Deep Reinforcement Learning for Semisupervised Hyperspectral Band Selection

Jie Feng, *Member, IEEE*, Di Li, Jing Gu, *Member, IEEE*, Xianghai Cao, *Member, IEEE*, Ronghua Shang, *Member, IEEE*, Xiangrong Zhang, *Senior Member, IEEE*, and Licheng Jiao, *Fellow, IEEE*

***Abstract*— Band selection is an important step in efficient processing of hyperspectral images (HSIs), which can be seen as the combination of powerful band search technique and effective evaluation criterion. The existing deep-learning-based methods make the network parameters sparse to search the spectral bands using threshold-based functions or regularization terms. These methods may lead to an intractable optimization problem. Furthermore, these methods need to repeatedly train deep networks for evaluating candidate band subsets. In this article, we formalize hyperspectral band selection as a reinforcement learning (RL) problem. Band search is regarded as a sequential decision-making process, where each state in the search space is a feasible band subset. To evaluate each state, a semisupervised convolutional neural network (CNN), called EvaluateNet, is constructed by adding the intraclass compactness constraint of both limited labeled and sufficient unlabeled samples. A simple stochastic band sampling method is designed to train EvaluateNet, making it possible to efficiently evaluate without any fine-tuning. In RL, new reward functions are defined by taking the EvaluateNet and the penalty of repeated selection into account. Finally, advantage actor–critic algorithms are designed to explore in the state space and select the band subset according to the expected accumulated reward. The experimental results on HSI data sets demonstrate the effectiveness and efficiency of the proposed algorithms for hyperspectral band selection.**

***Index Terms*— Actor–critic algorithm, band selection, deep reinforcement learning (DRL), hyperspectral image (HSI) classification, semisupervised learning.**

I. Introduction

THE emergence of hyperspectral remote sensing technology is one of the most iconic achievements in the history of remote sensing development. Hyperspectral sensors obtain the images with many very narrow and continuous spectral bands in the spectrum ranging from visible to thermal infrared [1]. Hyperspectral images (HSIs) realize the integration of spatial and spectral information, which makes fine ground object recognition possible. However, the excessive number of spectral bands contain some noisy and redundant ones, which have an adverse effect on the classification performance, the consumption of computing resources, transmission, and storage of HSIs. Therefore, it is necessary to reduce the dimensionality of HSIs before HSIs are processed and analyzed.

Feature selection is one of the main approaches of dimensionality reduction. It removes redundant and irrelevant features from original features without changing them. Feature selection of HSIs is also called band selection, which is appropriate for some specific applications of HSIs.

Generally, there are two key issues in feature selection methods: feature search and feature evaluation. Feature search provides candidate feature subsets by searching in the original feature space, while feature evaluation measures the scores of different feature subsets. There are three main types of feature search methods: exhaustive search, sequential search, and random search [2]. Exhaustive search tries all the possible feature combinations. The search space is the power set of all the features, which is computationally expensive as in [3]. Sequential search sequentially adds features to or removes features from the current feature subset until a desired number of features are selected, such as sequential forward selection (SFS). Cao *et al.* [4] proposed a band selection method by formalizing band selection as a dynamic classifier selection (DCS) problem, where the band subset corresponding to the best classifiers is selected using SFS. The random-searching-based methods randomly select possible band subsets to optimize the evaluation criterion under well-designed strategies. Representative methods are particle swarm optimization [5], genetic algorithm [6], clonal selection algorithm [7], and so on.

Many methods have been developed to evaluate the feature subsets. According to whether the label information of the samples is used in the process of feature evaluation, feature selection methods can be divided into supervised, unsupervised, and semisupervised. Supervised band selection methods aim to search the most discriminative band subset under the guidance of the class labels. However, the cost of acquiring the class labels for HSIs is expensive, so that the number of labeled samples is often very limited in practical applications.

Manuscript received August 3, 2020; revised November 30, 2020; accepted December 27, 2020. Date of publication February 19, 2021; date of current version December 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61871306, Grant 61836009, Grant 61772400, Grant 61773304, Grant 61703328, and Grant 61601397; in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2019JM-194; in part by the Joint Fund of the Equipment Research of Ministry of Education under Grant 6141A020337; in part by the Innovation Fund of Shanghai Aerospace Science and Technology under Grant SAST2019-093; and in part by the Aeronautical Science Fund of China under Grant 2019ZC081002. *(Corresponding author: Jie Feng.)*

The authors are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China (e-mail: jiefeng0109@163.com; dili@stu.xidian.edu.cn; xuer6126@126.com; xianghaicao@hotmail.com; rhshang@mail.xidian.edu.cn; xrzhang@mail.xidian.edu.cn; lchjiao@mail.xidian.edu.cn).

Digital Object Identifier 10.1109/TGRS.2021.3049372

The intuitive idea of supervised band selection is to directly use the classification performance as the evaluation criterion of band subsets, which requires a huge computational cost. There are also supervised methods using mutual information criteria [8], pairwise separability criteria [9], [10], distance criteria [11], coefficient-based criteria [12], and so on. The unsupervised band selection methods select the band combinations using the intrinsic properties of training samples, and their aim is to retain as much information as possible in original spectral bands. Sui *et al.* [13] proposed an unsupervised method considering both the overall accuracy (OA) and the redundancy. It incorporates the predicted OA and redundancy into the objective function through an adjusted weight. Wang *et al.* [14] considered band selection of HSIs as a combinatorial optimization process, where the purpose is to reconstruct the original data with low band correlation. There are unsupervised criteria, such as high-information criteria [15]–[17], low-correlation criteria [18], [19], large-dissimilarity criteria [20], [21], and linear prediction-based criteria [22]. Unsupervised methods usually get better generalization than supervised methods, but have poorer classification performance. Semisupervised methods inherit the advantages of both supervised and unsupervised methods. In HSIs, these semisupervised methods not only consider the label information of labeled samples but also use the intrinsic structural information of a large number of unlabeled samples. In DCS, the pseudo labels of some unlabeled samples are produced by support vector machine (SVM) classification followed by edge-preserving spatial filtering [9]. Then, the performance of band selection is improved by incorporating unlabeled samples with the pseudo labels. In [23], a semisupervised method using a hypergraph model (HM) was proposed for band selection of HSIs. In HM, a hypergraph model is built on all the samples of HSIs to measure the similarity among samples. Then, a label propagation algorithm is adopted to propagate the class labels to unlabeled samples.

Recently, deep learning is a new trend in the field of HSI processing, which can automatically and hierarchically learn from HSI data. In the field of HSI classification, deep-learning-based methods have demonstrated promising results [24], [25]. To improve the performance of deep learning for HSI classification, many deep-learning-based methods mainly focus on dealing with the small sample size problem [26], developing joint spatial–spectral extraction [27], and designing deeper network construction [28]. In the field of hyperspectral band selection, deep learning methods have achieved the state-of-the-art results. Some deep learning methods focus on achieving band selection through sparse network parameters [29], [30]. Feng *et al.* [29] proposed an end-to-end band selection method, named ternary weight convolutional neural network (TWCNN). In TWCNN, the weights of band selection layer are constrained to-1, 0, or 1 using the threshold-based function, which makes it difficult to optimize the network because the gradient is almost zero everywhere. In TWCNN, the gradient of the network is approximated using the straight through estimator (STE) [31], but the approximation in STE lacks interpretability. Some deep learning methods implement band selection through data reconstruction. Zhao *et al.* [32] proposed a band selection method called BS-Nets. In BS-Nets, band selection of HSIs is treated as a spectral reconstruction task, which assumes that all the spectral bands can be sparsely reconstructed using some informative ones. In BS-Nets, deep neural networks are designed to extract nonlinear interdependencies between spectral bands, and l_1 regularization term is added to make the weights of the deep networks sparse. But in practice, the sparsity of the weights is usually difficult to be guaranteed by only using the l_1 regularization in BS-Nets. Other deep learning methods focus on using the deep networks as the evaluator of candidate band subsets, such as explanations from convolutional neural networks (eCNN) [32], a convolutional neural network (CNN) method based on distance density (DDCNN) [33], and self-improving CNN (SICNN) [34]. eCNN and DDCNN use a full-band trained network to evaluate the band subsets without fine-tuning, which makes the evaluation inaccurate. In SICNN, the network is fine-tuned for each band subset, which requires excessive computing cost. Thus, these methods need to make a tradeoff between the accuracy and efficiency in the evaluation process.

In the past decades, reinforcement learning (RL) has become a hot direction of machine learning. Two key characteristics of RL are the trial-and-error search and the delayed reward [35]. In RL, an agent learns under the guidance of the reward. The reward is obtained from the interaction with the environment, and the agent is becoming more and more adaptable to the environment. The band selection task of HSIs can be considered as a combinatorial optimization problem that searches the band combination in a discrete space. RL has shown huge potential to solve combinatorial optimization problems [36]. An advantage of RL does not require much engineering and heuristic design [37]. Furthermore, since RL updates the parameters through trial and error [35], it does not require the expected reward to be differentiable and can deal with the search problem in a discrete space directly. Deep reinforcement learning (DRL) combines deep neural networks with RL architectures, which enables the agent to solve a wide range of complex decision-making tasks. Recently, DRL has made a great success in robotics, video games, education, transportation, finance, and healthcare [38].

In this article, a novel method named Reinforcement Learning for Semi-Supervised Band Selection (RLSBS) is proposed. In RLSBS, the band selection process is formalized as an RL problem. The sequential search of spectral bands can be regarded as a Markov decision process (MDP). In MDP, each state is a possible band subset, and each action is to determine which band is selected to the current state or end the selection while evaluating the current state. In RL, reward functions are defined by constructing an efficient semisupervised CNN, called EvaluateNet. EvaluateNet evaluates the performance of band subsets using both limited labeled samples and abundant unlabeled samples under the intraclass compactness constraint. For unlabeled samples, the center of each class is derived from averaging all the samples with adaptive fuzzy membership. Finally, advantage actor–critic (A2C) algorithms are designed to optimize the band subset by maximizing the expected cumulative reward in MDP. To solve the problem of selecting fixed

number of bands and adaptive number of bands, respectively, two algorithms called RLSBS-F and RLSBS-A are designed accordingly. The main contributions of this article are listed as follows:

1) The band selection of HSIs is formalized as an RL problem for the first time. RLSBS avoids the intractable optimization problem caused by sparse network parameters in the existing deep-learning-based band selection methods.
2) EvaluateNet can improve the evaluation efficiency without any fine-tuning. At the same time, it ensures the evaluation performance using a simple but effective stochastic band sampling.
3) To alleviate inaccurate evaluation caused by small sample size problem, EvaluateNet makes full use of a large number of unlabeled samples by penalizing the distances between the deep features of samples and their fuzzy class centers.
4) In RLSBS-A, a *stop* operation is designed in the action space and an incentive for exploration is designed in the reward so that RL agents can automatically determine the number of selected bands without prior information.

The rest of this article is organized as follows. Section II briefly introduces DRL. Section III describes the procedure of the proposed RLSBS-F and RLSBS-A in detail. After that, Section IV shows all the experiment design and result analysis on three HSI data sets. Finally, some concluding remarks and suggestions are provided for further work in Section V.

II. Background of DRL

An RL problem can be seen, an agent learns from the environment by interacting with it and acquiring numeric rewards through taking actions. The aim of RL is learning an agent, which can take actions in different environment states to maximize the expected accumulated reward. The standard RL problem can be mathematically formulated as an MDP, which involves the state, action, state transition function, reward, and discount factor. Specifically, at each discrete time step t, the agent receives the state s_t from the environment ε, and then it will take an action a_t on the environment ε according to its policy π. The environment outputs the new state s_{t+1} and the reward r_t of the current step.

Under the policy π, the accumulated reward is formulated as

$$R_t = \sum_{t\geq 0} \gamma^t r_t | \pi \tag{1}$$

where $\gamma \in (0, 1]$ is the discount factor.

The optimal policy π^* maximizing the expectation of accumulated reward R_t is

$$\pi^* = \arg\max_\pi \mathbb{E}[R_t]. \tag{2}$$

The RL methods mainly include value-based and policy-based methods. The value-based methods do not learn a policy directly, but learn a function to judge how good a state or a state–action pair is. Specifically, the value function is the expected accumulated reward at the state s using the policy π. It is calculated as

$$V^\pi(s) = \mathbb{E}[R_t | s_0 = s]. \tag{3}$$

The Q value function is the expected accumulated reward from taking an action a in the state s by using the policy π

$$Q^\pi(s, a) = \mathbb{E}[R_t | s_0 = s, a_0 = a]. \tag{4}$$

We can always find an optimal Q^* that satisfies the following Bellman equation with the value iteration algorithm [38]:

$$Q^*(s, a) = \mathbb{E}[r + \gamma \max_{a'} Q^*(s', a') | s, a]. \tag{5}$$

For complex RL problems, the value functions can be estimated by deep neural networks instead of tables including the value for each state or state–action pair. In deep Q-learning (DQN) [39], a CNN is used to estimate Q value function

$$Q^*(s, a) \approx Q(s, a; \theta) \tag{6}$$

where θ is the weights and biases of CNN. The ith loss function in the iteratively training of DQN can be calculated as follows:

$$L_i(\theta_i) = \mathbb{E}[(y_i - Q(s, a; \theta_i))^2] \tag{7}$$

where $y_i = \mathbb{E}[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a]$.

The value-based approaches do not always work well, because the action space may be complex or even continuous. The policy-based approaches find the best policy from a collection of policies directly. A representative of the policy-based methods is REINFORCE [40]. In REINFORCE, the expected accumulate reward for a policy π_θ with the parameter θ is

$$J(\theta) = \mathbb{E}[R_t | \pi_\theta]. \tag{8}$$

The parameter of the best policy is $\theta^* = \arg\max_\theta J(\theta)$. The gradient ascent can be used directly on $J(\theta)$ to optimize these parameters

$$\nabla_\theta J(\theta) \approx \sum_{t\geq 0} R_t \nabla_\theta \log \pi_\theta(a_t | s_t). \tag{9}$$

To further reduce the variance of the estimate while keeping it unbiased, a learned function called baseline is subtracted from R_t. At the same time, a discount factor γ is added to the steps of the accumulated reward. Equation (9) is modified as

$$\nabla_\theta J(\theta) \approx \sum_{t\geq 0} \left(\sum_{t'\geq t} \gamma^{t'-t} r_{t'} - b(s_t) \right) \nabla_\theta \log \pi_\theta(a_t | s_t). \tag{10}$$

The actor–critic (ac) approaches combine the policy-based methods and value-based methods by training a policy-based actor to learn a policy and a value-based critic to learn an estimate of the value to further reduce the variance. In actor–critic approaches, the discounted reward $\sum_{t'\geq t} \gamma^{t'-t} r_{t'}$ of (10) can be approximated by the Q value function

$$\sum_{t'\geq t} \gamma^{t'-t} r_{t'} \approx Q^\pi(s_{t'}, a_{t'}). \tag{11}$$

To learn a better baseline of actor–critic approaches, a method called advanced actor–critic (A2C) [41] was proposed. In A2C, the baseline is estimated by the learnable approximate value function

$$b(s_t) = V^\pi(s_t). \tag{12}$$

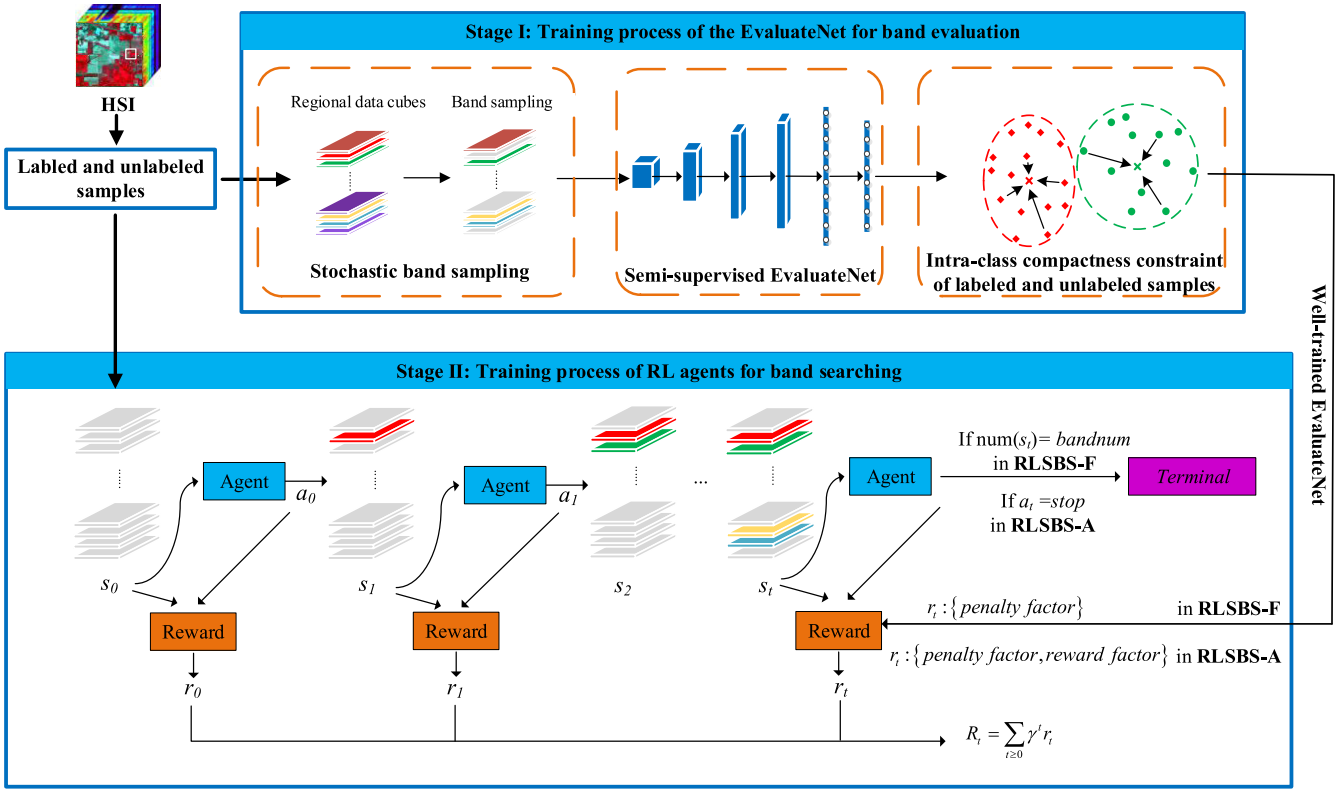


Fig. 1. Overview of the RLSBS framework. The gray bands denote the unselected spectral bands and the colored bands denote the selected ones.

$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t)$ is called the advantage function, which describes how much an action is better than expected. Asynchronous advantage actor–critic (A3C) implements asynchronous updates on the basis of A2C, which has similar performance as A2C.

III. Proposed RLSBS Methods

The band selection of HSIs is solved from the perspective of RL. RL agents are designed to search the band subset from original HSIs. In RL, the band search process is regarded as an MDP to maximize the expected accumulated reward. New reward functions are defined by constructing an efficient semisupervised CNN-based evaluation network, called EvaluateNet. At each time step, the agent selects a band and adds it to the current band subset until the agent decides to stop or the number of selected bands reaches the threshold.

The overview of the proposed RLSBS framework is shown in Fig. 1. It consists of two stages. In stage I, an EvaluateNet with high efficiency and effectiveness is constructed for band evaluation. A stochastic band sampling method is designed to train the EvaluateNet. To alleviate the small sample size problem, the intraclass compactness constraint of both labeled and unlabeled samples is designed into the loss function of the EvaluateNet. In stage II, the agents are trained to find the band subset that maximizes the expectation of the accumulated reward in RL. Here, two cases are considered. One is called RLSBS-A, which corresponds to the case of selecting adaptive number of bands. In this case, the agent can decide when to stop. The other called RLSBS-F corresponds to the case of selecting fixed number of bands. In this case, the agent stops when the number of selected bands reaches the threshold. When the agent stops according to these two cases, the reward function is defined to evaluate using the well-trained EvaluateNet. In RLSBS-F, a penalty factor is added to the reward function to suppress the repeated selection. In RLSBS-A, an additional reward factor is added to the reward function to encourage the agent to explore new bands.

A. MDP for Hyperspectral Band Selection

The first contribution of this work is formulizing hyperspectral band selection as an RL problem. We will introduce the case of RLSBS-A first, which selects adaptive number of spectral bands. The RL for band selection of HSIs is seen as an MDP, which involves the state, action, state transition function, reward, and discount factor.

Let $\mathcal{F}$ stand for the set of all the spectral bands and n represent the number of these spectral bands. f_i is any band in $\mathcal{F}$.

1) The State: In the band selection of HSIs, the state s can be any subset of the original band set $\mathcal{F}$, and the state space is the power set of $\mathcal{F}$ plus a Terminal state. Each state is represented by a band encoding. Specifically, the band encoding is a vector whose length is n. The corresponding elements of the selected bands are set to 1, and the others are set to 0.

2) The Action: In the case of RLSBS-A, two kinds of actions are included. One kind of actions is to continue to select a band from the band set $\mathcal{F}$, and the other is to stop.

In RLSBS-A, the policy of the RL agent determines the probability of which band is selected. The selection criterion is that the classification performance of the selected bands is as close as possible to that of all the original bands.

3) The State Transition Function: If any action is selected, the state will transit from one state to another, which is described as the state transition function. Specifically, if the action a_t is to select a band f_i that has already been selected in the current episode, the state s_t will not change in the next time step. If the selected band f_i has not been selected in the current episode yet, the state s_t will transit to a new one. In the new state s_{t+1}, the corresponding element of the newly selected band is set to 1. If the selected action is to stop or all the bands have been selected in RLSBS-A, the state will transit into the *Terminal* state, and the episode ends. The transition function of RLSBS-A is defined as

$$s_{t+1} = \begin{cases} \text{Terminal} & \text{if } a_t = \text{stop} \\ s_t & \text{if } a_t = \text{select a band } f_i \text{ already in } s_t \\ s_t + f_i & \text{if } a_t = \text{select a new band } f_i. \end{cases} \quad (13)$$

4) The Reward: In RL, one of the most distinguishing characteristics is to naturally handle the delayed reward [35]. In the SFS methods as [4], each selected band needs to be evaluated immediately in each step. In the RLSBS methods, the agent only needs to evaluate the performance of the final selected band subset if the next state is Terminal. In this time step, the reward function is defined to evaluate the selected band subset using the well-trained EvaluateNet. In RLSBS-A, a penalty factor $\alpha < 0$ is added to the reward function. It can avoid selecting repetitive bands that may cause the agent to fall into a loop. To further encourage the agent to explore more bands, a reward factor β is given when new bands are selected. The reward function of RLSBS-A is defined as follows:

$$r_t = \begin{cases} -(\text{loss} - \text{loss}_{\text{fullbands}}) & \text{if } a_t = \text{stop} \\ \alpha & \text{if } a_t = \text{select a band} \\ & \quad f_i \text{ already in } s_t \\ \beta & \text{if } a_t = \text{select a new band } f_i \end{cases} \quad (14)$$

where loss represents the loss of EvaluateNet, which measures how good the band subset selected in an episode is. The EvaluateNet will be introduced in detail in part B. To reduce the variance of the training of RL, the loss of EvaluateNet is subtracted by a baseline loss. The baseline loss $\text{loss}_{\text{fullbands}}$ represents the loss of all the spectral bands fed into the EvaluateNet.

In RLSBS-F, a fixed number of spectral bands are selected. In this case, the problem is much simpler. The state of RLSBS-F is the same as that of RLSBS-A. The action space of RLSBS-F deletes *stop*, and only retains to select a band f_i from the band set $\mathcal{F}$, and the corresponding state transition function is modified as

$$s_{t+1} = \begin{cases} \text{Terminal} & \text{if } \text{Num}(s_{t-1}) = \text{bandnum} \\ s_t & \text{if } a = \text{selected a band } f_i \text{already in } s_t \\ s_t + f_i & \text{if } a = \text{selected a new band } f_i \end{cases} \quad (15)$$

where $\text{Num}(\cdot)$ represents the number of nonzero elements in the current state. bandnum is the number of bands expected to be selected in RLSBS-F. Since only the selection is retained in the action, no additional rewards are given for exploration at this time. The reward function of RLSBS-F is modified as

$$r_t = \begin{cases} -(\text{loss} - \text{loss}_{fb}) & \text{if } \text{Num}(s_{t-1}) = \text{bandnum} \\ \alpha & \text{if } a = \text{selected a band} \\ & \quad f_i \text{ already in } s_t \\ 0 & \text{if } a = \text{selected a new band } f_i. \end{cases} \quad (16)$$

Actually, there are other kinds of actions for the agent to achieve band selection, such as dropping a band, adding several bands, or dropping first and then adding the bands. When different kinds of actions are designed, the reward function needs to be adjusted accordingly.

B. Semisupervised EvaluateNet for Efficient Band Evaluation

The existing deep learning methods [32]–[34] try to use the output of CNNs to evaluate the candidate band subsets in HSIs. Among these methods, SICNN [34] needs to train a specific CNN for each candidate band subset. Although training CNNs repeatedly obtains better classification performance, the computational cost is often unbearable. eCNN [32] and DDCNN [33] train the CNN with full bands and then evaluate the selected bands without any fine-tuning. These two methods reduce the computational cost, but it is difficult to fully dig out the CNN's classification ability to evaluate the candidate band subsets. In [42], an automatic network pruning method is proposed by evaluating the pruned networks directly without fine-tuning. Inspired by [42], a new semisupervised CNN called EvaluateNet is constructed to evaluate the candidate band subsets. EvaluateNet is trained using stochastic band sampling strategy. It does not need any fine-tuning, and thus it reduces the computational cost greatly while ensuring the evaluation performance.

The architecture of EvaluateNet is shown in Fig. 2. It includes a stochastic band sampling part, a spatial–spectral feature extraction part, and a classification part based on the intraclass compactness constraint. In HSIs, the training samples for EvaluateNet are represented as $X_e = \{x_1, \ldots, x_l, \ldots, x_N\}$ in an $R^{w \times w \times d}$ feature space, where each training sample $x_i (1 \le i \le N)$ is represented by a regional data cube, $w \times w$ is the size of the spatial regions, and d is the number of spectral bands. The training samples consist of l labeled training samples and $u = N - l$ unlabeled training samples. The labels of labeled training samples are denoted as $Y = \{y_1, \ldots, y_l\}$. $y_i \in \{1, 2, \ldots, K\}$, where K is the number of classes.

In the training phase of EvaluateNet, binary encoding vectors of 0 and 1 are randomly generated, and its encoding length is equal to the number of all the spectral bands in HSIs. Then, the stochastic band sampling is used to set the corresponding bands of training samples to zero according to binary encoding vectors. By training the EvaluateNet with stochastic band sampling, EvaluateNet can approximately seek the discriminative features for different band combinations. In the evaluation phase, instead of retraining or fine-tuning the EvaluateNet, we retain the selected bands and set the values of unselected bands to zero, and then feed the data into

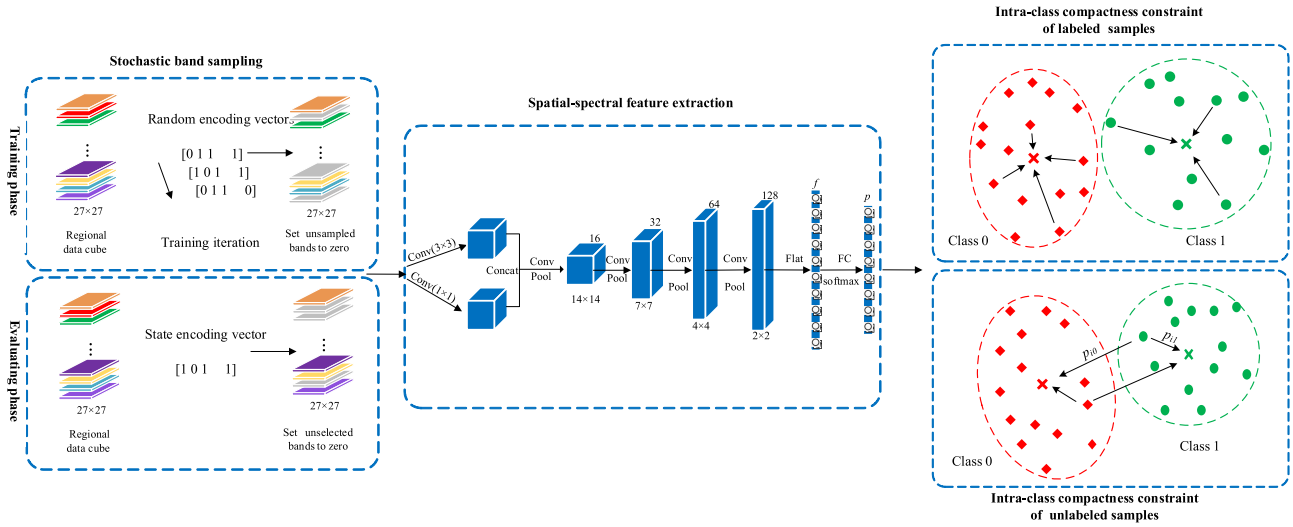


Fig. 2. Architecture of EvaluateNet. It consists of the stochastic band sampling part, spatial–spectral feature extraction part, and classification part based on the intraclass compactness constraint.

EvaluateNet and get the evaluation results directly. Without fine-tuning of EvaluateNet, the performance of candidate band subsets can be evaluated efficiently.

The network configuration for spatial–spectral feature extraction part of the EvaluateNet is a simple CNN, which is stacked by the convolutional layers, pooling layers, and fully connected layer. In multiscale convolutional layer, the convolutional layer with 1×1 kernels is designed to extract spectral features, and the convolution with 3×3 kernels is used to extract spatial features. After that, the spatial and spectral features are cascaded and fed into the next layer. To speed up the training, a batch normalization layer is added after each convolutional layer in the EvaluateNet.

In HSIs, the number of labeled training samples is extremely limited, which may limit the ability of the evaluator to extract discriminative features and result in a certain deviation in the evaluation of candidate band subsets. In addition, it is impossible to train all the band combinations for HSIs in EvaluateNet. In this case, it is required that the deep features extracted by EvaluateNet are not only separable but also discriminative. To alleviate these problems, a new intraclass constraint is defined and added to the loss function of the EvaluateNet by leveraging a large number of unlabeled samples. It prompts the EvaluateNet to narrow the deep features of both the labeled and unlabeled samples in the same class and their corresponding class centers. Thus, EvaluateNet can learn more discriminative features under limited labeled samples.

For labeled samples, it is easy to find the class center. But for the unlabeled samples of unknown classes, how to find the class center is the primary problem to be solved in the intraclass constraint. In the fuzzy c-means method [43], the fuzzy membership function is used to measure the membership degree of unlabeled samples belonging to any cluster center. Inspired by the fuzzy c-means, an adaptive fuzzy membership is designed for unlabeled samples. EvaluateNet updates the adaptive fuzzy membership, learns the class centers, and reduces the intraclass feature variation simultaneously.

In the last fully connected layer of EvaluateNet, softmax is used as the activation function. The output of the softmax function represents the probability distribution of samples belonging to all the different classes, where each element is in the range of [0, 1], and all the elements add up to 1. Here, this output is designed as the fuzzy membership of unlabeled samples, which can be learned from the network adaptively. Using the adaptive fuzzy membership, the intraclass compactness constraint for unlabeled training samples can be defined as follows:

$$L_u = \frac{1}{u} \sum_{j=1}^{K} \sum_{i=l+1}^{N} p_{ij} \| f_i - c_j \|^2 \tag{17}$$

$$c_j = \frac{\sum_{i=l+1}^{N} p_{ij} f_i}{\sum_{i=l+1}^{N} p_{ij}} \tag{18}$$

where f_i is the feature of unlabeled sample x_i extracted from the last convolutional layer in the spatial–spectral feature extraction part, as shown in Fig. 2. c_j represents the fuzzy feature center of the jth class. p_{ij} denotes the fuzzy membership, which is the probability that the feature f_i belongs to the known class c_j.

For labeled training samples, the intraclass compactness constraint is defined as

$$L_l = \frac{1}{l} \sum_{i=1}^{l} \| f_i - c_j \| \tag{19}$$

$$c_j = \frac{\sum_{y_i=j} f_i}{N_j} \tag{20}$$

where f_i denotes the corresponding feature for the labeled sample x_i. c_j represents the feature center of the jth class to which f_i belongs. N_j is the number of the samples of the jth class.

The cross-entropy loss is given for the classification of labeled training samples as follows:

$$L_c = -\frac{1}{l} \sum_{i=1}^{l} \sum_{j=1}^{K} y_i \log(p_{ij}). \tag{21}$$

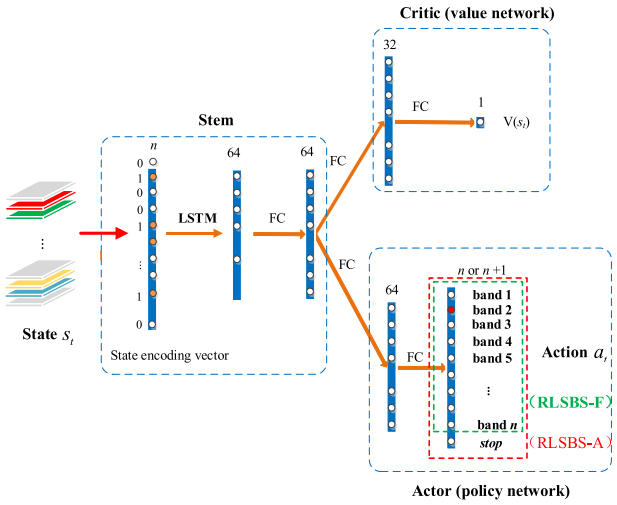


Fig. 3. Architecture of the A2C-based RL agents. The RL agent consists of a critic and an actor. In the case of RLSBS-A, actions include selecting bands and stop in the red dotted frame. In the case of RLSBS-F, actions only include selecting bands in the green dotted frame.

Both the cross-entropy loss and the intraclass constraint for both labeled and unlabeled training samples are used to jointly train the EvaluateNet. The multitask learning promotes the discriminative ability and generalization performance of EvaluateNet effectively. The final loss of the EvaluateNet is formulated as follows:

$$\text{loss} = L_c + L_l + L_u. \tag{22}$$

C. A2C-Based RL Agents for Band Searching

After the MDP process is defined and the semisupervised EvaluateNet for band evaluation is trained, A2C-based RL agents are designed to solve the band searching problem. A2C is a widely used RL method replacing the discounted cumulative award from REINFORCE [40] with the advantage function, which makes the update of the policy more stable. In this work, the agents of RLSBS-A and RLSBS-F consist of a stem network, an actor (policy network), and a critic (value network), as shown in Fig. 3. The stem network takes band encoding vectors instead of the original hyperspectral data as the input to reduce the complexity of band selection problem. A long short-term memory (LSTM) [44] layer is designed in the stem network to extract the long-term spectral dependence between bands in the state, which helps the agent avoid selecting bands with strong correlation.

In the case of RLSBS-F, the actor takes the features extracted from the stem network as the input and learns the policy. The policy predicts the selected probabilities of each band in the current step, shown in green dotted frame of Fig. 3. The critic outputs the value that indicates how good a state is in the training process. It reduces the variance from the unstable rewards of the RL agents. The architectures of the actor and critic are simple 1-D fully connected networks.

In the case of RLSBS-A, the policy learned by the actor predicts the selected probabilities of each band and a *stop* action, shown in red dotted frame of Fig. 3. When the *stop* action is selected, the state will transit to the *Terminal* state and the current episode ends.

TABLE I
PROCEDURE OF RLSBS-A AND RLSBS-F ALGORITHMS

INPUT: training set $X_{train} = \{X_e, X_a\}$, test set X_{test}

OUTPUT: The selected spectral bands and the labels of test samples

Begin
//Stage I
Training EvaluateNet with training set for EvaluateNet X_e
  Initialize all the *parameters* φ of EvaluateNet
  for every epoch
    // Stochastic band sampling
    Generate a stochastic binary encoding vector
    Set the corresponding bands of $X_e = \{X_{e\_l}, X_{e\_u}\}$ to zero according to binary encoding vector
    // Spatial-spectral feature extraction
    Forward propagation by convolutional layers
    Compute the loss by the equations (17)-(22)
    Compute the gradient $\nabla_\varphi loss$
    Gradient descent on φ using $\nabla_\varphi loss$
  end for

//Stage II
Training RL agents with training set for RL agent X_a
  Initialize all the *parameters* θ of the actor, θ_v of the critic, and step counter t=1
  for every episode:
    Reset gradients: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
    $t_{\text{start}} = t$
    Get state s_t
    while s_t is not *Terminal and* $t - t_{\text{start}} < t_{\max}$
      Perform a_t according to the policy $\pi_\theta(a_t \mid s_t)$
      Receive the reward r_t and new state s_{t+1}
    end while
    $R = \begin{cases} 0 & \text{for } \textit{Terminal } s_t \\ V(s_t, \theta_v) & \text{for non-}\textit{Terminal } s_t \end{cases}$
    for $i \in \{t-1, \ldots, t_{\text{start}}\}$ **do**
      $R \leftarrow r_i + \gamma R$
      Accumulate gradients of θ:
      $d\theta \leftarrow d\theta + \nabla_\theta \log \pi(a_i \mid s_i; \theta)\left(R - V(s_i; \theta_v)\right)$
      Accumulate gradients of θ_v:
      $d\theta_v \leftarrow d\theta_v + \partial\left(R - V(s_i; \theta_v)\right)^2 / \partial\theta_v$
    end for
    Update θ using $d\theta$ and update θ_v using $d\theta_v$
  end for

Inference the RL agent by the test samples X_{test}
Output the selected bands and the labels of test samples
End

The procedure of RLSBS-A and RLSBS-F is summarized in Table I. In HSIs, the training samples X_{train} are divided into two sets: the training set for EvaluateNet X_e and the training

TABLE II
CLASSES OF THE INDIAN PINES SCENE AND THEIR RESPECTIVE SAMPLE NUMBERS

#	Class	Samples
1	Alfalfa	46
2	Corn-notill	1428
3	Corn-mintill	830
4	Corn	237
5	Grass-pasture	483
6	Grass-trees	730
7	Grass-pasture-mowed	28
8	Hay-windrowed	478
9	Oats	20
10	Soybean-notill	972
11	Soybean-mintill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings-Grass-Trees-Drives	386
16	Stone-Steel-Towers	93
Total		10249

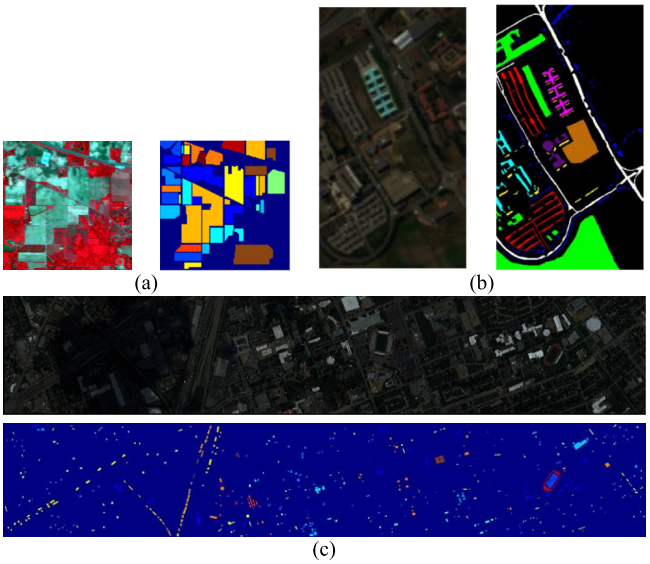


Fig. 4. False-color composite images and ground-truth classes of three data sets on the (a) Indian Pines, (b) Pavia University, and (c) University of Houston.

set for RL agents X_a. As is shown in Table I, the training set for EvaluateNet X_e is first used to train the EvaluateNet in stage I. In stage II, the training set for RL agents X_a is used to train the RL agents to search for the band subset. Once the training process of the RL agent is finished, we can get the final selected bands and the corresponding classification results by inferencing the RL agent.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this part, the performance of the proposed algorithms will be investigated on three widely used HSI data sets. We will first introduce these three data sets in part A, and then explain the experiment settings in part B. Parts C, D, and E will compare the proposed algorithms with the existing methods in terms of classification performance, sensitivity to the number of selected bands, and time consumption. The selected bands will be analyzed in part F. Part G provides the convergence analysis of the proposed RLSBS-A and RLSBS-F methods. In parts H and I, the computational complexity and the effectiveness of EvaluateNet will be analyzed, respectively. Part J investigates the performance of the comparison algorithms with disjoint train–test sets.

A. Data Description

Three HSI data sets are used to investigate the performance of the proposed RLSBS-A and RLSBS-F methods:

1) *Indian Pines:* This data set was gathered in northwestern Indiana by Airborne Visible Infrared Imaging Spectrometer, which consists of 145 × 145 pixels. Excluding the bands covering the water absorption: [104-108], [150-163], 220, there are 200 spectral bands. A total of 16 classes of the Indian Pines scene and their respective sample numbers are shown in Table II. Fig. 4(a) shows the false color image composited by bands 50, 27, and 17 and the ground truth. For this data set, 5% of the labeled data are randomly selected as the labeled training set, 10% of the labeled data are randomly selected as the unlabeled training set, and the rest is used as the test set.

TABLE III
CLASSES OF THE UNIVERSITY OF PAVIA SCENE AND THEIR RESPECTIVE SAMPLE NUMBERS

#	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947
Total		42776

2) *Pavia University:* This data set was captured by the Reflective Optical System Imaging Spectrometer over the Pavia University in northern Italy. There are 610 × 610 pixels and 103 spectral bands in this data set. As is shown in Table III, there are nine classes in the Pavia University scene. The false color image composited by bands 50, 27, and 17 and the ground truth of nine classes are shown in Fig. 4(b). For this data set, 3% of the labeled data are randomly selected as the labeled training set, 6% of the labeled data are randomly selected as the unlabeled training set, and the rest is used as the test set.
3) *University of Houston:* This data set was released in IEEE Geoscience and Remote Sensing Society (GRSS) data fusion contest in 2013, covering the University of Houston campus. The data set includes 15 classes. In all, 144 spectral bands with a spatial size of 349 × 1905 and a spatial resolution of 2.5 m per pixel are covered in the data set. The classes for the University of Houston

TABLE IV
CLASSES OF THE UNIVERSITY OF HOUSTON SCENE AND THEIR RESPECTIVE SAMPLE NUMBERS

#	Class	Samples
1	grass_healthy	1251
2	grass_stressed	1254
3	grass_synthetic	697
4	tree	1244
5	soil	1242
6	water	325
7	residential	1268
8	commercial	1244
9	road	1252
10	highway	1227
11	railway	1235
12	parking_lot1	1233
13	parking_lot2	469
14	tennis_court	428
15	running_track	660
Total		15029

scene and their sample numbers are shown in Table IV. Fig. 4(c) shows the false color image composited by bands 28, 45, and 65 and the ground truth. For this data set, 5% of the labeled data are randomly selected as the labeled training set, 10% of the labeled data are randomly selected as the unlabeled training set, and the rest is used as the test set.

B. Experimental Settings

To verify the performance of the proposed algorithms, seven band selection algorithms are used for comparison, which include four supervised methods: minimum-redundancy maximum relevance (mRMR) [8], DDCNN [33], SICNN [34], and TWCNN [29], one unsupervised method BS-Nets [45], and two semisupervised methods: HM [23] and DCS [4]. In addition, SVM with radial basis kernel function (SVM-RBF) is used to compare the full-band performance. The experiments are implemented in a computer equipped with Intel i7-7820X CPU, 64G DDR4 memory, and Nvidia RTX 2080Ti GPU. The proposed algorithms use Pytorch framework based on python language.

For SVM-RBF, the parameters are determined by grid search, and one-against-all strategy is used to extend it to the multiclassification case. For mRMR, SVM is chosen as the classifier. For SICNN, the parameters in fractional order Darwinian particle swarm optimization are suggested in [34], and the size of input regional data is set to 27×27. For DDCNN, all the parameters are set to the default values in the literature [33]. For HM, the parameters are obtained using the cross-validation on a subset of the training samples. For DCS, the number of the neighbors in K-nearest neighbor is set to 10. For BS-Nets, the regularization coefficient is set to 0.02, and the maximum number of epochs is set to 500. For TWCNN, the weight of the constraint term is set the same as that in [29]. For RLSBS-A and RLSBS-F, half of the labeled and unlabeled training samples $X_e = \{X_{e\_l}, X_{e\_u}\}$ are used as the training set for EvaluateNet, and the other half $X_a = \{X_{a\_l}, X_{a\_u}\}$ are used as the training set for the RL agent. The hyperparameters of RLSBS-A and RLSBS-F are determined by a trial-and-error procedure. The size of the input regional data cubes is 27×27. The learning rate in the training process of the EvaluateNet and RL agents is set to 0.0005 and 0.001, respectively. The number of iterations for stochastic band sampling is 8000. The parameter β of RLSBS-A is set to 1e-6, and the parameter α of RLSBS-A and RLSBS-F will be investigated in part F.

C. Analysis of the Classification Results

In this part, the OA, average accuracy (AA), and Kappa coefficient (Kappa) are used to evaluate the performance of different algorithms.

1) Indian Pines: For this data set, RLSBS-A adaptively determines the number of selected bands, and other algorithms select fixed 60 bands. Table V records the average classification results and the corresponding standard deviations of each algorithm by running 30 times independently. Among these results, the best ones are emphasized in gray regions.

In Table V, among the ten algorithms, BS-Nets and mRMR perform worse. BS-Nets uses deep neural networks to retain the most important bands by reconstructing the Indian Pines data, but it does not consider the label information in the band selection. In mRMR, mutual information-based band selection and SVM-based classification are independent of each other. DCS is superior to HM by transforming the problem of band selection into DCS. Compared with DCS and HM, deep-learning-based band selection methods, SICNN, DDCNN, TWCNN, RLSBS-F, and RLSBS-A, achieve better classification results due to the advantage of deep learning in nonlinear feature representation. SICNN surpasses DDCNN because it is difficult for DDCNN to use the CNN trained with full-band Indian Pines data to effectively evaluate different candidate band subsets, while SICNN is trained repeatedly to evaluate each band subset. TWCNN outperforms SICNN by integrating band selection, feature extraction, and classification into one network. Among five deep learning methods, RLSBS-F achieves at least 4.8% improvement in terms of OA index because of the powerful search capability of RL agents and the accurate evaluation ability of semisupervised EvaluateNet. On the basis of RLSBS-F, RLSBS-A further improves by 0.7% in terms of OA, 2.1% in terms of AA, and 0.8% in terms of Kappa by allowing the agents to automatically determine the number of selected bands. Furthermore, due to the effective usage of unlabeled samples, RLSBS-F and RLSBS-A perform better for the classes with a small number of samples, such as the Alfalfa and Oats classes.

Fig. 5 shows the visual classification results of all the ten algorithms. As shown in Fig. 5(a), (b), and (d)–(g), there are a/lot of noisy scattered misclassification points in RBF-SVM, mRMR, DDCNN, HM, DCS, and BS-Nets because of the input form of single sample. Compared with these methods, SICNN, TWCNN, RLSBS-F, and RLSBS-A use the regional cube as input and obtain smoother results. SICNN has more misclassifications in the classes with fewer samples, especially Alfalfa and Oats. Although TWCNN achieves good region homogeneity for most classes, it

TABLE V
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, SICNN, DDCNN, HM, DCS, BS-NETS, TWCNN, RLSBS-F, AND RLSBS-A ON THE INDIAN PINES DATA SET

Class	RBF-SVM	mRMR	SICNN	DDCNN	HM	DCS	BS-Nets	TWCNN	RLSBS-F	RLSBS-A
1	55.8±14.1	35.9±23.0	54±19.7	29.5±9.0	48.0±24.4	46.2±15.4	16.2±11.9	68.2±21.4	94.4±6.4	91.3±7.7
2	74.6±1.7	66.7±5.5	73.9±5.4	77.4±0.4	66.1±3.1	68.5±3.9	67.5±2.3	89.1±0.5	97.7±2.2	97.8±2.2
3	63.4±4.4	50.2±5.4	74.5±14.4	61.3±5.0	51.1±4.3	53.3±1.6	50.9±4.4	92.9±1.7	96.2±3.1	97.3±2.7
4	42.1±5.1	50.4±8.3	76.9±10	42.0±6.7	44.8±11.7	53.7±2.5	42.6±3.0	90.9±2.5	95.5±7.6	95.4±4.9
5	88.7±2.7	83.3±4.1	80.8±13.1	83.2±1.9	80.8±5.0	84.3±2.6	80.8±5.3	86.1±5.9	97.9±1.5	94.7±2.7
6	95.6±2.2	89.6±2.3	91.8±6.7	89.6±1.5	90.3±2.8	90.8±4.5	93.0±1.8	95.5±3.0	98.0±2.4	98.7±1.6
7	70.8±11.1	40.8±32.0	79.6±15.6	89.6±2.1	75.8±8.1	45.8±33.3	30.6±22.1	93.8±8.7	88.3±11.6	93.3±11.4
8	97.4±1.0	94.9±3.5	99.8±0.2	94.5±0.6	92.4±4.8	94.5±3.8	95.6±1.3	100.0±0.0	100.0±0.0	100.0±0.0
9	24.2±13.7	11.8±11.8	47.4±14.8	38.2±8.8	17.6±7.4	26.5±8.8	11.8±0.0	87.7±2.5	78.8±29.9	100.0±0.0
10	72.1±3.3	68.5±3.9	83.4±4.8	72.3±3.3	74.6±2.5	64.0±3.8	60.0±6.6	89.5±2.7	91.7±5.7	95.0±3.5
11	79.1±2.3	75.2±0.8	90.3±5.4	83.5±0.3	73.1±2.6	74.9±0.9	67.6±1.3	96.4±2.8	98.5±1.4	99.0±1.2
12	60.9±3.8	46.6±4.3	71.6±10.7	62.8±0.7	46.7±5.6	58.3±2.2	35.9±4.4	76.3±4.0	95.1±3.8	97.6±2.8
13	97.2±1.5	82.3±6.7	78.6±11.7	95.1±0.3	91.2±4.1	92.5±2.3	91.2±4.7	98.6±0.9	100.0±0.0	100.0±0.0
14	91.4±2.7	90.5±2.7	95.2±4.1	94.8±2.1	88.1±4.5	84.7±2.9	89.8±4.1	97.2±1.4	98.5±2.0	99.1±1.3
15	43.3±6.3	34.5±2.8	70.1±8.7	40.5±0.3	29.3±7.9	58.7±6.2	40.3±3.0	87.5±6.3	99.5±1.0	99.6±0.6
16	89.2±5.0	88.1±3.3	90.7±11.1	86.1±7.6	87.2±5.0	94.3±0.6	84.0±3.3	91.4±11.4	86.3±12.9	91.6±4.7
OA（%）	77.5±0.4	71.7±0.6	84.1±2.9	78.3±0.3	71.3±1.2	73.1±0.1	68.7±0.9	92.4±1.0	97.2±0.8	97.9±0.4
AA(%)	71.6±1.6	63.1±3.4	78.7±3.3	71.3±0.2	66.1±1.2	68.2±0.5	59.9±1.8	90.1±2.4	94.8±3.0	96.9±0.9
Kappa×100	74.3±0.5	67.6±0.7	81.9±3.3	75.1±0.3	67.2±1.4	69.2±0.0	64.3±1.0	91.3±1.1	96.8±0.9	97.6±0.5

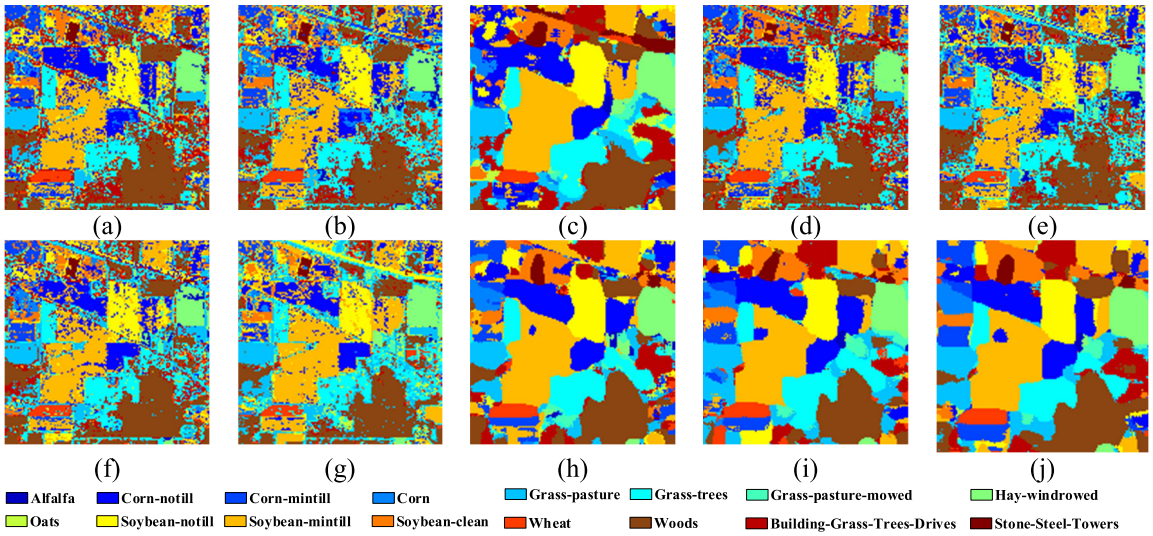


Fig. 5. Visual classification map of the Indian Pines data set. (a) RBF-SVM. (b) mRMR. (c) SICNN. (d) DDCNN. (e) HM. (f) DCS. (g) BS-Nets. (h) TWCNN. (i) RLSBS-F. (j) RLSBS-A.

misclassifies some classes as spatially adjacent classes, such as the corn-notill and soybean-mintill classes. RLSBS-F misclassifies some samples of the soybean-notill class as soybean-mintill. Compared with SICNN, TWCNN, and RLSBS-F, RLSBS-A provides a better distinction between the soybean-notill and soybean-mintill classes and obtains better boundary localization in the corn-notill and soybean-mintill classes.

2) Pavia University: For this data set, all the band selection algorithms except RLSBS-A select 30 spectral bands. Table VI lists the classification accuracies of all the classes and the OA, AA, and Kappa for all the ten algorithms. It can be seen that RLSBS-F and RLSBS-A have achieved the highest classification accuracies in most classes. It is difficult for other algorithms to classify the gravel and bare soil classes. In the classification for the gravel class, RLSBS-F and RLSBS-A have an increase of at least 34.3% and 33.4%. For the bare soil class, these two methods increase by at least 4.0%. RLSBS-A improves the classification performance by 3.8% in the OA index, 6.9% in the AA index, and 5.1% in the Kappa index compared with the best baseline among other comparison methods.

Fig. 6 shows the visual classification results of all the algorithms on the Pavia University data set. The bare soil and meadows classes are easily confused. As shown in Fig. 6(a)–(h), many samples belonging to the bare soil class are misclassified as the meadows class. RLSBS-F and RLSBS-A provide a better distinction for these two classes. There is more noise in the background region in the results of RBF-SVM, mRMR, DDCNN, HM, DCS, and BS-Nets. SICNN, TWCNN, RLSBS-F, and RLSBS-A improve the regional consistency of the background by fully using the spatial information. Compared with other methods, RLSBS-A provides more accurate classification for the samples in the

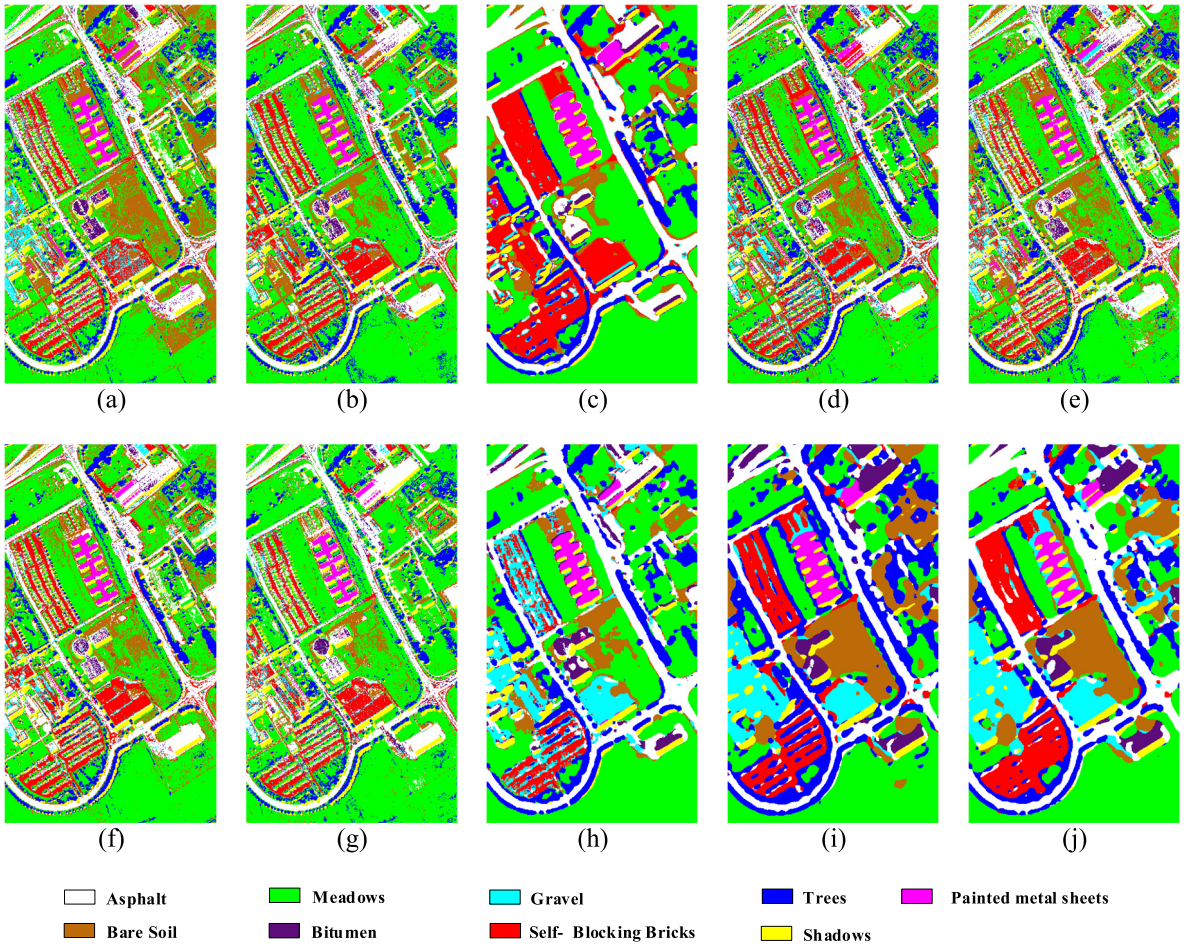


Fig. 6. Visual classification map of the Pavia University data set. (a) RBF-SVM. (b) mRMR. (c) SICNN. (d) DDCNN. (e) HM. (f) DCS. (g) BS-Nets. (h) TWCNN. (i) RLSBS-F. (j) RLSBS-A.

TABLE VI
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, SICNN, DDCNN, HM, DCS, BS-NETS, TWCNN, RLSBS-F, AND RLSBS-A ON THE PAVIA UNIVERSITY DATA SET

Class	RBF-SVM	mRMR	SICNN	DDCNN	HM	DCS	BS-Nets	TWCNN	RLSBS-F	RLSBS-A
1	89.2±2.7	85.5±1.9	87.7±1.1	88.3±1.2	88.7±0.4	84.6±3.3	84.6±5.8	96.2±0.9	100.0±0.0	99.8±0.2
2	95.4±0.5	95.1±2.1	95.7±0.1	95.6±1.7	96.7±0.1	95.1±1.3	97.2±0.6	99.6±0.2	100.0±0.0	99.9±0.0
3	62.1±5.0	31.0±8.1	63.6±0.9	43.5±14.8	33.4±13.3	49.1±3.7	27.5±8.4	65.2±8.9	99.5±1.0	98.6±0.4
4	86.2±1.8	76.8±5.4	92.0±0.1	80.9±3.6	79.1±4.2	82.4±1.3	78.4±1.8	97.6±0.6	97.8±0.7	99.4±0.4
5	98.8±0.1	99.0±0.4	99.4±0.2	99.3±0.2	99.0±0.2	99.1±0.2	99.2±0.2	100.0±0.0	100.0±0.0	99.9±0.2
6	75.4±3.0	40.3±7.9	80.1±1.1	57.3±7.4	49.3±0.7	53.2±3.9	31.5±8.9	96.0±1.3	100.0±0.0	100.0±0.0
7	66.9±8.4	55.2±6.0	67.3±5.4	59.7±8.0	46.4±11.0	69.8±5.5	45.4±12.7	84.8±3.3	99.3±1.5	98.4±1.6
8	82.1±5.0	81.8±1.7	81.2±1.9	83.1±6.8	83.9±7.8	82.1±1.5	80.4±2.7	94.4±0.3	99.6±0.5	99.5±0.2
9	99.2±0.2	99.8±0.2	99.8±0.1	99.8±0.2	99.9±0.1	99.8±0.1	99.8±0.1	98.6±1.0	93.3±4.8	99.3±0.8
OA（%）	88.0±0.4	80.5±0.5	88.9±0.1	84.4±0.5	83.0±0.2	83.7±0.5	79.8±0.9	95.9±0.8	99.6±0.1	99.7±0.1
AA(%)	83.9±0.6	73.8±0.3	85.2±0.5	78.6±1.5	75.2±1.6	79.5±0.5	71.6±2.0	92.5±1.5	98.8±0.6	99.4±0.2
Kappa×100	83.9±0.6	73.5±0.6	85.2±0.1	78.9±0.9	76.8±0.3	78.0±0.7	72.2±1.4	94.5±1.0	99.5±0.2	99.6±0.1

near-edge regions and generates more similar results to the ground truth in Fig. 4(b).

3) University of Houston: The labeled samples are more scattered throughout this data set than the previous two data sets. For this data set, all the fixed selection number of band selection algorithms select 40 spectral bands. Table VII shows the classification accuracies of all the classes and OA, AA, and Kappa for the ten algorithms. As is shown in Table VII, compared with other algorithms, the proposed RLSBS-F and RLSBS-A algorithms have a certain degree of improvement in the classification results of all the classes. In particular, in the tennis_court and running_track classes, these two algorithms obtain completely correct classification. Among all the ten algorithms, RLSBS-A achieves the best OA, AA, and Kappa indexes.

Fig. 7 shows the visual classification results of the University of Houston data set. Because the distribution of labeled samples on this data set is too scattered, the entire data set is

TABLE VII

CLASSIFICATION RESULTS OF RBF-SVM, MRMR, SICNN, DDCNN, HM, DCS, BS-NETS, TWCNN, RLSBS-F, AND RLSBS-A ON THE HOUSTON DATA SET

Class	RBF-SVM	mRMR	SICNN	DDCNN	HM	DCS	BS-NETs	TWCNN	RLSBS-F	RLSBS-A
1	93.1±1.7	95.4±1.3	97.3±1.0	94.4±2.3	88.3±4.0	97.1±0.6	91.1±7.0	93.1±3.0	97.0±0.9	99.4±0.3
2	96.7±0.4	89.5±5.0	96.5±1.7	96.4±2.6	98.6±0.1	95.9±2.1	89.7±4.5	97.8±1.3	97.6±1.3	99.0±0.8
3	97.4±0.3	87.4±4.0	99.7±0.3	98.3±1.4	93.0±0.9	98.3±1.8	88.5±0.3	99.8±0.2	100.0±0.0	99.9±0.3
4	92.6±2.1	90.9±0.3	96.3±0.1	94.6±1.5	98.3±0.0	95.8±0.3	95.8±2.9	92.5±2.3	97.2±0.6	98.8±1.3
5	97.2±0.2	98.4±0.5	98.5±0.6	97.3±1.2	98.7±0.4	98.6±0.3	96.9±0.0	99.3±0.4	99.6±0.4	100.0±0.0
6	94.2±0.4	94.6±2.9	96.9±2.7	96.0±0.7	90.6±4.7	89.7±4.6	87.5±7.5	88.1±1.2	97.1±1.8	97.4±2.2
7	94.6±0.6	73.0±3.1	86.9±3.2	89.8±0.3	81.0±0.2	89.5±2.4	78.2±1.6	93.3±1.5	93.8±0.8	91.1±1.4
8	76.2±3.8	75.0±3.6	83.2±0.6	78.4±2.0	75.8±7.6	80.3±3.9	77.8±1.5	88.5±6.1	92.4±2.0	94.6±1.2
9	89.5±1.7	75.0±1.7	82.6±1.9	80.3±1.2	79.2±3.5	83.5±1.1	77.5±4.5	90.0±3.7	96.8±1.5	96.0±0.2
10	95.4±2.8	77.8±1.6	87.5±1.7	87.7±1.0	80.0±4.0	87.9±1.8	79.9±2.7	94.3±4.2	100.0±0.0	97.4±2.4
11	96.8±0.6	76.1±1.2	86.6±1.7	80.0±1.2	74.1±3.3	85.3±1.7	77.3±0.9	97.1±1.5	96.9±0.4	100.0±0.0
12	86.4±3.6	72.1±4.2	84.3±1.5	78.6±2.6	82.2±0.0	76.4±4.6	74.8±2.7	92.5±2.7	94.5±1.9	96.3±1.5
13	73.1±8.3	33.1±6.3	40.4±2.3	44.0±0.9	46.8±9.9	39.7±8.4	27.5±2.0	93.2±5.1	93.6±0.5	99.2±1.0
14	99.0±0.1	97.0±1.4	99.3±0.1	98.6±0.0	95.6±3.3	97.2±2.7	95.9±0.3	96.3±2.6	100.0±0.0	100.0±0.0
15	99.5±0.3	94.5±2.3	98.9±0.0	99.3±0.4	98.1±0.3	98.6±0.5	96.4±1.6	96.3±4.3	100.0±0.0	100.0±0.0
OA（%）	92.1±0.7	82.2±0.2	89.7±0.2	87.9±0.1	85.7±0.3	88.6±0.3	83.3±0.5	94.1±1.3	96.9±0.3	97.7±0.3
AA(%)	92.1±1.2	82.0±0.2	89.0±0.2	87.6±0.1	85.4±0.5	87.6±0.4	82.3±0.7	94.1±1.4	97.1±0.3	98.0±0.3
Kappa×100	91.5±0.8	80.8±0.3	88.8±0.2	86.9±0.2	84.5±0.3	87.7±0.3	82.0±0.6	93.7±1.4	96.7±0.3	97.5±0.3

used to predict and fully show the visual classification map. There are more noisy points in the visual classification map of RBF-SVM, mRMR, DDCNN, HM, DCS, and BS-Nets. SICNN, TWCNN, RLSBS-F, and RLSBS-A improve the region homogeneity for most classes. RLSBS-A performs better in detail compared with SICNN, TWCNN, and RLSBS-F, especially in the highway class.

D. Sensitivity to the Number of Selected Bands

The sensitivity to different numbers of selected bands is investigated for all the fixed selection number of band selection algorithms. Fig. 8 shows the relationship between the number of selected bands and the OA value on three HSI data sets by running 30 times independently. RBF-SVM and RLSBS-A are not included because RBF-SVM uses all the spectral bands and RLSBS-A determines the number of selected bands by itself. For the Indian Pines data set, the number of selected bands is investigated in the range of [2, 200], with an interval of 20. For the Pavia University and University of Houston, the corresponding number of selected bands is in the range of [2, 100] and [2, 140] with an interval of 20, respectively.

As shown in Fig. 8, the OA value of mRMR, DDCNN, HM, DCS, and BS-Nets improves sharply as the number of selected bands changes from 2 to 60, while SICNN, TWCNN, and RLSBS-F improve slowly in terms of OA. That may be because SICNN, TWCNN, and RLSBS-F select more discriminative band subset. Moreover, these methods more fully extract spatial–spectral information when only few spectral bands are selected. The OA value of TWCNN and RLSBS-F does not improve significantly after 60 bands are selected. The OA value of SICNN has been gradually improving with the increasing number of selected bands. Compared with SICNN, TWCNN and RLSBS-F are more likely to select a compact band subset. Since TWCNN still has to handle the intractable optimization problem, even if more bands are selected, its classification performance is affected. Compared with other methods, RLSBS-F consistently shows the best classification performance in the entire range of different numbers of selected spectral bands.

TABLE VIII

RUNNING TIME OF RBF-SVM, MRMR, SICNN, DDCNN, HM, DCS, BS-NETS, TWCNN, RLSBS-F, AND RLSBS-A

Dataset	Method	Training Time(s)	Test Time(s)
Indian Pines	RBF-SVM	0.4±0.1	0.8±0.1
	mRMR	1.2±0.1	0.3±0.1
	SICNN	1,037.2±1.6	1.7±0.2
	DDCNN	26.3±3.7	1.4±0.3
	HM	3411.7±55.0	0.8±0.1
	DCS	11810.7±125.0	1.8±0.1
	BS-Nets	881.4±1.3	1.8±0.1
	TWCNN	330.9±9.1	1.4±0.0
	RLSBS-F	1219.6±7.0	1.1±0.1
	RLSBS-A	1125.8±25.3	1.1±0.1
Pavia University	RBF-SVM	0.5±0.1	1.2±0.1
	mRMR	1.9±0.1	0.8±0.1
	SICNN	1586.4.2±5.4	2.1±0.2
	DDCNN	36.2±4.1	1.5±0.1
	HM	10175.1±100.2	1.3±0.1
	DCS	16902.5±56.0	1.3±0.2
	BS-Nets	1723.6±4.75	4.9±0.4
	TWCNN	570.5±1.1	3.0±0.0
	RLSBS-F	1510.6±10.4	4.0±0.0
	RLSBS-A	1338.6±30.8	4.0±0.0
University of Houston	RBF-SVM	0.6±0.1	1.3±0.1
	mRMR	1.5±0.1	0.9±0.1
	SICNN	1206.2±3.4	2.7±0.2
	DDCNN	30.9±2.7	1.2±0.2
	HM	12798.0±64.0	1.5±0.2
	DCS	17161.4±150.5	1.3±0.2
	BS-Nets	889.6±2.0	1.5±0.2
	TWCNN	460.6±2.2	1.7±0.0
	RLSBS-F	744.9±1.9	1.6±0.0
	RLSBS-A	708.0±6.5	1.6±0.0

E. Investigation on Running Time

In this part, we investigate the running time of all the ten algorithms on three HSI data sets in Table VIII. In RLSBS-F

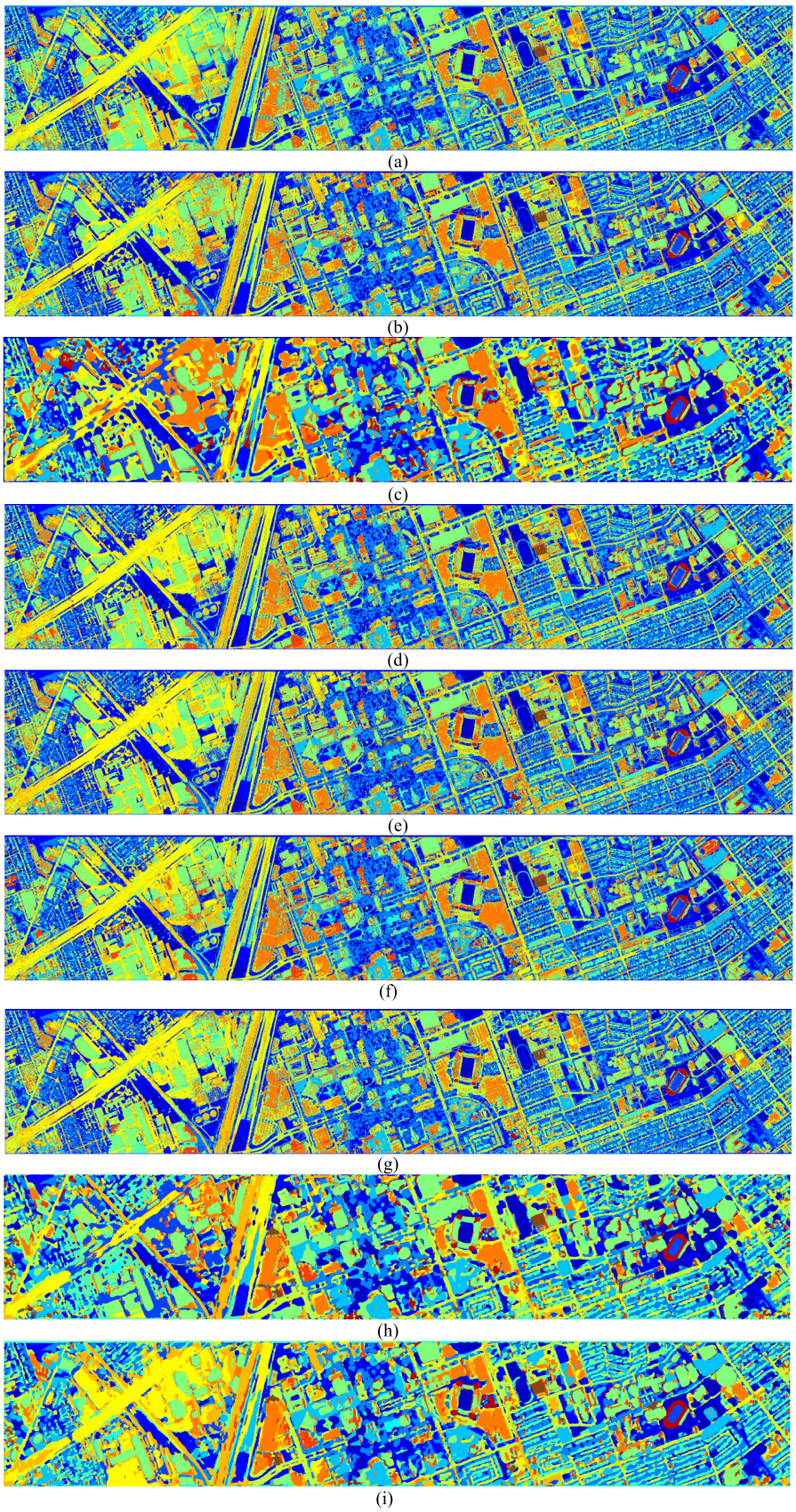


Fig. 7. Visual classification map of the University of Houston data set. (a) RBF-SVM. (b) mRMR. (c) SICNN. (d) DDCNN. (e) HM. (f) DCS. (g) BS-Nets. (h) TWCNN. (i) RLSBS-F.

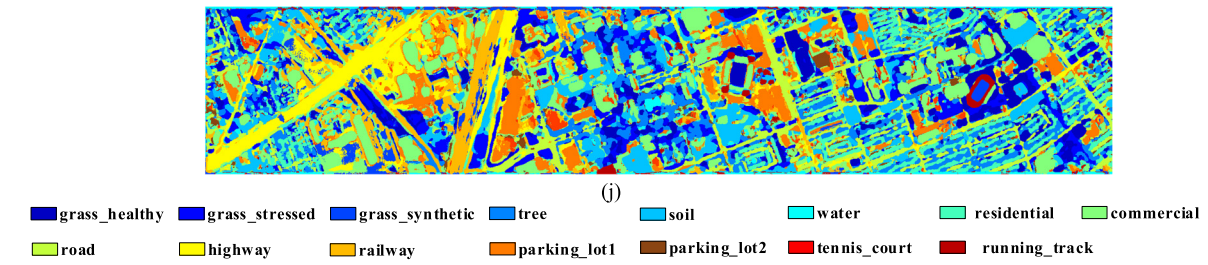


Fig. 7. *(Continued.)* Visual classification map of the University of Houston data set. (j) RLSBS-A.

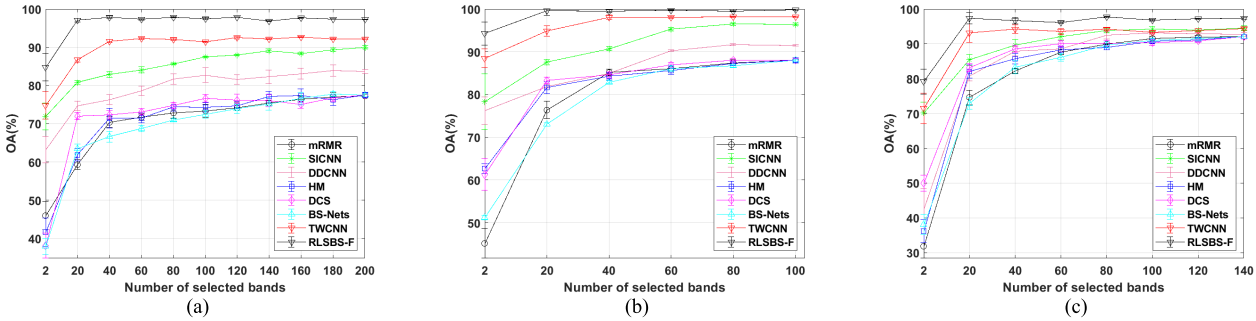


Fig. 8. Sensitivity to the number of selected bands. (a) Indian Pines. (b) Pavia University. (c) University of Houston.

and RLSBS-A, the running time includes the time cost by both EvaluateNet and *RL* agents. For the training time of Table VIII, RBF-SVM is the fastest without band selection process. mRMR takes less time than other band selection methods using efficient incremental search. Among five deep learning methods, DDCNN is the most efficient using the CNN well-trained by full-band data in advance. SICNN, BS-Nets, and TWCNN require more training time compared with DDCNN. The time of SICNN is mainly focused on repeatedly training CNNs for different candidate band subsets. BS-Nets focuses on the network optimization by minimizing the reconstruction error. TWCNN needs a lot of time to solve the intractable optimization problem. Compared with other methods, four semisupervised methods, HM, DCS, RLSBS-A, and RLSBS-F, cost more training time using a large number of unlabeled samples. HM needs a lot of computation to build a hypergraph, which results in a sharp increase in the training time with the increasing number of training samples. Thus, its training time on the Pavia University has greatly increased than that on the Indian Pines data set. DCS needs to train many base classifiers for different band combinations, which consumes excessive time. Compared with HM and DCS, RLSBS-A and RLSBS-F save the training time significantly through the design of efficient EvaluateNet, especially on the Pavia University and University of Houston data sets. RLSBS-A requires slightly more training time than RLSBS-F, because RLSBS-A makes the agents search in larger space and needs more iterations to converge. In the test time, all the algorithms can finish the test process in seconds, and the methods based on deep learning need more time on the inference of the networks.

F. Analysis of the Selected Spectral Bands

To show the selected bands more intuitively, all the fixed selection number of band selection algorithms are used to select 20 spectral bands on three HSI data sets. In Fig. 9, the upper part of each subfigure shows which bands are selected by each band selection algorithms, where the points of each line represent the locations of selected bands over the range of entire spectral bands. As shown in Fig. 9, the distribution of the bands selected by HM and BS-Nets is denser. In HSIs, the spectra are divided into hundreds of narrow and near-continuous spectral bands. Thus, there may be a high correlation between adjacent bands. The selection of these adjacent bands would result in more information redundancy. The information entropy of each band is shown in the below part of the subfigures in Fig. 9. The spectral bands with lower entropy values contain less amount of information. If a spectral band has lower entropy value among adjacent spectral bands, it is likely to be the noisy one. In SICNN, TWCNN, HM, and DDCNN, some spectral bands with low entropy values are selected. Compared with other algorithms, RLSBS-F can remove these noisy spectral bands successfully and select the spectral bands with more discrete distribution.

G. Convergence Analysis of RLSBS-F and RLSBS-A

To analyze the convergence of RLSBS-F and RLSBS-A, we train RLSBS-F and RLSBS-A iteratively for 5000 episodes on three data sets and record the reward value of RL agents with different values of parameter α. Furthermore, the loss of EvaluateNet in RLSBS-F and RLSBS-A is also recorded to analyze the discriminative ability of selected bands as the number of episodes changes. Fig. 10(a)–(f) shows the convergence curves of RLSBS-F and RLSBS-A, respectively. For RLSBS-F, 20 spectral bands are selected on all the three data sets. Note that the curves are smoothed by $x_t = 0.99 * x_{t-1} + 0.01 * x_t$, where x_t is the reward or loss of the current episode.

For both the methods, the reward and the loss of the EvaluateNet gradually tend to converge with iteration.

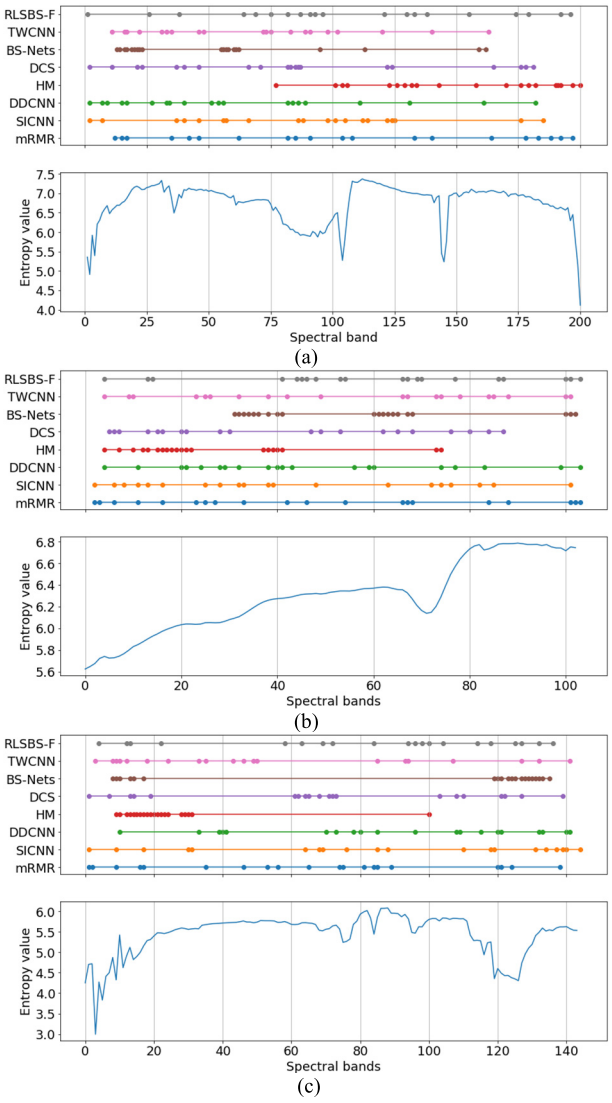


Fig. 9. Twenty spectral bands selected by different methods in the above subfigures and the entropy value of each band in the below subfigures on the (a) Indian Pines, (b) Pavia University, and (c) University of Houston.

Compared with Fig. 10(a) of Indian Pines and Fig. 10(c) of University of Houston, RLSBS-F converges faster as shown in Fig. 10(b) of Pavia University. That may be due to the fact that there are more training samples and less spectral bands in the Pavia University data set, which allows agents to find suitable band combinations more quickly. Generally, the loss of EvaluateNet accounts for a large proportion of the award. It can be seen that the reward described by the left of each subfigure changes almost synchronously with the loss described by the right subfigure. Additionally, more discriminative bands are selected when the RL agents tend to converge. As shown in Fig. 10(a)–(c), a suitable α value can speed up the convergence of RL agents and reduce the variance of the values of the reward and loss.

For RLSBS-A, it can be seen in (d)–(f) that α has a greater impact on RLSBS-F than on RLSBS-A, and the curve oscillation of RLSBS-A is also larger than RLSBS-F. The adaptive selection for the number of selected bands in RLSBS-A increases the complexity of the problem, which easily causes the RL agents to be unstable. RLSBS-A needs more iterations to train, but achieves lower loss of EvaluateNet. It shows that RLSBS-A can effectively and adaptively select the discriminative band subset.

TABLE IX

NUMBER OF LEARNABLE PARAMETERS AND FLOPS OF THE MAIN TYPES OF LEARNABLE LAYERS IN THE PROPOSED METHOD

	Learnable parameters	FLOPs
Convolution	$\left[M_{in}^{conv}S^2+1\right]M_{out}^{conv}$	$2F^2\left[M_{in}^{conv}S^2+1\right]M_{out}^{conv}$
Full Connection	$\left(M_{in}^{FC}+1\right)M_{out}^{FC}$	$2M_{in}^{FC}M_{out}^{FC}$
The proposed RL agent	$\sum_{i=1}^{l_{fa}}\left(M_{in_i}^{FC}+1\right)M_{out_i}^{FC}$	$\sum_{i=1}^{l_{fa}}2M_{in_i}^{FC}M_{out_i}^{FC}$
The proposed EvaluateNet	$\sum_{i=1}^{l_{ce}}\left\{\left[M_{in_i}^{conv}S_i^2+1\right]M_{out_i}^{conv}\right\}$ $+\sum_{i=1}^{l_{fe}}\left[\left(M_{in_i}^{FC}+1\right)M_{out_i}^{FC}\right]$	$\sum_{i=1}^{l_{ce}}2F_i^2(M_{in_i}^{conv}S_i^2+1)M_{out_i}^{conv}$ $+\sum_{i=1}^{l_{fe}}2M_{in_i}^{FC}M_{out_i}^{FC}$
The whole proposed method	141.8k	16.0M

H. Analysis of the Computational Complexity of RLSBS-F and RLSBS-A

In this part, the computational complexity is analyzed by measuring the number of learnable parameters and floating point operations (FLOPs) [46], [47]. In the proposed RLSBS-F and RLSBS-A methods, the parameters and computational cost mainly focus on two parts: RL agent and EvaluateNet. The RL agent is mainly stacked by the fully connected layers, and EvaluateNet is stacked by the convolutional layers, pooling layers, and fully connected layer. The corresponding results of the main types of learnable layers in the proposed method are recorded in Table IX. In Table IX, the last line lists the total number of learnable parameters and FLOPs of the proposed methods.

In Table IX, $S.$ represents the size of the convolution kernel. $F.$ indicates the spatial size of the output feature map. The number of input and output channels is represented as $M_{\text{in}.}$ and $M_{\text{out}.}$, respectively. In the proposed method, l_{fa} denotes the number of fully connected layers in the proposed RL agent. l_{ce} and l_{fe} represent the number of convolution layers and fully connected layers in the EvaluateNet, respectively.

In the proposed methods, the RL agent only needs a small number of parameters and FLOPs, and EvaluateNet only needs forward calculation in the band selection stage without fine-tuning, which greatly reduces the computational cost. Finally, the proposed method has only 141.8k learnable parameters and 16.0 M FLOPs.

I. Effectiveness Analysis of the EvaluateNet

In this part, the effectiveness of the EvaluateNet is analyzed by comparing with other evaluation criteria. Specifically, we use SVM [48], mRMR [8], and CNN [33] as the criteria of the proposed RL method to select the fixed number of bands and adaptive number of bands.

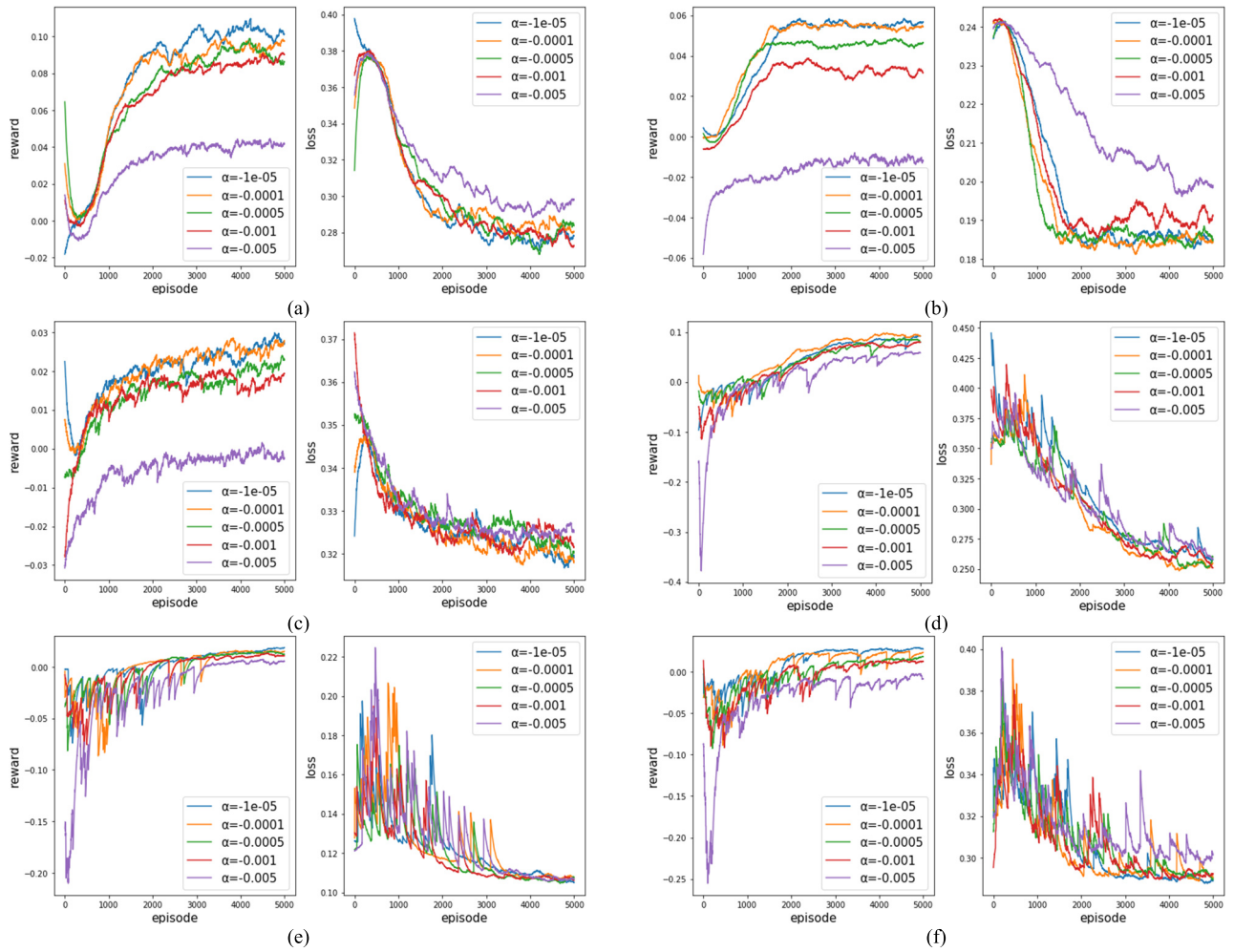


Fig. 10. Convergence curves of RLSBS-F and RLSBS-A. (a)–(c) show the convergence curves of RLSBS-F with different values of the parameter α on (a) Indian Pines, (b) Pavia University, and (c) University of Houston. (d)–(f) show the convergence curves of RLSBS-A with different values of the parameter α on (d) Indian Pines, (e) Pavia University, and (f) University of Houston.

TABLE X
OA RESULTS OF THE PROPOSED METHODS WITH OTHER EVALUATION CRITERIA

		RL+ SVM [51]	RL+ mRMR [10]	RL+ CNN [38]	RL+ EvaluateNet (ours)
Fixed number of selected bands	Indian Pines (80 bands)	76.2±0.2	74.8±0.2	90.3±0.9	97.2±0.8
	Pavia University (40 bands)	87.3±0.1	84.3±0.1	95.7±0.2	99.6±0.1
	University of Houston (60 bands)	85.7±0.3	82.1±0.1	90.7±0.3	96.9±0.3
Adaptive number of selected bands	Indian Pines	76.6±1.3	75.0±0.7	92.2±0.1	97.9±0.4
	Pavia University	87.6±0.2	84.3±0.2	96.6±0.2	99.7±0.1
	University of Houston	86.8±0.2	83.3±0.5	91.0±0.2	97.7±0.3

Table X shows the OA results of the RL methods with SVM, mRMR, and CNN criteria. As shown in Table X, compared with the RL method with mRMR, the RL method with SVM directly uses the classification accuracy of SVM as the evaluation criterion to achieve better classification performance. This is because mRMR is not directly related to the classification performance. The RL method with CNN uses the CNN well-trained by the full-band for band evaluation. By this way, this method may lead to biased performance evaluation while directly setting the unselected bands to zero in CNN. Among all the methods, the proposed RL method with EvaluateNet achieves the best classification results on all the three HSI data sets.

J. Analysis with Spatially Disjoint Training–Test Sets

Some previous works [25], [49], [50] have shown that the overlap between the training and test sets would lead to unfair or biased performance evaluation. In this part, using the controlled sampling strategy [50], spatially disjoint training and test sets are obtained in the three HSI data sets. Specifically, for

TABLE XI
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, SICNN, DDCNN, HM, DCS, BS-NETS, TWCNN, RLSBS-F, AND RLSBS-A ON THE INDIAN PINES DATA SET WITH DISJOINT TRAIN–TEST SETS

	RBF-SVM	mRMR	SICNN	DDCNN	HM	DCS	BS-NETs	TWCNN	RLSBS-F	RLSBS-A
OA（%）	77.4±1.5	71.1±0.7	82.0±2.7	75.1±1.1	72.9±1.0	73.1±2.0	70.9±2.6	83.2±2.9	86.7±1.0	88.3±0.8
AA(%)	75.2±4.1	72.3±2.6	82.2±0.4	74.4±2.1	70.4±2.1	69.1±1.0	71.5±2.9	81.9±4.2	85.5±2.0	87.7±0.8
Kappa×100	74.5±1.6	67.4±0.6	79.6±3.1	71.7±1.2	69.1±1.2	69.3±2.2	67.1±2.9	80.9±3.3	84.8±1.2	86.6±0.9

TABLE XII
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, SICNN, DDCNN, HM, DCS, BS-NETS, TWCNN, RLSBS-F, AND RLSBS-A ON THE PAVIA UNIVERSITY DATA SET WITH DISJOINT TRAIN–TEST SETS

	RBF-SVM	mRMR	SICNN	DDCNN	HM	DCS	BS-NETs	TWCNN	RLSBS-F	RLSBS-A
OA（%）	89.9±1.4	83.9±1.6	87.0±3.2	87.4±1.1	85.8±2.9	86.6±2.0	83.8±2.5	92.4±3.2	94.1±2.5	95.5±1.8
AA(%)	89.4±1.3	83.4±1.0	84.5±3.3	87.4±1.1	83.8±1.9	83.0±1.5	82.1±1.4	90.5±4.6	93.1±2.3	95.2±2.3
Kappa×100	86.6±1.9	78.9±1.9	82.7±4.4	83.3±1.4	81.2±3.7	82.2±2.6	78.6±3.0	89.9±4.2	92.2±3.3	94.0±2.4

TABLE XIII
CLASSIFICATION RESULTS OF RBF-SVM, MRMR, SICNN, DDCNN, HM, DCS, BS-NETS, TWCNN, RLSBS-F, AND RLSBS-A ON THE UNIVERSITY OF HOUSTON DATA SET WITH DISJOINT TRAIN–TEST SETS

	RBF-SVM	mRMR	SICNN	DDCNN	HM	DCS	BS-NETs	TWCNN	RLSBS-F	RLSBS-A
OA（%）	89.6±0.3	82.2±0.3	85.7±2.7	84.5±0.4	83.8±0.8	84.7±0.8	83.1±0.0	91.3±1.0	92.1±1.3	94.6±0.8
AA(%)	88.7±0.6	81.3±0.4	83.3±3.3	83.5±0.2	82.7±1.1	83.4±1.1	81.8±0.0	89.8±1.7	90.0±2.1	92.9±1.6
Kappa×100	88.7±0.3	80.7±0.3	84.6±2.9	83.2±0.4	82.4±0.9	83.5±0.9	81.7±0.0	90.6±1.1	91.5±1.4	94.1±0.9

Indian Pines and Pavia University data sets, 30% and another 30% of the labeled data are selected as the labeled training set and unlabeled training set, respectively, and the rest is used as the test set. For the University of Houston data set, 5% and another 10% of the labeled data are selected as the labeled training set and unlabeled training set, respectively, and the rest is used as the test set. Tables XI–XIII record the average classification results with disjoint train–test sets by running 30 times independently.

As shown in Tables XI–XIII, SICNN, TWCNN, RLSBS-F, and RLSBS-A perform better than mRMR, HM, DCS, and DDCNN by fully using the spatial information. It is worth noting that the improvement on the disjoint train–test sets is not as large as that on the random sampling train–test sets. This is because the original random sampling strategy makes the test samples and training samples have a higher spatial correlation. In the Indian Pines data set, the proposed RLSBS-F and RLSBS-A methods achieve at least 3.5% and 5.1% improvement in terms of the OA index. In the Pavia University and University of Houston data sets, the proposed RLSBS-F and RLSBS-A methods perform better in all the indexes.

V. CONCLUSION

In this article, a novel DRL-based framework for semisupervised hyperspectral band selection has been proposed. To solve the problem of band subset search and evaluation in band selection, two methods named RLSBS-F and RLSBS-A are designed to consider the selection of fixed and adaptive number of bands, respectively. The band subset searching problem is regarded as an MDP, which is optimized by A2C agents with LSTM. For the band evaluation, a fast semisupervise band evaluation network called EvaluateNet is designed, where band evaluation without fine-tuning is realized by stochastic band sampling. To alleviate inaccurate evaluation caused by the small sample size problem of HSIs, the intraclass constraint of unlabeled samples is defined by fuzzy division. The experimental results on three HSI data sets showed the effectiveness of the proposed algorithms. The proposed algorithms provide a new paradigm for hyperspectral band selection. In addition to the classification task, the proposed algorithms can be easily extended to target detection, semantic segmentation, and other tasks of HSIs by changing the structure of EvaluateNet. It deserves to be investigated as a possible future work.

REFERENCES

[1] C.-I. Chang, *Hyperspectral Data Exploitation: Theory and Applications*. Hoboken, NJ, USA: Wiley, 2007.
[2] X. Jia, B.-C. Kuo, and M. M. Crawford, "Feature mining for hyperspectral image classification," *Proc. IEEE*, vol. 101, no. 3, pp. 676–697, Mar. 2013.
[3] P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. Comput.*, vol. C-26, no. 9, pp. 917–922, Sep. 1977.
[4] X. Cao, C. Wei, Y. Ge, J. Feng, J. Zhao, and L. Jiao, "Semi-supervised hyperspectral band selection based on dynamic classifier selection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1289–1298, Apr. 2019.
[5] H. Su, Q. Du, G. Chen, and P. Du, "Optimized hyperspectral band selection using particle swarm optimization," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2659–2670, Jun. 2014.
[6] A. Shi, H. Gao, Z. He, M. Li, and L. Xu, "A hyperspectral band selection based on game theory and differential evolution algorithm," *Int. J. Smart Sens. Intell. Syst.*, vol. 9, no. 4, pp. 1971–1990, 2016.

[7] J. Feng, L. C. Jiao, X. Zhang, and T. Sun, "Hyperspectral band selection based on trivariate mutual information and clonal selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 7, pp. 4092–4105, Jul. 2014.

[8] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[9] R. Huang and M. He, "Band selection based on feature weighting for classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 2, pp. 156–159, Apr. 2005.

[10] B. Demir and S. Ertürk, "Phase correlation based redundancy removal in feature weighting band selection for hyperspectral images," *Int. J. Remote Sens.*, vol. 29, no. 6, pp. 1801–1807, Mar. 2008.

[11] N. Keshava, "Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 7, pp. 1552–1565, Jul. 2004.

[12] H. Zhai, H. Zhang, L. Zhang, and P. Li, "Laplacian-regularized low-rank subspace clustering for hyperspectral image band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1723–1740, Mar. 2019.

[13] C. Sui, Y. Tian, Y. Xu, and Y. Xie, "Unsupervised band selection by integrating the overall accuracy and redundancy," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 185–189, Jan. 2015.

[14] Q. Wang, F. Zhang, and X. Li, "Hyperspectral band selection via optimal neighborhood reconstruction," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8465–8476, Dec. 2020.

[15] P. Bajcsy and P. Groves, "Methodology for hyperspectral band selection," *Photogramm. Eng. Remote Sens.*, vol. 70, no. 7, pp. 793–802, Jul. 2004.

[16] J.-H. Kim, J. Kim, Y. Yang, S. Kim, and H. S. Kim, "Covariance-based band selection and its application to near-real-time hyperspectral target detection," *Opt. Eng.*, vol. 56, no. 5, May 2017, Art. no. 053101.

[17] C.-I. Chang, Q. Du, T.-L. Sun, and M. L. G. Althouse, "A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 6, pp. 2631–2641, Nov. 1999.

[18] C.-I. Chang and S. Wang, "Constrained band selection for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 6, pp. 1575–1585, Jun. 2006.

[19] H.-C. Li, C.-I. Chang, L. Wang, and Y. Li, "Constrained multiple band selection for hyperspectral imagery," in *Proc. IGARSS*, Jul. 2016, pp. 6149–6152.

[20] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

[21] S. Jia, G. Tang, J. Zhu, and Q. Li, "A novel ranking-based clustering approach for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 88–102, Jan. 2016.

[22] K. Tan, E. Li, Q. Du, and P. Du, "Hyperspectral image classification using band selection and morphological profiles," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 1, pp. 40–48, Jan. 2014.

[23] X. Bai, Z. Guo, Y. Wang, Z. Zhang, and J. Zhou, "Semisupervised hyperspectral band selection via spectral–spatial hypergraph model," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2774–2783, Jun. 2015.

[24] A. Vali, S. Comai, and M. Matteucci, "Deep learning for land use and land cover classification based on hyperspectral and multispectral Earth observation data: A review," *Remote Sens.*, vol. 12, no. 15, p. 2495, Aug. 2020.

[25] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019.

[26] H. Wang, C. Tao, J. Qi, H. Li, and Y. Tang, "Semi-supervised variational generative adversarial networks for hyperspectral image classification," in *Proc. IGARSS*, Jul. 2019, pp. 9792–9794.

[27] L. Zou, X. Zhu, C. Wu, Y. Liu, and L. Qu, "Spectral–spatial exploration for hyperspectral image classification via the fusion of fully convolutional networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 659–674, 2020.

[28] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.

[29] J. Feng, D. Li, J. Chen, X. Zhang, X. Tang, and X. Wu, "Hyperspectral band selection based on ternary weight convolutional neural network," in *Proc. IGARSS*, Jul. 2019, pp. 3804–3807.

[30] P. Ribalta Lorenzo, L. Tulczyjew, M. Marcinkiewicz, and J. Nalepa, "Hyperspectral band selection using attention-based convolutional neural networks," *IEEE Access*, vol. 8, pp. 42384–42403, 2020.

[31] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*. [Online]. Available: http://arxiv.org/abs/1308.3432

[32] L. Zhao, Y. Zeng, P. Liu, and G. He, "Band selection via explanations from convolutional neural networks," *IEEE Access*, vol. 8, pp. 56000–56014, 2020.

[33] Y. Zhan, D. Hu, H. Xing, and X. Yu, "Hyperspectral band selection based on deep convolutional neural network and distance density," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2365–2369, Dec. 2017.

[34] P. Ghamisi, Y. Chen, and X. X. Zhu, "A self-improving convolution neural network for the classification of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 10, pp. 1537–1541, Oct. 2016.

[35] R. S. Sutton, "Introduction: The challenge of reinforcement learning," in *Reinforcement Learning*. Boston, MA, USA: Springer, 1992, pp. 1–3.

[36] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," 2020, *arXiv:2003.03600*. [Online]. Available: http://arxiv.org/abs/2003.03600

[37] P. L. Ruvolo, I. Fasel, and J. R. Movellan, "Optimization on a budget: A reinforcement learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1385–1392.

[38] V. Francois-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," 2018, *arXiv:1811.12560*. [Online]. Available: http://arxiv.org/abs/1811.12560

[39] V. Mnih *et al.*, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: http://arxiv.org/abs/1312.5602

[40] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.

[41] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[42] Z. Liu *et al.*, "MetaPruning: Meta learning for automatic neural network channel pruning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3296–3305.

[43] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, Jan. 1984.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[45] Y. Cai, X. Liu, and Z. Cai, "BS-nets: An end-to-end framework for band selection of hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1969–1984, Mar. 2020.

[46] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.

[47] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," 2016, *arXiv:1611.06440*. [Online]. Available: http://arxiv.org/abs/1611.06440

[48] R. Archibald and G. Fann, "Feature selection and classification of hyperspectral images with support vector machines," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 674–677, Oct. 2007.

[49] R. Hansch, A. Ley, and O. Hellwich, "Correct and still wrong: The relationship between sampling strategies and the estimation of the generalization error," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3672–3675.

[50] J. Liang, J. Zhou, Y. Qian, L. Wen, X. Bai, and Y. Gao, "On the sampling strategy for evaluation of spectral–spatial methods in hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 862–880, Feb. 2017.

Jie Feng (Member, IEEE) received the B.S. degree from Chang'an University, Xi'an, China, in 2008, and the Ph.D. degree from Xidian University, Xi'an, China, in 2014.

She is an Associate Professor with the Laboratory of Intelligent Perception and Image Understanding, Xidian University. Her interests include remote sensing image processing, deep learning, and machine learning.

Di Li received the B.S. degree from Xidian University, Xian, China, in 2018, where he is pursuing the M.S. degree with the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education, School of Artificial Intelligence.

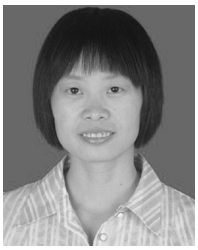
His interests include deep learning and remote sensing image processing.

Jing Gu (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 2007 and 2010, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, in 2016.

She is an Associate Professor with the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education of China, Xidian University. Her research interests include image processing, machine learning, and pattern recognition.

Xianghai Cao (Member, IEEE) received the B.E. and Ph.D. degrees from the School of Electronic Engineering, Xidian University, Xi'an, China, in 1999 and 2008, respectively.

Since 2008, he has been with Xidian University, where he is an Associate Professor with the School of Artificial Intelligence. His research interests include remote sensing image processing, pattern recognition, and deep learning.



Ronghua Shang (Member, IEEE) received the B.S. degree in information and computation science and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2003 and 2008, respectively.

She is a Professor with Xidian University. Her research interests include machine learning, pattern recognition evolutionary computation, image processing, and data mining.

Xiangrong Zhang (Senior Member, IEEE) received the B.S. and M.S. degrees in computer application technology from the School of Computer Science, Xidian University, Xi'an, China, in 1999 and 2003, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the School of Electronic Engineering, Xidian University, in 2006.

She is a Professor with the Key Laboratory of Intelligent Perception and Image Understanding, Ministry of Education, Xidian University. From January 2015 to March 2016, she was a Visiting Scientist with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. Her research interests include pattern recognition, machine learning, and remote sensing image analysis and understanding.

Licheng Jiao (Fellow, IEEE) received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively.

He has authored or coauthored more than 150 scientific articles. His research interests include image processing, natural computation, machine learning, and intelligent information processing. He was in charge of 40 important scientific research projects and published more than 20 monographs and a hundred articles in international journals and conferences.