Autoencoder-based Hyperspectral Anomaly Detection using Kernel Principal Component Pre-Processing

Katinka Müller, Vinay Chakravarthi Gogineni, Milica Orlandić, Stefan Werner
Dept. of Electronic Systems, Norwegian University of Science and Technology-NTNU, Norway
E-mails: katinkam@stud.ntnu.no, {vinay.gogineni, milica.orlandic, stefan.werner}@ntnu.no

***Abstract*—Anomaly detection in hyperspectral remote sensing applications has attracted colossal attention due to its ability to uncover small distinctive objects dispersed across large geographical areas in an unsupervised manner. Autoencoders (AEs) have recently been demonstrated as effective tools for detecting hyperspectral anomalies. Using pre-processing techniques along with AEs improves accuracy by removing noise and irrelevant information from the data and also improves computational efficiency by reducing the dimensionality of the data or transforming it into a more appropriate representation. Therefore, this paper proposes to utilize principal component analysis (PCA) and kernel PCA (KPCA) based pre-processing methods in conjunction with the autonomous hyperspectral anomaly detection autoencoder (AUTO-AD). Further, we propose using multiple kernels in KPCA-based pre-processing to capture the complexity of the data better. Although KPCA- and MKPCA-based pre-processing shows excellent results when combined with hyperspectral anomaly detection algorithms, their high computational cost becomes crucial in resource-constrained applications. As a solution, we use random Fourier features (RFF) to approximate KPCA-based pre-processing. We conduct a series of experiments on various datasets to demonstrate the performance of the proposed framework. The experiments reveal that utilizing KPCA as a pre-processing step lead to better results than state-of-the-art hyperspectral anomaly detection approaches.**

***Index Terms*—Hyperspectral imaging, anomaly detection, autoencoder, kernel principal component analysis, random Fourier features.**

I. Introduction

Hyperspectral imaging (HSI) is a technique that uses a wide range of electromagnetic spectrums to analyze each pixel [1]. Thus, it is possible to distinguish objects by taking advantage of the different absorption wavelengths of light for different materials. Hyperspectral anomaly detection (HAD) refers to the identification of pixels or sub-pixels with significantly different spectral characteristics from their neighboring pixels [2], [3]. In HSI, a pixel is a vector whose dimension is equal to the spectral dimension of the HSI. Remote sensing applications such as search-and-rescue, mine detection, and environmental monitoring rely heavily on the HAD. Consequently, there has been an increased interest in HAD as a research field, leading to the development of several state-of-the-art models using both traditional and deep learning methods [4].

This work is partly supported by the Research Council of Norway.

In a few traditional methods, such as the Reed-Xiaoli (RX) detector [5], the background of the HSI is modeled as a multidimensional Gaussian distribution, and the deviation of a test vector from this distribution is calculated using the Mahalanobis distance [6]. However, due to the complexity of HSI, the Gaussian distribution is not always sufficient to accurately model the background distribution. Representation-based methods, such as collaborative representation detection (CRD) [7], assume that background pixels can be modeled as a linear combination of the surrounding pixels while anomalies cannot. Additionally, tensor decomposition-based methods, such as prior-based tensor approximation (PTA) [8], considers the 3D structure of the HSI by treating the 3D hypercube as a third-order tensor. It is, however, difficult to generalize CRD and PTA to new datasets as both require manual parameter setting. A common issue with traditional methods is achieving good feature extraction for different data types.

HAD approaches based on deep learning can overcome this problem by learning automatically how to extract features based on a variety of datasets [9], [10]. Autoencoders (AEs), which reconstruct the original input and use the error as an anomaly score, have recently been explored in HAD [11]–[13]. A recently proposed autonomous hyperspectral anomaly detection network (AUTO-AD) utilizes the same concept of reconstructing the background for separating the anomalies [14]. Instead of using the hyperspectral image as input, AUTO-AD uses uniform noise to train the model, which has been shown to improve the performance of the model. AUTO-AD has demonstrated promising results in terms of computational cost and detection accuracy and has outperformed several state-of-the-art HAD models. The AUTO-AD, however, does not employ any form of pre-processing, which would potentially enhance its performance.

In this paper, we propose the use of PCA and KPCA as pre-processing methods along with AE in the AUTO-AD model. Using KPCA, the underlying structure of the data can be efficiently captured, which aids in identifying anomalies that may not be apparent in the raw data, making it easier for an AE to identify them [15]. A recent study has demonstrated that using multiple kernels can improve hyperspectral data representation, making background and anomalies more distinct [16]. Therefore, the paper proposes

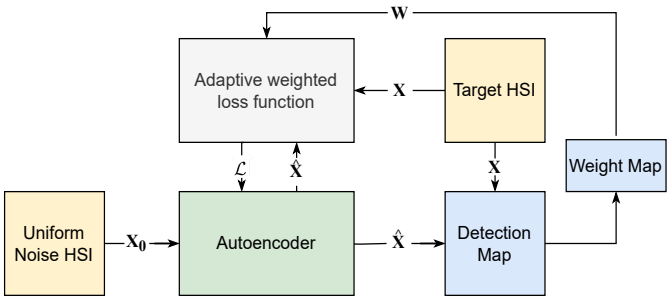


Fig. 1: Architecture of the autonomous hyperspectral anomaly detection (AUTO-AD).

to use multiple kernels in KPCA-based pre-processing, which results in multi-KPCA (MKPCA) pre-processing. Through this approach, various combinations of kernels can be used to capture different aspects of the data, therefore providing the AUTO-AD model with useful representations that could improve its performance. However, this advantage comes at the expense of high computational costs associated with KPCA when dealing with large datasets, which is even more pronounced in the case of MKPCA. To reduce the computational cost associated with KPCA and MKPCA, this paper adopts the principles of random Fourier features (RFF) [17]–[19]. We demonstrate the performance of the proposed HAD approach using real-world datasets.

II. Autonomous Hyperspectral Anomaly Detection (Auto-AD)

The AEs for HAD are based on the principle that anomaly pixels are less likely to be reconstructed than background pixels, i.e., the areas corresponding to anomalies result in high reconstruction error [14]. However, research has shown that the AE learns how to reconstruct the anomalies after a large number of iterations. To address this issue, the AUTO-AD uses an adaptive-weight loss function to suppress the anomalies further. The architecture of the AE proposed in the AUTO-AD is shown in Fig. 1. The AE consists of a decoder and encoder, which are built up using fully convolutional layers, together with skip connections to complement the encoder with spatial details.

The encoder consists of 15 convolutional layers, where the input to the encoder is a hypercube, of the same dimension as the target HSI, filled with uniform noise $\mathcal{U} \in [0, 0.1]$. The encoder uses convolutional filters to reduce dimensionality in the spatial and spectral domains. The decoder contains 11 convolutional layers together with up-sampling blocks that use nearest-neighbour interpolation to increase the spectral dimensionality. The input to the decoder is a concatenation of the image code in the encoder and outputs given from the skip connections. The decoder output contains the reconstructed background, which is sent as input to the adaptive weight-loss function.

Let $\mathbf{x}_{i,j} \in \mathbb{R}^{B\times 1}$ be the pixel in spatial position (i, j) in the target HSI $\mathbf{X} \in \mathbb{R}^{H\times W\times B}$ and $\hat{\mathbf{x}}_{i,j} \in \mathbb{R}^{B\times 1}$ is the corresponding output of the network. Then, the adaptive weight-loss function is given as

$$\mathcal{L} = \sum_{i=1}^{H}\sum_{i=j}^{W} ||(\mathbf{x}_{i,j} - \hat{\mathbf{x}}_{i,j})w_{i,j}||_2^2 \tag{1}$$

where $w_{i,j}$ is the weight corresponding to the pixel in spatial position (i, j) in the target HSI, which can be calculated as

$$w_{i,j} = \max(\text{vec}(\mathbf{E})) - e_{i,j}, \tag{2}$$

with $e_{i,j} = ||\mathbf{x}_{i,j} - \hat{\mathbf{x}}_{i,j}||_2^2$, vec$(\cdot)$ denotes the vectorization of the argument matrix, and

$$\mathbf{E} = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,W} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,W} \\ \vdots & \vdots & \ddots & \vdots \\ e_{H,1} & e_{H,2} & \cdots & e_{H,W} \end{bmatrix}. \tag{3}$$

The weights $w_{i,j}$ are initialized to one and will be updated according to (2) after a given amount of iterations P, which was set to 100 in the proposed AUTO-AD model. After each iteration, the loss will be fed backward to update the network parameters using the optimization algorithm [20]. The training stops when a particular stopping criterion is satisfied, e.g., after L_{max} iterations or when the average variation in loss is less than a certain threshold σ within the last I iterations:

$$\frac{1}{I}\sum_{i=k-I}^{k} |\mathcal{L}^{i+1} - \mathcal{L}^{i}| < \sigma. \tag{4}$$

where k signifies the number of iterations. In [14], I and L_{max} were set to 50 and 1000 iterations, respectively, while σ was set to 1.5×10^{-5}. With the AUTO-AD algorithm, we obtain a hyperspectral image containing the reconstructed background and a 2D detection map. The detection map is created by calculating the reconstruction loss between the target HSI and the reconstructed background. Brighter colors in the detection map indicate areas with higher reconstruction errors, while darker colors indicate lower reconstruction errors. Therefore, anomalies in the hyper-spectral image will appear as bright regions in the detection map.

III. Proposed HAD

This section presents the proposed HAD approach that utilizes PCA or KPCA-based pre-processing method along with AUTO-AD. Pre-processing can enhance anomaly detection accuracy by removing noise and redundant information from the data and may also improve computational efficiency by reducing the spectral dimension or transforming the data into a more appropriate representation. In addition, we propose utilizing multiple kernels in KPCA-based pre-processing to capture the intricacies of the data effectively. However, the computationally cost of KPCA and MKPCA-based pre-processing could pose significant challenges in resource-limited applications. To address this, we use random Fourier features (RFF) as an approximation technique for KPCA-based pre-processing. The proposed HAD approach is illustrated in Fig. 2.

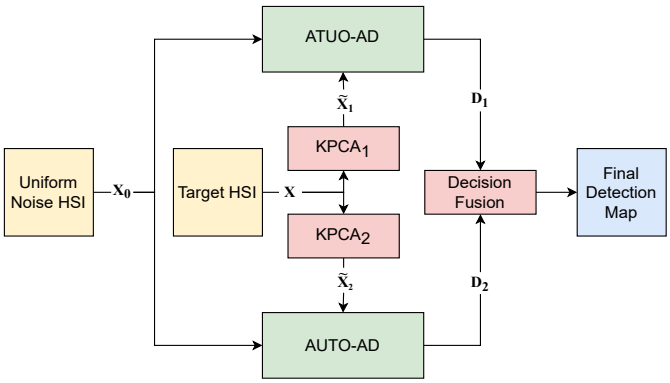


Fig. 2: Proposed implementation of AUTO-AD utilizing the MKPCA-based pre-processing.

A. Kernel Principal Component Analysis (KPCA)

Principal component analysis (PCA) is a mathematical technique that can be used for dimensionality reduction and feature selection for linearly separable data [21]. PCA transforms the data into a set of ξ principal components that are used to reduce the dimensionality while still containing most of the variation in the data. However, for nonlinearly separable data, the use of PCA is limited. In this case, a possible solution for dimensionality reduction is Kernel PCA. Using kernels, the data $\mathbf{x}$ can be projected onto a higher dimensional feature space as $\phi(\mathbf{x})$, where it becomes linearly separable [22]. The KPCA is similar to PCA, but with an additional step in which the data $\mathbf{x}$ is transformed into a higher dimensional space using the kernels. In the higher dimensional space, the inner product between two vectors $\mathbf{x}$ and $\mathbf{x}'$ can be computed using the kernel trick without explicitly transforming the data [15]. A continuous, symmetric, and positive-definite kernel $\kappa(\cdot)$ satisfies the Mercer's condition, which is given by

$$\kappa(\mathbf{x}, \mathbf{x}') = \phi^{\mathrm{T}}(\mathbf{x})\phi(\mathbf{x}'). \tag{5}$$

Some commonly used kernel functions are the radial basis function (RBF), Laplacian and sigmoid [15]. The first step in KPCA is to express the covariance matrix $\hat{\mathbf{C}}$ as an inner product between two transformed vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$ as follows:

$$\hat{\mathbf{C}} = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}_i)\phi^{\mathrm{T}}(\mathbf{x}_i') \tag{6}$$

where $N = W \times H$ is the number of pixels present in the HSI. The kernel matrix $\mathbf{K} = \{\kappa_i\}_{i=1}^{N}$, which is used to compute the principal components, can be set equal to the covariance matrix $\hat{\mathbf{C}}$ in (6). The principal components are found using the eigenvalue decomposition expressed as

$$\mathbf{KP} = \mathbf{\Lambda P} \tag{7}$$

where $\mathbf{P}$ contains the eigenvectors of $\mathbf{K}$, also referred to as the principal components, and $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues of $\mathbf{K}$ on its diagonal. For dimensionality reduction, only the first ξ principal components containing the most variance are used to transform the data. Applying KPCA with ξ principal components to the HSI $\mathbf{X}$ results in the new HSI $\widetilde{\mathbf{X}} \in \mathbb{R}^{H\times W\times \xi}$, which will be used instead of the original HSI for calculating the reconstruction loss $e_{i,j}$. Thus, the new adaptive weight-loss function is given by

$$\mathcal{L} = \sum_{i=1}^{H}\sum_{i=j}^{W} ||(\widetilde{\mathbf{x}}_{i,j} - \hat{\mathbf{x}}_{i,j})w_{i,j}||_2^2, . \tag{8}$$

Please note that the dimensionality of $\widetilde{\mathbf{x}}_{i,j}$ and $\hat{\mathbf{x}}_{i,j}$ is $\xi \times 1$ in (8).

B. Multi-Kernel PCA (MKPCA)

Multi-kernel learning combines multiple kernel functions to capture various aspects of the intricate structure of the data, which can enhance the background and anomaly separability [16]. Thus, we use MKL to obtain multiple detection maps, which are then fused together to yield the final detection map $\mathbf{D}$. To extract the detection maps, KPCA is first applied using different kernel functions to reduce the dimensionality, and used to compute the loss function in (8) of the AE separately. In the proposed MKPCA-based HAD approach, two kernel functions are utilized, resulting in two detection maps $\mathbf{D}_1$ and $\mathbf{D}_2$; these maps are fused using a convex combination to yield the final detection map $\mathbf{D}$ as follows

$$\mathbf{D} = \alpha\mathbf{D}_1 + (1-\alpha)\mathbf{D}_2 \tag{9}$$

where $\alpha \in [0, 1]$ is the fusion weight found using grid search.

By employing KPCA, it is possible to capture the complex underlying structure of the data, which helps identify the anomalies which may not be apparent in the raw data [22]. However, when N is large, the computation of a kernel matrix comes with a high computational cost [23].

C. KPCA using Random Fourier Features (RFF-KPCA)

RFF can be used to accelerate the computation of the kernel function when the amount of data samples, N, is large [17]. The kernel function (5) can be approximated as an inner product in the M-dimensional RFF space [18], [19]. This approximation becomes a finite-dimensional linear filtering problem that avoids the high computation of the kernel function. The RFF algorithm approximates the feature transformation $\phi(\cdot) \approx z(\cdot)$ used to compute the kernel function $\kappa(\mathbf{x}, \mathbf{x}')$, returning a new kernel function $\kappa(\mathbf{x}') = \phi^{\mathrm{T}}(\mathbf{x})\phi(\mathbf{x}) \approx z^{\mathrm{T}}(\mathbf{x})z(\mathbf{x}')$. The approximated transformation $\mathbf{z}(\mathbf{x})$, is given by

$$\begin{aligned} z(\mathbf{x}) = \sqrt{\frac{1}{M}}[&\cos(\mathbf{v}_1^{\mathrm{T}}\mathbf{x} + b_1)... \cos(\mathbf{v}_M^{\mathrm{T}}\mathbf{x} + b_M) \\ &\sin(\mathbf{v}_1^{\mathrm{T}}\mathbf{x} + b_1)... \sin(\mathbf{v}_M^{\mathrm{T}}\mathbf{x} + b_M)] \end{aligned} \tag{10}$$

where phase terms $\{b_i\}_{i=1}^{M}$ are drawn from a uniform distribution $\mathcal{U} \in [0, 2\pi]$. The RFF vectors $\{\mathbf{v_i}\}_{i=1}^{M}$ are drawn from a distribution $p(\mathbf{v})$ such that

$$\kappa(\mathbf{x} - \mathbf{x}') = \int p(\mathbf{v}\exp(j\mathbf{v}^{\mathrm{T}}(\mathbf{x} - \mathbf{x}')\, d\mathbf{v} \tag{11}$$

where $j^2 = -1$. The RFF vectors $\{\mathbf{v_i}\}_{i=1}^{M}$ can for example be the Gaussian or Laplacian distribution [17].

TABLE I: AUC scores of the proposed HAD approach with PCA, KPCA MKPCA, and RFF-MKPCA pre-processing methods. Also presented the AUC scores for RX, PTA, CRD, and AUTO-AD.

ABU scene	RX	PTA	CRD	Original	PCA	KPCA	MKPCA	RFF-MKPCA
Airport 1	0.8221	0.7331	0.9246	0.6941	0.9394	0.9537	0.9441	0.9333
Airport 2	0.8404	0.9096	0.8931	0.6764	0.9378	0.9789	0.9795	0.9287
Airport 3	0.9288	0.5476	0.9456	0.921	0.9411	0.9367	0.9440	0.9243
Airport 4	0.9526	0.9955	0.8664	0.5509	0.9879	0.9943	0.9947	0.9899
Airport Average	0.8860	0.7965	0.9074	0.7106	0.9515	**0.9659**	0.9656	0.9441
Beach 1	0.9828	0.9638	0.9882	0.9605	0.9730	0.9899	0.9889	0.9856
Beach 2	0.9106	0.8300	0.9257	0.9042	0.9702	0.9014	0.9072	0.9649
Beach 3	0.9998	0.9203	0.9932	0.9276	0.9959	0.9996	0.9999	0.9998
Beach 4	0.9887	0.9660	0.9417	0.9898	0.9704	0.9748	0.9746	0.9494
Beach Average	0.9705	0.9200	0.9622	0.9455	**0.9774**	0.9664	0.9677	0.9749
Urban 1	0.9907	0.9055	0.9887	0.9833	0.8816	0.9886	0.9914	0.9393
Urban 2	0.9946	0.9770	0.9294	0.9994	0.9962	0.9883	0.9945	0.9106
Urban 3	0.9513	0.8346	0.9414	0.7663	0.9750	0.9790	0.9791	0.9615
Urban 4	0.9887	0.8257	0.9549	0.9965	0.9866	0.9931	0.9923	0.9951
Urban 5	0.9692	0.8258	0.9371	0.9620	0.8834	0.9793	0.9788	0.9168
Urban Average	0.9789	0.8737	0.9503	0.9415	0.9446	0.9857	**0.9872**	0.9447

IV. Experimental Results

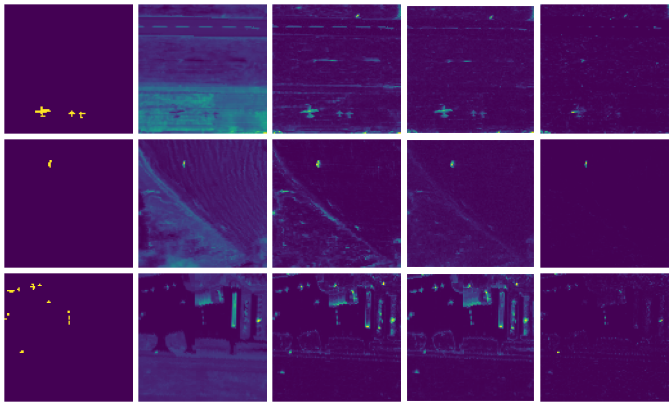
A series of experiments on various datasets are conducted to demonstrate the performance of the proposed HAD. The four pre-processing configurations, PCA, KPCA with the RBF kernel function, MKPCA with the RBF and Sigmoid kernel function and RFF-MKPCA, where RFF is used to construct the Gaussian and a Laplacian kernel function, are used. The experimental results are then compared with the original AUTO-AD and other state-of-the-art models by using the area under curve (AUC) metric [24].

A. Datasets

In this paper, thirteen HSI datasets from the public Airport-Urban-Beach (ABU) [25] are employed to verify the performance of the HAD. The image sizes are between 100×100 and 150×150 pixels with $102 - 207$ spectral bands. The HSIs have been manually extracted from images of the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), except for the Beach 4 scene taken from the Reflective Optical System Imaging Spectrometer (ROSIS-03). It is worth to mention that certain ABU datasets recognized under different names. For instance, Gulf Port is referred to as Airport 4, Texas Coast I and II are alternative names for Urban 1 and 2, Los Angeles 3 is another name for Airport 2, Beach 4 is commonly known as Pavia Center.

B. Parameter Settings

To find the number of principal components ξ for the pre-processing methods, tests are conducted for values of ξ between 10 and 200. For KPCA and MKPCA, tests are conducted using RBF, Laplacian, Sigmoid and Gaussian kernel functions. For PCA-based pre-processing, the highest detection score is achieved for $\xi = 20$. For KPCA, the number of components $\xi = 100$ together with the RBF kernel function are chosen based on the superior detection performance. MKPCA combining the Sigmoid and RBF kernel functions, each utilizing $\xi = 100$ principal components, provides the highest detection accuracy compared to the combination of other kernel functions. The RFF method with the dimension $M = 100$, resulting in KPCA with $\xi = 100$, provides the highest AUC detection by using Laplacian and Gaussian kernel functions.



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

Fig. 3: Detection map of airport 4 (top) , beach 3 (middle), and urban 3 (bottom). (a). ground truth, (b). conventional AUTO-AD , (c). with KPCA, (d). with MKPCA (e). with RFF-MKPCA.

C. Detection Performance

The proposed HAD is compared with other state-of-the-art HAD, RX [5], CRD [7] and PTA [8], and the original AUTO-AD model [14]. The parameters for CRD are set according to the parameters for Airport 4 in [26]. Parameters for PTA are set according to [8], whereas the RX model requires no parameter setting. The detection results are given in form of AUC scores in Table I and detection map plots in Fig. 3. The AUC scores are given for each dataset, and as an average over each scene.

By analyzing the average AUC scores of ABU scenes, it can be inferred that all four pre-processing methods performed better in terms of detection accuracy than the original AUTO-AD. As part of the proposed work, the experiments including the original AUTO-AD on the complete ABU dataset are

performed showing that there is no consistency in the detection performance for the different datasets, where the results range from 0.5509 for Airport 4 to 0.9994 for Urban 2. KPCA- and MKPCA-based pre-processing improve the anomaly detection accuracy of Airport scenes by 20% over the original AUTO-AD. In addition, detection results are consistent when these pre-processing techniques are employed. For Beach scenes, the highest AUC scores are achieved by the RX model, PCA- and RFF-MKPCA-based methods. This is expected as the background in the Beach scenes typically has less varying background than other scenes. These findings suggest that the linear methods such as PCA and RFF-MKPCA perform well when the underlying structure of the HSI is less complex. The detection maps for the original AUTO-AD and the AUTO-AD models with KPCA-based, MKPCA-based, and RFF-MKPCA-based pre-processing techniques, along with the ground truth are shown in Fig. 3. The detection maps are shown for Airport 4, Beach 3 and Urban 3. Overall, the RFF-MKPCA-based pre-processing method effectively eliminates most of the background, but also eliminates some of the anomalies, leading to high performance on Beach 3 but not on Airport 1 or Urban 3. This is also inherent in the AUC scores. Conversely, the original AUTO-AD has difficulty removing the background, which is a problem in all three datasets and is reflected in the AUC scores. Both KPCA and MKPCA have high AUC scores for all three scenes, and the detection maps correspond well with the scores. For the Urban 3 scene, even though MKPCA does not significantly suppress the background but rather preserves it to a large extent, the anomalies are still visible resulting in a high AUC score.

V. Conclusions

In this paper, we propose to use PCA-, KPCA- and MKPCA-base pre-processing methods to enhance the performance of the autonomous hyperspectral anomaly detection autoencoder (AUTO-AD). To reduce the computational cost associated with KPCA, random Fourier features(RFF) is used to approximate the KPCA-based pre-processing. Experiments were performed on the airport-urban-beach(ABU) datasets and compared to the original AUTO-AD and other state-of-art methods. The results showed that introducing pre-processing significantly increased the detection performance of the AUTO-AD, where the MKPCA-and KPCA-based pre-processing gave the most outstanding results.Additionally, results showed that PCA-and RFF-MKPCA-based outperformed other models when the underlying structure of the data was less complicated. In the future, the focus will be on decreasing the computational cost while still remaining high detection scores.

References

[1] G. ElMasry and D.-W. Sun, "Principles of hyperspectral imaging technology," in *Hyperspectral Imaging for Food Quality Analysis and Control*, D.-W. Sun, Ed. Academic Press, 2010, pp. 3–43.

[2] D. Stein, S. Beaven, L. Hoff, E. Winter, A. Schaum, and A. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 58–69, 2002.

[3] Y. Yuan, D. Ma, and Q. Wang, "Hyperspectral anomaly detection by graph pixel selection," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3123–3134, 2016.

[4] Y. Xu, L. Zhang, B. Du, and L. Zhang, "Hyperspectral anomaly detection based on machine learning: An overview," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Ses.*, vol. 15, pp. 3351–3364, 2022.

[5] I. 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, 1990.

[6] Q. Guo, B. Zhang, Q. Ran, L. Gao, J. Li, and A. Plaza, "Weighted-RXD and linear filter-based rxd: Improving background statistics estimation for anomaly detection in hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, pp. 2351–2366, 2014.

[7] W. Li and Q. Du, "Collaborative representation for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, pp. 1463–1474, 2015.

[8] L. Li, W. Li, Y. Qu, C. Zhao, R. Tao, and Q. Du, "Prior-based tensor approximation for anomaly detection in hyperspectral imagery," *IEEE Trans. Neural Net. Learn. Syst.*, vol. 33, no. 3, pp. 1037–1050, 2022.

[9] X. Fu, S. Jia, L. Zhuang, M. Xu, J. Zhou, and Q. Li, "Hyperspectral anomaly detection via deep plug-and-play denoising CNN regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, pp. 9553–9568, 2021.

[10] W. Li, G. Wu, and Q. Du, "Transferred deep learning for anomaly detection in hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 597–601, 2017.

[11] J. Zhang, Y. Xu, T. Zhan, Z. Wu, and Z. Wei, "Anomaly detection in hyperspectral image using 3d-convolutional variational autoencoder," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 2512–2515.

[12] G. Fan, Y. Ma, X. Mei, F. Fan, J. Huang, and J. Ma, "Hyperspectral anomaly detection with robust graph autoencoders," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[13] S. Chang, B. Du, and L. Zhang, "A sparse autoencoder based hyperspectral anomaly detection algorihtm using residual of reconstruction error," in *Proc. EEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 5488–5491.

[14] S. Wang, X. Wang, L. Zhang, and Y. Zhong, "Auto-AD: Autonomous hyperspectral anomaly detection network based on fully convolutional autoencoder," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022.

[15] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE J. Trans. Neural Net.*, vol. 12, no. 2, pp. 181–201, 2001.

[16] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Multiple kernel learning for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 1147–1160, 2011.

[17] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Info. Process. Syst.*, vol. 20, 2007.

[18] V. R. M. Elias, V. C. Gogineni, W. A. Martins, and S. Werner, "Adaptive graph filters in reproducing kernel hilbert spaces," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 7, pp. 62–74, 2021.

[19] V. C. Gogineni, V. R. M. Elias, W. A. Martins, and S. Werner, "Graph diffusion kernel lms using random fourier features," in *Proc. 54th Asilomar Conf. Signals, Syst., and Comput.*, 2020, pp. 1528–1532.

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Int. Conf. Learn. Represent.*, 2015.

[21] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1, pp. 37–52, 1987.

[22] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *J. Neural Comp.*, vol. 10, pp. 1299–1319, 1998.

[23] L. J. Cao, K. S. Chua, W. K. Chong, H. P. Lee, and Q. M. Gu, "A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine," *J. Neurocomputing*, vol. 55, pp. 321–336, 2003.

[24] C. Ferri, J. Hernández-Orallo, and P. Flach, "A coherent interpretation of auc as a measure of aggregated classification performance," in *Proc. Int. Conf. Machine Learn*, 2011, pp. 657–664.

[25] Data sets. [Online]. Available: http://xudongkang.weebly.com/data-sets.html

[26] X. Song, S. Aryal, K. M. Ting, Z. Liu, and B. He, "Spectral–spatial anomaly detection of hyperspectral data based on improved isolation forest," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.