Domain-Adaptive Few-Shot Learning for Hyperspectral Image Classification

Andi Zhang, Fang Liu, Jia Liu, Xu Tang, *Senior Member, IEEE*, Wenfei Gao, Donghui Li, and Liang Xiao, *Member, IEEE*

***Abstract*— Recently, hyperspectral image (HSI) classification by deep learning is flourishing. However, only a few labeled samples are available in practice since it is time-and-labor-consuming to label pixels in HSI (called target domain). This letter proposes a domain-adaptive few-shot learning (DAFSL) method to tackle this problem. Specifically, some other HSIs (called source domain) with large labeled samples are fully used as complementary information and a generative architecture is employed to adapt embedded features in the source domain to that of the target domain. We first perform domain adaptation with unsupervised learning. In detail, the embedded features are generated by the encoder of an autoencoder, where both source and target samples could be well recovered and the reconstruction loss is used to measure the gap between the source domain and the target domain. At the same time, the embedded features are put into a metric space for classification in the source domain and the encoder parameter is fine-tuned together with the classifier in the target domain with few labels so that both general and discriminative features are well captured. The experiment results show that DAFSL outperforms the other mainstream methods with limited labeled samples.**

***Index Terms*— Domain-adaptive (DA), few-shot learning (FSL), hyperspectral image (HSI) classification.**

I. Introduction

HYPERSPECTRAL image (HSI) performs important research value in view of its high-dimensional spectral and spatial properties. By exploring the abundant information in HSI, researchers could apply it to many fields [1], such as medical diagnosis and the aerospace field. Consequently, more attention has been paid to HSI analysis, such as classification.

In the early stage, most researchers take manually extracted features or traditional classifier for HSI classification, such as support vector machine (SVM) [2], which could achieve good performance to some extent but ignore the feature learning and representation.

Recently, deep learning methods exhibit superiority in image feature extraction [3]. Chen et al. [4] introduce stack autoencoder (SAE) to HSI classification, which performs training with unsupervised representation learning and supervised fine-tuning. However, the huge amount of parameters makes it difficult to train the network composed of a full connection (FC) layer.

In contrast, Mei et al. [5] propose 3-D autoencoder network for HSI classification. Different from the autoencoder structure above, the 3-D convolution kernel could establish the relation of the HSI cube with spectral-spatial information and decrease the training parameters to a great extent. Zhong et al. [6] replace the encoder part with a residual block and designs spectral–spatial residual network (SSRN), which further improves the HSI classification accuracy.

However, these methods often require adequate labeled HSI data, which is fairly limited because it is a time-and-labor-consuming matter. Consequently, researchers turn to the few-shot learning (FSL) [7] field to address the poor performance problem under insufficient labeled samples and many FSL methods [8], [9], [10] are widely applied to HSI classification. Liu et al. [11] propose the deep FSL (DFSL), which conducts the FSL in the metric space and achieves good performance with small labels. However, the DFSL could not solve the domain shift problem. Later, influenced by the work of DFSL, Li et al. [12] propose deep cross-domain FSL (DCFSL), and alleviates the domain shift problem between the source and target domain by conducting meta-learning alternately, which trains the conditional generative adversarial network to decrease the distribution difference. Although DCFSL tries to solve the domain shift by transfer learning to some extent, the knowledge from the source domain is not fully exploited. In this article, we carefully consider how to make full use of the source domain and propose DAFSL to achieve domain adaption by designing an autoencoder structure, where the source and target HSIs are put into the same network. By minimizing the reconstruction loss through alternate training, the gap between both source and target domains is diminishing, which could realize the knowledge transfer.

The major innovations of this letter are described in the following.

Manuscript received 27 August 2022; revised 16 October 2022; accepted 24 October 2022. Date of publication 26 October 2022; date of current version 22 November 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61802190, Grant 61906093, Grant 62171332, Grant 61871226, Grant 62071233, and Grant 62001226; in part by the Open Research Fund in 2021 of the Jiangsu Key Laboratory of Spectral Imaging and Intelligent Sense under Grant JSGP202101 and Grant JSGP202204; and in part by the Natural Science Foundation of Jiangsu Province, China, under Grant BK20190451. *(Corresponding author: Fang Liu.)*

Andi Zhang, Fang Liu, Jia Liu, Wenfei Gao, Donghui Li, and Liang Xiao are with the Jiangsu Key Laboratory of Spectral Imaging & Intelligent Sense, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: liufang_cs@njust.edu.cn).

Xu Tang is with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi’an 710071, China.

Digital Object Identifier 10.1109/LGRS.2022.3217502

1) To tackle the domain shift among source and target HSIs, we design the autoencoder structure. Both source and target HSIs share the same parameters, which help retain the common knowledge in the feature space to perform domain alignment.
2) In view of the unsupervised learning during the domain adaption by data reconstruction, the discriminative features could be neglected. Consequently, we adopt the fine-tuning strategy in the method. Supervised learning further improves the model the ability to discover the unique features of both domains, which helps to obtain better performance.

II. Proposed Approach

A. Framework of DAFSL

In the proposed DAFSL for HSI classification, the whole data set $\boldsymbol{D}_{\text{ALL}} = \boldsymbol{D_S} \bigcup \boldsymbol{D_T}$ consists of two parts. $\boldsymbol{D_S}$ represents the source HSIs consisting of multiple HSIs with abundant labeled samples while target HSI $\boldsymbol{D_T}$ is to be classified with the help of source HSIs.

As a preparation, to keep consistent with the classes of both domains, we randomly select classes equal to the class number in the target domain for training. And only a few target domain labels are used while source domain labels are sufficient. Then, both source and target HSIs are cut into image patches with the same size of width and height. Each patch stands for a training sample where the label is set to the same as the central pixel. The pretreated HSI patches of source and target domain are denoted as $\boldsymbol{P_s} \in R^{H\times W\times C_s}$ and $\boldsymbol{P_t} \in R^{H\times W\times C_t}$ respectively, where H and W are the height and width of image patches, while C_s and C_t represent the different channel numbers of source and target domain respectively. The process of the framework of DAFSL is shown in Fig. 1. Image patches are firstly modeled by the domain-adaptive (DA) module, which consists of a domain fusion part and a domain alignment part. The former part aims to obtain the latent features while the latter part is used to recover knowledge from the sharing of embedded features. With the optimization of reconstruction loss, the embedded features will capture the adaptive knowledge for the target HSI eventually. What is more, the embedded features conduct FSL through a classifier and the encoder parameters are fine-tuned together with the classifier. As a result, the embedded features not only obtain abundant knowledge from source domain data but also distinguish unseen target data.

B. DA Learning

Multiple source domains have been introduced to help classify the land covers in the target domain. In fact, the fit degree between the source and target domain is vague and leads to domain shift. To address the above existing problems, we propose DAFSL shown in Fig. 1 for domain adaption.

1) Domain Fusion: The domain fusion module (the structure included in the green box in Fig. 1) aims to map different domains to the same feature space.

In view of the inconsistent channels between the source and target domains, a spectral mapping function is used to keep data of the same size to ensure that they could be fed into the same network for training. The process of spectral dimensionality reduction can be formulated as

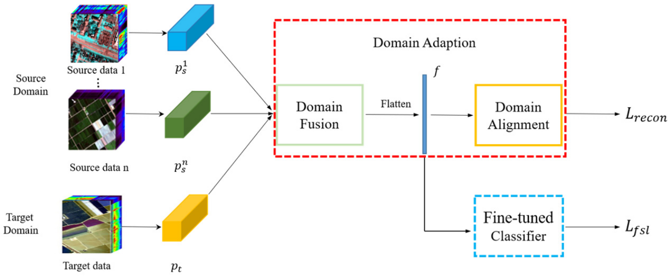


Fig. 1. Framework of DAFSL.

$$\hat{\boldsymbol{p}}_s{}^{H\times W\times C} = h_{\text{BN}}\left(\text{conv}_{1\times1}\left(\left(\hat{\boldsymbol{p}}_s{}^{H\times W\times C_s}\right)\right)\right) \tag{1}$$

$$\hat{\boldsymbol{p}}_t{}^{H\times W\times C} = h_{\text{BN}}\left(\text{conv}_{1\times1}\left(\left(\hat{\boldsymbol{p}}_t{}^{H\times W\times C_t}\right)\right)\right) \tag{2}$$

where h_{BN} and $\text{conv}_{1\times1}$ represent the batch normalization layer and 2-D convolution layer. $\hat{\boldsymbol{p}}_s$ and $\hat{\boldsymbol{p}}_t$ represent the image patches after dimensionality reduction in the source domain and target domain, respectively.

So far, the source and target domains could be integrated into the same network for feature learning. Considering the rich spatial-spectral features of HSI, a 3-D residual network is adopted as the encoder to map the source and target domains to the same metric space, and the knowledge of the different domains will be migrated to the parameters of the encoder. The process of domain fusion can be described as

$$\boldsymbol{f_s} = E_\varphi(\hat{\boldsymbol{p}}_s) \tag{3}$$

$$\boldsymbol{f_t} = E_\varphi(\hat{\boldsymbol{p}}_s) \tag{4}$$

where $\boldsymbol{f_s}$ and $\boldsymbol{f_t}$ are the corresponding embedding features, E is the encoder part, and φ is the learnable parameter of E.

2) Domain Alignment: After obtaining the representation of different domains in the feature space, we perform domain alignment in the form of unsupervised learning. Specifically, the $\boldsymbol{f_s}$ and $\boldsymbol{f_t}$ will recover the HSI patches in a domain alignment module (the structure included in the orange box in Fig. 1) with multiple transposed convolution and up-sampling layers. The process can be formulated as

$$\tilde{\boldsymbol{p}}_s = D_\theta(\boldsymbol{f_s}) \tag{5}$$

$$\tilde{\boldsymbol{p}}_t = D_\theta(\boldsymbol{f_t}) \tag{6}$$

where $\tilde{\boldsymbol{p}}_s$ and $\tilde{\boldsymbol{p}}_t$ are the reconstructed images of source HSI and target HSI patches respectively, D is the decoder network, and θ is the learnable parameter of D.

In the training phase, we feed the patches into the encoder alternatively in the form of source-target-source and apply reconstruction loss to force the domains to recover as well as possible. When the source HSI finishes one round of training, the target HSI will be fine-tuned based on the network parameters of the previous round. That is to say, the network will adjust the knowledge of the source domain for the target domain according to the reconstruction loss. With constant learning, the gap between the source and target domain is decreasing. Thus, the common knowledge of different domains is mixed into the φ and θ, which realize the knowledge transferred toward the target domain. Consequently, the process of image recovery is also the process of domain alignment. The domain alignment is given by

$$L^{\text{recon}} = \|\tilde{\boldsymbol{p}} - \hat{\boldsymbol{p}}\|_2^2 \tag{7}$$

where L^{recon} represents the reconstruction loss to measure the quality of reconstructed images.

C. DAFSL for HSI Classification

To further represent the whole knowledge from different domains. We conduct FSL to make the encoder learn the discriminative features at the same time.

After the division of support sets and query sets, samples are projected to a metric space via the encoder. It is noticed that the shot number per class in support set is configured to 5 in the experiment. As a result, we take the mean vector in the feature space to represent the prototype of each class. The prototype can be calculated by

$$\mathbf{c_k} = \frac{1}{|\boldsymbol{S_k}|} \sum_{(\boldsymbol{s},\boldsymbol{y}) \in \boldsymbol{S_k}} E_\varphi(\boldsymbol{s}) \tag{8}$$

where $\boldsymbol{c}_k$ is the prototype of class k and $\boldsymbol{s}$ denotes the samples from support set $\boldsymbol{S}$. Then the latent features of the samples from support sets are presented by a prototype. Furthermore, the distance between the query sets and support sets are calculated and could be converted to class probability distribution through the SoftMax function. The process can be formulated as

$$P_\theta(\boldsymbol{y} = \boldsymbol{k} \mid \boldsymbol{q} \in \boldsymbol{Q}) = \frac{\exp\bigl(-\operatorname{dist}\bigl(E_\varphi(\boldsymbol{q}), \boldsymbol{c_k}\bigr)}{\sum_{k=1}^{C} \exp\bigl(-\operatorname{dist}\bigl(E_\varphi(\boldsymbol{q}), \boldsymbol{c_k}\bigr)} \tag{9}$$

where $\boldsymbol{c_k}$ is the prototype of class k, dist$(*)$ denotes the Euclidean distance function to measure the distance between the $\boldsymbol{q}$ and $\boldsymbol{c_k}$, and $\boldsymbol{y}$ is the label of $\boldsymbol{q}$.

With the information of labels, the latent features could establish the relation of specific categories from different domains. The class probability distribution is then changed to FSL loss through cross entropy function as follows:

$$L^{\text{fsl}} = -\sum_{\boldsymbol{x},\boldsymbol{y} \in \boldsymbol{Q}.} \boldsymbol{b_k} \log p_\theta(\boldsymbol{y} = \boldsymbol{k} \mid \boldsymbol{x}) \tag{10}$$

where $\boldsymbol{b}_k$ is the corresponding one hot label of $\boldsymbol{x}$ and the parameters of the encoder are further fine-tuned by L^{fsl} to learn the special domain information. The whole loss of the network is the linear combination of reconstruction loss and FSL loss. The formulation can be described as

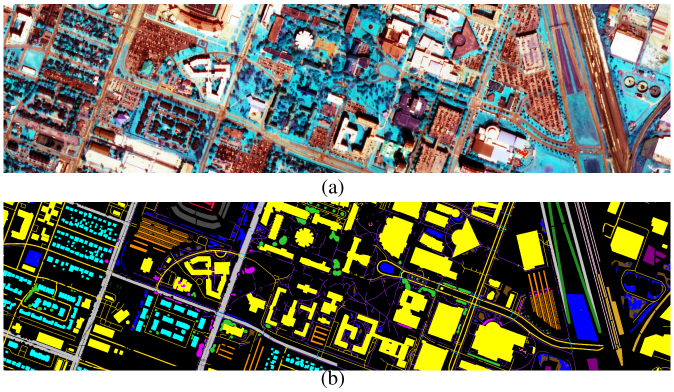
$$L^{\text{whole}} = \lambda L^{\text{recon}} + (1-\lambda) L^{\text{fsl}} \tag{11}$$

where λ denotes the hyper balance parameter.

By updating the L^{whole}, the learned architecture is able to capture both the general and the discriminative features of different domains. Algorithm 1 could describe the whole process of DAFSL.

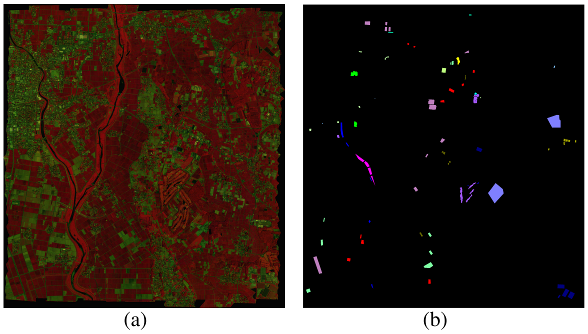
Algorithm 1 Process of DAFSL

```
Require: Distribution of tasks: $P(D_k), k \in [1, n]$
Require: learning rate $\alpha, \beta$
Initialize the parameters $\varphi$ and $\theta$ in the network
while epoch $i < I$ do
  for all $D_k$ do
    Samples from $D_k$ to constitute episode task $T_j$
    for all task $T_j$ do
      Calculate $\nabla_\varphi L_{T_j}(E_\varphi)$ with $L_{fsl}$ and update parameters in layers with gradient:
      $\varphi'_j = \varphi - \alpha_\varphi L_{T_j}(E_\varphi)$
      Calculate $\nabla_{(\varphi,\theta)} L_{T_j}(E_\varphi, D_\theta)$ with $L^{recon}$ and update parameters in layers with gradient:
      $\theta'_j = \theta - \nabla_{(\varphi,\theta)} L_{T_j}(E_\varphi, D_\theta)$
      $\varphi''_j = \varphi - \nabla_{(\varphi,\theta)} L_{T_j}(E_\varphi, D_\theta)$
    end for
    $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_j \sim P(D_k)} \mathcal{L}_{\mathcal{T}_j}(E_{\theta'_j})$
    $\varphi \leftarrow \varphi - \beta \nabla_\varphi \sum_{\mathcal{T}_j \sim P(D_k)} \mathcal{L}_{\mathcal{T}_j}(E_{\varphi''_j})$
    $-\beta \nabla_{(\varphi,\theta)} \sum_{\mathcal{T}_j \sim P(D_k)} \mathcal{L}_{\mathcal{T}_j}(E_{\varphi''_j}, D_{\theta'_j})$
  end for
end while
```


(a)

(b)

Fig. 2. HoustonU dataset. (a) Red green blue (RGB) image. (b) Ground-truth map.


(a)

(b)

Fig. 3. Chikusei dataset. (a) RGB image. (b) Ground-truth map.

III. Experiments

The experiments are conducted on the Salinas Valley (SV) dataset with the source data of the Chikusei and HoustonU datasets. Chikusei dataset consists of 2517 × 2335 pixels with 128 bands and contains 19 classes. HoustonU dataset consists of 2384 × 601 pixels with 50 bands and contains 20 classes. SV consists of 512 × 217 pixels with 220 bands and contains 16 classes. Chikusei and HoustonU datasets are set as source data while the SV dataset is the target data. The corresponding datasets are shown in Figs. 2–4.

A. Parameter Setting

In the training phase, we randomly select 16 classes from each source domain to keep consistent with the SV dataset, five labeled samples per class of target HSI and 200 labeled

TABLE I
CLASSIFICATION RESULTS ON SV DATASET

Method / Class name	SVM [2]	3DCNN [5]	SSRN [6]	DFSL [11]	DCFSL [12]	RN-FSC [13]	DAFSL
Brocoli_green_weeds_11	96.75	99.90	97.55	98.05	**100**	98.75	98.80
Brocoli_green_weeds_2	86.42	99.27	98.97	97.64	**99.87**	99.26	98.70
Fallow	83.11	97.82	92.47	95.33	**99.09**	80.32	98.38
allow_rough_plow	99.11	99.78	96.33	99.71	**100**	95.32	99.93
Fallo_smooth	93.63	93.27	94.12	91.73	**95.62**	92.16	95.62
Stubble	99.17	100	99.14	100	99.90	99.12	**100**
Celery	94.44	99.64	99.96	99.08	**99.97**	99.41	98.57
Grapes_untrasined	64.15	44.43	86.90	57.07	70.76	52.74	**82.71**
Soil_vinyard_develop	96.12	99.92	99.12	99.48	**99.61**	97.68	98.39
Corn_senesced_green_weeds	70.39	87.72	92.34	87.78	91.02	**91.66**	89.73
Lecture_romaine_4wk	91.77	99.34	94.13	**99.91**	98.59	99.21	99.44
Lecture_romaine_5wk	98.13	**99.95**	99.14	99.32	99.79	99.84	99.12
Lecture_romaine_6wk	97.22	98.79	95.78	99.12	98.57	98.13	**99.34**
Lecture_romaine_7wk	97.52	**99.81**	99.13	98.50	99.62	95.14	99.34
Vinyard_untrained	60.24	79.23	55.97	71.11	59.31	61.32	**79.83**
Vinyard_vertical_trellis	80.26	99.38	98.91	98.67	99.00	97.12	**99.50**
OA (%)	80.75	84.87	86.33	85.37	87.5	83.98	**92.29**
AA (%)	89.12	93.92	93.24	93.28	94.44	92.82	**96.09**
Kappa*100	79.27	83.32	84.95	83.81	86.1	84.74	**91.43**

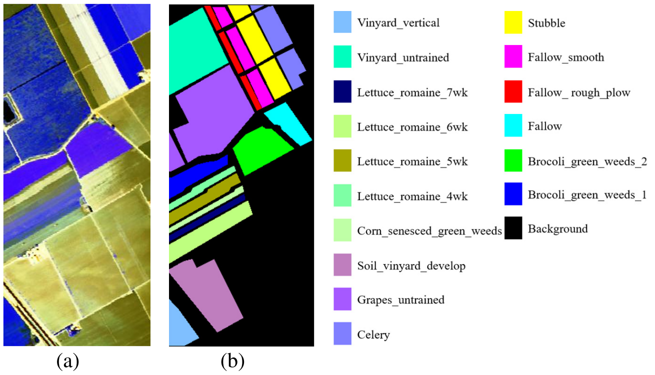


Fig. 4. SV dataset. (a) RGB image. (b) Ground-truth map.

samples per class of each source domain are picked to constitute the training set while the rest of target samples are used for the test. The original HSIs are divided into image patches of size 9 × 9 and the parameter settings of the encoder are the same as [11] to ensure fairness. The decoder network takes the symmetrical network as the encoder. The parameter λ is set to 0.7. And the same adam optimizer is selected as [11] where learning rate α is set to 0.001 and the decay rate of β is set to the default value of (0.9, 0.999). Each comparing experiment is trained by 20000 epochs. All the experiments are performed on the GPU of NVIDIA GeForce RTX 3070 with 8G memory. The code is implemented on the open-source software framework Pytorch with python 3.6.

B. Comparison Experiments and Result Exhibition

In order to validate the effectiveness of the DAFSL method. We compare DAFSL with traditional method SVM [2], deep learning method 3D-CNN [5], SSRN [6]. And FSL methods including DFSL [11], DCFSL [12], and relation network for HSI few-shot classification (RN-FSC) [13]. As for DFSL, DCFSL, and RN-FSC, the same training set of 200 labeled samples for each source HSI and five labeled samples are

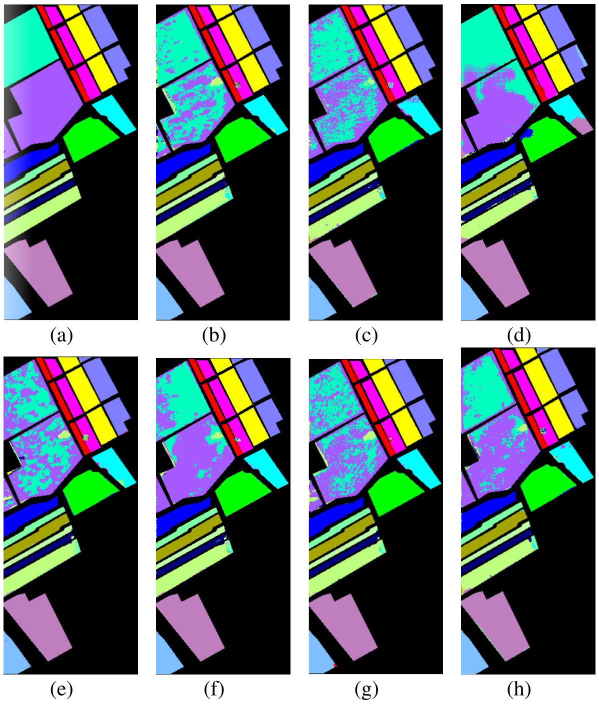


Fig. 5. Results on SV dataset. (a) Ground-truth. (b) SVM [2]. (c) 3D-CNN [5]. (d) SSRN [6]. (e) DFSL [11]. (f) DCFSL [12]. (g) RN-FSC [13]. (h) DAFSL.

TABLE II
TRAINING TIME (S), INFERENCE TIME, FLOPS, AND PARAMETERS ON DIFFERENT METHODS

Dataset	Method	SVM [2]	3DCNN [5]	SSRN [6]	DFSL [11]	DCFSL [12]	RN-FSC [13]	DAFSL
SV	Traning time	0.31	72.34	86.91	812.85	1647.12	581.26	1963.89
	Inference time	0.65	6.82	40.26	4.77	5.06	50.18	5.23
	FLOPs	-	216635	992309	42536869	7125246	49523212	52144239
	Parameters	-	127745	199153	48954	4326939	316944	4674235

chosen as the training set in the target domain to ensure fairness. While In SVM, 3D-CNN and SSRN, the source domain cannot be available. Consequently, the training set only contains five labeled samples of the target domain. All the results are shown in Fig. 5 and Table I. As for the methods without cross domain, due to the inability to utilize the source domain data, almost all the indicators are worse than these FSL methods. And for FSL methods, although the accuracy of proposed DAFSL in some classes is not as high as other FSL methods, the classification accuracy of each class is very balanced, which is attributable to the domain adaption and fine-tuning strategy. The DA learning could greatly improve the accuracy of the algorithm because it introduces multiple source HSIs and tries to make full use of the knowledge to help classify the target land covers. At the same time, the fine-tuning strategy by training the source and target HSIs according to respective labels makes the general knowledge retained in the latent features, which obtains a more balanced performance than other methods. And the accuracy is relatively higher.

C. Computation Complexity

To exhibit the efficiency of all the methods, we further compare the training times, inference times, floating-point operations (FLOPS), and parameters in Table II.

It is shown that 3D-CNN takes less time to train and test than SVM because the network requires many training epochs to get converged. The long-time iteration could also obtain the inner feature of HSIs. The SSRN introduces a deep residual block to get better performance. Consequently, the time and parameters of the network are larger than 3D-CNN. As for DFSL, DCFSL, RN-FSC, and DAFSL, time is expensively spent on transfer learning with source domain data. It can be observed that although our DAFSL takes the longest time and largest parameters than other methods, the most effort has been done to acquire the best performance.

IV. CONCLUSION

This letter puts forward a new method to handle hyperspectral classification with few labels. The DAFSL consists of two important modules. The DA module fuses different domains in the latent space by an encoder, and the recovery of HSIs with decoder by reconstruction loss makes the general domain retained in the parameters of networks. There is a significant improvement in classification accuracy for each category over other algorithms according to Table I and Fig. 5, which shows the effectiveness of domain adaptation of the DA module. Then the fine-tuning module puts the embedding features in a metric space to conduct the supervised classification with few labels according to the distance. Consequently, the parameters of the encoder are further fine-tuned to learn the discriminative features of both the source domain and target domain. Experiments on HSIs indicate that the proposed method outperforms the mainstream algorithms.

REFERENCES

[1] 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.

[2] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[3] Z. Li et al., "Deep multilayer fusion dense network for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1258–1270, 2020.

[4] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[5] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3D convolutional autoencoder for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6808–6820, Sep. 2019.

[6] 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.

[7] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, May 2021, doi: 10.1145/3386252.

[8] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, Aug. 2021.

[9] Y. Zhang, W. Li, M. Zhang, Y. Qu, R. Tao, and H. Qi, "Topological structure and semantic information transfer network for cross-scene hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 16, 2021, doi: 10.1109/TNNLS.2021.3109872.

[10] Y. Zhang, W. Li, R. Tao, J. Peng, Q. Du, and Z. Cai, "Cross-scene hyperspectral image classification with discriminative cooperative alignment," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 9646–9660, Nov. 2021.

[11] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2018.

[12] Z. Li, M. Liu, Y. Chen, Y. Xu, W. Li, and Q. Du, "Deep cross-domain few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2021.

[13] K. Gao, B. Liu, X. Yu, J. Qin, P. Zhang, and X. Tan, "Deep relation network for hyperspectral image few-shot classification," *Remote Sens.*, vol. 12, no. 6, p. 923, Mar. 2020.