Lightweight Autonomous Autoencoders for Timely Hyperspectral Anomaly Detection

Vinay Chakravarthi Gogineni, *Senior Member, IEEE*, Katinka Müller, Milica Orlandić, *Member, IEEE*, and Stefan Werner, *Fellow, IEEE*

***Abstract*— Autoencoders (AEs) have attracted significant attention for hyperspectral anomaly detection (HAD) in remote sensing applications due to their ability to unveil small, unique objects scattered across large geographical regions in an unsupervised manner. However, the training and inference processes of AEs are computationally demanding, posing challenges for efficient HAD in resource-constrained onboard applications. Various optimization techniques and parallel computing approaches have been proposed to alleviate the computational burden and enhance the feasibility of AEs for real-time applications in HAD. In this letter, we first present an efficient lightweight autonomous autoencoder (LAutoAE) that addresses the computational challenges of the autonomous hyperspectral anomaly detection autoencoder (AUTO-AD) while maintaining a similar anomaly detection accuracy. To further enhance the accuracy, we introduce LAutoAE+, which integrates kernel principal component analysis (KPCA)-based preprocessing methods with the LAutoAE. Experiments on diverse datasets demonstrate that the proposed LAutoAE and LAutoAE+ achieve comparable or superior detection performance compared with conventional Auto-AD, while also achieving reductions of 87% and 89.4%, respectively, in the number of learnable parameters.**

***Index Terms*— Anomaly detection, autoencoder, computational efficiency, hyperspectral imaging, lightweight architectures.**

I. Introduction

HYPERSPECTRAL imaging (HSI) facilitates the capture of detailed spectral information of objects or scenes across a wide range of wavelengths [1], [2]. Thus, it enables the identification of materials, detection of anomalies, and extraction of quantitative data, making it valuable in various fields such as agriculture [3], surveillance [4], and remote sensing [5], [6]. By leveraging spectral information in HSI, hyperspectral anomaly detection (HAD) aims to identify abnormal or anomalous features within a scene [7]. HAD methods can be broadly categorized into statistical and deep learning approaches [8]. Of these, statistical methods, including Reed-Xiaoli (RX) [9], kernel isolation forest detection (KIFD) [10], and spectral-spatial anomaly detection algorithm using an improved iForest detection (SSIIFD) [11] rely on predefined statistical models to estimate the distribution of data and identify anomalies based on deviations from the models. However, statistical methods often assume a specific distribution for the data, such as Gaussian or multivariate Gaussian, which may not always hold true for hyperspectral data. In contrast, deep learning approaches learn patterns directly from data without explicit statistical assumptions.

Autoencoders (AEs) have emerged as effective deep learning approaches for HAD, leveraging their ability to learn compact representations of high-dimensional spectral data and reconstruct the original HSI background in an unsupervised manner [12]. Notable AEs proposed for HAD include the robust graph AE (RGAE) [13], guided AE (GAE) [14], spectral difference guided graph attention autoencoder (SDGATA) [15], and the autonomous hyperspectral anomaly detection autoencoder (AUTO-AD) [16], in which the background is reconstructed and the anomalies appear as reconstruction errors. The RGAE enhances robustness against anomaly targets and noise by integrating graph regularization, whereas the GAE diminishes the AE's representation capability for anomaly targets through a guided module. The SDGATA consists of a graph attention (GAT) network encoder, guided by spectral differences, and a GAT decoder incorporating a background purification method to mitigate anomaly reconstruction. The AUTO-AD is a fully convolutional network that effectively mitigates the reconstruction of anomalous targets through an adaptive loss function. These models demonstrate superior detection performance compared to state-of-the-art methods in HAD; however, their practicality is often hindered by their high computational complexity, posing challenges related to power consumption, mechanical size, and heat dissipation when dealing with onboard detection [17].

To address this challenge, there is growing interest in the development of lightweight deep learning models specifically tailored for HAD. These models aim to minimize the computational burden without compromising anomaly detection accuracy. To this end, this letter proposes two lightweight AE models for HAD. The major contributions are as follows.

1) We propose a lightweight autonomous AE (LAutoAE) that effectively addresses the computational challenges associated with AUTO-AD. The proposed LAutoAE

Manuscript received 14 September 2023; revised 1 January 2024; accepted 8 January 2024. Date of publication 18 January 2024; date of current version 31 January 2024. This work was supported in part by the Research Council of Norway Project through ARIEL under grant 333229. *(Vinay Chakravarthi Gogineni and Katinka Müller contributed equally to this work.) (Corresponding author: Vinay Chakravarthi Gogineni.)*

Vinay Chakravarthi Gogineni is with the SDU Applied AI and Data Science, The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, 5230 Odense, Denmark (e-mail: vigo@mmmi.sdu.dk).

Katinka Müller and Milica Orlandić are with the Department of Electronic Systems, Norwegian University of Science and Technology, 7034 Trondheim, Norway (e-mail: katinkamul@gmail.com; milica.orlandic@ntnu.no).

Stefan Werner is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7034 Trondheim, Norway, and also with the Department of Information and Communications Engineering, Aalto University, 00076 Espoo, Finland (e-mail: stefan.werner@ntnu.no).

Digital Object Identifier 10.1109/LGRS.2024.3355471

achieves a remarkable 87% reduction in the number of learnable parameters compared to AUTO-AD while maintaining comparable performance.

2) To further enhance performance, we propose LAutoAE+, an enhanced version of LAutoAE. LAutoAE+ utilizes a kernel principal component analysis (KPCA)-based preprocessing block that can enhance the representation of anomalies. The LAutoAE+ achieves 89.4% reduction in the number of learnable parameters compared to AUTO-AD while achieving similar or even better performance.

II. Preliminaries

Consider a target HSI denoted by $\mathbf{X} \in \mathbb{R}^{H\times W\times B}$, where H and W represent the spatial dimensions, and B represents the spectral dimension. Utilizing AEs for HAD involves capitalizing on the inherent difficulty that AEs face in reconstructing anomalies. In AE-based HAD methods, the input HSI $\mathbf{X}'$ is encoded into a lower dimensional latent space $\mathbf{Z}$, i.e., $\mathbf{Z} = f_e(\mathbf{X}')$. The latent variable is then decoded into reconstructed HSI $\hat{\mathbf{X}}$ as $\hat{\mathbf{X}} = f_d(\mathbf{Z})$. Here, $f_e(\cdot)$ and $f_d(\cdot)$ describe encoder and decoder operations, respectively. Due to their significant deviation in spectral signatures compared to the surrounding areas, anomalies are challenging for AEs to reconstruct accurately. Consequently, anomalous pixels in the original HSI look like the background in the reconstructed HSI. Thus, in the detection map, i.e., $\mathbf{X} - \hat{\mathbf{X}}$, anomalies manifest as residuals.

A. Autonomous Hyperspectral Anomaly Detection (Auto-AD)

After a large number of iterations, AEs, however, can become proficient in reconstructing anomalies, resulting in poor detection performance. The recently proposed AUTO-AD [16] effectively addresses this issue by intelligently suppressing inadvertently reconstructed anomalies during the HAD process. The AUTO-AD model consists of a 15-layer encoder that takes a hypercube, filled with uniform noise ranging from 0 to 0.1, with the same dimensions as the target HSI. The encoder employs convolutional filters to reduce dimensionality in the spatial and spectral domains. On the other hand, the 11-layer decoder increases spatial and spectral dimensionality by utilizing convolutional layers and upsampling blocks that employ nearest-neighbor interpolation. The decoder takes the concatenation of the image code from the encoder and the outputs from the skip connections to generate the reconstructed background. The AUTO-AD effectively reconstructs the background by mitigating the anomaly reconstruction through an adaptive weight-loss function, defined as

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

In the above, the weight $w_{i,j}$ for the pixel at spatial position (i, j) in the target HSI is calculated as

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

where $\mathbf{E}$ is a matrix of size $H \times W$ with (i, j)th element $e_{i,j} = ||\mathbf{x}_{i,j} - \hat{\mathbf{x}}_{i,j}||_2^2$, and vec$(\cdot)$ represents the vectorization operation applied to the argument matrix. The weights $w_{i,j}$ are initialized to one and are updated according to (2) after a given number of iterations P, e.g., 100 in the AUTO-AD model. After each iteration, the loss will be fed backward to update the network parameters using the Adam optimizer [19].

Despite its excellent detection performance, AUTO-AD's substantial demand for computational resources limits its practicality in resource-constrained applications like onboard airborne and spaceborne platforms. Hence, there is a significant need to develop a lightweight AE that reduces computational requirements while maintaining a detection performance comparable to that of AUTO-AD.

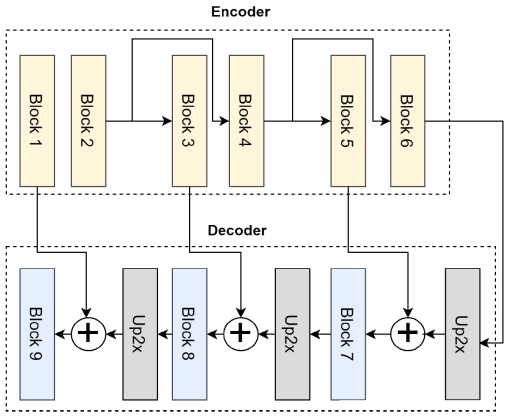

Fig. 1. LAutoAE architecture.

III. Proposed LAutoAE for HAD

The LAutoAE is a fully convolutional AE with skip connections, comprising only ten convolutional layers, which is a reduction of over half compared to the AUTO-AD model's 26 convolutional layers. Furthermore, as we move through the encoder layers from the leftmost to the rightmost side, there is a gradual decrease in the number of spectral bands. This leads to a significant reduction in the number of learnable parameters within the AE, resulting in lower computational costs. The architecture of the proposed LAutoAE is depicted in Fig. 1. It consists of six blocks in the encoder, three blocks in the decoder, and three upsampling blocks.

A. Encoder

The details of the encoder blocks are presented in Fig. 2(a). Each block in the encoder contains a convolutional layer that reduces the number of channels from input to output. The convolutional layer is denoted as conv$(C_{\text{in}} - C_{\text{out}}, k, s, p)$, where C_{in}, C_{out}, k, s, and p represent the number of input channels, number of output channels, kernel size, stride, and padding, respectively. The number of input channels for six encoder blocks are $\{B, B, E_1, E_1, E_2, E_2\}$, while the number of output channels are $\{E_1, E_1, E_2, E_2, E_3, E_3\}$, respectively. To ensure the effectiveness of the encoder, the input channels of these convolutional layers should satisfy the condition $B \geq E_1 \geq E_2 \geq E_3$. Furthermore, batch normalization is employed in each encoder block to stabilize the training process, and the LeakyReLU activation function is utilized to introduce

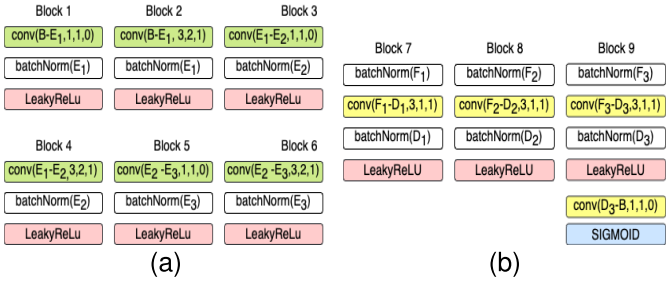

Fig. 2. Layer-wise details of the LAutoAE architecture. (a) Encoder. (b) Decoder.

nonlinearity to the network. Blocks 1, 3, and 5 complement the decoder with spatial details through skip connections.

B. Decoder

The details of the three decoder blocks are depicted in Fig. 2(b). Each of these three blocks consists of two batch normalization steps, one convolutional layer, and a LeakyReLU activation function. The ninth block includes an additional convolutional filter with a sigmoid activation function. This extra convolutional layer in ninth block ensures that the reconstructed background $\widehat{\mathbf{X}}$ contains the same number of spectral bands as that of input HSI $\mathbf{X}$. The sigmoid activation function restricts the network's output, representing the reconstructed background, to a range of [0, 1].

In addition to these three blocks, the decoder has three spatial upsampling blocks that utilize nearest-neighbor interpolation. These blocks ensure that the spatial dimensions of the outputs of the decoder blocks match those of the spatial dimensions of the outputs of the encoder blocks arriving via skip connections. They also guarantee that the reconstructed output of the decoder has the same spatial dimension as the input HSI fed to the proposed LAutoAE.

The LAutoAE architecture includes three skip connections that involve concatenating the encoder and decoder outputs. This enables the encoder to provide spatial information to complement the decoder. The participation of the first, third, and fifth blocks of the encoder in the concatenation process results in F_i number of input channels for the decoder blocks, given by

$$F_i = \begin{cases} E_3 + E_3, & \text{for } i = 1 \\ E_2 + D_1, & \text{for } i = 2 \\ E_1 + D_2, & \text{for } i = 3. \end{cases} \tag{3}$$

The number of input channels for the three decoder blocks is $\{F_1, F_2, F_3\}$, while the number of output channels is $\{D_1, D_2, B\}$, respectively. To ensure a better detection performance, we must set $B \geq D_3 \geq D_2 \geq D_1$.

IV. Proposed LAutoAE+ for HAD

As AEs tend to become adept at reconstructing anomalies through multiple iterations of the HAD process, it is advantageous to eliminate extreme outliers from the target HSI before initiating AEs training for HAD. This preprocessing serves as a purification step, preventing HAD models from excessively concentrating on outliers. Therefore, to further enhance the detection performance of LAutoAE, KPCA-based preprocessing can be integrated. The resulting model is referred to as LAutoAE+. The KPCA transforms the target HSI into a higher dimensional feature space, which can enhance the representation of anomalies that are not apparent in the raw data of the HSI. This enhanced representation can result in a higher separability between the anomalies and background pixels, leading to improved detection performance. A basic structure of the LAutoAE+ is presented in Fig. 3.

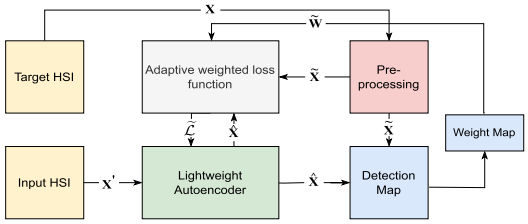

Fig. 3. Architecture and functionality of the kernel PCA-based LAutoAE+ for HAD.

Kernels enable the transformation of data $\mathbf{x}$ into a high-dimensional feature space denoted as $\phi(\mathbf{x})$, where it can be linearly separable [20]. KPCA closely resembles PCA, but it involves an additional step in which the data $\mathbf{x}$ is projected into a high-dimensional space using kernels. In this higher dimensional space, the kernel trick can be utilized to compute the inner product between two vectors $\mathbf{x}$ and $\mathbf{x}'$ without explicitly transforming the data [21]. A continuous, symmetric, and positive-definite kernel $\kappa(\cdot)$ satisfies the Mercer's condition. Commonly employed kernel functions include the radial basis function (RBF), Laplacian, and sigmoid [21]. The initial step in KPCA involves representing covariance matrix $\hat{\mathbf{C}}$ as an inner product between two transformed vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$ in the following manner:

$$\hat{\mathbf{C}} = \frac{1}{N} \sum_{i=1}^{N} \phi(\mathbf{x}_i) \phi^{\mathrm{T}}(\mathbf{x}_i') \tag{4}$$

where $N = W \times H$, representing the number of pixels in the HSI. For computing the principal components, the kernel matrix $\mathbf{K}$ can be set equal to the covariance matrix $\hat{\mathbf{C}}$ as defined in (4). The principal components are determined through eigenvalue decomposition, which is expressed as

$$\mathbf{KP} = \mathbf{\Lambda P} \tag{5}$$

where $\mathbf{P}$ comprises the eigenvectors of $\mathbf{K}$, also known as the principal components, and $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of $\mathbf{K}$ along its diagonal. For dimensionality reduction, only the initial ξ principal components, which contain the most variance, are employed to transform the data. Applying KPCA with ξ principal components to the original target HSI $\mathbf{X}$ yields a new target HSI $\tilde{\mathbf{X}} \in \mathbb{R}^{H \times W \times \xi}$, which is used instead of the original HSI to calculate the reconstruction loss $e_{i,j}$. Consequently, the new adaptive weight-loss function is given by

$$\mathcal{L} = \sum_{i=1}^{H} \sum_{i=j}^{W} ||(\widetilde{\mathbf{x}}_{i,j} - \hat{\mathbf{x}}_{i,j}) w_{i,j}||_2^2 \tag{6}$$

where the dimensionality of $\widetilde{\mathbf{x}}_{i,j}$ and $\hat{\mathbf{x}}_{i,j}$ is $\xi \times 1$.

TABLE I
AUC SCORES OF THE PROPOSED LAUTOAE AND KPCA-BASED LAUTOAE (LAUTOAE+), RX [9], KIFD [10], SSIFD [11], AND AUTO-AD [16] FOR VARIOUS ABU SCENES

ABU Scene	Captured Place	Image Size	RX	KIFD	SSIFD	AUTO-AD	LAutoAE	LAutoAE+
Airport 1	Los Angelas	$100 \times 100 \times 205$	0.8221	0.9192	0.8991	0.6941	0.8048	0.9474
Airport 2	Los Angelas	$100 \times 100 \times 205$	0.8404	0.9755	0.9616	0.6764	0.8324	0.9811
Airport 3	Los Angelas	$100 \times 100 \times 205$	0.9288	0.9553	0.9594	0.9210	0.8740	0.9622
Airport 4	Gulfport	$100 \times 100 \times 191$	0.9526	0.9914	0.9976	0.5509	0.9180	0.9954
Airport Avg.			0.8860	0.9604	0.9544	0.7106	0.8573	**0.9715**
Beach 1	Cat Island	$150 \times 150 \times 188$	0.9807	0.9849	0.9560	0.9605	0.9868	0.9894
Beach 2	San Diego	$100 \times 100 \times 193$	0.9106	0.9038	0.9383	0.9042	0.9617	0.9124
Beach 3	Bay Champagne	$100 \times 100 \times 188$	0.9998	0.9891	0.99997	0.9276	0.9916	0.9995
Beach 4	Pavia	$150 \times 150 \times 102$	0.9538	0.9738	0.9488	0.9898	0.9814	0.9761
Beach Avg.			0.9612	0.9629	0.9608	0.9455	**0.9804**	0.9693
Urban 1	Texas Coast	$100 \times 100 \times 204$	0.9907	0.9865	0.9697	0.9833	0.8436	0.9907
Urban 2	Texas Coast	$100 \times 100 \times 207$	0.9946	0.9863	0.9971	0.9940	0.8835	0.9894
Urban 3	Gainesville	$100 \times 100 \times 191$	0.9513	0.9550	0.9649	0.7663	0.8977	0.9845
Urban 4	Los Angelas	$100 \times 100 \times 205$	0.9887	0.9822	0.9899	0.9965	0.9485	0.9931
Urban 5	Los Angelas	$100 \times 100 \times 205$	0.9692	0.9772	0.9541	0.9620	0.8061	0.9797
Urban Avg.			0.9789	0.9775	0.9751	0.9415	0.8759	**0.9875**
Computational Time			1 sec.	98 sec.	28 sec.	362 sec.	**181** sec.	**181** sec.
Learnable Parameters						3250125	**429755**	**343025**

V. EXPERIMENTAL RESULTS

In this section, we demonstrate the detection performance and computational efficiency of the proposed LAutoAE and LAutoAE+ by conducting experiments on various HSI datasets from the public airport-urban-beach (ABU) [18]. In these datasets, HSI spatial dimensions vary from 100×100 to 150×150, with the number of spectral bands ranging from 102 to 207. All experiments were performed on a system equipped with 12th Gen Intel[1] Core[2] i5-12 500H processor operating at 2500 MHz. The system comprises 12 physical cores and 16 logical processors, with a total of 16GB RAM. The performance of the proposed HAD models is assessed using a 2-D-receiver operating characteristic (ROC) curve and the corresponding area under the curve (AUC) values are utilized as performance metrics [22].

The input and output channel configurations in the encoder blocks of LAutoAE and LAutoAE+ were determined through experimental testing. The experiments suggested that the spectral dimensionality reduction in the encoder enhances detection performance and reduces computational time. Therefore, we set $\{E_1, E_2, E_3\} = \{75, 50, 25\}$ in the encoder. In the decoder, we set to $\{D_1, D_2, D_3\} = \{25, 50, 75\}$. The kernel filter size in the convolutional layers was set to 3×3. In LAutoAE+, an RBF kernel was chosen for the KPCA-based preprocessing and only $\xi = 100$ principal components were used. The negative slope of the LeakyReLU activation function was set to 0.2. Furthermore, the HAD process was halted if the residual error does not improve for ten consecutive epochs. This method yielded an average number of epochs of 785 for AUTO-AD, 935 for the proposed LAutoAE, and 558 for LAutoAE+. The decrease in epochs, coupled with a reduction in learnable parameters (as given in Table I), signifies a notable reduction in computational cost.

The proposed HAD models along with the state-of-the-art statistical models and AUTO-AD model were simulated on the datasets mentioned above, and the corresponding AUC scores and an average over each scene are presented in Table I. The ROC plots of these models for the first Airport, second Beach, and third Urban scenes are presented in Fig. 4. From the average AUC scores of ABU scenes and ROC plots, it is evident that the AUTO-AD demonstrates inconsistent performance across different ABU scenes. In scenes with less varying backgrounds, such as the Urban 2 scene, the AUTO-AD achieves a high AUC score of 0.9940. However, in scenes with highly complex backgrounds like Airport 4, the AUC score drops significantly to just 0.5509. On the contrary, the proposed models, especially LAutoAE+, exhibit better detection accuracy across all scenes compared to AUTO-AD. The KPCA-based preprocessing in LAutoAE+ enhances the representation of anomalies, thereby resulting in excellent detection performance across all scenes, except for Beach 2. The AUC of LAutoAE+ is observed to be lower than that of other proposed LAutoAE in this particular scene. In Beach

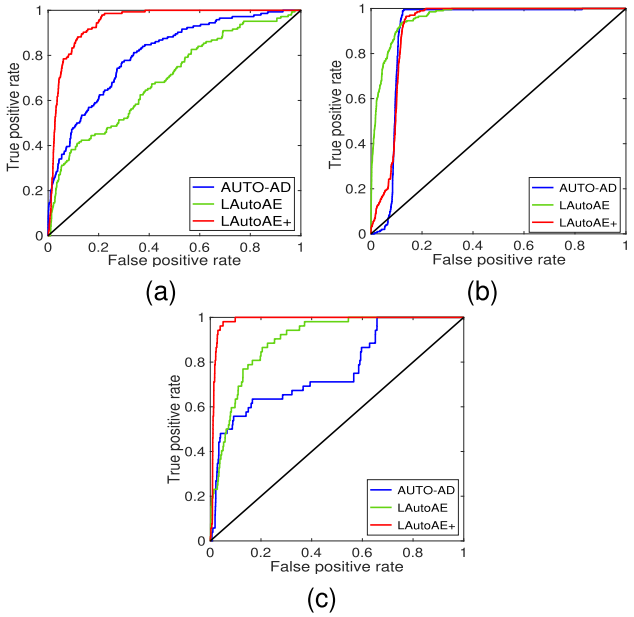

Fig. 4. ROC plots of the proposed HAD models and AUTO-AD for ABU scenes. (a) Airport 1. (b) Beach 2 (c). Urban 3.

[1]Registered trademark.
[2]Trademarked.

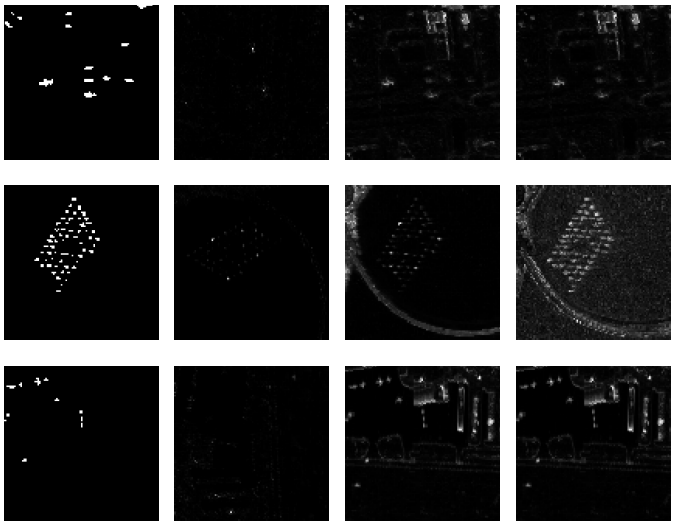

Fig. 5. (Top) Detection maps of Airport 1, (middle) Beach 2, and (bottom) Urban 3 for ABU scenes. (a) Ground truth. (b) AUTO-AD. (c) LAutoAE. (d) LAutoAE+.

2 scene, LAutoAE+ correctly classifies most of the anomalies but with slightly increased false alarm rate. Consequently, the AUC of LAutoAE+ is diminished for Beach 2 scene.

For Airport 1, Beach 2, and Urban 3 scenes, the detection maps generated by the proposed HAD models and AUTO-AD, along with the corresponding ground truth, are illustrated in Fig. 5. Based on these detection maps, it is evident that the LAutoAE+ model successfully eliminates a significant portion of the background, which is also confirmed through AUC scores. On the contrary, the AUTO-AD faces challenges in removing the background, especially noticeable in Airport 1 and Beach 2 scenes. This difficulty is also reflected in the AUC scores, where the AUTO-AD exhibits lower performance compared to other models. However, it is worth noting that while LAutoAE exhibits better AUC scores than AUTO-AD, it also inadvertently removes anomalies.

It is important to emphasize that the proposed HAD models achieved this improved performance even with reduced computational complexity. Under the aforementioned parameter settings, the proposed LAutoAE and LAutoAE+ models successfully reduce the learnable parameters by 87% and 89.4%, respectively. Furthermore, the computational time is decreased by 50% compared to that of the AUTO-AD model.

VI. Conclusion

In this letter, we proposed two lightweight HAD models that effectively address the computational challenges associated with AUTO-AD while maintaining comparable detection performance. We first introduced LAutoAE, which achieves detection performance similar to AUTO-AD while reducing the number of convolutional layers by over half. We further presented an enhanced version of LAutoAE, named LAutoAE+, which incorporates KPCA-based preprocessing into LAutoAE. This additional step improves detection performance by enhancing the representation of anomalies. Experiments conducted on diverse datasets revealed that the proposed LAutoAE and LAutoAE+ achieve similar or even better detection performance than Auto-AD, while requiring fewer computational resources. These findings highlight the potential of the proposed lightweight models to enable efficient HAD in resource-constrained airborne and spaceborne applications.

References

[1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 17–28, Jan. 2002, doi: 10.1109/79.974718.
[2] P. Ghamisi et al., "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.
[3] B. Lu, P. Dao, J. Liu, Y. He, and J. Shang, "Recent advances of hyperspectral imaging technology and applications in agriculture," *Remote Sens.*, vol. 12, no. 16, p. 2659, Aug. 2020.
[4] M. Shimoni, R. Haelterman, and C. Perneel, "Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 101–117, Jun. 2019.
[5] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.
[6] S.-E. Qian, "Overview of hyperspectral imaging remote sensing from satellites," in *Advances in Hyperspectral Image Processing Techniques*. Hoboken, NJ, USA: Wiley, 2023, pp. 41–66.
[7] S. Matteoli, M. Diani, and J. Theiler, "An overview of background modeling for detection of targets and anomalies in hyperspectral remotely sensed imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2317–2336, Jun. 2014.
[8] H. Su, Z. Wu, H. Zhang, and Q. Du, "Hyperspectral anomaly detection: A survey," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 64–90, Mar. 2022.
[9] I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Oct. 1990.
[10] S. Li, K. Zhang, P. Duan, and X. Kang, "Hyperspectral anomaly detection with kernel isolation forest," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 319–329, Jan. 2020.
[11] X. Song, S. Aryal, K. M. Ting, Z. Liu, and B. He, "Spectral–spatial anomaly detection of hyperspectral data based on improved isolation forest," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022, Art. no. 5516016.
[12] Y. Xu, L. Zhang, B. Du, and L. Zhang, "Hyperspectral anomaly detection based on machine learning: An overview," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 3351–3364, 2022.
[13] G. Fan, Y. Ma, X. Mei, F. Fan, J. Huang, and J. Ma, "Hyperspectral anomaly detection with robust graph autoencoders," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022, Art. no. 5511314.
[14] P. Xiang, S. Ali, S. K. Jung, and H. Zhou, "Hyperspectral anomaly detection with guided autoencoder," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2022, Art. no. 5538818.
[15] K. Li et al., "Spectral difference guided graph attention autoencoder for hyperspectral anomaly detection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–17, 2023.
[16] S. Wang, X. Wang, L. Zhang, and Y. Zhong, "Auto-AD: Autonomous hyperspectral anomaly detection network based on fully convolutional autoencoder," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022, Art. no. 5503314.
[17] N. Ma, X. Yu, Y. Peng, and S. Wang, "A lightweight hyperspectral image anomaly detector for real-time mission," *Remote Sens.*, vol. 11, no. 13, p. 1622, Jul. 2019.
[18] *ABU Data Set*. Accessed: Aug. 2023. [Online]. Available: http://xudongkang.weebly.com/data-sets.html
[19] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–13.
[20] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[21] K.-R. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.
[22] C. Ferri, J. Hernández-Orallo, and P. Flach, "A coherent interpretation of AUC as a measure of aggregated classification performance," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 657–664.