remote sensing

MDPI

Article

SAOCNN: Self-Attention and One-Class Neural Networks for Hyperspectral Anomaly Detection

Jinshen Wang [1], Tongbin Ouyang [2], Yuxiao Duan [1] and Linyan Cui [1,*]

[1] Department of Aerospace Information Engineering, Beihang University, Beijing 100191, China
[2] Wuhan Digital Engineering Institute, Wuhan 430205, China
* Correspondence: cuily@buaa.edu.cn

Abstract: Hyperspectral anomaly detection is a popular research direction for hyperspectral images; however, it is problematic because it separates the background and anomaly without prior target information. Currently, deep neural networks are used as an extractor to mine intrinsic features in hyperspectral images, which can be fed into separate anomaly detection methods to improve their performances. However, this hybrid approach is suboptimal because the subsequent detector is unable to drive the data representation in hidden layers, which makes it a challenge to maximize the capabilities of deep neural networks when extracting the underlying features customized for anomaly detection. To address this issue, a novel unsupervised, self-attention-based, one-class neural network (SAOCNN) is proposed in this paper. SAOCNN consists of two components: a novel feature extraction network and a one-class SVM (OC-SVM) anomaly detection method, which are interconnected and jointly trained by the OC-SVM-like loss function. The adoption of co-training updates the feature extraction network together with the anomaly detector, thus improving the whole network's detection performance. Considering that the prominent feature of an anomaly lies in its difference from the background, we designed a deep neural extraction network to learn more comprehensive hyperspectral image features, including spectral, global correlation, and local spatial features. To accomplish this goal, we adopted an adversarial autoencoder to produce the residual image with highlighted anomaly targets and a suppressed background, which is input into an improved non-local module to adaptively select the useful global information in the whole deep feature space. In addition, we incorporated a two-layer convolutional network to obtain local features. SAOCNN maps the original hyperspectral data to a learned feature space with better anomaly separation from the background, making it possible for the hyperplane to separate them. Our experiments on six public hyperspectral datasets demonstrate the state-of-the-art performance and superiority of our proposed SAOCNN when extracting deep potential features, which are more conducive to anomaly detection.

Keywords: self-attention mechanism; one-class support vector machine; anomaly detection; hyperspectral image

check for updates

Citation: Wang, J.; Ouyang, T.; Duan, Y.; Cui, L. SAOCNN: Self-Attention and One-Class Neural Networks for Hyperspectral Anomaly Detection. *Remote Sens.* **2022**, *14*, 5555. https://doi.org/10.3390/rs14215555

Academic Editor: Edoardo Pasolli

Received: 13 September 2022
Accepted: 28 October 2022
Published: 3 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Hyperspectral images are a set of three-dimensional data and involve two-dimensional spatial information and abundant spectral information with hundreds of narrow contiguous bands [1]. Hyperspectral images have greater advantages in the areas of ground object classification [2–4], change detection [5,6], and target detection [7–9], depending on their powerful space–spectral information expression capabilities compared with other remote sensing images [10], such as visible and infrared light. According to whether the spectrum information of targets is known, target detection can be divided into either supervised target detection or anomaly detection [11,12]. Anomaly detection divides the pixels into anomaly and normal (background) without prior spectral information of the targets [13]. Generally, anomalies in hyperspectral images refer to the spectral irregularities caused by the presence of atypical ground objects [14]. The anomalies have not been authoritatively

defined by researchers and are widely agreed upon as targets whose spectral information clearly deviates from the background distribution, such as airplanes, vehicles, and other man-made objects in natural scenes [15,16].

Traditional methods of hyperspectral anomaly detection are generally divided into three categories: background modeling-based methods, distance-based methods, and representation-based methods [17,18].

The background modeling method is based on statistical theory. Assuming that the background obeys a certain distribution, statistical methods can be used to measure the likelihood of test pixels to be anomalous after estimating its parameters.

The well-known Reed–Xiaoli (RX) detector is a classic background modeling method [19]. It builds a statistical model of the background through the mean and covariance matrix of the whole image with the assumption that the background obeys a multivariate Gaussian distribution. Then, it detects anomalies by measuring the Mahalanobis distance between the test pixel spectrum and background spectral distribution. However, a whole-image-based background modeling method may be inaccurate with a complex background consisting of a variety of different distributions. Therefore, the Local RX (LRX) method uses samples from the local area around the test pixel to perform statistical background modeling [20]. Another issue with the RX method is that the background disobeys the multivariate Gaussian distribution in most cases. Therefore, Kwon and Nas [21] proposed a detection method based on the kernel function (Kernel RX, KRX), which maps hyperspectral image data into a higher dimensional feature space. Their results show that hyperspectral data tend to obey the multivariate Gaussian distribution in such a kernel space. Many improved RX detectors have appeared in follow-up studies, such as Segmented-RX [22], Weighted-RX, and linear filter-based RX detectors [23], etc.

In hyperspectral anomaly methods based on distance, pixels are grouped according to distance, and the pixels that deviate from the center of the cluster are considered anomalies. Distance-based approaches can be further classified into: support vector machine (SVM)-based, clustering-based, and graph-based methods. The SVM-based method obtains a hyperplane with the smallest distance in background samples by a support vector data description (SVDD) [24], and the distance between the test pixel and the center of the hyperplane in the higher feature space is considered the anomalous score [25]. In the clustering-based method, the original hyperspectral data are clustered by k-means [26], fuzzy clustering [27] or other approaches, and the Mahalanobis distance is calculated for each cluster to estimate the anomalous scores of test pixels. Graph theory [28], which reflects the internal structure of the data, is introduced in the graph-based method. Pixels in hyperspectral images are considered as the set of vertices of a graph and are connected with an edge, becoming inseparable when the similarity exceeds a given threshold. The large-component pixels of the graph form the background set; therefore, the test pixel's anomalous score is regarded as the distance from the nearest point in the background graph [29,30].

To avoid estimating the background with an inappropriate data distribution or evaluating the anomalies just by cluster distance, another method based on representation uses the linear combination of elements in the background dictionary to express each pixel in a hyperspectral image. Because the background and anomalies belong to different clusters, the representation of the background pixels will perform better than the anomalies through the constructed background-based dictionary, so that the residual image of the represented and original images is considered a measure of anomalies. Sparse (SR) [31], collaborative (CR) [32], and low-rank representation (LRR) [33] are all representation methods.

SR theory considers that signals are sparse [34], i.e., the features of a background sample are far fewer than its number. Therefore, the background pixels can be expressed by as few linear combinations of elements as possible in a given over-complete dictionary, which is constructed by a large amount of background sample data. In the SR-based anomaly detection method, the reconstruction of anomalies through a learned dictionary is worse than that of the background [13], and various anomaly detection models based on sparse representation theory have been proposed [35–37]. Furthermore, a CR-based anomaly detector was proposed based on the assumption that the collaboration between

dictionary atoms is more considerable than the competition between them [38]. The number of dictionary atoms used to represent test pixels is required to be as small as possible in SR theory, while all dictionary atoms are used in CR theory. CR-based methods have rapidly developed many extensions [39–41]. The over-complete background dictionary in SR and CR theory is usually obtained by a dual-window strategy [42], which indicates that only local attributes are used to constrain the coding vector; however, global structure information is not considered. In addition, setting the dual window to a reasonable size without polluting anomalies is difficult when constructing the dictionary in the absence of prior information.

Different from the two theories above, LRR assumes that each pixel in hyperspectral data can be expressed as a linear representation of the basis vectors formed by the dictionary [43]. In real hyperspectral images, the number of categories is far fewer than the number of pixels; therefore, the matrix's rows or columns are strongly correlated, creating a low-rank hyperspectral image. Based on LRR theory, a hyperspectral image is decomposed into a low-rank background matrix and a sparse anomaly matrix with the assumption that the background is in a low-dimensional subspace and considered to be highly correlated; furthermore, anomalies only occupy a small part of the image scene and have a low occurrence probability [44]. Other algorithms were presented to better represent and decompose the background and anomaly matrices [45–47] and to find the lowest rank representation of all pixels and use the sparse constraint anomalies under the background dictionary.

In recent years, deep learning has gradually become popular in the field of computer vision and has achieved many outstanding study results. Due to its excellent performance in the modeling and feature extraction of complex data, deep learning has been introduced into hyperspectral anomaly detection. A deep neural network's main role in anomaly detection is to extract the essential features of the original hyperspectral image as a feature extractor while reducing the dimensionality.

Most researchers proposed generative models, including autoencoders (AEs) [48], deep belief networks (DBNs), [49], and generative adversarial networks (GANs) [50], to minimize the error between the original and reconstructed spectra to extract features in the hidden layer or reconstruction spaces. Subsequently, traditional anomaly detectors (such as RX, CRD, LRR, etc.) have been applied to the features extracted by deep neural networks to perform the final detection. Arisoy et al. [51] utilized a GAN model to generate a synthetic background image similar to the original and an RX detector to detect anomalies on the residual image of synthetic and original images. Jiang et al. [52] used an adversarial autoencoder (AAE) to reconstruct the hyperspectral image and combined the morphological filter and RX detector on the residual image of original and reconstructed images to detect anomalies. AAE was also adopted in the spectral adversarial feature learning (SAFL) architecture proposed by Xie et al. [53]. Self-attention mechanisms [54] have been widely employed for anomaly detection in texts [55], videos [56], and industrial images [57] because of the effective learning capabilities. Jiang [58] proposed a manifold-constrained, multi-head, self-attention, variational autoencoder (MMS-VAE) method for hyperspectral anomaly detection, which introduced a self-attention mechanism to focus on abnormal areas by learning context-related information.

The main problem for methods based on deep learning is that the objective functions do not jointly optimize the feature extractor and anomaly detector, which makes the deep neural network unable to exert its advantages.

To address the above problems, we propose a new hyperspectral anomaly detection model based on deep learning to jointly optimize the feature extractor and hyperplane layer (i.e., the anomaly detector) in this paper. Accordingly, we present a new model with an extractor to obtain both global and local features in hyperspectral images, in addition to an anomaly weight map that roughly evaluates the probability of each pixel to be anomalous and enhances possible anomalous regions. Specifically, we designed our proposed network structure and loss function as an extension of the basic one-class support vector machine (OC-SVM) algorithm [24,59]. OC-SVM is a variant of the traditional SVM algorithm in which all training samples are considered as positive and the original data are mapped

to a new high-dimensional space through the kernel function, solving the problem of unbalanced samples in two-class classification. Our proposed model replaces the kernel function in the original OC-SVM with a feature extractor based on a deep neural network with optimizable parameters, making it possible to jointly train the feature extraction network with the subsequent hyperplane layer for a one-class objective.

In the feature extractor, both global and local features are obtained, and the original data are mapped into a new feature space. We introduced an attention mechanism to mine the feature correlation of each pixel with its neighbor in the image as the global information. In addition, we utilized an anomaly weight map to enhance the possible anomalous regions, considering that low-SNR targets may decrease anomaly detection performance. Moreover, we also designed a local feature extraction block because a local spatial structure helps to improve the efficiency of feature utilization [60,61]. In the hyperplane layer, most of the pixels are considered as background, while the origin of the high-dimensional space is representative of the anomaly points. Finally, we trained the best hyperplane to separate the background from the anomaly by maximizing the distance between the hyperplane and the representative point for anomaly.

Compared with other existing state-of-the-art hyperspectral anomaly detection methods, the main contributions of our new proposed method are summarized as follows:

1. We propose a new hyperspectral anomaly detection model and an unsupervised training framework. Whereas other methods separated them, we combine and co-optimize the feature extraction network and hyperplane layer through the one-class objective function, which maximizes the advantages of a deep neural network to extract features customized for the anomaly detector.
2. Our model simultaneously extracts global and local image features to achieve higher feature utilization levels. We calculate the relevant information between pixels based on self-attention mechanisms to obtain the image's global features; however, we also use a local feature extraction block to mine the local spatial information of each pixel's neighboring regions.
3. We apply the anomaly weight map in the self-attention mechanism to enhance the possible anomalous regions in the original image. The differences between targets and the background are highlighted through the anomaly weight map, which helps the final anomaly detection results, especially when the target is weak and confounded by the background.

The remainder of our article is organized as follows. Section 2 presents the proposed method in detail. Section 3 introduces experimental settings and provides the detection results. Section 4 discusses the efficiency of the adversarial learning and joint training methods. Finally, Section 5 presents our conclusions.

2. Proposed Method

This section will provide a brief introduction of our proposed hyperspectral anomaly detection method based on self-attention and one-class neural networks (SAOCNNs). The SAOCNN contains two components: an attention-oriented feature extraction network and a hyperplane layer based on the OCSVM network (OCSVM-Net). The loss function constrains the extracted features to make the new high-dimensional feature space more separable by maximizing the distance between the hyperplane layer and anomaly representative point, achieving the objective of joint training feature extraction and hyperplane optimization.

2.1. Overview

Figure 1 shows the complete flow chart of our proposed method. Suppose that the original hyperspectral data are reshaped to $\boldsymbol{X} = [x_1, x_2, x_3, \cdots, x_N] \in \mathbb{R}^{k \times N}$, the image contains N pixels (the image size is $H \times W$, the total number of pixels is $N = H \times W$), and each pixel has k bands.

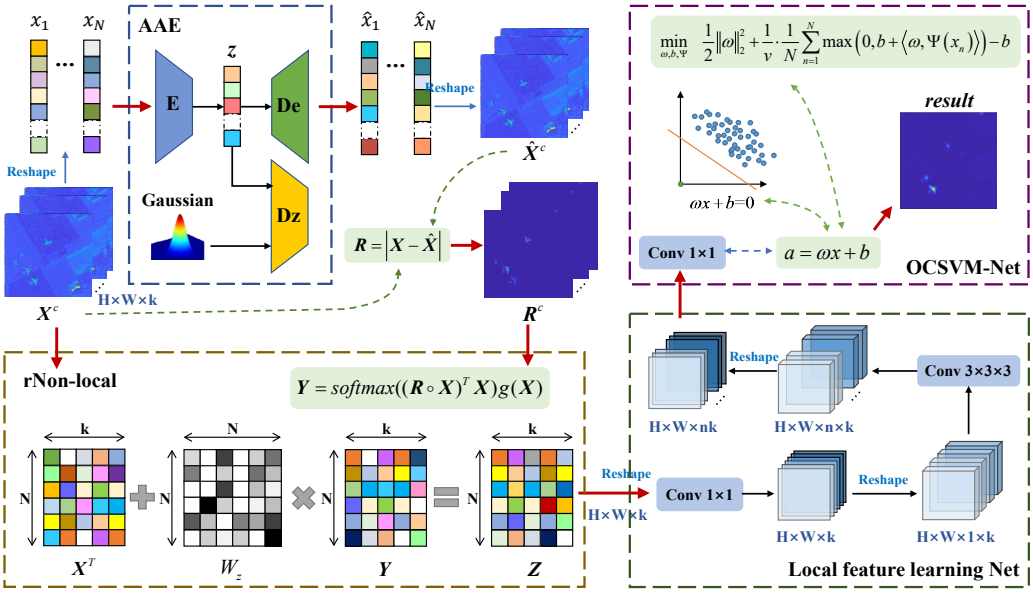


Figure 1. Flow chart of proposed method for hyperspectral anomaly detection consisting of two parts: the feature extraction network, which includes AAE, rNon-local, and Local feature learning Net, and the anomaly detection network as OCSVM-Net. $\boldsymbol{X}^c$ is the original 3D hyperspectral matrix reshaped with $N = H \times W$ pixels and k bands. Here, superscript c only serves as a symbol to represent the 3D matrix with no practical meaning. Same applies to reconstructed image $\hat{\boldsymbol{X}}^c$ and residual image $\boldsymbol{R}^c$.

The original hyperspectral image $\boldsymbol{X}$ is first input into an adversarial autoencoder (AAE) [62] to obtain the reconstructed image $\hat{\boldsymbol{X}}$. Due to the majority of background pixels in training samples and the constraint of the feature discriminator Dz in the hidden layer, the reconstruction performance of the background is better than that of anomalies in $\hat{\boldsymbol{X}}$, which will be explained in detail in Section 2.2.1. The residual image $\boldsymbol{R}$ is obtained by the residual between the original $\boldsymbol{X}$ and reconstruction $\hat{\boldsymbol{X}}$, in which the residual value of possible anomalous pixels is greater than that of background pixels.

The residual $\boldsymbol{R}$ is considered an anomaly weight map and is input into a residual non-local network (rNon-local) together with the original $\boldsymbol{X}$. In rNon-local, the anomaly weight map is added to enhance the possible anomaly areas in the original $\boldsymbol{X}$, which makes the anomaly area features more prominent.

The global features of each pixel are extracted in rNon-local, while the local features are extracted in the subsequent local feature learning network, including spectral features, spectral correlations of adjacent spectral segments, and spatial correlations of adjacent pixels. The whole feature extraction network maps the hyperspectral data to a new feature space through the local feature learning network, making the anomalies and background easier to separate with hyperplanes.

The feature extraction network is followed by OCSVM-Net as the hyperplane layer to train a hyperplane that separates the anomaly from the background. The OCSVM-Net evaluates the possibility of an anomaly according to the distance between the feature vector of each pixel mapped in the feature space and the hyperplane. During the whole training process, the hyperplane layer and feature extraction network are repeatedly optimized and iterated according to the loss function, obtaining the final hyperspectral anomaly detection model.

2.2. Attention-Oriented Feature Extraction Network

As shown in Figure 1, this module consists of three parts: AAE, rNon-local, and Local feature learning Net. In most hyperspectral anomaly detection methods based on deep learning, the networks are usually constructed from stacked fully connected layers because a single pixel is just a multi-dimensional vector considered as a separate sample,

leading to only spectral features being extracted in each pixel. However, the similarity between background pixels is greater than between anomaly and background pixels, which means the global correlation of background pixels is better than that of anomalous pixels. Therefore, we extracted global correlation features based on self-attention mechanisms to achieve better detection results in the rNon-local network. We added the residual map of the original and reconstructed images generated by the AAE as the anomaly weight map to enhance the possible anomaly regions, which further improves the performance of global feature extractor. In addition, local structure information is extracted through Local feature learning Net in the whole feature extraction block to improve feature utilization, including local-region spectral and spatial features extraction with two- and three-dimensional convolutional layers, respectively.

2.2.1. Anomaly Weight Map Generated Block

AAE combines the generative adversarial network's (GAN) generalization ability [50] with the autoencoder's (AE) reconstruction ability [48], utilizing the potential adversarial learning method to train the autoencoder. The anomaly is distinguished in the reconstructed map due to the higher reconstruction error of the anomaly compared with the background.

(1) Autoencoder:

AE is a two-layer neural network model for unsupervised learning features to accurately reconstruct the original input by minimizing the reconstructed error between the output $\hat{\boldsymbol{x}}$ and input vectors $\boldsymbol{x}$. The AE is divided into two parts, namely, encoder E and decoder De: the encoder is a structure in which $\boldsymbol{x}$ is projected from the input to hidden layer through a mapping function $f(\cdot)$, and the decoder is a structure in which $\boldsymbol{z}$ is projected from hidden to output layer through a mapping function $\hat{f}(\cdot)$:

$$\begin{aligned} \boldsymbol{z} &= f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \\ \hat{\boldsymbol{x}} &= \hat{f}(\hat{\boldsymbol{W}}\boldsymbol{z} + \hat{\boldsymbol{b}}) \end{aligned} \tag{1}$$

where $\boldsymbol{W}$ and $\hat{\boldsymbol{W}}$ are weight matrices; $\boldsymbol{b}$ and $\hat{\boldsymbol{b}}$ are deviation vectors.

(2) Generative Adversarial Networks:

GAN is a generative model that learns specific data distributions, which consists of two networks: generator G and discriminator D. Let the real data $\boldsymbol{x}$ fit the p_{data} distribution, the generator map the fake data $\boldsymbol{z}$ conforming to the p_z distribution to the expected distribution, and make it as close to p_{data} distribution as possible; subsequently, the discriminator attempts to identify the real samples in these two distributions:

$$\min_{\mathrm{G}} \max_{\mathrm{D}} V(\mathrm{G},\mathrm{D}) = \min_{\mathrm{G}} \max_{D} \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log \mathrm{D}(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(z)}[\log(1-\mathrm{D}(\mathrm{G}(\boldsymbol{z})))] \tag{2}$$

where V(G,D) is maximized when optimizing the discriminator and minimized when optimizing the generator. Through continuous repetitive training, the model can reach convergence and minimize the Jensen–Shannon (JS) difference between the two distributions [52].

In hyperspectral anomaly detection, the anomaly and background belong to anomaly and background clusters in original feature space, respectively, which are supposed to belong to hidden-layer cluster A and hidden-layer cluster B, respectively, in the hidden feature space. The reconstruction process of AE for anomalous pixels can be explained as the mapping stage in Figure 2a, while the background pixels' reconstruction process is demonstrated in Figure 2b. We expected that the background-cluster learning is better than the anomaly cluster in AE, which means the residual value between reconstruction $\hat{\boldsymbol{X}}$ and original $\boldsymbol{X}$ anomalous pixels will be greater than that of background pixels, therefore highlighting the target and suppressing the background in the residual image $\boldsymbol{R}$.

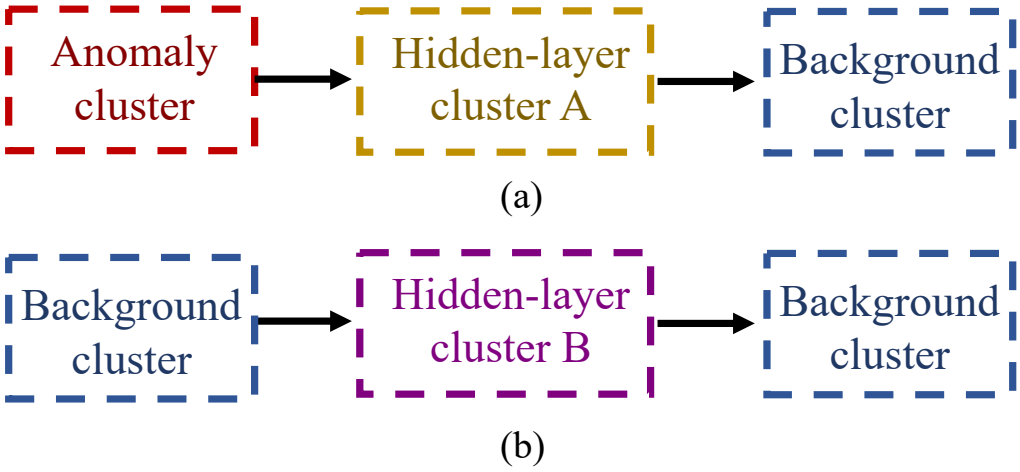


Figure 2. Reconstruction process of AE in hyperspectral data. (**a**) Reconstruction of anomaly. (**b**) Reconstruction of background.

However, the reconstruction ability of AE is too strong in real situations, meaning both mapping processes are learned efficiently during training; consequently, the residual map cannot effectively highlight the target. Therefore, the feature discriminator Dz is introduced, and encoder E is equivalent to the corresponding generator. The Gaussian distribution is selected as the prior distribution in discriminator Dz because we assumed that the hyperspectral image background pixels obey Gaussian probability [19]. Adversarial learning method forces constrain hidden features z and make all hidden-layer samples belong to the same hidden-layer cluster G. Because the number of background samples is much higher than the number of anomalies, the decoder De only learns the mapping from hidden-layer cluster G to the background cluster, as demonstrated in Figure 3. The reconstruction error of anomalous pixels will be larger, effectively highlighting the anomalies in residual images $\boldsymbol{R}$.

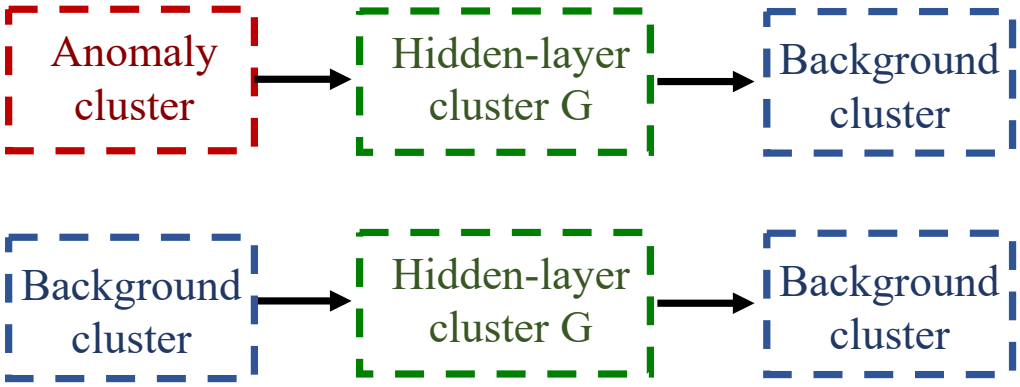


Figure 3. Reconstruction process of AAE in hyperspectral data.

The specific structure of AAE in our proposed model is shown in Figure 4, and it is composed of an encoder E, a decoder De, and a feature discriminator Dz. The encoding–decoding process is considered the reconstruction of the original data $\boldsymbol{X}$. The reconstruction loss function is the Huber loss [63]:

$$L_\delta(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \begin{cases} \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}}) & \text{for } |\boldsymbol{x} - \hat{\boldsymbol{x}}| \le \delta \\ \delta|\boldsymbol{x} - \hat{\boldsymbol{x}}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{3}$$

where δ represents the hyperparameters when training.

The discriminator's loss function is Wgan-gp [64]:

$$L_{\text{Gan}} = E_{\tilde{z} \sim P_{\text{gauss}}}[\text{Dz}(\tilde{z})] - E_{z \sim P_E}[\text{Dz}(z)] - \lambda E_{t \sim P_t}\left[(\|\text{Dz}(t)\|_2 - 1)^2\right] \tag{4}$$

where $\tilde{z}$ is the Gaussian sampling data; P_{gauss} is the Gaussian distribution to which the Gaussian sampling data belongs; P_E is the data distribution of the hidden-layer space; $\text{Dz}(\cdot)$ represents the output in Dz; $\boldsymbol{t}$ is a uniform sampling of the linear space P_t along the sampling points between P_{gauss} and P_E; and $E[\cdot]$ represents the mathematical expectation function.

In our proposed model, we used a two-dimensional convolutional layer with a kernel size of 1 × 1 instead of a fully connected layer. The group normalization [65] and the LeakyReLU [66] activation function are applied between each layer.

LeakyReLU is a variant of ReLU (rectified line unit):

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{for } x \geq 0 \\ \alpha x & \text{otherwise} \end{cases} \tag{5}$$

where the hyperparameter α is set to 0.2 in the method.

Because the 1 × 1 convolution kernel is implemented instead of a fully connected layer, the training batch is very small, and group normalization is selected in the network:

$$\text{GN}(\boldsymbol{x}) = \frac{\boldsymbol{x} - E(\boldsymbol{x})}{\sqrt{\text{Var}(\boldsymbol{x}) + \varepsilon}} * \gamma + \beta \tag{6}$$

where $E(\boldsymbol{x})$ is the average value; $\text{Var}(\boldsymbol{x})$ is the standard deviation; and ε, γ, and β are the learned parameters.

In summary, the normalization and activation between adjacent convolutional layers can be expressed as:

$$\boldsymbol{y} = \text{LeakyReLU}(\text{GN}(\boldsymbol{x})) \tag{7}$$

where $\boldsymbol{x}$ represents the output of the previous convolutional layer and $\boldsymbol{y}$ represents the input of the next convolutional layer.

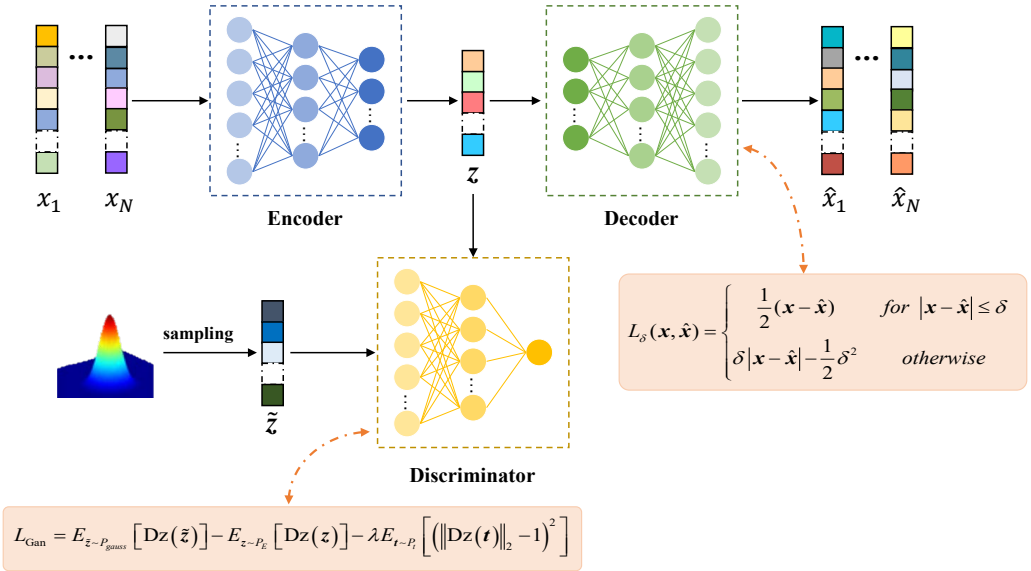


Figure 4. Details of network and loss functions based on AAE. Huber loss is provided for the encoder–decoder, and Wgan-gp loss is provided for the discriminator.

2.2.2. Residual Non-Local Network

The original non-local network [67] flow chart is shown as Figure 5a, and assuming $\boldsymbol{X} = [x_1, x_2, x_3, \cdots, x_N]$ as the input and $\boldsymbol{Y} = [y_1, y_2, y_3, \cdots, y_N]$ as the output, the original non-local network extracting global features is

$$y_i = \frac{1}{C(\boldsymbol{x})} \sum_{\forall j} f(\boldsymbol{x}_i, \boldsymbol{x}_j) g(\boldsymbol{x}_j) \tag{8}$$

where i is the index of the output position; j is the index of enumerating all possible positions; $\boldsymbol{x}$ is the input signal; and $\boldsymbol{y}$ is the output signal. The function $g(\cdot)$ is used to calculate the feature representation of $\boldsymbol{x}_j$, and the function $f(\cdot)$ is used to calculate the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $\mathcal{C}(\boldsymbol{x})$ is a function used for normalization. In the original non-local network, the similarity between the current pixel and all other pixels in the image is calculated through the function $f(\cdot)$ to obtain the global correlation features of the current pixel by connecting local and non-local information.

However, the global features extracted by the original non-local network cannot effectively highlight the anomaly when the target is weak, which we demonstrated in an ablation experiment (Section 3.5). The non-local network is improved to a residual non-local network in order to solve the above problems (Figure 5b). The input of rNon-local is set to the original hyperspectral image $\boldsymbol{X}$ and the anomaly weight map as the residual $\boldsymbol{R}$, which roughly evaluates the anomaly probability of each pixel in the original image. The size of the original image and anomaly weight map is reshaped into $k \times N$, i.e., $\boldsymbol{X}, \boldsymbol{R} \in \mathbb{R}^{k \times N}$. The Hadamard product of the two is presented to obtain the anomaly enhancement map $\boldsymbol{V} = \boldsymbol{R} \circ \boldsymbol{X} \in \mathbb{R}^{k \times N}$. The possible anomalous area in $\boldsymbol{V}$ has been enhanced compared with $\boldsymbol{X}$.

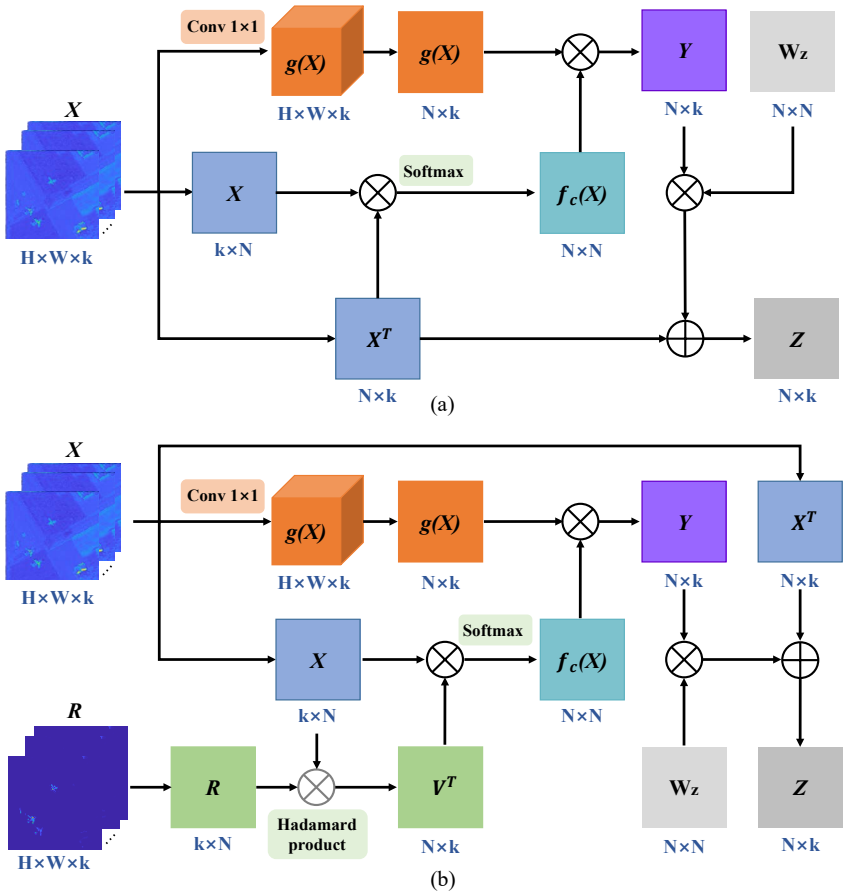


Figure 5. (**a**) Original non-local network with input of $\boldsymbol{X}$. (**b**) Improved rNon-local in proposed method with input of $\boldsymbol{X}$ and residual $\boldsymbol{R}$.

The similarity between vector $\boldsymbol{v}_i$ and $\boldsymbol{x}_j$ is calculated and the Gaussian function is chosen:

$$f(\boldsymbol{v}_i, \boldsymbol{x}_j) = e^{\boldsymbol{v}_i^T \boldsymbol{x}_j} \tag{9}$$

Softmax is used to perform normalization; therefore, the rNon-local is equivalent to the self-attention mechanism. The global feature response of each position in the output of the rNon-local is

$$\begin{aligned} \boldsymbol{y}_i &= \frac{1}{\mathcal{C}_i} \sum_{\forall j} f\left(\boldsymbol{r}_i \circ \boldsymbol{x}_i, \boldsymbol{x}_j\right) g\left(\boldsymbol{x}_j\right) \\ \mathcal{C}_i &= \sum_{\forall j} f\left(\boldsymbol{r}_i \circ \boldsymbol{x}_i, \boldsymbol{x}_j\right) \end{aligned} \tag{10}$$

The integral global correlation feature $\boldsymbol{Y} \in \mathbb{R}^{N \times k}$ is

$$\boldsymbol{Y} = \text{softmax}\Big((\boldsymbol{R} \circ \boldsymbol{X})^T \boldsymbol{X}\Big) g(\boldsymbol{X}) \tag{11}$$

Finally, the output $\boldsymbol{Z} \in \mathbb{R}^{N \times k}$ of rNon-local is expressed with the residual connection:

$$\boldsymbol{Z} = \boldsymbol{W}_z \boldsymbol{Y} + \boldsymbol{X}^T \tag{12}$$

where $\boldsymbol{W}_z$ is the weight matrix of the residual connection, $\boldsymbol{Y}$ is given in Equation (11), and "$+\boldsymbol{X}^T$" represents the residual connection, which allows the insertion of a new non-local block and maintains the initial network when it is a zero matrix.

2.2.3. Local Feature Extraction Block

Because the rNon-local network with the anomaly weight-map-generated block only learns the global correlation features of hyperspectral data, the Local feature learning Net, which is composed of a two- and three-dimensional convolution layer with kernel sizes of 1 × 1 and 3 × 3 × 3, respectively, is applied to further extract local features. The two-dimensional convolution layer is used to extract the spectral features of each pixel, while the three-dimensional convolution layer simultaneously extracts the spatial features of adjacent pixels and the spectral correlation of adjacent bands.

The two convolutional layers are followed by group normalization layers and the activation function before being input to the next layers. The convolutional layers' outputs in the local feature extraction block are normalized after self-multiplication:

$$\boldsymbol{y} = \text{LeakyReLU}(\text{GN}(\boldsymbol{x} \circ \boldsymbol{x})) \tag{13}$$

where $\boldsymbol{x} \circ \boldsymbol{x}$ stands for the self-multiplication of $\boldsymbol{x}$, which makes the distribution of samples much denser in the feature space; therefore, it is more conducive to the division of the hyperplane in the next step.

The three-dimensional convolutional layer is implemented on the expanded data with one extra dimension. Compared with the 3 × 3 two-dimensional convolution that is directly performed on the original data, the three-dimensional convolution can not only fuse the spatial adjacent pixels, similar to two-dimensional convolution, but it also performs the convolution on the spectral dimension to extract the spectral correlation features of adjacent bands, which cannot be achieved by the simple two-dimensional convolution.

Because the three-dimensional convolutional layer extracts the spatial features of targets by a convolution kernel with a set size, the performance of this convolution layer will be affected by the anomaly size and background complexity. The detection results of the model with a three-dimensional convolution layer, and the results of the model without it, are compared in Section 3.5.

2.3. OCSVM-Net-Based Hyperplane Layer

The final hyperplane layer and the objective function are proposed based on the OC-SVM algorithm [59], in which all data points are considered as positive instances, while the origin is the only negative instance. When the hyperspectral image $\boldsymbol{X}$ is directly used as the input of the hyperplane layer, suppose the output is the final anomaly score $A = [a_1, a_2, a_3 \cdots, a_N] \in \mathbb{R}^N$, in which the value of a is the anomaly score of each pixel. The linear decision function of this layer is defined as:

$$a_n = \boldsymbol{\omega}^T \boldsymbol{x}_n + b \tag{14}$$

where ω and b are the weight and bias parameters, respectively, and $\boldsymbol{\omega}^T \boldsymbol{x}_n + b = 0$ is the hyperplane represented by this layer.

We expected that the hyperplane would be far away from the origin, which represents the anomaly, and that as many as possible background vectors would be on the same side of the hyperplane. That is, the pixel may be a background pixel when a is negative or close to 0, while the larger the value of a is, the more likely the pixel is anomalous. The optimization function can be expressed as:

$$\min_{\boldsymbol{\omega},b} \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 + \frac{1}{v} \cdot \frac{1}{N} \sum_{n=1}^{N} \max\left(0, \boldsymbol{\omega}^T \boldsymbol{x}_n + b\right) - b \tag{15}$$

where $v \in (0,1)$ is a parameter that weighs the two problems of maximizing the distance between the hyperplane, origin, and the data allowed to pass through the hyperplane.

Equation (15) is the optimization function when the hyperplane layer is trained separately, and the optimization function of the traditional OC-SVM is

$$\min_{\boldsymbol{\omega},b} \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 + \frac{1}{v} \cdot \frac{1}{N} \sum_{n=1}^{N} \max(0, \langle \boldsymbol{\omega}, \Phi(\boldsymbol{x}_n) \rangle + b) - b \tag{16}$$

where $\Phi(\cdot)$ represents the kernel function to map the input data into a high-dimensional space, in which the background and anomalies are linearly separable. The kernel function is a fixed function, and its selection has a considerable influence on the performance of a traditional OC-SVM in anomaly detection.

Inspired by the one-class neural network (OC-NN) model [68], which designs a neural architecture based on an OC-SVM-equivalent loss function, the feature mapping method in our proposed model is a feature extraction network in place of a kernel function, which is the main difference compared with the traditional OC-SVM. Although they play the same role, the parameters in the proposed feature extraction network can be optimized because of the deep neutral network's learning ability; however, the parameters in a manually selected kernel function are fixed. Therefore, the feature extraction network is added into the optimization function of the proposed model:

$$\min_{\boldsymbol{\omega},b,\Psi} \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 + \frac{1}{v} \cdot \frac{1}{N} \sum_{n=1}^{N} \max(0, \langle \boldsymbol{\omega}, \Psi(\boldsymbol{x}_n) \rangle + b) - b \tag{17}$$

where $\Psi(\cdot)$ represents the feature extraction network, which replaces $\Phi(\cdot)$ in Equation (16) and makes it possible to leverage and refine the features learned by the extraction network.

The hyperplane layer is a two-dimensional convolutional layer with a 1×1 convolution kernel. The loss function is calculated through Equation (17):

$$L_{\mathrm{OCSVM}} = \frac{1}{2}\|\boldsymbol{\omega}\|_2^2 + \frac{1}{v} \cdot \frac{1}{N} \sum_{n=1}^{N} \max(0, \langle \boldsymbol{\omega}, \Psi(\boldsymbol{x}_n) \rangle + b) - b \tag{18}$$

Because $\boldsymbol{\omega}$, b, and Ψ are all in the form of network parameters, the error back propagation algorithm is used to minimize the loss function during training. The loss function in Equation (18) drives the optimization of the hyperplane parameters and feature extraction network parameters simultaneously, making the features obtained by the extraction network more conducive to the anomaly detector. The OCSVM loss function makes it possible for the feature extraction network to update together with the anomaly detector, i.e., OCSVM-Net.

2.4. Training Steps

The pseudocode of the SAOCNN is summarized as Algorithm 1, in which LFL-Net stands for the local feature learning network. The hyperspectral images with no or few anomalous pixels are used as the training sample. The AAE part is trained separately in the

pre-training period to make the network training more stable. Then, it is trained together with the rNon-local, LFL-Net, and OCSVM-Net. The number of times that the AAE is trained alone is AAE_epoch_num. When the number of training epochs is larger than the AAE_epoch_num, a joint training approach is carried out, which updates and iterates the OCSVM-Net with the feature extraction network. The AAE is first trained under the constraint of L_δ and L_{GAN} and then updated in the back propagation of L_{OCSVM}.

Algorithm 1: Framework of SAOCNN

Input: Training sample (original HSI) $Y = [y_1, y_2, y_3 \cdots, y_{H\times W}] \in \mathbb{R}^{H\times W\times k}$.
Output: Models E, De, Dz, rNon-local, LFL-Net, OCSVM-Net, Anomaly score map $A = [a_1, a_2, a_3 \cdots, a_{H\times W}] \in \mathbb{R}^{H\times W}$.

```
Initialize the network with random weight;
foreach epoch do
    AE update: E, De by L_δ
    Feature discriminator update: E, Dz by L_GAN
    if epoch > AAE_pre_num then
        OC-SVM update: E, De, rNon-local, LFL-Net, OCSVM-Net by L_OCSVM
    end
    Back-propagate L_δ, L_GAN, L_OCSVM
    To change E, De, Dz, rNon-local, LFL-Net, OCSVM-Net.
end
```

3. Experiments Section

We used six real hyperspectral images and five methods for comparison, including traditional anomaly detection methods, such as GRX [19], CRD [39], and LSAD [69], and two recent anomaly detection methods based on deep learning, namely, HADGAN [52] and LREN [47], to verify the performance of our proposed hyperspectral anomaly detection algorithm. Because three-dimensional convolution can play different roles according to the anomaly size in images, we also introduced another version without the use of three-dimensional convolution for spatial feature extraction named SAOCNN_NS, where "NS" stands for "no spatial features".

3.1. Datasets Description

We evaluated our proposed method by six public hyperspectral datasets, including Coast [15], Pavia [70], Washington DC Mall, HYDICE [71], Salinas, and San Diego [72].

Coast: Coast dataset was acquired by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor, which contains 100 × 100 spectral vectors, each with 207 channels, covering a range of 450 to 1350 nm. There are 141 bands left after removing some noise interference bands in the data. Buildings of different sizes in the image were marked as anomalies, and the total number of anomalous pixels was 155, accounting for 1.55%; furthermore, the number of pixels occupied by one single target was less than 20.

Pavia: Pavia dataset was acquired by Reflective Optics System Imaging Spectrometer (ROSIS) sensor over the Pavia center area. A sub-image with 100 × 100 px, 102 spectral bands in a wavelength ranging from 430 to 860 nm, and with a 1.3 m spatial resolution was selected from the hyperspectral data. The background was composed of rivers and bridges, and the anomalous targets were vehicles driving on the bridge, as well as bare soil near the bridge piers. The anomaly distribution map was obtained in the ENVI software by manual annotation, and the total number of anomalous pixels was 61, accounting for 0.61%; furthermore, the pixel number of each anomalous target ranged from 4 to 15.

Washington DC Mall: This dataset is composed of airborne hyperspectral data for Washington DC Mall. Its size is 1208 × 307 and contained 191 bands after discarding some useless bands. We selected a sub-image with 100 × 100 px where the main background

was vegetation, and the anomalous targets were the two man-made buildings in the image, occupying 7 and 12 pixels, respectively, accounting for 0.19% in total.

HYDICE urban: This dataset was collected by the Hyperspectral Digital Imagery Collection Experiment (HYDICE) airborne sensor for an urban area. A sub-image of this dataset was selected, which had 80 × 100 pixels and 162 spectral bands in a wavelength ranging from 400 to 2500 nm. The background contained vegetation, highways, and parking lots, and the image's anomalous targets were vehicles. The total number of anomalous pixels was 19, accounting for 0.24%; furthermore, the pixel number of each anomalous target ranged from 1 to 4.

Salinas Valley: This dataset was collected by the high-spatial-resolution, 224-band AVIRIS sensor over Salinas Valley, California. It had 204 bands after discarding the 20 water absorption bands. The original data size was 512 × 217, and we selected the 200 × 200 sub-image where the background was mainly various crops and roads between farmland; furthermore, the anomalous targets were man-made houses of different sizes. We obtained the anomaly distribution map in ENVI software by manual annotation, and the total number of anomalous pixels was 102, accounting for 0.22%; furthermore, the size of one single target ranged from 10 to 50 pixels.

San Diego: This dataset was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor from San Diego, CA, USA. The spatial resolution was 3.5 m per pixel. We selected the 90 × 90 sub-image, which contained 189 bands and a spectral range from 400 to 2500 nm. The background included aprons, grass, roofs, and shadows. The three aircraft were considered in the scene as anomalous targets, and they occupied 134 pixels, accounting for 1.34%; furthermore, the size of one single target ranges from 30 to 50 pixels.

In the hyperspectral image datasets above, we considered anomalous targets in San Diego and Salinas as planar targets, while smaller targets in others are considered small targets.

3.2. Evaluation Metric

We used receiver operating characteristic (ROC) curves and the area under the ROC curve (AUC) to evaluate detection results [73] and quantitatively analyze and compare the detection performance of our proposed method with other methods.Two types of ROC curves are applied in our experiment through the three parameters of P_d, P_f, and τ, which represent the true positive rate, the false positive rate, and the detection threshold, respectively. The first type is the ROC curve of (P_d, P_f), which indicates the trade-off between P_d and P_f and the closer curve (P_d, P_f) is to the upper-left corner; furthermore, the closer the area is to 1, the better the detection performance. The other type is the ROC curve of (P_f, τ), which reports P_f at each threshold, and the closer curve (P_f, τ) is to the lower-left corner, the closer the area is to 0, which means the false alarm rate is lower.

In addition, we also used box plots [74] to show the difference in the separation between the background and anomalies of each method in a more intuitive way. In the box plot, the red and green boxes, respectively, represent the anomaly data distribution and background in the detection map, where the bottom and top of each box represent 25% and 75% of the sample. Thus, a greater distance between the red and green boxes means a better separation between background and anomalies, and a narrower green box indicates a better background suppression.

3.3. Experimental Setting

As mentioned in Section 2, our proposed method consists of two parts: a feature extraction network and an OC-SVM layer. In the feature extraction network, both the E and De of the AAE are composed of four convolutional layers with kernel sizes of 1 × 1 and the number of hidden-layer spatial features set to 25; meanwhile, the Dz of the AAE contains three convolutional layers with kernel sizes of 1 × 1 and one average pooling layer. We applied a two-dimensional convolutional layer with a convolution kernel size of 1 × 1 and a three-dimensional convolution layer with a convolution kernel size of 3 × 3 × 3 in the

local feature extraction block, reducing the number of original hyperspectral image features to 100 dimensions. In the OC-SVM layer, we employed a two-dimensional convolutional layer with a convolution kernel size of 1 × 1 instead of the single fully connected network.

Adam is used to optimize the network parameters during the training process. The network learning rate is set to 10^{-5}, which decays exponentially during training. In the optimization functions, we set v in Equation (18) to 0.1 and λ in Wgan-GP (Equation (4)) to 10. We used Pytorch to implement our approach, and all the experiments were performed on a GeForce GTX 1080Ti graphics card and an Intel(R) Core (TM) i9-7920U CPU machine with 64 GB of RAM.

3.4. Experimental Results

Anomaly detection on six hyperspectral image datasets using the two proposed algorithms, namely, the SAOCNN and SAOCNN_NS, and the five contrast algorithms were performed. Figure 6 shows the anomaly detection maps of each algorithm. Figure 7 is the ROC curve (P_d, P_f) and Table 1 lists the AUC_f (AUC of curve (P_d, P_f)) for each algorithm, in which the largest AUC_f values are indicated in bold and the second largest are underlined. Figure 8 is the ROC curve (P_f, τ), and Table 2 lists the AUC_τ (AUC of curve (P_f, τ)) for each algorithm, in which the minimums are indicated in bold and the second smallest are underlined. In addition, the box plot in Figure 9 can be used to analyze the separation of the background and anomalies of each algorithm on the six datasets.

3.4.1. Performance for Small Target

The three hyperspectral image datasets of Coast, Pavia, and Washington DC Mall have small targets and simple backgrounds; therefore, the anomaly detection in these images is the simplest to perform. Figure 6a–c, shows that the targets in detection maps of our proposed methods are prominent; moreover, the suppression on the background is better. The AUC_f values of the proposed methods are approximate to the ideal value of 1 in Table 1; furthermore, they are usually the maximum or the second largest of all of the methods. In addition, the AUC_τ values of SAOCNN are about 0.001 which is smaller than other methods in Table 2 and indicates that the SAOCNN has a lower false alarm rate. Compared with CRD and LSAD, the detection results of our methods have a higher contrast between anomalies and background and have a better suppression of the background according to our visual perception of the detection maps in Figure 6 and the box plot shown in Figure 9, especially in the Pavia dataset. Compared with GRX, although the ROC curves (P_d, P_f) of SAOCNN are very close to GRX in Figure 7, the SAOCNN is faster than approach 1, and the AUC_f values of our proposed methods are slightly higher than those of GRX. Compared with LREN and HADGAN, which were recently proposed as deep learning methods, the detection maps of our proposed methods consist of fewer backgrounds and more anomalies. Moreover, the ROC curves of (P_d, P_f) for SAOCNN_NS are presented at the top-left corner, indicating that our proposed network is more effective for extracting the deep features of anomalies while suppressing the background compared with LREN and HADGAN do. Although, the ROC curves of (P_d, P_f) for HADGAN and SAOCNN_NS in the DC dataset are very close, and the former reaches 1 the fastest; it can be seen from the box plot that the green boxes of the proposed method are at a lower level, which indicates a better background suppression.

In the HYDICE dataset, the targets are smaller at a pixel level and the background is more complex compared with the Coast, Pavia, and Washington DC Mall datasets. Compared with the traditional LSAD, CRD, GRX, and the deep-learning-based HADGAN, our proposed methods produce visual results that are closer to the reference images. Moreover, the green boxes of our proposed methods are narrower in the box plot shown in Figure 9, indicating a better efficiency in background suppression. Compared with the LREN, our proposed method's background suppression is slightly inferior in detection maps. The LREN and SAOCNN are very close and lower than others—AUC_τ of 0.0088 and 0.0138 shown in Table 2. However, the ROC curves of (P_d, P_f) shown in Figure 7 demonstrate that the true positive rate P_d of SAOCNN_NS is the fastest to reach 1, while

P_d of the LREN is the last to reach 1 because some pixels are not detected. We found that the LREN may have a lower false alarm rate due to its better background suppression; however, it causes missed detection for some targets and makes the AUC_f value smaller. Our proposed methods present a superior balance between detection and false alarm rates.

Table 1. AUC_f of SAOCNN, SAOCNN_NS, HADGAN, LREN, LSAD, CRD, and GRX. (The maximum is in bold and the second largest is underlined).

	SAOCNN	SAOCNN_NS	HADGAN	LREN	LSAD	CRD	GRX
Coast	0.9962	**0.9993**	0.9925	0.9966	0.9261	0.9213	<u>0.9969</u>
Pavia	**0.9977**	<u>0.9962</u>	0.9922	0.9948	0.9541	0.9135	0.9885
DC	0.9991	<u>0.9997</u>	**0.9999**	0.9479	0.9766	0.9298	0.9989
HYDICE	0.9908	**0.9997**	0.9918	0.9779	<u>0.9929</u>	0.9868	0.9763
Salinas	**0.9990**	0.9812	0.9944	0.9702	0.9465	0.9474	<u>0.9949</u>
SanDiego	**0.9962**	<u>0.9884</u>	0.9773	0.9359	0.7645	0.7870	0.9106
Average	**0.9965**	<u>0.9941</u>	0.9913	0.9705	0.9268	0.9143	0.9777

Table 2. AUC_τ of SAOCNN, SAOCNN_NS, HADGAN, LREN, LSAD, CRD, and GRX. (The minimum is in bold and the second smallest is underlined).

	SAOCNN	SAOCNN_NS	HADGAN	LREN	LSAD	CRD	GRX
Coast	**0.0015**	0.0254	0.0133	0.0091	0.0042	<u>0.0041</u>	0.0088
Pavia	**0.0011**	<u>0.0026</u>	0.0185	0.0423	0.1485	0.1171	0.0259
DC	**0.0027**	<u>0.0240</u>	0.0395	0.1759	0.0697	0.0552	0.0452
HYDICE	<u>0.0138</u>	0.0492	0.0749	**0.0088**	0.0693	0.0351	0.0380
Salinas	<u>0.0020</u>	0.0170	**0.0016**	0.0213	0.0082	0.0102	0.0022
SanDiego	<u>0.0389</u>	0.1464	0.1122	0.1937	0.2177	0.1459	**0.0297**
Average	**0.0100**	0.0441	0.0433	0.0752	0.0863	0.0613	<u>0.0250</u>

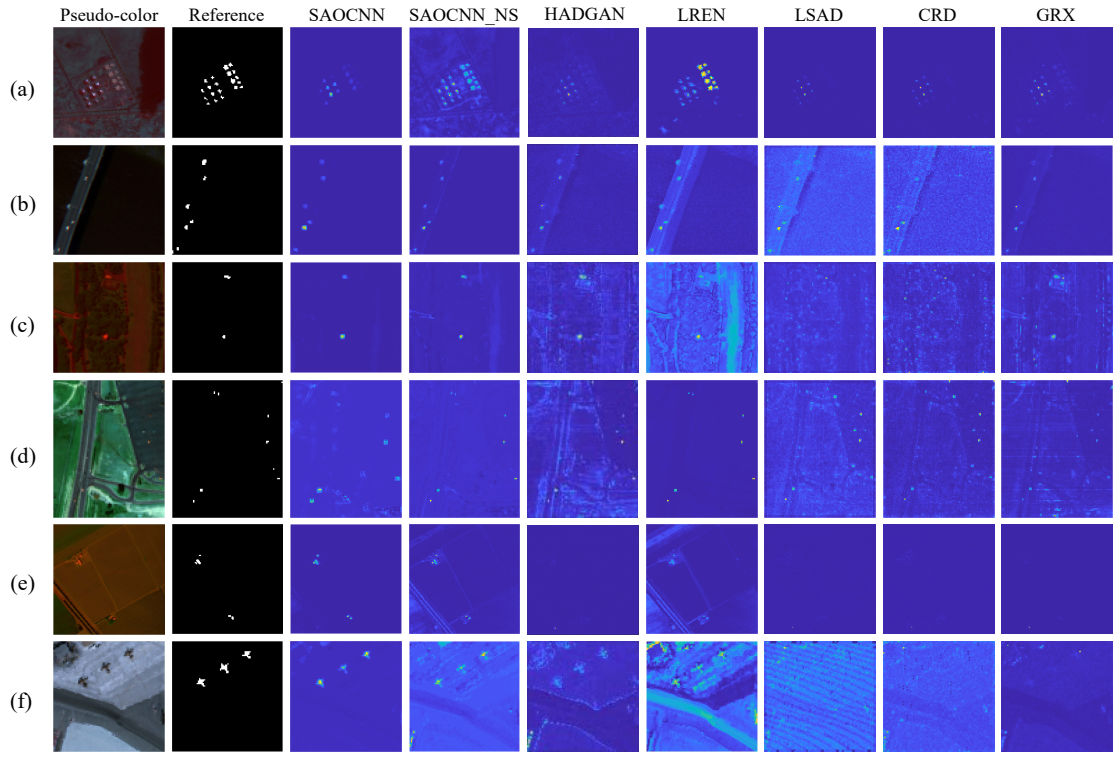


Figure 6. Anomaly detection maps of SAOCNN, SAOCNN_NS, HADGAN, LREN, LSAD, CRD, and GRX. (**a**) Coast. (**b**) Pavia. (**c**) DC Mall. (**d**) HYDICE. (**e**) Salinas. (**f**) San Diego.

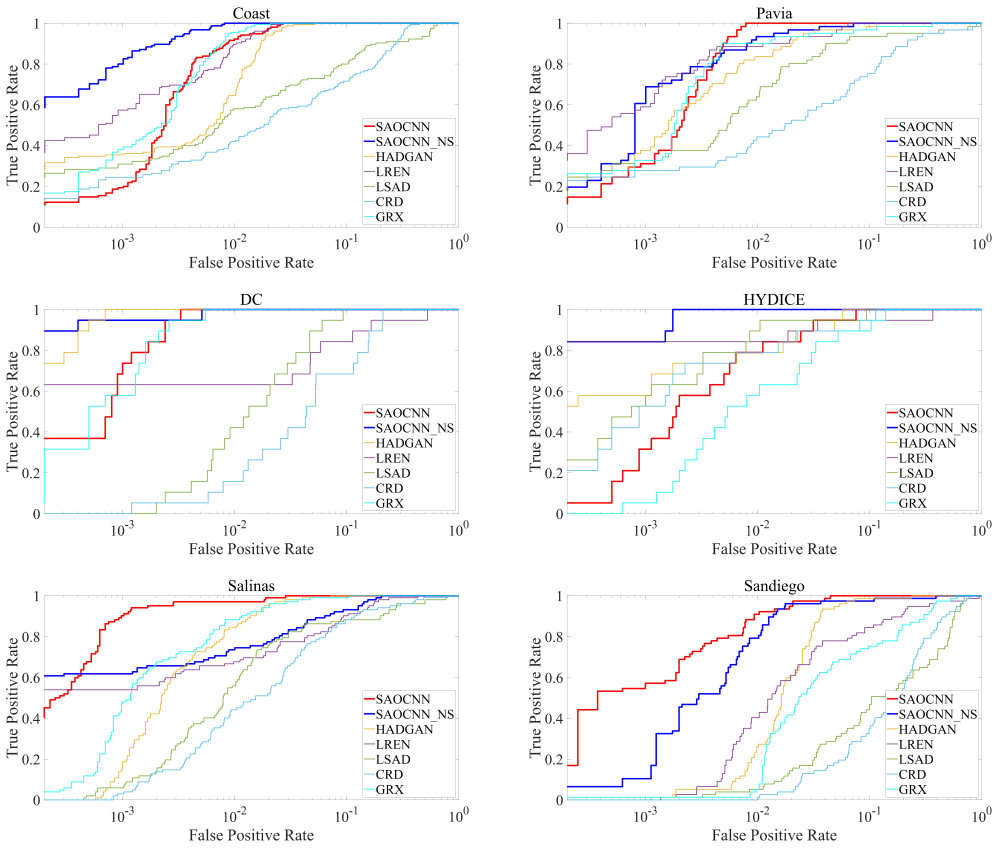


Figure 7. ROC of (P_d, P_f) for SAOCNN, SAOCNN_NS, HADGAN, LREN, LSAD, CRD, and GRX.

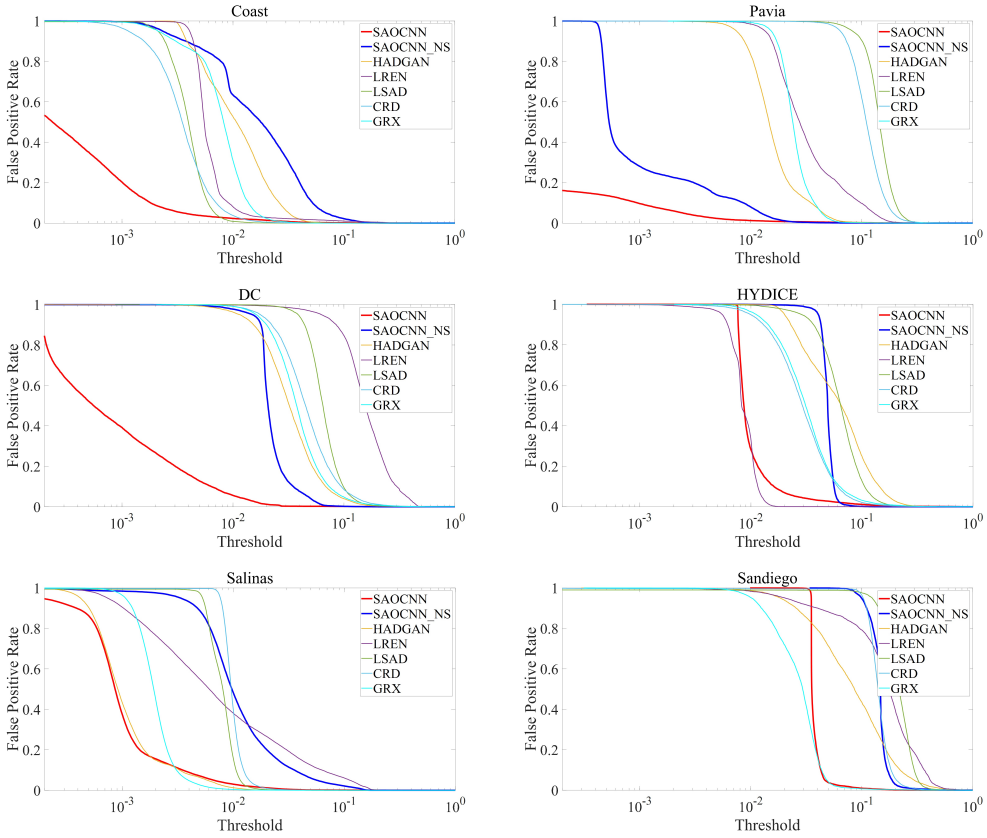


Figure 8. ROC of (P_f, τ) for SAOCNN, SAOCNN_NS, HADGAN, LREN, LSAD, CRD, and GRX.

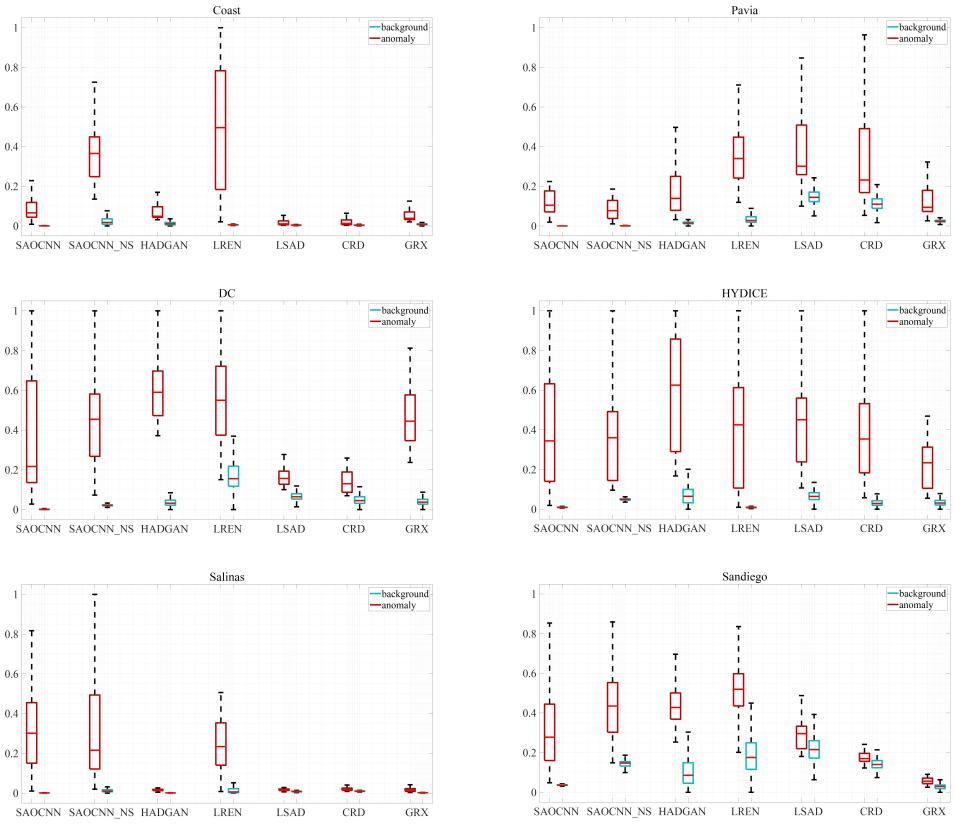


Figure 9. Box plots for SAOCNN, SAOCNN_NS, HADGAN, LREN, LSAD, CRD, and GRX.

For small targets, our methods perform well and have better background suppression rates with proper target prominence in both simple and complex backgrounds compared with other contrast methods. We compared the four ROC curves and found that curves of the proposed SAOCNN_NS are basically the closest to the top-left corner, and the proposed SAOCNN can quickly reach 1 in most of the datasets, except for HYDICE. The detection maps in Figure 6 show that targets detected by the SAOCNN are larger than the real ones annotated in reference maps, and the main reason is that a three-dimensional convolution kernel is added to the local feature extraction network. Especially in the HYDICE dataset, most of the anomalies are point targets, while the sizes of detected targets in the detection map appear to be much larger. In addition, the role of the three-dimensional convolution kernel in the network is to extract spatial features (such as texture and shape features), which the point target is lacking in. However, the background in this dataset is more complicated, with obvious spatial background features, and is therefore more easily confused with targets, which means that the SAOCNN presents a lower detection probability compared with others when the false alarm rate is low in the detection results. Without three-dimensional convolution, targets detected by the SAOCNN_NS seem to be at a proper size in Figure 6, and it presents a better balance between detection probability and false alarm rate in Figure 7. However, the SAOCNN_NS does not perform as well as the SAOCNN in background suppression, with the latter presenting a lower false alarm rate in Figure 8.

3.4.2. Performance for Planar Target

The Salinas dataset has larger targets and a simpler background. The detection results of each algorithm on the dataset are shown in Figure 6e. Targets in the detection maps of SAOCNN are the most prominent, and the SAOCNN_NS and LREN background suppression levels are not as high as SAOCNN. Targets in the detection maps of HADGAN, LSAD, CRD, and GRX are almost invisible, and there are obvious false alarm pixels. Figure 7 shows the SAOCNN ROC curve of (P_d, P_f) is at the top-left corner and is the

first one to reach 1, with the largest AUC value of 0.9990. From the box plot shown in Figure 9, the HADGAN, LSAD, CRD, and GRX red boxes and green boxes are very close, with very low positions. However, the red boxes of our proposed method are relatively high, indicating that the four contrast methods are inferior when separating anomalies from the background. Compared with the LREN, the SAOCNN_NS has a similar performance level in all aspects. However, upon adding a three-dimensional convolution kernel, the AUC_f value of SAOCNN improves by 2.968% compared with the LREN. Meanwhile, its background suppression is also better, with a narrower green box in shown in the box plot.

In the San Diego dataset, the targets are three airplanes that are not only larger but also have certain shape features, and the background is more complicated, with some roof areas that may be easily detected as anomalies compared with other datasets. From the detection results shown in Figure 6f, the targets detected by the SAOCNN are very prominent and completely retain their basic shape structures with superior background suppression. Although targets are detected in the SAOCNN_NS, the suppression of the roof areas in the background is imperfect. The HADGAN and LREN detection maps only highlight part of the anomalous pixels and preserve some background details, while targets in the LSAD, CRD, and GRX methods are almost invisible. In the ROC curves of (P_d, P_f) in Figure 7, the SAOCNN is at the top-left corner and is the fastest to approach 1, and the probability of its detection is always higher than in other methods when the false alarm rate is about 0–0.01. For AUC_f in Table 1, the SAOCNN has the largest value (0.9962), followed by the SAOCNN_NS (0.9884), which is a considerable increase compared with other methods. Figure 8 shows that the ROC curve of (P_f, τ) of GRX is slightly lower than the SAOCNN, with Table 2 showing the lowest AUC_τ value as 0.0297. This may be because the GRX green box in the box plot of Figure 9 is smaller, resulting in a lower false alarm rate at a lower threshold. However, the red box of the targets in GRX is lower and extremely close to the green box, while our proposed methods showed red and green boxes farther apart, which better distinguished the anomaly from the background.

For planar targets, the two versions of our proposed method have better target prominence and recognize the approximate shape of anomalies in the detection maps. However, the anomalous targets in other contrast methods are usually submerged in the background, which is not conducive to detection. Comparing the two proposed methods, the SAOCNN_NS does not present as well on background suppression as the SAOCNN. Especially when the background is complex, the SAOCNN has high levels of superiority because of assistance from the three-dimensional convolution kernel, which extracts the favorable spatial features of targets and inhibits the unfavorable features.

3.4.3. Detection Accuracy and Efficiency

Table 1 lists the average AUC_f for all algorithms of the six datasets. The average SAOCNN and SAOCNN_NS AUC_f values are 0.9965 and 0.9941, respectively, while the HADGAN, LREN, LSAD, CRD, and GRX values are 0.9913, 0.9705, 0.9268, 0.9143, and 0.9777, respectively, which indicates that the two versions of our method are superior. The AUC_f values of our proposed methods are close to the ideal value 1 for all datasets, even if the lowest detection accuracy is 0.9812 for the SAOCNN_NS in the Salinas dataset. Table 2 lists the average AUC_τ for all algorithms. The average minimum AUC_τ value of the SAOCNN is 0.0100, followed by GRX at 0.0250. The SAOCNN_NS is 0.0441, which is larger than the HADGAN and GRX but less than the LREN, LSAD, and CRD. In summary, the SAOCNN has superior performance for accuracy in anomaly detection, with higher detection and lower false alarm rates compared with the other methods, which demonstrates that the anomalies are sufficiently recognized and match the detection requirements.

Both the HADGAN and SAOCNN introduce the AAE to reconstruct hyperspectral data and highlight anomalies. Their difference is that the traditional anomaly detector RX is combined in the subsequent step to obtain the final anomaly detection map in the HADGAN, while we proposed OCSVM-Net to train together with a feature extraction network to obtain the results in the SAOCNN. Through the above experiments, it can

be seen that although they both effectively highlighted anomalies, the box plot showed that the HADGAN method is far less effective in suppressing the background than the SAOCNN, with obvious background texture features displayed in the detection maps. This proves that the idea of jointly training in the SAOCNN may extract deep features that are more conducive to detecting anomalies, trading a balance between abnormal prominence and background suppression.

Due to the random neural network training, we repeated our network training experiments 10 times to verify the credibility of our experimental results. The obtained results, including average and standard deviation (std), are shown in Table 3, and they prove that our proposed methods are also stable.

In addition to accuracy, the calculation efficiency is also a considerable aspect of detection performance. Table 4 records the operation time of each algorithm for six datasets. The shortest time value in the table is displayed in bold, and the second is underlined. Our proposed methods have the shortest operation times, mainly due to the fewer network layers and the acceleration of the GPU. Compared with the SAOCNN, the SAOCNN_NS has no three-dimensional convolution layer, leading to a faster calculation. Although the LREN also utilizes a deep neural network, it has the longest operation time. Networks in the LREN are only used for dimension reduction, feature extraction, and dictionary generation. However, an anomaly detector is still the traditional method based on low-rank representation and solves weight coefficients through iterative optimization, which greatly reduces the operation efficiency.

Table 3. AUC_f of SAOCNN and SAOCNN_NS in repeated experiments.

	SAOCNN		SAOCNN_NS	
	Average	**Std**	**Average**	**Std**
Coast	0.987853	0.004966	0.988417	0.017487
Pavia	0.996868	0.000658	0.991104	0.004563
DC	0.998792	0.000318	0.995962	0.003893
HYDICE	0.975760	0.026773	0.983838	0.025194
Salinas	0.997116	0.001316	0.969657	0.008905
SanDiego	0.983805	0.006330	0.969812	0.020098

Table 4. Operation time in seconds of SAOCNN, SAOCNN_NS, HADGAN, LREN, LSAD, CRD, and GRX. (The minimum is in bold and the second is underlined).

	SAOCNN	**SAOCNN_NS**	**HADGAN**	**LREN**	**LSAD**	**CRD**	**GRX**
Coast	<u>0.0229</u>	**0.0129**	1.8826	99.9463	12.4239	1.5438	0.0548
Pavia	<u>0.0189</u>	**0.0099**	1.4586	99.4378	7.0819	0.8629	0.0532
DC	0.1415	**0.0569**	2.2735	101.0966	12.9656	1.3515	<u>0.1023</u>
HYDICE	<u>0.0169</u>	**0.0079**	1.6805	73.0596	10.3797	1.1974	0.0596
Salinas	<u>0.2263</u>	**0.1117**	7.5416	384.9072	57.3966	6.5627	0.2836
SanDiego	0.1132	**0.0149**	1.9432	73.5684	47.0938	5.5229	<u>0.0666</u>

3.5. Ablation Study

To verify the effectiveness of the major components in SAOCNN, we designed a set of ablation experiments based on the benchmark OC-NN [68]. In this experiment, the following combinations are used for comparison:

- **SAOCNN**: our complete proposed method based on OC-NN, in which the non-local method and three-dimensional convolutional layer using anomalous weight map is added;
- **SAOCNN_NS**: rNon-local mechanism with an anomaly weight map is added in OC-NN, but without a three-dimensional convolutional layer to extract spatial features. "NS" stands for "No spatial feature";

- **OCNN**: the original OCNN is used as a benchmark for comparing the detection results of ablation experiments. However, the difference is that the loss function is Equation (18) and error back propagation is presented to optimize the parameters;
- **OCNN_S**: a three-dimensional convolutional layer is added to extract spatial features based on OCNN. "S" stands for "Spatial feature";
- **OCNN_NL**: the original non-local network is added to extract global features based on OCNN. "NL" stands for "non-local network";
- **SAOCNN_NA**: the original non-local network and a three-dimensional convolutional layer are added based on OCNN. Compared with the complete algorithm, the residual weight map of AAE is not used in the non-local network. "NA" stands for "No AAE".

The detection results of each module combination in the ablation experiment are shown in Figure 10. The ROC curves of (P_d, P_f) are shown in Figure 11, and the corresponding AUC values are listed in Table 5. Finally, the box plot is used in Figure 12 to show the separation between the background and anomalies on different datasets in each module combination. In addition, the repeated experiment is likewise presented in Table 6 to verify the reliability of results.

(1) AAE

The AAE is the first module in the whole network and is applied to generate the anomaly weight map to highlight the possible anomalous areas. By comparing the SAOCNN and SAOCNN_NA, we can see that the ROC curves of SAOCNN are higher than that of SAOCNN_NA and reach the ideal 1 faster (Figure 11). The average AUC_f value increased by 0.433%, which indicated that AAE's assistance can improve the detection probability of targets with a low false alarm rate.

(2) rNon-local Network

The main function of non-local network is to obtain global features by calculating the correlation between the test pixel and all other pixels in the image. In our ablation study, the OCNN_NL adds the ordinary non-local network based on the benchmark OCNN. The average AUC_f value of the OCNN is 0.029 larger than the OCNN_NL in Table 5, which indicates that the simple non-local network cannot improve the detection performance of the OCNN. Furthermore, the background suppression in the OCNN_NL detection maps is worse than that of the OCNN in Figure 10. The main reason for this may be that the non-local network enhances both the targets and background by extracting the global features of each pixel in a way that is not directed at anomalies.

Table 5. AUC_f of ablation study. (The maximum is in bold and the second largest is underlined).

	SAOCNN	SAOCNN_NS	OCNN	OCNN_S	OCNN_NL	SAOCNN_NL
Coast	0.9962	**0.9993**	<u>0.9993</u>	0.9934	0.9992	0.9959
Pavia	0.9977	0.9962	0.9967	**0.9985**	0.9946	<u>0.9978</u>
DC	0.9991	<u>0.9997</u>	0.9997	0.9993	**0.9999**	0.9991
HYDICE	0.9908	**0.9997**	0.9936	0.9877	<u>0.9977</u>	0.9693
Salinas	**0.9990**	0.9812	0.9778	0.9980	0.9588	<u>0.9981</u>
SanDiego	**0.9962**	0.9884	0.9844	0.9862	0.9842	<u>0.9927</u>
Average	**0.9965**	<u>0.9941</u>	0.9919	0.9939	0.9891	0.9922

We added the residual image reconstructed by the AAE to improve the non-local structure as rNon-local, which refers to the SAOCNN_NS in the experiment, enhancing the possible anomalous areas in the feature map produced by the non-local network. Comparing the AUC_f of the SAOCNN_NS and OCNN in Table 5, the SAOCNN_NS detection results are better compared with the OCNN on each dataset, with the average value increased by 0.222%. From the box plot in Figure 12, we observed that the distances between the red and the green boxes of the SAOCNN_NS are longer, which suggests that it is necessary to combine the AAE and rNon-local because the use of the anomaly weight map in rNon-local effectively enhances the features of anomalous areas.

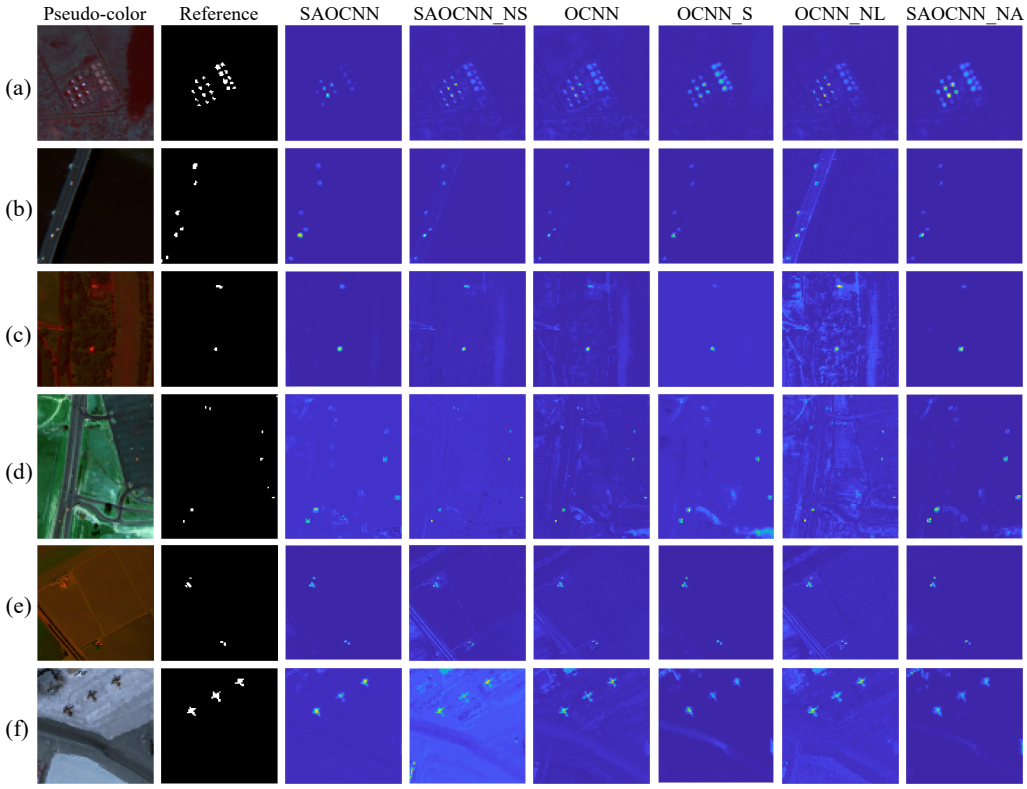


Figure 10. Anomaly detection maps of ablation study. (**a**) Coast. (**b**) Pavia. (**c**) DC Mall. (**d**) HYDICE. (**e**) Salinas (**f**) San Diego.

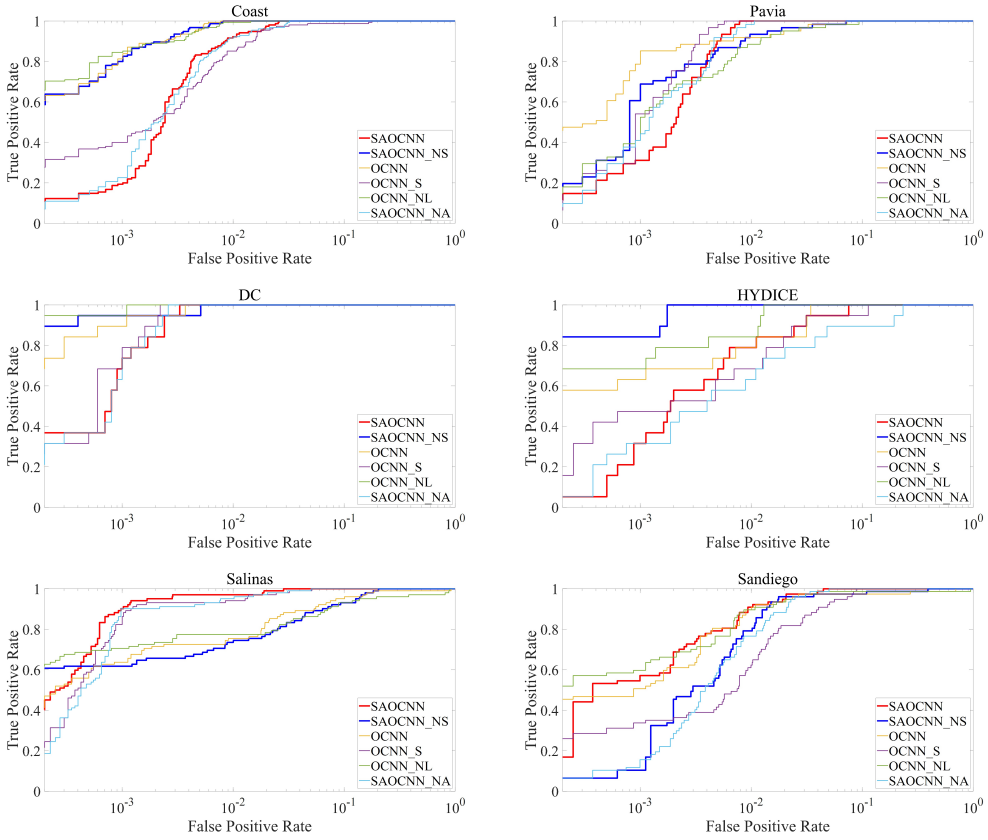


Figure 11. ROC of (P_d, P_f) for ablation study.

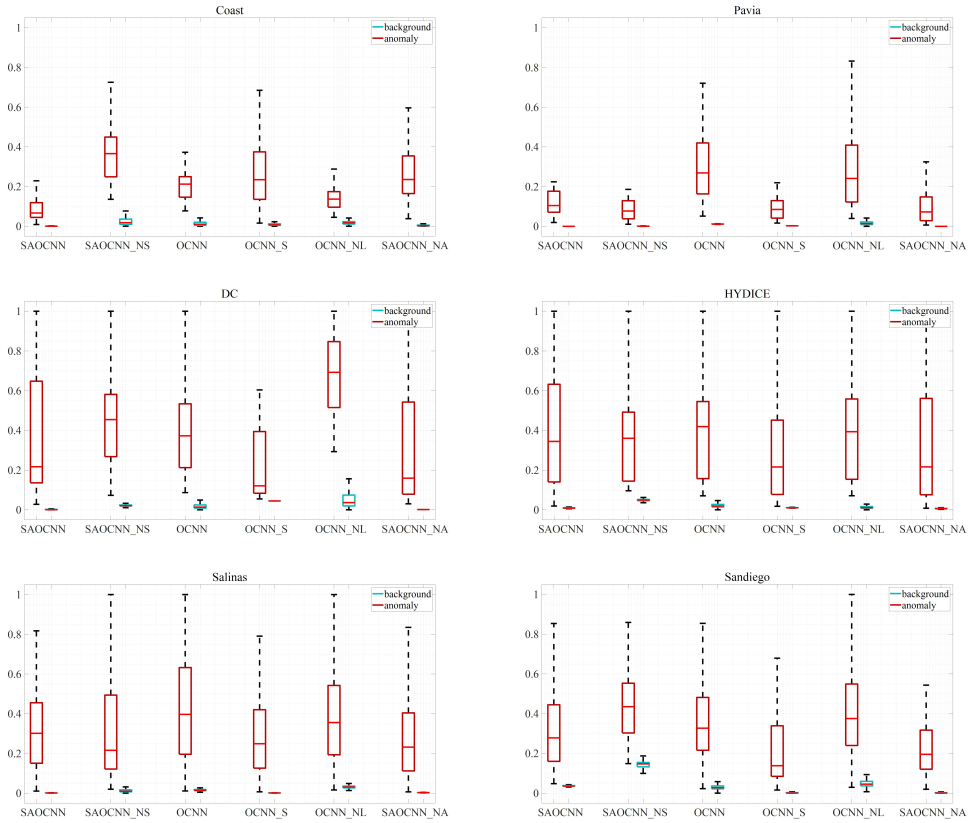


Figure 12. Box plot of ablation study.

(3) The 3D convolutional layer

The OCNN_S adds a three-dimensional convolutional layer to extract spatial features based on the OCNN. The average AUC_f value of OCNN_S is 0.202% higher than that of the OCNN in Table 5. The ROC curves of OCNN_S reach 1 faster compared with the OCNN in the four datasets shown in Figure 11, except for Coast and HYDICE. The green boxes of OCNN_S are narrower than that of the OCNN in the box plot comparison in Figure 12, which means that OCNN_S demonstrates a better background suppression than the OCNN.

In the Coast dataset, the target sizes in the detection map of OCNN_S are much larger than the real ones, causing lower AUC_f values because of the three-dimensional convolution kernel (Section 3.4.1). In the HYDICE dataset, the background contains more spatial features compared with the anomaly. The three-dimensional convolution layer used in the OCNN_S cannot extract enough spatial information from the target's very small pixels, resulting in a few false alarm targets in the detection map of OCNN_S (Figure 10d). For the datasets Salinas and San Diego with larger targets and certain spatial features, the performance of OCNN_S is considerably better than that of the OCNN. Figure 10e,f shows that targets are more prominent in the detection maps of OCNN_S; furthermore, its background suppression is also better.

It can be concluded that a three-dimensional convolutional layer is more advantageous for targets with shape structures and can suppress backgrounds that lack spatial features, thus increasing the contrast between the background and targets.

(4) Non-local network combined with 3D convolutional layer

A non-local network and a three-dimensional convolutional layer are used in the SAOCNN_NA. Table 5 shows that the optimal method in the average AUC_f is the OCNN_S, followed by the SAOCNN_NA and OCNN. The three methods' AUC_f values are very close in the Coast, Pavia, and Washington DC Mall datasets; however, the OCNN performs best in the HYDICE dataset, and the SAOCNN_NA is the best in the Salinas and San

Diego datasets. The reason may be that the ordinary non-local network amplifies both the advantages and disadvantages of the three-dimensional convolutional layer.

In summary, the non-local network and the three-dimensional convolution layer we added to the OC-NN can effectively improve anomaly detection performance; furthermore, the anomaly weight map of AAE is added to enhance the possible anomalous regions in the feature map extracted by the improved rNon-local network. Therefore, the detection performance of SAOCNN reached the optimal level, with average AUC_f values exceeding 0.464% of the benchmark OCNN, and the AUC_f is significantly increased, especially in the Salinas and San Diego datasets.

Table 6. AUC_f of OCNN, OCNN_S, OCNN_NL, and SAOCNN_NA in repeated experiments.

	OCNN		OCNN_S		OCNN_NL		SAOCNN_NA	
	Average	**Std**	**Average**	**Std**	**Average**	**Std**	**Average**	**Std**
Coast	0.978943	0.031274	0.979385	0.019868	0.990201	0.018966	0.957941	0.028588
Pavia	0.964453	0.036719	0.983363	0.017115	0.970988	0.021899	0.953301	0.020565
DC	0.995598	0.005574	0.965748	0.068885	0.994500	0.008199	0.996005	0.003954
HYDICE	0.966670	0.033513	0.936631	0.019346	0.992880	0.007649	0.920196	0.013655
Salinas	0.975203	0.006805	0.995252	0.007107	0.962751	0.021042	0.994950	0.002550
SanDiego	0.953151	0.011778	0.953332	0.017886	0.968287	0.016956	0.970973	0.012985

4. Discussion

We carried out a training analysis with four scenarios to verify the effectiveness of the training tricks in our proposed methods, including adversarial learning and joint training approaches. The AUC_f values for each dataset are shown in Table 7. Figure 13 displays the ROC curves (P_d, P_f), and Figure 14 shows the box plots for each scenario. In addition, the repeated experiment is also presented in Table 8.

- **AAE+OCSVM**: our proposed method with adversarial learning (i.e., an additional discriminator Dz) and joint training for the whole network. "+OCSVM" stands for the rest part, except for the anomaly weight-map-generated block;
- **AAE+OCSVM(se)**: a two-step training version of our proposed network. The AAE is first trained with L_δ and L_{GAN} separately, and then the trained AAE is fixed and used to help train the rest part with L_{OCSVM}. "se" stands for separate;
- **AE+OCSVM**: the AE is used to generate the anomaly weight map with the basic encoder–decoder structure and loss function L_δ;
- **AE+OCSVM(se)**: a two-step training version of the third scenario.

The four schemes are trained in identical epochs to assure the rationality of our experiment. In the two-step approaches, the AAE or AE are trained in 500 epochs separately, and the subsequent parts are trained in additional 500 epochs with the fixed AAE or AE. The co-training approaches train in 500 epochs for the whole network.

(1) Adversarial learning method

As discussed in Section 2.2.1, the adversarial training approach is presented in an anomaly weight-map-generated block to highlight the anomalous area in a residual image, with an additional discriminator Dz that helps constrain the hidden feature $\boldsymbol{z}$ of AE to follow the background distribution, which is concerned to obey Gaussian probability. We compared scenario (A) and (C) to validate the effectiveness of the adversarial learning method. Table 7 shows that the average value of AUC_f is improved by 0.321% when adding the adversarial training. The AAE obtains better detection results of up 1.064%, especially for the San Diego dataset with planer targets and complex backgrounds.

Although the AUC_f values do not significantly increase, we noticed by comparing ROC curves in Figure 13 that the red curves are at the upper left of the yellow curves in most datasets, and the detection probability is much higher when the false alarm rate is about $0–10^{-2}$. For the Pavia dataset, the AE performed better than AAE in ROC curves; however, we found that the red box of AAE is farther from the green box than that of

AE from the box plot comparison in Figure 14. This suggests that the AAE highlighted anomalies better, although it is not obvious from the AUC_f perspective.

In conclusion, our comparison results indicate that additional adversarial training helps yield a better anomaly weight map, which highlights the possible anomalous areas and improves the final detection performance.

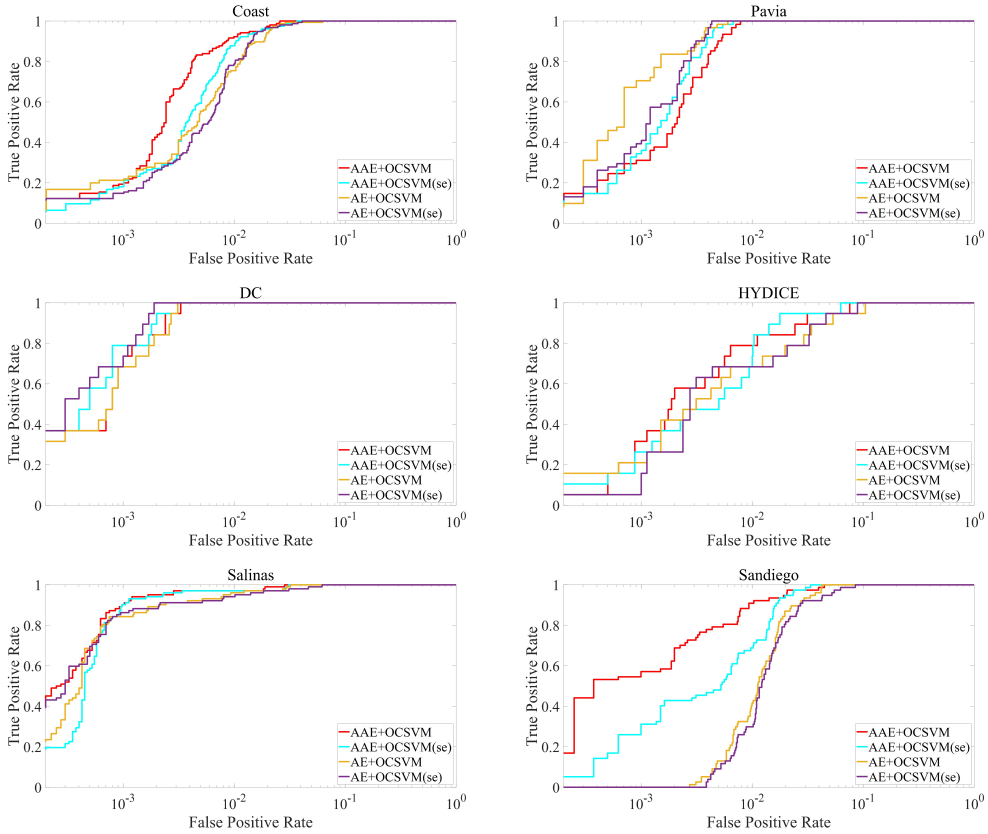


Figure 13. ROC of (P_d, P_f) for training study.

Table 7. AUC_f of training study. (The maximum is in bold and the second largest is underlined).

	AAE+OCSVM	AAE+OCSVM(se)	AE+OCSVM	AE+OCSVM(se)
Coast	**0.9962**	0.9946	0.9932	0.9931
Pavia	0.9977	0.9982	**0.9989**	0.9985
DC	0.9991	0.9993	0.9990	**0.9994**
HYDICE	0.9908	**0.9914**	0.9852	0.9861
Salinas	**0.9990**	0.9986	0.9983	0.9977
SanDiego	**0.9962**	0.9928	0.9856	0.9839
Average	**0.9965**	0.9959	0.9933	0.9931

Table 8. AUC_f of OCNN, OCNN_S, OCNN_NL, and SAOCNN_NA in repeated experiments.

	AAE+OCSVM(se)		AE+OCSVM		AE+OCSVM(se)	
	Average	Std	Average	Std	Average	Std
Coast	0.991742	0.005183	0.989259	0.003943	0.992701	0.000304
Pavia	0.996883	0.003059	0.997840	0.000697	0.997279	0.002112
DC	0.999121	0.000133	0.998961	0.000422	0.999260	0.000136
HYDICE	0.985342	0.007138	0.975644	0.005904	0.981095	0.005834
Salinas	0.997680	0.001136	0.996643	0.002436	0.996969	0.000538
SanDiego	0.987765	0.003566	0.982193	0.001865	0.982184	0.001213

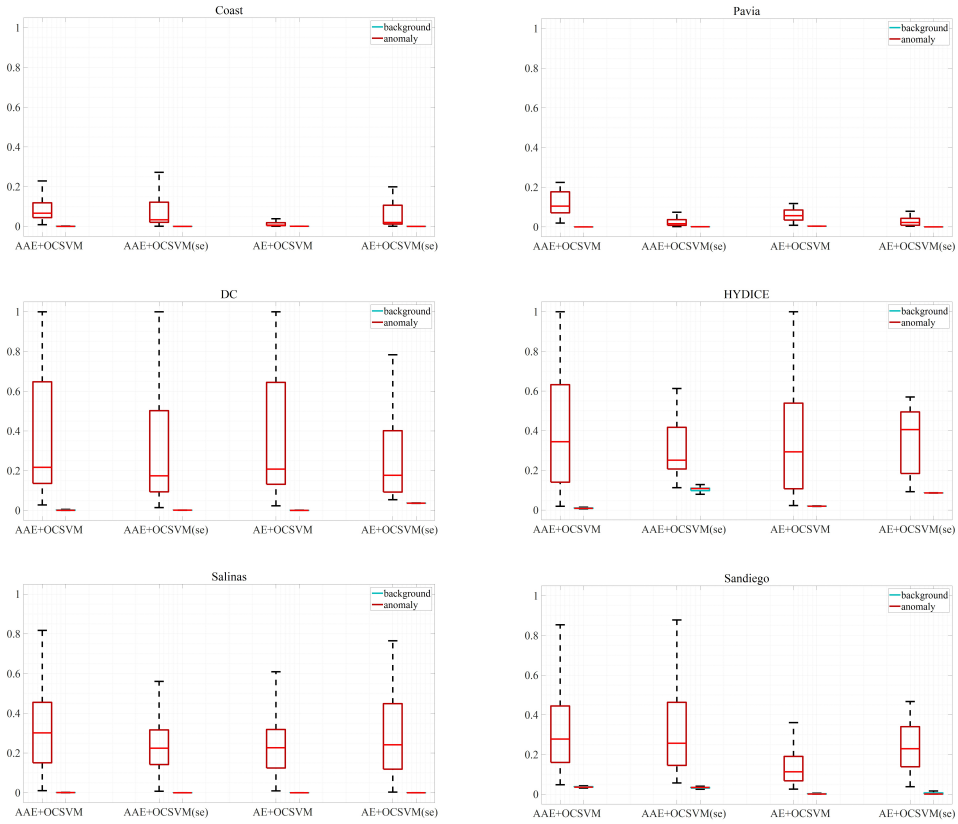


Figure 14. Box plot of training study.

(2) Joint training method

To better demonstrate the superiority of joint optimizing the feature extractor and subsequent anomaly detector in our proposed method, we present a two-step training version of our proposed network for comparison. As described in Section 2.4, when training the whole SAOCNN, the AAE should be trained separately for the AAE_pre_num epochs at first, and then the subsequent network, including the OCSVM layers, can be updated together with the feature extraction network through L_{OCSVM}. Such joint training drives the feature extractor to learn additional deep details for specifically anomaly detection tasks, which cannot be achieved with the two-step training method.

In our experiment, we used the two-step training version for both scenario (A) and (C). Figure 7 shows that the joint training methods demonstrate superior performance than the related separate training methods. The average AUC_f values are increased by 0.060% and 0.020% for the AAE and AE versions, respectively, when implementing joint training. For our proposed method, the AUC_f values increased by 0.342% and 0.343% in the Coast and San Diego datasets, respectively. The ROC curves in Figure 13 show that the red curves, which represent our proposed method, are located in the upper-left corner in these two datasets. This shows that the joint training approaches present a prime balance between high detection and low false alarm rates. For other datasets, although the ROC curves of the two types are close, the comparison of the box plot in Figure 14 shows that the red boxes of the co-training approaches are farther from the green boxes than that of the two-step training approach, which means the joint training is capable of better highlighting anomalies. In the HYDICE dataset, co-training approaches can also suppress the background better because the green box of the two joint training methods is much closer to 0 in the box plot.

The overall results show that the L_{OCSVM} joint training approach in our proposed methods help the whole SAOCNN to extract the relevant anomaly features compared with the two-step training approach. In the HADGAN [52] and SAFL [53] methods, the feature

extractor and anomaly detector are separated, and the latter makes no contribution to constraining the feature extraction network. This two-step training approach is necessary for these methods because the anomaly detector still adopts the traditional method. However, in our method, a one-class network is applied for anomaly detection and updated together with the whole network through L_{OCSVM}. The joint training method is not only more concise but also enables the anomaly detector to better guide the feature extraction network and make the anomalies more prominent while suppressing the background in the detection maps.

5. Conclusions

In our study, we proposed an unsupervised deep learning model (SAOCNN) for hyperspectral anomaly detection. The advantage of an SAOCNN is that a one-class objective function trains a neural network that maps hyperspectral data to a feature space specially constructed for an anomaly detection task. To capture rich features in hyperspectral images, the improved residual non-local network with an added adversarial autoencoder extracts global information and enhances possible anomalous regions, while the local feature extraction network is sufficient to obtain local spatial information. SAOCNN adopts a training mode to jointly optimize a feature extractor and anomaly detector and accurately separates anomalies from the background according to the hyperplane learned in the feature space. After conducting experiments on six public hyperspectral datasets. It can be concluded that SAOCNN outperforms state-of-the-art methods with higher accuracy and lower false alarm rates, in addition to having faster calculation speeds, whether from a qualitative or quantitative perspective. In the future, we may try new deep learning techniques, such as Transformer, in our proposed unsupervised model to improve its performance.

Author Contributions: Conceptualization, J.W. and T.O.; methodology, J.W. and T.O.; software, T.O.; validation, J.W., T.O., Y.D. and L.C.; formal analysis, T.O. and Y.D.; investigation, T.O. and Y.D.; resources, J.W. and T.O.; data curation, T.O.; writing—original draft preparation, Y.D.; writing—review and editing, J.W. and Y.D.; visualization, Y.D.; supervision, J.W. and L.C.; project administration, J.W.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Fundamental Research Funds for the Central Universities.

Data Availability Statement: Publicly available datasets were analyzed in this study. The Coast dataset is available at: http://xudongkang.weebly.com/ (accessed on 27 September 2021). The Pavia and Salinas datasets are available at: https://www.ehu.eus/ccwintco/ (accessed on 23 September 2021). The Washington DC Mall dataset is publicly available at: https://engineering.purdue.edu/~biehl/MultiSpec/ (accessed on 28 September 2021). The HYDICE dataset is publicly available at: https://sites.google.com/site/feiyunzhuhomepage/ (accessed on 13 September 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Borengasser, M.; Hungate, W.S.; Watkins, R. *Hyperspectral Remote Sensing: Principles and Applications*; CRC Press: Boca Raton, FL, USA, 2007.
2. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 120–147. [CrossRef]
3. Tuia, D.; Flamary, R.; Courty, N. Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 272–285. [CrossRef]
4. Su, H.; Yu, Y.; Du, Q.; Du, P. Ensemble learning for hyperspectral image classification using tangent collaborative representation. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3778–3790. [CrossRef]
5. Wu, C.; Du, B.; Zhang, L. A subspace-based change detection method for hyperspectral images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 815–830. [CrossRef]
6. Liu, S.; Bruzzone, L.; Bovolo, F.; Du, P. Hierarchical unsupervised change detection in multitemporal hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 244–260.
7. Nasrabadi, N.M. Hyperspectral target detection: An overview of current and future challenges. *IEEE Signal Process. Mag.* **2013**, *31*, 34–44. [CrossRef]

8. Axelsson, M.; Friman, O.; Haavardsholm, T.V.; Renhorn, I. Target detection in hyperspectral imagery using forward modeling and in-scene information. *ISPRS J. Photogramm. Remote Sens.* **2016**, *119*, 124–134. [CrossRef]
9. Jiao, C.; Chen, C.; McGarvey, R.G.; Bohlman, S.; Jiao, L.; Zare, A. Multiple instance hybrid estimator for hyperspectral target characterization and sub-pixel target detection. *ISPRS J. Photogramm. Remote Sens.* **2018**, *146*, 235–250. [CrossRef]
10. Campbell, J.B.; Wynne, R.H. *Introduction to Remote Sensing*; Guilford Press: New York, NY, USA, 2011.
11. Chang, C.I.; Chiang, S.S. Anomaly detection and classification for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 1314–1325. [CrossRef]
12. Li, Z.; Zhang, Y. A new hyperspectral anomaly detection method based on higher order statistics and adaptive cosine estimator. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 661–665.
13. Li, J.; Zhang, H.; Zhang, L.; Ma, L. Hyperspectral anomaly detection by the use of background joint sparse representation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2523–2533.
14. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–58.
15. Kang, X.; Zhang, X.; Li, S.; Li, K.; Li, J.; Benediktsson, J.A. Hyperspectral anomaly detection with attribute and edge-preserving filters. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5600–5611.
16. Matteoli, S.; Diani, M.; Theiler, J. An overview of background modeling for detection of targets and anomalies in hyperspectral remotely sensed imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2317–2336. [CrossRef]
17. Su, H.; Wu, Z.; Zhu, A.X.; Du, Q. Low rank and collaborative representation for hyperspectral anomaly detection via robust dictionary construction. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 195–211. [CrossRef]
18. Su, H.; Wu, Z.; Zhang, H.; Du, Q. Hyperspectral anomaly detection: A survey. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 64–90.
19. Reed, I.S.; Yu, X. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution. *IEEE Trans. Acoust. Speech Signal Process.* **1990**, *38*, 1760–1770.
20. Borghys, D.; Kåsen, I.; Achard, V.; Perneel, C. Comparative evaluation of hyperspectral anomaly detectors in different types of background. In Proceedings of the Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII, Baltimore, MD, USA, 23–27 April 2012; Volume 8390, pp. 803–814.
21. Kwon, H.; Nasrabadi, N.M. Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 388–397. [CrossRef]
22. Matteoli, S.; Diani, M.; Corsini, G. Improved estimation of local background covariance matrix for anomaly detection in hyperspectral images. *Opt. Eng.* **2010**, *49*, 46201. [CrossRef]
23. Guo, Q.; Zhang, B.; Ran, Q.; Gao, L.; Li, J.; Plaza, A. Weighted-RXD and linear filter-based RXD: Improving background statistics estimation for anomaly detection in hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2351–2366. [CrossRef]
24. Tax, D.M.; Duin, R.P. Support vector data description. *Mach. Learn.* **2004**, *54*, 45–66. [CrossRef]
25. Banerjee, A.; Burlina, P.; Diehl, C. A support vector method for anomaly detection in hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 2282–2291. [CrossRef]
26. Carlotto, M.J. A cluster-based approach for detecting man-made objects and changes in imagery. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 374–387. [CrossRef]
27. Hytla, P.C.; Hardie, R.C.; Eismann, M.T.; Meola, J. Anomaly detection in hyperspectral imagery: Comparison of methods using diurnal and seasonal data. *J. Appl. Remote Sens.* **2009**, *3*, 033546. [CrossRef]
28. West, D.B. *Introduction to Graph Theory*; Prentice Hall: Upper Saddle River, NJ, USA, 2001; Volume 2.
29. Messinger, D.W.; Albano, J. A graph theoretic approach to anomaly detection in hyperspectral imagery. In Proceedings of the 2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Lisbon, Portugal, 6–9 June 2011; pp. 1–4.
30. Song, S.; Yang, Y.; Zhou, H.; Chan, J.C.W. Hyperspectral Anomaly Detection via Graph Dictionary-Based Low Rank Decomposition with Texture Feature Extraction. *Remote Sens.* **2020**, *12*, 3966.
31. Elad, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 2.
32. Zhang, L.; Yang, M.; Feng, X. Sparse representation or collaborative representation: Which helps face recognition? In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 471–478.
33. Liu, G.; Lin, Z.; Yu, Y. Robust subspace segmentation by low-rank representation. In Proceedings of the International Conference on International Conference on Machine Learning Citeseer, Haifa, Israel, 21–24 June 2010; Volume 1, p. 8.
34. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 210–227. [CrossRef]
35. Huyan, N.; Zhang, X.; Zhou, H.; Jiao, L. Hyperspectral anomaly detection via background and potential anomaly dictionaries construction. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2263–2276. [CrossRef]
36. Li, F.; Zhang, X.; Zhang, L.; Jiang, D.; Zhang, Y. Exploiting structured sparsity for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4050–4064. [CrossRef]
37. Soofbaf, S.R.; Sahebi, M.R.; Mojaradi, B. A sliding window-based joint sparse representation (swjsr) method for hyperspectral anomaly detection. *Remote Sens.* **2018**, *10*, 434. [CrossRef]
38. Zhang, L.; Yang, M.; Feng, X.; Ma, Y.; Zhang, D. Collaborative representation based classification for face recognition. *arXiv* **2012**, arxiv:1204.2358.

39. Vafadar, M.; Ghassemian, H. Hyperspectral anomaly detection using outlier removal from collaborative representation. In Proceedings of the 2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA), Shahrekord, Iran, 19–20 April 2017; pp. 13–19.
40. Su, H.; Wu, Z.; Du, Q.; Du, P. Hyperspectral anomaly detection using collaborative representation with outlier removal. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 5029–5038. [CrossRef]
41. Tu, B.; Li, N.; Liao, Z.; Ou, X.; Zhang, G. Hyperspectral anomaly detection via spatial density background purification. *Remote Sens.* **2019**, *11*, 2618. [CrossRef]
42. Li, W.; Du, Q. Collaborative representation for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 1463–1474.
43. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 171–184. [CrossRef]
44. Zhang, Y.; Du, B.; Zhang, L.; Wang, S. A low-rank and sparse matrix decomposition-based Mahalanobis distance method for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2015**, *54*, 1376–1389.
45. Qu, Y.; Wang, W.; Guo, R.; Ayhan, B.; Kwan, C.; Vance, S.; Qi, H. Hyperspectral Anomaly Detection through Spectral Unmixing and Dictionary-Based Low-Rank Decomposition. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4391–4405. doi: 10.1109/TGRS.2018.2818159. [CrossRef]
46. Li, L.; Li, W.; Du, Q.; Tao, R. Low-rank and sparse decomposition with mixture of Gaussian for hyperspectral anomaly detection. *IEEE Trans. Cybern.* **2020**, *51*, 4363–4372. [CrossRef]
47. Jiang, K.; Xie, W.; Lei, J.; Jiang, T.; Li, Y. LREN: Low-rank embedded network for sample-free hyperspectral anomaly detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 4139–4146.
48. Hinton, G.E.; Zemel, R. Autoencoders, minimum description length and Helmholtz free energy. *Adv. Neural Inf. Process. Syst.* **1993**, *6*. Available online: https://proceedings.neurips.cc/paper/1993/hash/9e3cfc48eccf81a0d57663e129aef3cb-Abstract.html (accessed on 12 September 2022).
49. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
50. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
51. Arisoy, S.; Nasrabadi, N.M.; Kayabol, K. GAN-based hyperspectral anomaly detection. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Online, 18–22 January 2021; pp. 1891–1895.
52. Jiang, T.; Li, Y.; Xie, W.; Du, Q. Discriminative reconstruction constrained generative adversarial network for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4666–4679. [CrossRef]
53. Xie, W.; Liu, B.; Li, Y.; Lei, J.; Chang, C.I.; He, G. Spectral adversarial feature learning for anomaly detection in hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2352–2365. [CrossRef]
54. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (accessed on 12 September 2022).
55. Ruff, L.; Zemlyanskiy, Y.; Vandermeulen, R.; Schnake, T.; Kloft, M. Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4061–4071.
56. Purwanto, D.; Chen, Y.T.; Fang, W.H. Dance with self-attention: A new look of conditional random fields on anomaly detection in videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 173–183.
57. Wu, K.; Zhu, L.; Shi, W.; Wang, W.; Wu, J. Self-Attention Memory-Augmented Wavelet-CNN for Anomaly Detection. *IEEE Trans. Circuits Syst. Video Technol.* 2022, *in press*. [CrossRef]
58. Jiang, H. A Manifold Constrained Multi-Head Self-Attention Variational Autoencoder Method for Hyperspectral Anomaly Detection. In Proceedings of the 2021 International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA), Huaihua, China, 10–12 December 2021; pp. 11–17.
59. Schölkopf, B.; Smola, A.J.; Bach, F. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2002.
60. Cheng, T.; Wang, B. Graph and total variation regularized low-rank representation for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 391–406. [CrossRef]
61. Wu, Z.; Zhu, W.; Chanussot, J.; Xu, Y.; Osher, S. Hyperspectral anomaly detection via global and local joint modeling of background. *IEEE Trans. Signal Process.* **2019**, *67*, 3858–3869. [CrossRef]
62. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.J. Adversarial Autoencoders. *arXiv* **2015**, arXiv:1511.05644
63. Huber, P.J. Robust estimation of a location parameter. In *Breakthroughs in Statistics*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 492–518.
64. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: https://proceedings.neurips.cc/paper/2017/hash/892c3b1c6dccd52936e27cbd0ff683d6-Abstract.html (accessed on 12 September 2022).

65. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
66. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the International Conference on International Conference on Machine Learning Citeseer, Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.
67. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
68. Chalapathy, R.; Menon, A.K.; Chawla, S. Anomaly detection using one-class neural networks. *arXiv* **2018**, arxiv:1802.06360.
69. Tan, K.; Hou, Z.; Wu, F.; Du, Q.; Chen, Y. Anomaly detection for hyperspectral imagery based on the regularized subspace method and collaborative representation. *Remote Sens.* **2019**, *11*, 1318. [CrossRef]
70. Billor, N.; Hadi, A.S.; Velleman, P.F. BACON: Blocked adaptive computationally efficient outlier nominators. *Comput. Stat. Data Anal.* **2000**, *34*, 279–298.
71. Zhu, F.; Wang, Y.; Xiang, S.; Fan, B.; Pan, C. Structured sparse method for hyperspectral unmixing. *ISPRS J. Photogramm. Remote Sens.* **2014**, *88*, 101–118.
72. Xu, Y.; Wu, Z.; Li, J.; Plaza, A.; Wei, Z. Anomaly detection in hyperspectral images based on low-rank and sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2015**, *54*, 1990–2000.
73. Kerekes, J. Receiver operating characteristic curve confidence intervals and regions. *IEEE Geosci. Remote Sens. Lett.* **2008**, *5*, 251–255. [CrossRef]
74. Williamson, D.F.; Parker, R.A.; Kendrick, J.S. The box plot: A simple visual method to interpret data. *Ann. Intern. Med.* **1989**, *110*, 916–921. [PubMed]