EST and PCA are around 92% average detection. FLD clearly performs the worst at around an average detection rate of 87%.

Table 9.2 indicates that at low FAR (between 0 and 0.1 FAR), KPCA performs the best among all detectors with an average detection rate of 87%. KEST and KRX perform relatively similarly at average detection rates of approximately 83%

Full Range (0-1)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.9393	93.93%	KRX	0.9501	95.01%
PCA	0.9208	92.08%	KPCA	0.9524	95.24%
FLD	0.8691	86.91%	KFD	0.9393	93.93%
EST	0.9211	92.11%	KEST	0.9567	95.67%

Table 9.1: AUC and average detection rates under the ROC curves for the DR-II results calculated over the full FAR range [0,1].

Low FAR Range (0-0.1)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.0741	74.10%	KRX	0.08159	81.59%
PCA	0.07113	71.13%	KPCA	0.08713	87.13%
FLD	0.06075	60.75%	KFD	0.07933	79.33%
EST	0.07115	71.15%	KEST	0.08298	82.98%

Table 9.2: AUC and average detection rates under the ROC curves for the DR-II results calculated over the FAR range [0,0.1].

and 82%, respectively. KFD has an average detection rate of 79%. Both EST and PCA exhibit an average detection rate of approximately 71% while RX performs at around 74%. FLD again performs worst among all detectors with an average detection rate of 61%. These results confirm the fact that even at low FAR, the nonlinear detectors each perform better than their respective linear counterparts. In addition, as a whole, all four nonlinear detectors also perform better than all four linear methods.

9.2.2.2 FR-I Results and Analysis

The ground truth for the FR-I HYDICE image is shown in Figure 9.9(a). It clearly shows the location of the fourteen 'targets of interest'. All eight algorithms were implemented for this image and the best outputs for each can be seen in Figures

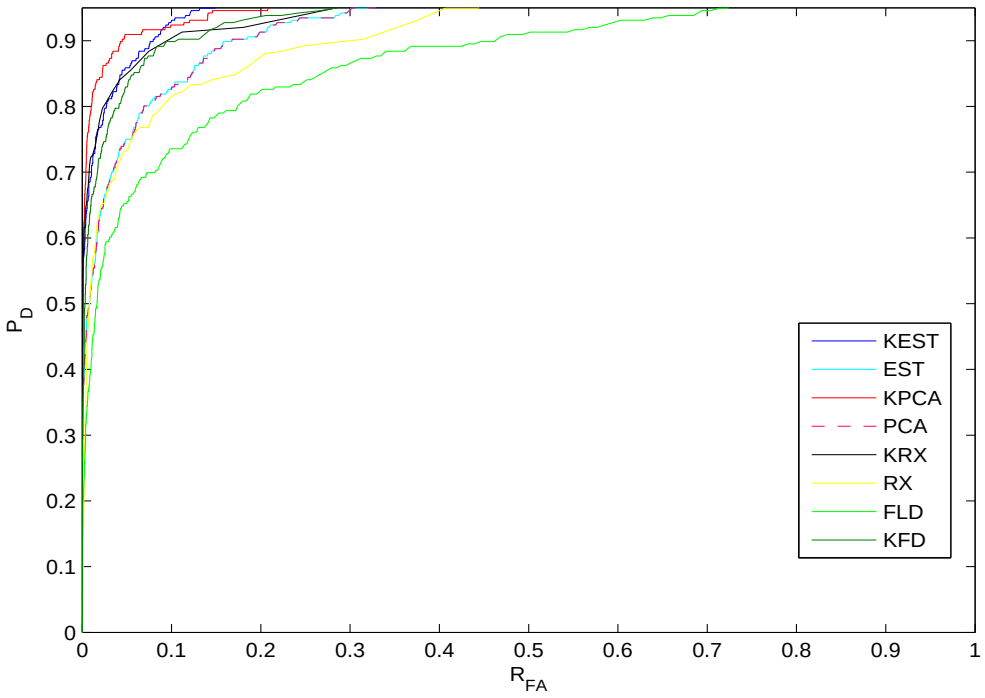


Figure 9.7: ROC curves for the DR-II image.

9.9(b) - 9.9(i). Again, for visual purposes as well as for easier comparison, a binary threshold was used on each image where the threshold in each image corresponds to the level which achieves an 80% correct detection rate for that image.

The results shown are the best results obtained using the detectors outlined in Chapters 4, 6, and 7. As in the analysis for the DR-II HYDICE image, the optimal values for the kernel parameter σ and the number of eigenvectors to use were calculated using the algorithms described in Chapter 8. The best results for PCA and KPCA were achieved using the same number of eigenvectors as above. That is, for PCA, the first 6 eigenvectors using the OWR spectra were used in Equation (4.11) and for KPCA, the first 6 eigenvectors using the OWR spectra were used in Equation (6.24). For EST, the first three negative eigenvectors were

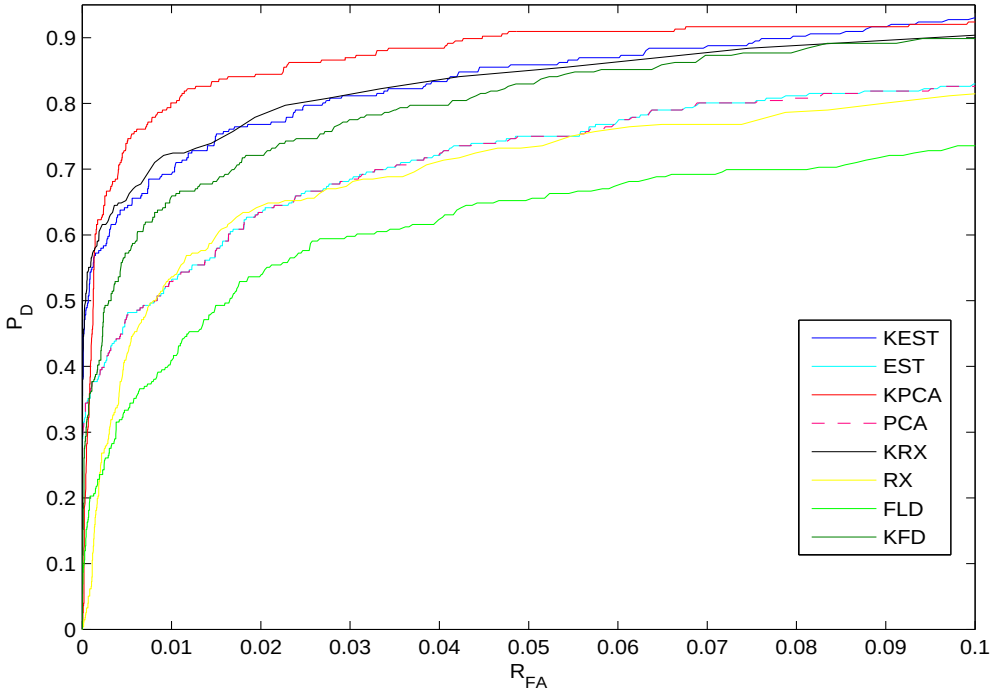
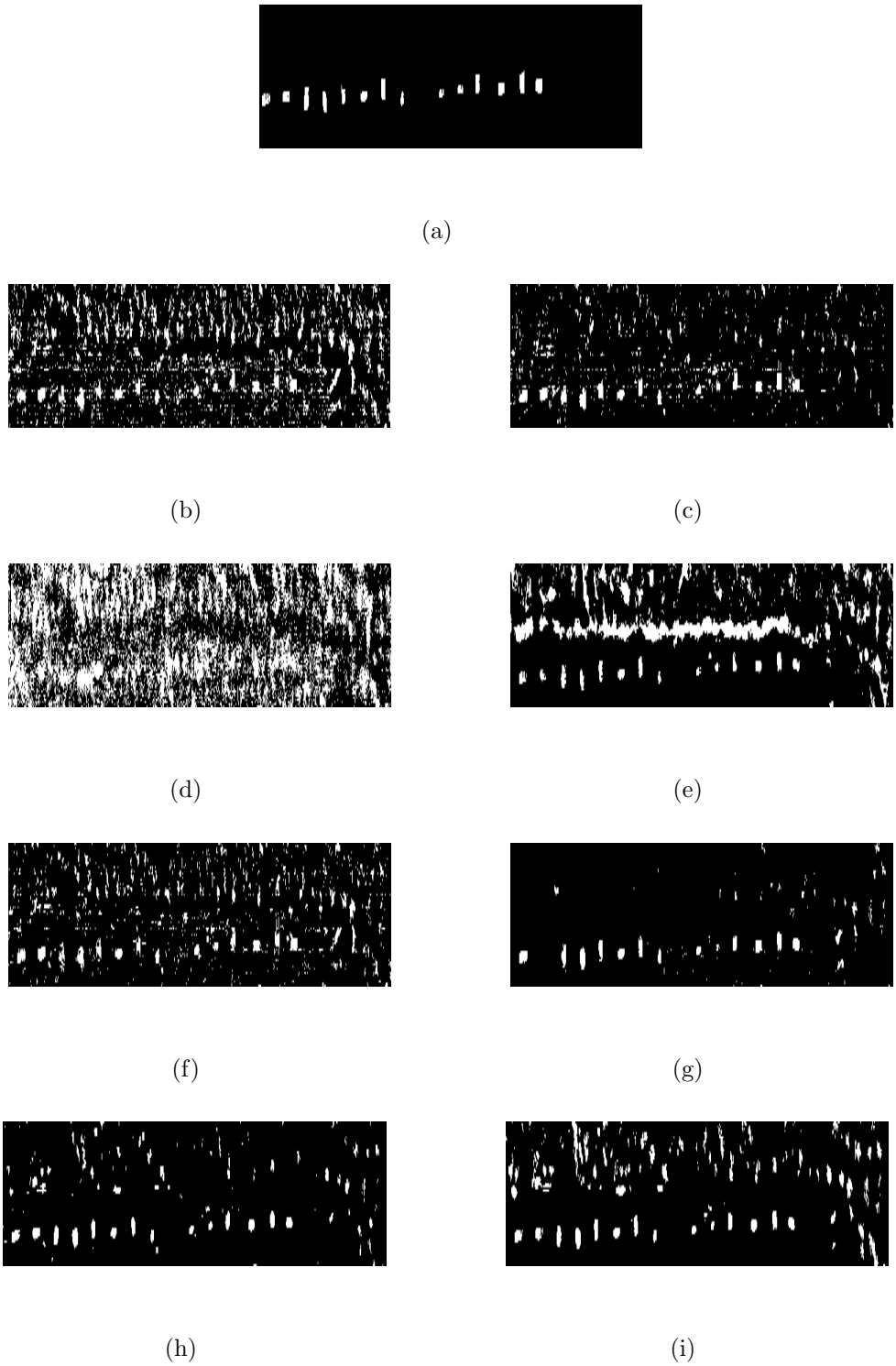


Figure 9.8: ROC curves for the DR-II image at very low false alarm rates.

used in Equation (4.23) and for KEST the first 3 positive eigenvectors were used in Equation (6.54). For RX and KRX, the highest 50 eigenvectors were kept when implementing the inverse covariance matrix.

From the images, it is clear that KPCA performs the best among all eight algorithms for this image since it detects very few background clutter regions. The results for KFD, KEST, and PCA also appear to perform very well with slightly more false alarms detected at this detection rate. KEST appears to perform much better than EST for this image as EST exhibits a large amount of false alarms around the treeline region. A region similar to this one could prove to be problematic for anomaly detectors as there is an abrupt change from foliage material to a shadowed grassy region. Once again, the result using FLD is the poorest among all detectors.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 9.9: (a) Ground truth for the FR-I HYDICE image. Output results at 80% detection rate using (b) RX, (c) PCA, (d) FLD, (e) EST, (f) KRX, (g) KPCA, (h) KFD, and (i) KEST.

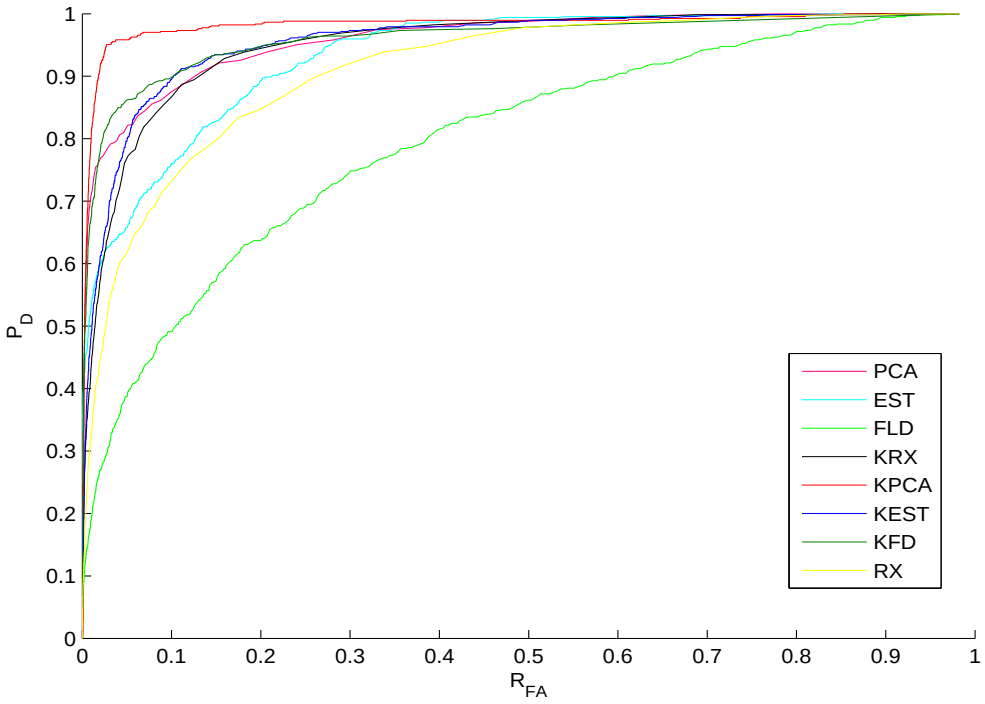


Figure 9.10: ROC curves for the FR-I image.

The ROC curves for each of the eight algorithms over all false alarm rates as well as at low FAR are shown in Figures 9.10 and 9.11, respectively. These figures confirm the analysis given above based on the output images. It is easy to see that KPCA performs the best both at low FAR and over the entire range and that FLD performs the worst by far. At very low R_{FA}, PCA outperforms all algorithms except KPCA and KFD. The ROC curve results indicate that each of the nonlinear algorithms performs better compared with their respective linear versions. That is, KPCA does better than PCA, KRX does better than RX, etc. However, since PCA performs well for this image, it cannot be said that all nonlinear versions as a whole perform this task better than the four linear methods.

The areas under the curves calculated over the full range of FAR (0-1) and

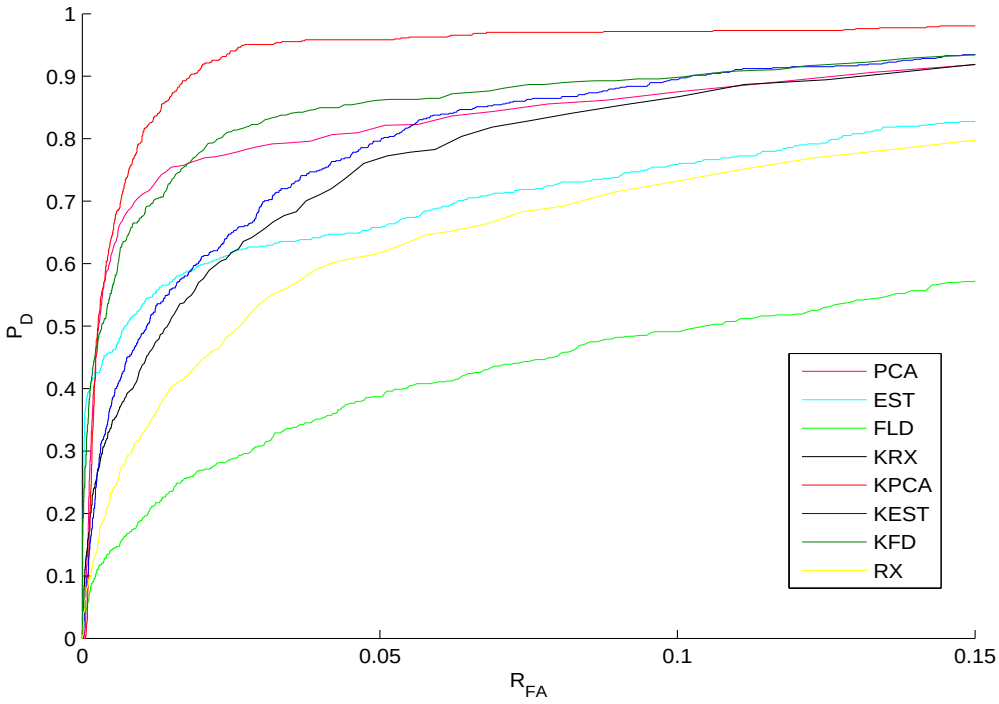


Figure 9.11: ROC curves for the FR-I image at very low false alarm rates.

over the subinterval (0-0.15) are shown in Tables 9.3 and 9.4. The AUC statistics confirm the analysis provided for the ROC curves. That is, each nonlinear algorithm performs better than its respective linear method. KPCA shows the best performance on the FR-I image with an average detection rate of about 93% at low R_{FA}. KFD also performs very well at about 85% detection rate in this range. For this image, PCA performs at a rate of approximately 82.5%, better than all other detectors except for KPCA and KFD.

9.2.3 AHI Imagery

The other three images are from Hawaii's Airborne Hyperspectral Imagery (AHI) database. These hyperspectral cubes contain 70 spectral bands and span

Full Range (0-1)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.9122	91.22%	KRX	0.934	93.40%
PCA	0.9424	94.24%	KPCA	0.9653	96.53%
FLD	0.7803	78.03%	KFD	0.9425	94.25%
EST	0.9183	91.83%	KEST	0.9387	93.87%

Table 9.3: AUC and average detection rates under the ROC curves for the FR-I results calculated over the FAR range [0,1].

Low FAR Range (0-0.15)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.0951	63.40%	KRX	0.114	76.00%
PCA	0.1237	82.47%	KPCA	0.1399	93.27%
FLD	0.062	41.33%	KFD	0.1276	85.07%
EST	0.105	70.00%	KEST	0.119	79.33%

Table 9.4: AUC and average detection rates under the ROC curves for the FR-I results calculated over the FAR range [0,0.15].

the long-wave infrared (LWIR) frequency range (8 - 11.5 μm). Thus, a spectral resolution of 50 nm is provided by the sensor. The AHI-1, AHI-2, and AHI-3 images used in this thesis are shown in Figures 9.12, 9.17, and 9.21, respectively. The images shown are only a portion of a each hyperspectral image; these smaller regions were used in order to reduce computation times. The light blue objects in the images are fiducials, man-made objects used for post-collection calibration and ground truthing. These objects are not considered as targets to detect and pre-processing is done in order to remove their effect on the results.

9.2.3.1 AHI-1 Results and Analysis

The ground truth for the AHI-1 image is shown in Figure 9.13(a). It shows the locations of the thirty-five 'targets of interest' - buried mines in this case. All

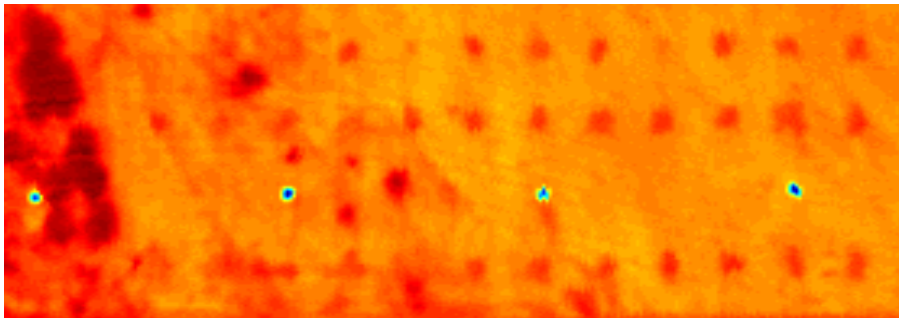


Figure 9.12: A portion of a hyperspectral mine image from the AHI database.

eight algorithms were once again implemented for this image and the best outputs for each can be seen in Figures 9.13(b) - 9.13(i). Once again, for visual purposes as well as for easier comparison, a binary threshold was used on each image where the threshold in each image corresponds to the level which achieves an 80% correct detection rate for that image.

The results shown are the best results obtained using the detectors outlined above. As in the analysis for the HYDICE images, the optimal values for the kernel parameter σ and the number of eigenvectors to use were calculated using the algorithms described in Chapter 8. For PCA, the first six eigenvectors using the OWR spectra were used in Equation (4.11) and for KPCA, the first six eigenvectors using the OWR spectra were used in Equation (6.24). For EST, the first five positive eigenvectors were used in Equation (4.22) and for KEST the first five positive eigenvectors were used in Equation (6.54). For RX, the first 50 eigenvectors were kept when implementing the inverse covariance matrix. Similarly, for KRX, the first 50 eigenvectors were used as $\mathcal{B}$ in Equation (7.14).

The ROC curves for each of the eight algorithms over all false alarm rates as

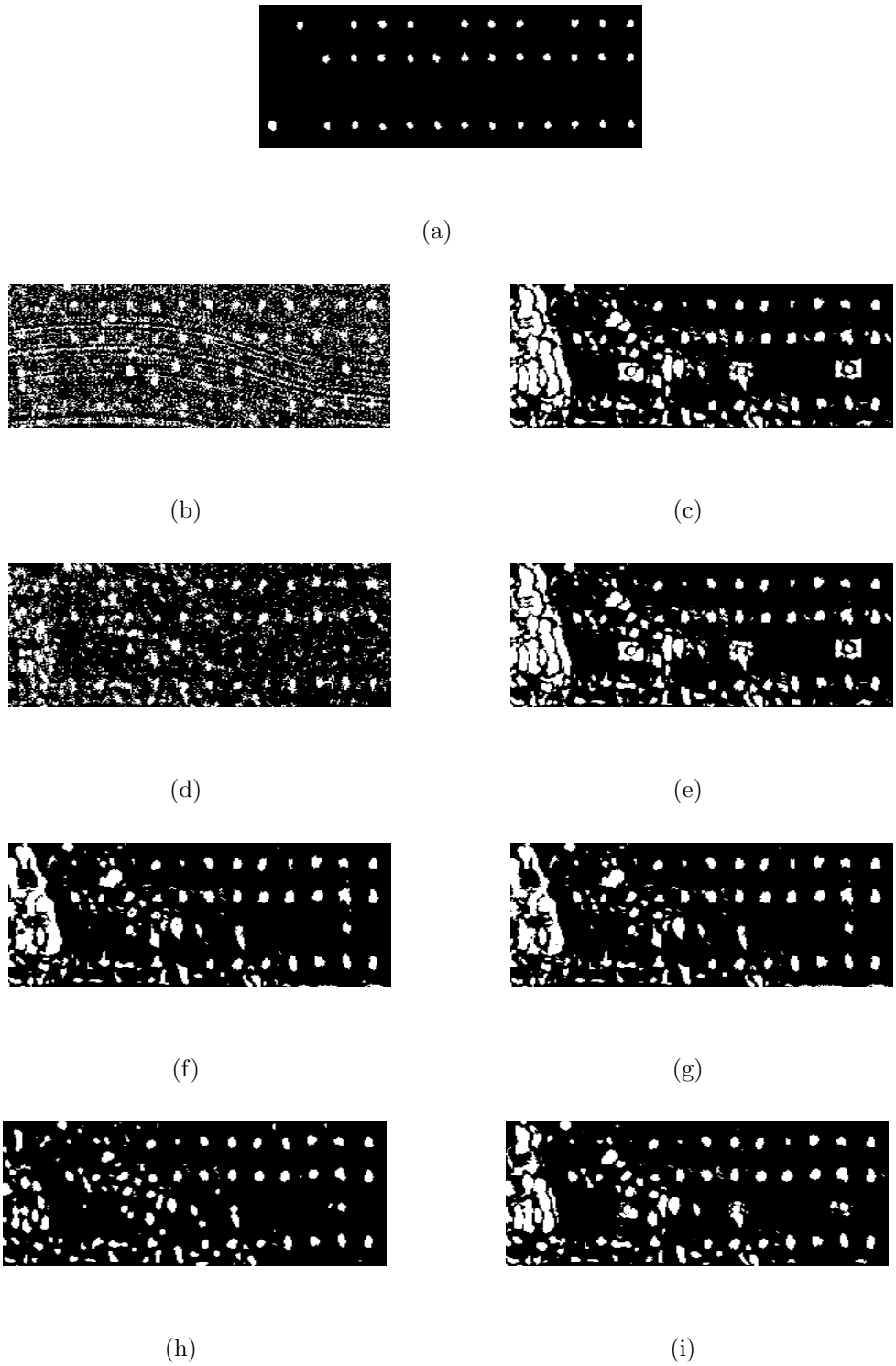


Figure 9.13: (a) Ground truth for the AHI-1 image. Output results at 80% detection rate using (b) RX, (c) PCA, (d) FLD, (e) EST, (f) KRX, (g) KPCA, (h) KFD, and (i) KEST.

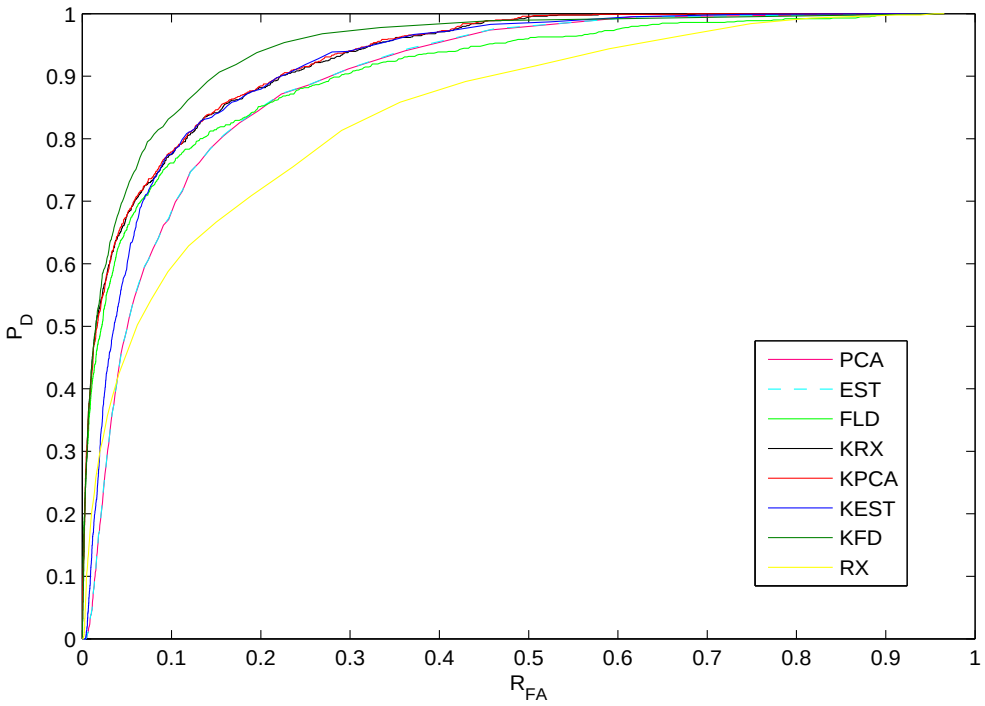


Figure 9.14: ROC curves for the AHI-1 image.

well as at low FAR are shown in Figures 9.14 and 9.15, respectively. In addition, the areas under the curves over all false alarm rates and at low FAR are presented in Tables 9.5 and 9.6, respectively. From the images, ROC curves, and AUC analysis, it can easily be seen that for this image over all detection rates, KFD performs the best among all methods at about 94.5% average detection rate while RX clearly exhibits the worst detection performance here at around 79.5% average detection rate. KPCA and KRX both perform very similarly throughout the entire image providing a detection rate of about 93%. From the ROC analysis, it can be seen that after a false alarm rate of 0.08, the KPCA, KRX, and KEST detectors all exhibit almost exactly identical performances. Among the linear methods, FLD performs the best at around 91% average detection rate. Both PCA and EST have

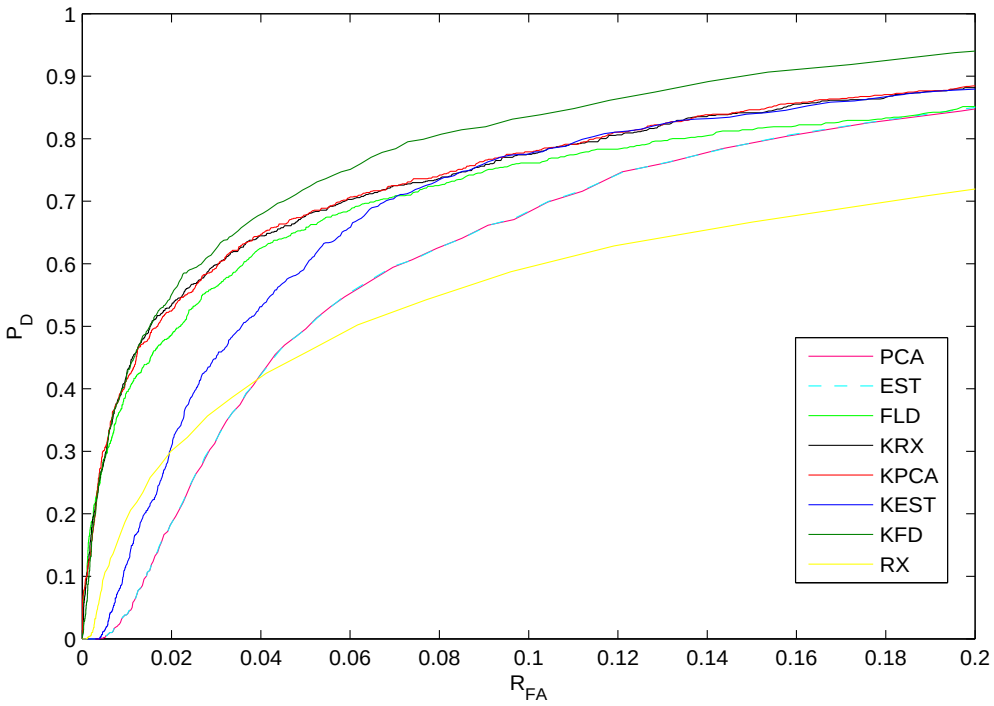


Figure 9.15: ROC curves for the AHI-1 image at very low false alarm rates.

an average detection rate of about 90%. Comparing overall detection statistics, all four nonlinear methods achieve better performance than all four linear methods.

However, at low false alarm rates (defined to be [0,0.2] for this image), FLD achieves a higher detection rate than KEST at around 70.5% compared to 68%. The other three nonlinear algorithms all perform better than all four linear methods. The RX detector really suffers from a high false alarm rate, only achieving an average detection rate of 54% over this range. This is due to a regularization issue in the RX-algorithm that has to deal with the fact that an inverse of a nearly singular matrix must be found. Once again, each nonlinear detector performs at a higher level than its linear counterpart. However, the increases from each linear detector performance to each nonlinear detector performance is not as large as in the HYDICE images.

In fact, the highest overall rate in the low R_{FA} range is KFD at around 77.5%. This is much lower than the highest detection rate for other images. The reason for this is most likely explained by two facts. The first is that in the HYDICE images, the targets were surface tanks which were clearly visible in the open. In this AHI image (and the two that follow), the targets of interest were either buried mines or disturbed soil. This means that the objects were either not visible in plain sight, or were not even placed at all (holes were dug, but nothing was placed in them). The second explanation is the large anomalous area detected on the left side of the images in Figure 9.13. This region corresponds to the darker regions in Figure 9.12. Further analysis leads to the conclusion that the terrain of these areas are vastly different spectrally than the background. Figure 9.16 shows the average spectra of background pixels compared with the average of some of the spectra in the dark region. It is easy to see that the spectral properties of the dark region are significantly different from those of the background immediately surrounding this area. This explains why these pixels are labeled as anomalies in the detector outputs and why a large number of false alarms are generated in this region. While they are in fact anomalies (with respect to the background), they are not considered targets. Thus, the nonlinear detectors suffer greatly from false alarms in this region, hindering their overall detection rates. From Figure 9.13(h), it can be seen that KFD does not generate a lot of false alarms in this region, helping it to achieve a higher detection performance than all other detectors for this image.

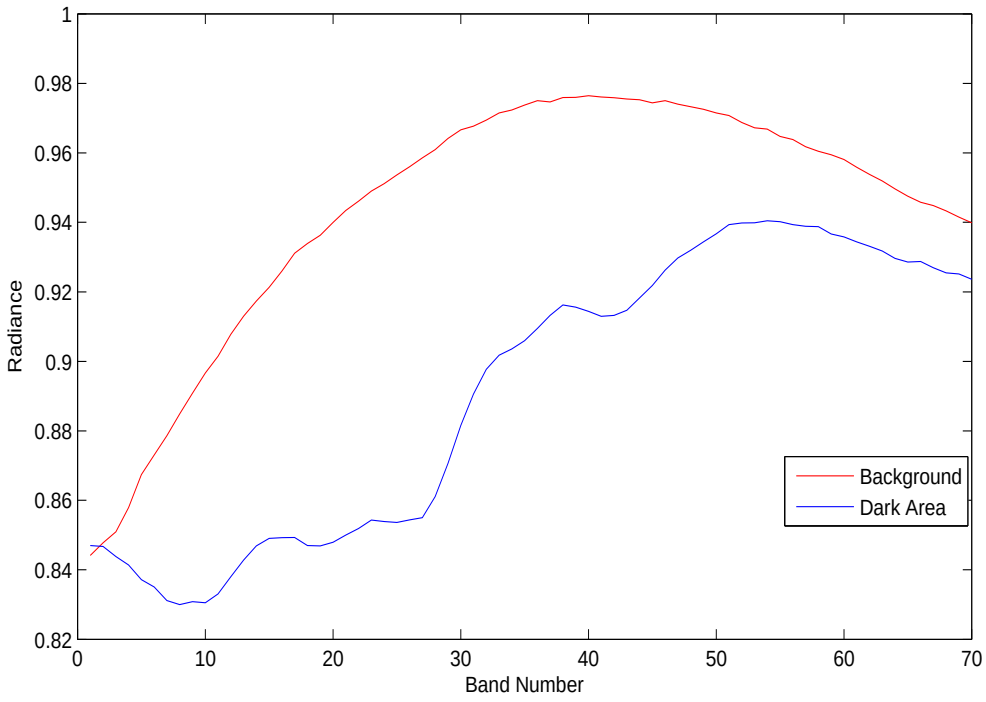


Figure 9.16: Average spectrum of background pixels versus average spectrum of dark area pixels in the AHI-1 image. The extreme difference in statistical properties between these two averages explains the high number of false alarms generated in this region.

Full Range (0-1)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.8467	84.67%	KRX	0.931	93.10%
PCA	0.897	89.70%	KPCA	0.933	93.30%
FLD	0.91	91.00%	KFD	0.946	94.60%
EST	0.897	89.70%	KEST	0.92	92.00%

Table 9.5: AUC and average detection rates under the ROC curves for the AHI-1 results calculated over the entire FAR range.

9.2.3.2 AHI-2 Results and Analysis

The ground truth for the AHI-2 image is shown in Figure 9.18(a). It shows the locations of the nineteen targets of interest'. All eight algorithms were once again

Low FAR Range (0-0.2)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.108	54.00%	KRX	0.146	73.00%
PCA	0.118	59.00%	KPCA	0.147	73.50%
FLD	0.141	70.50%	KFD	0.155	77.50%
EST	0.118	59.00%	KEST	0.136	68.00%

Table 9.6: AUC and average detection rates under the ROC curves for the AHI-1 results calculated over the low FAR range [0,0.2].

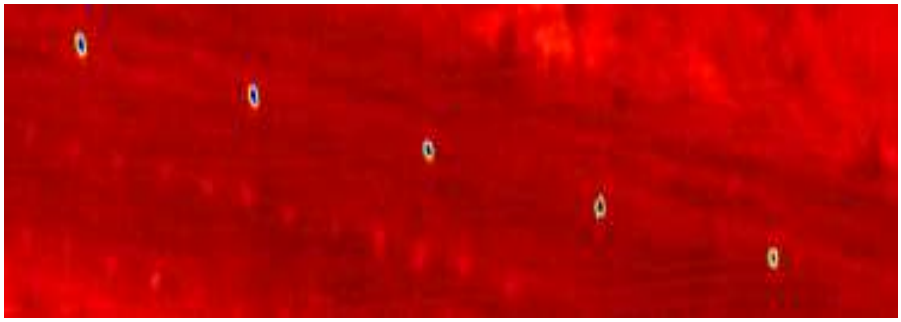
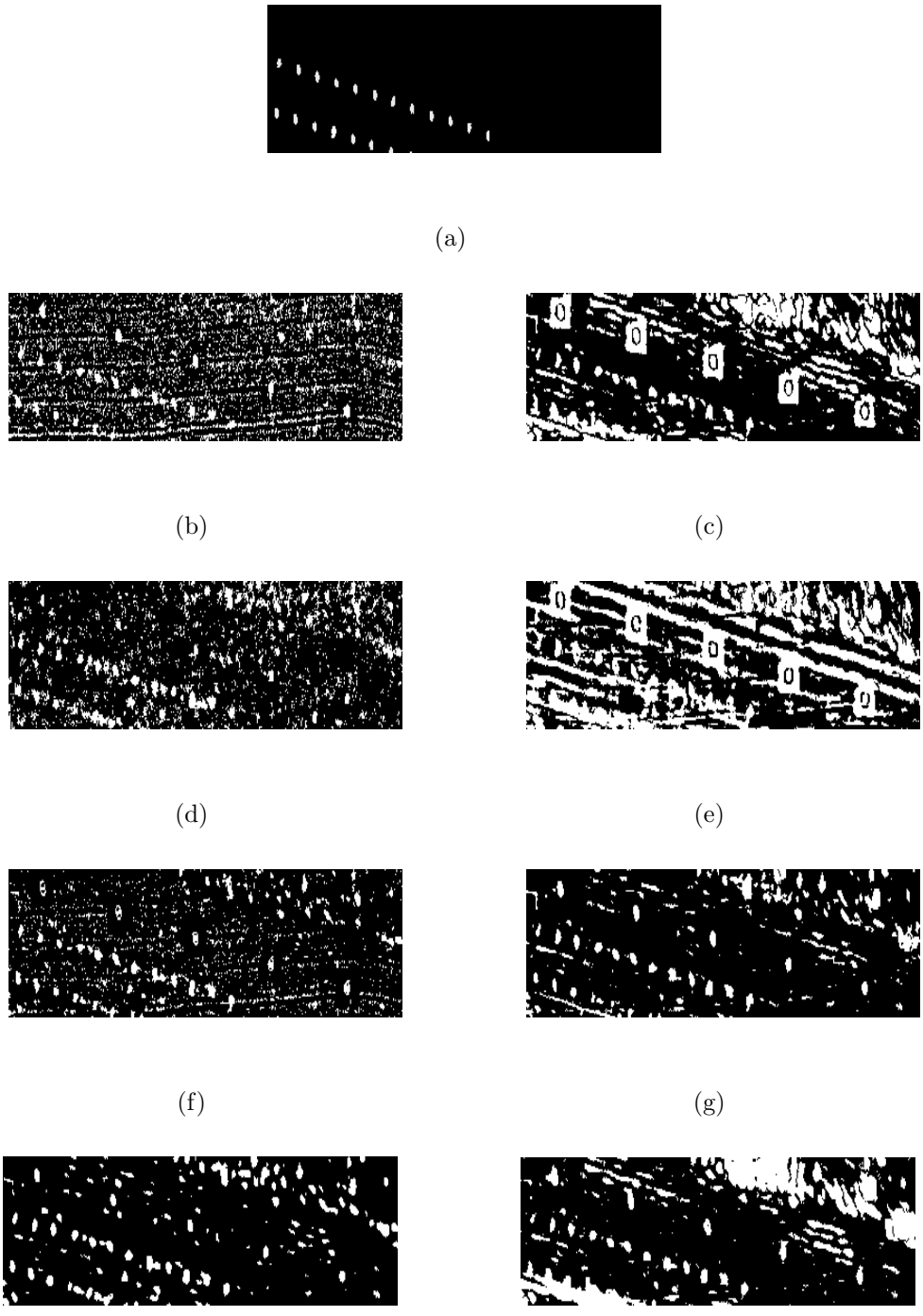


Figure 9.17: A portion of a hyperspectral mine image from the AHI database. This image is labeled AHI-2 in this thesis.

implemented for this image and the best outputs for each can be seen in Figures 9.18(b) - 9.18(i). A similar binary threshold as above was again used on each image where the threshold in each image corresponds to the level which achieves an 80% correct detection rate for that image. The results shown are the best results obtained using the detectors outlined above. For methods with configurable parameters, the same parameters are used as for the AHI-1 image above.

The ROC curves for each of the eight algorithms over all false alarm rates as well as at low FAR are shown in Figures 9.19 and 9.20, respectively. Furthermore, the areas under the curves over all false alarm rates and at low FAR are presented in Tables 9.7 and 9.8, respectively. From the images, ROC curves, and AUC results,



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 9.18: (a) Ground truth for the AHI-2 image. Output results at 80% detection rate using (b) RX, (c) PCA, (d) FLD, (e) EST, (f) KRX, (g) KPCA, (h) KFD, and (i) KEST.

we conclude that KFD performs the best among all methods at about 92% average detection rate while EST clearly exhibits the worst detection performance here at around 71% average detection rate. KPCA and KRX once again perform very similarly throughout the entire image providing a detection rate of about 90.5%. From the ROC analysis, it can be seen that after a false alarm rate of 0.3, the KPCA, KRX, and KEST detectors all exhibit very similar performances. After this rate, none can be considered a clear favorite. Among the linear methods, FLD performs the best at around 89% average detection rate. The AUC results indicate that RX outperforms PCA over the whole FAR range. In this sense, the output image for RX and PCA (Figures 9.18(b)-9.18(c), respectively) are a little bit misleading. RX looks relatively noisy compared to PCA; however, the ROC curves and AUC analysis indicates that the performance distinction is made at low FAR and as mentioned, these images are thresholded at a level which generates an 80% detection rate. PCA, EST, and KEST (the three worst performing methods at low FAR) all exhibit very large false-alarm clusters. These areas are the reasons for these algorithms' poor detection performances. At low FAR, KRX, KPCA, and KFD all perform about the same at around a 78.5% average detection rate. Comparing overall detection statistics, each nonlinear method once again achieves a better performance than its respective linear method. However, since FLD outperforms KEST both at low FAR and over the entire FAR range, the results from this image do not indicate that all of the nonlinear methods here perform better than all of the linear methods.

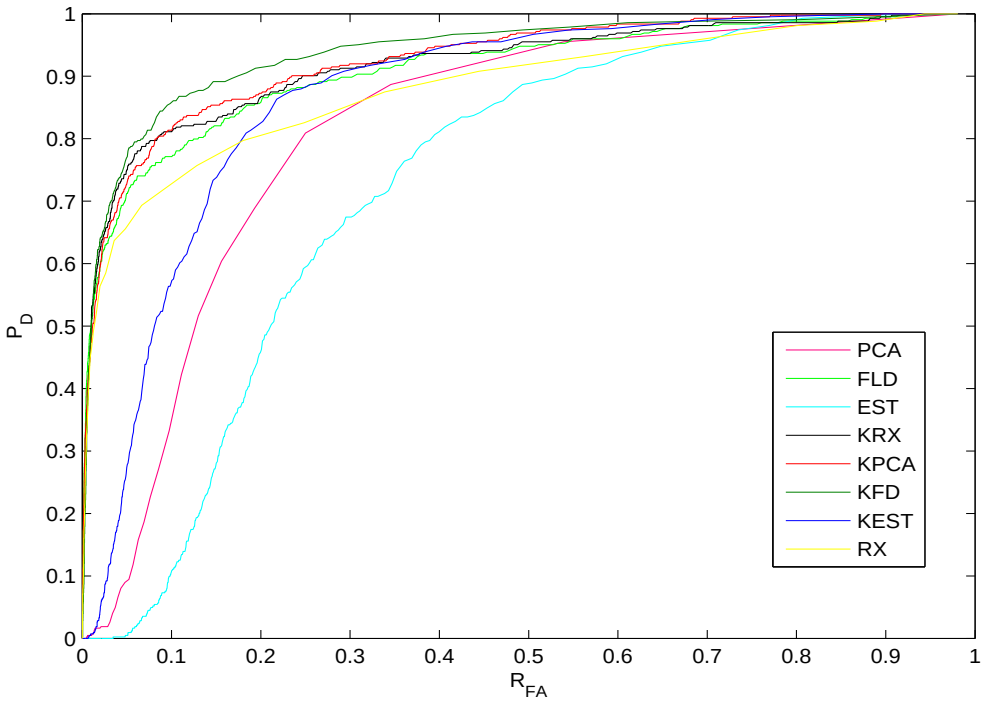


Figure 9.19: ROC curves for the AHI-2 image.

Full Range (0-1)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.8834	88.34%	KRX	0.901	90.10%
PCA	0.798	79.80%	KPCA	0.908	90.80%
FLD	0.893	89.30%	KFD	0.921	92.10%
EST	0.711	71.10%	KEST	0.851	85.10%

Table 9.7: AUC and average detection rates under the ROC curves for the AHI-2 results calculated over the entire FAR range.

Low FAR Range (0-0.25)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.179	71.60%	KRX	0.196	78.40%
PCA	0.107	42.80%	KPCA	0.197	78.80%
FLD	0.191	76.40%	KFD	0.197	78.80%
EST	0.057	22.80%	KEST	0.138	55.20%

Table 9.8: AUC and average detection rates under the ROC curves for the AHI-2 results calculated over the low FAR range [0,0.25].

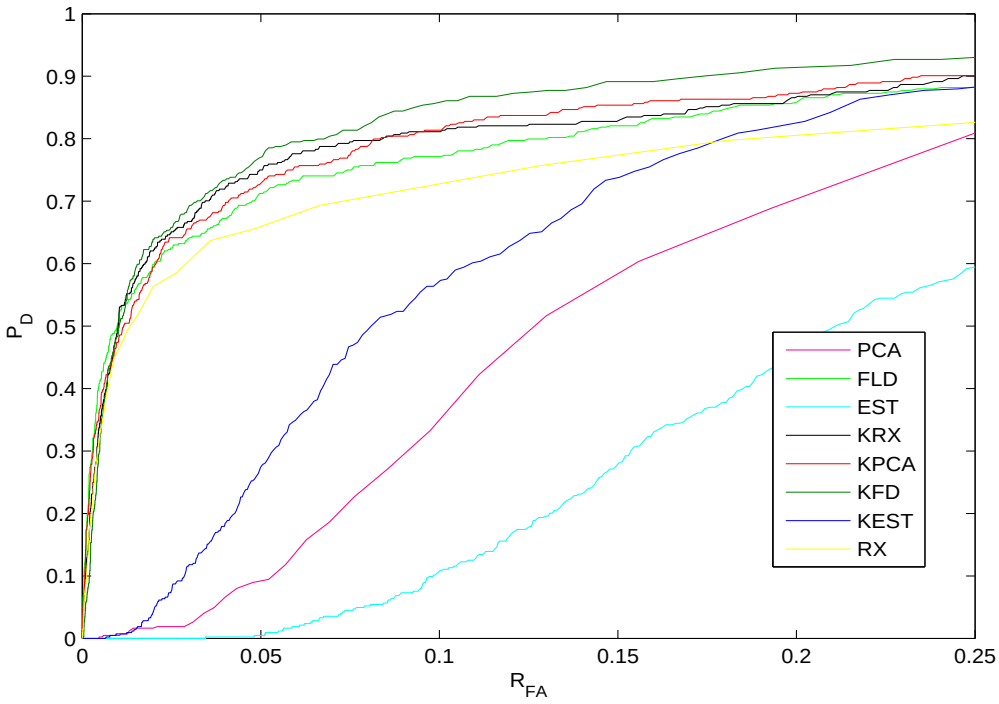


Figure 9.20: ROC curves for the AHI-2 image at low false alarm rates.

9.2.3.3 AHI-3 Results and Analysis

The ground truth for the AHI-3 image is shown in Figure 9.22(a). It shows the locations of the seventy-six 'targets of interest'. All eight algorithms were once again implemented for this image and the best outputs for each are displayed in Figures 9.22(b) - 9.22(i). A similar binary threshold as above was again used on each image where the threshold in each image corresponds to the level which achieves an 80% correct detection rate for that image. The results shown are the best results obtained using the detectors outlined above. For methods with configurable parameters, the same parameters are used as for the AHI-1 image above.

The ROC curves for each of the eight algorithms over all false alarm rates as well as at low FAR are shown in Figures 9.23 and 9.24, respectively. Furthermore,

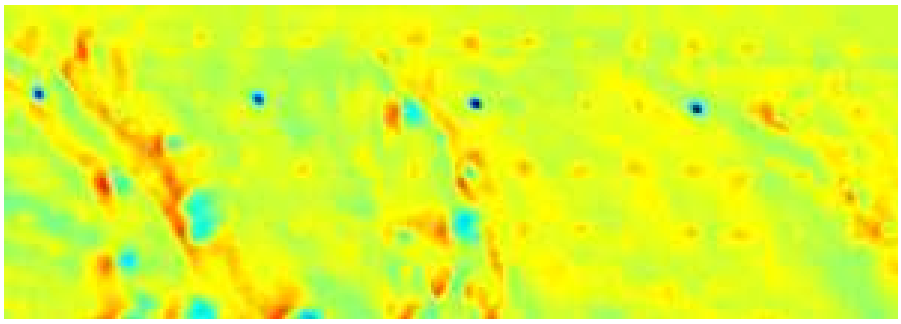


Figure 9.21: A portion of a hyperspectral mine image from the AHI database. This image is labeled AHI-3 in this thesis.

the areas under the curves over all false alarm rates and at low FAR are presented in Tables 9.7 and 9.8, respectively. From the images, ROC curves, and AUC analysis, it appears as if these algorithms have an extremely difficult time in detecting the target pixels. One reason for this could be due to the significant increase in the number of pixels to detect. Looking at the original image in Figure 9.21, another possible reason is that is much harder to locate the targets visually - an indication that the target spectra are much more similar to the local background than in the previous images. A very large number of false alarms are generate throughout the entire image for all of the methods, significantly reducing the performances for this image. It can be concluded that KFD performs the best overall for this image with an average detection rate of about 82.54%. However, its performance rate is not too much higher than KPCA or FLD. FLD once again is the best linear method and performs better than two of the nonlinear methods (KRX and KEST) over all false alarm rates as well as at low FAR with average detection rates of 80.5% and 66.7%, respectively.

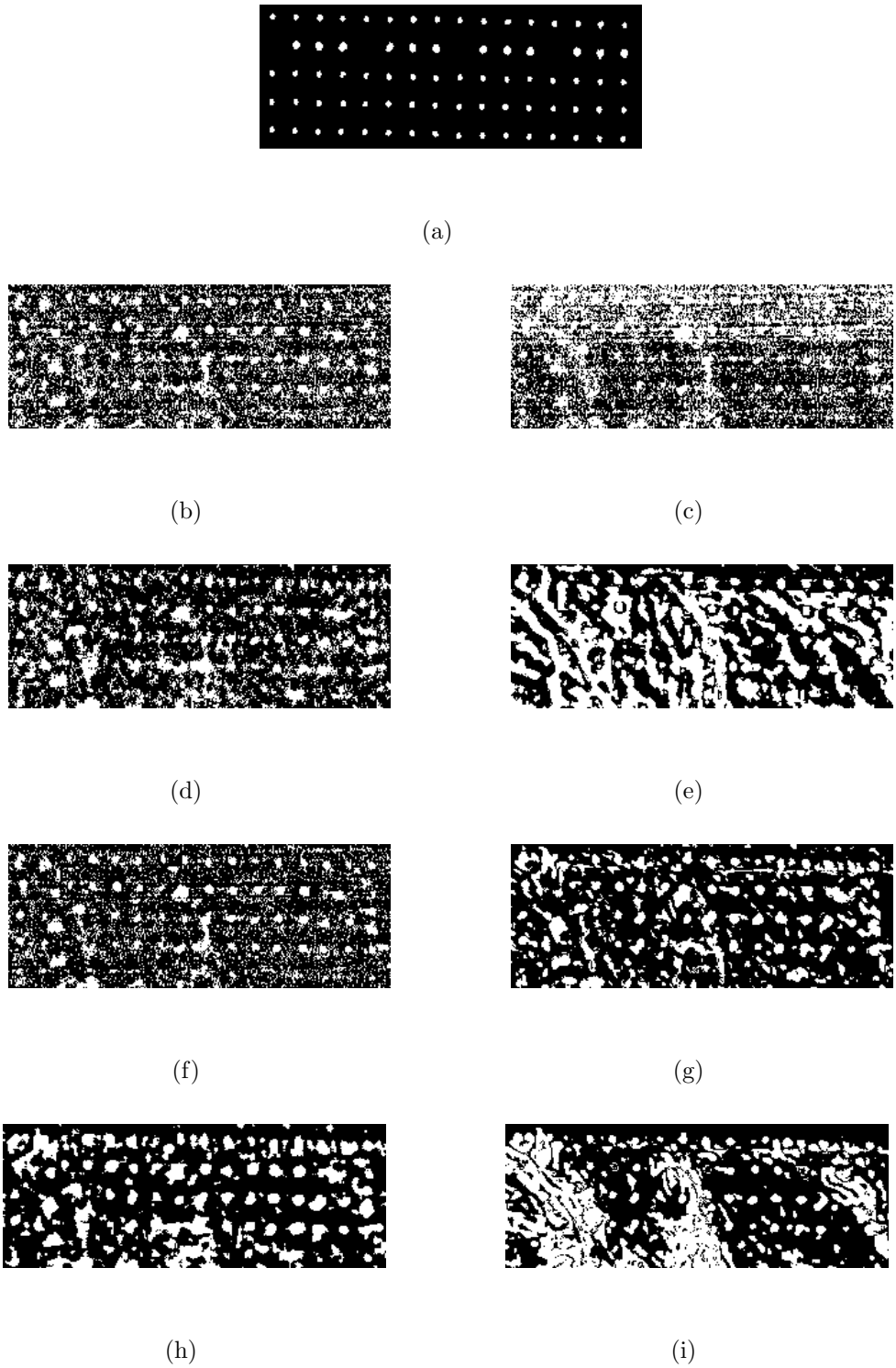


Figure 9.22: (a) Ground truth for the AHI-3 image. Output results at 80% detection rate using (b) RX, (c) PCA, (d) FLD, (e) EST, (f) KRX, (g) KPCA, (h) KFD, and (i) KEST.

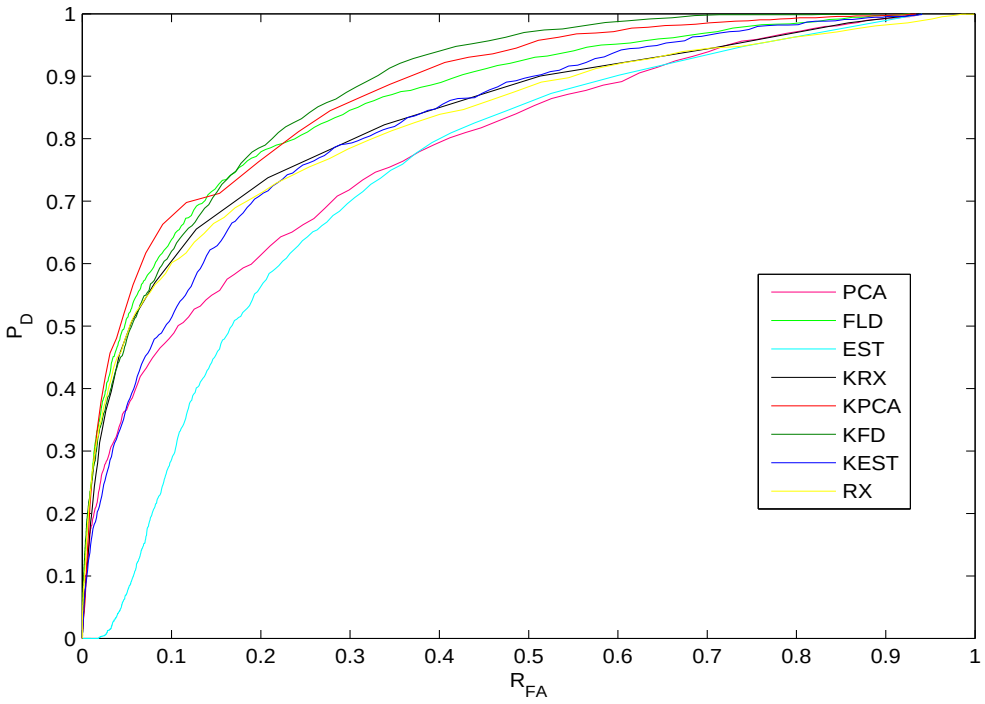


Figure 9.23: ROC curves for the AHI-3 image.

At low FAR, the detection performances for KPCA, KFD, and FLD are very similar with KPCA doing slightly better than the other two methods. In fact, at a FAR between 0.025 and 0.15, KPCA is the best method for this image. However, for FAR after 0.175, KFD performs the best among all methods. There is *very* small FAR range in which FLD is actually the best method. The performances of KFD and FLD can essentially be considered equal at low FAR. This might be due to a regularization issue in KFD. In order to fully investigate this effect, more analysis on the exact statistical properties of the image are required. Once again overall, each nonlinear method outperforms its corresponding linear method. However, it cannot be concluded that all of the nonlinear methods outperform all of the linear methods.

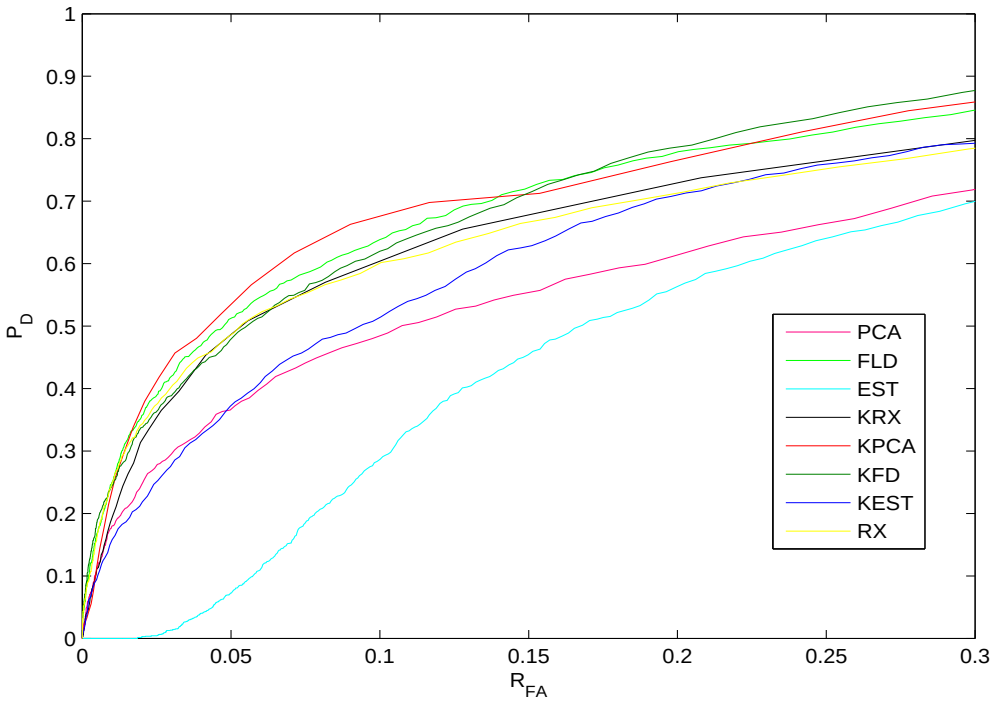


Figure 9.24: ROC curves for the AHI-3 image at low false alarm rates.

Full Range (0-1)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.8283	82.83%	KRX	0.776	77.60%
PCA	0.728	72.80%	KPCA	0.819	81.90%
FLD	0.806	80.60%	KFD	0.824	82.40%
EST	0.689	68.90%	KEST	0.767	76.70%

Table 9.9: AUC and average detection rates under the ROC curves for the AHI-3 results calculated over the entire FAR range.

Low FAR Range (0-0.3)					
Algorithm	**AUC**	**Avg. Det. Rate**	**Algorithm**	**AUC**	**Avg. Det. Rate**
RX	0.186	62.00%	KRX	0.184	61.33%
PCA	0.157	52.33%	KPCA	0.202	67.33%
FLD	0.2	66.67%	KFD	0.199	66.33%
EST	0.118	39.33%	KEST	0.173	57.67%

Table 9.10: AUC and average detection rates under the ROC curves for the AHI-3 results calculated over the low FAR range [0,0.3].

9.3 Further Analysis

The results shown above lead to the conclusion that each nonlinear method outperforms its corresponding linear method. The natural follow question might be to ask how significant this increase is. Since ROC analysis provides an immense amount of information, there are many other ways to analyze the data presented above. For example, one can calculate the percentage increase in detection probability at a specific false alarm rate from each linear method to its corresponding nonlinear method. Naturally, this increase will be different at various false alarm rates. However, this kind of analysis requires one to pick an appropriate FAR at which to perform these calculations. Unfortunately, without knowing the exact application for which this technology is being used, it is very difficult to know which FAR to choose. For example, if the application calls for accurate detection, a very low FAR would be required. On the other hand, if the application is such that a larger amount of false alarms are acceptable, then one might choose a slightly higher FAR. The point is that at these two distinct false alarm rates, the 'best' methods might possibly be different.

In addition, the results above indicate that the best performing method - while almost always a nonlinear method - is very much image dependent. If one were given an image to analyze and forced to choose one algorithm, the user must be aware of which method gives the highest probability of performing the best detection. In order to precisely answer this question, it would be necessary to assign some sort of value to each image measuring the ease of pixel detection. However, to simplify

Average Overall Detection Rates					
Method	**Avg. AUC**	**Avg. Det. Rate**	**Method**	**Avg. AUC**	**Avg. Det. Rate**
RX	0.8819	88.19%	KRX	0.89842	89.84%
PCA	0.85724	85.72%	KPCA	0.91554	91.55%
FLD	0.85168	85.17%	KFD	0.91456	91.46%
EST	0.82728	82.73%	KEST	0.88668	88.67%

Table 9.11: This table provides the average overall detection rates for each of the eight methods. These values were calculated by averaging the detection rates over each of the five images for each method.

matters, equal weights are assumed and the average probability of detection over all five images of each method is calculated; this is accomplished by averaging the detection rates for each of the methods over all images. These results are shown in Table 9.11.

From this table, it appears that both KPCA and KFD perform very well with average overall detection rates of 91.5%. KRX and KEST are not very far behind with average detection rates of around 89.8% and 88.7%, respectively. Based on these results, each nonlinear method outperforms its corresponding linear method. In addition, on average (over these five hyperspectral images) all nonlinear methods outperform all four linear methods, a result that cannot be confirmed based on the results of each individual image. This is only an average performance. Thus, it would seem that KPCA or KFD should be chosen as the method of choice if only one algorithm could be used at a time. Of the eight methods here, using one of those two algorithms would yield the highest chance of achieving the highest detection performance.

9.4 Implementation Time

In target detection applications, processing time becomes a major factor, especially when used in military settings. The correct detection of targets of interest in a timely and efficient manner is often critical to the success of a mission. One would expect that a large number of hyperspectral bands significantly adds to computation time. This in fact turns out to be the case; the inherent high-dimensionality of hyperspectral data does lead to a slight computational burden. However, it is the necessity in many of the applications above to compute and invert large covariance matrices that eventually is the main cause of the heavy processing load. In addition, the eigen-decomposition of large matrices adds to the processing times. In Table 9.12, relative average processing times (APT) per pixel are reported for each of the eight methods. The same results are plotted in Figure 9.25. Average processing times are computed using all measured times from all pixels in the two HYDICE hyperspectral data sets. Times have been normalized to the APT of the RX-algorithm since this is the benchmark anomaly detection algorithm. All of the algorithms are implemented using MATLAB. The code has not been optimized. For comparison purposes, the exact times are not needed - only the relative processing times. (In addition, due to certain restrictions, exact processing times cannot be reported here).

From the chart, it is clear that each of the four kernel-based algorithms is considerably computationally heavier than their respective linear methods. This can be easily attributed to the larger matrix inversions and eigen-decompositions

Method	Relative Avg. Time
RX	1
PCA	0.552
FLD	4.174
EST	1.544
KRX	7.750
KPCA	14.008
KFD	30.203
KEST	17.851

Table 9.12: Relative implementation times per pixel for all eight methods on the HYDICE images. Each spectral vector is of length 150. All times are relative to the average processing time for the RX-algorithm.

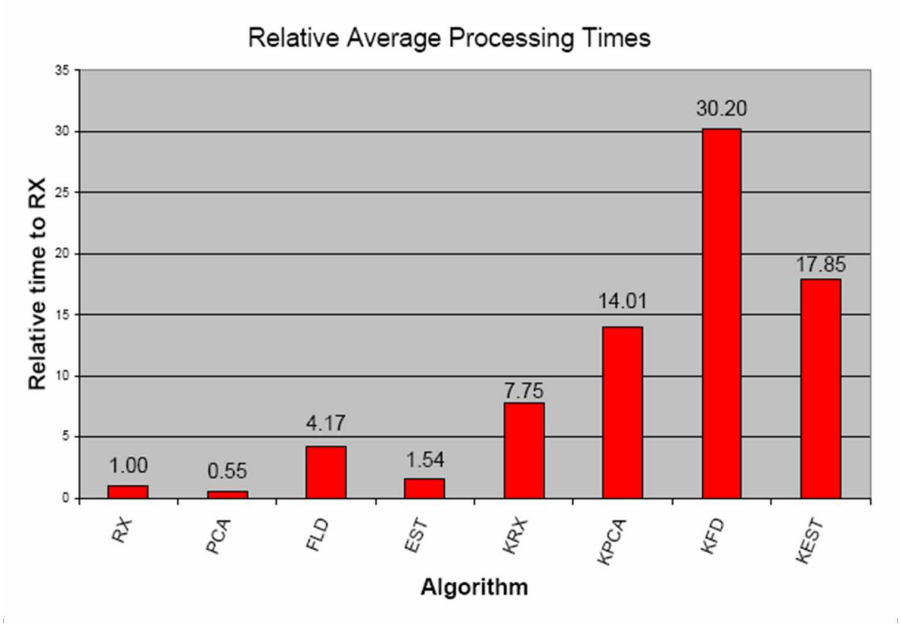


Figure 9.25: Relative implementation times per pixel for all eight methods on the HYDICE images. Each spectral vector is of length 150. All times are relative to the average processing time for the RX-algorithm.

that are necessary in these algorithms.

The most computationally intensive algorithm is the KFD algorithm. It takes about 30 times longer per pixel than the RX-algorithm. Additionally, the FLD

method is the slowest linear technique. These observations are the result of both a large scale inversion *in addition to* an eigen-decomposition. It is the only method that requires both processes. Comparing each of the nonlinear algorithms with only its linear counterpart, the data show that the KPCA algorithm has the largest increase in APT, taking about 25 times longer than PCA. The KRX algorithm takes about 8 times as long as the RX-algorithm while the KFD requires a little more than 7 times the processing time than FLD. Finally, KEST takes about 11.5 times longer to run than EST. If the image being processed is relatively large, these computation time increases can become very significant. In fact, they can mean the difference between running in a few minutes to taking many hours to process one image.

It should be noted that the above results are for data that has 150 spectral bands. The question then becomes whether increasing or decreasing the number of spectral bands alters the implementation time performance. It turns out that altering the number of spectral bands only affects the computation time of the linear methods. Decreasing the number of bands (to 70 as in the case of the AHI data sets) decreases the run time of the linear methods but has no effect on the run-time of the nonlinear methods. This is due to the fact that a nonlinear method implementation time is dependent on, as mentioned above, inverting and/or eigen-decomposing a large Gram matrix. The size of this Gram matrix is not dependent on the length of the spectral vector; rather, it depends on the size of the IWR and OWR. Therefore, since the dual window size remains constant throughout the experiments, decreasing the number of spectral bands in a data cube will decrease the run-time of the linear methods but will have no affect on the nonlinear methods. To reduce the run-time of

the nonlinear methods, a k-means clustering algorithm could be used to reduce the number of background samples. However, no such method was used in this thesis.

From these results, it is clear that while for the most part the nonlinear methods provide better anomaly detection than their linear counterparts, they are much more computationally intensive. In applications where real time detection results are required, a kernel method will have a difficult time meeting these specifications without further software optimizations.

Chapter 10

Conclusion

This thesis provided a performance characterization of linear and nonlinear subspace-based anomaly detection algorithms for hyperspectral imaging. Initially, three linear algorithms were used to generate projection vectors onto which samples from the inner window region and outer window region of a dual window centered at the test pixel were projected. In addition, the popular RX-anomaly detector was implemented. Each of these algorithms was then mapped into a high-dimensional feature space in an attempt to exploit the higher-order correlation between the spectral characteristics of the pixels. The nonlinear algorithms in the feature space needed to be rewritten in terms of kernels - otherwise, the extremely high-dimensionality made computation impractical. With each nonlinear algorithm formulated in terms of kernel functions of the data in the original input space, eight anomaly detection algorithms were briefly explained and implemented using five hyperspectral data cubes containing a varying number of 'targets of interest'. Performance comparisons were made using ROC curve and area under the curve (AUC) analysis.

In addition to a performance comparison, this thesis also provided a time comparison of the methods. Using the same machine and processing power for all algorithms, results indicate that the kernel methods require significantly more time to completely process each image. This added computational cost varied by

algorithm and was caused by the need to invert and eigen-decompose larger matrices.

Results from the five images show a generally improved performance for the nonlinear algorithms compared with their corresponding linear algorithms. The best performing algorithm, while always a kernel-based method, proved to be image dependent. Overall, however, KPCA and KFD showed the highest average detection rates over all five hyperspectral images. This superior detection abilities come with a high computational cost. In addition, with a large number of configurable parameters for some of the methods and no way to analytically optimize these parameters, it becomes difficult to compare the algorithms against each other with a high degree of certainty. Further research should examine the task of optimizing the parameters used in these algorithms (i.e. - kernel parameter, number of eigenvectors used for projection vectors, dual window size, etc.). In addition, more hyperspectral imagery could being tested in order to formulate a more accurate comparison of the algorithms. Finally, performances of these nonlinear methods should be compared with other common nonlinear anomaly detectors such as the support vector machine (SVM) novelty detector.

Chapter A

Centering a Matrix in Feature Space

In the kernel methods described in Chapters 6 and 7, it is occasionally assumed that the data are centered in the feature space. However, since there is no specific knowledge of the actual mapping Φ, there is no way to explicitly center the data in the highly-dimensional $\mathcal{F}$. To circumvent this problem, the kernel (Gram) matrix $\mathbf{K}$ must first be calculated using the uncentered input data. (Exactly how the Gram matrix is calculated is detailed in each section where appropriate). At that point, a technique shown in [28] provides a centered Gram matrix $\hat{\mathbf{K}}$ using the uncentered Gram matrix $\mathbf{K}$. This is accomplished for any general $M \times M$ Gram matrix using the relationship

$$\hat{\mathbf{K}} = (\mathbf{K} - \mathbf{1}_M\mathbf{K} - \mathbf{K}\mathbf{1}_M + \mathbf{1}_M\mathbf{K}\mathbf{1}_M) \tag{A.1}$$

where $(\mathbf{1}_M)$ is an $M \times M$ with each element equal to $1/M$.

Bibliography

[1] P. Shippert, "Why use hyperspectral imagery," *Photogrammetric Engineering and Remote Sensing*, pp. 377–380, Apr. 2004.

[2] D. W. J. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 58–69, Jan. 2002.

[3] H. Kwon, N. Nasrabadi, and S. Der, "Adaptive anomaly detection using subspace separation for hyperspectral imagery," *Optical Engineering*, vol. 42, no. 11, pp. 3342–3351, Nov. 2003.

[4] A. Banerjee, P. Burlina, and C. Diehl, "A support vector method for anomaly detectino in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 44, no. 8, pp. 2282–2291, Aug. 2006.

[5] K. I. Ranney and M. Soumekh, "Hyperspectral anomaly detection within the signal subspace," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 3, pp. 312–316, July 2006.

[6] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Processing Magazing*, vol. 19, no. 1, pp. 29–43, Jan. 2002.

[7] L. L. Scharf and B. Friedlander, "Matched subspace detectors," *IEEE Trans. on Signal Processing*, vol. 42, no. 8, pp. 2146–2157, Aug. 1994.

[8] D. Manolakis, D. Marden, and G. A. Shaw, "Hyperspectral image processing for automatic target detection applications," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 79–116, 2003.

[9] H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 43, no. 2, pp. 388–397, Feb. 2005.

[10] J. C. Harsanyi and C.-I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subpsace projection approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 4, pp. 779–785, July 1994.

[11] H. Kwon and N. Nasrabadi, "Kernel-based subpixel target detection in hyperspectral images," in *Proc. of IEEE Joint Conference on Neural Networks*, Budapest, Hungary, July 2004, pp. 717–722.

[12] ——, "Kernel orthogonal subspace projection for hyperspectral signal classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 12, pp. 2952–2962, Dec. 2005.

[13] ——, "Kernel spectral matched filter for hyperspectral target detection," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Philadelphia, PA, March 2005.

[14] C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. New York: Kluwer Academics/Plenum Publishers, 2003.

[15] C.-I. Chang and S.-S. Chiang, "Anomaly detection and classification for hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 40, no. 6, pp. 1314–1325, June 2002.

[16] E. A. Ashton and A. Schaum, "Algorithms for the detection of subpixel targets in multispectral imagery," *Photogram. Eng. Remote Sensing*, pp. 723–731, July 1998.

[17] R. N. Clark and et. al, "The `U.S. Geological Survey Digital Spectral Library Open File Report` 03-395," 2003, http://speclab.cr.usgs.gov/spectral-lib.html.

[18] D. Stein, J. Schoonmaker, and E. Coolbaugh, "Hyperspectral imaging for intelligence, surveillance, and reconnaissance," Space and Naval Warfare Systems Command, Sand Diego, CA, Tech. Rep., Aug. 2001.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: John Wiley & Sons, Inc., 2001.

[20] I. T. Joliffe, *Principal Component Analysis*. Berlin, Germany: Springer-Verlag, 1986.

[21] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing*, 1999, pp. 41–48.

[22] D. Torrieri, "The eigenspace transform for neural network classifiers," *Neural Networks*, vol. 12, no. 3, pp. 419–427, Apr. 1999.

[23] L. A. Chan, N. M. Nasrabadi, and D. Torrieri, "Eigenspace transform for automatic clutter rejection," *Optical Engineering*, pp. 564–573, 2001.

[24] B. Scholkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[25] B. E. Boser, I. M. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed. ACM Press, July 1992, pp. 144–152.

[26] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," *Philosophical Transactions of the Royal Society, London*, pp. A 209:415–446, 1909.

[27] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, UK: Cambridge University Press, 2004.

[28] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Kernel principal component analysis," *Neural Computation*, no. 10, pp. 1299–1319, 1999.

[29] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller, "Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces," *IEEE Trans. on Pattern Anal. and Machine Intelligence*, vol. 25, no. 5, pp. 623–628, May 2003.

[30] I. S. Reed and X. Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Oct. 1990.

[31] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[32] C. Marzban, "A comment on the roc curve and the aread under it as performance measures," June 2004, *http://www.nhn.ou.edu/˜marzban/ROC.pdf*.