Kernel-Based Detection Techniques for Hyperspectral Imagery

Nasser. M. Nasrabadi

Research Laboratory, ATTN: RDRL-SES-E
2800 Powder Mill Rd., Adelphi, MD 20783

ABSTRACT

In this paper we implement various linear and nonlinear subspace-based anomaly detectors for hyperspectral imagery. First, a dual window technique is used to separate the local area around each pixel into two regions - an inner-window region (IWR) and an outer-window region (OWR). Pixel spectra from each region are projected onto a subspace which is defined by projection bases that can be generated in several ways. Here we use three common pattern classification techniques (Principal Component Analysis (PCA), Fisher Linear Discriminant (FLD) Analysis, and the Eigenspace Separation Transform (EST)) to generate projection vectors. In addition to these three algorithms, the well-known Reed-Xiaoli (RX) anomaly detector is also implemented. Each of the four linear methods is then implicitly defined in a high- (possibly infinite-) dimensional feature space by using a nonlinear mapping associated with a kernel function. Using a common machine-learning technique known as the *kernel trick* all dot products in the feature space are replaced with a Mercer kernel function defined in terms of the original input data space. To determine how anomalous a given pixel is, we then project the current test pixel spectra and the spectral mean vector of the OWR onto the linear and nonlinear projection vecotrs in order to exploit the statistical differences between the IWR and OWR pixels. Anomalies are detected if the separation of the projection of the current test pixel spectra and the OWR mean spectra are greater than a certain threshold. Comparisons are made using receiver operating characteristics (ROC) curves.

Keywords: Anomaly detection, hyperspectral imagery, Eigenspace Separation Transform (EST), kernel-based machine learning, kernel PCA, kernel Fisher discriminant, kernels.

1. INTRODUCTION

Anomaly detectors are pattern recognition schemes that are used to detect objects that might be of military interest. Almost all anomaly detectors attempt to locate anything that looks different spatially or spectrally from its surroundings using a dual rectangular window approach.[1] In spectral anomaly detection algorithms, pixels (materials) that have a significantly different spectral signature from their neighboring background clutter pixels are identified as spectral anomalies. Spectral anomaly detection algorithms[1–8] could also use spectral signatures to detect anomalies embedded within background clutter with a very low signal-to-noise ratio. In spectral anomaly detectors, no prior knowledge of the target spectral signature is utilized or assumed.

One way of designing an anomaly detector is by projecting the input spectra onto a subspace whose bases are defined by some projection vectors. In[1] researchers compared subspace-based anomaly detection algorithms using projection vectors which were generated using three common pattern recognition techniques - Principal Component Analysis (PCA),[9] Fisher Linear Discriminant (FLD) Analysis,[10] and the Eigenspace Separation Transform (EST).[11] In addition, they compared the performance of a standard anomaly detection algorithm, the so called Reed-Xiaoli (RX) detector,[5] with the performances of their subspace-based anomaly detectors.

In many situations, however, a linear classifier is not always sufficient; that is, most real-world data are not linearly separable. Furthermore, most data do not fit the Gaussian distribution assumption made by the RX algorithm. However, by using a nonlinear mapping to transform each spectrum into a high-dimensional (possibly infinite-dimensional) feature space we can potentially exploit higher order correlation between the spectral bands, something that is not possible in the linear anomaly detectors. The resulting linear hyperplane separating the anomalies from the background in the high-dimensional feature space corresponds to a nonlinear boundary in the original input space. Unfortunately, it is computationally infeasible to carry out any algorithms in this high dimensional feature space. However, this problem can be circumvented by using a common machine learning

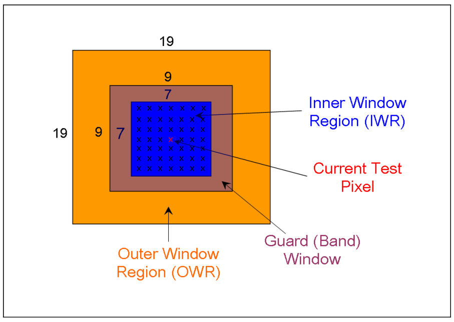


Figure 1. An example of a dual window with guard band. The numbers represent the length in pixels making up the side of each window. Each 'x' in the IWR represents one pixel. The pixel in red represents the current test pixel. The figure is not necessarily drawn to scale.

technique known as *kernelization.* By kernelizing the algorithm, all dot products between mapped vectors in the feature space are instead computed using a predetermined kernel function on the input data. Moreover, this technique will significantly simplify the mathematical computation.

This paper also examines the performance of the kernel versions of each of the four methods (PCA, FLD, EST and RX) described above and applies each one to the anomaly detection problem. More specifically, Kernel-RX (KRX),[6] Kernel Principal Component Analysis (KPCA),[12] Kernel Fisher Discriminant (KFD),[13] and Kernel Eigenspace Separation Transform (KEST), which is introduced in this paper, are all implemented and their performances are compared against each other as well as against their linear counterparts.

This paper is structured in the following manner. In Section 2 an introduction to subspace-based anomaly detection can be found and brief descriptions of the three linear methods which are used to generate the projection vectors are given. Section 3 contains a brief introduction to kernel-based learning techniques while Section 4 extends each of the three linear methods from Section 2 into their respective nonlinear kernelized methods. The RX and Kernel-RX Algorithms are presented in Section 5. Results and analysis of all four methods and their kernel version as applied to simulated data as well as multiple hyperspectral data sets can be found in Section 6. Finally, concluding remarks are made in Section 7.

2. LINEAR SUBSPACE-BASED ANOMALY DETECTION

One common method used in many anomaly detection algorithms is the dual-window approach Its use is predicated on the fact that it exploits 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 the inner window 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 (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 model.[8] In subspace-based anomaly detection techniques projection (basis) vectors are generated using the statistical properties of the IWR and OWR covariance matrices.

Using the eigen-value decomposition of the covariance matrices of IWR and OWR spectra it is possible to generate basis vectors for a subspace onto which vectors from the IWR and OWR are projected for discrimination. Denote a spectral vector within the IWR of a dual window centered at a 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 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_{in}}]$ is of size $J \times N_{in}$ and contains the spectra of each one of these samples as one of its N_{in} columns. 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 $\mathbf{Y} = \left[\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{N_{out}}\right]$ is one whose columns are the spectral vectors of the pixels in the OWR representing the background clutter samples. The background clutter statistics are estimated using the spectra of the pixels in the OWR. The covariance matrices of the IWR and OWR spectra are given by

$$\begin{aligned} \mathbf{C}_X &= \frac{1}{N_{in}-1}\left(\mathbf{X}-\hat{\boldsymbol{\mu}}_X\right)\left(\mathbf{X}-\hat{\boldsymbol{\mu}}_X\right)^T && (1) \\ \mathbf{C}_Y &= \frac{1}{N_{out}-1}\left(\mathbf{Y}-\hat{\boldsymbol{\mu}}_Y\right)\left(\mathbf{Y}-\hat{\boldsymbol{\mu}}_Y\right)^T. && (2) \end{aligned}$$

where $\hat{\boldsymbol{\mu}}_X$ and $\hat{\boldsymbol{\mu}}_Y$ are 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 projection separation statistic for a input test pixel denoted by $\mathbf{r}$ is calculated using

$$s' = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T \mathbf{W}\mathbf{W}^T (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{3}$$

where $\mathbf{W} = [\mathbf{w}_1\, \mathbf{w}_2\, \ldots\, \mathbf{w}_m]$ is a matrix whose columns are m projection vectors. The product $\mathbf{W}\mathbf{W}^T$ is known as a projection operator and represents a subspace characterizing the spectra used to generate the projection vectors $\mathbf{w}_i$. An anomaly is detected if the projection separation, s', is greater than some threshold, η. It is also possible to project the difference $(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})$ onto the complement subspace $(\mathbf{I} - \mathbf{W}\mathbf{W}^T)$ given by

$$s' = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T (\mathbf{I} - \mathbf{W}\mathbf{W}^T)(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{4}$$

Equation (4) is only used in some algorithms (e.g. - PCA and EST). In the experimental results section, only the best results between Equations (3) and (4) are reported and mention will be made regarding which equation was used. In the following subsections three different methods are used to generate the projection vectors, $\mathbf{W}$, in order to obtain the projection operator in the Equations (3) and (4) .

2.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the most commonly used methods for feature extraction and dimensionality reduction. The underlying goal is to find a projection which best represents some input data in the least-squared sense.[9] In order to generate the projection vectors, $\mathbf{w}_i$, the background clutter covariance matrix, $\mathbf{C}_{\mathbf{Y}}$, is written in terms of its eigenvectors $\mathbf{V}$ and their corresponding eigenvalues $\boldsymbol{\Lambda}$ as

$$\mathbf{C}_{\mathbf{Y}} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T. \tag{5}$$

the first m eigenvectors with the highest corresponding eigenvalues form the projection vectors. Thus,

$$\mathbf{W}_{PCA} = \tilde{\mathbf{V}} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m] \tag{6}$$

where m is a configurable constant. Altering the value of m (i.e. - changing the number of eigenvectors used) will change the performance of the anomaly detector as shown by experimental results.

Using Equation (6) as the projection vectors and substituting this result into Equation (3) or (4) we obtain the corresponding projections of the input onto the subspace and the complement subspace. The PCA anomaly detector is given by

$$\mathbf{PCA}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T \left(\mathbf{W}_{PCA}\mathbf{W}_{PCA}^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{7}$$

$$\mathbf{PCA}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T \left(\mathbf{I} - \mathbf{W}_{PCA}\mathbf{W}_{PCA}^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{8}$$

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, $\hat{\boldsymbol{\mu}}_{\mathbf{Y}}$, should ideally be large if the dual window is centered on an anomalous target.

The algorithm outlined above can also be developed using samples collected from the IWR. In this case, Equations (7) and (8) remain the same with the exception that the projection vectors $\mathbf{w}_i$ are generated using spectral information contained in the IWR rather than the OWR. In this paper, Equations (7) and (8) are referred to as the 'PCA Algorithm' or simply 'PCA'. In Section 6, only the best result among the four possible choices for PCA is given.

2.2 Fisher Linear Discriminant Analysis

Although PCA has proven to be very useful for efficient representation of data, it does not exploit the information in the IWR and OWR at the same time in order to generate the target or background subspaces. However, Fisher Linear Discriminant (FLD) Analysis,[10] which attempts to seek an optimal direction for discriminating between IWR and OWR data samples, does. First, the between-class scatter matrix is defined as $\mathbf{S}_B = (\hat{\boldsymbol{\mu}}_X - \hat{\boldsymbol{\mu}}_Y)(\hat{\boldsymbol{\mu}}_X - \hat{\boldsymbol{\mu}}_Y)^T$ while the within-class scatter matrix can be written as $\mathbf{S}_W = \mathbf{C}_X + \mathbf{C}_Y$ where $\mathbf{C}_X$ and $\mathbf{C}_Y$ are the covariance matrices of the samples in the IWR and OWR defined by Equations (1) and (2), 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.

In order to calculate the optimal discrimination direction, $\mathbf{w}^*$, the criterion function

$$\mathbf{w}^* = \max_{\mathbf{w}} \mathrm{J}(\mathbf{w}) = \frac{|\mathbf{w}^T \mathbf{S}_B \mathbf{w}|}{|\mathbf{w}^T \mathbf{S}_W \mathbf{w}|} \tag{9}$$

needs to be maximized over all possible $\mathbf{w}$ and has been shown[10] to be given by

$$\mathbf{w_F} = \mathbf{w}^* = \mathbf{S}_W^{-1}(\boldsymbol{\mu}_{in} - \boldsymbol{\mu}_{out}) \tag{10}$$

Using Equation (10) as the projection vector and substituting this result into Equation (3) gives

$$\mathbf{FLD}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)^T \left(\mathbf{w}_F \mathbf{w}_F^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_Y). \tag{11}$$

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 paper, Equation (11) is referred to as the 'FLD Algorithm' or simply 'FLD'.

2.3 Eigenspace Separation Transform

The Eigenspace Separation Transform (EST) was developed by Torrieri[11] as a preprocessing technique to extract features for neural network classifiers and has been successfully used by researchers for automatic clutter rejection.[14] Like PCA, EST aims to extract features from a training set by projecting the input patterns onto a lower-dimensional orthogonal subspace. In this paper it is used to generate projection vectors in order to separate target pixels from background clutter.

In EST algorithm we first compute the $J \times J$ Difference Correlation (DCOR) matrix

$$\hat{\mathbf{M}} \quad = \quad \mathbf{R}_X - \mathbf{R}_Y = \frac{1}{N_{in}}\mathbf{X}\mathbf{X}^T - \frac{1}{N_{out}}\mathbf{Y}\mathbf{Y}^T \tag{12}$$

where $\hat{\mathbf{M}}$ is simply the difference of 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[1] .

The eigenvalue decomposition of DCOR can be rewritten in block-matrix form in terms of its positive and negative eigenvalues and eigenvectors as

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{V}_+ & \mathbf{V}_- \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_+ & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_- \end{bmatrix} \begin{bmatrix} \mathbf{V}_+^T \\ \mathbf{V}_-^T \end{bmatrix} \tag{13}$$

where the columns of $\mathbf{V}_+$ and $\mathbf{V}_-$ are the eigenvectors with their corresponding non-zero positive ($\boldsymbol{\Lambda}_+$) and negative ($\boldsymbol{\Lambda}_-$) eigenvalues, respectively.

The matrix $\mathbf{W}_{\text{EST}}$ is then chosen to be the set of m positive or negative eigenvectors associated with $\hat{\mathbf{M}}$. The choice of which set to use hinges on which set of eigenvalues (positive or negative) has the largest absolute sum. Thus, the EST projection vectors are given as

$$\mathbf{W}_{\text{EST}} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m] \tag{14}$$

where $\mathbf{v}_i$ $(i = 1, \ldots, m)$ are the m most significant (either positive or negative) eigenvectors or $\hat{\mathbf{M}}$.

Using Equation (14) as the projection vectors and substituting this result into Equation (3) gives

$$\mathbf{EST}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)^T \left(\mathbf{W}_{EST}\mathbf{W}_{EST}^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_Y). \tag{15}$$

It is also possible to project onto the complement subspace, $\left(\mathbf{I} - \mathbf{W}_{EST}\mathbf{W}_{EST}^T\right)$. Thus, substituting Equation (14) into Equation (4) yields

$$\mathbf{EST}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}})^T \left(\mathbf{I} - \mathbf{W}_{EST}\mathbf{W}_{EST}^T\right)(\mathbf{r} - \hat{\boldsymbol{\mu}}_{\mathbf{Y}}). \tag{16}$$

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 paper, Equations (15) and (16) are referred to as the 'EST Algorithm' or simply 'EST'. In the experimental results in Section 6, only the best results among the four possible choices of EST are presented.

3. KERNEL LEARNING THEORY

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,

$$\begin{aligned} \Phi &: \chi \to \mathcal{F} \\ \mathbf{x} &\mapsto \Phi(\mathbf{x}). \end{aligned} \tag{17}$$

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

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle. \tag{18}$$

Equation (18), which is commonly referred to in machine learning literature as the *kernel trick*,[15] 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 (18) illustrates that all dot products in $\mathcal{F}$ can be replaced by an appropriately chosen *Mercer* kernel function k.[16] For a more comprehensive discussion about the properties of various types of kernels and for more information on kernel-based learning in general, see one of the many references devoted to kernel methods.[15,16]

4. KERNEL SUBSPACE-BASED ANOMALY DETECTION

In this Section, each of the linear subspace-based methods in Sections 2.1-2.3 is extended into the feature space $\mathcal{F}$ and then kernelized by replacing all dot products in the feature space by kernel functions using the kernel trick in Equation (18). The following subsections present a derivation of each of the kernelized algorithms. Using a nonlinear mapping $\boldsymbol{\Phi}$, the original data in the input space defined by $\mathbf{X}$ and $\mathbf{Y}$ are mapped into the feature space $\mathcal{F}$ and denoted by

$$\begin{aligned} \mathbf{X}_\Phi &= \Phi(\mathbf{X}) = [\Phi(\mathbf{x}_1)\,\Phi(\mathbf{x}_2)\,\ldots\,\Phi(\mathbf{x}_{N_{in}})] \qquad (19) \\ \mathbf{Y}_\Phi &= \Phi(\mathbf{Y}) = \left[\Phi(\mathbf{y}_1)\,\Phi(\mathbf{y}_2)\,\ldots\,\Phi(\mathbf{y}_{N_{out}})\right] \qquad (20) \end{aligned}$$

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 represented by $\hat{\boldsymbol{\mu}}_{X_\Phi}$ and $\hat{\boldsymbol{\mu}}_{Y_\Phi}$, respectively. For many of the following methods, it is assumed that the mapped data is centered in $\mathcal{F}$. Thus, denote each centered vector for the IWR in $\mathcal{F}$ as $\Phi_c(\mathbf{x}_i) = \Phi(\mathbf{x}_i) - \hat{\boldsymbol{\mu}}_{X_\Phi}$, $i = 1, \ldots, N_{in}$ and similarly for the OWR spectra (i.e. - $\Phi_c(\mathbf{y}_j) = \Phi(\mathbf{y}_j) - \hat{\boldsymbol{\mu}}_{Y_\Phi}$, $j = 1, \ldots, N_{out}$). Then, let $\mathbf{X}_{c\Phi}$ and $\mathbf{Y}_{c\Phi}$ be matrices whose columns are the centered IWR and OWR in the feature space, respectively. Also, let $\mathbf{C}_{X_\Phi}$ and $\mathbf{C}_{Y_\Phi}$ be the covariance matrices of the centered spectra in the feature space. The projection of the mapped test pixel spectra $\Phi(\mathbf{r})$ onto a linear subspace in the feature space which is equivalent to a nonlinear subspace in the original input domain is given by

$$s' = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \mathbf{W}_\Phi \mathbf{W}_\Phi^T (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}) \tag{21}$$

where $\mathbf{W}_\Phi = \left[\mathbf{w}_\Phi^1 \, \mathbf{w}_\Phi^1 \, \ldots \, \mathbf{w}_\Phi^m\right]$ is a matrix whose columns are the set of m projection vectors in $\mathcal{F}$. Similarly, projection onto the complement subspace is given by

$$s' = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \left(\mathbf{I}_\Phi - \mathbf{W}_\Phi \mathbf{W}_\Phi^T\right) (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}) \tag{22}$$

The following subsections outline three methods that can provide the projection vectors $\mathbf{W}_\Phi = \left[\mathbf{w}_\Phi^1 \, \mathbf{w}_\Phi^1 \, \ldots \, \mathbf{w}_\Phi^m\right]$ in order to perform nonlinear anomaly detection using a nonlinear subspace. The methods employed are simply nonlinear extensions of each of the three algorithms detailed in Section 2.

4.1 Kernel Principal Component Analysis

In this section, the PCA 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 to make the problem computationally feasible. As in the linear PCA algorithm, KPCA can be formulated using either the IWR or OWR spectra to formulate the KPCA projection vectors in the feature space.

In order to find the PCA eigenvectors in the feature space, simply solve the mapped version of Equation (5); specifically, the eigenvalues and eigenvectors of $\mathbf{C}_{\mathbf{Y}_\Phi}$ in $\mathcal{F}$ can be found by solving

$$\mathbf{C}_{\mathbf{Y}_\Phi} = \mathbf{V}_\Phi \boldsymbol{\Lambda}_\Phi \mathbf{V}_\Phi^T \tag{23}$$

where $\boldsymbol{\Lambda}_\Phi = diag\left(\lambda_\Phi^1 \, \lambda_\Phi^2 \, \ldots \, \lambda_\Phi^p\right)$ and $\mathbf{V}_\Phi = \left[\mathbf{v}_\Phi^1 \, \mathbf{v}_\Phi^2 \, \ldots \, \mathbf{v}_\Phi^p\right]$ contain only the p nonzero eigenvalues and corresponding eigenvectors of $\mathbf{C}_{\mathbf{Y}_\Phi}$. All eigenvectors in the feature space lie in the span of the vectors in $\mathbf{Y}_{c\Phi}$. Therefore, they can be represented as

$$\mathbf{V}_\Phi \quad = \quad {\boldsymbol{\Lambda}_\Phi}^{-1/2} \mathbf{Y}_{c\Phi} \mathbf{A} \tag{24}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1 \, \boldsymbol{\alpha}_2 \cdots \boldsymbol{\alpha}_{N_{out}}]$ is a matrix whose columns are the nonzero eigenvectors of the centered Gram (kernel) matrix $\mathbf{K}_c$ and $\boldsymbol{\Lambda}_\Phi$ contains the associated nonzero eigenvalues. The centered kernel matrix can be calculated by $\mathbf{K}_c = (\mathbf{K} - \mathbf{1}_{N_{out}}\mathbf{K} - \mathbf{K}\mathbf{1}_{N_{out}} + \mathbf{1}_{N_{out}}\mathbf{K}\mathbf{1}_{N_{out}})$ where $\mathbf{K}$ is the kernel matrix whose elements are $(\mathbf{K})_{ij} = k(\mathbf{y}_i, \mathbf{y}_j)$ with $i, j = 1, \ldots, N_{out}$ and $(\mathbf{1}_{N_{out}})$ is an $N_{out} \times N_{out}$ with each element equal to $1/N_{out}$. It is known[12] that the eigenvalue decomposition of the centered kernel matrix is given by

$$\mathbf{K}_c = \mathbf{A} \boldsymbol{\Lambda}_\Phi \mathbf{A}^T \tag{25}$$

Utilizing only the m most significant eigenvectors in the features space they can be written as

$$\mathbf{W}_{PCA} = \tilde{\mathbf{V}}_\Phi = \mathbf{Y}_{c\Phi} \tilde{\mathbf{A}} \tag{26}$$

where $\tilde{\mathbf{A}} = [\boldsymbol{\alpha}_1 \, \boldsymbol{\alpha}_2 \cdots \boldsymbol{\alpha}_m]$ are the m most significant eigenvectors of $\mathbf{K}_c$ normalized by the square roots of their respective eigenvalues. The vectors $\tilde{\mathbf{V}}_\Phi$ are then used as projection vectors in the feature space ($\mathbf{W}_\Phi$) in Equation (21). Substituting Equation (26) into Equation (21) gives

$$\begin{aligned} \mathbf{KPCA}(\mathbf{r}) &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}))^T \left(\tilde{\mathbf{V}}_\Phi \tilde{\mathbf{V}}_\Phi^T\right) (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \\ &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \mathbf{Y}_{c\Phi} \tilde{\mathbf{A}}_{KPCA} \tilde{\mathbf{A}}_{KPCA}^T \mathbf{Y}_{c\Phi}^T (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \end{aligned} \tag{27}$$

For notational simplicity, let $\mathbf{K}_{\mathbf{Yr}}^T = \Phi(\mathbf{r})^T \mathbf{Y}_{c\Phi}$ and $\mathbf{K}_{\mathbf{Y}\hat{\boldsymbol{\mu}}}^T = \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}^T \mathbf{Y}_{c\Phi}$; these are commonly referred to as empirical kernel expansions. Substituting these results into Equation (27) results in

$$\mathbf{KPCA}(\mathbf{r}) = \left(\mathbf{K}_{\mathbf{Yr}}^T - \mathbf{K}_{\mathbf{Y}\hat{\boldsymbol{\mu}}}^T\right)^T \tilde{\mathbf{A}}_{KPCA} \tilde{\mathbf{A}}_{KPCA}^T \left(\mathbf{K}_{\mathbf{Yr}}^T - \mathbf{K}_{\mathbf{Y}\hat{\boldsymbol{\mu}}}^T\right). \tag{28}$$

It should be pointed out that as in the linear case, we can also project onto the complement subspace $(\mathbf{I} - \tilde{\mathbf{V}}_\Phi \tilde{\mathbf{V}}_\Phi^T)$ in the feature space. As mentioned above, the KPCA algorithm can be formulated using the IWR spectra rather than the OWR spectra to generate the projection vectors $\mathbf{W}_\Phi$ in the feature space.

4.2 Kernel Fisher Discriminant Analysis

It has been shown[13, 15] how to extend FLD analysis to its nonlinear version by using the kernel trick to compute the Fisher discriminant in the feature space. Defining FLD in the feature space is equivalent to maximizing the cost function given by

$$\mathrm{J}(\mathbf{w}_\Phi) = \frac{|\mathbf{w}_\Phi^T \mathbf{S}_B^\Phi \mathbf{w}_\Phi|}{|\mathbf{w}_\Phi^T \mathbf{S}_W^\Phi \mathbf{w}_\Phi|} \tag{29}$$

where $\mathbf{w}_\Phi$ is the projection vector, $\mathbf{S}_W^\Phi = \mathbf{C}_{X_\Phi} + \mathbf{C}_{Y_\Phi}$ and $\mathbf{S}_B^\Phi = (\hat{\boldsymbol{\mu}}_{X_\Phi} - \hat{\boldsymbol{\mu}}_{Y_\Phi})(\hat{\boldsymbol{\mu}}_{X_\Phi} - \hat{\boldsymbol{\mu}}_{Y_\Phi})^T$ are the within-class and between-class scatter matrices, respectively, in $\mathcal{F}$.

Finding an optimal $\mathbf{w}_\Phi$ by maximizing Equation (29) is not mathematically tractable considering the simple fact that the feature space is of high (possibly infinite) dimensionality. Fortunately, we can reformulate this problem in terms of dot products in the feature space and then replace them with kernel functions. Based on reproducing kernel theory, any solution $\mathbf{w}_\Phi$ to Equation (29) can be expanded as

$$\mathbf{w}_\Phi = \sum_{i=1}^{N_{TOT}} \alpha_i \Phi(\mathbf{z}_i) = \mathbf{Z}_\Phi \boldsymbol{\alpha} \tag{30}$$

where $N_{TOT} = N_{in} + N_{out}$ and $\mathbf{Z}_\Phi = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{N_{TOT}}] = \left[\Phi_c(\mathbf{x}_1), \ldots, \Phi_c(\mathbf{x}_{N_{in}}), \Phi_c(\mathbf{y}_1), \ldots, \Phi_c(\mathbf{y}_{N_{out}})\right]$ is a matrix whose columns are the mapped vectors in $\mathcal{F}$ of the corresponding spectra in both the IWR and OWR concatenated together and $\boldsymbol{\alpha}$ is the KFD vector in the feature space.

Combining the definition of $\hat{\boldsymbol{\mu}}_{\mathbf{X}_\Phi}$ and Equation (30) yields

$$\mathbf{w}_\Phi^T \hat{\boldsymbol{\mu}}_{\mathbf{X}_\Phi} = \boldsymbol{\alpha}^T \mathbf{M}_{in} \tag{31}$$

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 (30) gives

$$\mathbf{w}_\Phi^T \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi} = \boldsymbol{\alpha}^T \mathbf{M}_{out} \tag{32}$$

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 (Equation (18)).

By using the definition of $\mathbf{S}_B^\Phi$ and Equations (31) and (32), the numerator of Equation (29) can be written as

$$\mathbf{w}_\Phi^T \mathbf{S}_\mathbf{B}^\Phi \mathbf{w}_\Phi = \boldsymbol{\alpha}^T \mathbf{A} \boldsymbol{\alpha} \tag{33}$$

where $\mathbf{A} = (\mathbf{M}_{in} - \mathbf{M}_{out})(\mathbf{M}_{in} - \mathbf{M}_{out})^T$. Using a similar argument, the denominator of Equation (29) can be rewritten as

$$\mathbf{w}_\Phi^T \mathbf{S}_\mathbf{W}^\Phi \mathbf{w}_\Phi = \boldsymbol{\alpha}^T \mathbf{B} \boldsymbol{\alpha} \tag{34}$$

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 $\mathbf{K}_{out}$ are defined to be

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

A combination of Equations (33) and (34) means that Fisher's discriminant in $\mathcal{F}$ can now be found by maximizing

$$\mathrm{J}(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \mathbf{A} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{B} \boldsymbol{\alpha}}. \tag{35}$$

As in the solution to the analogous problem in the input space (Equation (9)), Equation (35) can be solved simply by finding the leading eigenvector, $\boldsymbol{\alpha}_{KFD}$, of $\mathbf{B}^{-1}\mathbf{A}$. Thus, the identity in Equation (30) becomes $\mathbf{w}_{\Phi} = \mathbf{w}_{F_{\Phi}} = \mathbf{Z}_{\Phi} \boldsymbol{\alpha}_{KFD}$.

Substituting this result into Equation (21), gives

$$\begin{aligned} \mathbf{KFD}(\mathbf{r}) &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_{\Phi}})^T \left(\mathbf{w}_{F_{\Phi}} \mathbf{w}_{F_{\Phi}}^T \right) (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_{\Phi}}). \\ &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_{\Phi}})^T \mathbf{Z}_{\Phi} \boldsymbol{\alpha}_{KFD} \boldsymbol{\alpha}_{KFD}^T \mathbf{Z}_{\Phi}^T (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_{\Phi}}). \end{aligned} \tag{36}$$

To simplify this equation, let

$$\Phi(\mathbf{r})^T \mathbf{Z}_{\Phi} = \Phi(\mathbf{r})^T \left[\Phi(\mathbf{x}_1)\ \Phi(\mathbf{x}_2)\ \ldots\ \ \Phi(\mathbf{x}_{N_{TOT}}) \right] = \mathbf{k}\left(\mathbf{Z}, \mathbf{r}\right)^T = \mathbf{K}_{\mathbf{Zr}} \tag{37}$$

and

$$\hat{\boldsymbol{\mu}}_{\mathbf{Y}_{\Phi}}^T \mathbf{Z}_{\Phi} = \frac{1}{N_{out}} \sum_{\mathbf{y} \in OWR} \Phi(\mathbf{y})^T \left[\Phi(\mathbf{z}_1)\, \Phi(\mathbf{z}_2)\ \ldots\ \Phi(\mathbf{z}_{N_{TOT}}) \right] = \frac{1}{N_{out}} \sum_{\mathbf{y} \in OWR} \mathbf{k}\left(\mathbf{Z}, \mathbf{y}\right)^T = \mathbf{K}_{\mathbf{Z}\boldsymbol{\mu}}. \tag{38}$$

Using Equations (37) and (38), Equation (36) becomes

$$\mathbf{KFD}(\mathbf{r}) = \left(\mathbf{K}_{\mathbf{Zr}}^T - \mathbf{K}_{\mathbf{Z}\boldsymbol{\mu}}^T \right)^T \boldsymbol{\alpha}_{KFD} \boldsymbol{\alpha}_{KFD}^T \left(\mathbf{K}_{\mathbf{Zr}}^T - \mathbf{K}_{\mathbf{Z}\boldsymbol{\mu}}^T \right) \tag{39}$$

Equation (39) is the equation used for the Kernel Fisher Discriminant algorithm in this paper.

4.3 Kernel Eigenspace Separation Transform

In this section, EST is defined in the feature space $\mathcal{F}$ and then reformulated solely in terms of dot products. Once again, the kernel trick is utilized to convert it into its kernel version. The difference correlation matrix (DCOR) $\mathbf{R}$ for the input data in the feature space can be written as

$$\begin{aligned} \mathbf{R}_{\Phi} &= \mathbf{R}_{\mathbf{X}_{\Phi}} - \mathbf{R}_{\mathbf{Y}_{\Phi}} = \frac{1}{N_{in}} \Phi(\mathbf{X}) \Phi(\mathbf{X})^T - \frac{1}{N_{out}} \Phi(\mathbf{Y}) \Phi(\mathbf{Y})^T \\ &= \begin{bmatrix} \Phi(\mathbf{X}) & -\Phi(\mathbf{Y}) \end{bmatrix} \begin{bmatrix} \Phi(\mathbf{X})^T / N_{in} \\ \Phi(\mathbf{Y})^T / N_{out} \end{bmatrix}. \end{aligned} \tag{40}$$

Here, the correlation matrix in the feature space of the first class 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 second class 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

$$\mathbf{R}_{\Phi} = \begin{bmatrix} \mathbf{V}_{+_{\Phi}} & \mathbf{V}_{-_{\Phi}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_{+_{\Phi}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{-_{\Phi}} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{+_{\Phi}}^T \\ \mathbf{V}_{-_{\Phi}}^T \end{bmatrix} \tag{41}$$

where the columns of $\mathbf{V}_{+_{\Phi}}$ and $\mathbf{V}_{-_{\Phi}}$ are the eigenvectors in the feature space with their corresponding non-zero positive ($\boldsymbol{\Lambda}_{+_{\Phi}}$) and negative ($\boldsymbol{\Lambda}_{-_{\Phi}}$) 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 $\boldsymbol{\Lambda}_{\Phi}$ which satisfy the equation

$$\boldsymbol{\Lambda}_{\Phi} \mathbf{V}_{\Phi} = \mathbf{R}_{\Phi} \mathbf{V}_{\Phi}. \tag{42}$$

Kernelization of EST in the Feature Space

Due to the (possibly) extreme high-dimensionality of the feature space, (42) 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

$$\begin{aligned}\mathbf{v}_\Phi^k &= \frac{1}{\sqrt{N_{in}}}\sum_{i=1}^{N_{in}} \alpha_i^k \Phi(\mathbf{x}_i)\lambda_i^{-\frac{1}{2}} - \frac{1}{\sqrt{N_{out}}}\sum_{j=1}^{N_{out}} \beta_j^k \Phi(\mathbf{y}_j)\lambda_j^{-\frac{1}{2}} \\ &= \frac{1}{\sqrt{N_{in}}}\Phi(\mathbf{X})\boldsymbol{\alpha}^k \boldsymbol{\Lambda}_+^{-\frac{1}{2}} - \frac{1}{\sqrt{N_{out}}}\Phi(\mathbf{Y})\boldsymbol{\beta}^k \boldsymbol{\Lambda}_-^{-\frac{1}{2}} \qquad (43)\end{aligned}$$

where the expansion coefficients, $\boldsymbol{\alpha}^k$ and $\boldsymbol{\beta}^k$, are defined as $\boldsymbol{\alpha}^k = (\alpha_1^k, \alpha_2^k, \dots, \alpha_{N_{in}}^k)^T$ and $\boldsymbol{\beta}^k = (\beta_1^k, \beta_2^k, \dots, \beta_{N_{out}}^k)^T$ for $k = 1, \dots, N_t$ where $N_t = N_{in} + N_{out}$. Equation (43) can be used to write all eigenvectors with non-zero eigenvalues as

$$\begin{aligned}\mathbf{V}_\Phi &= \begin{bmatrix}\mathbf{v}_\Phi^1 & \mathbf{v}_\Phi^2 & \cdots & \mathbf{v}_\Phi^{N_t}\end{bmatrix} \\ &= \Phi(\mathbf{X})\mathbf{A}\boldsymbol{\Lambda}_+^{-\frac{1}{2}} - \Phi(\mathbf{Y})\mathbf{B}\boldsymbol{\Lambda}_-^{-\frac{1}{2}} \\ &= \begin{bmatrix}\Phi(\mathbf{X}) & -\Phi(\mathbf{Y})\end{bmatrix}\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_+^{-\frac{1}{2}} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Lambda}_-^{-\frac{1}{2}}\end{bmatrix} \\ &= \begin{bmatrix}\Phi(\mathbf{X}) & -\Phi(\mathbf{Y})\end{bmatrix}\mathbf{D} \qquad (44)\end{aligned}$$

where

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

and the columns of

$$\mathbf{D} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}_+^{-\frac{1}{2}} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Lambda}_-^{-\frac{1}{2}}\end{bmatrix} = \begin{bmatrix}\frac{\boldsymbol{\alpha}^1}{\sqrt{N_{in}}} & \frac{\boldsymbol{\alpha}^2}{\sqrt{N_{in}}} & \cdots & \frac{\boldsymbol{\alpha}^{N_t}}{\sqrt{N_{in}}}\\ \frac{\boldsymbol{\beta}^1}{\sqrt{N_{out}}} & \frac{\boldsymbol{\beta}^2}{\sqrt{N_{out}}} & \cdots & \frac{\boldsymbol{\beta}^{N_t}}{\sqrt{N_{out}}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Lambda}\end{bmatrix}^{-\frac{1}{2}} \qquad (45)$$

represent the eigenvectors of a kernel matrix associated with the kernelized version of EST (shown below). By substituting equations (40) and (44) into (42) and using the kernel trick from Equation (18) to simplify we obtain

$$\boldsymbol{\Lambda}\big[\Phi(\mathbf{X}) \;\; -\Phi(\mathbf{Y})\big]\mathbf{D} = \big[\Phi(\mathbf{X}) \;\; -\Phi(\mathbf{Y})\big]\begin{bmatrix}\frac{\mathbf{K_{XX}}}{N_{in}} & -\frac{\mathbf{K_{XY}}}{N_{in}}\\ \frac{\mathbf{K_{YX}}}{N_{out}} & -\frac{\mathbf{K_{YY}}}{N_{out}}\end{bmatrix}\mathbf{D} \qquad (46)$$

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 (46) by $[\Phi(\mathbf{X}) \quad \Phi(\mathbf{Y})]^T$ and again using (18) to simplify produces

$$\boldsymbol{\Lambda}\begin{bmatrix}\frac{\mathbf{K_{XX}}}{N_{in}} & -\frac{\mathbf{K_{XY}}}{N_{in}}\\ \frac{\mathbf{K_{YX}}}{N_{out}} & -\frac{\mathbf{K_{YY}}}{N_{out}}\end{bmatrix}\mathbf{D} = \begin{bmatrix}\frac{\mathbf{K_{XX}}}{N_{in}} & -\frac{\mathbf{K_{XY}}}{N_{in}}\\ \frac{\mathbf{K_{YX}}}{N_{out}} & -\frac{\mathbf{K_{YY}}}{N_{out}}\end{bmatrix}^2\mathbf{D}. \qquad (47)$$

Solving equation (47) is tantamount to finding the eigenvectors and eigenvalues of the kernel matrix

$$\mathbf{K}_{KEST} = \begin{bmatrix} \frac{\mathbf{K_{XX}}}{N_{in}} & -\frac{\mathbf{K_{XY}}}{N_{in}} \\ \frac{\mathbf{K_{YX}}}{N_{out}} & -\frac{\mathbf{K_{YY}}}{N_{out}} \end{bmatrix} = \tilde{\mathbf{D}}\mathbf{\Lambda}\tilde{\mathbf{D}}^T. \tag{48}$$

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 (43). 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

$$\begin{aligned} \mathbf{W}_{KEST} = \mathbf{W}^{+}_{KEST} &= [\mathbf{d}_1\, \mathbf{d}_2 \cdots \mathbf{d}_m] \\ \mathbf{W}_{KEST} = \mathbf{W}^{-}_{KEST} &= [\mathbf{d}_{N_t}\, \mathbf{d}_{N_t-1} \cdots \mathbf{d}_{N_t-m+1}] \end{aligned} \tag{49}$$

where, as with KPCA, m is a configurable constant. 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 2.3.

Substituting Equation (49) for $\mathbf{D}$ in Equation (44) (i.e. - only using the first m positive or negative eigenvectors) and using this result as the projection vectors, $\mathbf{W}_\Phi$, in Equation (21) yields

$$\begin{aligned} \mathbf{KEST}(\mathbf{r}) &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \left(\mathbf{V}_\Phi \mathbf{V}_\Phi^T\right) (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}). \\ &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi})^T \Phi(\bar{\mathbf{Z}}) \mathbf{W}_{KEST} \mathbf{W}^T_{KEST} \Phi(\bar{\mathbf{Z}})^T \left(\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{Y}_\Phi}\right). \end{aligned} \tag{50}$$

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

$$\begin{aligned} \Phi(\mathbf{r})^T\Phi(\mathbf{Z}) &= \Phi(\mathbf{r})^T \left[\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_{N_{in}}), -\Phi(\mathbf{y}_1), \ldots, -\Phi(\mathbf{y}_{N_{out}})\right] \\ &= \left[k(\mathbf{x}_1, \mathbf{r}), \ldots, k(\mathbf{x}_{N_{in}}, \mathbf{r}), -k(\mathbf{y}_1, \mathbf{r}), \ldots, -k(\mathbf{y}_{N_{out}}, \mathbf{r})\right] \\ &= \mathbf{k}(\bar{\mathbf{Z}}, \mathbf{r})^T = \mathbf{K}_{\bar{\mathbf{Z}}\mathbf{r}} \end{aligned} \tag{51}$$

where the second equal sign is as a direct result of using the kernel trick in Equation (18). The vector $\mathbf{k}(\bar{\mathbf{Z}}, \mathbf{r})^T$ is commonly referred to as the empirical kernel map of an input vector $\mathbf{r}$.[15] Similarly, define

$$\begin{aligned} \hat{\boldsymbol{\mu}}^T_{\mathbf{Y}_\Phi}\Phi(\bar{\mathbf{Z}}) &= \frac{1}{N_{out}} \sum_{\mathbf{y}\in OWR} \Phi(\mathbf{y})^T \left[\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_{N_{in}}), -\Phi(\mathbf{y}_1), \ldots, -\Phi(\mathbf{y}_{N_{out}})\right] \\ &= \frac{1}{N_{out}} \sum_{\mathbf{y}\in OWR} \mathbf{k}\left(\bar{\mathbf{Z}}, \mathbf{y}\right)^T = \mathbf{K}_{\bar{\mathbf{Z}}\hat{\boldsymbol{\mu}}}. \end{aligned} \tag{52}$$

Using Equations (51) and (52), Equation (50) becomes

$$\mathbf{KEST}(\mathbf{r}) = \left(\mathbf{K}^T_{\bar{\mathbf{Z}}\mathbf{r}} - \mathbf{K}^T_{\bar{\mathbf{Z}}\hat{\boldsymbol{\mu}}}\right)^T \mathbf{W}_{KEST} \mathbf{W}^T_{KEST} \left(\mathbf{K}^T_{\bar{\mathbf{Z}}\mathbf{r}} - \mathbf{K}^T_{\bar{\mathbf{Z}}\hat{\boldsymbol{\mu}}}\right) \tag{53}$$

As in the case of linear EST in Section 2.3, as well as in KPCA, it is also possible to project onto the complement subspace $(\mathbf{I} - \mathbf{V}_\Phi \mathbf{V}^T_\Phi)$ in the feature space as an extension of Equation (4).

5. RX AND KERNEL-RX ANOMALY DETECTORS

The RX anomaly detector introduced by Reed and Yu[5] has become the benchmark for hyperspectral anomaly detection because of its natural assumption that neither the target spectrum nor the covariance matrix of the background clutter need to be known. The RX-algorithm is based on comparing the difference between the test spectrum and the spectra of the immediate background samples. It is similar to the Mahalanobis distance measure and is given by

$$\mathbf{RX}(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y)^T \mathbf{C}_Y^{-1} (\mathbf{r} - \hat{\boldsymbol{\mu}}_Y). \tag{54}$$

where $\mathbf{r}$ is the test sample, $\hat{\boldsymbol{\mu}}_Y$ and $\mathbf{C}_Y$ are the spectral mean and covariance of the background clutter samples in the OWR. Similarly, the RX-algorithm can be defined in the feature space as

$$\mathbf{RX}(\Phi(\mathbf{r})) = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{Y_\Phi})^T \hat{\mathbf{C}}_{b_\Phi}^{-1} (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{Y_\Phi}) \tag{55}$$

where $\hat{\mathbf{C}}_{Y_\Phi}$ is the estimated covariance matrix of the background clutter and $\hat{\boldsymbol{\mu}}_{Y_\Phi}$ is the mean of the background clutter samples in the feature space. Equation (55) corresponds to a linear detector in the feature space; however, it corresponds to a nonlinear detector in the original input space. Unfortunately, Equation (55) cannot be directly implemented because of the high dimensionality of the feature space. The kernel version of the RX algorithm was obtained in[6] and is given as

$$\mathbf{KRX}(\mathbf{r}) = \left(\mathbf{K}_{\mathbf{Yr}}^T - \mathbf{K}_{\hat{\boldsymbol{\mu}}_Y}^T\right)^T \hat{\mathbf{K}}_{\mathbf{Y}}^{-1} \left(\mathbf{K}_{\mathbf{Yr}}^T - \mathbf{K}_{\hat{\boldsymbol{\mu}}_Y}^T\right). \tag{56}$$

where $\mathbf{K}_{\mathbf{Yr}} = \Phi(\mathbf{r})^T \mathbf{Y}_{c\Phi}$, $K_{\hat{\boldsymbol{\mu}}_{\mathbf{Y}}} = \hat{\boldsymbol{\mu}}_{Y_\Phi}^T \mathbf{Y}_{c\Phi}$ and $\hat{\mathbf{K}}_{\mathbf{Y}}$ is the estimated centered kernel matrix. Equation (56) is the kernel-RX equation used in this paper.

6. RESULTS

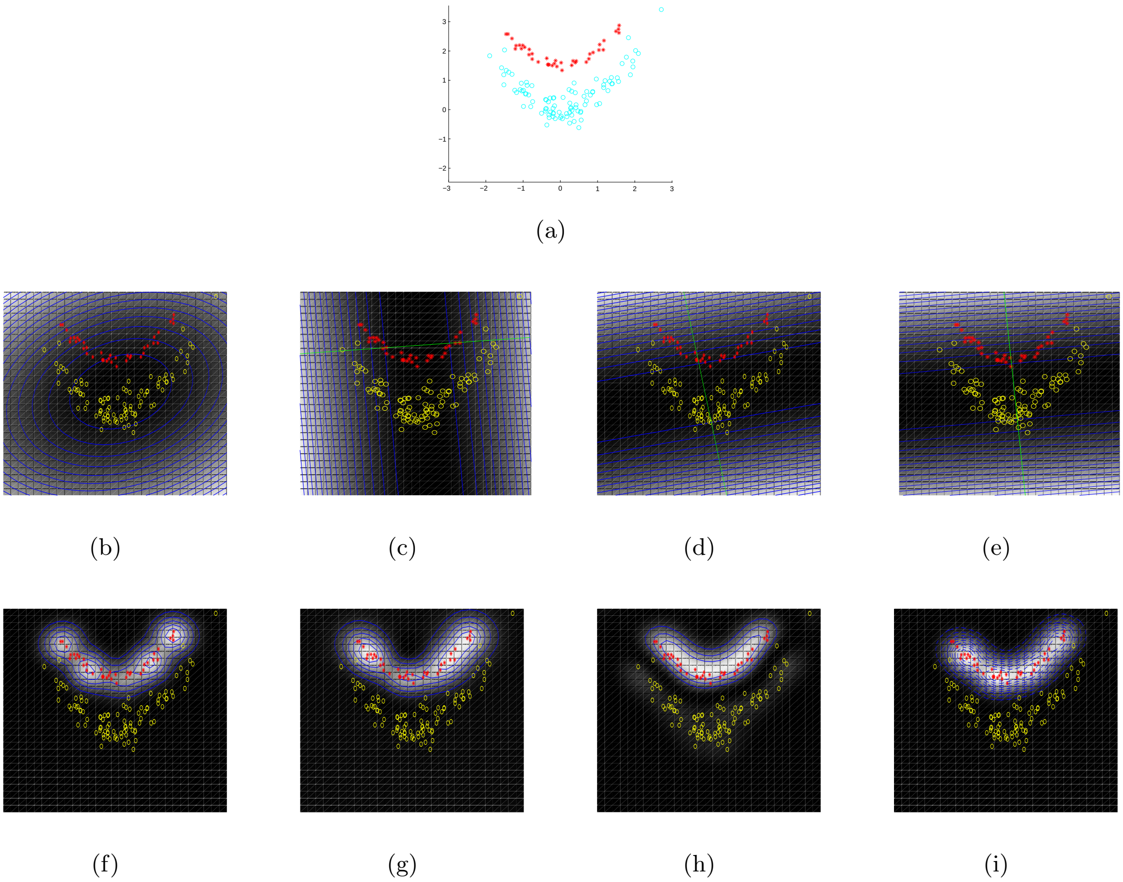
In this Section, each of the eight equations is implemented using both simulated illustrative toy data as well as real hyperspectral imagery from the Hyperspectral Digital Imagery Collection Experiment (HYDICE) and Airborne Hyperspectral Imager (AHI) data sets. For this paper, we use a Gaussian Radial Basis Function (RBF) kernel which takes the form $k(\mathbf{x}, \mathbf{y}) = \texttt{exp}[-||\mathbf{x} - \mathbf{y}||^2/2\sigma^2]$ where $\sigma > 0$ is a critical kernel parameter representing the width of the Gaussian kernel. This parameter must be chosen so that the RBF function can full exploit the data variations. In this paper, the value of σ was determined experimentally for each algorithm and for each image using a cross-validation technique. Performance results using ROC curve analysis for each of the eight methods are provided and compared.

6.1 Simulated Data

Each of the eight algorithms are implemented here as discrimination methods on an illustrative toy data set. The data set, shown in Figure 2(a), 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 2(b)-2(i). 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 2(c), 2(d), and 2(e) 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 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 generate decision boundaries which very



(a)

(b) (c) (d) (e)

(f) (g) (h) (i)

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

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.

6.2 Hyperspectral Imagery

Three real hyperspectral images from two different HSI sensor databases were used to 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 third is from the University of Hawaii's Airborne Hyperspectral Imager (AHI) sensor.

Before any processing is done, all spectra in each image are normalized so that all values in 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.[6] In all algorithms a dual window was used to collect data. To provide consistency, 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 vectors available for further processing.

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, 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. The probability of detection is defined as $P_D = \frac{N_{hit}}{N_T}$ and the false alarm rate is calculated by $R_{FA} = \frac{N_{miss}}{N_{TP}}$ 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. For visual purposes, all outputs shown below have been binary thresholded at a value which corresponds to an 80% detection rate for that image.

6.3 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 here; bands 1-22, 102-108, 137-151, and 195-210 have been removed. 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 3(a) and 4(a), respectively.

6.3.1 DR-II Results and Analysis

The ground truth for the DR-II image is shown in Figure 3(b). 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 3(c) - 3(j). The results shown are the best results obtained using the eight detectors outlined above. For PCA, the first 6 eigenvectors using the OWR spectra were used in Equation (8). For KPCA, the first 6 eigenvectors using the OWR spectra were used in the complement subspace form of Equation (28). For EST and KEST, the first 3 positive eigenvectors were used in Equations (15) and (53), respectively.

The ROC curves at low FAR for each of the eight methods can be seen in Figure 5(a). From these results, it appears that each of the four nonlinear methods performs better than its respective linear counterpart. 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.

6.3.2 FR-I Results and Analysis

The ground truth for the FR-I HYDICE image is shown in Figure 4(b). 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 4(c) - 4(j). The results shown are the best results obtained using the eight detectors outlined above. For PCA, the first 6 eigenvectors using the OWR spectra were used in Equation (8) and for KPCA, the first 6 eigenvectors using the OWR spectra were used in the complement subspace form of Equation (28). For EST, the first three negative eigenvectors were used in Equation (16) and for KEST the first 3 positive eigenvectors were used in Equation (53).

The ROC curves for each of the eight algorithms at low FAR are shown in Figure 5(b). From these results, 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 appearing at this detection rate. At very low R_{FA}, PCA outperforms all algorithms except KPCA and KFD. 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. The results for this image indicate that each of the nonlinear algorithms performs better compared with its respective linear version. 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.

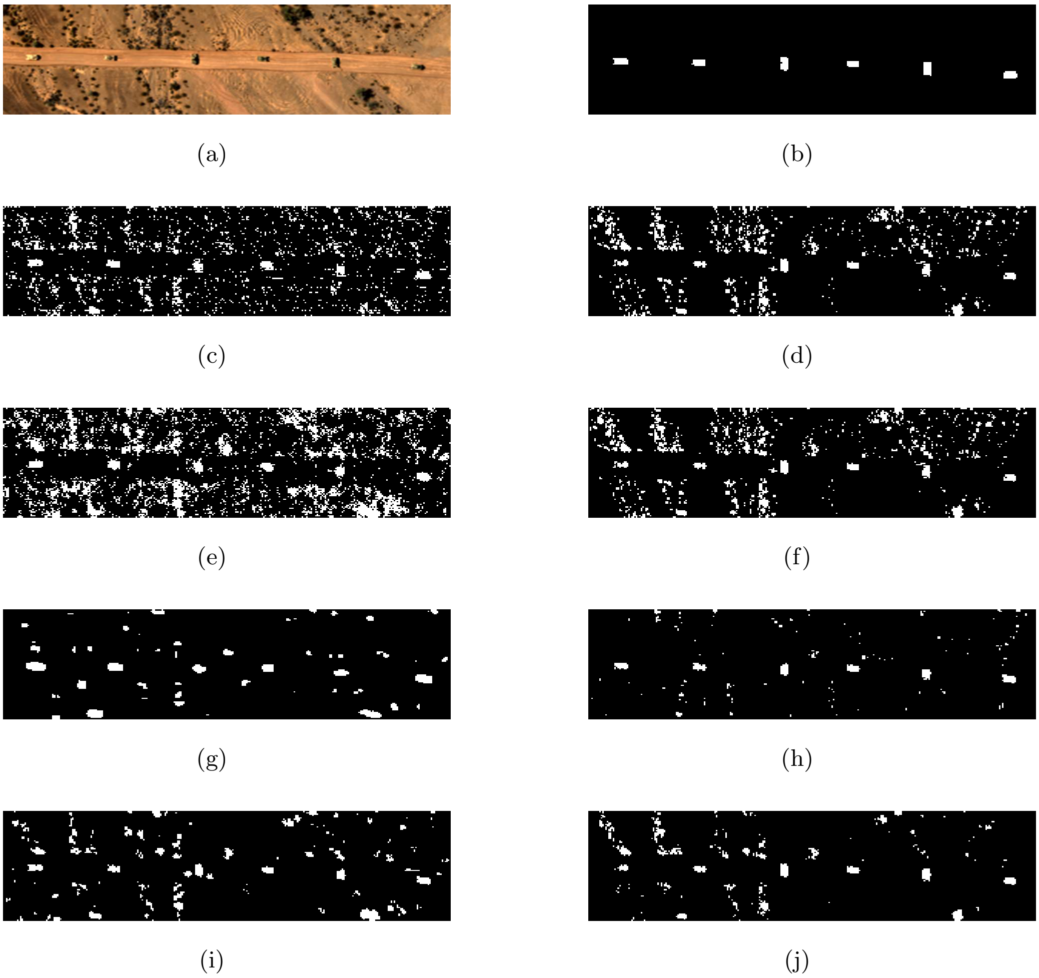
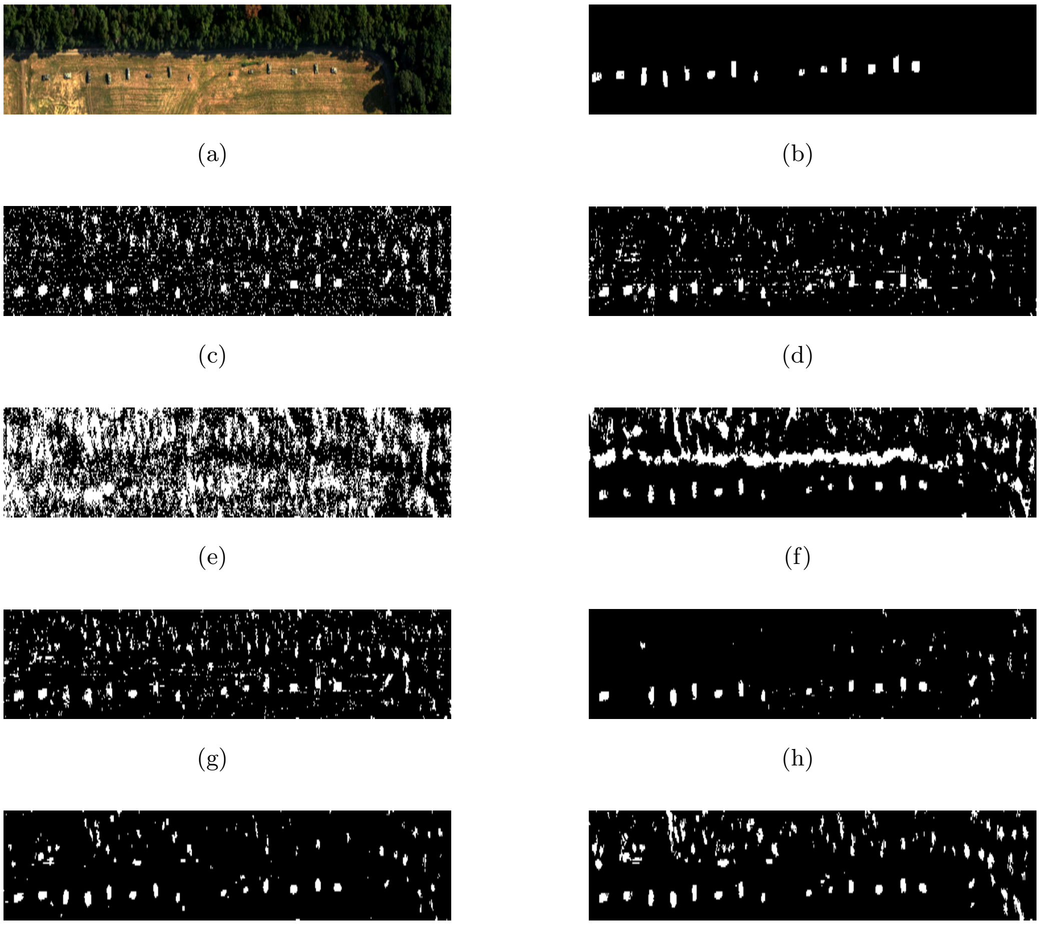


Figure 3. (a) Original DR-II HYDICE image. (b) Ground truth for the DR-II HYDICE image. Output results at 80% detection rate using (c) RX, (d) PCA, (e) FLD, (f) EST, (g) KRX, (h) KPCA, (i) KFD, and (j) KEST.

6.4 AHI Imagery and Results

The third image is from Hawaii's Airborne Hyperspectral Imagery (AHI) database. This hyperspectral cube contains 70 spectral bands and spans 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 image used in this paper is shown in Figure 6(a). The ground truth for the AHI-1 image is shown in Figure 6(b). It shows the locations of the thirty-five 'targets of interest' - mines in this case. All eight algorithms were once again implemented for this image and the best outputs for each can be seen in Figures 6(c) - 6(j). The results shown are again the best results obtained using the eight detectors outlined above. For PCA, the first six eigenvectors using the OWR spectra were used in Equation (8) and for KPCA, the first six eigenvectors using the OWR spectra were used in the complement subspace form of Equation (28). For EST, the first five positive eigenvectors were used in Equation (15) and for KEST the first five positive eigenvectors were used in Equation (53).

The ROC curves for each of the eight algorithms at low FAR are shown in Figure 5(c). From the images and ROC curves it can be seen that for this image at low FAR, KFD performs the best among all methods while RX clearly really suffers from a large number of false alarms and thus exhibits the worst detection performance. In general, at low false alarm rates, KFD, KPCA, and KRX all perform better than the other detectors. However, at very low FAR FLD actually achieves a higher detection rate than KEST. Nonetheless, each nonlinear detector performs at a higher level than its linear counterpart. However, the performance increase from each linear detector compared to its corresponding nonlinear detector is not as significant as in the HYDICE images. The



(a) (b)

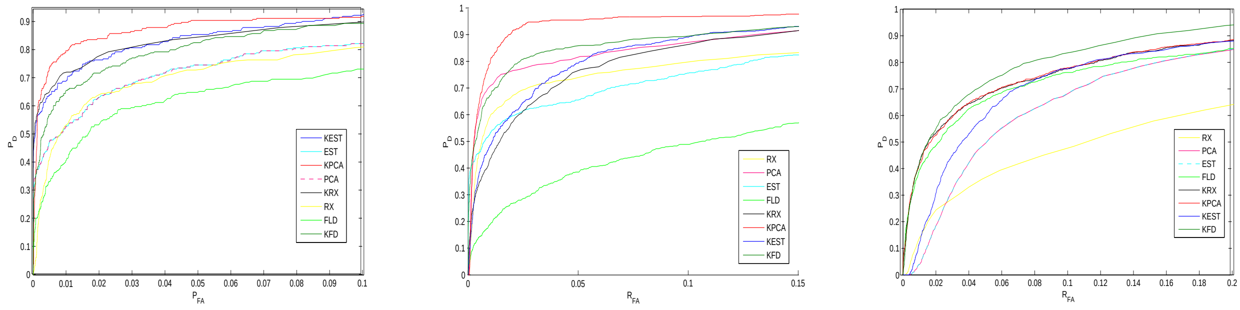
(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 4. (a) Original FR-I HYDICE image. (b) Ground truth for the FR-I HYDICE image. Output results at 80% detection rate using (c) RX, (d) PCA, (e) FLD, (f) EST, (g) KRX, (h) KPCA, (i) KFD, and (j) KEST.



(a) (b) (c)

Figure 5. ROC curves for the (a) DR-II (b) FR-I and (c) AHI-1 image at low false alarm rates.

reason for this is most likely explained by the large anomalous areas detected on the left side of the images in Figure 6. This region corresponds to the darker regions in Figure 6(a). Further analysis leads to the conclusion that the terrain of these areas are vastly different spectrally than the background; that is, the spectral properties of the dark region differ greatly 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

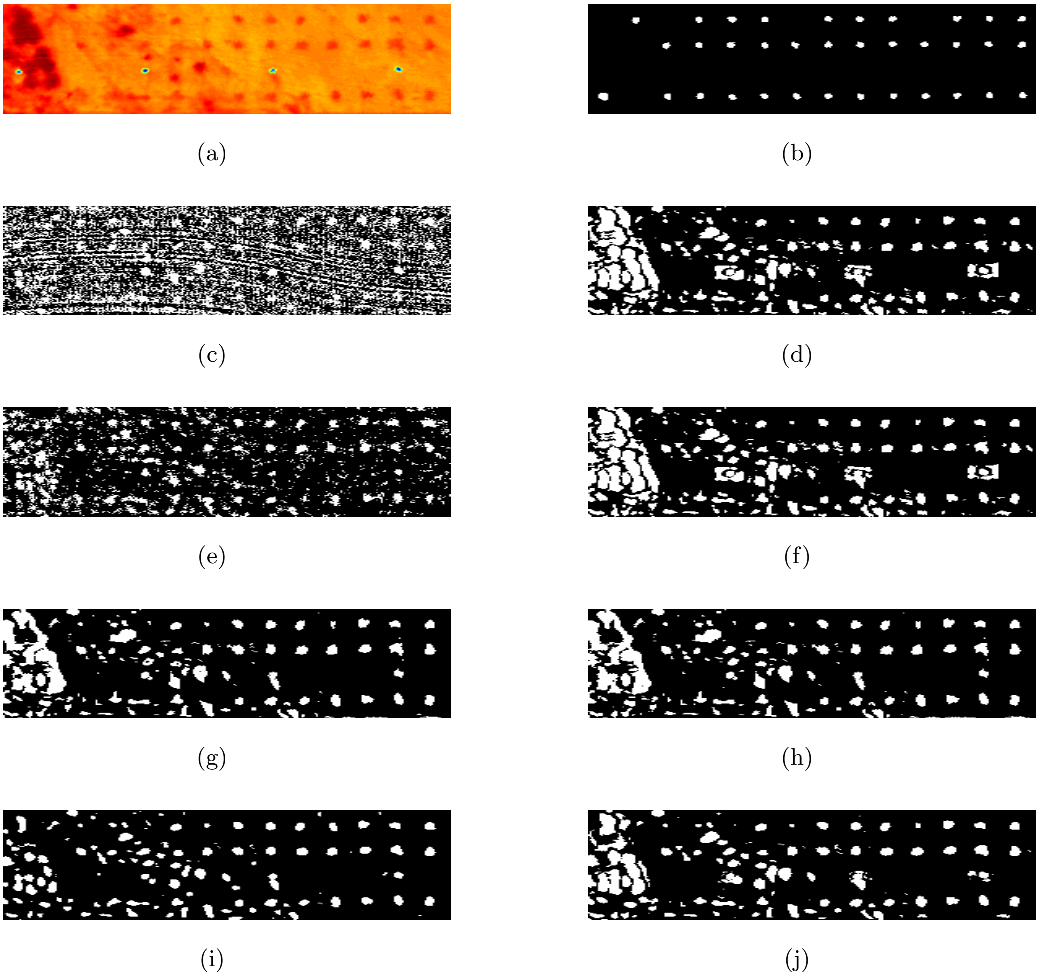


Figure 6. (a) Original AHI-1 image. (b) Ground truth for the AHI-1 image. Output results at 80% detection rate using (c) RX, (d) PCA, (e) FLD, (f) EST, (g) KRX, (h) KPCA, (i) KFD, and (j) KEST.

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 6(i), 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.

7. CONCLUSIONS

This paper provided a performance characterization of nonlinear kernel-based methods for hyperspectral anomaly detection. Four 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. 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 then needed to be rewritten in terms of kernels functions of the data in the original input space. All eight anomaly detection algorithms were briefly explained and implemented using three hyperspectral data cubes containing a varying number of 'targets of interest'.

The results from the three hyperspectral images did provide a rough sense that the kernel-based algorithms could achieve better detection levels than their respective linear methods. 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 is being tested in order to formulate a more accurate comparison of the algorithms.

REFERENCES

1. H. Kwon, N. Nasrabadi, and S. Der, "Adaptive anomaly detection using subspace separation for hyperspectral imagery," *Optical Engineering* **42**, pp. 3342–3351, Nov. 2003.
2. D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Processing Magazing* **19**, pp. 29–43, Jan. 2002.
3. 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* **19**, pp. 58–69, Jan. 2002.
4. D. Manolakis, D. Marden, and G. A. Shaw, "Hyperspectral image processing for automatic target detection applications," *Lincoln Laboratory Journal* **14**(1), pp. 79–116, 2003.
5. 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.* **38**, pp. 1760–1770, Oct. 1990.
6. H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing* **43**, pp. 388–397, Feb. 2005.
7. A. Banerjee, P. Burlina, and C. Diehl, "A support vector method for anomaly detectino in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing* **44**, pp. 2282–2291, Aug. 2006.
8. K. I. Ranney and M. Soumekh, "Hyperspectral anomaly detection within the signal subspace," *IEEE Geoscience and Remote Sensing Letters* **3**, pp. 312–316, July 2006.
9. I. T. Joliffe, *Principal Component Analysis*, Springer-Verlag, Berlin, Germany, 1986.
10. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, Inc., New York, 2 ed., 2001.
11. D. Torrieri, "The eigenspace transform for neural network classifiers," *Neural Networks* **12**, pp. 419–427, Apr. 1999.
12. B. Schölkopf, A. J. Smola, and K.-R. Müller, "Kernel principal component analysis," *Neural Computation* (10), pp. 1299–1319, 1999.
13. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Constructing descriptive and discriminative non-linear feature: Rayleigh coefficients in kernel feature space," *IEEE Trans. on Pattern Analysis and Machine Intell.* **25**(5), pp. 623–628.
14. L. A. Chan, N. M. Nasrabadi, and D. Torrieri, "Eigenspace transform for automatic clutter rejection," *Optical Engineering* , pp. 564–573, 2001.
15. B. Scholkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.
16. J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, 2004.