remote sensing

MDPI

Article

CRNN: Collaborative Representation Neural Networks for Hyperspectral Anomaly Detection

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

[1] Department of Aerospace Information Engineering, Beihang University, Beijing 100191, China; 17375124@buaa.edu.cn
[2] Wuhan Digital Engineering Institute, Wuhan 430205, China; oytb@buaa.edu.cn
* Correspondence: wangjinshen@buaa.edu.cn
† These authors contributed equally to this work.

Abstract: Hyperspectral anomaly detection aims to separate anomalies and backgrounds without prior knowledge. The collaborative representation (CR)-based hyperspectral anomaly detection methods have gained significant interest and development because of their interpretability and high detection rate. However, the traditional CR presents a low utilization rate for deep latent features in hyperspectral images, making the dictionary construction and the optimization of weight matrix sub-optimal. Due to the excellent capacity of neural networks for generation, we formulate the deep learning-based method into CR optimization in both global and local streams, and propose a novel hyperspectral anomaly detection method based on collaborative representation neural networks (CRNN) in this paper. In order to gain a complete background dictionary and avoid the pollution of anomalies, the global dictionary is collected in the global stream by optimizing the dictionary atom loss, while the local background dictionary is obtained by using a sliding dual window. Based on the two dictionaries, our two-stream networks are trained to learn the global and local representation of hyperspectral data by optimizing the objective function of CR. The detection result is calculated by the fusion of residual maps of original and represented data in the two streams. In addition, an autoencoder is introduced to obtain the hidden feature considered as the dense expression of the original hyperspectral image, and a feature extraction network is concerned to further learn the comprehensive features. Compared with the shallow learning CR, the proposed CRNN learns the dictionary and the representation weight matrix in neural networks to increase the detection performance, and the fixed network parameters instead of the complex matrix operations in traditional CR bring a high inference efficiency. The experiments on six public hyperspectral datasets prove that our proposed CRNN presents the state-of-the-art performance.

Keywords: collaborative representation; autoencoder; anomaly detection; hyperspectral image

Citation: Duan, Y.; Ouyang, T.; Wang, J. CRNN: Collaborative Representation Neural Networks for Hyperspectral Anomaly Detection. *Remote Sens.* **2023**, *15*, 3357. https://doi.org/10.3390/rs15133357

Academic Editors: Chein-I Chang, Shengwei Zhong and Shuhan Chen

Received: 15 May 2023
Revised: 26 June 2023
Accepted: 29 June 2023
Published: 30 June 2023

1. Introduction

Hyperspectral remote sensing integrates two sensing technologies of panchromatic imaging and spectrometry to fully exploit the spatial and spectral features [1]. Hyperspectral images record nearly continuous spectral bands and better reflect the physical properties of the objects, which helps to identify subtle differences between various materials. Therefore, they are widely employed in both civil and military fields, such as image classification [2–4] and target detection [5–7]. As one of the vital research branches of target detection, hyperspectral anomaly detection distinguishes anomalies from background in unsupervised way for lack of prior knowledge [8]. Due to the challenges in establishing a comprehensive and accurate spectrum database for targets, hyperspectral anomaly detection has received a lot of interest and has great practical prospects [9].

In hyperspectral anomaly detection, the anomaly is typically considered as the target with a notably different spectral features from the surrounding background pixels [1,10].

In real hyperspectral images, it is difficult to consider all background clusters since the background occupies most of the pixels and has a complex composition [11]. Moreover, hyperspectral images are highly unbalanced, and the occurrence probability of anomalies is low, which makes it impossible to extract the intrinsic features of anomalies. Therefore, the main challenge in hyperspectral anomaly detection is to effectively suppress the background while distinguishing the anomalies.

The existing anomaly detection methods exploit the property of anomalies with low probability and high deviation from the surrounding background cluster to distinguish anomalies from background [10,12]. The most widely known benchmark algorithm is the Reed–Xiaoli (RX) detector [13]. RX utilizes the whole image to build background model under the presumption that the background is distributed according to a multivariate Gaussian distribution. The probability of anomalies is measured by the Mahalanobis distance between the test pixel and the background. The statistical theory-based hyperspectral anomaly detection methods use a simple mathematical model and have low computational consumption. Subsequently, researchers have proposed many improved algorithms based on RX, such as Kernel RX [14], Local RX [15], Dual-Window RX [16], and Weighted RX [17]. In addition to the generalized likelihood ratio test-based RX, Chang [9] developed dummy variable trick (DVT) to convert hyperspectral target detection to anomaly detection, and also introduced effective anomaly space (EAS) [18] to remove the background while retain the anomaly. Cluster-based hyperspectral anomaly detectors [19,20], the support vector data description (SVDD) [21], and graph-based methods [22,23] take the distance to measure the anomaly score. Ref. [24] introduced a fractional Fourier entropy (FrFE)-based hyperspectral anomaly detection method to distinguish anomalies from background. To utilize the spatial features in the hyperspectral data, an attribute and edge-preserving filtering detector (AED) [25] and other filtering-based methods [26,27] were proposed. Tensor-based methods treat the hyperspectral cube as a third-order tensor and utilize the spectral and spatial information [28]. A prior-based tensor approximation (PTA) method for hyperspectral anomaly detection is proposed in [29], which decomposed the hyperspectral data into a background tensor and an anomaly tensor under a low-rank and a piecewise-smooth prior. Iterative spectral–spatial hyperspectral anomaly detector (ISSHAD) [30] extracted spectral–spatial information via anomaly detection maps and gave feedback to create a new expanded data cube in iteration.

The representation-based methods, including sparse representation (SR) [31–33], collaborative representation (CR) [34–40], and low-rank representation (LRR) [41–45], try to use a linear combination of atoms in the constructed background dictionary to represent the hyperspectral images, and assume that the background pixels will be represented better than anomalous pixels since the clusters of anomalies are far different from the background clusters. Therefore, representation-based methods usually use the residual map of the represented image and the original image to detect the anomaly.

The SR-based approaches assume that only a small subset of atoms in the given over-complete dictionary contribute to the representation under the sparsity constraint. Li et al. [32] adopted principal component analysis (PCA) [46] to learn a robust background dictionary and proposed a SR model based on the re-weighted Laplace prior. The CR-based methods utilize all of the background dictionary atoms to represent each pixel with the l_2-norm [47] to constrain the representation coefficients for better background modeling capability [34]. The CR-based methods have received many developments due to their computational simplicity and good performance, but the pollution of anomalous atom caused by the sliding dual window limits the performance of the method. To solve this issue, researchers have devised lots of methods. Vafadar and Ghassemian [35] removed anomalous pixels that deviate noticeably from the distribution of most pixels when constructing the background dictionary. Other methods were also tried in [36–39] to exclude potential anomalous pixels and obtain a pure background. In contrast to the popular dual window strategy, Wu et al. [40] introduced a practical non-global dictionary and proposed a new relaxed CR hyperspectral anomaly detector. The LRR-based approaches divide the

hyperspectral images into background and anomalies with the assumption that the background pixels are low rank because they are widely dispersed and homogeneous, while the anomalies are few and sparse. The LRR theory determines the lowest rank representation of the background based on the background dictionary, and constrains the anomaly with sparsity [41]. Xu et al. [42] combined low-rank and sparse representation (LRASR) and constructed a stable and discriminative dictionary. Zhang et al. [43] implemented low-rank and sparse decomposition (LRaSMD) through the Go Decomposition (GoDec) [48] approach and adopted the RX detector to detect anomalies. Cheng et al, [44] proposed a graph and total variation regularized LRR-based model (GTVLRR) to retain the local geometrical and spatial structure. LRR-based methods are still widely used due to their high detection performance [11,45,49].

With the development of artificial intelligence, researchers have introduced deep learning into hyperspectral anomaly detection [50]. Due to the small proportion of anomalies in hyperspectral data that limits their contribution to model training, and the absence of prior spectral knowledge, most current deep learning-based hyperspectral anomaly detection methods chose unsupervised learning [49,51–57].

The deep neural networks are introduced to learn the intrinsic expression in the hyperspectral data to take advantage of their capacity on complex data modeling and feature extraction. The traditional method is then performed on the extracted features to detect anomalies. Lei et al. [51] utilized a deep brief network (DBN) [58] to automatically learn discriminative features and high-level representations in the spectral domain and fed them into an RX detector. Based on the generative models of autoencoder (AE) [59] and Generative Adversarial Networks (GAN) [60], the original hyperspectral data are reconstructed and the residuals of the original and the reconstructed data are used for anomaly detection. The AE-based methods have received a lot of attention recently. In [52], manifold learning is introduced to an AE network for hyperspectral anomaly detection. Jiang et al. [49] proposed a low-rank embedded network (LREN) to learn spectral features via an AE, and the LRR-based anomaly detector is used in the deep feature space. Xiang et al. [61] proposed a guided AE (GAE) which introduced a guided image to suppress the feature learning of anomalies and used the reconstruction error as the metric. GAN-based methods introduce adversarial learning [53] into background reconstruction. Jiang et al. [54] proposed GAN for hyperspectral anomaly detection (HADGAN) and reconstructed the hyperspectral data via an adversarial autoencoder (AAE) [62]. The morphological filter and RX were implemented on the residual image. Xie et al. [55] also used the AAE for reconstruction, but chose to use the morphological method on the hidden layer features.

Since the above methods usually detect anomalies in two stages, and deep features are not extracted specially for the hyperspectral anomaly detection tasks, the separate training of the feature extractor and the anomaly detector hinders deep neural networks from attaining their full potential. Therefore, an autonomous hyperspectral anomaly detection network (Auto-AD) is proposed in [56] to avoid the pre-processing and post-processing in other two-stage networks. Xie et al. [57] proposed an end-to-end Spectral Distribution-Aware Estimation network (SDEN) to directly estimate anomalies, which jointly learned the two parts of the feature extraction network and the anomaly detector. Meanwhile, in addition to the above unsupervised methods, weakly supervised methods are gradually becoming an attractive option. Most of the current weakly supervised methods firstly filtered out background pseudo-samples for training based on the estimated background distribution [10,11,63,64].

The traditional hyperspectral anomaly detection methods have a reasonable mathematical foundation and interpretability, but the shallow learning method somehow limits their detection performance. While the deep learning-based methods have a high rate in feature utilization and strong generalization ability, their interpretability is weaker than that of traditional algorithms. Therefore, a new hyperspectral anomaly detection method based on Collaborative Representation Neural Networks (CRNN) is proposed in this paper.

Our motivation is to implement traditional representation theory of hyperspectral anomaly detection into the framework of deep neural networks.

To be specific, the fairly complete background dictionaries of global and local streams are constructed, and the representation weight matrices are learned by optimizing the objective function of CR in the two-stream networks. The global background dictionary is a generalization of the global background, which is constructed based on all pixels of the entire image. To avoid the pollution of the anomalous pixels, the global CR networks is trained to obtain the most representative dictionary atoms under the constraint of the dictionary atom loss function. The local background dictionary focuses on the local information surrounding the test pixel, which is obtained by a sliding dual window. The global and local representation images of the test pixel are calculated through the linear combination of atoms in the two dictionaries, respectively. In traditional CR-based methods, the optimal solution of weight coefficients usually involves matrix inversion and other complex operations, which would bring large computational load and time consumption. While, in our proposed method, the two-stream networks learn the weight matrix via iterative optimization during training and the fixed parameters in inference, which leads to a high calculation efficiency. Finally, the anomaly score of each pixel can be measured by the residual input data and its representation through networks based on the jointly considered global and local background dictionary.

In addition, CRNN firstly introduces an AE to reconstruct the original input hyperspectral data to obtain the dimensionality reduction and dense expression of the original image. The global and local CR is then operated on the hidden feature map of the AE. Moreover, we design a feature extraction network to further learn the comprehensive spectral–spatial features in the hidden feature map.

The main contributions of this paper are as follows:

1. We propose a new collaborative representation neural network for hyperspectral anomaly detection that combines the mathematical foundation of CR theory with the effective feature learning capacity of deep neural networks. CRNN optimizes the objective function of CR through iterative training, and improves the detection performance with faster inference speed.
2. We perform both global and local streams of CR, and fuse the detection maps to obtain the final result comprehensively. The global dictionary is collected in the networks by optimizing the dictionary atom loss to refine the background without the pollution of anomalies. While the local dictionary by a sliding dual window reflects the background neighboring the test pixel.
3. We utilize a feature extraction network to generate the comprehensive feature map, including spectral–spatial features from both global and local perspectives. The deep latent feature space helps to learn the representation weigh matrices more effectively.

The remainder of our article is organized as follows. Section 2 reviews some related works. Section 3 introduces the proposed method. Section 4 provides the experiments in detail. Section 5 makes the discussion. Finally, the conclusion is presented in Section 6.

2. Related Works

We briefly review major works on CR-based detector [34] and AE [59] in this section.

2.1. Collaborative Representation-Based Detector

For a given hyperspectral dataset, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the i-th pixel in the data, the CR theory believes that the approximate representation $\hat{\boldsymbol{y}}$ can be calculated via the linear combination of background pixels in the surrounding area of the test pixel $\boldsymbol{y}$,

$$\hat{\boldsymbol{y}} = \boldsymbol{X}_s \alpha \tag{1}$$

where $\hat{\boldsymbol{y}}$ is the collaborative representation of the test pixel. $\boldsymbol{X}_s$ is the collection of background pixels around the test pixel through a dual window. $\alpha \in \mathbb{R}^{s \times 1}$ is the weight vector.

The Euclidean distance r between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ is usually introduced to measure the probability of anomaly

$$r = \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2 = \|\boldsymbol{y} - \boldsymbol{X}_s\alpha\|_2 \tag{2}$$

where $\|\cdot\|_2$ is the l_2-norm.

Collaboration means that the constraint on the weight vector α is that $\|\boldsymbol{y} - \boldsymbol{X}_s\alpha\|_2^2$ is the smallest, while $\|\alpha\|_2^2$ is the smallest, so the objective function is

$$\min_{\alpha} \|\boldsymbol{y} - \boldsymbol{X}_s\alpha\|_2^2 + \lambda\|\alpha\|_2^2 \tag{3}$$

where λ is the Lagrange multiplier. l_2-norm provides each element in α a tiny value and guarantees that all atoms take part in the representation.

The formula (3) is equal to:

$$\min_{\alpha} \left[\alpha^T \left(\boldsymbol{X}_s^T\boldsymbol{X}_s + \lambda I\right)\alpha - 2\alpha^T\boldsymbol{X}_s^T\boldsymbol{y}\right] \tag{4}$$

The above formula (4) belongs to the convex function optimization problem and exists as a closed-form solution for α. Setting the derivative to 0 gains the optimal solution for α,

$$\alpha = \left(\boldsymbol{X}_s^T\boldsymbol{X}_s + \lambda I\right)^{-1}\boldsymbol{X}_s^T\boldsymbol{y} \tag{5}$$

2.2. Autoencoder

AE is an unsupervised deep learning-based neural network model and consists of the encoder and the decoder. The encoder firstly maps the input vector $\boldsymbol{x}$ from input layer to hidden layer, while the decoder is a structure where the vector $\boldsymbol{z}$ in hidden layer is mapped to the output vector $\hat{\boldsymbol{x}}$,

$$\begin{aligned} \boldsymbol{z} &= f(\boldsymbol{x};\theta) \\ \hat{\boldsymbol{x}} &= \hat{f}(\boldsymbol{z};\hat{\theta}) \end{aligned} \tag{6}$$

where $f(\cdot;\theta)$ and $\hat{f}(\cdot;\hat{\theta})$ are the mapping functions of the encoder and the decoder.

In order to accurately reconstruct the input vector $\boldsymbol{x}$, the divergence between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ is as small as possible during training, so that the hidden $\boldsymbol{z}$ can well reflect the feature of $\boldsymbol{x}$. The loss function L_R is as follows:

$$L_R = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2 \tag{7}$$

3. Proposed Method

In this section, a hyperspectral anomaly detection method named CRNN is proposed. The flow chart is presented in Figure 1. Firstly, an AE reconstructs the original input hyperspectral data $\boldsymbol{X}$. The hidden feature $\boldsymbol{Z}$ during the reconstruction process serves as the basis for subsequent anomaly detection. Secondly, the comprehensive feature extraction network further learns the global and local spectral–spatial feature in $\boldsymbol{Z}$ to obtain comprehensive feature map $\boldsymbol{Z}^c$. The CR learning network includes global stream and local stream. The global background dictionary D^g and the local background dictionary D^l are both obtained from $\boldsymbol{Z}$, while the global representation weight α^g and local representation weight α^l are learned based on $\boldsymbol{Z}^c$. Finally, the anomaly score map $\boldsymbol{A}$ is calculated in the detection fusion module.

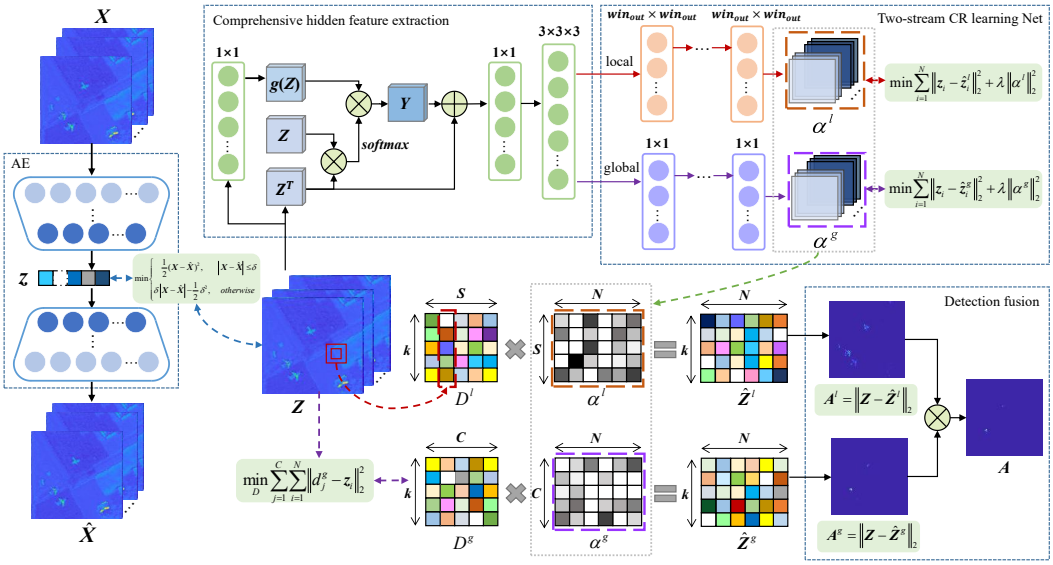


Figure 1. An overview of the proposed CRNN. It consists of four parts: AE, comprehensive feature extraction, two-stream CR learning network, and detection fusion.

3.1. Hidden Feature Map Generation

High spectral resolution and continuous spectral bands in hyperspectral data produce the issue of highly correlated and redundant bands. Therefore, the original hyperspectral data are densely expressed before anomaly detection to improve computational efficiency, specifically through the encoding process in an AE [59].

Supposing the original hyperspectral data as $\boldsymbol{X} \in \mathbb{R}^{H\times W\times K}$, where K is the number of bands of the original data, and N is the total amount of pixels in the image that $N = H \times W$ (H and W are the height and width, respectively). As shown in Figure 1, the reshaped $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{K\times N}$ is firstly input into an AE, and the output reconstructed image is $\hat{\boldsymbol{X}} = [\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \cdots, \hat{\boldsymbol{x}}_N] \in \mathbb{R}^{K\times N}$. In the encoder E, the original hyperspectral data $\boldsymbol{X}$ are firstly encoded into a lower-dimensional and denser hidden variable $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N] \in \mathbb{R}^{k\times N}$, where k is the number of bands of the hidden feature map. This hidden variable $\boldsymbol{Z}$ is mapped back to the original hyperspectral space in the decoder De for the reconstruction $\hat{\boldsymbol{X}}$. Huber loss L_δ [65] is chosen as the loss function,

$$L_\delta(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \begin{cases} \frac{1}{2}(\boldsymbol{x} - \hat{\boldsymbol{x}})^2 & \text{for } |\boldsymbol{x} - \hat{\boldsymbol{x}}| \le \delta \\ \delta|\boldsymbol{x} - \hat{\boldsymbol{x}}| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases} \tag{8}$$

where δ is the hyperparameters.

The structures of E and De are similar as both are composed of l_a numbers of 1×1 two-dimensional convolutional layers. The hidden feature map $\boldsymbol{Z}$ is obtained from the output of the l_a-th layer in E,

$$\begin{aligned} \boldsymbol{X}_{i+1} &= \sigma\left(GN\left(f_i^E\left(\boldsymbol{X}_i; \Theta_i^E\right)\right)\right), \quad i = 1, \ldots, l_a - 1 \\ \boldsymbol{Z} &= f_{l_a}^E\left(\boldsymbol{X}_{l_a}; \Theta_{l_a}^E\right) \end{aligned} \tag{9}$$

where $\boldsymbol{X}_i$ is the input datum of the i-th layer, $f_i^E\left(\cdot; \Theta_i^E\right)$ denotes the two-dimensional convolution with parameters Θ_i^E of the i-th layer in encoder E, $GN(\cdot)$ is the group normalization operation [66], and σ is a leakly rectified linear unit (Leakly ReLU) activation function [67].

3.2. Comprehensive Feature Extraction

Since the AE only considers the spectral dimension of each pixel during the reconstruction without the global and local structural information, it is necessary to further extract the features of $\boldsymbol{Z}$ to mine its deep potential features.

3.2.1. Non-Local Module

A non-local network [68] is introduced to learn global correlation features. $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N] \in \mathbb{R}^{k \times N}$ is the input of the module, and $\boldsymbol{Y}^g = \left[\boldsymbol{y}_1^g, \boldsymbol{y}_2^g, \cdots, \boldsymbol{y}_N^g\right]^T \in \mathbb{R}^{N \times k}$ is the global feature calculated by the module,

$$\boldsymbol{y}_i^g = \frac{1}{\mathcal{C}_i} \sum_{\forall j} f(\boldsymbol{z}_i, \boldsymbol{z}_j) g(\boldsymbol{z}_j) \tag{10}$$

where i and j are the indices of the pixel position; the function $f(\cdot)$ calculates the similarity between the two pixels, and Gaussian function is chosen in our model, $f(\boldsymbol{z}_i, \boldsymbol{z}_j) = e^{\boldsymbol{z}_i^T \boldsymbol{z}_j}$; the function $g(\cdot)$ denotes the feature representation, which is implemented by a 1×1 convolutional layer; $\mathcal{C}_i$ is the normalization factor, $C_i = \sum_{\forall j} f(\boldsymbol{z}_i, \boldsymbol{z}_j)$.

Therefore, the non-local module is equivalent to the self-attention mechanism [69], normalized in the form of softmax [70]. The global correlation feature $\boldsymbol{Y}^g$ can be expressed as:

$$\boldsymbol{Y}^g = \text{softmax}\left(\boldsymbol{Z}^T \boldsymbol{Z}\right) g(\boldsymbol{Z}) \tag{11}$$

Finally, the output feature map $\boldsymbol{Z}^{nl} \in \mathbb{R}^{N \times k}$ of the non-local module with the residual connection is

$$\boldsymbol{Z}^{nl} = \boldsymbol{W}\boldsymbol{Y}^g + \boldsymbol{Z}^T \tag{12}$$

where $\boldsymbol{W}$ denotes the weight matrix learned by the network.

3.2.2. Local Feature Learning Module

The local feature learning module consists of two convolutional layers, a two-dimensional one and a three-dimensional one, as shown in Figure 1. In the 1×1 two-dimensional convolution layer, the spectral features for each pixel are learned as $\boldsymbol{V} \in \mathbb{R}^{H \times W \times k}$. In the $3 \times 3 \times 3$ three-dimensional convolutional layer, the local spectral–spatial features are extracted on the expanded data $\boldsymbol{V}^e \in \mathbb{R}^{H \times W \times 1 \times k}$ with an additional dimension. The three-dimensional convolution not only fuses the spatial neighboring pixels to learn the local spatial features, like the 3×3 two-dimensional convolution, but also extracts the local spectral features via the convolution on the adjacent spectral bands, since some anomalies typically occur in the local spectral bands.

For the reshaped input feature $\boldsymbol{Z}^{nl} \in \mathbb{R}^{H \times W \times k}$, the comprehensive feature map $\boldsymbol{Z}^c \in \mathbb{R}^{H \times W \times nk}$ is obtained as followed

$$\begin{aligned} \boldsymbol{V} &= \sigma\left(GN\left(f_{\text{conv2}}\left(\boldsymbol{Z}^{nl}; \Theta_{\text{conv2}}\right)\right)\right) \\ \boldsymbol{Z}^c &= \sigma(GN(f_{\text{conv3}}\left(\boldsymbol{V}^e; \Theta_{\text{conv3}}\right))) \end{aligned} \tag{13}$$

where $f_{\text{conv2}}(\cdot; \Theta_{\text{conv2}})$ denotes the two-dimensional convolution and parameters Θ_{conv2} and $f_{\text{conv3}}(\cdot; \Theta_{\text{conv3}})$ denote the three-dimensional convolution with parameters Θ_{conv3}.

3.3. Two-stream Collaborative Representation Learning Networks

Based on the hidden layer features $\boldsymbol{Z} \in \mathbb{R}^{k \times N}$, we design a two-stream CR learning network for global and local streams.

3.3.1. Global Collaborative Representation Learning

The calculation of the global collaborative representation image $\hat{\boldsymbol{Z}}^g \in \mathbb{R}^{k \times N}$ is

$$\hat{\boldsymbol{Z}}^g = D^g \alpha^g \tag{14}$$

where $D^g \in \mathbb{R}^{k \times C}$ is the global background dictionary; $\alpha^g \in \mathbb{R}^{C \times N}$ is the global representation weight matrix; and C is the amount of dictionary atoms.

The global representation weight matrix α^g is obtained through the global CR learning networks based on the comprehensive feature map $\boldsymbol{Z}^c$, by l_b layers of 1×1 convolutional layer,

$$\begin{aligned} \boldsymbol{Z}_{i+1}^g &= \sigma\left(GN\left(f_i^{GC}\left(\boldsymbol{Z}_i^g; \Theta_i^{GC}\right)\right)\right), \quad i = 1, \ldots, l_b - 1 \\ \alpha^g &= f_{l_b}^{GC}\left(\boldsymbol{Z}_{l_b}^g; \Theta_{l_b}^{GC}\right) \end{aligned} \tag{15}$$

where $\boldsymbol{Z}_i^g$ is the input datum of the i-th layer in the global CR learning networks, and $\boldsymbol{Z}_1^g = \boldsymbol{Z}^c$.

In order to make the error between the global representation $\hat{\boldsymbol{Z}}^g$ of the network and the hidden feature map $\boldsymbol{Z}$ as small as possible, the loss function L_{Global} of the global CR learning network is designed according to the CR objective function in formula (3)

$$L_{Global} = \sum_{i=1}^{N} \left\| \boldsymbol{z}_i - \hat{\boldsymbol{z}}_i^g \right\|_2^2 + \lambda \|\alpha^g\|_2^2 \tag{16}$$

where $\boldsymbol{z}_i$ is the i-th pixel in $\boldsymbol{Z}$, and $\hat{\boldsymbol{z}}_i^g$ is the i-th pixel in $\hat{\boldsymbol{Z}}^g$.

Moreover, we design a loss function L_{Dic} to constrain the dictionary atoms and make them more representative in D^g

$$L_{Dic} = \sum_{j=1}^{C} \sum_{i=1}^{N} \left\| d_j^g - \boldsymbol{z}_i \right\|_2^2 \tag{17}$$

where d_j^g is the j-th atom in D^g. The purpose of this constraint is to minimize the distance from each dictionary atom to the hidden feature $\boldsymbol{Z}$.

3.3.2. Local Collaborative Representation Learning

Different from the global CR learning networks, the local background dictionary is obtained via a sliding dual window on the hidden feature $\boldsymbol{Z}$, with the sizes of inner and outer windows as win_{in} and win_{out}, respectively. For the i-th pixel in $\boldsymbol{Z}$, the local collaborative representation $\hat{\boldsymbol{Z}}^l = \left\{ \hat{\boldsymbol{z}}_1^l, \hat{\boldsymbol{z}}_2^l, \ldots, \hat{\boldsymbol{z}}_N^l \right\} \in \mathbb{R}^{k \times N}$ is calculated as

$$\hat{\boldsymbol{z}}_i^l = D_i^l \alpha_i^l \quad i = 1, 2, 3 \cdots N \tag{18}$$

where $\hat{\boldsymbol{z}}_i^l$ is the i-th pixel of $\hat{\boldsymbol{Z}}^l$; $D_i^l \in \mathbb{R}^{k \times S}$ denotes the local background dictionary of the i-th pixel, which is a collection of the feature vectors between the inner and outer windows centered on the i-th pixel in $\boldsymbol{Z}$; $\alpha_i^l \in \mathbb{R}^{S \times 1}$ is the weight coefficient of the i-th pixel in matrix $\alpha^l \in \mathbb{R}^{S \times N}$; and S represents the number of background pixels between the inner and outer windows that $S = win_{\text{out}} \times win_{\text{out}} - win_{\text{in}} \times win_{\text{in}}$.

Since the weight matrix α^l depends on the background pixels in the dual window, the convolutional kernel size in local CR learning networks is set to $win_{\text{out}} \times win_{\text{out}}$. The number of layers is still l_b the same as the global CR learning networks, and the calculation of α_l is similar with α_g in Equation (15).

The loss function of local collaborative representation $\hat{\boldsymbol{Z}}^l$ is also designed according to the objective function (3):

$$L_{Local} = \sum_{i=1}^{N} \left\| \boldsymbol{z}_i - \hat{\boldsymbol{z}}_i^l \right\|_2^2 + \lambda \left\| \alpha^l \right\|_2^2 \tag{19}$$

Therefore, combining the two streams of global and local, the objective function of CR learning networks is obtained as follows

$$L_{CR} = L_{Dic} + \lambda_1 L_{Global} + \lambda_2 L_{Local} \tag{20}$$

where λ_1 and λ_2 are the factors of the trade-off between the global and local streams.

Due to the introduction of learning layers in the neural network, the convex optimization problem in Formula (3) is transformed into a non-convex loss function in (20). The deep learning-based algorithm seeks multiple possible local optimal solutions during network iteration, instead of being restricted to a unique closed-form solution. In this way, the generalization performance of the detector would be enhanced and noise interference would be more difficult.

3.4. Detection Fusion

The Euclidean distance between the hidden feature $\boldsymbol{Z}$ and the global representation $\hat{\boldsymbol{Z}}^g$ or the local representation $\hat{\boldsymbol{Z}}^l$ is calculated, respectively, to obtain the global detection result $\boldsymbol{A}^g \in \mathbb{R}^{H\times W}$ and the local detection result $\boldsymbol{A}^l \in \mathbb{R}^{H\times W}$

$$A_i^g = \left\| \boldsymbol{z}_i - \hat{\boldsymbol{z}}_i^g \right\|_2 \quad i = 1, 2, 3 \cdots N \tag{21}$$

$$A_i^l = \left\| \boldsymbol{z}_i - \hat{\boldsymbol{z}}_i^l \right\|_2 \quad i = 1, 2, 3 \cdots N \tag{22}$$

where A_i^g is the i-th pixel in $\boldsymbol{A}^g$, and A_i^l is the i-th pixel in $\boldsymbol{A}^l$.

The final anomaly score map $\boldsymbol{A} \in \mathbb{R}^{H\times W}$ will be obtained by element-wise multiplying of $\boldsymbol{A}^g$ and $\boldsymbol{A}^l$ to achieve a higher detection accuracy:

$$\boldsymbol{A} = \boldsymbol{A}^g \odot \boldsymbol{A}^l \tag{23}$$

The proposed CRNN method is described in Algorithm 1. CFL-Net stands for the comprehensive feature extraction networks and CRL-Net stands for the two-stream CR learning networks. AE_pre_num is epochs that the AE is trained individually during the pre-training phase to strengthen the stability of network training. AE_pre_num = 10.

Algorithm 1: Algorithm flow diagram of CRNN

Input: original Hyperspectral data $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{H\times W}] \in \mathbb{R}^{H\times W\times K}$.
Output: Anomaly score map $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_{H\times W}] \in \mathbb{R}^{H\times W}$.

```
Initialize the network with random weight;
foreach epoch do
    AE update: E, De by L_δ
    if epoch > AE_pre_num then
        CRNN update: CFL-Net, CRL-Net by L_CR
    end
    Back-propagate L_δ, L_CR
    To change E, De, CFL-Net, CRL-Net.
end
```

4. Experiments

To validate the detection performance of our proposed CRNN, we perform experiments on six real hyperspectral images and compare with eight previous algorithms.

4.1. Experimental Settings

4.1.1. Datasets Description

Six datasets are employed in the experiment, including Beach [25], Urban [25], XiongAn [71], HYDICE [72], Gulfport [25], and SanDiego [42]. Table 1 lists the details of each dataset. The targets in the images of Beach, Urban, XiongAn, and HYDICE are small. While the anomalies in the datasets of Gulfport and SanDiego are relatively large and have specific shape features, which are planar targets.

Table 1. Details of the six hyperspectral datasets in the experiment.

Dataset	Sensor	Resolution	Spatial Size	Bands	Anomaly Type	Anomaly Proportion	Anomaly Size
Beach	AVIRIS [1]	4.4 m	100 × 100	188	boat	0.61%	11
Urban	AVIRIS [1]	17.2 m	100 × 100	204	buildings	0.67%	2–14
HYDICE	HYDICE [2]	1 m	80 × 100	162	vehicles	0.24%	1–4
XiongAn	Gaofeng [3]	0.5 m	100 × 100	250	vehicle	0.10%	10
Gulfport	AVIRIS [1]	3.4 m	100 × 100	188	3 airplanes	0.60%	10–40
SanDiego	AVIRIS [1]	3.5 m	100 × 100	188	3 airplanes	0.34%	30–50

[1] Airborne Visible Infrared Imaging Spectrometer. [2] Hyperspectral Digital Imagery Collection Experiment. [3] Gaofeng Full Spectrum Multi-modal Imaging Spectrometer.

4.1.2. Evaluation Criteria

To quantitatively evaluate the detection performance, the three-dimensional receiver operating characteristic (3D ROC) curves and the area under the ROC curve (AUC) [73] are introduced. The 3D ROC curves depict the trade-off among the three parameters of the true positive rate P_d, the false positive rate P_f, and the detection threshold τ. Three types of two-dimensional ROC (2D ROC) curves decomposed from the 3D ROC curve are also applied in the experiment, including (P_d, P_f), (P_d, τ), and (P_f, τ). Correspondingly, three types of AUC values are applied, and their values range in [0, 1]. The $\mathrm{AUC}_{(\mathrm{D,F})}$ (AUC of curve (P_d, P_f)) value approaching to 1 indicates a perfect effectiveness, the $\mathrm{AUC}_{(\mathrm{D},\tau)}$ (AUC of curve (P_d, τ)) value closer to 1 evaluates the better detection probability, and the $\mathrm{AUC}_{(\mathrm{F},\tau)}$ (AUC of curve (P_f, τ)) value approaching 0 indicates an outstanding performance on background suppression. In addition, another five AUC values [74] are introduced to further evaluate the detector of joint anomaly detection (JAD), joint background suppression (JBS), anomaly detection with background suppression (ADBS), overall anomaly detection probability (OADP), and signal-to-noise probability ratio (SNPR). They are denoted as $\mathrm{AUC}_{\mathrm{JAD}}$, $\mathrm{AUC}_{\mathrm{JBS}}$, $\mathrm{AUC}_{\mathrm{ADBS}}$, $\mathrm{AUC}_{\mathrm{OADP}}$, and $\mathrm{AUC}_{\mathrm{SNPR}}$, respectively. The five AUC values are calculated in the following equation, and larger AUC values indicate a better performance. The values of $\mathrm{AUC}_{\mathrm{JAD}}$, $\mathrm{AUC}_{\mathrm{JBS}}$, and $\mathrm{AUC}_{\mathrm{ADBS}}$ range in [0, 2], while $\mathrm{AUC}_{\mathrm{OADP}}$ value is between [0, 3].

$$\begin{aligned}
\mathrm{AUC}_{\mathrm{JAD}} &= \mathrm{AUC}_{(\mathrm{D,F})} + \mathrm{AUC}_{(\mathrm{D},\tau)} \\
\mathrm{AUC}_{\mathrm{JBS}} &= \mathrm{AUC}_{(\mathrm{D,F})} + \left(1 - \mathrm{AUC}_{(\mathrm{F},\tau)}\right) \\
\mathrm{AUC}_{\mathrm{ADBS}} &= \mathrm{AUC}_{(\mathrm{D},\tau)} + \left(1 - \mathrm{AUC}_{(\mathrm{F},\tau)}\right) \\
\mathrm{AUC}_{\mathrm{OADP}} &= \mathrm{AUC}_{(\mathrm{D,F})} + \mathrm{AUC}_{(\mathrm{D},\tau)} + \left(1 - \mathrm{AUC}_{(\mathrm{F},\tau)}\right) \\
\mathrm{AUC}_{\mathrm{SNPR}} &= \frac{\mathrm{AUC}_{(\mathrm{D},\tau)}}{\mathrm{AUC}_{(\mathrm{F},\tau)}}
\end{aligned} \tag{24}$$

4.1.3. Compared Method

The comparison methods include traditional anomaly detection methods of the benchmark algorithm such as global RX (GRX) [13], and four representation-based methods, such as the original CRD [34], CRD in EAS (EAS_CR) [18], relaxed CRD (RCRD) [40], and the local summation unsupervised nearest regularized subspace anomaly detector (LSUNRS) [38]. We also experimented with recent deep learning-based anomaly detection methods, such as LREN [49], HADGAN [54], and GAE [61]. The comparison algorithms used in this paper are all verified based on the source code released by the original author. In order to assure the equitable results for the comparison experiment, the parameters of the CRD,EAS_CR, RCRD, LSUNRS, LREN, HADGAN, and GAE in the experiment were all optimal.

4.1.4. Implementation Details

Our approach is implemented in Pytorch 1.8, and all the experiments were tested on a workstation with an Nvidia GeForce GTX 1080-Ti GPU and an Intel i9 CPU. According to the network structure described in Section 3, the specific experiment settings are as follows. Both the encoder and the decoder use the structure of a 4-layer convolutional network, i.e., $l_a = 4$. The channel dimension of each layer is configured as E{K, 100, 50, 20, k} and De{k, 20, 50, 100, K}, where K is the number of input hyperspectral channels, and k is the feature dimension of $\boldsymbol{Z}$ as $k = 10$. The comprehensive feature $\boldsymbol{Z}^c \in \mathbb{R}^{H \times W \times nk}$ is obtained in the feature extraction networks, where $n = 5$. In CR learning networks, both the global and local streams adopt the 3-layer convolutional network, i.e., $l_b = 3$. The channel number configuration is GC{nk, k, k, C} and LC{nk, k, k, S}. δ in the Huber loss L_δ in Equation (8) is set as 1. λ_1 and λ_2 in the loss function L_{CR} in Equation (20) are set to 0.1 and 0.1. During training, the number of epochs is set as 500 to balance the detection accuracy and the calculation load, and the Adam algorithm is chosen to optimize the parameters of networks. The network learning rate is initially set to 10^{-4} and an exponential decay would be performed after 100 epochs. The above parameters are the default in the experiment and can be adjusted in other versions of this experiment.

4.2. Detection Performance

The anomaly detection maps of the proposed CRNN and the other eight algorithms of the six datasets are displayed in Figure 2. For a further and more intuitive analysis of the quantitative performance of each algorithm, the 3D ROC curves and the three types of 2D ROC curves, including (P_d, P_f), (P_d, τ), and (P_f, τ), are shown in Figure 3. The 2D ROC curves of (P_d, P_f) are set in a logarithmic scale for better visualization. The eight AUC performance comparisons for each dataset are shown in Table 2, and the average values of the AUC metrics over six datasets are listed in Table 3.

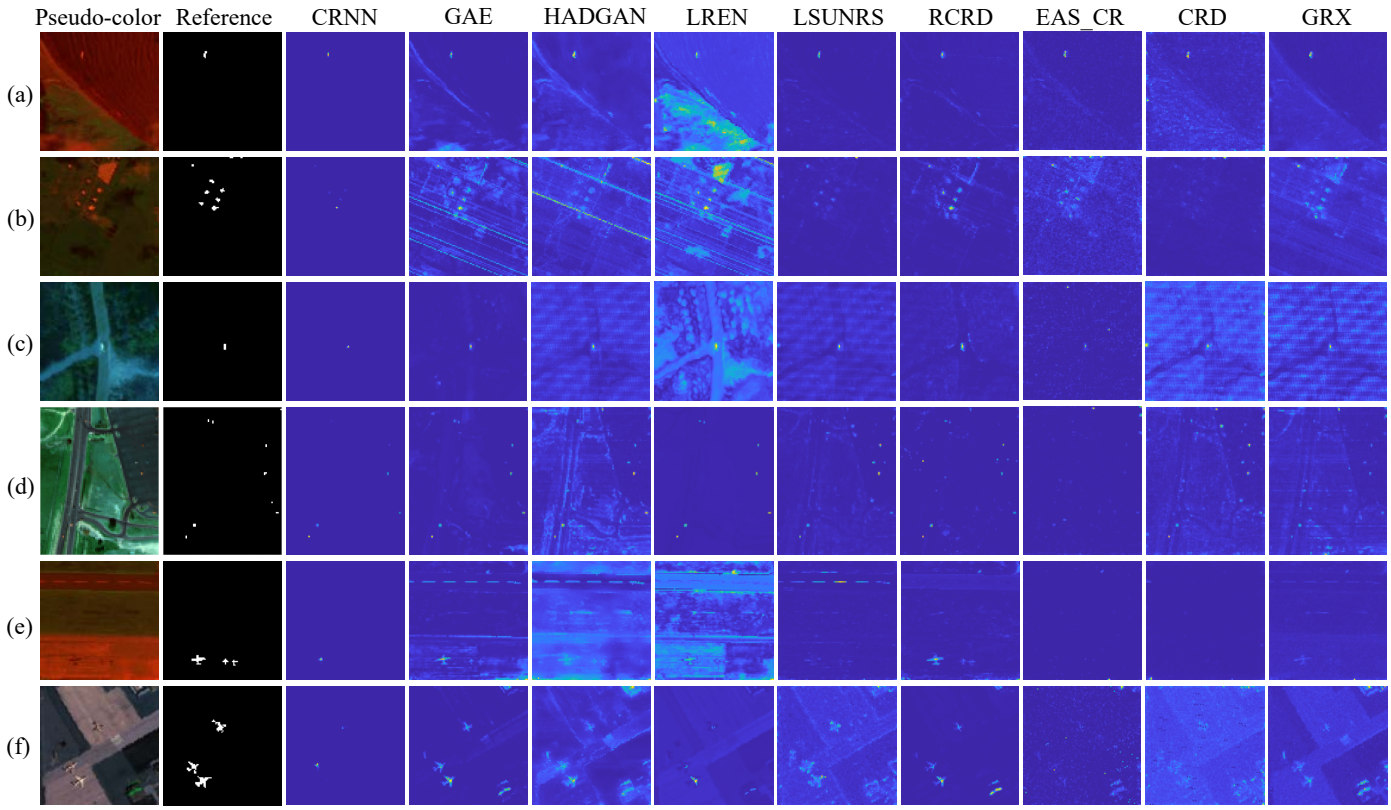


Figure 2. Pseudo-color images, references, and detection maps of the proposed CRNN and compared algorithms on six datasets. (**a**) Beach. (**b**) Urban. (**c**) XiongAn. (**d**) HYDICE. (**e**) Gulfport. (**f**) SanDiego.

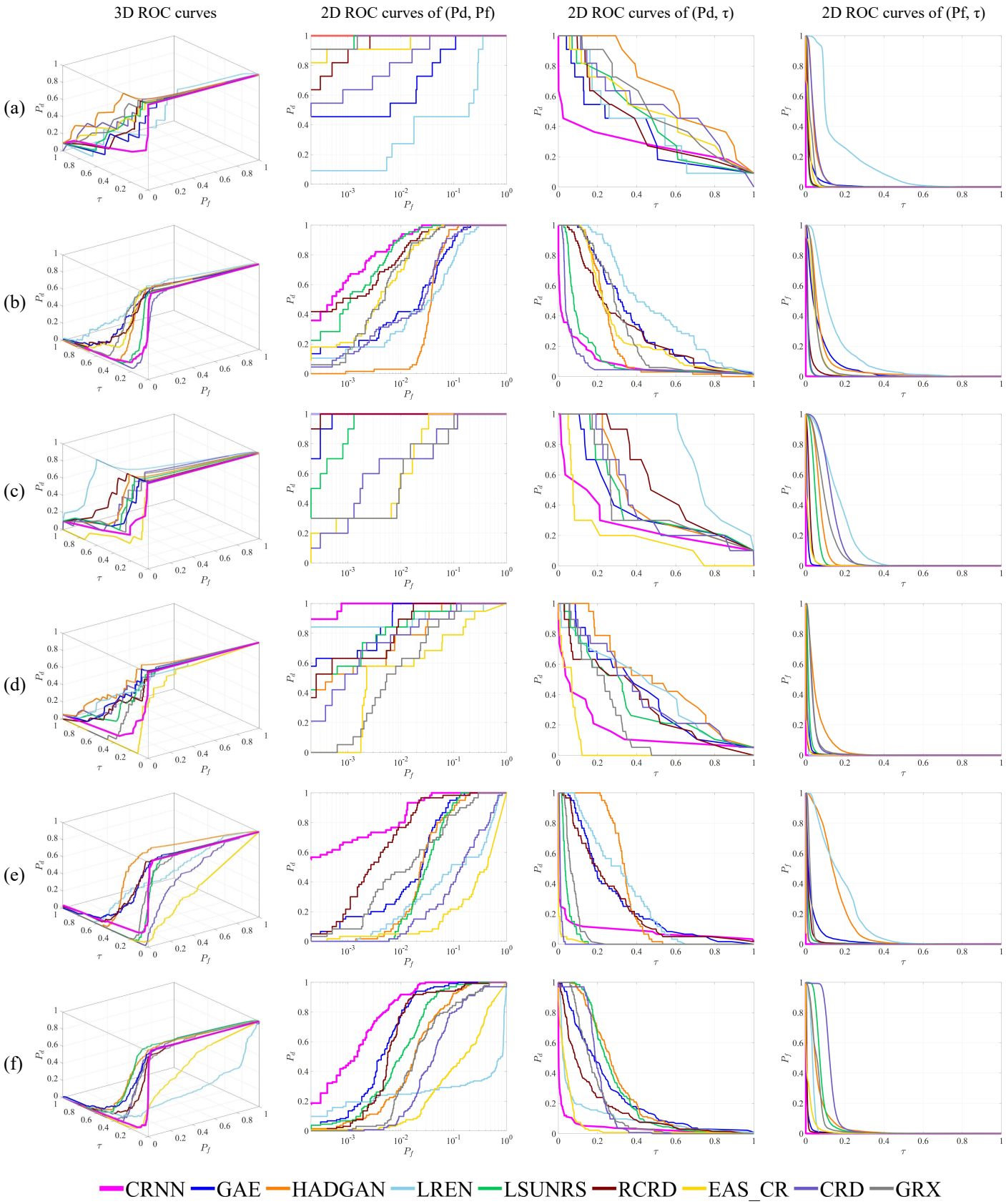


Figure 3. 3D ROC curves and the generated three 2D ROC curves of the proposed CRNN and compared algorithms on six datasets. (**a**) Beach. (**b**) Urban. (**c**) XiongAn. (**d**) HYDICE. (**e**) Gulfport. (**f**) SanDiego.

Table 2. AUC values of the proposed CRNN and compared algorithms on six datasets. (The best of each dataset is in bold).

Dataset	Method	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
Beach	CRNN	**1.0000**	0.2803	**7.75×10^{-9}**	1.2803	**2.0000**	1.2803	2.2803	**3.62×10^{7}**
	CRNN_t [1]	1.0000	0.7747	0.0014	1.7747	1.9986	1.7733	2.7733	553.6985
	GAE [61]	0.9773	0.3542	0.0176	1.3315	1.9596	1.3366	2.3139	20.0874
	HADGAN [54]	**1.0000**	**0.6598**	0.0448	**1.6597**	1.9552	**1.6150**	**2.6150**	14.7397
	LREN [49]	0.8451	0.4071	0.1662	1.2521	1.6788	1.2408	2.0859	2.4489
	LSUNRS [38]	0.9998	0.4414	0.0092	1.4413	1.9907	1.4323	2.4321	48.1575
	RCRD [40]	0.9996	0.4036	0.0114	1.4032	1.9882	1.3922	2.3918	35.3348
	EAS_CR [18]	0.9986	0.5020	0.0149	1.5005	1.9837	1.4871	2.4857	33.7063
	CRD [34]	0.9944	0.5351	0.0469	1.5295	1.9474	1.4882	2.4825	11.4035
	GRX [13]	0.9998	0.5314	0.0259	1.5313	1.9739	1.5055	2.5053	20.4912
Urban	CRNN	**0.9973**	0.0899	**2.45×10^{5}**	1.0872	**1.9973**	1.0899	2.0872	**3.66×10^{3}**
	CRNN_t [1]	0.9973	0.6896	0.0277	1.6869	1.9696	1.6619	2.6592	24.8633
	GAE [61]	0.9622	0.3695	0.0666	1.3317	1.8956	1.3029	2.2651	5.5496
	HADGAN [54]	0.9584	0.2483	0.0763	1.2067	1.8822	1.1720	2.1304	3.2555
	LREN [49]	0.9399	**0.4838**	0.1406	**1.4237**	1.7993	**1.3432**	2.2831	3.4411
	LSUNRS [38]	0.9963	0.1352	0.0105	1.1315	1.9858	1.1247	2.1210	12.8765
	RCRD [40]	0.9937	0.3162	0.0160	1.3099	1.9777	1.3003	**2.2939**	19.8251
	EAS_CR [18]	0.9911	0.3082	0.0496	1.2993	1.9414	1.2586	2.2496	6.2098
	CRD [34]	0.9655	0.0848	0.0127	1.0503	1.9528	1.0721	2.0376	6.6565
	GRX [13]	0.9907	0.3143	0.0556	1.3049	1.9351	1.2587	2.2494	5.6570
XiongAn	CRNN	**1.0000**	0.2718	**7.48×10^{-6}**	1.2718	**2.0000**	1.2718	2.2718	**3.63×10^{4}**
	CRNN_t [1]	1.0000	0.8708	0.2460	1.8708	1.7540	1.6247	2.6247	3.5392
	GAE [61]	0.9999	0.4014	0.0079	1.4013	1.9920	1.3935	2.3934	50.6621
	HADGAN [54]	**1.0000**	0.4634	0.0718	1.4634	1.9282	1.3917	2.3916	6.4562
	LREN [49]	**1.0000**	**0.7751**	0.1641	**1.7751**	1.8359	**1.6109**	**2.6109**	4.7220
	LSUNRS [38]	0.9996	0.4296	0.0499	1.4292	1.9497	1.3797	2.3793	8.6172
	RCRD [40]	0.9999	0.5710	0.0209	1.5709	1.9791	1.5501	2.5501	27.3467
	EAS_CR [18]	0.9881	0.1909	0.0105	1.1790	1.9776	1.1804	2.1685	18.1455
	CRD [34]	0.9783	0.4413	0.1315	1.4196	1.8469	1.3098	2.2881	3.3569
	GRX [13]	0.9765	0.4222	0.1000	1.3986	1.8765	1.3222	2.2987	4.2235
HYDICE	CRNN	**0.9999**	0.1570	**9.75×10^{-7}**	1.1570	**1.9999**	1.1570	2.1570	**1.61×10^{5}**
	CRNN_t [1]	0.9999	0.8132	0.0210	1.8131	1.9790	1.7923	2.7922	38.8089
	GAE [61]	0.9982	0.3973	0.0099	1.3955	1.9883	1.3874	2.3856	40.1662
	HADGAN [54]	0.9903	**0.5024**	0.0571	**1.4927**	1.9332	**1.4453**	**2.4356**	8.7944
	LREN [49]	0.9784	0.4271	0.0093	1.4055	1.9691	1.4179	2.3963	46.1746
	LSUNRS [38]	0.9929	0.3449	0.0166	1.3378	1.9763	1.3283	2.3212	20.7202
	RCRD [40]	0.9962	0.3380	0.0059	1.3342	1.9903	1.3322	2.3284	57.7771
	EAS_CR [18]	0.9233	0.0554	0.0037	0.9787	1.9196	1.0517	1.9751	15.0686
	CRD [34]	0.9868	0.4117	0.0351	1.3985	1.9517	1.3765	2.3634	11.7149
	GRX [13]	0.9763	0.2184	0.0380	1.1947	1.9382	1.1803	2.1566	5.7406
Gulfport	CRNN	**0.9955**	0.0843	**3.32×10^{9}**	1.0799	**1.9955**	1.0843	2.0799	**2.54×10^{7}**
	CRNN_t [1]	0.9955	0.4825	0.0105	1.4780	1.9851	1.4720	2.4676	46.1582
	GAE [61]	0.9690	0.2675	0.0314	1.2364	1.9376	1.2361	2.2051	8.5239
	HADGAN [54]	0.9602	**0.3380**	0.1496	**1.2982**	1.8106	1.1884	2.1486	2.2588
	LREN [49]	0.7521	0.3104	0.1739	1.0626	1.5782	1.1365	1.8887	1.7851
	LSUNRS [38]	0.9589	0.0455	0.0139	1.0044	1.9450	1.0316	1.9905	3.2769
	RCRD [40]	0.9899	0.2552	0.0160	1.2451	1.9739	**1.2392**	**2.2291**	15.9448
	EAS_CR [18]	0.5714	0.0059	0.0015	0.5773	1.5698	1.0044	1.5758	3.8782
	CRD [34]	0.7762	0.0112	0.0069	0.7874	1.7693	1.0043	1.7805	1.6246
	GRX [13]	0.9526	0.0736	0.0248	1.0262	1.9278	1.0489	2.0015	2.9743

Table 2. *Cont.*

Dataset	Method	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
SanDiego	CRNN	**0.9965**	0.0368	$\mathbf{1.18 \times 10^{-5}}$	1.0333	**1.9965**	1.0368	2.0333	$\mathbf{3.11 \times 10^{3}}$
	CRNN_t [1]	0.9965	0.5481	0.0178	1.5446	1.9787	1.5303	2.5268	30.7612
	GAE [61]	0.9910	0.2380	0.0079	1.2290	1.9831	1.2301	**2.2211**	30.1127
	HADGAN [54]	0.9644	**0.2898**	0.0653	1.2542	1.8991	**1.2244**	2.1888	4.4353
	LREN [49]	0.4387	0.1041	0.0346	0.5428	1.4041	1.0695	1.5082	3.0098
	LSUNRS [38]	0.9796	0.2756	0.0789	**1.2552**	1.9007	1.1967	2.1763	3.4914
	RCRD [40]	0.9825	0.1575	0.0084	1.1399	1.9740	1.1490	2.1315	18.6983
	EAS_CR [18]	0.7515	0.0464	0.0149	0.7979	1.7366	1.0316	1.7831	3.1202
	CRD [34]	0.9071	0.1993	0.1272	1.1064	1.7799	1.0721	1.9792	1.5671
	GRX [13]	0.9403	0.1778	0.0589	1.1181	1.8814	1.1189	2.0592	3.0176

[1] A logarithmic transformation was performed on the result maps of CRNN, and this method is only used as a supplementary illustration and does not participate in the comparison of other methods.

Table 3. Average AUC values of the proposed CRNN and compared algorithms over six datasets. (The best is in bold).

	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
CRNN	**0.9982**	0.1534	$\mathbf{7.47 \times 10^{-6}}$	1.1516	**1.9982**	1.1534	2.1516	$\mathbf{1.03 \times 10^{7}}$
CRNN_t [1]	0.9982	0.6965	0.0541	1.6947	1.9441	1.6424	2.6406	116.3049
GAE [61]	0.9829	0.3380	0.0236	1.3209	1.9594	1.3144	2.2974	25.8503
HADGAN [54]	0.9789	0.4169	0.0775	**1.3958**	1.9014	**1.3395**	2.3183	6.6567
LREN [49]	0.8257	**0.4179**	0.1148	1.2436	1.7109	1.3032	2.1289	10.2636
LSUNRS [38]	0.9879	0.2787	0.0298	1.2666	1.9580	1.2489	2.2367	16.1900
RCRD [40]	0.9936	0.3403	0.0131	1.3339	1.9805	1.3272	**2.3208**	29.1545
EAS_CR [18]	0.8706	0.1848	0.0159	1.0555	1.8548	1.1690	2.0396	13.3548
CRD [34]	0.9347	0.2806	0.0601	1.2153	1.8747	1.2205	2.1552	6.0539
GRX [13]	0.9727	0.2896	0.0505	1.2623	1.9222	1.2391	2.2118	7.0174

[1] A logarithmic transformation was performed on the result maps of CRNN, and this method is only used as a supplementary illustration and does not participate in the comparison of other methods.

4.2.1. Detection Performance for Small Targets

The detection maps of the Beach dataset are shown in Figure 2a; the target in the detection map of CRNN is prominent, and the suppression of the background is the best since CRNN can effectively suppress the false alarms in the background of this dataset. In the detection maps of other methods, the beach at the bottom left is a more obvious false alarm, especially in LREN. Figure 2b shows the detection maps of the Urban dataset and the result is that CRNN presents outstanding targets and better background suppression. The results of GAE, HADGAN, LREN, RCRD, EAS_CR, and GRX present obvious background texture and banding noise; in particular, the result of LREN presents obvious false detection. The background and targets are both at a low brightness level in the result maps of CRD and LSUNRS. However, the ROC curves (P_d, P_f) of CRNN and LSUNRS are quite close when P_d is reaching 1. It may be due to the disparity in the brightness of different targets in the detection result of CRNN, which means only some anomalous pixels are visually obvious. The detection maps of the XiongAn dataset are shown in Figure 2c; the background suppression is not so effective in the results of the eight compared methods, especially in LREN, and the detection maps of HADGAN, LSUNRS, CRD, and GRX present evident banding noise. Compared with CRD, the two CR-based methods of RCRD and EAS_CR improve the background suppression, but still present some false alarms in detection maps. While the result of CRNN not only presents a perfect suppression on the background but also has a strong anti-interference ability against noise. The detection maps of the HYDICE

image are shown in Figure 2d. The detection results of CRNN and LREN are best when dealing with prominent targets and good background suppression, while LREN has some instances of missed detection and the detection score of some anomalous pixels is low. The background suppression in other methods is poor; in addition, the background texture is clear in HADGAN, CRD, and GRX, while some points of false alarm present higher brightness in RCRD and EAS_CR.

For small anomalous targets, the proposed CRNN produces visual results closest to the reference compared with other seven state-of-the-art methods. In Figure 3, the ROC curves of (P_d, P_f) are all the first to reach 1, and the ROC curves of (P_f, τ) are all the closest to the lower left corner. It is clear from the study above that the proposed CRNN can clearly highlight the small anomalous targets in the detection maps while effectively suppressing the background, since it utilizes neural networks to learn features that are more conducive to anomaly detection.

4.2.2. Detection Performance for Planar Targets

As the detection maps of the Gulfport dataset show in Figure 2e, it can be seen that only the larger aircraft is significant in the detection results of CRNN, and the other two smaller ones are difficult to observe directly but still present higher values than the background. However, in the result maps of traditional detectors of GRX, CRD, EAS_CR, and LSUNRS, the saliency of targets is poor, and obvious false alarm points occur. However, in GAE, HADGAN, and LREN, the networks only learn the texture features of the hyperspectral image, causing a poor detection result. Targets in the detection maps of RCRD are obvious, and the ROC curves (P_d, P_f) of CRNN and RCRD are close, but CRNN is better at background suppression. The detection results of the SanDiego dataset are shown in Figure 2f. The detection map of CRNN presents clear targets and perfect background suppression, but the anomaly scores vary greatly between the three airplanes. In the detection results of GRX, CRD, EAS_CR, LSUNRS, and HADGAN, the targets are lost in the background and obvious noise, causing some false alarms. Although some pixels of the three anomalies have higher brightness in the result of GAE, LREN, and RCRD, the background suppression is not so good that the roof on the lower right turns to be a false alarm target.

For planar targets on the Gulfport and SanDiego datasets, the ROC curves of (P_d, P_f), and (P_f, τ) are all the closest to the upper left corner and the lower left corner, respectively. It can be concluded that CRNN also presents a perfect detection accuracy with the lowest false alarm rate for planar targets such as airplanes. Although the target texture is not obvious in the detection maps of CRNN, the outline of the aircraft can be clearly seen after performing the logarithmic transformation on the detection maps as show in Figure 4, which proves that the anomalies and the background are still separable. The logarithmic transformation is as follows

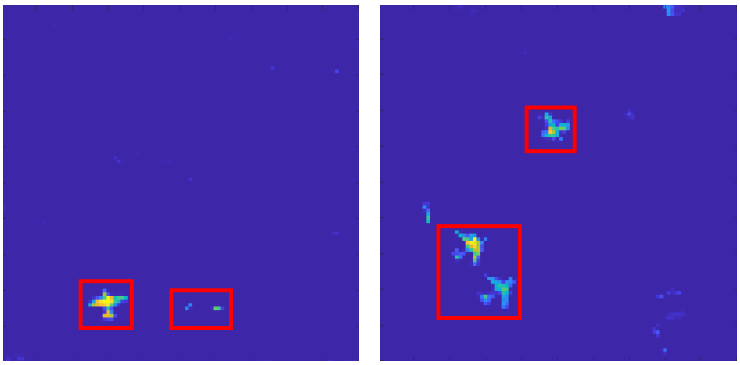
$$R = \log_{v+1}(1 + v \cdot I) \tag{25}$$

where $(v + 1)$ is the base of the logarithm. I and R denote the input and output images, respectively, and the values are all in the range of [0, 1].

4.2.3. AUC Performance

According to the comparison of the average AUC values in Table 3, CRNN has the highest $AUC_{(D,F)}$ value of 0.9982, which is 6.79% higher than that of the traditional CRD, indicating that CRNN has perfect detection efficiency and finds the balance between improving the detection rate and reducing the false alarm rate. At the same time, CRNN also shows the smallest $AUC_{(F,\tau)}$ score, and the largest AUC_{JBS} and AUC_{SNPR}, which are 7.47×10^{-6}, 1.9982, 1.03×10^{7}, respectively. It indicates that CRNN has a strong ability to suppress the background. However, the performance of CRNN in $AUC_{(D,\tau)}$, AUC_{JAD}, AUC_{ADBS}, and AUC_{OADP} needs to be improved, and the values of the latter three are mainly related to the former. This may be because the high efficiency of background suppression in CRNN also has an impact on some anomalous pixels, and the detection rate

needs to rely on a lower threshold, causing the poor performance on (P_d, τ) curves and $AUC_{(D,\tau)}$. However, it does not mean that the background and anomalies are inseparable. The average AUC_{SNPR} of the detection maps reaches seven orders of magnitude, and the anomaly scores of background pixels is concentrated around 10^{-11} to 10^{-7}. The order of magnitude of the potential anomalous pixels spans a large range that the anomaly scores of some pixels are quite close to 1, while other pixels may reach about 10^{-4}. As a supplementary illustration, since the orders of magnitude of the background and partial anomalies are small, we tried logarithmic transformation on the result maps of CRNN as the image enhancement strategy to re-evaluate the AUC performance. The logarithmic transformation is in shown in Equation (25) and v is set to 10^2 in all datasets. This version is not included in the comparison of other methods in Table 3 because it is only used as the post-processing method. According to the results of AUC, logarithmic transformation searches a balance between $AUC_{(D,\tau)}$ and $AUC_{(F,\tau)}$, and improves the target detectability.



(a) (b)

Figure 4. Detection maps after the logarithmic transformation. The red boxes are manually added to mark the targets. (**a**) Gulfport ($v = 10^7$). (**b**) SanDiego ($v = 10^4$).

In addition, we repeated 10 experiments for each dataset to measure the uncertainty of network training. The mean and variance of the three AUC values of $AUC_{(D,F)}$, $AUC_{(D,\tau)}$, and $AUC_{(F,\tau)}$ are shown in Table 4, which verifies the credibility of the experimental results for our proposed model.

Table 4. $AUC_{(D,F)}$ $AUC_{(D,\tau)}$ and $AUC_{(F,\tau)}$ values of CRNN in repeated experiments for each dataset.

	$AUC_{(D,F)}$		$AUC_{(D,\tau)}$		$AUC_{(F,\tau)}$	
	Mean	**Variance(%)**	**Mean**	**Variance(%)**	**Mean**	**Variance(%)**
Beach	0.99950	0.04849	0.22151	4.75515	4.25×10^{-5}	0.00432
Urban	0.99642	0.10841	0.07129	1.24275	1.51×10^{-5}	0.00110
XiongAn	0.99994	0.00611	0.2561	4.02692	1.90×10^{-5}	0.00242
HYDICE	0.99988	0.00953	0.18799	3.54861	4.80×10^{-6}	0.00070
Gulfport	0.98564	1.36994	0.08248	2.59059	2.52×10^{-4}	0.02073
SanDiego	0.98954	0.43000	0.02573	1.11803	2.70×10^{-4}	0.01517

4.3. Computing Time

In addition to the accuracy of anomaly detection, computing time is also an important part of algorithm efficiency. The computing time of the proposed CRNN and the compared eight algorithms is listed Table 5. For deep learning-based methods, the training epochs of CRNN, GAE, and HADGAN are all set to 500. We counted their training time on GPU and inference time tested on CPU. While LREN has no module for testing, so we only recorded the entire computing time including network training on GPU. The training epochs of LREN are 1000 according to [49]. For non-deep learning methods, LSUNRS, RCRD, EAS_CR, CRD, and RX, were implemented on MATLAB R2019b.

Table 5. Computing time in second of the proposed CRNN and compared algorithms. (The content in brackets in columns 2–4 represents the inference time of the network at test).

	CRNN	**GAE** [61]	**HADGAN** [54]	**LREN** [49]	**LSUNRS** [38]	**RCRD** [40]	**EAS_CR** [18]	**CRD** [34]	**GRX** [13]
Beach	49.8535 (0.0279)	47.7064 (6.4304)	674.0821 (1.8471)	100.2248	18.7829	38.2593	2.5918	1.5395	0.0854
Urban	60.6503 (0.0259)	48.6605 (6.4776)	677.5559 (2.0005)	109.0844	19.3547	40.2936	3.1943	1.6119	0.0896
XiongAn	21.0506 (0.0558)	56.7479 (7.9990)	673.0219 (2.6153)	102.5597	30.9906	51.7006	3.3039	8.8312	0.1826
HYDICE	19.2016 (0.0189)	26.0721 (5.2584)	693.5843 (1.6805)	73.0596	14.5599	25.5156	1.6684	1.2025	0.0669
Gulfport	70.5033 (0.0488)	47.8685 (6.4286)	676.432 (1.8710)	100.8034	18.8575	39.4005	2.6521	1.5377	0.0807
SanDiego	31.5995 (0.0229)	48.6626 (6.7521)	676.4849 (1.8861)	101.3657	19.0317	36.6709	2.1232	1.5573	0.0840

The comparison of computing times in Table 5 are just for reference due to the differences in the implementation of these methods, but it still shows the high efficiency of our proposed CRNN. Compared with the three deep learning-based method, the inference time of CRNN is the shortest, mainly because of the simple linear computation of matrices in a few network layers. GAE calculates the guided images through a spectral similarity method that takes most of the computation time. HADGAN directly used the traditional detector on the residual image and still retains the computational cost of traditional algorithms. LREN needs to consume part of computing time in solving the optimization of LRR.

5. Discussion

In this section, discussions of the ablation study and parameter analysis are made.

5.1. Ablation Study

To verify the effectiveness of the two-stream CR learning networks, including the fusion strategy and the joint training strategy in the proposed CRNN, we set a series of ablation experiments for research. In this experiment, the following scenarios were compared.

- CRNN: the proposed CRNN model, whose detection result is $\boldsymbol{A} = \boldsymbol{A}^g \odot \boldsymbol{A}^l$;
- CRNN+: the CRNN model in additive fusion version, whose detection result is $\boldsymbol{A}^+ = \boldsymbol{A}^g + \boldsymbol{A}^l$;
- LCRNN: the local stream of CR networks in CRNN, whose detection result is $\boldsymbol{A}^l$;
- GCRNN: the global stream of CR networks in CRNN, whose detection result is $\boldsymbol{A}^g$;
- CRNN_se: the local stream and global stream are separately trained and fused to obtain the detection result. It is a separate training version for CRNN;
- LCRNN_se: the local stream of CR networks in CRNN_se;
- GCRNN_se: the global stream of CR networks in CRNN_se.

Table 6 lists the average of the eight AUC values for the ablation study. In addition, the detailed eight AUC values on each dataset are presented in the Appendix A. It is suggested that either the local CR networks or the global CR networks have outstanding detection performance. Their average $\mathrm{AUC}_{(\mathrm{D,F})}$ values are all above 0.9940, which are obviously better than the other eight comparison algorithms. Compared with other six scenarios of CRNN+, LCRNN, GCRNN, CRNN_se, LCRNN_se, and GCRNN_se, the average $\mathrm{AUC}_{(\mathrm{D,F})}$ scores of CRNN are, respectively, improved by 0.06%, 0.08%, 0.13%, 0.03%, 0.40%, and 0.10%.

Table 6. Average AUC values over six datasets for the ablation study. (The best is in bold).

	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
CRNN	**0.9982**	0.1534	$\mathbf{7.47 \times 10^{-6}}$	1.1516	**1.9982**	1.1534	2.1516	$\mathbf{1.03 \times 10^{7}}$
CRNN+	0.9976	**0.2880**	1.29×10^{-3}	**1.2856**	1.9963	**1.2867**	**2.2843**	1.01×10^{4}
LCRNN	0.9973	0.2503	1.61×10^{-3}	1.2475	1.9956	1.2487	2.2459	1.04×10^{4}
GCRNN	0.9968	0.1921	2.51×10^{-4}	1.1889	1.9965	1.1919	2.1887	1.03×10^{4}
CRNN_se	0.9978	0.1400	3.77×10^{-5}	1.1377	1.9977	1.1399	2.1377	1.24×10^{6}
LCRNN_se	0.9942	0.2770	3.61×10^{-3}	1.2712	1.9906	1.2734	2.2676	1.25×10^{2}
GCRNN_se	0.9972	0.1912	6.45×10^{-4}	1.1884	1.9966	1.1906	2.1878	2.13×10^{4}

(1) Global CR and Local CR

The anomaly detection maps for six datasets of LCRNN and GCRNN are shown in Figure 5. Compared with the global CR networks, the detection results of the local CR networks present more false alarm targets and worse suppression on the background. This may be, mainly, because the local CR networks are composed of the local background dictionary, which is more likely to highlight specific areas in the background, such as the white area in HYDICE, the runway in Gulfport, and the roof in SanDiego. In addition, the anomaly scores of different targets in the results of local CR networks are relatively close, while the anomaly scores of different targets vary greatly in the results of global CR networks.

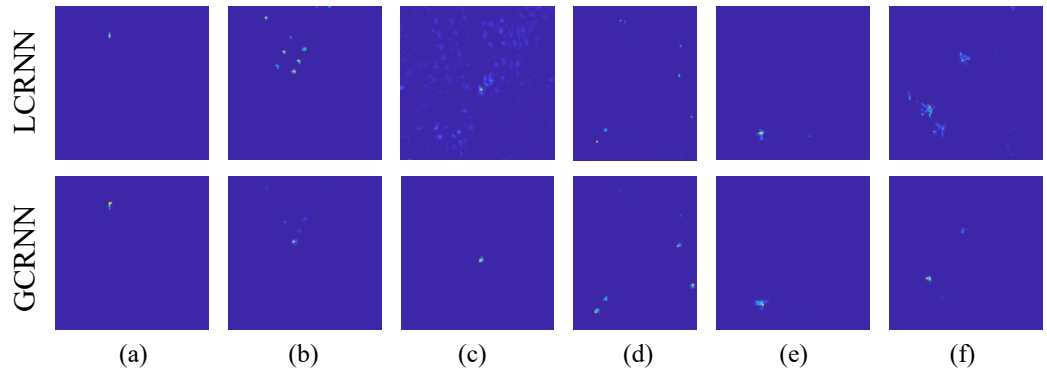


Figure 5. Detection maps of LCRNN and GCRNN. (**a**) Beach. (**b**) Urban. (**c**) XiongAn. (**d**) HYDICE. (**e**) Gulfport. (**f**) SanDiego.

(2) Detection Fusion of Global CR and Local CR

From the quantitative perspective, CRNN obtained the best $AUC_{(D,F)}$ values when compared to LCRNN and GCRNN. The results of separate training versions also presents the similar findings that the average $AUC_{(D,F)}$ values of CRNN_se have, respectively, improved by 0.36% and 0.06% compared with LCRNN_se and GCRNN_se. This demonstrates that the detection results fused with the local and global ones is better than that of the two alone. Similar to the random forest theory [75], it is possible to create a stronger and better detector by comprehensively analyzing the results of several detectors with good detection performance. In addition, the multiplicative fusion strategy used in proposed CRNN is better than additive fusion in CRNN+. As shown in Figure 5, the background suppression in the result maps of LCRNN is not as good as that of GCRNN, so multiplicative fusion can offer the advantages of global result maps and better background suppression compared with additive fusion.

(3) Joint Training and Separate Training

According to the average $AUC_{(D,F)}$ value, GCRNN is 0.04% smaller compared with GCRNNse, while the score of LCRNN is 0.3% higher than LCRNNse. For the final fusion

result, the $AUC_{(D,F)}$ score of CRNN is improved by 0.03% compared with the separate training version. Therefore, the joint training of the local and global streams of the networks is better than separate training in general. In our opinion, this may be because of the comprehensive usage of the global and local streams with a better balance in joint training. It optimizes each part in the networks through the objective function of L_{CR}, and extracts features more conducive to detection performance.

5.2. Parameter Analysis

Under the hyperparameters configured in Section 4.1.4, the performance of CRNN is mainly related to two key parameters, C as the amount of the global background dictionary atoms and S as the amount of the local background dictionary atoms, where S is mainly determined by the scale of the dual windows.

Since the size of C affects the ability of the global dictionary to generalize the background of the entire image to a certain extent, it is vital to set C at the appropriate value to ensure the comprehensiveness of the background and avoid anomalous pollution. The parameter C is analyzed in the range of {5, 10, 15, 20}. In Figure 6, it is shown by multiple experiments that the anomaly detection performance reaches the optimum when C is 15. In addition, the analysis of parameter C on the eight types of AUC values for the six datasets are listed in the Appendix B.

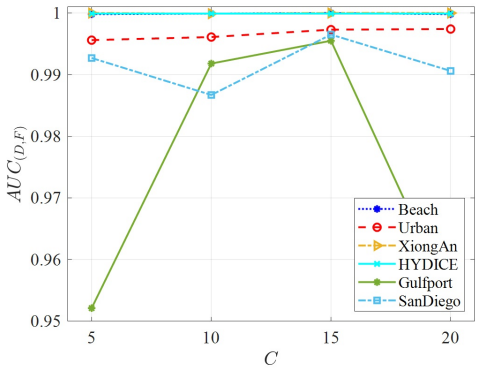


Figure 6. Analysis of parameter C on $AUC_{(D,F)}$ value.

The detection performance of the local CR networks is mainly affected by the scale of inner and outer windows, which is generally correlated with the targets size. The larger the size of the windows presents a higher computational cost and a slower inference speed for the networks. In the experiments, the range of the inner window size is set in [3, 9], and the range of the outer window size is in [2 + win_{in}, 8 + win_{in}] relatively. As shown in Figure 7, for the Beach, Urban, XiongAn and HYDICE datasets that only contain small targets, the detection results are not sensitive to the scale of the inner and outer windows, and the $AUC_{(D,F)}$ values are all above 0.99. For the Gulfport and SanDiego datasets with planar targets, the $AUC_{(D,F)}$ values remain at a suboptimal level when the scale of inner windows is much smaller than the size of anomalies, and they are significantly reduced when the inner window size is excessively wider than the anomaly. However, the selection of the optimal window size still involves a certain dynamic range. According to the experimental results, it is suggested that the range of the inner window is set in [5, 7], and the range of outer window is [9, 11] for the planar target.

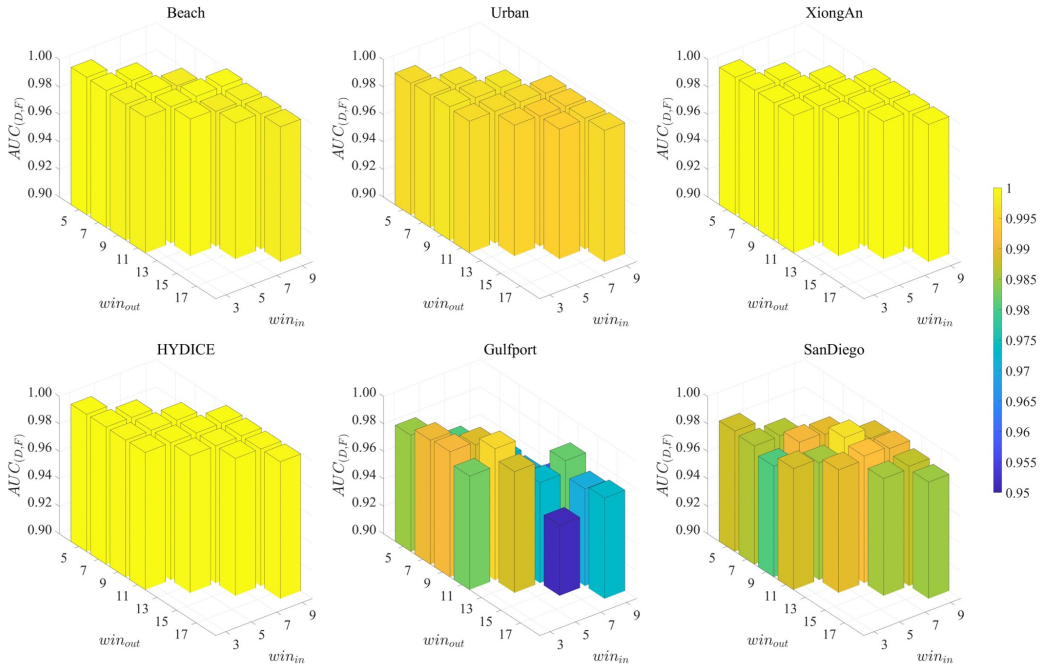


Figure 7. Analysis of parameters win_{in} and win_{out} over $\mathrm{AUC}_{(\mathrm{D,F})}$ value on six datasets.

6. Conclusions

In this paper, we proposed a deep learning-based hyperspectral anomaly detection method named CRNN, which implements CR theory into neural networks. CRNN mines the deep features that are helpful for the background construction and representation weight optimization in CR, which enhances the detection performance and efficiency. Firstly, an autoencoder is introduced to reconstruct the original hyperspectral data and obtain a dense expression. Based on the hidden feature map in the AE, the feature extraction network is used to further extract its comprehensive feature. Then, the two-stream global and local CR learning networks are ideally designed to learn the representation data of the hidden features under the objective function of CR. Finally, the anomaly detection result is contributed by fusing the residual maps of original and representation data in the two streams. Compared with state-of-the-art methods, experiments on six real hyperspectral images show that CRNN presents a perfect balance on detection accuracy and false alarm rate with a faster inference speed. In the future, we may try a more reasonable structure for CR learning networks instead of only applying convolutional networks to learn features, and also study a better fusion strategy to balance the global and local detection results.

Author Contributions: Conceptualization, Y.D., T.O., and J.W.; methodology, Y.D. and T.O.; software, Y.D. and T.O.; validation, Y.D., T.O., and J.W.; formal analysis, Y.D.; investigation, Y.D. and T.O.; resources, T.O. and J.W.; data curation, Y.D. and T.O.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D. and J.W.; visualization, Y.D.; supervision, J.W.; 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 under Grant YWF-21-JC-01.

Data Availability Statement: Publicly available datasets were analyzed in this study. The Beach, Urban, and Gulfport datasets are available at: http://xudongkang.weebly.com/ (accessed on 6 April 2023). The XiongAn dataset is available at: http://www.hrs-cas.com/a/share/shujuchanpin/2019/0501/1049.html (accessed on 6 April 2023). The HYDICE dataset is publicly available at: https://sites.google.com/site/feiyunzhuhomepage/ (accessed on 6 April 2023).

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

Appendix A

Table A1. AUC values for the ablation study on Beach dataset. (The best is in bold).

	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
CRNN	**1.0000**	0.2803	$\mathbf{7.75 \times 10^{-9}}$	1.2803	**2.0000**	1.2803	2.2803	$\mathbf{3.62 \times 10^{7}}$
CRNN+	**1.0000**	**0.4354**	8.35×10^{-6}	**1.4354**	**2.0000**	**1.4354**	**2.4354**	5.21×10^{4}
LCRNN	0.9999	0.2663	5.00×10^{-6}	1.2662	1.9999	1.2663	2.2662	5.32×10^{4}
GCRNN	**1.0000**	0.4141	7.97×10^{-6}	1.4141	**2.0000**	1.4141	2.4141	5.20×10^{4}
CRNN_se	**1.0000**	0.1791	2.66×10^{-8}	1.1791	**2.0000**	1.1791	2.1791	6.74×10^{6}
LCRNN_se	0.9984	0.2710	6.83×10^{-4}	1.2694	1.9978	1.2703	2.2688	3.97×10^{2}
GCRNN_se	**1.0000**	0.2968	2.72×10^{-6}	1.2968	**2.0000**	1.2968	2.2968	1.09×10^{5}

Table A2. AUC values for the ablation study on Urban dataset. (The best is in bold).

	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
CRNN	0.9973	0.0899	2.45×10^{-5}	1.0872	1.9973	1.0899	2.0872	3.66×10^{3}
CRNN+	0.9964	0.1604	1.06×10^{-4}	1.1567	1.9963	1.1603	2.1566	1.51×10^{3}
LCRNN	0.9960	0.2255	1.22×10^{-4}	1.2215	1.9958	1.2254	2.2214	1.85×10^{3}
GCRNN	0.9961	0.0640	7.01×10^{-5}	1.0602	1.9961	1.0640	2.0601	9.14×10^{2}
CRNN_se	**0.9979**	0.0504	$\mathbf{1.30 \times 10^{-6}}$	1.0483	**1.9979**	1.0504	2.0483	$\mathbf{3.88 \times 10^{4}}$
LCRNN_se	0.9935	**0.2495**	2.09×10^{-3}	**1.2430**	1.9914	**1.2474**	**2.2410**	1.20×10^{2}
GCRNN_se	0.9964	0.0828	1.87×10^{-5}	1.0792	1.9964	1.0827	2.0791	4.43×10^{3}

Table A3. AUC values for the ablation study on XiongAn dataset. (The best is in bold).

	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
CRNN	**1.0000**	0.2718	$\mathbf{7.48 \times 10^{-6}}$	1.2718	**2.0000**	1.2718	2.2718	$\mathbf{3.63 \times 10^{4}}$
CRNN+	**1.0000**	0.5173	5.70×10^{-3}	1.5173	1.9943	1.5116	2.5116	90.7
LCRNN	0.9999	0.4930	8.02×10^{-3}	1.4929	1.9919	1.4850	2.4849	61.5
GCRNN	0.9998	0.2706	1.38×10^{-4}	1.2704	1.9997	1.2704	2.2703	1.96×10^{3}
CRNN_se	**1.0000**	0.3773	4.89×10^{-5}	1.3773	**2.0000**	1.3773	2.3773	7.71×10^{3}
LCRNN_se	0.9999	**0.5628**	6.30×10^{-3}	**1.5627**	1.9936	**1.5565**	**2.5565**	89.4
GCRNN_se	**1.0000**	0.4389	3.07×10^{-3}	1.4389	1.9969	1.4359	2.4359	1.43×10^{2}

Table A4. AUC values for the ablation study on HYDICE dataset. (The best is in bold).

	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
CRNN	**0.9999**	0.1570	$\mathbf{9.75 \times 10^{-7}}$	1.1570	**1.9999**	1.1570	2.1570	$\mathbf{1.61 \times 10^{5}}$
CRNN+	0.9990	**0.3518**	1.05×10^{-3}	**1.3508**	1.9979	**1.3508**	**2.3498**	3.36×10^{2}
LCRNN	0.9998	0.3038	4.51×10^{-4}	1.3037	1.9994	1.3034	2.3032	6.74×10^{2}
GCRNN	0.9981	0.2572	1.19×10^{-3}	1.2553	1.9969	1.2560	2.2541	2.15×10^{2}
CRNN_se	0.9997	0.1562	1.62×10^{-6}	1.1559	1.9997	1.1562	2.1558	9.63×10^{4}
LCRNN_se	0.9948	0.3070	2.94×10^{-3}	1.3018	1.9918	1.3041	2.2989	1.04×10^{2}
GCRNN_se	0.9990	0.2150	2.41×10^{-4}	1.2140	1.9988	1.2147	2.2138	8.92×10^{2}

Table A5. AUC values for the ablation study on Gulfport dataset. (The best is in bold).

	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
CRNN	0.9955	0.0843	$\mathbf{3.32 \times 10^{-9}}$	1.0799	1.9955	1.0843	2.0799	$\mathbf{2.54 \times 10^{7}}$
CRNN+	0.9961	0.1347	2.18×10^{-5}	1.1308	1.9961	1.1347	2.1308	6.19×10^{3}
LCRNN	0.9953	0.0934	1.45×10^{-5}	1.0887	1.9953	1.0933	2.0887	6.42×10^{3}
GCRNN	0.9904	0.0956	1.64×10^{-5}	1.0861	1.9904	1.0956	2.0861	5.81×10^{3}
CRNN_se	**0.9977**	0.0445	7.68×10^{-8}	1.0422	**1.9977**	1.0445	2.0422	5.79×10^{5}
LCRNN_se	0.9930	**0.2548**	7.33×10^{-3}	**1.2478**	1.9856	**1.2475**	**2.2404**	34.7
GCRNN_se	0.9963	0.0504	3.90×10^{-6}	1.0468	1.9963	1.0504	2.0467	1.29×10^{4}

Table A6. AUC values for the ablation study on SanDiego dataset. (The best is in bold).

	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
CRNN	**0.9965**	0.0368	$\mathbf{1.18 \times 10^{-5}}$	1.0333	**1.9965**	1.0368	2.0333	$\mathbf{3.11 \times 10^{3}}$
CRNN+	0.9940	**0.1284**	8.51×10^{-4}	**1.1224**	1.9932	**1.1275**	**2.1216**	1.51×10^{2}
LCRNN	0.9926	0.1197	1.05×10^{-3}	1.1123	1.9916	1.1186	2.1112	1.14×10^{2}
GCRNN	0.9960	0.0513	7.98×10^{-5}	1.0473	1.9959	1.0512	2.0472	6.43×10^{2}
CRNN_se	0.9915	0.0322	1.74×10^{-4}	1.0237	1.9913	1.0321	2.0236	1.85×10^{2}
LCRNN_se	0.9855	0.0167	2.31×10^{-3}	1.0022	1.9832	1.0144	1.9999	7.22
GCRNN_se	0.9915	0.0634	5.38×10^{-4}	1.0549	1.9909	1.0629	2.0544	1.18×10^{2}

Appendix B

Table A7. Analysis of parameter C on the eight AUC values. (The best of each dataset is in bold).

Dataset	C	$AUC_{(D,F)}$ ↑	$AUC_{(D,\tau)}$ ↑	$AUC_{(F,\tau)}$ ↓	AUC_{JAD} ↑	AUC_{JBS} ↑	AUC_{ADBS} ↑	AUC_{OADP} ↑	AUC_{SNPR} ↑
Beach	5	0.9999	0.1939	2.70×10^{-5}	1.1938	1.9998	1.1939	2.1937	7.17×10^{3}
	10	0.9999	0.2776	5.54×10^{-5}	1.2775	1.9998	1.2776	2.2775	5.01×10^{3}
	15	**1.0000**	**0.2803**	$\mathbf{7.75 \times 10^{-9}}$	**1.2803**	**2.0000**	**1.2803**	**2.2803**	$\mathbf{3.62 \times 10^{7}}$
	20	0.9998	0.2025	3.58×10^{-5}	1.2023	1.9998	1.2025	2.2023	5.66×10^{3}
Urban	5	0.9956	0.0668	$\mathbf{7.81 \times 10^{-6}}$	1.0624	1.9956	1.0668	2.0624	8.55×10^{3}
	10	0.9961	0.0479	2.03×10^{-5}	1.0439	1.9960	1.0479	2.0439	2.36×10^{3}
	15	0.9973	**0.0899**	2.45×10^{-5}	**1.0872**	1.9973	**1.0899**	**2.0872**	3.66×10^{3}
	20	**0.9974**	0.0879	7.96×10^{-6}	1.0853	**1.9974**	1.0879	2.0853	$\mathbf{1.10 \times 10^{4}}$
XiongAn	5	**1.0000**	0.2931	5.53×10^{-6}	1.2931	**2.0000**	1.2931	2.2931	5.30×10^{4}
	10	0.9999	**0.3401**	$\mathbf{1.77 \times 10^{-6}}$	**1.3401**	1.9999	**1.3401**	**2.3400**	$\mathbf{1.92 \times 10^{5}}$
	15	**1.0000**	0.2718	7.48×10^{-6}	1.2718	**2.0000**	1.2718	2.2718	3.63×10^{4}
	20	**1.0000**	0.2789	5.50×10^{6}	1.2789	**2.0000**	1.2789	2.2788	5.07×10^{4}
HYDICE	5	**0.9999**	0.1788	$\mathbf{3.61 \times 10^{-7}}$	1.1788	**2.0000**	1.1788	2.1788	$\mathbf{4.95 \times 10^{5}}$
	10	**0.9999**	0.1873	2.22×10^{-6}	1.1872	1.9999	1.1873	2.1872	8.43×10^{4}
	15	**0.9999**	0.1570	9.75×10^{-7}	1.1570	1.9999	1.1570	2.1570	1.61×10^{5}
	20	**0.9999**	**0.2152**	1.34×10^{-6}	**1.2151**	1.9999	**1.2152**	**2.2151**	1.60×10^{5}
Gulfport	5	0.9521	0.0098	2.54×10^{-4}	0.9619	1.9519	1.0095	1.9616	38.4
	10	0.9919	**0.0883**	2.99×10^{-5}	**1.0802**	1.9918	**1.0883**	**2.0802**	2.95×10^{3}
	15	**0.9955**	0.0843	$\mathbf{3.32 \times 10^{-9}}$	1.0799	**1.9955**	1.0843	2.0799	$\mathbf{2.54 \times 10^{7}}$
	20	0.9534	0.0077	7.88×10^{-4}	0.9611	1.9526	1.0069	1.9604	9.79
SanDiego	5	0.9928	0.0249	4.76×10^{-4}	1.0176	1.9923	1.0244	2.0172	52.3
	10	0.9866	0.0012	3.81×10^{-4}	0.9878	1.9862	1.0008	1.9874	3.12
	15	**0.9965**	**0.0368**	$\mathbf{1.18 \times 10^{-5}}$	**1.0333**	**1.9965**	**1.0368**	**2.0333**	$\mathbf{3.11 \times 10^{3}}$
	20	0.9906	0.0143	2.85×10^{-4}	1.0049	1.9903	1.0140	2.0046	50.2

References

1. Eismann, M.T.; Stocker, A.D.; Nasrabadi, N.M. Automated Hyperspectral Cueing for Civilian Search and Rescue. *Proc. IEEE* **2009**, *97*, 1031–1055.
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.
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.
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.
5. Nasrabadi, N.M. Hyperspectral target detection: An overview of current and future challenges. *IEEE Signal Process. Mag.* **2013**, *31*, 34–44.
6. 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.
7. 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.
8. Chang, C.I. Hyperspectral Anomaly Detection: A Dual Theory of Hyperspectral Target Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–20. https://doi.org/10.1109/TGRS.2021.3086768.
9. Chang, C.I. Target-to-Anomaly Conversion for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–28. https://doi.org/10.1109/TGRS.2022.3211696.
10. Li, K.; Ling, Q.; Qin, Y.; Wang, Y.; Cai, Y.; Lin, Z.; An, W. Spectral-Spatial Deep Support Vector Data Description for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. https://doi.org/10.1109/TGRS.2022.3144192.
11. Xie, W.; Zhang, X.; Li, Y.; Lei, J.; Li, J.; Du, Q. Weakly Supervised Low-Rank Representation for Hyperspectral Anomaly Detection. *IEEE Trans. Cybern.* **2021**, *51*, 3889–3900. https://doi.org/10.1109/TCYB.2021.3065070.
12. Su, H.; Wu, Z.; Zhang, H.; Du, Q. Hyperspectral anomaly detection: A survey. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 64–90.
13. 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.
14. Kwon, H.; Nasrabadi, N.M. Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 388–397.
15. 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, 24 May 2012; Volume 8390, pp. 803–814.
16. Li, W.; Du, Q. Decision fusion for dual-window-based hyperspectral anomaly detector. *J. Appl. Remote Sens.* **2015**, *9*, 097297.
17. 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.
18. Chang, C.I. Effective Anomaly Space for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–24. https://doi.org/10.1109/TGRS.2022.3161632.
19. Carlotto, M.J. A cluster-based approach for detecting human-made objects and changes in imagery. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 374–387.
20. 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.
21. 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.
22. 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.
23. Yuan, Y.; Ma, D.; Wang, Q. Hyperspectral anomaly detection by graph pixel selection. *IEEE Trans. Cybern.* **2015**, *46*, 3123–3134.
24. Tao, R.; Zhao, X.; Li, W.; Li, H.C.; Du, Q. Hyperspectral Anomaly Detection by Fractional Fourier Entropy. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 4920–4929. https://doi.org/10.1109/JSTARS.2019.2940278.
25. 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.
26. Xie, W.; Jiang, T.; Li, Y.; Jia, X.; Lei, J. Structure Tensor and Guided Filtering-Based Algorithm for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4218–4230. https://doi.org/10.1109/TGRS.2018.2890212.
27. Cheng, X.; Wen, M.; Gao, C.; Wang, Y. Hyperspectral Anomaly Detection Based on Wasserstein Distance and Spatial Filtering. *Remote Sens.* **2022**, *14*, 2730. https://doi.org/10.3390/rs14122730.
28. Shang, W.; Jouni, M.; Wu, Z.; Xu, Y.; Dalla Mura, M.; Wei, Z. Hyperspectral Anomaly Detection Based on Regularized Background Abundance Tensor Decomposition. *Remote Sens.* **2023**, *15*, 1679. https://doi.org/10.3390/rs15061679.
29. Li, L.; Li, W.; Qu, Y.; Zhao, C.; Tao, R.; Du, Q. Prior-Based Tensor Approximation for Anomaly Detection in Hyperspectral Imagery. *IEEE Trans. Neural Networks Learn. Syst.* **2022**, *33*, 1037–1050. https://doi.org/10.1109/TNNLS.2020.3038659.

30. Chang, C.I.; Lin, C.Y.; Chung, P.C.; Hu, P.F. Iterative Spectral–Spatial Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–30. https://doi.org/10.1109/TGRS.2023.3247660.
31. 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.
32. 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.
33. 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.
34. Li, W.; Du, Q. Collaborative representation for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 1463–1474.
35. 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.
36. 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.
37. Tu, B.; Li, N.; Liao, Z.; Ou, X.; Zhang, G. Hyperspectral anomaly detection via spatial density background purification. *Remote Sens.* **2019**, *11*, 2618.
38. 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.
39. Zhao, C.; Li, C.; Feng, S.; Su, N.; Li, W. A Spectral–Spatial Anomaly Target Detection Method Based on Fractional Fourier Transform and Saliency Weighted Collaborative Representation for Hyperspectral Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 5982–5997. https://doi.org/10.1109/JSTARS.2020.3028372.
40. Wu, Z.; Su, H.; Tao, X.; Han, L.; Paoletti, M.E.; Haut, J.M.; Plaza, J.; Plaza, A. Hyperspectral Anomaly Detection With Relaxed Collaborative Representation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–17. https://doi.org/10.1109/TGRS.2022.3190327.
41. Sun, W.; Liu, C.; Li, J.; Lai, Y.M.; Li, W. Low-rank and sparse matrix decomposition-based anomaly detection for hyperspectral imagery. *J. Appl. Remote Sens.* **2014**, *8*, 083641.
42. 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.
43. 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.
44. Cheng, T.; Wang, B. Graph and Total Variation Regularized Low-Rank Representation for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 391–406. https://doi.org/10.1109/TGRS.2019.2936609.
45. 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.
46. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459.
47. Wu, F.; Yang, X.H.; Packard, A.; Becker, G. Induced L2-norm control for LPV systems with bounded parameter variation rates. *Int. J. Robust Nonlinear Control* **1996**, *6*, 983–998.
48. Zhou, T.; Tao, D. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In Proceedings of the Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, WA, USA, 28 June 28–2 July 2011.
49. 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, Vancouver, Canada, 2–9 February 2021; Volume 35, pp. 4139–4146.
50. Hu, X.; Xie, C.; Fan, Z.; Duan, Q.; Zhang, D.; Jiang, L.; Wei, X.; Hong, D.; Li, G.; Zeng, X.; et al. Hyperspectral anomaly detection using deep learning: A review. *Remote Sens.* **2022**, *14*, 1973.
51. Lei, J.; Xie, W.; Yang, J.; Li, Y.; Chang, C.I. Spectral–spatial feature extraction for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8131–8143.
52. Lu, X.; Zhang, W.; Huang, J. Exploiting Embedding Manifold of Autoencoders for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 1527–1537. https://doi.org/10.1109/TGRS.2019.2944419.
53. Arisoy, S.; Nasrabadi, N.M.; Kayabol, K. GAN-based hyperspectral anomaly detection. In Proceedings of the 2020 28th European Signal Processing Conference (EUSIPCO), Amsterdam, The Netherlands, 18–21 January 2021, pp. 1891–1895.
54. 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.
55. 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.
56. Wang, S.; Wang, X.; Zhang, L.; Zhong, Y. Auto-AD: Autonomous Hyperspectral Anomaly Detection Network Based on Fully Convolutional Autoencoder. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. https://doi.org/10.1109/TGRS.2021.3057721.
57. Xie, W.; Fan, S.; Qu, J.; Wu, X.; Lu, Y.; Du, Q. Spectral Distribution-Aware Estimation Network for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–12. https://doi.org/10.1109/TGRS.2021.3089711.
58. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507.

59. Hinton, G.E.; Zemel, R. Autoencoders, minimum description length and Helmholtz free energy. *Adv. Neural Inf. Process. Syst.* **1994**, *6*, 3–30.
60. 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.
61. Xiang, P.; Ali, S.; Jung, S.K.; Zhou, H. Hyperspectral Anomaly Detection With Guided Autoencoder. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–18. https://doi.org/10.1109/TGRS.2022.3207165.
62. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial autoencoders. *arXiv* **2015**, arXiv:1511.05644.
63. Xie, W.; Liu, B.; Li, Y.; Lei, J.; Du, Q. Autoencoder and Adversarial-Learning-Based Semisupervised Background Estimation for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5416–5427. https://doi.org/10.1109/TGRS.2020.2965995.
64. Li, Y.; Jiang, T.; Xie, W.; Lei, J.; Du, Q. Sparse Coding-Inspired GAN for Hyperspectral Anomaly Detection in Weakly Supervised Learning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. https://doi.org/10.1109/TGRS.2021.3102048.
65. Huber, P.J. Robust estimation of a location parameter. In *Breakthroughs in Statistics*; Springer: Berlin, Germany, 1992; pp. 492–518.
66. 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.
67. Maas, A.L.; Hannun, A.Y.; Ng, A.Y.; et al. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the ICML, Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.
68. 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.
69. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 16–18 April 2019; pp. 7354–7363.
70. Jang, E.; Gu, S.; Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv* **2016**, arXiv:1611.01144.
71. Senling, C.Y.L.X.Y.W.; Peng, Z. Aerial hyperspectral remote sensing classification dataset of Xiongan New Area (Matiwan Village). *J. Remote Sens.* **2020**, *24*, 1299–1306. (In Chinese) https://doi.org/10.11834/jrs.20209065.
72. 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.
73. Chang, C.I. An Effective Evaluation Tool for Hyperspectral Target Detection: 3D Receiver Operating Characteristic Curve Analysis. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5131–5153. https://doi.org/10.1109/TGRS.2020.3021671.
74. Chang, C.I. Comprehensive Analysis of Receiver Operating Characteristic (ROC) Curves for Hyperspectral Anomaly Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–24. https://doi.org/10.1109/TGRS.2022.3211786.
75. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31.