Cross-Domain Land Cover Classification of Remote Sensing Images Based on Full-Level Domain Adaptation

Yuke Meng, Zhanliang Yuan, Jian Yang, Peizhuo Liu, Jian Yan, Hongbo Zhu, Zhigao Ma, Zhenzhao Jiang, Zhouwei Zhang, and Xiaofei Mi

***Abstract*—The use of remote sensing images for land cover classification is an important and challenging pixel-level classification task. However, the different distribution of the same land cover categories across different datasets, the accuracy of the classification is significantly reduced when a classification model trained on one dataset is used directly on another dataset. To address the issue, numerous unsupervised domain adaptation (UDA) methods have been proposed. However, the existing UDA methods focus mainly on natural images and are not suited to remote sensing images with large variations in spectral information and texture features. Therefore, we develop a new full-level domain adaptation network (FLDA-NET) applicable for cross-domain land cover classification of remote sensing images. It aligns the source domain and target domain through a two-stage process encompassing image-level, feature-level, and output-level. In stage I, we align at image-level by converting the source domain image to the target domain image style. In stage II, at the feature-level we align the entropy of the two domain features. At the output-level we do not use simple global alignment, but category-level alignment. Furthermore, a self-training strategy based on superpixel segmentation and softmax probability is proposed to further enhance the model's performance on the target domain. Extensive experiments on our proposed FLDA-NET are performed on the Potsdam and Vaihingen datasets and compared with other advanced UDA methods. The outcome demonstrates that this approach greatly improves the ability of cross-domain land cover classification in remote sensing images.**

***Index Terms*—Domain adaptation, generative adversarial network (GAN), land cover, remote sensing (RS), semantic segmentation.**

I. Introduction

With the growth of remote sensing (RS) technology, the quality and resolution of RS images have gradually improved. At the same time, the difficulty of obtaining these images has significantly decreased, making it easy for people to acquire huge numbers of images with varying spectral characteristics and resolutions. These images provide an important information base for understanding the Earth system at various scales. Land cover classification refers to the process of classifying and identifying surface objects using RS images and categorizing them into different land cover types. It has important applications in forest resource monitoring, urban planning, and climate change assessment [1], [2], [3]. Land cover classification is an important and challenging pixel-level classification task. On low and medium resolution images, traditional approaches use machine learning algorithms, such as decision trees [4] and support vector machines [5] to achieve land cover classification [6], [7], [8]. However, with the increase in image resolution, traditional machine learning algorithms are not applicable to high-resolution images and their classification accuracy is often poor [9]. In recent years, with its powerful feature extraction and classification capabilities, deep learning technology has been increasingly applied in the field of RS interpretation [10], [11], [12]. Many semantic segmentation models are widely applied for land cover classification tasks, such as the full convolutional network (FCN) [13], U-Net [14] and SegNet [15] with encoder-decoder structure, and DeepLab family of networks [16]. These models have achieved excellent performance. However, in practical applications, there are some problems that cause these models not to be better applied.

First, land cover classification using deep learning approaches typically requires sufficient training samples [17]. This means that huge numbers of pixel-level labels must be manually labeled, which is very expensive. Second, deep learning models are particularly sensitive to changes in data distribution, which requires that the training data (i.e., the source domain data) and the test data (i.e., the target domain data) adhere to the hypothesis of being independently and equivalently distributed. However,

Manuscript received 20 March 2024; revised 12 May 2024; accepted 28 May 2024. Date of publication 31 May 2024; date of current version 18 June 2024. This work was supported in part by the National Key R&D Program of China under Grant 2021YFE0117300, in part by the Civil Aerospace Technology Preresearch Project of China's 14th Five-Year Plan, and in part by the Major Project of High Resolution Earth Observation System under Grant 30-Y60B01-9003-22/23. *(Corresponding author: Xiaofei Mi.)*

Yuke Meng, Zhanliang Yuan, Zhigao Ma, and Zhenzhao Jiang are with the School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China (e-mail: 212104020034@home.hpu.edu.cn; yuan6400@hpu.edu.cn; zhigaoma2@gmail.com; jiangzhenzhao9527@gmail.com).

Jian Yang, Jian Yan, Zhouwei Zhang, and Xiaofei Mi are with the National Engineering Laboratory for Satellite Remote Sensing Applications, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: yangjian@aircas.ac.cn; yanjian97@mails.ucas.ac.cn; zhangzw@aircas.ac.cn; mixf@aircas.ac.cn).

Peizhuo Liu is with the Beihang University, Beijing 100191, China (e-mail: liupeizhuo@buaa.edu.cn).

Hongbo Zhu is with the North China Institute of Aerospace Engineering, Langfang 065000, China (e-mail: zhb999322@stumail.nciae.edu.cn).

Code is available online at https://github.com/Myk1011/FLDA-NET.

Digital Object Identifier 10.1109/JSTARS.2024.3407808

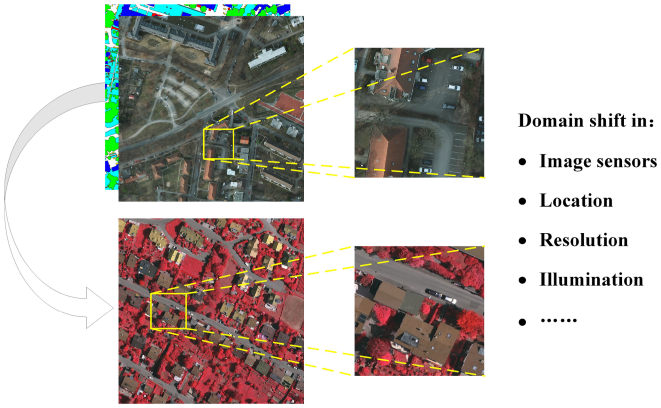


Fig. 1. Domain shift between RS images.

in practical applications, differences in imaging sensors, geographical location, spatial resolution, and light intensity lead to a large gap in the appearance and structural features between different RS images, resulting in the phenomenon of domain shift. In Fig. 1, we demonstrate this situation. At this point, if the land classification model that has been trained on the source domain images is applied to directly another set of very different target domain images, the model's performance will significantly decline. This is because different imaging sensors and light intensities can result in the same land cover class having completely different color information and texture features on different data. Therefore, it is necessary to relabel new image data each time we train the model, and cannot utilize the existing label data to its full extent.

To overcome these limitations, unsupervised domain adaptation (UDA) methods in computer vision have been developed to address the problem of excessive data distribution differences. UDA, a subset of transfer learning, seeks to reduce the data distribution gap between the two domains when the tasks are the same, leveraging labeled data from the source domain to improve the model's accuracy and generalization on the unlabeled target domain [18], [19]. Recently, adversarial learning [20] has been frequently employed in UDA tasks. Depending on the level and manner of the adversarial loss action, domain adaptation methods based on adversarial learning are categorized into image-level, feature-level, and output-level [21]. Image-level domain adaptation methods mainly transform the image style into the image style of the target domain on the basis of preserving the semantic information [22], [23], [24], [25]. Feature-level domain adaptation methods use adversarial learning to reduce the differences of data distributions from different domains in a common feature space [23], [26], [27]. On the other hand, output-level domain adaptation methods are similar to feature-level methods, but they use adversarial training on the output prediction probability distributions to make the prediction probability of the two domains close to the same [28], [29].

Although these UDA methods mentioned above already showed some success in semantic segmentation tasks, they are all performed on common natural images [23], [24], [28], [30]. In contrast, RS images possess richer spectral and textural information, and have special attributes that differ from natural images [31]. Therefore, the above UDA methods may not be well suitable to land cover classification. In recent years, some UDA approaches have been suggested in the remote sensing field. [32], [33], [34], [35]. However, most of these methods are based on single-level or dual-level alignment. In addition, these methods only consider whole image differences and ignore local category differences when performing domain adaptation training. In fact, when performing cross-domain land cover classification, the difficulty of domain adaptation varies between different land cover types due to factors such as imaging sensors and resolution. Following an extended period of domain adaptation training, some classes may have adapted, whereas others are difficult to adapt. In this case, considering only the adaptation of the whole image will result in negative migration of the already adapted classes.

To solve the above limitations, we introduce a novel full-level domain adaptation network (FLDA-NET) for the task of cross-domain land cover classification in RS images. Specifically, we divide into two stages to align the source and target domains. In Stage I, we use the classical image translation network [22] to align the source and target domains at the image-level, which does not need to pair the images of the two domains while maintaining a good transformation effect. Therefore, it greatly reduces the difficulty of data collection and preprocessing, and is very suitable for use on RS images. The Stage II minimizes the data distribution disparities between the two domains by means of adversarial learning. At the feature-level, we adjust the entropy distributions of the features to approach consistency, indirectly minimizing the entropy of the prediction results. At the output-level, we align categories that are difficult to adapt by weighting the adversarial loss, while preventing negative migration of already aligned categories. Finally, we introduce a new self-training strategy (SPST) based on superpixel segmentation and softmax probability. It is used on our FLDA-NET to generate high-confidence pseudolabels in the target domain, and the segmentation model is fine-tuned using these pseudo-labels to further enhance the model's generalization. In conclusion, the key contributions of this article can be outlined as follows.

1) A new full-level UDA method, FLDA-NET, is proposed for the task of cross-domain land cover classification of RS images, which uses adversarial learning to align the data distributions of the two domains in two stages at the image-level, feature-level, and output-level to achieve land cover classification over unlabeled target domain.
2) A SPST strategy is proposed based on the superpixel-based segmentation and softmax probability, by which high-confidence predictions are selected as pseudolabels, which will be used to continue the training of model after the domain adaptation in order to enhance the model's performance within the target domain.
3) Compared with other state-of-the-art UDA methods, our method is shown to perform more by performing experiments on the Potsdam and Vaihingen datasets. The impact of each level in our method is also discussed to ensure its effectiveness.

The rest of this article is organized as follows. Related work is described in Section II. We outline our suggested approach in Section III. Section IV presents the results and analysis of the experiment. Section V conducts the discussion. Finally, Section VI concludes this article.

II. Related Work

A. Land Cover Classification

RS land cover classification aims to assign each image element in an image to a predefined land cover class by analyzing and identifying the spatial distribution features of different feature types in the image. For this purpose, traditional methods can use spectral and textural features of individual image elements for classification [36]. However, with the increasing spatial resolution of RS images, land cover classification relies not only on spectral features of pixels, but more on contextual information and spatial relationships between land cover types [34].

Deep learning technology has shown impressive advantages in land cover classification in RS images recently. Compared with traditional land cover classification methods, deep learning methods have more powerful feature learning capabilities and can automatically extract features from images. FCN [13] is the first network to address semantic segmentation tasks end-to-end. Compared with traditional convolutional neural network, FCN can take inputs of any size and produce pixel-level prediction results of the same dimensions, which makes FCN well suited for semantic segmentation tasks. The structure of FCN consists of convolutional and upsampling layers, which can efficiently learn the spatial information of an image and generate pixel-level prediction results. Subsequently, more and more deep learning networks have been proposed for semantic segmentation tasks. Representative ones include U-Net [14] and SegNet [15] with encoder-decoder structure, the DeepLab family of networks [16], and so on. These networks have been successfully used in land cover classification tasks with excellent performance. However, these deep learning models need to mark a lot of pixel-level labels for training, which is costly and takes time. On the other hand, significant data distribution differences between sets of RS images are due to differences in sensors, geographical location, and resolution and light intensity. In this case, the classification ability of a land cover classification model trained on one set of data is significantly degraded when applied directly to another different set of data. As a result, it is required to relabel the data for each new dataset, which does not utilize the existing labeled data to its full extent.

B. Generative Adversarial Network

Generative adversarial network (GAN) [20] has shown strong capabilities in image generation and unsupervised learning. Therefore, UDA methods based on adversarial learning have been proposed to solve the issue of lack of labels in RS images. Initially, they were mainly applied to scene classification of RS images [37], [38] and later more and more researchers applied them to semantic segmentation tasks [39], [40]. These studies proved that adversarial learning based UDA has good results in processing RS images with different domains. Therefore, we make full use of its advantages and apply adversarial learning to sensing images cross-domain land cover classification.

Specifically, GAN typically comprises a generator G and a discriminator D, each representing a separate model. In the adversarial learning process, G is responsible for generating fake data distributions with the intention of generating data that closely resembles real data, making it impossible for the discriminator to accurately differentiate between real and fake data. D is responsible for determining whether the entered data is real data or generated fake data, and its goal is to discriminate as accurately as possible between real data and fake data. A dynamic game relationship is formed between G and D. A dynamic equilibrium point is reached through constant competition and confrontation. Its formula is described as follows:

$$\min_G \max_D L_{\text{GAN}}(G, D) = E_{x\sim P_{\text{data}}}[\log D(x)] + E_{x\sim P_G}[\log(1 - D(G(x)))] \quad (1)$$

where P_{data} represents real data and P_G represents generated fake data. In the adversarial learning process, the generator G and the discriminator D have opposite optimization objectives. Minimizing the loss $L_{\text{GAN}}(G, D)$ by optimizing the parameters of G and maximizing the loss $L_{\text{GAN}}(G, D)$ by optimizing the parameters of D. By alternately training G and D, they eventually reach a balanced state.

C. Unsupervised Domain Adaptation

In order to remove the domain shift phenomenon and solve the problem of the lack of labels in the target domain, many GAN-based UDA methods have been developed for semantic segmentation tasks and deliver impressive performance results. Guo et al. [41] implemented feature-level domain transformation from virtual to real images using recurrently consistent GANs, introduced a dynamic perceptual network to enhance the quality of image generation. Tsai et al. [28] presented an adversarial domain adaptation approach for semantic segmentation that utilizes multilevel adversarial learning in the output level to improve the model's performance in the target domain. Li et al. [25], unlike the previous ones, proposed a new bidirectional learning network for the semantic segmentation tasks. The segmentation model and the image translation model can be optimized for one another through the use of the bidirectional learning technique. Vu et al. [30] presented a new entropy-based UDA approach that aligns the entropy of semantic prediction results at the output-level to achieve domain adaptation. While the aforementioned UDA methods have demonstrated encouraging outcomes in natural image semantic segmentation, they cannot be directly applicable to cross-domain land cover classification tasks because they overlook the spectral information, texture features, and spatial resolution properties of RS images.

With the wide application of UDA methods in natural image, more and more research work is focused on applying UDA methods to RS images. Makkar et al. [42] introduced a framework called ADDA, based on adversarial learning, to address complex tasks such as hyperspectral image classification

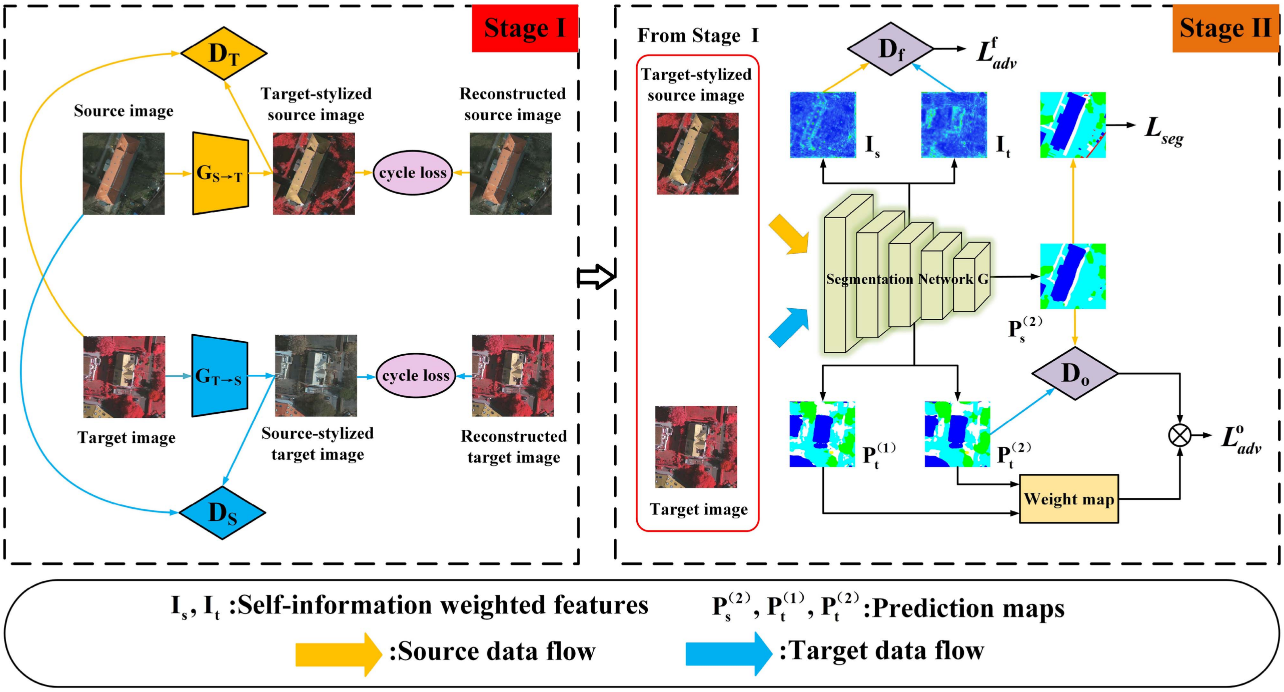

Stage I
D_T
Source image
Target-stylized source image
Reconstructed source image
G_S→T
cycle loss
Target image
G_T→S
Source-stylized target image
Reconstructed target image
cycle loss
D_S
Stage II
From Stage I
Target-stylized source image
Target image
D_f
L^f_adv
I_s
I_t
L_seg
Segmentation Network G
P_s^(2)
D_o
P_t^(1)
P_t^(2)
Weight map
L^o_adv
I_s, I_t :Self-information weighted features
P_s^(2), P_t^(1), P_t^(2):Prediction maps
:Source data flow
:Target data flow

Fig. 2. Our proposed overall architecture for FLDA-NET.

and large-scale semantic segmentation for RS images. Liu and Wang [43] proposed a novel network SFnet-DA that embeds domain adaptation into a selective self-attention mechanism to improve the extraction of water bodies with drastic scale changes. Zhang et al. [33] proposed a staged domain adaptation approach to align the semantic prediction results through GANs, both interdomain and intradomain. On the other hand, the model introduced by [44] applies structured domain adaptation into synthetic image generation and road segmentation in RS images by integrating feature pyramid networks into GANs to minimize the differences between two domain images. Yan et al. [45] jointly considered the information from both domains and proposed a triplet adversarial domain adaptation approach to discriminate at the output-level whether the two sets of segmentation prediction results are from the same domain or different domains. However, these UDA methods tend to implement domain adaptation only for single or dual levels. This is insufficient for RS images with complex structures and large distributional differences. Ji et al. [46] implemented full space domain adaptation for land cover classification of multisource RS images based on a GAN that makes full use of available vector maps. Peng et al. [47] implemented building extraction from high-resolution RS images using full-level domain adaptation, using Wallis filter method to transform the style of source domain images at the image-level, introducing a discriminator to constrain the features at the feature-level, and using a mean teacher model at the output-level to further improve the model effect. But these full-level UDA methods pursue global alignment and ignore local differences during adversarial learning. This can lead to negative migration of some already adapted categories when performing cross-domain land cover classification. Therefore, our work overcomes the above limitations and significantly improves the performance of cross-domain land cover classification of RS images.

III. Proposed Method

We frequently come across such a scenario when classifying land cover from RS images. An unlabeled target domain dataset X_T is needed for classification of land cover, while another source domain dataset (X_S, Y_S) has similar classification scenarios with sample labels. However, due to differences in sensors, geographical location, resolution and light intensity, etc. between them, the model trained on the data from the source domain cannot be directly used on the data from the target domain, and thus the existing labeled data cannot be fully utilized. For this reason, we translate the problem into a UDA task. Our goal is to learn supervised semantic information from the source domain and leverage this knowledge to construct a robust land cover classification model capable of excelling in the target.

Our proposed FLDA-NET is illustrated in Fig. 2. It is mainly tailored to classify land cover across domains in RS images. This framework consists of two stages. Stage I performs image-level domain adaptation, which is described in Section III-A. This stage mainly performs image style transfer, enabling the generation of source domain images stylized to align with the target domain's visual characteristics. Stage II takes the stylized source domain images in the target domain style, along with the source domain labels and target domain images as input. Here, adversarial learning is leveraged to ensure domain alignment at both the feature and output levels, which is given in Section III-B.

Meanwhile, in order to enhance the model accuracy in the target domain after full-level domain adaptation, a self-training (ST) approach is employed for fine-tuning the segmentation model, as given in Section III-C.

A. Stage I: Image-Level Domain Adaptation

In Stage I, our goal is to achieve image-level domain adaptation to convert the image style of the source domain images X_S with label Y_S to the target domain X_T, enabling them to mimic the characteristics of the target domain image (sensor type, spatial resolution, light intensity, etc.). Subsequently, by using the segmentation model trained on the source domain images transformed to the target domain style, we are able to achieve superior target domain performance.

We have followed the network structure proposed in [22] to implement an unpaired image translation network. The aim of this article is to use this network to achieve style translation from source domain images to target domain images, and to convert the source domain training dataset to the target domain style. We introduce a generator $G_{S\to T}$, which is utilized to align the distributions of the two domain images, and the generated result $G_{S\to T}(x_s)$ is used to deceive the discriminator. Conversely, the discriminator D_T outputs a binary value of 0 or 1 to determine the domain of the input data, with a loss function according to (2). During adversarial learning, the generator $G_{S\to T}$ is expected to produce source domain images with a style that mimics the target domain to deceive the discriminator D_T, whereas the discriminator D_T is expected to increase its discriminative power to distinguish the true target domain image x_t from $G_{S\to T}(x_s)$. The two are alternately trained to continuously improve the generation and discrimination capabilities and eventually reach a balanced state.

$$\begin{aligned} L_{\text{GAN}}^{s\to t}(G_{S\to T}, D_T, X_T, X_S) &= E_{x_t\sim X_T}[\log D_T(x_t)] \\ &+ E_{x_s\sim X_S}[\log(1 - D_T(G_{S\to T}(x_s)))]. \end{aligned} \tag{2}$$

However, the style conversion of the image in the actual process will not be able to be carried out as expected, this is because in (2), although x_s is successfully generated as the style $G_{S\to T}(x_s)$ of the target domain image, it cannot inherently assure that $G_{S\to T}(x_s)$ can still preserve the original content or structure of the sample x_s. For this reason, we use the cyclic consistency loss [22] to ensure that the original information from the source domain image is preserved during the image style conversion process, as shown in (3). In this process, we introduce another generator $G_{T\to S}$ that retranslates the stylized source domain image in the target domain back to the source domain's style according to the same loss $L_{\text{GAN}}^{t\to s}$, thus achieving cyclic consistency, i.e., $G_{T\to S}(G_{S\to T}(x_s)) \approx x_s$.

$$\begin{aligned} &L_{\text{cyc}}(G_{S\to T}, G_{T\to S}, X_S, X_T) \\ &= E_{x_s\sim X_S}[\| G_{T\to S}(G_{S\to T}(x_s)) - x_s \|_1] \\ &\quad + E_{x_t\sim X_T}[\| G_{S\to T}(G_{T\to S}(x_t)) - x_t \|_1]. \end{aligned} \tag{3}$$

B. Stage II: Full-Level Domain Adaptation

Although we performed image-level domain adaptation in Stage I to obtained a stylized source domain image resembling the target domain's characteristics, there is still a domain shift phenomenon [28]. This is because single-level domain adaptation solely at the image-level alone is not sufficient, feature-level and output-level can provide stronger guidance. Here, we utilize adversarial learning to achieve full-level domain adaptation by inputting target domain stylized source domain image and labels, along with target domain images. Our network encompasses a segmentation network G and two discriminators D_f and D_o. G consists of a feature extractor E and two classifiers C_1 and C_2, and the specific structure is shown in Fig. 3. The feature extractor E uses ResNet 101 as a backbone network, which consists mainly of an input convolutional layer and 33 residual blocks. The classifiers C_1 and C_2 adopt the atrous spatial pyramid pooling structure, and the null rates of the pyramid pooling are 6, 12, 18, and 24, respectively. The discriminators D_f and D_o are composed of four convolutional layers and a domain classifier.

1) Feature-Level Domain Adaptation: Information entropy is a concept in information theory used to measure the uncertainty or amount of information in a set of data. The higher the information entropy, the more uncertain the information and harder it is to predict. The lower the information entropy, the more certain the information and easier it is to predict. Therefore, a trained model will generate predictions with high entropy in the target domain and low entropy in the source domain. To address this discrepancy, we utilize adversarial learning to make the feature entropy distributions of the two domains close to each other, indirectly reducing the entropy of predictions in the target domain, thus achieving feature-level domain adaptation.

Similar to [30], we can obtain high-dimensional features f_s and f_t from the feature extractor E. Self-information, as a component of information entropy, serves to quantify the information of individual events. Its self-information can be defined as $-\log f_s$ and $-\log f_t$. Therefore, we can obtain self-information-weighted pixel-level features $I_s = -f_s \cdot \log f_s$ and $I_t = -f_t \cdot \log f_t$. We introduce the feature discriminator D_f, which takes I_s and I_t as input to perform the domain classification output. In the adversarial learning process, the entropy distributions of f_s and f_t gradually converge, enabling G to learn domain-invariant features and achieve feature-level domain adaptation. The loss calculation formula is as follows

$$L_{adv}^{f}(G, D_f) = -E[\log(D_f(I_s))] - E[\log(1 - D_f(I_t))] \tag{4}$$

2) Output-Level Domain Adaptation: As in [28], previous output-level domain adaptation was to globally align the predictions rather than locally. There is a significant problem with this approach. Because different categories have different domain shifts during domain adaptation, it is possible that certain categories are easy to adapt and certain categories are difficult to adapt. In the pursuit of global alignment, existing local alignment may be destroyed, resulting in negative migration of those classes that are already aligned.

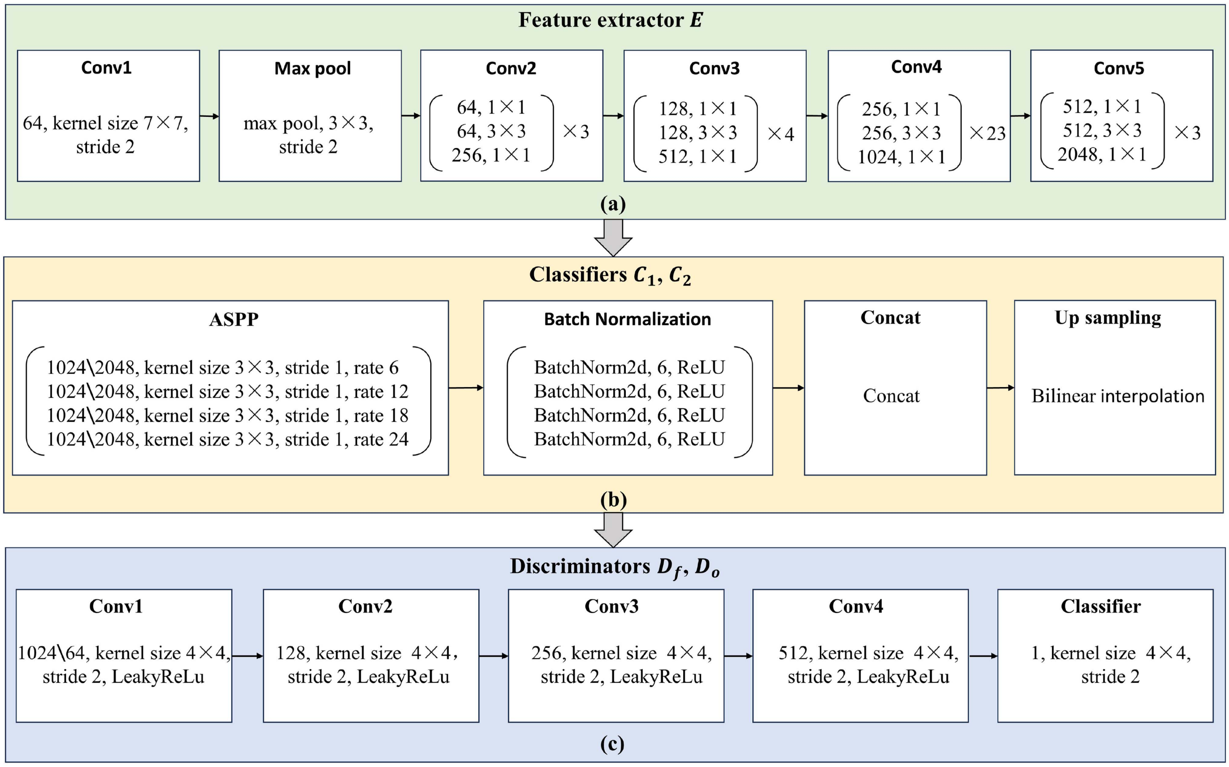


Fig. 3. Structure of the different branches in Stage II. (a) Represents the feature extractor E. (b) Represents the classifiers C_1 and C_2. (c) Represents the discriminators D_f and D_o.

Inspired by [29], at the output-level we use the cotraining algorithm [48] for category-level alignment. In the source-domain data stream, for a source-domain image $x_s \in X_S$, we input the last two layers of the network obtained by the feature extractor E to C_1 and C_2 to produce pixel-level predictions $p_s^{(1)}$ and $p_s^{(2)}$, respectively. Then, introduce the output-level discriminator D_o. Use $p_s^{(2)}$ as input to generate adversarial losses, learning domain invariance. At the same time, $p_s^{(2)}$ is used to calculate segmentation losses with the source domain label $y_s \in Y_S$, guiding the supervised semantic information of the segmentation network G. For segmentation loss, we jointly use soft cross-entropy and dice losses, as shown in (5) and (6). The soft cross-entropy loss employs cross-entropy with label smoothing to improve the model's generalization capability, whereas the Dice loss can alleviate class imbalances to some extent.

$$L_{\text{CE}}(x_s) = -\sum_{i=1}^{H\times W}\sum_{c=1}^{C} y_s^{(i,c)}\log(P_s^{(i,c)}). \tag{5}$$

$$L_{\text{Dice}}(x_s) = 1 - \frac{2\,|Y_s \cap P_s|}{|Y_s| + |P_s|}. \tag{6}$$

In the target domain data stream, we also input a target domain image $x_t \in X_T$ into the segmentation network G to obtain predictions $p_t^{(1)}$ and $p_t^{(2)}$ of the two classifiers.However, the target domain, in contrast to the source domain, lacks label guidance for segmentation loss, and relying on the discriminator alone to perform the global alignment will result in negative migration of some already adapted categories. Therefore, we generate a discrepancy map $M(p_t^{(1)}, p_t^{(2)})$ from $p_t^{(1)}$ and $p_t^{(2)}$, where M represents the cosine distance between the two predictions.Similarly, we input $p_t^{(2)}$ into the discriminator D_o to obtain the adversarial loss map L_{adv}, and then perform an element-by-element multiplication between $M(p_t^{(1)}, p_t^{(2)})$ and L_{adv} to obtain the weighted adversarial loss $\sum_{i=1}^{H}\sum_{i=1}^{W}(1-\cos(p_{i,j}^{(1)}, p_{i,j}^{(2)}))\times L_{\text{adv}_{i,j}}$, where ($i$, j) denotes traversing all pixels in the map. When $M(p_t^{(1)}, p_t^{(2)})$ is large, it indicates that the semantic information of the two prediction results is inconsistent and the category is not yet better adapted between the two domains, so a larger loss weight is assigned to encourage the segmentation network G to deceive the discriminator D_o. When $M(p_t^{(1)}, p_t^{(2)})$ is small, it indicates that the inconsistency of the semantic information of the two prediction results is not a serious problem and the category might have been adaptable, and thus a smaller loss weight is assigned to ignore the adversarial penalty of D_o. With this weighting, our network can focus more on the categories that are difficult to adapt at the output-level, thus achieving category-level alignment. Finally, the adversarial loss at the output-level is calculated as follows:

$$\begin{aligned} L_{\text{adv}}^{o}(G, D_o) &= -E[\log(D_o(G(X_S)))] \\ &\quad - E[(\lambda_m M(P_t^{(1)}, P_t^{(2)}) + \epsilon)\log(1 - D_o(G(X_T)))] \end{aligned} \tag{7}$$

where λ_m is the weight that controls the category-level adversarial loss. In order to stabilize the training process, we include a decimal ϵ in the weights.

In Stage II, we achieve full-level domain adaptation by jointly minimizing all the above losses. That is, the supervised segmentation loss for source domain, the feature-level adversarial loss on both domains, and the output-level adversarial loss. Thus, our Stage II objective function is as follows:

$$L = \min[L_{\text{CE}} + L_{\text{Dice}} + \lambda^{f}_{\text{adv}} L^{f}_{\text{adv}} + \lambda^{o}_{\text{adv}} L^{o}_{\text{adv}}] \tag{8}$$

where λ^{f}_{adv} and λ^{o}_{adv} denote the weights for balancing against losses.

C. Target Domain Self-Training

After full-level domain adaptation, our model FLDA-NET demonstrates better predictive capabilities in the target domain. The model's performance exhibits significant improvements compared to the baseline segmentation model. However, in the absence of available labels for the target domain, this could lead the segmentation model to tend to make overly confident predictions on the source domain. When applied to the target domain, this may result in poor prediction outcomes. ST is a semisupervised learning method that focuses on using existing models to make predictions on unlabeled data, using high-confidence predictions as pseudolabels, and then using the pseudolabels to continue training the model and improve its performance. To achieve this, we suggest a new ST strategy (SPST) based on superpixel segmentation and softmax probability to assign pseudolabels. This approach achieves better performance on the target domain by selecting high-confidence pseudolabels and fine-tuning the segmentation model that is already adapted to the domain.

Superpixel segmentation is an image segmentation technique for grouping pixels with similar color, texture, or luminance characteristics in an image [49]. Initially, we employ the Felzenszwalb superpixel segmentation algorithm [50] to generate a superpixel segmentation result for the target domain image. Then, we compare this result with the predicted results obtained for the target domain to obtain category labels for each superpixel block in each image. Taking a block as an example, we can obtain the one that contains the most category labels within that pixel block. Based on the object-oriented segmentation criterion, we believe that the categories within a pixel block should be consistent. Therefore, we consider this category as a high-confidence category and the rest as low-confidence categories. The number of low-confidence categories is then counted as the error value for that pixel block. The statistics for each superpixel block within an image by this method can obtain the error value of the whole image as in (9). Finally, the error values for each image on the target domain are sorted in ascending order, and we take the top $p\%$ of images with smaller error values as the final samples that can be used for self-training.

$$\text{error} = \sum_{i=1}^{S} N_i - \max\left(\sum_{j=1}^{C} s_{i,j}\right) \tag{9}$$

where S denotes the total number of superpixel blocks of the whole image, C denotes the number of categories, N_i denotes the total number of pixels in block i, and $s_{i,j}$ denotes the number of pixels of category j in block i.

Algorithm 1: Generation of Pseudo-Labels in SPST.

```
Input : The target image $\mathbf{X_t}$, the target predicted
 probability map $\mathbf{P_t}$, the hyperparameters p and $\mu$
Output : The pseudo-labels $\hat{\mathbf{Y}}_\mathbf{t}$
Obtain the result of superpixel segmentation:
  $\mathbf{S_t} = \text{Felzenszwalb}(\mathbf{X_t})$
Obtain the label map: $\mathbf{L_t} = \text{argmax}(\mathbf{P_t})$
Obtain the error value of each image with (9): $\mathbf{X_{error}}$
Sort $\mathbf{X_{error}}$ in ascending order and choose for the top
  $p\%$ of elements as $\mathbf{D_t} = \{x_t, p_t, l_t\}$
for $D_i$ to $D_t$ do
  Initialise to get a pseudo-labels map with the same
  dimensions as $l_i$ and all pixel values of 255: $\hat{y}_i$ Where
  these pixels with a value of 255 will be ignored during
  the training process
  for j = 1 to C do
    Pick predicted probability $p_j$ of class j from $p_i$
    according to $l_i$: $p_j = p_i[\text{where}(l_i == j)]$
    Sort $p_j$ in ascending order and select the value of
    $1 - \mu$ as threshold $T_j$
    Select the label mask: $M_j = \text{where}(p_j > T_j)$
    Generate the pseudo-labels of class j: $\hat{y}_j = l_i[M_j]$
  end
end
return $\hat{\mathbf{Y}}_\mathbf{t}$
```

Although the error values of the prediction results of the selected self-training samples is relatively low, there are still some misclassified pixels in these samples, which may cause the model to learn incorrect knowledge and gradually deviate. To address this problem, we assess the confidence level of each pixel using softmax probability and remove those pixels with lower confidence in the pseudolabels to generate the reliable pseudolabels. Specifically, for each category, we sort the softmax probability values of its predictions in ascending order. μ is a superparameter that defines the proportion of retained pixels (μ is a percentage), and we choose the softmax probability value of the category at $1 - \mu$ as the threshold for that category. Pixels with softmax probability values greater than the threshold are retained, and assign a value of 255 to pixels with less than the threshold, which is ignored in subsequent training. This method ensures that we remove the low-confidence pixels while retaining a sufficient number of pixels. Algorithm 1 shows the whole process.

IV. Experiments and Analyses

In this section, we first introduce the dataset used, the experimental setup, and define the evaluation metrics employed. Then, we compare with other advanced UDA methods to substantiate the efficacy of our proposed approach. Finally, we delve into ablation experiments to showcase the significance of each level in our approach. At the same time, we also discuss the effect of hyperparameters on our method under different settings.

TABLE I
PERCENTAGE OF PIXELS PER CATEGORY IN THE DATASET

Datasets	Percentage of categories					
	Imp surf	Tree	Building	Car	Low veg	Clutter
Potsdam	28.2%	15.4%	26.1%	1.7%	24.6%	4.1%
Vaihingen	28.5%	23.0%	26.4%	1.2%	20.2%	0.8%

A. Datasets Description

1) Vaihingen Dataset: The Vaihingen Dataset (Vaihingen) is a 2-D semantic segmentation dataset for the ISPRS Vaihingen Challenge [51]. It collects an area of 1.38 km^2 of Vaihingen city and consists of 33 3-band IRRG (near-infrared, red, and green) VHR aerial images with a GSD of 9 cm. The average size of each image was approximately 2494 × 2064 [52]. Out of these, only 16 images are provided with semantic labels, consisting of six classes: impervious surfaces (Imp surf), trees (Tree), buildings (Building), cars (Car), low vegetation (Low veg), and clutter (Clutter). We choose these 16 images as the training set, whereas the remaining 17 images as the test set and crop them into 512 × 512 blocks, resulting in 210 training images and 249 test images.

2) Potsdam Dataset: The Potsdam Dataset (Potsdam) is a 2-D semantic segmentation dataset for the ISPRS Potsdam Challenge [53]. It collects an area of 3.42 km^2 of Potsdam city and consists of 38 4-band IRRGB (near-infrared, red, green, and blue) VHR aerial images with a GSD of 5 cm. Each image has a size of 6000 × 6000. Its only 24 images provide semantic labels and it contains the same six classes as the Vaihingen dataset. We selected these 24 images as the training set and the remaining 14 images as the test set. Due to certain classes in the aerial images having specific scale ranges [54], we resample the Potsdam dataset to obtain the same spatial resolution as the Vaihingen dataset. This ensures both datasets maintain consistency in resolution. We select two combinations of bands [IR, R, G] and [R, G, B] from the four bands and cropped them into 512 × 512 blocks, resulting in 813 training images and 482 test images.

Table I displays the two datasets of training data in each category number percentage of the total. Specifically, in the Potsdam training dataset, the percentages of impervious surfaces, trees, buildings, cars, low vegetation, and clutter are 28.2%, 15.4%, 26.1%, 1.7%, 24.6%, and 4.1%, respectively. In the Vaihingen training dataset, they are 28.5%, 23.0%, 26.4%, 1.2%, 20.2%, and 0.8%, respectively. The two datasets are characterized by an imbalance of sample categories. For example, the cars and clutter have significantly lower percentages in both datasets than the other categories. Fig. 4 shows some sample examples for the Potsdam and Vaihingen datasets.

B. Experimental Settings

1) Network Architecture: In Stage I, we follow the architecture of CycleGAN [22] and use instance normalization instead of batch normalization. In Stage II, we use DeepLab-V2 pretrained on the ImageNet [55] dataset with ResNet 101 [56] as the backbone network as the segmentation network, with pyramid-pooled nulls of 6, 12, 18, and 24, respectively. The network structure of the feature-level discriminator and the output-level discriminator is similar to that of [57], consisting of four convolutional layers and a domain classifier, each with convolutional kernel size of 4 and stride size of 2. No batch normalization layer is used. A LeakyReLU [58] activation function with parameter 0.2 is added after each convolutional layer except the last. The specific structure is shown in Fig. 3.

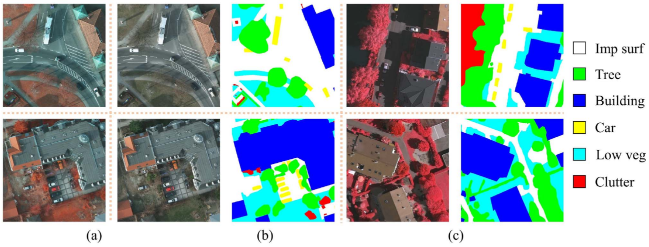


Fig. 4. Sample images and labels for the Potsdam dataset and the Vaihingen dataset. (a) Potsdam dataset consisting of [IR, R, G]. (b) Potsdam dataset consisting of [R, G, B]. (c) Vaihingen dataset consisting of [IR, R, G].

2) Training Details: In this article, our proposed approach was developed using the PyTorch framework [59], which provides an efficient programming interface written in Python, and the experiments were performed on a workstation equipped with an RTX 3090 GPU. The segmentation network is trained using an SGD optimizer [60], whose initial learning rate is set to 2.5×10^{-4}, momentum is 0.9, weight decay is 5×10^{-4}, and learning rate decreases linearly with training times. For the two discriminators, we employed the Adam optimizer with beta parameters set to 0.9 and 0.99. The same learning rate tuning strategy as for the segmentation network is used, the learning rate is set to 1×10^{-4}. The batchsize is set to 2 and the hyperparameters $\lambda_{\mathrm{adv}}^{f}$ and $\lambda_{\mathrm{adv}}^{o}$ are set to 0.001 and 0.01, respectively. In Stage I training, CycleGAN is used to style transform the source domain images, and Stage II uses the style transformed source domain images and labels, as well as the target domain images, as inputs for 50 000 training iterations. In the ST phase, the learning rate is set to 6×10^{-4} and the segmentation model underwent 20 epochs of fine-tuning. Notably, only the target domain data was used to optimize the network parameters during this phase. The hyperparameters p is set to 80 and μ to 0.75. To ensure fairness, the same experimental conditions and settings were used for the other UDA methods compared.

C. Evaluation Metrics

To quantitatively evaluate the model's performance, we employ overall accuracy (OA), mean F1 score (mF1), mean intersection over union (mIoU), and Kappa coefficient (Kappa) as assessment measures. The formulae for each metric and intermediate variable are shown below.

1) Overall Accuracy (OA): OA represents the proportion of samples showing correct classification in all categories and is a comprehensive assessment of the models' overall classification performance, calculated as follows:

$$\mathrm{OA} = \frac{1}{C}\sum_{c=1}^{C}\frac{\mathrm{TP}_c + \mathrm{TN}_c}{\mathrm{TP}_c + \mathrm{TN}_c + \mathrm{FP}_c + \mathrm{FN}_c}. \tag{10}$$

In all formulas, TP indicates how many categories that the model correctly predicted to be positive, TN indicates how many categories that the model correctly predicted to be negative, FP indicates how many negative categories that the model incorrectly predicted to be positive, FN indicates how many positive categories that the model incorrectly predicted to be negative, and C signals the total number of categories in the dataset.

2) Mean F1 Score (mF1): The F1 score combines precision and recall and its mean is calculated as follows:

$$\mathrm{mF1} = \frac{1}{C}\sum_{c=1}^{C}\frac{2 \times \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{11}$$

$$\mathrm{precision} = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c} \tag{12}$$

$$\mathrm{recall} = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}. \tag{13}$$

3) Mean IoU (mIoU): IoU is utilized to gauge the degree of similarity between predicted results and true value labels, and its mean is calculated as follows:

$$\mathrm{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c}. \tag{14}$$

4) Kappa Coefficient (Kappa): The kappa coefficient is a metric derived from the confusion matrix to assess the consistency between the predicted results and the true labels, taking into account chance. It can solve the problem of inaccurate scoring due to sample imbalance, and its calculation formula is as follows:

$$\mathrm{Kappa} = \frac{(\mathrm{OA}) - P_e}{1 - P_e} \tag{15}$$

$$P_e = \frac{\sum_{i=1}^{C}(N_{i,j} \times \sum_{j=1}^{C} N_{i,j})}{N \times N} \tag{16}$$

where N indicates the total pixel count, higher values of the above metrics indicate better model performance.

D. Hyperparameters Sensitivity Analysis

To investigate the impact of different hyperparameters on the overall segmentation performance of the model, we conducted extensive experiments in the scenario of Potsdam (RGB) → Vaihingen (IRRG). We analyzed the sensitivity of the adversarial loss weights λ^f_{adv} and λ^o_{adv}, as well as hyperparameters p and μ in ST.

1) Effect Of Weighting Against Loss: The hyperparameters λ^f_{adv} and λ^o_{adv} represent the weights of the feature-level adversarial loss and the output-level adversarial loss, respectively. To verify the effect of their different settings on the model, hyperparameter tuning experiments are performed based on five parameter sets. Table II shows their effects on the mF1 and IoU of the model. It is evident that the best performance of the model can be obtained by setting λ^f_{adv} to 0.001 and λ^o_{adv} to 0.01. The mF1 and IoU can reach 68.6% and 53.4%, respectively. The performance of our model decreases significantly when a larger weight is given to the feature-level adversarial loss. It is proven that output-level domain adaptation can better align the source and target domains compared with the feature-level. The model also shows some performance degradation when both weights are increased at the same time, which may be caused by giving the adversarial loss greater clout. Significant decreases in mF1 and mIoU also occurred when the weights were too little, indicating the significance of adversarial loss to the model.

TABLE II
EFFECT OF HYPERPARAMETERS λ^f_{ADV} AND λ^o_{ADV} ON MODEL PERFORMANCE FROM POTSDAM TO VAIHINGEN SCENARIOS

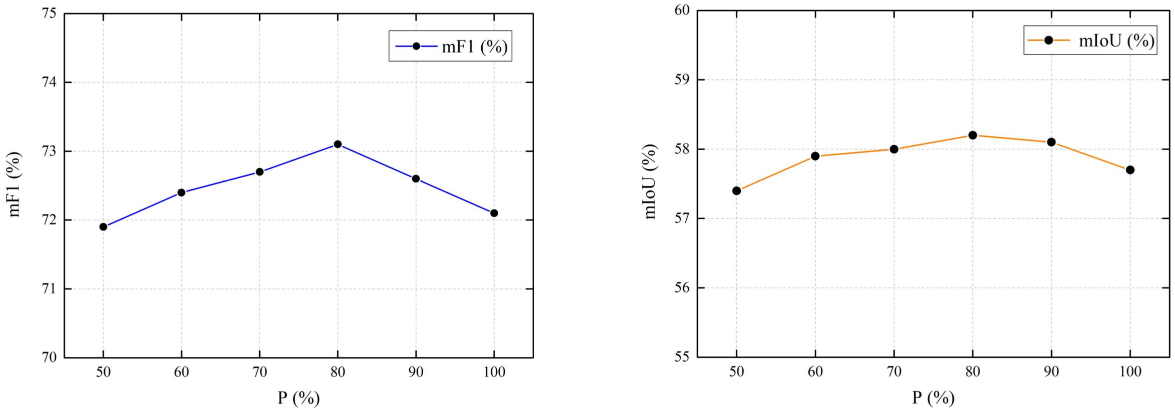
Source: Potsdam (RGB), Target: Vaihingen (IRRG)					
λ^f_{adv} /λ^o_{adv}	0.0001/0.001	0.001/0.001	0.001/0.01	0.01/0.001	0.005/0.05
mF1	65.5	66.9	**68.6**	64.0	66.4
mIoU	50.0	51.6	**53.4**	48.5	50.5

Bold values represent the best values.

2) Sensitivity Analysis Of ST Parameters: The hyperparameters p and μ need to be adjusted during the ST process. Fig. 5 shows the effect of various values of p on the model's performance in terms of mF1 and mIoU metrics. The greatest results for the model is obtained when p is set to 0.8, where mF1 and mIoU can reach 73.1% and 58.9%, respectively. If p is either too small or too large, leads to a decline in the model's effectiveness. This is because if p is taken very small, the pseudolabels generated for the ST process that can be utilized will be very small and may not be sufficient to train. On the contrary, if p is taken very large, the pseudolabels will contain too many incorrect labels, which will cause the performance of the model to deteriorate.

We also need to perform a sensitivity analysis on the hyperparameter μ. If μ is too small, the pseudolabels will retain all the high-confidence pixels, but there may not be enough pixels to train the data in the target domain. Conversely, if μ is too large, the pseudolabels will retain too many wrong pixels. This will mislead the segmentation network during training and cause the model to perform poorly. Therefore, we discuss the hyperparameter μ in Fig. 6. Observing the figure, the prediction confidence decreases very slowly as the pixel ratio increases from 45% to 75%. However, when the pixel ratio increases from 75% to 95%, the prediction confidence decreases rapidly. Based on these analyses, we set the parameter μ to 0.75 to balance between the quantity and quality of pseudolabeled pixels.

To further investigate the sensitivity of the hyperparameter μ, we choose different values of μ to produce pseudolabels for model training, The outcomes of the experiments are given in Table III. The best performance of the model is obtained when we take μ as 75%, and mF1 and mIoU can reach 73.1% and 58.9%. If μ is taken too small or too large, the model's performance decreases, which confirms our analysis above.



(a)

(b)

Fig. 5. Sensitivity analysis of the hyperparameter p in SPST. (a) Effect of p on mF1. (b) Effect of p on mIoU.

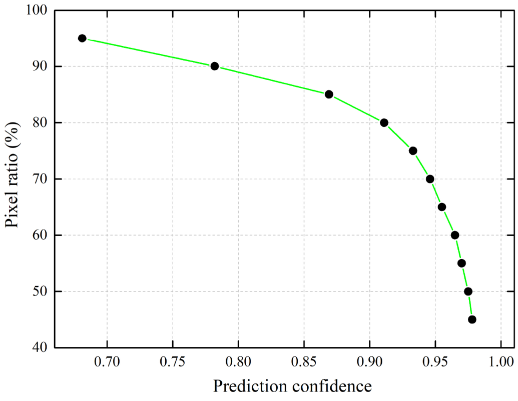


Fig. 6. Relationship between prediction confidence and pixel ratio.

TABLE IV
DIFFERENT DOMAIN ADAPTATION LEVELS USED IN THE COMPARISON METHOD

Method	Image-level	Feature-level	Output-level
CycleGAN [22]	✓	×	×
MinEnt [30]	×	×	×
AdvEnt [30]	×	×	✓
CyCADA [23]	✓	✓	×
AdaptSegNet [28]	×	×	✓
BDL [25]	✓	×	✓
CLAN [29]	×	×	✓
FLDA-NET	✓	✓	✓

TABLE III
EFFECT OF HYPERPARAMETER μ ON MODEL PERFORMANCE FROM POTSDAM TO VAIHINGEN SCENARIOS

Source: Potsdam (RGB), Target: Vaihingen (IRRG)								
μ	60%	65%	70%	75%	80%	85%	90%	100%
mF1	72.1	72.3	72.7	**73.1**	72.8	72.6	72.4	72.2
mIoU	57.9	57.7	58.3	**58.9**	58.5	58.2	58.0	57.8

Bold values represent the best values.

E. Results and Analysis

In this section, we conduct three domain adaptation experiments using the Potsdam and Vaihingen datasets: Potsdam (IRRG) → Vaihingen (IRRG), Potsdam (RGB) → Vaihingen (IRRG), and Vaihingen (IRRG) → Potsdam (RGB). Among them, the first experiment has the same band combination between the two data, and the reasons for the domain shift mainly include the differences in geographical locations, spatial resolution and light intensity, and is mainly aimed at the problem of cross-domain land cover classification in different regions. The latter two experiments have a larger domain shift due to the different bands between the data, which also include differences in the imaging sensors. To confirm our proposed method, we compare it against current state-of-the-art UDA methods. These include the baseline models Deeplab-V2, CycleGAN, MinEnt, AdvEnt, CyCADA, AdaptSegNet, BDL, and CLAN. Table IV summarizes the different level combinations of each UDA method.

1) Potsdam (IRRG) → Vaihingen (IRRG): In this scenario, we perform experiments from the source domain Potsdam (IRRG) to the target domain Vaihingen (IRRG). All comparison methods utilize Deeplab-V2 for the segmentation network, and since CycleGAN is an image style transformation network, we directly use Deeplab-V2 for training on the source domain data after stylization of the target domain.

Table V shows the experimental results for this scenario. The OA, mF1, mIoU, and Kappa of the baseline method are 63.0%, 56.2%, 40.8%, and 50.3%, respectively. Compared with the results of Deeplab-V2, a baseline model trained exclusively using source domain data, all domain adaptation methods were significantly improved. This result suggests that UDA methods can reduce domain bias between different domains to some extent and improve model performance, reflecting the importance of domain adaptation. Notably, our model FLDA-NET

TABLE V
Evaluation Results (%) for All Methods From Potsdam(IRRG) to Vaihingen(IRRG) Scenarios

Source: Potsdam (IRRG) , Target: Vaihingen (IRRG)														
Method	Imp surf		Tree		Building		Car		Low veg		OA	mF1	mIoU	Kappa
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU				
Deeplab-V2(baseline)	64.8	47.9	73.5	58.1	69.1	52.7	36.8	22.6	36.7	22.5	63.0	56.2	40.8	50.3
CycleGAN	66.2	49.5	56.7	39.6	72.0	56.3	58.3	41.1	50.2	33.5	62.7	60.5	44.0	50.4
MinEnt	65.3	48.5	69.4	53.2	66.4	49.7	48.8	32.3	44.4	28.5	62.3	58.9	42.4	49.7
AdvEnt	66.5	49.8	69.9	53.7	73.0	57.5	49.6	33.0	46.7	30.4	64.7	61.1	44.9	53.1
CyCADA	68.2	51.8	67.2	50.1	69.1	52.7	49.1	32.5	49.9	33.2	63.9	60.7	44.2	52.0
AdaptSegNet	69.6	53.4	67.8	51.3	70.9	55.0	53.2	36.3	49.4	32.8	65.1	62.2	45.8	53.5
BDL	73.2	57.7	72.6	57.0	75.7	60.9	56.0	38.9	53.7	36.7	69.6	66.2	50.2	59.4
CLAN	72.3	56.6	70.8	54.8	75.5	60.7	56.6	39.4	53.1	36.2	68.4	65.7	49.5	58.0
FLDA-NET	82.2	69.8	69.4	53.2	86.3	75.9	65.1	48.2	65.6	48.8	76.6	73.7	59.2	68.9
FLDA-NET+SPST	**84.1**	**72.5**	**78.1**	**64.1**	**88.0**	**78.5**	**66.8**	**50.2**	**70.0**	**53.9**	**80.5**	**77.4**	**63.8**	**74.1**
Supervised	88.8	79.9	86.4	76.0	92.6	86.2	80.5	67.4	77.8	63.7	86.9	85.2	74.6	82.6

Bold values are the best values.

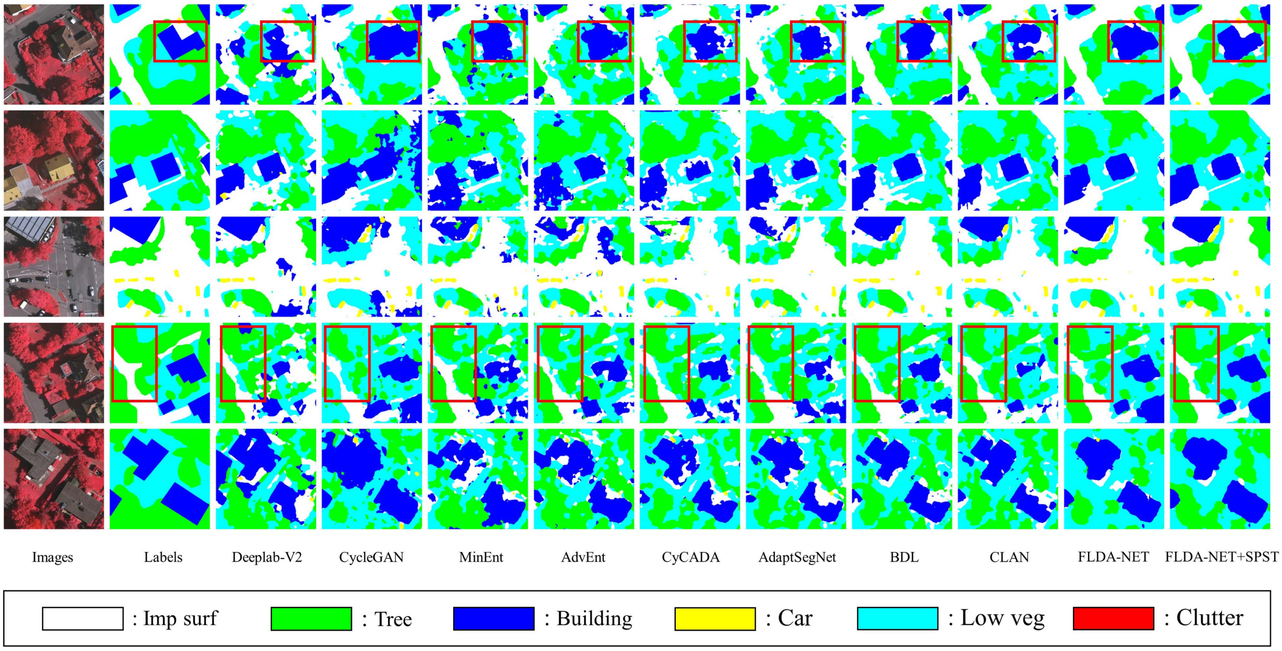


Fig. 7. Representative results for all methods from Potsdam (IRRG) to Vaihingen (IRRG) scenarios. White, green, blue, yellow, cyan, and red represent Imp surf, Tree, Building, Car, Low veg, and Clutter, respectively. Clutter is not involved in training.

outperforms other UDA methods in all four evaluation metrics, and our best model FLDA-NET+SPST can achieve OA, mF1, mIoU, and Kappa of 80.5%, 77.4%, 63.8%, and 74.1%, respectively. Compared with the baseline model, it improves by 17.5%, 21.2%, 23%, and 23.8%, respectively. Compared with the best performing BDL, our model improves OA, mF1, mIoU, and Kappa by 10.9%, 11.2%, 13.6%, and 14.7%, respectively. From a single category perspective, our approach adjusts the adaptation degree to different categories and achieves the highest F1 and IoU. Each category has increased by 10.9%, 5.5%, 12.3%, 8.5%, and 16.3% compared with the second place in F1. In terms of IoU, the improvements compared with the second place are 14.8%, 7.1%, 17.6%, 9.1%, and 17.2%, respectively.

Fig. 7 shows representative results of our method and various UDA methods in the Potsdam (IRRG) to Vaihingen (IRRG) scenarios. In the illustration, white, green, blue, yellow, cyan, and red regions represent impervious surfaces, trees, buildings, cars, low vegetation, and clutter, respectively. Among them, the results of the baseline model Deeplab-V2 show significant differences compared with the labels. After applying the UDA method, the segmentation performance has been improved to varying degrees. Our method FLDA-NET gives the best visual results, with the presence of fewer missed and wrong segments and higher completeness.

2) Potsdam (RGB) → Vaihingen (IRRG): In this scenario, we perform experiments from the source domain Potsdam (RGB)

TABLE VI
EVALUATION RESULTS (%) FOR ALL METHODS FROM POTSDAM(RGB) TO VAIHINGEN(IRRG) SCENARIOS

Source: Potsdam (RGB) , Target: Vaihingen (IRRG)														
Method	Imp surf		Tree		Building		Car		Low veg		OA	mF1	mIoU	Kappa
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU				
Deeplab-V2(baseline)	23.2	13.1	47.2	30.8	64.7	47.8	22.7	12.8	38.4	23.8	45.0	39.2	25.7	27.1
CycleGAN	49.4	32.8	30.8	18.2	62.9	45.9	36.5	22.3	37.3	22.9	47.6	43.4	28.4	30.4
MinEnt	50.4	33.7	46.7	30.5	58.6	41.4	47.0	30.7	37.9	23.4	49.8	48.1	31.9	32.9
AdvEnt	58.1	40.9	63.7	46.7	73.2	57.7	50.8	34.1	41.2	25.9	59.5	57.4	41.1	46.1
CyCADA	58.8	41.6	59.3	42.2	70.8	54.8	52.1	35.2	38.7	24.0	57.8	55.9	39.6	43.9
AdaptSegNet	61.8	44.7	69.3	53.0	76.3	61.7	52.3	35.4	47.3	31.0	64.1	61.4	45.2	52.4
BDL	63.7	46.7	66.8	50.1	77.0	62.6	48.8	32.2	48.2	31.8	64.0	60.9	44.7	52.3
CLAN	63.9	46.9	71.6	55.8	73.1	57.6	49.7	33.1	49.0	32.5	64.5	61.5	45.2	52.9
FLDA-NET	73.6	58.3	67.1	50.5	86.8	76.6	63.1	46.1	52.6	35.7	70.8	68.6	53.4	61.3
FLDA-NET+SPST	**75.8**	**61.0**	**76.5**	**61.9**	**90.1**	**82.0**	**67.6**	**51.1**	**55.5**	**38.4**	**75.1**	**73.1**	**58.9**	**67.0**
Supervised	88.8	79.9	86.4	76.0	92.6	86.2	80.5	67.4	77.8	63.7	86.9	85.2	74.6	82.6

Bold values are the best values.

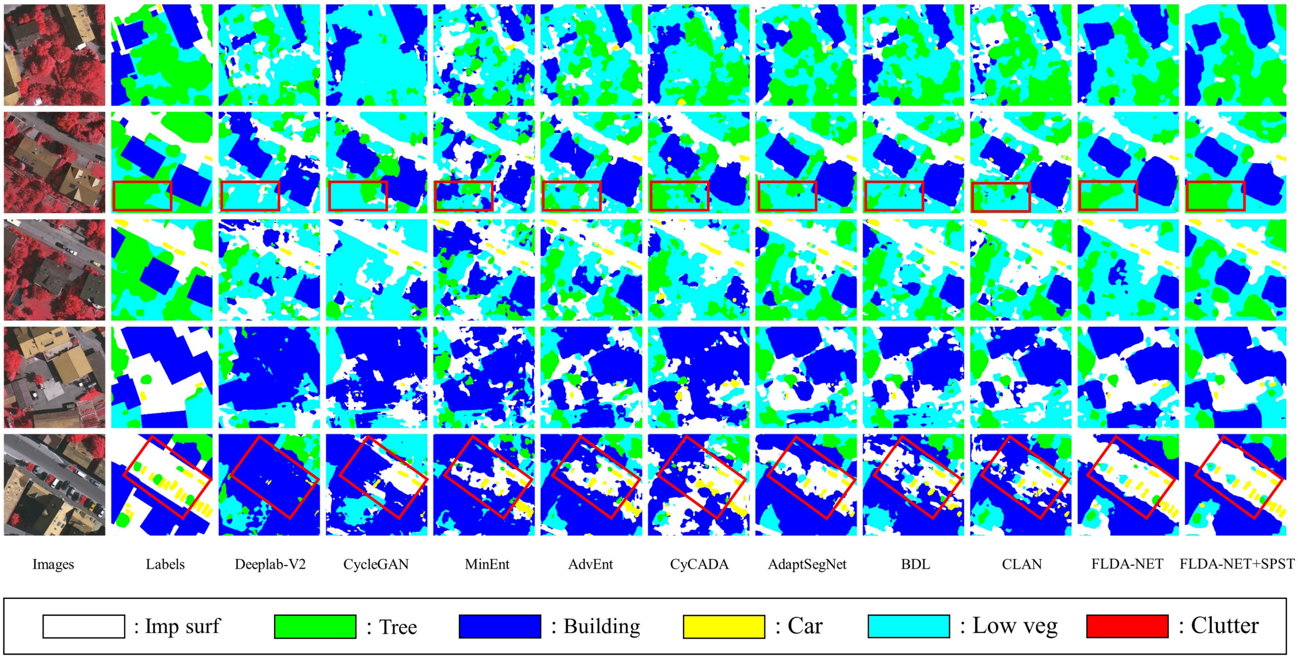


Fig. 8. Representative results for all methods from Potsdam (RGB) to Vaihingen (IRRG) scenarios. White, green, blue, yellow, cyan, and red represent Imp surf, Tree, Building, Car, Low veg, and Clutter, respectively. Clutter is not involved in training.

to the target domain Vaihingen (IRRG). The specific experimental results are detailed in Table VI, where the OA, mF1, mIoU, and Kappa of the baseline method can only reach 45.0%, 39.2%, 25.7%, and 27.1%, respectively. Our best model FLDA-NET+SPST can achieve OA, mF1, mIoU and Kappa of 75.1%, 73.1%, 58.9% and 67.0%, respectively. Compared with the baseline model, it improves by 30.1%, 33.9%, 33.2% and 39.9%, respectively. However, compared with the fully supervised training, there is still a certain gap between our method and it. This is due to the large domain difference between the datasets in terms of imaging sensors, geographical locations, spatial resolutions, etc. Our model surpasses all other UDA methods in achieving the highest scores for all evaluation metrics. Compared with CLAN, which has the highest performance, our model improves by 10.6%, 11.6%, 13.7%, and 14.1% for OA, mF1, mIoU, and Kappa, respectively. Also, our model achieves the highest F1 and IoU in each category. These advantages indicate that our approach performs well in the cross-domain land cover classification from Potsdam (RGB) to Vaihingen (IRRG).

Fig. 8 displays representative results of our method alongside other UDA methods for this scene. The baseline model Deeplab-V2 suffers from a severe domain shift problem and produces poor results due to the fact that this scenario has a larger domain shift compared to the previous one. This is shown

TABLE VII
Evaluation Results (%) for All Methods From Vaihingen(IRRG) to Potsdam(RGB) Scenarios

Source: Vaihingen (IRRG), Target: Potsdam (RGB)														
Method	Imp surf		Tree		Building		Car		Low veg		OA	mF1	mIoU	Kappa
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU				
Deeplab-V2(baseline)	26.6	15.3	16.4	8.9	41.1	25.8	62.7	45.7	15.0	8.1	32.1	32.3	20.8	10.0
CycleGAN	49.7	33.0	35.8	21.8	52.7	35.8	50.7	33.9	9.0	4.7	44.2	39.6	25.9	24.1
MinEnt	57.4	40.3	45.3	29.3	43.7	28.0	49.4	32.8	20.1	11.2	44.4	43.2	28.3	25.3
AdvEnt	57.7	40.5	48.7	32.2	44.3	28.4	49.6	33.0	16.5	9.0	44.7	43.4	28.6	26.0
CyCADA	61.0	43.9	55.4	38.3	40.7	25.5	55.6	38.5	10.9	5.7	46.3	44.7	30.4	28.3
AdaptSegNet	61.7	44.8	56.1	39.0	40.8	25.6	53.4	36.4	14.1	7.6	45.9	45.3	30.7	27.7
BDL	66.4	49.7	**58.1**	**41.0**	48.3	31.9	68.5	52.0	8.8	4.6	49.8	50.0	35.8	33.2
CLAN	60.4	43.2	43.3	27.6	42.5	27.0	71.2	55.3	15.4	8.3	44.5	46.5	32.3	25.2
FLDA-NET	71.5	55.7	56.5	39.4	82.1	69.6	63.1	46.1	34.3	20.7	63.7	61.5	46.3	51.8
FLDA-NET+SPST	**72.0**	**56.2**	52.2	35.4	**85.4**	**74.5**	**72.2**	**56.5**	**41.7**	**26.3**	**65.4**	**64.7**	**49.8**	**53.9**
Supervised	91.6	84.4	84.6	73.3	95.9	92.1	90.0	81.9	85.1	74.1	89.9	89.4	81.2	86.5

Bold values are the best values.

by the many noisy points in the figure, together with large areas of misprediction. After domain adaptation, this problem was largely mitigated. However, some single-level UDA methods still have quite a lot of noisy segmentation, such as CycleGAN and AdvEnt. Compared to them, our method obtains better prediction results. Especially, the categories of building and car have more complete and smooth edge parts.

3) Vaihingen (IRRG) → Potsdam (RGB): In order to verify the credibility of our methods, we conducted another experiment from the source domain Vaihingen (IRRG) to the target domain Potsdam (RGB). Similarly, all comparison methods utilize Deeplab-V2 for the segmentation network. Table VII describes the results of the performance comparison between our method and other UDA methods. The results clearly illustrate that the performance of the baseline model Deeplab-V2 is very poor. OA, mF1, mIoU, and Kappa can only reach 32.1%, 32.3%, 20.8%, and 10.0%, respectively. Our method also excels with the highest OA, mF1, mIoU, and Kappa of 65.4%, 64.7%, 49.8%, and 53.9%, respectively. Compared with the baseline model, it improves by 33.3%, 32.4%, 29.0%, and 43.9%, respectively. Even compared with the second ranked BDL, our method improves by 15.6%, 14.7%, 14.0%, and 20.7% on the four evaluation metrics, respectively. Meanwhile, on most of the categories, our method achieves the highest scores for both F1 and IoU. However, our method is slightly lower than BDL in the tree category, due to the strong similarity between the tree and low vegetation categories, which can easily be confused when performing cross-domain land cover classification. However, our method can achieve 41.7% and 26.3% F1 and IoU for low vegetation. However, BDL can only achieve 8.8% and 4.6%, which proves that the method does not effectively discriminate between these two categories, but only classifies most of them into the tree category. On the other hand, our method can reconcile these two categories to some extent.

Fig. 9 shows representative results of our method and various UDA methods in the Vaihingen (IRRG) to Potsdam (RGB) scenarios. As can be observed that not only the baseline model but also the other methods show a lot of noise segmentation and wrong segmentation. This is because, compared with the previous scenario, the Vaihingen dataset as the source domain has fewer training samples and provides less guidance in terms of segmentation loss, resulting in some single-level or dual-level methods not being able to align the source domain and the target domain well. However, our approach successfully attenuates the domain difference between the two datasets and shows good segmentation results. This is further evidence that our approach can still achieve the goal of cross-domain land cover classification with fewer samples.

V. Discussion

Due to the relative complexity of our model approach, it is necessary to more thoroughly assess and analyze the effectiveness of each level in our model. We conducted extensive experiments in the Potsdam (RGB) → Vaihingen (IRRG) scenarios. Simultaneously, we also conducted a complexity analysis of different UDA methods, comprehensively comparing the relationship between evaluation results and computational efficiency of different methods.

A. Each Level of Effectiveness

In our proposed method, the cross-domain land cover classification is accomplished by aligning the source and target domains at the image, feature, and output levels. To demonstrate the efficacy of each level, we conducted extensive ablation experiments. Table VIII shows the evaluation results of mF1 and mIoU across different levels. Where, no adaptation represents the baseline model, which trains the segmentation network using only the source domain data. Stage I (IL) represents Stage I image-level domain adaptation. Stage II (FL) and Stage II (OL) represent Stage II feature-level and output-level domain adaptation, respectively. SPST represents our proposed self-training strategy. The table shows that the model performs poorly when we do not use the UDA method. When we use only image-level domain adaptation, there is some enhancement in contrast to the baseline model. When we use the output-level domain adaptation,

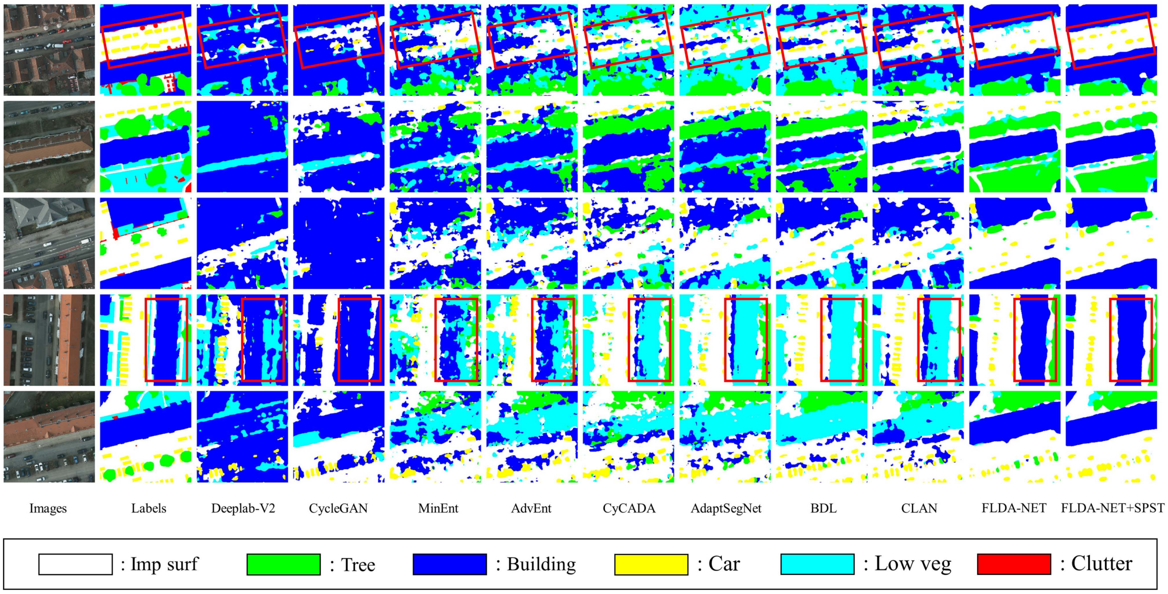


Fig. 9. Representative results for all methods from Vaihingen (IRRG) to Potsdam (RGB) scenarios. White, green, blue, yellow, cyan, and red represent Imp surf, Tree, Building, Car, Low veg, and Clutter, respectively. Clutter is not involved in training.

TABLE VIII
Impact of Different Level Combinations From Potsdam to Vaihingen Scenarios

Source: Potsdam (RGB), Target: Vaihingen (IRRG)							
Method	No adaptation	Stage I (IL)	Stage II (FL)	Stage II (OL)	SPST	mF1	mIoU
Baseline	✓					39.2	25.7
FLDA-NET		✓				43.4	28.4
				✓		66.3	51.3
		✓		✓		67.1	51.9
		✓	✓	✓		68.6	53.4
		✓	✓	✓	✓	**73.1**	**58.9**

"No Adaptation" represents the baseline approach without using any domain adaptation. "Stage I (IL)" represents Stage I image-level domain adaptation. "Stage II (FL)" represents Stage II feature-level domain adaptation. "Stage II (OL)" represents Stage II output-level domain adaptation.
Bold values represent the best values.

we observe that the model performance increases significantly. Both mF1 and mIoU are improved by 22.9% compared with image-level domain adaptation. Thus, it may be concluded that output-level domain adaptation proves to be more effective in aligning the source and target domains.

By comparing Stage I (IL) + Stage II (OL) and Stage II(OL), we can see that dual-level alignment achieves better results compared with single-level alignment. By further increasing the feature-level alignment, our full-level domain adaptation approach achieves the best performance with mF1 and mIoU of 68.6% and 53.4%, respectively. Compared with Stage I (IL) + Stage II (OL), there is a 1.5% improvement in both mF1 and mIoU. Finally, by adding the SPST self-training strategy and fine-tuning the already domain-adapted segmentation model through the utilization of high-confidence pseudolabels in the target domain, the model shows better segmentation on the target domain, and the mF1 and IoU can reach 73.1% and 58.9%, respectively.

B. Complexity Analysis

We performed a complexity analysis of different UDA methods in the Potsdam (RGB) → Vaihingen (IRRG) scenario and comprehensively compared the relationship between evaluation results and computational efficiency of different methods. All experiments were performed on a single NVIDIA RTX 3090 GPU. Table IX provides the computational complexity, number of parameters, training time, and mF1 evaluation results for each method. It can be seen that MinEnt has the lowest computational complexity, number of model parameters, and training time for the model due to the fact that it only uses a segmentation network and does not introduce a discriminator compared with the other methods. However, its scores for mF1 is also the lowest, proving that adversarial learning has good results for cross-domain land cover classification. The other methods have increased computational complexity, number of model parameters, and training time due to the introduction of discriminators using adversarial learning. Our method FLDA-NET, employing

TABLE IX
Complexity Analysis of Different Methods

Source: Potsdam (RGB), Target: Vaihingen (IRRG)				
Method	FLOPs(G)	Params(M)	Time(h)	mF1
MinEnt	**186.12**	**43.16**	**2.92**	48.1
AdvEnt	199.81	48.70	3.34	57.4
CyCADA	191.04	50.85	6.57	55.9
AdaptSegNet	197.94	48.26	5.42	61.4
BDL	192.03	45.71	5.67	60.9
CLAN	193.90	45.93	5.58	61.5
FLDA-NET	280.78	106.92	10.15	68.6
FLDA-NET+SPST	280.78	106.92	10.49	**73.1**

Bold values represent the best values.

a two-stage training mode and introducing two discriminators, thus imposes higher requirements on computational complexity, number of model parameters, and training time compared with other methods. However, our method achieves the highest mF1 at the cost of sacrificing complexity. In addition, our method FLDA-NET+SPST further enhances model performance without increasing the computational complexity and the number of model parameters, achieving the highest mF1 of 73.1%.

VI. Conclusion

We introduce a full-level domain adaptation approach called FLDA-NET to solve the challenge of cross-domain land cover classification in RS images. The method enhances the performance of cross-domain land cover classification by aligning the distribution of source and target domain data at the image-level, feature-level, and output-level in two stages. To improve the classifier's discriminative capabilities for each land cover category in the target domain, we present a ST method called SPST. This method is based on superpixel segmentation and softmax probability to assign pseudolabels. These pseudolabels are then utilized to enhance the classification performance through applying them to the proposed FLDA-NET. We conducted two-way experiments on the Potsdam and Vaihingen datasets, and experiment results show that the method outperforms currently available UDA methods. We also conducted a series of ablation experiments to validate the efficacy of each level in our proposed method.

However, our proposed method is relatively complex and requires multiple stages, as well as numerous hyperparameters. In our future research, we plan to develop an end-to-end approach to enhance the performance of cross-domain land cover classification for RS images. In addition, we will explore strategies for automating the selection of hyperparameter weights to further increase the model algorithm's automation level.

References

[1] J. Cihlar, "Land cover mapping of large areas from satellites: Status and research priorities," *Int. J. Remote Sens.*, vol. 21, no. 6/7, pp. 1093–1114, 2000.

[2] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sens. Environ.*, vol. 80, no. 1, pp. 185–201, 2002.

[3] R. G. Congalton, J. Gu, K. Yadav, P. Thenkabail, and M. Ozdogan, "Global land cover mapping: A review and uncertainty analysis," *Remote Sens.*, vol. 6, no. 12, pp. 12 070–12 093, 2014.

[4] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, May/Jun. 1991.

[5] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Netw.*, vol. 17, no. 1, pp. 113–126, 2004.

[6] J. Lafferty et al., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. Int. Conf. Mach. Learn.*, 2001, pp. 282–289.

[7] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, 2005.

[8] M. Punia, P. K. Joshi, and M. Porwal, "Decision tree classification of land use land cover for Delhi, India using IRS-P6 AWiFS data," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5577–5583, 2011.

[9] M. Pal and P. M. Mather, "A comparison of decision tree and backpropagation neural network classifiers for land use classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2002, pp. 503–505.

[10] J. Wang, M. Zhang, W. Li, and R. Tao, "A multistage information complementary fusion network based on flexible-mixup for HSI-X image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 14, 2023, doi: 10.1109/TNNLS.2023.3300903.

[11] M. Zhang, W. Li, X. Zhao, H. Liu, R. Tao, and Q. Du, "Morphological transformation and spatial-logical aggregation for tree species classification using hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5501212.

[12] W. Li, J. Wang, Y. Gao, M. Zhang, R. Tao, and B. Zhang, "Graph-feature-enhanced selective assignment network for hyperspectral and multispectral data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5526914.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.

[15] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[16] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[17] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[18] S. B.-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, pp. 151–175, 2010.

[19] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[20] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.

[21] J. Zhu, Y. Guo, G. Sun, L. Yang, M. Deng, and J. Chen, "Unsupervised domain adaptation semantic segmentation of high-resolution remote sensing imagery with invariant domain-level prototype memory," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5603518.

[22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

[23] J. Hoffman et al., "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1989–1998.

[24] Z. Wu et al., "DCAN: Dual channel-wise alignment networks for unsupervised scene adaptation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 518–534.

[25] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6929–6938.

[26] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "FCNs in the wild: Pixel-level adversarial and constraint-based adaptation," 2016, *arXiv:1612.02649*.

[27] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. F. Wang, and M. Sun, "No more discrimination: Cross city adaptation of road scene segmenters supplementary material," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2010–2020.

[28] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7472–7481.

[29] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2502–2511.

[30] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2512–2521.

[31] D. V. Sang and N. D. Minh, "Fully residual convolutional neural networks for aerial image segmentation," in *Proc. 9th Int. Symp. Inf. Commun. Technol.*, 2018, pp. 289–296.

[32] D. Wittich and F. Rottensteiner, "Appearance based deep domain adaptation for the classification of aerial images," *ISPRS J. Photogrammetry Remote Sens.*, vol. 180, pp. 82–102, 2021.

[33] L. Zhang, M. Lan, J. Zhang, and D. Tao, "Stagewise unsupervised domain adaptation with adversarial self-training for road segmentation of remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5609413.

[34] Y. Li, T. Shi, Y. Zhang, W. Chen, Z. Wang, and H. Li, "Learning deep semantic segmentation network under multiple weakly-supervised constraints for cross-domain remote sensing image semantic segmentation," *ISPRS J. Photogrammetry Remote Sens.*, vol. 175, pp. 20–33, 2021.

[35] J. Chen, J. Zhu, Y. Guo, G. Sun, Y. Zhang, and M. Deng, "Unsupervised domain adaptation for semantic segmentation of high-resolution remote sensing imagery driven by category-certainty attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5616915.

[36] L. Ma, M. Li, X. Ma, L. Cheng, P. Du, and Y. Liu, "A review of supervised object-based land-cover image classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 130, pp. 277–293, 2017.

[37] R. Adayel, Y. Bazi, H. Alhichri, and N. Alajlan, "Deep open-set domain adaptation for cross-scene classification based on adversarial learning and Pareto ranking," *Remote Sens.*, vol. 12, no. 11, 2020, Art. no. 1716.

[38] W. Liu and F. Su, "A novel unsupervised adversarial domain adaptation network for remotely sensed scene classification," *Int. J. Remote Sens.*, vol. 41, no. 16, pp. 6099–6116, 2020.

[39] B. Benjdira, Y. Bazi, A. Koubaa, and K. Ouni, "Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images," *Remote Sens.*, vol. 11, no. 11, 2019, Art. no. 1369.

[40] L. Shi, Z. Wang, B. Pan, and Z. Shi, "An end-to-end network for remote sensing imagery semantic segmentation via joint pixel-and representation-level domain adaptation," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 11, pp. 1896–1900, Nov. 2021.

[41] X. Guo et al., "GAN-based virtual-to-real image translation for urban scene semantic segmentation," *Neurocomputing*, vol. 394, pp. 127–135, 2020.

[42] N. Makkar, L. Yang, and S. Prasad, "Adversarial learning based discriminative domain adaptation for geospatial image analysis," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 150–162, 2022.

[43] J. Liu and Y. Wang, "Water body extraction in remote sensing imagery using domain adaptation-based network embedding selective self-attention and multi-scale feature fusion," *Remote Sens.*, vol. 14, no. 15, 2022, Art. no. 3538.

[44] P. Shamsolmoali, M. Zareapoor, H. Zhou, R. Wang, and J. Yang, "Road segmentation for remote sensing images using adversarial spatial pyramid networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4673–4688, Jun. 2021.

[45] L. Yan, B. Fan, H. Liu, C. Huo, S. Xiang, and C. Pan, "Triplet adversarial domain adaptation for pixel-level classification of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3558–3573, May 2020.

[46] S. Ji, D. Wang, and M. Luo, "Generative adversarial network-based full-space domain adaptation for land cover classification from multiple-source remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 3816–3828, May 2021.

[47] D. Peng, H. Guan, Y. Zang, and L. Bruzzone, "Full-level domain adaptation for building extraction in very-high-resolution optical remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5607317.

[48] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.

[49] F. He, M. P. Mahmud, A. Z. Kouzani, A. Anwar, F. Jiang, and S. H. Ling, "An improved SLIC algorithm for segmentation of microscopic cell images," *Biomed. Signal Process. Control*, vol. 73, 2022, Art. no. 103464.

[50] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, pp. 167–181, 2004.

[51] "International society for photogrammetry and remote sensing 2D semantic labeling - Vaihingen data." Accessed: Feb. 23, 2024. [Online]. Available: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx

[52] L. Wang, R. Li, C. Duan, C. Zhang, X. Meng, and S. Fang, "A novel transformer based semantic segmentation scheme for fine-resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Feb. 21, 2024, Art. no. 6506105.

[53] "International society for photogrammetry and remote sensing 2D semantic labeling contest - Potsdam." Accessed: Feb. 21, 2024. [Online]. Available: https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx

[54] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogrammetry Remote Sens.*, vol. 145, pp. 3–22, 2018.

[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. F.-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[57] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5967–5976.

[58] A. L. Maas et al., "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1–5.

[59] "Pytorch." Accessed: 2024. [Online]. Available: https://pytorch.org

[60] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

Yuke Meng received the B.E. degree in surveying and mapping engineering from the Henan University of Engineering, Zhengzhou, China, in 2021. He is currently working toward the M.E. degree in surveying and mapping engineering with the School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo, China.

His research interests include computer vision, remote sensing image processing, and deep learning.

Zhanliang Yuan received the Ph.D. degree in surveying and mapping from the Henan Polytechnic University, Jiaozuo, China, in 2015.

He is a Visiting Scholar with the University of New South Wales, Kensington, NSW, Australia, and is currently a Professor with Henan Polytechnic University, Jiaozuo, China. His research interests include smart cities, remote sensing data analysis, and 3-D visualization.

Jian Yang received the Ph.D. degree in cartography and geography information system from Institute of Remote Sensing Application, Chinese Academy of Sciences, Beijing, China, in 2008.

He is currently a Professor with Aerospace Information Research Institute, Chinese Academy of Sciences. His main research interests include remotely sensing imagery processing and analyzing algorithms on high spatial resolution RS image, including multi-feature extraction, multisource data fusion, land cover classification, and change detection of urban region.

Peizhuo Liu received the B.E. and M.S. degrees in instrument science and technology from the Harbin Institute of Technology, Harbin, China, in 2013 and 2015, respectively. He is currently working toward the doctorate degree in vision measurement and remote sensing with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing, China.

His research interests include deep learning, self-supervised learning, semisupervised learning, remote sensing, and semantic segmentation.

Jian Yan received the B.E. degree in engineering of automobile from Tsinghua University, Beijing, China, in 2019. He is currently working toward the doctoral degree in quantitative remote sensing with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing.

He is a Visiting Ph.D. Student with DISI, University of Trento, Trento, Italy. His research interests include multimodal remote sensing image processing, land cover classification, and artificial intelligence.

Hongbo Zhu received the B.E. degree in spatial information and digital technology from the North China Institute of Aerospace Engineering, LangFang, China, in 2021. He is currently working toward the M.E degree in aeronautical and astronautical science and technology with the school of remote sensing and information engineering, North China Institute of Aerospace Engineering langfang, China.

His research interests include sensing image processing, deep learning, and land cover classification.

Zhigao Ma received the B.E. degree in surveying and mapping engineering from Henan Polytechnic University, Jiaozuo, China, in 2021. He is currently working toward the M.E. degree in surveying and mapping engineering with the School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo, China.

His research interests include remote sensing image processing and point cloud deep learning.

Zhenzhao Jiang received the B.E. degree in surveying and mapping engineering from the Henan University of Urban Construction, Pingdingshan, China, in 2019. He is currently working toward the M.E. degree in surveying and mapping engineering with the School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo, China.

His research interests include remote sensing image processing, deep learning and change detection.

Zhouwei Zhang received the M.S. degree in aerospace engineering from Aerospace Engineering, Dresden University of Technology, Dresden, Germany, in 2006, and the Ph.D. degree in cartography and GIS from Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, in 2011.

His research interests include quantum computing, remote sensing image processing, and deep learning.

Xiaofei Mi received the Ph.D. degree in signal and information processing from Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China, in 2020.

She is currently an Assistant Professor with National Engineering Laboratory of Remote Sensing Satellite Applications, Beijing. Her research interests include land cover classification and change detection, including image processing and remote sensing application.