







A Parallel Algorithm for Hyperspectral Target Detection Based on Weighted Alternating Direction Method of Multiplier

Kun Yu , Shanshan Wu , Zebin Wu , Senior Member, IEEE, Jin Sun , Member, IEEE, Yi Zhang , Yang Xu , Member, IEEE, and Zhihui Wei , Member, IEEE

Abstract—Target detection for hyperspectral images (HSIs) is one of the significant techniques in remote sensing data processing. Targets generally comprise various object categories with complex features and of varying sizes. Target detection is often used in complex application scenarios in which accurately and efficiently acquiring detection results can be challenging. The development of advanced target detection approaches is becoming increasingly necessary in both military and civilian fields. This article proposes an alternating direction method of multiplier (ADMM)-based parallel approach for hyperspectral target detection. Different from existing methods performing target detection solely on HSIs, our approach performs the fusion of hyperspectral and multispectral data to leverage both spectral and spatial information prior. For each task or data partition, the parallel processing of the computation load on multiple computing nodes can substantially reduce the computation time. In addition, we introduce a novel weighted ADMM, which takes the influence of different variables on convergence into account, to further enhance the computational efficiency of the target detection model. Experiments on real-world HSI datasets demonstrate that our proposed parallel method not only produces more accurate detection results than direct detection methods, but also achieves significant acceleration ratio compared with the serial processing flow.

Index Terms—Alternating direction method of multiplier (ADMM), fusion, hyperspectral image (HSI), parallel, target detection.

Manuscript received 7 June 2023; revised 19 July 2023; accepted 20 August 2023. Date of publication 6 September 2023; date of current version 14 September 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62071233, Grant 61971223, and Grant 61976117, in part by the Jiangsu Provincial Natural Science Foundation of China under Grant BK20211570, Grant BK20180018, and Grant BK20191409, in part by the Fundamental Research Funds for the Central Universities under Grant 30917015104, Grant 30919011103, Grant 30919011402, Grant 30921011209, and Grant JSGP202204, and in part by the Key Projects of University Natural Science Fund of Jiangsu Province under Grant 19KJA360001. (Corresponding author: Zebin Wu.)

Kun Yu, Jin Sun, Yi Zhang, Yang Xu, and Zhihui Wei are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: 319106010139@njust.edu.cn; sunj@njust.edu.cn; yzhang@njust.edu.cn; xuyangth90@njust.edu.cn; gswei@njust.edu.cn).

Shanshan Wu is with the Nanjing Research Institute of Electronics Engineering (NRIEE), Nanjing 210000, China (e-mail: wushanshan@cetc.com.cn).

Zebin Wu is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: wuzb@njust.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3312523

I. INTRODUCTION

HYPERSPECTRAL images (HSIs), a 3-D data cube, with the abundant spectral and spatial information. Each pixel in HSIs consists of a continuous spectrum that is closely related to the material of the object and its environment. HSIs are mapped to the Earth at the spatial resolution of meters and sub-meters, and their outstanding advantage lies in the inclusion of fine spatial and texture features. In addition to using reflectance on different bands for radiation dimension analysis, it is more important to exploit texture, structure and context features to dig the relationships and rules contained in pixels, so as to provide support for ground object classification, target recognition, and scene analysis [1]. Over the past decades, hyperspectral target detection has been widely used and shown tremendous advantages in many fields [2], such as military target reconnaissance, vegetation research, and geological surveying. Hyperspectral target detection, which attracts a variety of signal processing and machine learning algorithms [3], [4], facing various challenges and emerging different solutions and strategies [5], [6], [7].

Assuming prior information about the target is available, the fundamental issue in hyperspectral target detection is to separate the target of interest from the background based on the spectral differences between different types of materials [8]. Specifically, target detection involves finding a scientific and effective method to measure and compare the spectral differences in the detection. The typical processing flow of classical hyperspectral target detection algorithms involves deriving decision functions through various mathematical analyses of the data, with the aim of highlighting the target and suppressing the background [9], [10]. To generate the detection output, the vector representation of the test pixels is input into the decision function and compared with a threshold to determine the presence of the target [11], [12]. The main methods for detection can be classified into four categories. The first category is based on spectral similarity measurement, with classical algorithms, such as spectral angle mapper [13]. This algorithm calculates the cosine of the angle between the target spectrum and the test spectrum, and if the cosine value is within a certain threshold, it indicates a high probability that the test spectrum is the target. The second category is based on projection, with projection operators being an important tool in hyperspectral data processing for highlighting

targets, suppressing noise, and improving signal-to-noise ratio. Representative algorithms include orthogonal subspace projection [14], which projects the spectral vector onto the orthogonal subspace of the background to achieve good detection performance. Another algorithm is constrained energy minimization (CEM) [15], which designs a finite impulse response filter that minimizes the average energy of background samples passing through the filter, while satisfying a constant filter response for the signal. The third category is based on statistical methods, which assume that the background and target follow different probability distributions and construct likelihood ratio tests, such as matched filtering [16] and adaptive matched subspace detector [17]. The last category is based on low-rank and sparse methods. The combined sparse and collaborative representation algorithm [10] uses sparse representation for the target and collaborative representation for the background to obtain detection results through the residual between the two representations. The low-rank regularized least squares-matched subspace detector [18] adds a low-rank constraint to the least squares method, limits the background samples, and generates results under the generalized likelihood ratio test.

Under reasonable assumptions, traditional algorithms exhibit robustness in hyperspectral target detection; however, most of them rely heavily on specific model assumptions [19]. If the detection algorithm or model does not sufficiently incorporate the authentic information of the land features, it can lead to a significant degradation in the performance of the detection process. In practice, hyperspectral data contain various types of objects, and the theoretical model of traditional methods is constrained by prior conditions [20], limiting its capacity to utilize data information, which substantially affects the performance of detection [21]. The detection method based on data fusion takes full advantage of the spectral information provided by hyperspectral data and, to a certain extent, compensates for the deficiency in spatial resolution inherent in such hyperspectral data. Senchuri et al. [22] proposed a multisensory hyperspectral and LiDAR data fusion approach for road edge detection utilizing machine learning algorithms, including support vector machines, random forests, and convolutional neural networks (CNN), in urban environments. Jones proposed a novel data fusion based approach for mapping quartz and carbonate veins in drill cores, leveraging diverse image analysis techniques in conjunction with the mineral abundance data derived from the unmixing of the thermal infrared and short-wave infrared spectra. The empirical results illustrate that the advanced data fusion strategy was successful in logging the quartz and carbonate veins [23].

In recent years, deep learning has rapidly developed. By constructing complex neural network models to analyze the features of hyperspectral datasets, deep learning has shown excellent performance in HSI processing [24], [25], [26]. With the emergence of large-scale datasets and the thriving development of deep neural networks in the field of visual images, hyperspectral target detection technology based on deep learning has been continuously innovated. By extracting deep abstract features, deep learning technology combines the advantages of imaging spectroscopy technology and 2-D target detection

technology, and further explores the spatial and spectral information of HSIs. Xie et al. [27] proposed a deep learning algorithm for spectral bands selection, which employs a variational autoencoder (VAE) to select the bands with abundant target information and reconstructs the image for detection. To better utilize the widely distributed background spectral elements in HSIs, they also proposed a background-aware target detection method, training the network on background pixels and reconstructing it for better detection. Wang et al. [28] proposed a target detection method based on CNN, which constructs a novel HSI change detection framework based on 3D-Wavelet domain active CNN. Zhu et al. [29] proposed a two-stream convolutional network-based target detector, which utilizes the two-stream convolutional networks to extract the spectral information in HSI.

HSI provides a comprehensive representation of objects due to the richness of information provided by the multiple spectral bands. However, it also leads to an extensive volume of data and redundant information in the image. Consequently, hyperspectral target detection methods often suffer from problems, such as long computation time and insufficient memory, on a single machine. Distributed machine learning provides an effective solution to these problems, as data and models are properly divided among different work nodes in a distributed environment. By designing efficient parallel communication schemes, different work nodes can cooperatively and simultaneously complete the calculation tasks. Ren and Chang [30] capitalized on the characteristic that the high-order automatic anomaly detection algorithm may descend into a local extreme point after randomly selecting the initial vector, facilitating parallel detection under diverse situations. Liu et al. [31] optimized the target detection algorithm based on low-rank sparse representation. They first segmented the image using the narrow dependency of hyperspectral data on the high-performance computing platform Spark, then used parallel clustering algorithm to cluster the pixels of HSIs, and finally computed the clustered data in parallel, significantly improving the speed and scalability of the method. In a similar vein, Zhang et al. [32] proposed a distributed and parallel implementation of the collaborative Tucker3 tensor decomposition for hyperspectral computational imaging, achieving exceptional accuracy and significantly enhancing computational efficiency when dealing with large-scale HSI datasets. Moreover, some researchers have harnessed the advantages of hardware to achieve parallel and efficient execution of HSI detection tasks. Du et al. [33] proposed an efficient RX method based on GPU, capitalizing on the rapid computing speed of the GPU and the recursive computing characteristics of the RX algorithm, vastly improving the efficiency of hyperspectral target detection. Macias et al. [34] presented a hardware optimized implementation for FPGAs of the automatic target detection and classification algorithm using the Gram-Schmidt method for orthogonalization purposes. In conclusion, the exploration and optimization of hyperspectral target detection methods leveraging distributed machine learning and hardware advancements are paving the way for solving current issues faced in handling large volumes of hyperspectral data efficiently and effectively.



Fig. 1. Target detection serial execution flow based on fusion.

HSI provides multilevel, multidirectional, and multitemporal information for features of land cover. It has been widely applied in various applications, such as urban planning, precision agriculture, and military reconnaissance. However, the processing of hyperspectral data poses significant challenges, such as a large number of spectral bands, large data volume, redundant information, and time-consuming processing on a single machine. In this situation, this article proposes a parallel target detection method after consulting aforementioned ideas. We first consider the problem of insufficient spatial resolution in HSIs, inspired by the CEM and data fusion method in [35], and we perform the fusion of hyperspectral data and multispectral data after the coarse detection. According to the fusion, high-resolution HSIs are obtained, which contain both spatial and spectral information and are beneficial for achieving high accuracy detection results. To overcome the difficulties of large data volume and insufficient computing resources, we develop a distributed implementation of the proposed method. In detection, the training process of VAE is performed in parallel based on data segmentation, and the weight is calculated by the average of each node. The execution of fusion is also in parallel manner, by coordinating tasks among multiple working nodes, the fusion process is performed simultaneously on each hyperspectral data block and its corresponding multispectral data block. ADMM as the optimized algorithm of fusion model, whose process is the most time-consuming part of our method, is modified to tackle the issue of multiple parameters alternating iteration. Different from the sequential update of variables in ADMM, our method takes into account the influence of each variable on convergence and the time required for each variable update. The update frequency of each variable is determined by its influence. Specifically, this work makes the following contributions.

- 1) We propose a novel object detection method to solve the problem of insufficient spatial resolution in HSIs.
- 2) We develop a parallel implementation to improve the processing efficiency of hyperspectral big data on high-performance computing platforms.
- 3) We propose an optimized ADMM to reduce the processing time for fusion by modifying the update frequency of variables. This approach effectively reduces unnecessary updates of unimportant variables, resulting in more efficient computation.

The rest of this article is organized as follows. Section II provides a detailed introduction of the proposed remote sensing image processing method. Section III presents a parallel implementation of the detection method. Section IV shows the experimental results that demonstrate the reliability and acceleration effects of the method. Finally, Section V concludes this article.

II. ALGORITHM

The flowchart of the proposed method is shown in Fig. 1. The input of algorithm includes an HSI, a corresponding multispectral image (MSI), and the target spectrum. The steps of the detection can be divided into the following.

- 1) Use a target detection method based on background reconstruction to perform rough detection on the HSI and preliminarily select the possible areas where the targets may exist.
- 2) Fuse the selected areas with the corresponding MSIs to obtain high-resolution HSIs.
- 3) Perform detailed detection on the fused image to obtain the corresponding pixels of the target and generate the detection results.

The following sections will introduce the serial processes of the algorithm.

A. Target Detection Process Based on Background Reconstruction

The well-known CEM algorithm has shown success in detection. Main idea behind the CEM is to design a finite impulse response filter that minimizes the energy of the background samples, subject to the constraint that the filter response to a known target signal is constant. Let $\mathbf{Y} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_H}] \in \mathbb{R}^{N_H \times L_H}$ denotes hyperspectral data matrix with N_H pixels and L_H spectral bands, prior signal $\mathbf{d} \in \mathbb{R}^{L_H \times 1}$ and filter $\mathbf{w} = [w_1, w_2, \dots, w_{L_H}]^T$. CEM solves the following problem:

$$\begin{aligned} \min \mathbf{w}^T \mathbf{R} \mathbf{w} \\ \text{s.t. } \mathbf{w}^T \mathbf{d} = 1 \end{aligned} \quad (1)$$

where $\mathbf{R} = (1/N)\mathbf{Y}\mathbf{Y}^T$ represents the autocorrelation matrix of the background. The constrained optimization problem above can be solved by Lagrange multiplier method and acquire the following equation:

$$\mathbf{w} = \frac{\mathbf{R}^{-1} \mathbf{d}}{\mathbf{d}^T \mathbf{R}^{-1} \mathbf{d}}. \quad (2)$$

After \mathbf{w} is obtained, calculate the spectral response value $\mathbf{w}^T \mathbf{Y}$. If the spectral value of corresponding pixel is exceeding than the threshold set in advance, the pixel is considered as the target. CEM treats the original image as the background for computing the linear filter while the number of background samples is far more than targets. Thus, there is a slight error between the result calculated and expected. To retrieve information solely from background samples, this article employs CEM detection algorithm based on background reconstruction. This method

generates data that closely mirror the original input, facilitating the reestablishment of background pixels. When leveraged for target detection in conjunction with background reconstruction, this approach exhibits superior performance. Remarkably, in certain image detection scenarios—such as those found in San Diego—it can surpass traditional CEM algorithms by up to five units in terms of area under curve (AUC) values. This highlights the considerable advantage of employing reconstructed background for target detection, suggesting an innovative and more accurate strategy in complex imaging environments. In order to ensure that the reconstructed samples contain only background information, and to make the reconstructed background more realistic, a constraint energy minimization regularization term is introduced in the VAE network. This allows the background samples to be reconstructed accurately, while the reconstruction error of nonbackground samples is relatively large. The specific formula is

$$\mathcal{L}_{\text{cem_loss}} = \sum_i \left(\frac{\mathbf{R}^{-1}\mathbf{d}}{\mathbf{d}^T \mathbf{R}^{-1}\mathbf{d}} \right)^T \mathbf{x}_i'. \quad (3)$$

As the loss function of detection method based on background reconstruction is

$$\mathcal{L} = \mathcal{L}_{\text{VAE}} + \rho \mathcal{L}_{\text{cem_loss}}. \quad (4)$$

\mathcal{L}_{VAE} belong to the original VAE, gradient descent (GD) can be used to optimize the model above efficiently.

In general, the process of target detection based on background reconstruction includes the following steps.

- 1) Conduct preliminary detection on the input hyperspectral data using the CEM algorithm.
- 2) Random select background samples for background reconstruction.
- 3) Perform target detection using the residual between the original image and the reconstructed background.

B. HSI Super-Resolution

HSI super-resolution is an important technology that generate high-resolution hyperspectral images (HR-HSIs) by fusing low-resolution hyperspectral images (LR-HSIs) and high-resolution multispectral images (HR-MSIs). HR-HSIs not only offer detailed object outlines, but also encompass rich spectral information that can be utilized to precisely identify object characteristics and aid in target detection. Therefore, target detection based on high-resolution HSIs is more accurate and better suited for important tasks such as military target detection. Due to the limitations of imaging sensors, it is difficult to balance the spectral resolution and spatial resolution of an image. For HSI, the spectral resolution is high, but the spatial resolution is low, while traditional panchromatic or MSIs achieves high spatial resolution. To obtain HR-HSIs, a more cost-effective solution is to fuse hyperspectral and MSIs. In addition to the methods extended from hypersharpening technology, HSI super-resolution mainly includes model-based methods [36], [37], [38], [39], [40] and learning-based methods [41], [42], [43], [44], [45].

The HR-HSI $\mathbf{X} \in \mathbb{R}^{N_M \times L_M}$ is obtained by fusing the LR-HSI $\mathbf{Y} \in \mathbb{R}^{N_H \times L_H}$ and the corresponding HR-MSI $\mathbf{Z} \in$

$\mathbb{R}^{N_M \times L_M}$. N_M and L_M are the number of pixels and bands in HR-MSI separately. \mathbf{Y} and \mathbf{Z} can be modeled as spatially degraded and spectrally degraded versions of the desired \mathbf{X} . Specifically, this can be described as

$$\mathbf{Y} = \mathbf{X}\mathbf{B}\mathbf{H} + \mathbf{E}_Y \quad (5)$$

$$\mathbf{Z} = \mathbf{R}\mathbf{X} + \mathbf{E}_Z \quad (6)$$

where \mathbf{B} represents the spatial blurring operator, \mathbf{H} represents the spatial downsampling operator, \mathbf{R} represents the spectral response of the multispectral sensor, and \mathbf{E}_X and \mathbf{E}_Y are the residuals. In some papers, the equation above can be combined as a regularization model composed of two fidelity, and written as

$$\min_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathbf{X}\mathbf{B}\mathbf{H}\|_F^2 + \frac{1}{2} \|\mathbf{Z} - \mathbf{R}\mathbf{X}\|_F^2 \quad (7)$$

where $\|\cdot\|_F$ is the Frobenius norm. As linear spectral mixing model (LSMM) introduced to the fusion, (7) can be rewritten as

$$\min_{\mathbf{A}, \mathbf{S}} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{S}\mathbf{B}\mathbf{H}\|_F^2 + \frac{1}{2} \|\mathbf{Z} - \mathbf{R}\mathbf{A}\mathbf{S}\|_F^2 \quad (8)$$

where \mathbf{A} represents the endmember matrix and \mathbf{S} represents the abundance matrix. To make sense physically, \mathbf{A} and \mathbf{S} are supposed to be nonnegative. For model-based methods, the main idea is to establish a regularization model based on the mechanisms of spectral and spatial degradation. In this article, the matrix decomposition fusion (MDF) framework is adopted, which decomposes the HSI matrix into two unconstrained matrices. Optimizing the following model to obtain \mathbf{A} and \mathbf{S} preliminary:

$$\min_{\mathbf{A}, \mathbf{S}} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F^2. \quad (9)$$

In this equation, \mathbf{A} and \mathbf{S} are not the final solution for the fusion, and the function of them is estimating \mathbf{X} . To measure the performance of an $\mathbf{A}\mathbf{S}$, MDF method using region-based low-rank regularization (RLR-MDF) introduces the root-mean-squared error (RMSE), which is a common metric used in regression analysis to measure the difference between predicted values and actual values. The RMSE is defined as follows:

$$\|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F^2 / \sqrt{N_H L_H}. \quad (10)$$

The RMSE results show that MDF produces more exact outputs than nonnegative matrix factorization. Specifically, RLR-MDF solves the following optimization problems:

$$\min_{\mathbf{A}, \mathbf{S}} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{S}\mathbf{B}\mathbf{H}\|_F^2 + \frac{1}{2} \|\mathbf{Z} - \mathbf{R}\mathbf{A}\mathbf{S}\|_F^2 + \lambda \mathcal{S}(\mathbf{S}) \quad (11)$$

where $\mathcal{S}(\mathbf{S})$ is the proposed regularizer

$$\mathcal{S}(\mathbf{S}) = \sum_{k=1}^K \|\mathbf{S}_{(k)}\|_*. \quad (12)$$

$\|\mathbf{S}_{(k)}\|_*$ denotes the nuclear norm, and k is the number of superpixel. Because \mathbf{Z} indicates the spatial information of \mathbf{X} reliably, it makes sense to transfer the spatial relationship of \mathbf{Z} to \mathbf{S} . As whole, the complete RLR-MDF can be divided into four steps: 1) initialize the spectral matrix. Since the spectral information is mainly contained in \mathbf{Y} , we can use the VCA

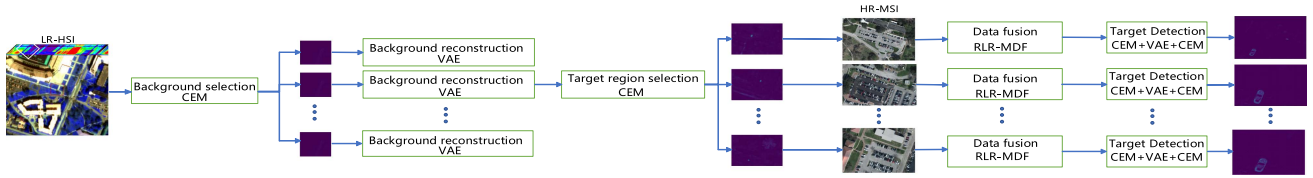


Fig. 2. Target detection parallel execution flow based on fusion.

algorithm to decompose \mathbf{Y} and obtain the initial matrix \mathbf{A} . 2) Estimate the spatial matrix \mathbf{S} , using ADMM to solve the problem described in (11), we can obtain \mathbf{S} . 3) update the spectral matrix \mathbf{A} based on the outputs from step 2. To acquire the results, solve the LS problem according to (13). 4) Reconstruction of HR-HIS using \mathbf{A} and \mathbf{S} directly.

$$\min_{\mathbf{A}} \|\mathbf{Y} - \mathbf{ASBH}\|_F^2 + \lambda \|\mathbf{A}\|_F^2. \quad (13)$$

Remarkably, the MDF framework bears some resemblance to the CS-based fusion methods. The first step can be viewed as identifying an appropriate conversion, while the second step can be seen as a procedure for injecting details. The third step is employed to adaptively modify the conversion, and the final step entails a reconstruction process.

III. PARALLEL IMPLEMENTATION

In this section, we describe the parallel version of our proposed algorithm, starting with a parallel flow of our proposed detection method. Then, we explain the parallel patterns of the VAE algorithm for background reconstruction. Finally, we provide a detailed description of the optimization of the ADMM algorithm in the fusion process.

A. Parallel Implementation of the Hyperspectral Target Detection

Fig. 2 shows the complete parallel process for high-resolution hyperspectral target detection based on background reconstruction. For the input hyperspectral data, background selection is performed first. The CEM algorithm is used to roughly determine the location of the target and select background pixels, then selected background pixels are partitioned for data-block training. During the VAE background reconstruction process, data are distributed to different nodes for training, and the model averaging (MA) method is used to reduce the training time. After training is completed, the results are consolidated and a new background spectrum is generated. Finally, the CEM algorithm is used again to detect the reconstructed background image and determine the area to which the target belongs. At this point, the coarse detection of hyperspectral targets is completed.

After obtaining the coarse detection results for hyperspectral targets, corresponding areas need to be selected from the multi-spectral data. Since there are usually many types of targets, such as vehicles, trees, and houses, the coarse-cutting results may contain multiple regions. In the fusion process, we use image regions as task units and assign different regions to different computing nodes for fusion and detection, achieving completely independent task parallelism. This greatly improves the task

acceleration ratio. On each computing node, the HR-HSI block and corresponding MSI are obtained first. Then, the RLR-MDF algorithm is used to fuse data and obtain a HR-HSI. During the fusion process, the proposed novel weighted ADMM method is used for computation acceleration. After the fusion is completed, target detection is performed on the HR-HSI to obtain the final clear detection results.

Different from the traditional target detection methods based on the model, this article uses the data fused from hyperspectral and MSIs. Compared with other methods, our approach yields clearer and higher accuracy results which takes full advantages of rich spectral information from hyperspectral data and the high spatial resolution from multispectral data. In order to accurately identify the target, we have implemented strategies pertaining to data parallelism. We fully exploit the characteristic of data redundancy in VAE and use MA strategy to achieve parallel optimization during the training process. In the fusion process, the parallel unit becomes the target region selected in the coarse detection, which ensures tasks on each node are independent without the need for communication. Furthermore, we also use a weight-based method to determine the update frequency of variables in ADMM during the fusion process, greatly reducing computation time. In summary, our approach achieves high accuracy and efficiency simultaneously.

B. Model-Averaging VAE

The algorithm proposed in this article includes two target detection processes based on background reconstruction, and utilizes VAE that requires input of background samples. However, since the background samples constitute a high proportion of the original image, the number of pixels require training is very large, leading to significant consumption of computational resources and time. Considering the characteristics of widely distributed background targets are numerous and complex, this article employs MA to perform parallel optimization of the GD in the VAE. MA is a simple and straightforward parallel mechanism that requires only one communication between nodes to interact with the models after the training process is completed. Each computing node can work independently without waiting for others. MA has excellent acceleration effects, almost achieving linear acceleration ratios. However, the model parameter bias is usually large since nodes do not need information exchange during the training process. But in this article, the input of VAE network used for background reconstruction is background samples, which have a large data volume and information redundancy. Even after segmentation, the sample distribution on every node is similar to the original image sample distribution.

Therefore, using the MA model for optimizing VAE training can not only speed up the training but also ensure the stability of accuracy after parallel computation. The formulation of MA can be written as follows:

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (14)$$

where \bar{y} represents the average predicted result, N represents the number of models or compute nodes, and $\sum_{i=1}^N y_i$ represents the sum of all predicted results from the models.

C. Novel Weighted ADMM

ADMM is used to solving constrained minimization problems, which is widely applied in fields, such as data mining, machine learning, and image analysis [46], [47], [48]. This method decomposes the original problem into several smaller, easier-to-solve sub-problems, and obtains the global solution through the collaboration of these subproblems. ADMM is primarily used to solve (11) in RLR-MDF. To accomplish this, three variables: \mathbf{V}_1 , \mathbf{V}_2 , and \mathbf{V}_3 are introduced, which allow us to express the original problem as follows:

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3} f(\mathbf{S}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3) \\ \text{s.t. } \mathbf{V}_1 = \mathbf{S}\mathbf{B}, \mathbf{V}_2 = \mathbf{S}, \mathbf{V}_3 = \mathbf{S} \end{aligned} \quad (15)$$

where

$$\begin{aligned} f(\mathbf{S}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3) = \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{V}_1\mathbf{H}\|_F^2 \\ + \frac{1}{2} \|\mathbf{Z} - \mathbf{R}\mathbf{A}\mathbf{V}_2\|_F^2 + \lambda \mathcal{S}(\mathbf{V}_3). \end{aligned} \quad (16)$$

The augmented Lagrangian function (15) can be written as

$$\begin{aligned} \mathcal{L}(\mathbf{S}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3, \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3) \\ = f(\mathbf{S}, \mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3) + \frac{\mu}{2} \|\mathbf{S}\mathbf{B} - \mathbf{V}_1 - \mathbf{D}_1\|_F^2 \\ + \frac{\mu}{2} \|\mathbf{S} - \mathbf{V}_2 - \mathbf{D}_2\|_F^2 + \|\mathbf{S} - \mathbf{V}_3 - \mathbf{D}_3\|_F^2 \end{aligned} \quad (17)$$

where $\mu > 0$ is the penalty parameter, the auxiliary variables $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3 \in \mathbb{R}^{J \times N_H}$ are used in an iterative process to refine the optimization procedure, and J is the number of endmember in HSI. The iterative process can be expressed as follows.

1) *Solving S subproblem:* Optimizing \mathcal{L} associated with \mathbf{S} can be written as

$$\begin{aligned} \mathbf{S}^{(t+1)} = \arg \min_{\mathbf{S}} \frac{\mu}{2} \|\mathbf{S}\mathbf{B} - \mathbf{V}_1^{(t)} - \mathbf{D}_1^{(t)}\|_F^2 \\ + \frac{\mu}{2} \|\mathbf{S} - \mathbf{V}_2^{(t)} - \mathbf{D}_2^{(t)}\|_F^2 + \frac{\mu}{2} \|\mathbf{S} - \mathbf{V}_3^{(t)} - \mathbf{D}_3^{(t)}\|_F^2 \end{aligned} \quad (18)$$

and the formula for solving the above problem is

$$\begin{aligned} \mathbf{S}^{(t+1)} = (\mathbf{B}\mathbf{B}^T + 2\mathbf{I})^{-1} ((\mathbf{V}_1^{(t)} + \mathbf{D}_1^{(t)})\mathbf{B}^T \\ + \mathbf{V}_2^{(t)} + \mathbf{D}_2^{(t)} + \mathbf{V}_3^{(t)} + \mathbf{D}_3^{(t)}). \end{aligned} \quad (19)$$

This computation can be carried out efficiently using the fast Fourier transform (FFT), while the first term $(\mathbf{B}\mathbf{B}^T + 2\mathbf{I})^{-1}$ can be precomputed in advance.

2) *Solving V₁ subproblem:* Optimizing \mathcal{L} associated with \mathbf{V}_1 can be written as

$$\begin{aligned} \mathbf{V}_1^{(t+1)} = \arg \min_{\mathbf{V}_1} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{V}_1\mathbf{H}\|_F^2 \\ + \frac{\mu}{2} \|\mathbf{S}^{(t+1)}\mathbf{B} - \mathbf{V}_1 - \mathbf{D}_1^{(t)}\|_F^2 \end{aligned} \quad (20)$$

and the formula for solving the above problem is

$$\begin{aligned} \mathbf{V}_1^{(t+1)}\mathbf{H} = (\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1} \\ (\mathbf{A}^T\mathbf{Y} + \mu(\mathbf{S}^{(t+1)}\mathbf{B} - \mathbf{D}_1^{(t)}))\mathbf{H} \end{aligned} \quad (21)$$

$$\mathbf{V}_1^{(t+1)}\bar{\mathbf{H}} = (\mathbf{S}^{(t+1)}\mathbf{B} - \mathbf{D}_1^{(t)})\bar{\mathbf{H}}. \quad (22)$$

RLF-MDF utilizes the masking matrix \mathbf{H} to partition \mathbf{V}_1 into two submatrices, $\mathbf{V}_1\mathbf{H}$ and $\mathbf{V}_1\bar{\mathbf{H}}$, where $\bar{\mathbf{H}}$ is the complement matrix of \mathbf{H} . FFT can be employed to expedite the computation process effectively, while the $(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}$ and $\mathbf{A}^T\mathbf{Y}$ can be precomputed beforehand.

3) *Solving V₂ subproblem:* Optimizing \mathcal{L} associated with \mathbf{V}_2 can be written as

$$\begin{aligned} \mathbf{V}_2^{(t+1)} = \arg \min_{\mathbf{V}_2} \frac{1}{2} \|\mathbf{Z} - \mathbf{R}\mathbf{A}\mathbf{V}_2\|_F^2 \\ + \frac{\mu}{2} \|\mathbf{S}^{(t+1)} - \mathbf{V}_2 - \mathbf{D}_2^{(t)}\|_F^2 \end{aligned} \quad (23)$$

and the formula for solving the above problem is

$$\begin{aligned} \mathbf{V}_2^{(t+1)} = (\mathbf{A}^T\mathbf{R}^T\mathbf{R}\mathbf{A} + \mu\mathbf{I})^{-1} (\mathbf{A}^T\mathbf{R}^T\mathbf{Z} \\ + \mu(\mathbf{S}^{(t+1)} - \mathbf{D}_2^{(t)})). \end{aligned} \quad (24)$$

4) *Solving V₃ subproblem:* Optimizing \mathcal{L} with respect to \mathbf{V}_3 can be written as

$$\begin{aligned} \mathbf{V}_3^{(t+1)} = \arg \min_{\mathbf{V}_3} \lambda \sum_{k=1}^K \|\mathbf{V}_{3(k)}\|_* \\ + \frac{\mu}{2} \|\mathbf{S}^{(t+1)} - \mathbf{V}_3 - \mathbf{D}_3^{(t)}\|_F^2 \end{aligned} \quad (25)$$

and the solution for solving the above problem is

$$\mathbf{V}_{3(k)}^{(t+1)} = \mathcal{D}_{\lambda/\mu}(\mathbf{S}_{(k)}^{(t+1)} - \mathbf{D}_{3(k)}^{(t)}) \quad (26)$$

where $\mathcal{D}_{\lambda/\mu}(\cdot)$ represents the singular value threshold function.

5) *Updating auxiliary multipliers:* The multipliers associated with \mathcal{L} is calculated as follows:

$$\mathbf{D}_1^{(t+1)} = \mathbf{D}_1^{(t)} - (\mathbf{S}^{(t+1)}\mathbf{B} - \mathbf{V}_1^{(t+1)}) \quad (27)$$

$$\mathbf{D}_2^{(t+1)} = \mathbf{D}_2^{(t)} - (\mathbf{S}^{(t+1)} - \mathbf{V}_2^{(t+1)}) \quad (28)$$

$$\mathbf{D}_3^{(t+1)} = \mathbf{D}_3^{(t)} - (\mathbf{S}^{(t+1)} - \mathbf{V}_3^{(t+1)}). \quad (29)$$

For RLR-MDF method, the most time-consuming step is solving (15), with a complexity of $\mathcal{O}(JN_H \log N_W + J^2N_H)$ per iteration for the overall process. N_W is width in hyperspectral

Algorithm 1: Weighted ADMM.

Require: LR-HSI Y , HR-MSI Z , B , H , R , λ , J , K , interval vector p , spectral matrix A and superpixel-level labels of Z 's pixels

- 1: Initial $S^{(0)}$, $V_1^{(0)}$, $V_2^{(0)}$, $V_3^{(0)}$, $D_1^{(0)}$, $D_2^{(0)}$, $D_3^{(0)}$ by zero matrix, $\mu = 0.05$, $maxIter = 200$, $\epsilon = 10^{-3}$
- 2: **for** each $i \in [1, maxIter]$ **do**
- 3: Update $S^{(t)}$ by (19)
- 4: **if** $i \% p_1 == 0$ **then**
- 5: Update $V_1^{(t)}$ by (21) and (22)
- 6: **end if**
- 7: **if** $i \% p_2 == 0$ **then**
- 8: Update $V_2^{(t)}$ by (24)
- 9: **end if**
- 10: **if** $i \% p_3 == 0$ **then**
- 11: Update $V_3^{(t)}$ by (26)
- 12: **end if**
- 13: **if** $i \% p_4 == 0$ **then**
- 14: Update $D_1^{(t)}$ by (27)
- 15: **end if**
- 16: **if** $i \% p_5 == 0$ **then**
- 17: Update $D_2^{(t)}$ by (28)
- 18: **end if**
- 19: **if** $i \% p_6 == 0$ **then**
- 20: Update $D_3^{(t)}$ by (29)
- 21: **end if**
- 22: **if** $\|AS^{(t-1)} - AS^{(t)}\|_F / \|AS^{(t-1)}\|_F < \epsilon$ **then**
- 23: **break**
- 24: **end if**
- 25: **end for**
- 26: **return** The spectral matrix S

3-D cube. Although the problems in steps 1 and 2 can be quickly computed using FFT or other methods, the calculation is unacceptable due to the complexity and the large number of variables involved in the optimization process. To reduce the computing time during each iteration, this article proposes a novel weighted alternating direction method of multipliers. Specifically, during the optimization, every time S is updated, not all introduced variables and auxiliary variables need to be updated while the importance of each variable for convergence of S to the optimal point is not equal. Some variables may have already converged to their optimal values and their updates may have little impact on the convergence of S . Regarding this phenomenon, some methods use partial matrix updates to reduce the computation for less important variables [49]. Here, we propose assigning a weight to each variable, which determines the frequency of its update. The complete optimization algorithm is presented in Algorithm 1.

The limitation of ADMM is that the update steps involve interdependent variables, and each step requires information from the previous step. Consequently, the steps must be executed sequentially, making parallelism difficult to achieve. Moreover, optimizing the algorithm is challenging since the convergence

of the algorithm depends on the parameters' values at each step, and changing them may affect the convergence. To mitigate this limitation, researchers have proposed various approaches, such as developing parallel ADMM algorithms, modifying the update rules, or using acceleration techniques. However, these techniques require a deep understanding of the algorithm's principles and properties and careful consideration of the problem at hand. In particular, optimizing the ADMM algorithm requires balancing the tradeoff between convergence speed and computational complexity. Unlike traditional ADMM, the proposed algorithm differs in the way variables are updated. It considers that the importance of each variable to the convergence of S is different. Some updates of variables can significantly accelerate the convergence of S , while others only have a minimal impact on convergence. Therefore, the algorithm employs an interval strategy for updating variables. If a variable is deemed essential, there will be few intervals before it gets to convergence. Conversely, if a variable is unimportant, it will have many intervals. This interval strategy reduces the computation frequency of some variables, thereby reducing the computation time and workload. Below, we will introduce the calculation of weights and setting of intervals.

Calculation of weight is an important part of this article. In the context of processing large and complex hyperspectral data, it is common practice to segment the data into smaller chunks for more efficient computation. To reduce the computation time required for calculating the weights, only a small portion of the data is selected for weight calculation. In the feature fusion, the impact of each variable on the final result is taken into consideration. After each variable update, the change in S and its impact V on the fusion accuracy are calculated, and the weights are then determined based on the average impact of these variables.

$$V_{\text{aff}} = \|AS^{V-1} - AS^V\|_F / \|AS^V\|_F, \quad (30)$$

where V_{aff} represents the affect of variables calculated in the processes of problem (14). S^{V-1} and S^V represent the value of S before and after the update of V . In other words, we figure out the every S after variables updated, then acquire the affect V_{aff} . Finally, means of V_{aff} are calculated and used as weights, which can be written as

$$c = V_{\text{sum}} / T \quad (31)$$

where T is the number of iteration and c is the weight.

Then, we introduce the setting of update interval. In the weight-based ADMM update method, the iteration frequency of variables with lower importance is reduced by setting intervals. After calculating the weights, the variable with the highest weight is selected and updated once per iteration, while the remaining variables can be updated using the following method. The update interval p for each variable is computed based on its weight. Specifically, the weight c_i of variable is first calculated, and then, the update interval p_i is set as follows:

$$p_i = c_{\text{max}} / c_i \quad (32)$$

Here, c_{max} denotes the maximum weight among all variables, and c_i denotes weights of different variables. The variable is

updated every p_i iterations, except for the variable with the highest weight, which is updated every iteration. This strategy reduces the computation frequency of variables with lower importance, while ensuring that the variable with the highest weight is updated frequently to facilitate convergence.

In this paragraph, we analyze the computational complexity. First, we calculate the computational complexity of each variable update. The complexity of the subproblem for \mathcal{S} is $\mathcal{O}(JN_H \log N_W)$, \mathbf{V}_1 is $\mathcal{O}(JN_H \log N_W + J^2N_H)$, \mathbf{V}_2 is $\mathcal{O}(J^2N_H)$, and \mathbf{V}_3 is $\mathcal{O}(J^2N_H)$. If the updates are performed sequentially, the total complexity is $\mathcal{O}(JN_H \log N_W + J^2N_H)$. Nevertheless, the proposed method maintains a complexity of $\mathcal{O}(J^2N_H)$ during iterations that do not reach the interval, and only increases to $\mathcal{O}(JN_H \log N_W + J^2N_H)$ after the interval has been reached. Since the intervals of \mathbf{V}_1 and \mathbf{V}_2 are both ten in this experiment, the proposed method efficiently solves the original problem. In terms of convergence performance, the ADMM update scheme proposed in this article bears resemblance to asynchronous ADMM. For a detailed proof of convergence, interested readers are referred to [50] or relevant literature.

In addition, there are three points that need to be clarified. First, since the update formula for auxiliary variables mainly involves matrix addition and subtraction, the time consumption of auxiliary variables update in each iteration is very low. Therefore, the update frequency of auxiliary variables can also be set to once per iteration. Second, the optimization algorithm proposed in this article has similar to asynchronous communication, and some variables may have significant deviations due to lack of updates over a long period of time. Therefore, a maximum delay “maxiter” is set in this article. Third, the setting of the update frequency does not need to be very accurate. In most cases, the algorithm will converge as the number of iterations increases. The main purpose here is to reduce the number of updates for variables that have smaller convergence effects.

IV. EXPERIMENTS

In this section, we conducted experiments on a real hyperspectral datasets to demonstrate the reliability and performance of the proposed method. The content mainly includes an introduction to the dataset, verification of the effectiveness of the proposed method on the dataset, a time comparison between serial and parallel methods, and optimization improvements in parallel computing. All experiments were conducted on a computing cluster with 128 cores, 3.40 GHz, and 600 GB RAM.

A. Datasets

The real remote sensing HSI and MSI datasets used in the experiment were obtained by the HSI Analysis Laboratory at the University of Houston and the National Center for Airborne Laser Mapping in the United States. The spectral range covers 380 to 1050 nm, with a total of 48 bands and 20 types of ground cover. In this article, an HSI image with dimensions of $601 \times 596 \times 48$ and a corresponding MSI image with dimensions of $12020 \times 11920 \times 3$ were selected as reference images to detect vehicles of different colors in the images.

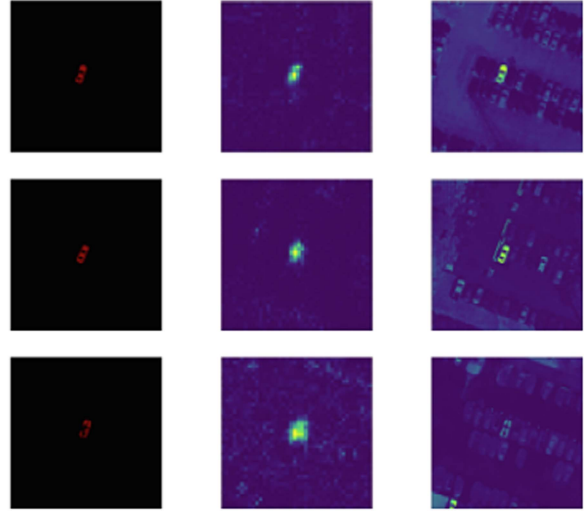


Fig. 3. Detection results of yellow cars.

B. Precision Results

The experimental results are primarily determined by three components: rough detection time, fusion time, and postfusion detection time. The size of the input data is the decisive factor in the rough detection time, while the running time of the other parts depends on the number of target areas that selected by fusion and detection. Target areas can be understood as the number of vehicles that need to be detected. Therefore, in this experiment, we selected eight fusion detection targets after the coarse detection for origin HSI. After that, we expanded the data through reasonable splicing, and linearly increased the number of fusion target detection. Thus, we have successfully validated the proposed approach presented in this article, which demonstrates its effectiveness in processing large-scale datasets.

In the experiment, we selected red and yellow cars as the targets, and red cars are more than that of yellow cars. Cars occupy fewer pixels compared with other targets, such as houses, which better reflects the accuracy of the proposed method in this study. For the hyperspectral target detection, we use the receiver operating characteristic (ROC) curve to evaluate the detection performance. The ROC curve describes the detection performance in terms of the true positive rate (P_D) and false positive rate (P_F) at each threshold θ . The curve illustrates the relationship between the detection rate and the false alarm rate. The area under the ROC curve (AUC) is used as a metric to measure the detection performance. A larger AUC value for the ROC curve indicates better detection performance. P_D and P_F are calculated as follows:

$$P_D = TP/(TP + FN), P_F = FP/(FP + TN) \quad (33)$$

where true positive (TP) represents the number of correctly detected target pixels, false positive (FP) represents the number of false alarm pixels, false negative (FN) represents the number of missed target pixels, and true negative (TN) represents the number of correctly detected background pixels.

Figs. 3 and 4 illustrate the detection results of yellow and red cars, respectively. Among them, the column on the left is the

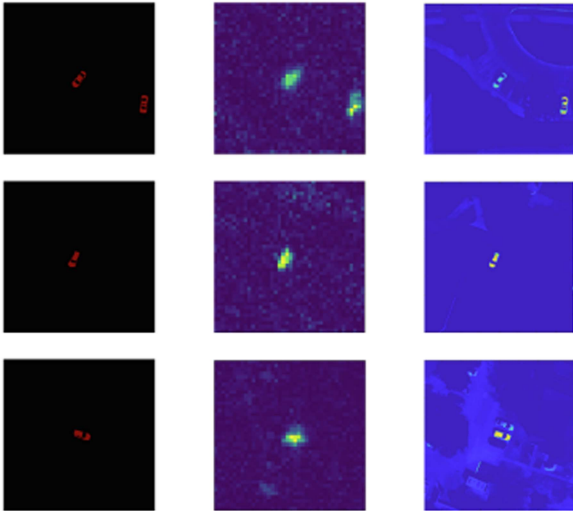


Fig. 4. Detection results of red cars.

TABLE I
AUC VALUES OF YELLOW CARS

Target	Target1	Target2	Target3
LR-HSI	0.8823	0.8301	0.9684
HR-HSI	0.9908	0.9834	0.9733

TABLE II
AUC VALUES OF RED CARS

Target	Target1	Target2	Target3
LR-HSI	0.8013	0.8411	0.9446
HR-HSI	0.9137	0.9724	0.9958

ground truth image, the middle column is the detection result before fusion, and the column on the right is the detection result after fusion. It is manifest that prefusion detection results are hazy and laborious to identify, posing a significant challenge in ascertaining the precision of target detection. Conversely, postfusion detection results are sharp and exact, facilitating easy identification of the test objects. When juxtaposed with the ground truth, the fused results demonstrate heightened accuracy. Tables I and II present the accuracy metrics, with LR-HIS signifying the nonfused detection results, and HR-HIS denoting the fused detection outcomes. The accuracy metrics suggest that the fused target detection methodology proposed herein markedly enhances the performance pertaining to accuracy and exhibits stellar performance. Elaborating further on the superiority of the proposed approach in this article, it becomes evident when compared with conventional data fusion detection methods. Traditional techniques often struggle with the tradeoff between spatial resolution and spectral resolution; however, our method effectively integrates these two aspects, yielding clearer and more accurate detection results. Moreover, our approach leverages advanced fusion algorithms, which not only preserve the spectral characteristics of the hyperspectral data but also enhance the spatial details obtained from multispectral data. This symbiotic integration substantially amplifies the detection precision and overall performance, thus establishing the notable advantage of our proposed methodology.

TABLE III
RUNNING TIME OF DIFFERENT VARIABLES UPDATE MODES

	Group 1	Group 2	Group 3	Group 4
V1	1	10	10	8
V2	1	10	10	10
V3	1	1	1	3
D1	1	5	1	2
D2	1	1	1	2
D3	1	5	1	6
Time	133s	105s	98s	176s

C. Time Results

In this section, we conducted a comparative analysis of the runtime of the proposed weight-based alternating iterative update method and the sequential update method in a single-machine environment. Four different update ways were employed for comparison, including sequential update (Group 1), weight-based update (Group 2), weight-and-time-based update (Group 3), and random update (Group 4). Table III presents the experimental results, where the four groups respectively control the variables update in the algorithm, and the data following the variables represent the update intervals in different groups. The results indicate that the proposed weight-based alternating iterative method outperforms the sequential execution in equal conditions, reducing the computation time by 25 s for single-machine execution, and this improvement is also observed in parallel execution. In addition, in Group 3, we considered the time cost of a single-step iteration and further improved the algorithm's performance by increasing the update frequency of dual variables and reducing the time needed for dual variable updates. This approach led to an even greater reduction in computation time compared with the weight-based update method used in Group 2. Group 4 shows that random variable updates do not achieve the expected acceleration effect, even the time is longer than before.

In addition to comparing the runtime performance, we also analyzed the convergence curve in the four groups. The four line charts in Fig. 5 correspond to the four different variable update methods. It can be observed from the chart that in Group 1, the ADMM method gradually converges as the number of iterations increases, and the difference between before and after iterations fluctuates and decreases. The proposed methods in Groups 2 and 3 still maintain the same decreasing trend while reducing the computational cost, indicating that the weight-based alternating iterative update method effectively achieves convergence.

Overall, the proposed weight-based alternating iterative update method exhibits excellent performance in terms of algorithmic efficiency, achieving good acceleration and fast convergence rates. This will be useful for tackling more complex optimization problems on large-scale datasets. In this section, we conducted experiments with data of varying sizes to test the performance of the proposed parallel algorithm on big data. These data were expanded by copying them to multiples of 2, 4, 6, 8, and 10, and the corresponding fusion regions and number of detection targets were increased. We tested the algorithm on different data sizes with varying numbers of computing resources (compute nodes), specifically 1, 2, 4, 8, and 16, which represent exponential

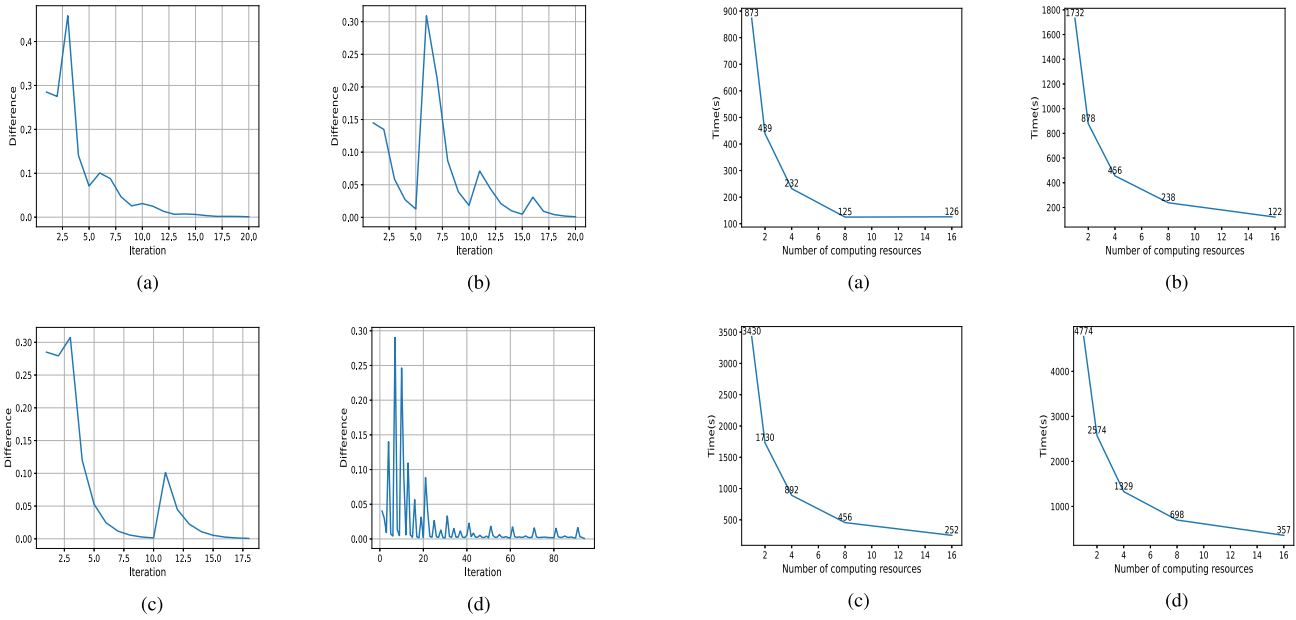


Fig. 5. Convergence curve. (a) Group1. (b) Group2. (c) Group3. (d) Group4.

growth, making it easier to observe the relationship between the increase in computing resources and the time it takes to perform calculations. As shown in the Fig. 6, the runtime of the parallel algorithm decreases almost linearly with the number of compute resources. For instance, for 4 times larger data, the runtime on a single machine is 3430 s. After adding one compute resource, the algorithm becomes parallel, and the runtime is reduced to 1730 s, saving 1700 s of time and achieving almost linear speedup. When the number of compute nodes is increased to 4, the runtime is reduced to 892 s, with a speedup ratio of 3.84. When the number of compute nodes is increased to 8, the runtime is reduced to 456 s, with a speedup ratio of 7.52. Finally, when the number of compute nodes is increased to 16, the runtime is reduced to 252 s, with a speedup ratio of 13.61. Even when the number of compute resources reaches dozens, the proposed fusion detection parallel algorithm still exhibits a high speedup ratio. Similar speedup effects were observed for data of other sizes. Considering that increasing compute resources increases data I/O between nodes, the experiment demonstrates that the proposed method can achieve excellent speedup when processing large data.

In the original data processing [see Fig. 6(a)], the speedup reached a bottleneck after eight compute resources because only eight detection targets were selected during the data fusion and detection process, resulting in only eight tasks being split. In this case, even if the number of compute resources is increased, the runtime will not decrease. In addition, we can reasonably predict the speedup situation when the data volume is sufficiently large. As long as the number of compute resources is less than the number of task partitions in the data, the method can reduce the computation time when the number of compute resources is increased. The speedup ratio will only stop increasing when the number of task partitions and the number of compute resources are the same.

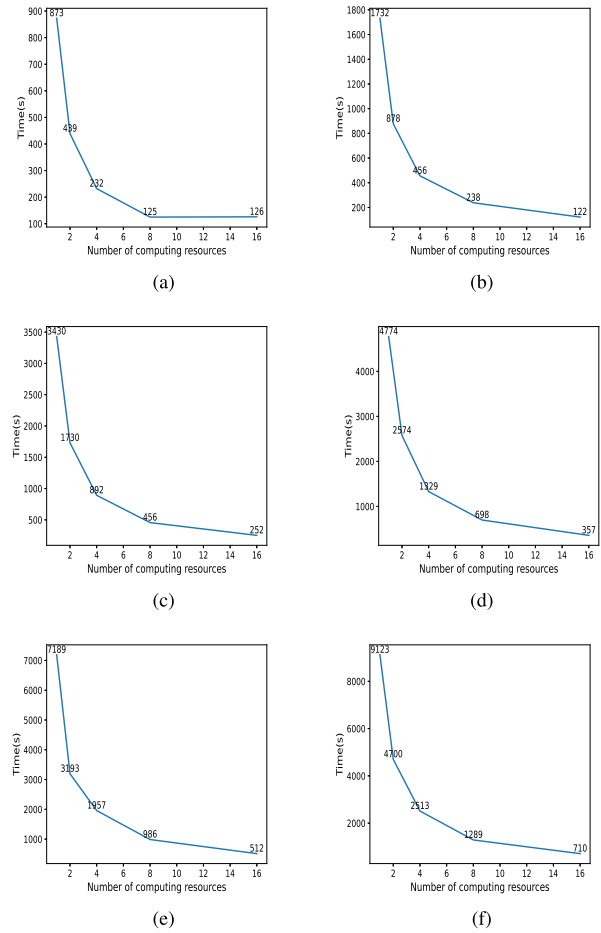


Fig. 6. Big data execution time under different computing resources. (a) Original data. (b) Twice the original data. (c) 4 times the original data. (d) 6 times the original data. (e) 8 times the original data. (f) 10 times the original data.

V. CONCLUSION

This article presents a parallel target detection method designed to tackle the challenges of poor detection performance and long processing times in the context of low-resolution hyperspectral imaging. To improve detection accuracy, CEM algorithm is employed, which utilizes background reconstruction. In addition, to overcome the low spatial resolution of hyperspectral data, multispectral and hyperspectral data are fused, leveraging the advantages of both spectral data types for clear and accurate target detection. Facing the challenge of managing large, information-rich remote sensing image data, a parallel optimization scheme is proposed for the detection method, which greatly enhances algorithmic efficiency on processing large datasets. The extensive experiments conducted demonstrate that our target detection method is both accurate and robust, particularly when handling large datasets. As such, our approach offers a novel solution to the challenges associated with target detection in the context of low-resolution hyperspectral imaging. By leveraging parallel optimization, spectral data fusion, and weight-based updates, we have demonstrated a highly effective target detection method capable of handling large, complex datasets in a practical setting.

REFERENCES

- [1] L. Zhang and L. Zhang, "Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 2, pp. 270–294, Jun. 2022.
- [2] A. Nisha and A. Anitha, "Current advances in hyperspectral remote sensing in urban planning," in *Proc. 3rd Int. Conf. Intell. Comput. Instrum. Control Technol.*, 2022, pp. 94–98.
- [3] D. Zhu, B. Du, and L. Zhang, "Target dictionary construction-based sparse representation hyperspectral target detection methods," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1254–1264, Apr. 2019.
- [4] Q. Du and C.-I. Chang, "A signal-decomposed and interference-annihilated approach to hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 4, pp. 892–906, Apr. 2004.
- [5] Y. Zhang, K. Wu, B. Du, and X. Hu, "Multitask learning-based reliability analysis for hyperspectral target detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2135–2147, Jul. 2019.
- [6] W. Sun, C. Liu, J. Li, Y. M. Lai, and W. Li, "Low-rank and sparse matrix decomposition-based anomaly detection for hyperspectral imagery," *J. Appl. Remote Sens.*, vol. 8, no. 1, 2014, Art. no. 083641.
- [7] D. Hong, L. Gao, X. Wu, J. Yao, and B. Zhang, "Revisiting graph convolutional networks with mini-batch sampling for hyperspectral image classification," in *Proc. 11th Workshop Hyperspectral Imag. Signal Process.: Evol. Remote Sens.*, 2021, pp. 1–5.
- [8] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 29–43, Jan. 2002.
- [9] X. Zhang, B. Hu, Z. Pan, and X. Zheng, "Dictionary learning based target detection for hyperspectral image," in *Proc. 5th Symp. Novel Optoelectron. Detection Technol. Appl.*, 2019, p. 49.
- [10] W. Li, Q. Du, and B. Zhang, "Combined sparse and collaborative representation for hyperspectral target detection," *Pattern Recognit.*, vol. 48, no. 12, pp. 3904–3916, Dec. 2015.
- [11] S. Matteoli, N. Acito, M. Diani, and G. Corsini, "An automatic approach to adaptive local background estimation and suppression in hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 2, pp. 790–800, Feb. 2011.
- [12] B. Du, Y. Zhang, L. Zhang, and D. Tao, "Beyond the sparsity-based target detector: A hybrid sparsity and statistics-based detector for hyperspectral images," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5345–5357, Nov. 2016.
- [13] M. A. Cho, P. Debba, R. Mathieu, L. Naidoo, J. van Aardt, and G. P. Asner, "Improving discrimination of savanna tree species through a multiple-endmember spectral angle mapper approach: Canopy-level analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4133–4142, Nov. 2010.
- [14] J. Harsanyi and C.-I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, Jul. 1994.
- [15] W. H. Farrand and J. C. Harsanyi, "Mapping the distribution of mine tailings in the Coeur d'Alene river valley, Idaho, through the use of a constrained energy minimization technique," *Remote Sens. Environ.*, vol. 59, no. 1, pp. 64–76, 1997.
- [16] F. Robey, D. Fuhrmann, E. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 1, pp. 208–216, Jan. 1992.
- [17] D. Manolakis, C. Siracusa, and G. Shaw, "Hyperspectral subpixel target detection using linear mixing model," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1392–1409, Jul. 2001.
- [18] Y. Xu, Z. Wu, F. Xiao, T. Zhan, and Z. Wei, "A target detection method based on low-rank regularized least squares model for hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 8, pp. 1129–1133, Aug. 2016.
- [19] L. Gao, X. Sun, X. Sun, L. Zhuang, Q. Du, and B. Zhang, "Hyperspectral anomaly detection based on chessboard topology," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Feb. 2023, Art. no. 5505016.
- [20] L. Zhang, L. Zhang, D. Tao, and X. Huang, "Sparse transfer manifold embedding for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 1030–1043, Feb. 2014.
- [21] L. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Hyperspectral remote sensing image subpixel target detection based on supervised metric learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4955–4965, Aug. 2014.
- [22] R. Senchuri, A. Kuras, and I. Burud, "Machine learning methods for road edge detection on fused airborne hyperspectral and LiDAR data," in *Proc. 11th Workshop Hyperspectral Imag. Signal Process.: Evol. Remote Sens.*, 2021, pp. 1–5.
- [23] D. Wang et al., "Automated vein detection for drill core analysis by fusion of hyperspectral and visible image data," in *Proc. 23rd Int. Conf. Mechatron. Mach. Vis. Pract.*, 2016, pp. 1–6.
- [24] L. Gao, D. Wang, L. Zhuang, X. Sun, M. Huang, and A. Plaza, "BS3LNET: A new blind-spot self-supervised learning network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5504218.
- [25] W. Rao, L. Gao, Y. Qu, X. Sun, B. Zhang, and J. Chanussot, "Siamese transformer network for hyperspectral image target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Mar. 2022, Art. no. 5526419.
- [26] B. Xi et al., "Multi-direction networks with attentional spectral prior for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2022, Art. no. 5500915.
- [27] W. Xie, J. Lei, J. Yang, Y. Li, Q. Du, and Z. Li, "Deep latent spectral representation learning-based hyperspectral band selection for target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 2015–2026, Mar. 2020.
- [28] X. Wang, C. Xing, Y. Feng, R. Song, and Z. Mu, "A novel hyperspectral image change detection framework based on 3D-wavelet domain active convolutional neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 4332–4335.
- [29] D. Zhu, B. Du, and L. Zhang, "Two-stream convolutional networks for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6907–6921, Aug. 2021.
- [30] H. Ren and Y.-L. Chang, "A parallel approach for initialization of high-order statistics anomaly detection in hyperspectral imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2008, pp. II-1017–II-1020.
- [31] J. Liu et al., "A distributed and parallel anomaly detection in hyperspectral images based on low-rank and sparse representation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 2861–2864.
- [32] L. Zhang, Z. Wu, J. Sun, Y. Xu, and Z. Wei, "A distributed and parallel method of hyperspectral computational imaging via collaborative Tucker3 tensor decomposition," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2022, pp. 1808–1811.
- [33] Q. Du, B. Tang, W. Xie, and W. Li, "Parallel and distributed computing for anomaly detection from hyperspectral remote sensing imagery," *Proc. IEEE*, vol. 109, no. 8, pp. 1306–1319, Aug. 2021.
- [34] R. Macias, S. Bernabé, D. Báscones, and C. González, "FPGA implementation of a hardware optimized automatic target detection and classification algorithm for hyperspectral image analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Jul. 2022, Art. no. 6011605.
- [35] J. Liu, Z. Wu, L. Xiao, J. Sun, and H. Yan, "A truncated matrix decomposition for hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 29, pp. 8028–8042, Jul. 2020.
- [36] L. Zhang, W. Wei, C. Bai, Y. Gao, and Y. Zhang, "Exploiting clustering manifold structure for hyperspectral imagery super-resolution," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 5969–5982, Dec. 2018.
- [37] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal patch tensor sparse representation for hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3034–3047, Jun. 2019.
- [38] R. Dian, S. Li, and L. Fang, "Learning a low tensor-rank representation for hyperspectral image super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2672–2683, Sep. 2019.
- [39] R. Dian and S. Li, "Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5135–5146, Oct. 2019.
- [40] J. Wright, M. Ben-Ezra, Y. Matsushita, Y.-W. Tai, R. Kawakami, and K. Ikeuchi, "High-resolution hyperspectral imaging via matrix factorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2329–2336.
- [41] J.-F. Hu, T.-Z. Huang, L.-J. Deng, T.-X. Jiang, G. Vivone, and J. Chanussot, "Hyperspectral image super-resolution via deep spatio-spectral attention convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7251–7265, Dec. 2022.
- [42] F. Palsson, J. R. Sveinsson, and M. O. Ulfarsson, "Multispectral and hyperspectral image fusion using a 3-D-convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 639–643, May 2017.
- [43] W. Dong, C. Zhou, F. Wu, J. Wu, G. Shi, and X. Li, "Model-guided deep hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 30, pp. 5754–5768, May 2021.

[44] J. Yang, Y. Zhao, J. C.-W. Chan, and C. Yi, "Hyperspectral image classification using two-channel deep convolutional neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 5079–5082.

[45] T. Huang, W. Dong, J. Wu, L. Li, X. Li, and G. Shi, "Deep hyperspectral image fusion network with iterative spatio-spectral regularization," *IEEE Trans. Comput. Imag.*, vol. 8, pp. 201–214, Feb. 2022.

[46] V. H. Pham and H.-S. Ahn, "Distributed stochastic MPC traffic signal control for urban networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8079–8096, Aug. 2023.

[47] T. Rahman, Y. Xu, and Z. Qu, "Continuous-domain real-time distributed ADMM algorithm for aggregator scheduling and voltage stability in distribution network," *IEEE Trans. Automat. Sci. Eng.*, vol. 19, no. 1, pp. 60–69, Jan. 2022.

[48] X. He, Y. Zhao, and T. Huang, "Optimizing the dynamic economic dispatch problem by the distributed consensus-based ADMM approach," *IEEE Trans. Ind. Inform.*, vol. 16, no. 5, pp. 3210–3221, May 2020.

[49] M. G. McGaffin and J. A. Fessler, "Alternating dual updates algorithm for X-ray CT reconstruction on the GPU," *IEEE Trans. Comput. Imag.*, vol. 1, no. 3, pp. 186–199, Sep. 2015.

[50] T.-H. Chang, W.-C. Liao, M. Hong, and X. Wang, "Asynchronous distributed ADMM for large-scale optimization—part II: Linear convergence analysis and numerical performance," *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 3131–3144, Jun. 2016.



Kun Yu was born in Jiangsu, China, in 1995. He received the B.Sc. degree in computer science and technology from the School of Computer Science and Technology, Huaiyin Normal University, Huaiyin, China, in 2018. He is currently working toward the Ph.D. degree in parallel computing with the Nanjing University of Science and Technology, Nanjing, China.

His research interests include hyperspectral image processing, parallel computing, and big data processing.



Shanshan Wu received the B.Sc. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2003.

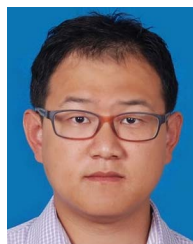
She is currently an advanced Researcher with the Nanjing Research Institute of Electronics Engineering. Her research interests include image processing and intelligent computing.



Zebin Wu (Senior Member, IEEE) received the B.Sc. and Ph.D. in computer science and technology from Nanjing University of Science and Technology, Nanjing, China, in 2003 and 2007, respectively.

He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. From August 2018 to September 2018, he was a Visiting Scholar with the GIPSA-lab, Grenoble INP, Grenoble, France, Université Grenoble Alpes, Grenoble. He was a Visiting Scholar with the Department of Mathematics, University of California Los Angeles, Los Angeles, CA, USA, from August 2016 to September 2016 and from July 2017 to August 2017. From 2014 to 2015, he was a Visiting Scholar with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, Cáceres, Spain. His research interests include hyperspectral image processing, parallel computing, big data processing, and their applications in railway foreign object detection.

His research interests include hyperspectral image processing, parallel computing, and big data processing.



Jin Sun (Member, IEEE) received the B.S. and M.S. degrees in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 2004 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Arizona, Tucson, AZ, USA, in 2011.

He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. From 2012 to 2014, he was with Orora Design Technologies, Inc., as a Member of technical staff. His research interests include cloud computing, edge computing and edge intelligence, and embedded systems.



Yi Zhang received the B.S. and Ph.D. degrees in computer science from the School of Computer Science and Engineering, Southeast University, Nanjing, China, in 2005 and 2011, respectively.

He is currently an Associate Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing. In 2009, he was an Intern with the IBM China Research Laboratory, Beijing, China. In 2011, he joined Huawei Tech. Co., Nanjing, as a Technical Research Staff Member. His research interests include project scheduling, workflow optimization, and resource management and allocation in cloud computing, mobile computing, and edge computing.



Yang Xu (Member, IEEE) received the B.Sc. degree in applied mathematics and the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2011 and 2016, respectively.

He is currently a Lecturer with the School of Computer Science and Engineering, NUST. His research interests include hyperspectral image classification, hyperspectral detection, image processing, machine learning, and their applications in railway foreign object detection.



Zhihui Wei (Member, IEEE) was born in Jiangsu, China, in 1963. He received the B.Sc. and M.Sc. degrees in applied mathematics and the Ph.D. degree in communication and information system from South East University, Nanjing, China, in 1983, 1986, and 2003, respectively.

He is currently a Professor and a Doctoral Supervisor with the Nanjing University of Science and Technology (NUST), Nanjing. His research interests include partial differential equations, mathematical image processing, multiscale analysis, sparse representation, compressive sensing, and their applications in railway foreign object detection.