

Empowering Change Vector Analysis with Autoencoding in Bi-temporal Hyperspectral Images

Annalisa Appice^{1,2}(✉), Nicola Di Mauro¹, Francesco Lomuscio¹, and Donato Malerba^{1,2}

¹ Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro via Orabona, 4 - 70126 Bari - Italy

² Consorzio Interuniversitario Nazionale per l'Informatica - CINI
annalisa.appice@uniba.it,
nicola.dimauro@uniba.it, f.lomuscio7@studenti.uniba.it,
donato.malerba@uniba.it

Abstract. Modern sensor technologies are capable of covering large surfaces of the Earth with exceptional spatial, spectral, and temporal resolutions. With the significant development of hyperspectral imaging technology, change detection, which discovers change knowledge about Earth surface, has emerged as a hot topic in remote sensing. In this paper, we propose a deep learning enhanced change detection methodology that leverages the power of traditional change vector analysis techniques by gaining in accuracy with autoencoding neural networks and clustering. Preliminary experiments performed evaluating the proposed methodology with benchmark data provide encouraging results, also when compared to recent state-of-the-art change vector analysis competitors.

1 Introduction

HyperSpectral Image analysis (HSI) for Change Detection (CD), HSI-CD in brief, is a process that identifies the difference between two, or multiple, hyperspectral images acquired with the same spectral sensor technology over the same geographical scene at different times. This process has relevant applications in assessing natural disasters, as well as monitoring crops. Although several CD and HSI analysis methods have been described in the recent literature, this does not mean that the HSI-CD problem can be readily solved.

The HSI-CD problem is made inherently complex by the high dimensionality of the HSI data. In particular, the existence of hundreds of narrow continuous bands makes computationally expensive identifying the changes in a high-dimensional feature space [15]. Another challenge is that the number of change detection datasets is very limited—acquiring a label for each pixel is labor-intensive and time-consuming [7].

To deal with both these challenges—the high dimensionality and the lack of ground truth labels—change vector analysis techniques have been already fitted

to CD-HSI problems. These techniques are mainly based on image differencing operators and rely on the computation of distances or angle between pairs of HSI pixels corresponding to the same position of the scene in the compared images. They commonly apply unsupervised threshold algorithms, such as in the Otsu’s algorithm [12], in order to separate changed regions from unchanged background based on the distance range [10]. Complex spatial-aware algorithms, e.g. watershed, have been recently explored in combination with distance mappers, in order to improve the accuracy of change vector detection techniques [9].

On the other hand, with the increasing popularity of deep artificial neural networks (NNs), growing interest has arisen in using deep learning also in HS image analysis due to its powerful learning ability in presence of high dimensions [14]. However, as a result of the limited number of labeled datasets, it is difficult to actually train the whole NN architecture with ground-truth label information. At present, change detection approaches based on deep learning can be mainly divided into the following three categories: a) a deep network is trained with pseudo-training samples yielded by other unsupervised change detection methods [18], b) a deep network is trained in an unsupervised way without a-priori labels [20], and c) a deep network, pre-trained from a different classification dataset, is subsequently used for change detection in an unsupervised way [2]. Nevertheless the current investigations of deep learning in HSI-CD problems are still insufficient.

Based upon the emerging developments of unsupervised deep learning approaches [16] and the success of change vector analysis techniques in HSI-CD [10, 9], we have decided to explore the use of NNs autoencoders in combination with the change vector analysis. Autoencoders have been already investigated as a deep learning approach for anomaly detection [21] also in HS image analysis [19]. The autoencoding technique has been also investigated in HSI classification [11]. However, a few studies consider autoencoders for change detection and they mainly apply autoencoders as a feature learning stage for training a classification model with ground-truth labels acquired in a supervised setting [5, 9]. Instead we design a fully unsupervised methodology, where we exploit an autoencoder NN to deal with the high dimension of HSI data by mapping HSI samples into a low-dimensional manifold, where changed samples can be accurately separated from the unchanged ones. In particular, we enhance the change information modeled by a traditional image differencing operator with the features engineered through the hidden encoded code of an autoencoder. We also resort to a Gaussian Mixture Model, in order to separate regions where a change is occurred in the scene from the unchanged background. Finally we exploit the spatial autocorrelation, in order to refine the change result and produce the CD map of the scene.

We note that autoencoding and change vector analysis techniques have already been explored in the HSI-CD imaging literature. However, to the best of our knowledge, the novelty of this study is the specific formulation adopted for autoencoding within the change vector analysis, as well as the effectiveness of the combination of these components in a methodology that actually outperforms

the accuracy of state-of-the-art change vector analysis competitors in benchmark bi-temporal HSI data. In particular, this study paves the way for proving that the proposed formulation of autoencoding, image differencing and clustering is an effective unsupervised means to accurately detect regions of changed pixels in a scene based of HS information. In general, our methodology gains in accuracy when compared to traditional change vector analysis techniques, which are also unsupervised, but neglecting NNs representations.

The remainder of this paper is organized as follows. The proposed methodology is illustrated in Section 2. Section 3 provides the details of the experiments, which are carried out in this study, and their results, along with important discussions, are reported. Finally, Section 4 draw the conclusions and future developments.

2 Methodology

The proposed methodology is named AICA – Autoencoding of bi-temporal Hyperspectral Images for Change Vector Analysis. It works on both \mathbf{X}^1 and \mathbf{X}^2 , which are bi-temporal HSI acquisitions of a $m \times n$ scene and outputs a CD map of the scene. Both \mathbf{X}^1 and \mathbf{X}^2 are two co-registered HS images, which describe spectral data acquired for the same geographic scene. They are acquired at two distinct consecutive time periods by using the same spectral sensor technology. Every HS image is represented as a tensor of $m \times n$ pixels described with k spectral bands. A pixel denotes an area of around a few square meters of the Earth’s surface—it is a function of the sensor’s spatial resolution—which is unequivocally referenced with spatial coordinates (i, j) with $1 \leq i \leq m$ and $1 \leq j \leq n$, according to the usual matrix representation. Specifically, for both \mathbf{X}^1 and \mathbf{X}^2 , the pixel indexed by row i and column j contains data sensed on that resolution cell over k spectral bands. Every spectral band is a numeric feature proportional to the surface reflectance and for a given band. The CD map is a binary $m \times n$ matrix assigning each pixel (i, j) of the scene to a binary label, i.e., “change” or 1, and “no-change” or 0.

In principle, the learning process performed by AICA is divided into into three stages:

1. performing change vector analysis by applying a distance operator, in order to compute pixelwise the image differencing between \mathbf{X}^1 and \mathbf{X}^2 ;
2. training an autoencoder NN architecture on the fusion of \mathbf{X}^1 and \mathbf{X}^2 and performing a clustering analysis of the scene based on information generated with image differencing and image fusion autoencoding;
3. producing the CD map from clustering and applying a spatial correction procedure based on the expected spatial autocorrelation of the CD labels.

2.1 Image differencing

Image differencing is performed by applying mathematical functions that compare pixel-by-pixel the spectral band values of both \mathbf{X}^1 and \mathbf{X}^2 , in order to

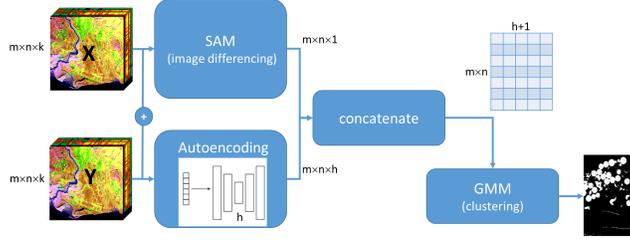


Fig. 1. AICA - Autoencoding of bi-temporal Hyperspectral Images for Change Vector Analysis

quantify the range of the change. Pixels associated with changed regions tend to produce distance values significantly higher than the ones associated to unchanged regions. In this study, we consider the Spectral Angle Mapper (SAM) to compute the distance as it is simple and fast to compute [6]. It has been widely used in change vector analysis [8–10] mainly for material identification. In addition, as pointed out by [13], the computation of the SAM algorithm is independent of the number of spectral bands and insensitive to sunlight. Let us consider the pixel in position (i, j) , the distance $\text{SAM}(i, j)$ measures the angle between the bi-temporal spectral vectors associated with (i, j) in both \mathbf{X}^1 and \mathbf{X}^2 , respectively. Formally, this angle is computed as follows:

$$\text{SAM}(i, j) = \arccos \frac{|\mathbf{X}_{ij}^1 \cdot \mathbf{X}_{ij}^2|}{|\mathbf{X}_{ij}^1| \cdot |\mathbf{X}_{ij}^2|}, \quad (1)$$

where \mathbf{X}_{ij}^1 and \mathbf{X}_{ij}^2 denote the vectors of spectral bands acquired for the pixel at position (i, j) of \mathbf{X}^1 and \mathbf{X}^2 , respectively.

2.2 Autoencoding and clustering

The autoencoding technique is computed by using a NN architecture consisting of an encoder function $\mathbf{h} = f(\mathbf{x})$ —mapping the input \mathbf{x} to an hidden code \mathbf{h} —and a decoder producing the reconstructed input $\hat{\mathbf{x}} = g(\mathbf{h})$, learned by minimizing a loss function $\mathcal{L}(\mathbf{x}, g(f(\mathbf{x}))) = \mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$ penalizing \mathbf{x} for being dissimilar from $\hat{\mathbf{x}}$ such as $\mathcal{L}_{\text{mse}}(\mathbf{x}, \hat{\mathbf{x}}) = 1/n \sum_i \|x_i - \hat{x}_i\|^2$ [1]. In this study, the autoencoder NN inputs all $m \times n$ pixels of the sensed scene spanned on the feature space that is derived by the fusion of \mathbf{X}^1 and \mathbf{X}^2 . Specifically, we use the fusion operator already considered in [9] computing $\mathbf{X}^- = \mathbf{X}_1 - \mathbf{X}_2$ so that the k -th fusion feature associated with the pixel in position (i, j) in \mathbf{X}^- is determined as follows:

$$X_{ijh}^- = X_{ijk}^1 - X_{ijk}^2, \quad (2)$$

where X_{ijk}^1 and X_{ijk}^2 denote the values measured at the k -th spectral band in vectors \mathbf{X}_{ij}^1 and \mathbf{X}_{ij}^2 , respectively.

After the completion of the autoencoder training on \mathbf{X}^- , we select the pixel representations on the encoding level \mathbf{h} of the trained autoencoder. These encoding features, augmented with the SAM feature and computed during the image differencing stage, are jointly processed across a clustering analysis. As a clustering algorithm, we consider the popular Gaussian Mixture Model (GMM) algorithm. Gaussian models have been already used as pivotal components of various HSI analysis systems [4]. They represent the probability density of the data with a weighted summation of a finite number of Gaussian densities with different means and standard deviations (or covariance matrices in case of multivariate GMM). In clustering, Gaussian models allow us to group data into a finite number of Gaussian clusters by modeling the cluster conditional probability with maximum likelihood. In particular, the GMM algorithm estimates the clustering parameters from the training data using the iterative Expectation-Maximization (EM) algorithm. Based upon the estimated parameters, the GMM algorithm determines the likelihoods of every pixel to belong to each clusters. However, in this study, instead of using decoupled two-stage training and the standard EM algorithm to determine likelihoods, we apply the approach recently introduced in [21]. This jointly optimizes the parameters of the deep autoencoder and the mixture model simultaneously in an end-to-end fashion, leveraging a separate estimation network to facilitate the parameter learning of the mixture model. In particular, this joint optimization helps the autoencoder escape from less attractive local optima and to a further reduction of reconstruction error. We fixed the number of clusters to 2, since we have to separate changed pixels from unchanged ones.

2.3 CD map production and spatial correction

After the pixels have been separated into two clusters— each pixel is assigned with the cluster for which the highest likelihood has been estimated—we have to decide the label (changed vs not-changes) to be assigned to each cluster and consequently to the pixels grouped in each cluster. To this purpose, we calculate the average change indicator of each cluster as the average SAM distance for each pixel assigned to the study cluster, that is:

$$\text{changel}(C_k) = \frac{\sum_{(i,j) \in C_k} \text{SAM}(i,j)}{|C_k|}, \quad (3)$$

where C_k is a pixel cluster and $|C_k|$ denotes the number of pixels grouped in C_k . According to the traditional theory inspiring change vector analysis techniques, pixels and, consequently clusters, with the highest value of SAM should delineate the changed regions in the study scene. Therefore, let $C_{max} = \underset{C_i, i=1,2}{\text{argmax}} \text{changel}(C_i)$ be the cluster associated with the highest average change indicator. Pixels are assigned to label “changed” if they are grouped in C_{max} , “not-changed” otherwise.

Final considerations concern the fact that the brute application of image differencing, autoencoding and clustering to produce the CD map will neglect the spatial arrangement of pixels. In particular, it may occasionally yield spurious, isolated assignments of pixels to clusters. To avoid this issue, we may apply the principle of local auto-correlation congruence of objects [17]—detected clusters are generally expanding over contiguous areas. Based on this principle, we may decide to change the assignment of pixels that strongly disagree with surrounding assignments. This mainly corresponds to perform a spatial-aware correction of the original clusters—based CD assignment. Based upon this correction, each pixel may be re-assigned to the cluster (and to its label) that originally groups the most part of its neighboring pixels.

3 Experimental study

We consider a benchmark co-registered bi-temporal HS dataset, in order to validate the effectiveness both in terms of accuracy and efficiency. This dataset has been already considered in the empirical studies of [10] and [9].

3.1 Data

The dataset is acquired by using the Hyperion sensor. This is a space-borne system carried on the EO-1 satellite, which includes 242 spectral bands covering wavelengths between 0.4 and 2.5 μm . The spectral range is divided into two intervals: the VNIR range (that includes 70 bands with wavelengths ranging from 356 to 1058 nm) and the SWIR range (that consists of 172 bands with wavelengths between 852 and 2577 nm). The spectral and spatial resolution of this sensor is about 10 nm and 30 m, respectively, over a 7.5-km strip. The originally sensed data (without any pre-elaboration) are available at the U.S. Geological Survey³. The study area covers an irrigated agricultural field in Hermiston City, Umatilla County, Oregon, USA. This area provides benchmark agricultural scenes, which are frequently used in the evaluation of the effectiveness of HS change detection algorithms. The land-cover types are soil, irrigated fields, river, buildings and types of cultivated land and grassland. The HS images are sensed at 2004 and 2007, respectively. The data consist of 390×200 pixels co-registered by using a multilayer fractional Fourier transform [3]. As no band selection is performed by the competitors, all 242 spectral bands are considered also in this study.

3.2 Implementation details and experimental set-up

AICA is written in Python 3.7. The autoencoder architecture consists of three hidden layers. The encoder reduces the input feature space through layers with 141 and 1 neurons in the bottleneck layer. The decoder maps the bottleneck signals back to the input space through two layers with 141 and 242 neurons.

³ <https://www.usgs.gov/>

Table 1. Performance analysis of AICA on Hermiston data: OA, FAR and MAR by varying the size of layer \mathbf{h} between 1 and 4 neurons in the autoencoding architecture.

\mathbf{h}	OA	FAR	MAR
1	0.9887	0.0045	0.0604
2	0.9390	0.0365	0.2269
3	0.9198	0.0893	0.0173
4	0.9070	0.1009	0.0384

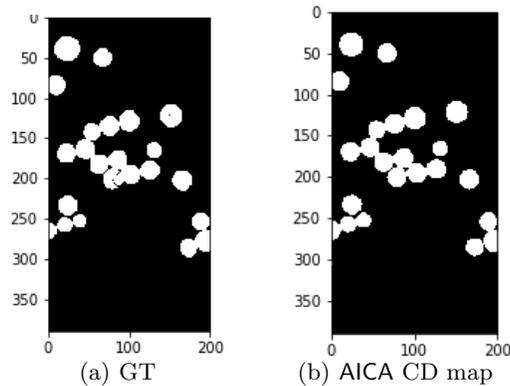


Fig. 2. Hermiston bi-temporal scene: ground truth (GT) in Figure 2(a) of the change occurred between 2004 and 2007; the separation of white changed objects from black background computed by AICA in Figure 2(b).

The mean squared error (mse) has been used as the loss function to be optimized with the ADAM algorithm. The hyperbolic tangent (\tanh) has been selected as activation function for each layer. The fitting procedure is iterated over 200 epochs.

The accuracy performance is evaluated with Overall Accuracy (OA), Missed Alarm Reate (MAR—changed pixels wrongly assigned to the un-changed background) and False Alarm Rate (FAR—un-changed pixels wrongly labeled within the region of the occurred change). The efficiency performance is evaluated with the computational time (TIME) spent in seconds completing the iterative learning process on notebook, CPU i7-6700 3.40 GHz, 16.0 GB RAM, running Windows 10 Pro. The performance metrics are measured on five trials.

3.3 Results and discussion

We start this investigation by analyzing the sensitivity of the performance of AICA to the size of the bottleneck layer \mathbf{h} in the autoencoding architecture of AICA. Results in terms of OA, FAR and MAR are reported in Table 1. They show that the highest performance is achieved when \mathbf{h} includes 1 neuron only. The CD map of this configuration is plotted in Figure 2(b). Our interpretation is that

Table 2. Performance analysis of spatial correction of AICA on Hermiston data: OA and TIME when AICA is run with both the spatial correction enabled (row 1) and the spatial correction disabled (row 2). AICA is run in the baseline configuration with $h = 1$.

spatial correction	OA	TIME (secs)
enabled	0.9887	372.25
disabled	0.9874	369.12

this configuration is actually able to isolate the information that describes at the best the change occurred in the study scene. As the CD accuracy progressively decreases as the size of \mathbf{h} increases, we draw the hypothesis that by augmenting the number of neurons in the level \mathbf{h} , useless noise is added to the representation of the change extracted through the autoencoding.

We proceed this investigation by verifying if the spatial correction actually gains in CD accuracy and how this operation affects the computation time spent processing the HSI data and producing the CD map. The OA and TIME of AICA are reported in Table 2. Results confirm the ability of spatial correction of improving the accuracy by correcting a few spurious cluster assignments. In addition, this gain in accuracy is at the expenses of a slight addition to the burden of computation. Finally, we also note that the entire CD process performed by AICA is completed spending about six minutes.

We conclude this analysis by comparing the accuracy of AICA to that of recent vector-analysis change competitors that have been reported in the literature and evaluated on this dataset [10, 9] respectively. As the authors of these papers have provided the pre-processed datasets they analysed, we are able to perform this comparative analysis in a safe environment, where the accuracy of the compared methods is actually evaluated under the same pre-elaboration of the data. In particular, authors in [10] report the performance of a fully unsupervised approach that combines the SAM distance algorithm with a threshold algorithm based on the Otsu’s approach. On the other hand, authors in [9] illustrate the performance of a fully unsupervised approach that combines a watershed algorithm with the SAM distance algorithm and the Otsu’s algorithm. Both competitors use spatial correction. Results are reported in Table 3. We note that AICA outperforms both competitors (included the one accounting for spatial information through watershed) in terms of both OA and FAR, while it performs slightly worse than competitors in terms of MAR. This means that the higher accuracy is coupled with a reduction of the number of false alarms, although this may cause an higher number of missed alarms. In general, the overall result confirms the effectiveness of our idea of enhancing the change vector analysis with the information extracted at the encoder level of an autoencoding architecture.

4 Conclusion

This paper illustrates a fully unsupervised methodology for analyzing bi-temporal HSI data of a geographical scene and detecting the changed regions of the scene.

Table 3. Compared competitors: approach (column 1), OA (column 3), FA (column 4) and MA (column 5).

competitor description	OA	FAR	MAR
SAM+Otsu[10]	0.9840	0.0122	0.0319
Watershed+SAM+Otsu[9]	0.9870	0.0078	0.0355
AICA	0.9887	0.0045	0.0604

The proposed methodology enhances the traditional change vector analysis, already considered in the recent literature to address HSI-CD problems, with relevant representation of change information extract through an autoencoding NN architecture. Spatial autocorrelation is also taken into account, in order to gain in CD accuracy. The preliminary experimental study is performed bi-temporal HSI data collected into a benchmark agricultural scene. They reveal that the proposed methodology is able to provide competitive accuracy compared to recent state-of-the-art change vector analysis techniques. In fact, with the encouraging performance of the proposed methodology, precise land-use land-cover (or cropping pattern) changes may be identified. Some directions for further work are still to be explored. New data scenarios may be considered for the empirical evaluation. In addition, the spatial information that is actually used in the post-processing phase only, may be also exploited during the autoencoding and clustering phases, in order to improve the ability of extracting a valuable representation of the change occurred in the data. Finally, we plan to expand the proposed methodology from bi-temporal to multi-temporal applications.

Acknowledgements

This work fulfills the research objectives of the PON Ricerca e Innovazione 2014–2020 project “CLOSE – Close to the Earth” (ARS01_001413), funded by the Italian Ministry for Universities and Research (MIUR). The authors thank Lopez Fandino Javier for providing the data used in the empirical evaluation of this study. These data are free available for download at <https://gitlab.citius.usc.es/hiperespectral/ChangeDetectionDataset>.

References

1. D. Charte, F. Charte, S. Garca, M. J. del Jesus, and F. Herrera. A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78 – 96, 2018.
2. K. L. de Jong and A. S. Bosman. Unsupervised change detection in satellite images using convolutional neural networks. *CoRR*, abs/1812.05815, 2018.
3. A. S. Garea, A. Ordonez, D. B. Heras, and F. Arguello. Hyperview: an open source desktop application for hyperspectral remote-sensing data processing. *International Journal of Remote Sensing*, 37(23):5533–5550, 2016.
4. U. B. Gewali, S. T. Monteiro, and E. Saber. Machine learning based hyperspectral image analysis: A survey. *CoRR*, abs/1802.08701, 2018.

5. M. Gong, H. Yang, and P. Zhang. Feature learning and change feature classification based on deep learning for ternary change detection in sar images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 129:212 – 225, 2017.
6. M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80:91 – 106, 2013.
7. X. Li, Z. Yuan, and Q. Wang. Unsupervised deep noise modeling for hyperspectral image change detection. *Remote Sensing*, 11(3), 2019.
8. S. Liu, L. Bruzzone, F. Bovolo, and P. Du. Hierarchical unsupervised change detection in multitemporal hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):244–260, 2015.
9. J. Lopez-Fandino, A. S. Garea, D. B. Heras, and F. Argüello. Stacked autoencoders for multiclass change detection in hyperspectral images. In *2018 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2018, Valencia, Spain, July 22-27, 2018*, pages 1906–1909. IEEE, 2018.
10. J. Lopez-Fandino, D. B. Heras, F. Arguello, and R. J. Duro. Cuda multiclass change detection for remote sensing hyperspectral images using extended morphological profiles. In *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, volume 1, pages 404–409, 2017.
11. S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du. Unsupervised spatial-spectral feature learning by 3d convolutional autoencoder for hyperspectral classification. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–13, 2019.
12. N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, 9:62–66, 1979.
13. S. T. Seydi and M. Hasanlou. A new land-cover match-based change detection for hyperspectral imagery. *European Journal of Remote Sensing*, 50(1):517–533, 2017.
14. A. Signoroni, M. Savardi, A. Baronio, and S. Benini. Deep learning meets hyperspectral image analysis: A multidisciplinary review. *Journal of Imaging*, 5(5), 2019.
15. A. Song, J. Choi, Y. Han, and Y. Kim. Change detection in hyperspectral images using recurrent 3d fully convolutional networks. *Remote Sensing*, 10(11), 2018.
16. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018:1–13, 2018.
17. J. Wang, S. Liu, and S. Zhang. A novel saliency-based object segmentation method for seriously degenerated images. In *2015 IEEE International Conference on Information and Automation*, pages 1172–1177, 2015.
18. Q. Wang, Z. Yuan, Q. Du, and X. Li. Getnet: A general end-to-end 2-d cnn framework for hyperspectral image change detection. *IEEE Transactions on Geoscience and Remote Sensing*, 57(1):3–13, 2019.
19. L. Zhang and B. Cheng. A stacked autoencoders-based adaptive subspace model for hyperspectral anomaly detection. *Infrared Physics and Technology*, 96:52 – 60, 2019.
20. W. Zhao, Z. Wang, M. Gong, and J. Liu. Discriminative feature learning for unsupervised change detection in heterogeneous images based on a coupled neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7066–7080, 2017.
21. B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.