AdaptMatch: Adaptive Matching for Semisupervised Binary Segmentation of Remote Sensing Images

Wei Huang, Yilei Shi, *Member, IEEE*, Zhitong Xiong, *Member, IEEE*, and Xiao Xiang Zhu, *Fellow, IEEE*

***Abstract*— There are various binary semantic segmentation tasks in remote sensing (RS) that aim to extract the foreground areas of interest, such as buildings and roads, from the background in satellite images. In particular, semisupervised learning (SSL), which can use limited labeled data to guide a large amount of unlabeled data for model training, can significantly promote the fast applications of these tasks in practice. However, due to the predominance of the background in RS images, the foreground only accounts for a small proportion of the pixels. It poses a challenge: models are biased toward the majority class of the background, leading to poor performance on the minority class of the foreground. To address this issue, this article proposes a novel and effective SSL framework, adaptive matching (AdaptMatch), for RS binary segmentation. AdaptMatch calculates individual and adaptive thresholds of the foreground and background based on their convergence difficulty in an online manner at the training stage; the adaptive thresholds are then used to select the high-confidence pseudo-labeled data of the two classes for model self-training in turn. Extensive experiments are conducted on two widely studied RS binary segmentation tasks, building footprint extraction and road extraction, to demonstrate the effectiveness and generalizability of the proposed method. The results show that the proposed AdaptMatch achieves superior performance compared with some state-of-the-art semisupervised methods in RS binary segmentation tasks. The codes will be publicly available at https://github.com/zhu-xlab/AdaptMatch.**

***Index Terms*— Adaptive threshold, binary segmentation, building footprint extraction, remote sensing (RS), road extraction, semisupervised learning (SSL).**

I. Introduction

IN THE remote sensing (RS) fields, there are various binary semantic segmentation tasks that aim to extract the foreground regions of interest, such as building footprints [1], [2], roads [3], [4], changes [5], [6], [7], [8], [9], and landslides [10], [11], from the background in satellite or aerial images. These tasks have a broad range of practical applications in urban planning [12], [13], hazard assessment [14], [15], and environmental monitoring [16], [17], [18]. However, they demand a substantial volume of numerous manually-labeled data at the pixel level, which is laborious and hard to obtain in practice. From this perspective, semisupervised learning (SSL), especially semisupervised semantic segmentation (SSS), can significantly promote the speed of these tasks because it can use only a few labeled data to guide a large number of unlabeled data for model training, thereby reducing the heavy dependence on annotations.

The core of SSS is its approach to utilizing unlabeled data for model training. The different methods for doing so can be broadly categorized into two technical routes. One is the consistency-based method [19], [20], [21], [22], [23], [24]. This approach aims to enforce the prediction agreement between the original images/features/model and the perturbed counterparts, which can promote models to learn robust feature representation and predictions that are free of noise and perturbation. The other is the self-training-based method [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38]. This type of method tries to assign the unlabeled data pseudo-labels and use them for pseudo-supervised training on models, where typically thresholds are set to only allow high-confidence unlabeled data for training to reduce the impact of wrong pseudo-labels.

Although the above SSS methods can provide some promising paradigms for semisupervised binary segmentation of RS images, they ignore the **imbalanced distribution** of the foreground and background. Typically, the foreground of interest only occupies a small proportion of the entire image, while all the remaining areas are considered background. The imbalanced distribution limits the direct application of these SSS methods, especially the self-training-based methods, in RS binary segmentation tasks. Moreover, there is a phenomenon of *confirmation bias* [24] in SSL, in which incorrect pseudo-labels of unlabeled data can be confirmed and memorized by models and the model segmentation performance then decreases after being trained by these wrong pseudo-labels. Unfortunately, imbalanced distribution further exacerbates the *confirmation bias* problem and thus degrades model performance significantly.

To alleviate the problem, in this article, we propose an adaptive matching (AdaptMatch) framework for semisupervised RS binary segmentation tasks, which aims to create relatively

Manuscript received 30 June 2023; revised 23 August 2023 and 24 September 2023; accepted 9 October 2023. Date of publication 28 November 2023; date of current version 29 November 2023. This work was supported in part by the German Research Foundation (DFG GZ: ZH 498/18-1) under Project 519016653; in part by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV) Based on a Resolution of the German Bundestag (EKAPEx) under Grant 67KI32002B; and in part by the German Federal Ministry of Education and Research (BMBF) in the Framework of the International Future AI Laboratory "AI4EO—Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond" under Grant 01DD20001. *(Corresponding author: Xiao Xiang Zhu.)*

Wei Huang, Zhitong Xiong, and Xiao Xiang Zhu are with the Chair of Data Science in Earth Observation, Technical University of Munich, 80333 Munich, Germany (e-mail: w2wei.huang@tum.de; zhitong.xiong@tum.de; xiaoxiang.zhu@tum.de).

Yilei Shi is with the School of Engineering and Design, Technical University of Munich (TUM), 80333 Munich, Germany (e-mail: yilei.shi@tum.de).

Digital Object Identifier 10.1109/TGRS.2023.3332490

balanced self-training on models between the foreground and background based on their convergence difficulty during the training stage. There are three parts to AdaptMatch: 1) supervised learning, which only uses labeled data for supervised model training; 2) pseudo-supervised learning, which further selects the high-confidence unlabeled data by thresholds for model self-training; and 3) an adaptive threshold mechanism, which calculates the adaptive and individual thresholds of the foreground and background in an online manner for more accurate and fine-grained self-training of the unlabeled data.

Of these three parts, pseudo-supervised learning and adaptive threshold mechanism are semisupervised. Specifically, the high-confidence pseudo-labels of weakly augmented RS images are used to supervise the segmentation of strongly augmented counterparts, i.e., pseudo-supervised training, based on FixMatch [34]. This approach integrates the benefits of both the self-training of high-confidence pseudo-labeled data and the consistency learning between weakly and strongly augmented images. However, the thresholds of different classes are fixed as the same value, typically 0.95, which is not suitable for imbalanced RS semisupervised binary segmentation. To address this issue, the adaptive threshold mechanism tries to calculate adaptive thresholds for more balanced pseudo-supervised learning by designing a novel strategy of class-wise prediction accumulation and convergence-difficulty calculation from both the labeled and the unlabeled training data. The calculated thresholds have three advantages over the fixed ones. First, the thresholds of the foreground and background are individual, which allows a class-wise selection of pseudo-labels of the unlabeled data. Second, the thresholds are dynamic at different training stages based on the convergence difficulty of the two classes, so they can adjust more precisely for different datasets in different training states. Third, AdaptMatch does not depend on any particular model and can be easily combined with various advanced segmentation models including both convolutional neural network (CNN) and vision transformer (ViT).

To evaluate the effectiveness and generalizability of the proposed AdaptMatch, extensive experiments are conducted on two widely studied RS binary segmentation tasks, building footprint extraction and road extraction. In comparison with other state-of-the-art SSS methods, AdaptMatch shows superior and more robust performance on several widely used datasets, including two building footprint datasets, Inria [39] and Massachusetts, and two road datasets, WHU_Roads and DeepGlobe_Roads. The comparison experiment results verify the superiority and robustness of the proposed method in semisupervised RS binary segmentation.

The contributions of this article can be summarized in two main areas, as follows.

1) We develop a novel SSL framework for RS binary segmentation, AdaptMatch, to alleviate the imbalanced distribution between foreground and background of RS images, which limits the effective self-training of unlabeled data. AdaptMatch calculates individual thresholds of the foreground and background based on their convergence difficulties during training. This allows it to adaptively select relatively balanced class-wise pseudo-labels and achieve more reasonable self-training of unlabeled data.
2) The proposed AdaptMatch is a model-agnostic optimization mechanism that can be easily combined with various models. Besides, it achieves superior results in two widely used RS binary segmentation tasks in the semisupervised setting when compared with some state-of-the-art SSS methods.

II. Related Works

In this section, we briefly review some related works from two perspectives: SSL and RS semisupervised semantic/binary segmentation.

A. Semisupervised Learning

SSL is a hot topic in image processing because it can largely reduce the dependence on pixel-level labeled data, which is labor-intensive and time-consuming. As mentioned in Section I, there are two main types of semisupervised methods, as discussed below.

1) Self-training-based methods [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [38]. Lee et al. [26] applied the pseudo-labels of unlabeled data to deep neural networks for model pseudo-supervised training, in a departure from the supervised training of labeled data in earlier eras. MixMatch [27] generates low-entropy labels from multiple data-augmented unlabeled examples and then mixes the labeled and unlabeled data for co-training. To get better pseudo labels, meta pseudo label, which utilizes a teacher network to generate pseudo-labels of unlabeled data to teach a student network, is proposed [28]. From the perspective of the long-tailed class distribution, He et al. [32] designed an effective distribution alignment and random sampling (DARS) strategy to produce unbiased pseudo-labels matching the true class distribution, Guan et al. [33] proposed an unbiased subclass regularization network (USRN) to alleviate the class imbalance problem by learning class-unbiased segmentation from balanced subclass distributions. In addition, FixMatch [34] predicts the pseudo-label from a weakly augmented image and uses it to supervise the classification of a strongly augmented version of the same image; here, it is worth noting that a fixed threshold is used to select high-confidence unlabeled samples for training. To get more accurate pseudo-labels, some works aim to generate dynamic thresholds of different classes for semisupervised image classification, notably FlexMatch [36] and SoftMatch [35]. Recently, UniMatch [37] highlighted the FixMatch in SSS with new state-of-the-art results.

2) Consistency-based methods [19], [20], [21], [22], [23], [24]. These methods aim to guide the models to learn feature representation and probability prediction free of perturbations. Here the perturbation varies and these methods can be applied at different levels, including images [19], features [20], and even models [21], [22]. Cross-consistency training (CCT) is proposed in [20] for SSS, which enforces the consistency of the predictions among the clear features and different types of perturbed features. A novel consistency regularization approach, cross pseudo supervision (CPS) [21] is designed to

impose consistency on two segmentation networks perturbed with different initialization but with the same architecture for the same input image; subsequently, a new conflict-based cross-view consistency (CCVC) method [22] aimed at enforcing the two heterogeneous subnets to learn consistency features from irrelevant views by introducing a feature discrepancy loss. To address the prediction accuracy problem of consistency learning methods, Liu et al. [23] extend the mean-teacher (MT) model with a new auxiliary teacher, and replace MT's mean square error (mse) with a stricter confidence-weighted cross-entropy (Conf-CE) loss. Finally, a redesign of pseudo-labeling is proposed in [19] to generate well-calibrated structured pseudo-labels with unlabeled or weakly labeled data for consistency training.

Besides, some works focus on cross-domain information transfer to reduce the demand for target domain labels. For example, the integration of graph information extraction in few-shot learning shows excellent domain adaptive performance in [40]. A domain generalization framework for hyperspectral images is designed to break through the limitations of traditional domain adaptive techniques in [41].

B. RS Semisupervised Binary Segmentation

Many researchers have also explored SSL in RS semantic/binary segmentation tasks in the past few years. Sun et al. [42] devised a boundary-aware SSS network, which integrates the channel-weighted multiscale feature module that balances semantic and spatial information and the boundary attention module, which weights the features with rich semantic boundary. Wang et al. [43] introduced a consistency regularization training method for RS SSS and employ the newly learned model for an average update of pseudo-label (AUP); Wang et al. [44] subsequently introduced cross pseudo-supervision into RS semantic segmentation and optimize it in an alternative manner. Zhang et al. [45] designed a transformation consistency regularization method to encourage consistent predictions under different random spatial transformations or perturbations, including rotation, patch shuffle, and CutMix [46], [47]. Zhang et al. proposed a feature and prediction alignment method in [48] and joint self-training [49] and rebalanced consistency learning for semisupervised change detection. Desai and Ghose [50] proposed an active learning-based sampling strategy to select high-representation data for land cover classification. Consistency learning has also been widely attempted as a regularization in change detection [51], semantic segmentation [52], and building footprint extraction.

The mainstream RS semisupervised semantic/binary segmentation methods are based on consistency learning. Unlike these methods, the proposed AdaptMatch is based on self-training and aims to alleviate the imbalance problem during training. Compared with consistency-based methods, self-training-based methods have their advantage and disadvantages. The advantage is that self-training can learn more discriminative predictions via the pseudo-supervision of two opposite pseudo-labels (foreground and background); in contrast, consistency-based methods mainly focus on consistent possibility predictions, which does not contribute to prediction discrimination as much as self-training. However, self-training-based methods usually suffer from some inevitable wrong pseudo-labels during training. Particularly, in RS binary segmentation, the imbalanced distribution of foreground and background exacerbates the pseudo-label misassignment from the minor foreground to the dominant background. The proposed AadptMatch can achieve relatively balanced self-training of unlabeled data via class-aware thresholds. As a result, AdaptMatch can decrease the negative impact of wrong pseudo-labels while increasing the prediction discrimination.

III. AdaptMatch-Based Semisupervised Binary Segmentation

In this section, some notations of semisupervised binary segmentation are given. Then, the shared segmentation model is introduced in brief. Finally, AdaptMatch is introduced in detail, with the whole workflow shown in Fig. 1.

A. Notations

In the SSL setting, there are two subsets of the training data: a limited-labeled training set $\mathcal{D}^l$ and an unlabeled set $\mathcal{D}^u$. Their sample sets are denoted as $\mathcal{D}^l = \{(\mathbf{x}^l, \mathbf{y}^l)\}_{i=1}^{N_l}$, and $\mathcal{D}^u = \{(\mathbf{x}^u)\}_{i=1}^{N_u}$, respectively, where $\mathbf{x}$, $\mathbf{y}$, and N are an image, its pixel-wise labels, and the sample number of its set, respectively. Here N_l is much smaller than N_u, that is, there are significantly fewer limited-labeled data than unlabeled data. The available labels y^l and unavailable labels $\mathbf{y}^u$ share the same class space $\{0, 1\}$, where 0 is the background and 1 is the foreground. In AdaptMatch, there are two individual thresholds of the foreground and background, denoted as $\{\tau^F, \tau^B\}$, for the adaptive pseudo-labeling of unlabeled data. The calculation of $\{\tau^F, \tau^B\}$ is based on the historical predictions of both the labeled and unlabeled data, and therefore two corresponding memory banks, denoted as $\{\mathcal{M}^F, \mathcal{M}^B\}$, are used to store these historical predictions. The core of this article focuses on utilizing the unlabeled data $\mathcal{D}^u$ effectively for model self-training at a balanced ratio of the foreground and background via the adaptive thresholds $\{\tau^F, \tau^B\}$.

B. Shared Segmentation Model

The core of semisupervised segmentation (SSS) is to design an efficient strategy to utilize unlabeled data for model training, which is free of particular model architectures. Starting from this principle, the proposed method can be combined with various neural networks and significantly boost their performance, as verified in Section IV-C. To make a fair evaluation, the comparison experiments among the proposed method and other SSS methods are carried out on the same encoder–decoder architecture, i.e., SegFormer_B2. In detail, a semantic segmentation model consists of an encoder $\mathcal{E}$ and a decoder $\mathcal{G}$. $\mathcal{E}$ is used to extract a high-level semantic feature map $\mathbf{f} \in \mathbb{R}^{H/s \times W/s \times C}$ from a given image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, where $[H, W]$ is the spatial size and s is the spatial scale ratio determined by certain segmentation models; $\mathcal{G}$ is used to

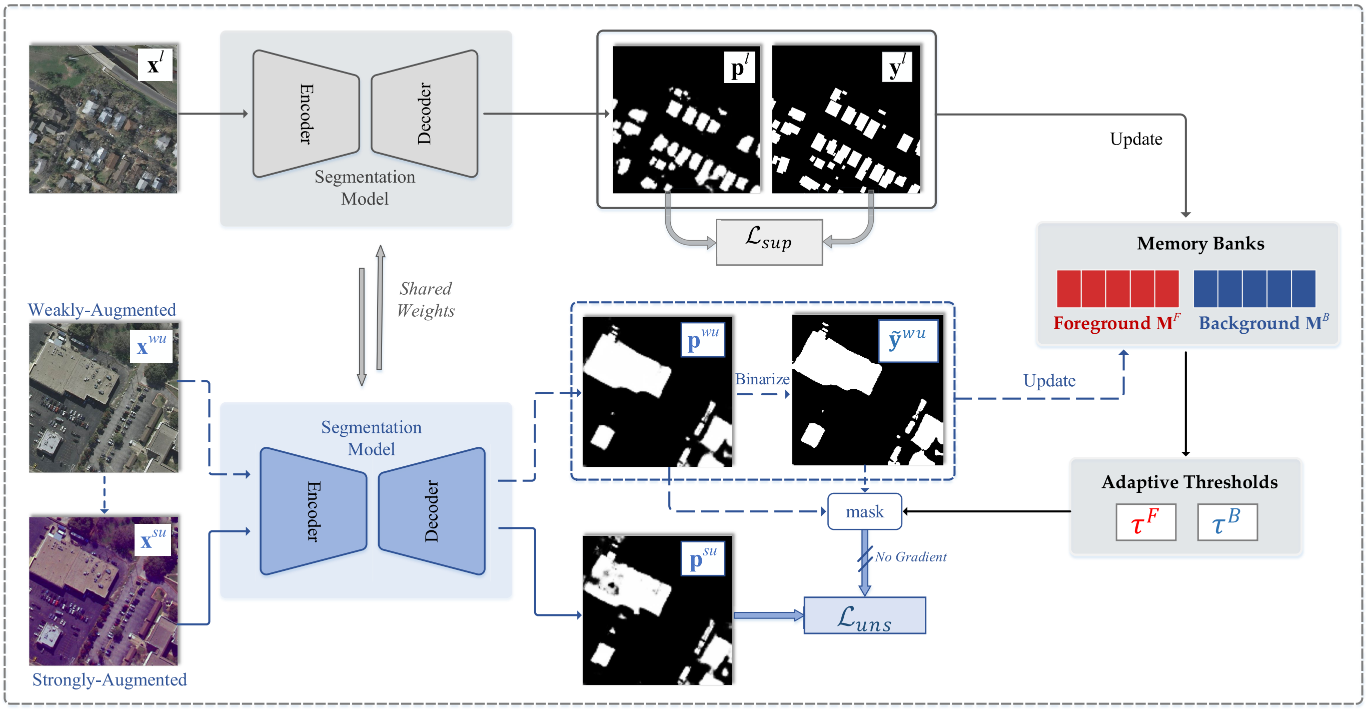


Fig. 1. Workflow of the proposed AdaptMatch for RS semisupervised binary segmentation. There are three branches for joint supervised and pseudo-supervised training on the segmentation model: 1) a labeled branch that uses labeled data for supervised training; 2) a weakly augmented unlabeled branch that generates trustworthy pseudo-labels from weakly augmented unlabeled data, selected by adaptive thresholds; and 3) a strongly augmented unlabeled branch for pseudo-supervised training that enforces predictions of the strongly augmented unlabeled data agreeing with the pseudo-labels generated from (2).

make a binary prediction map $\mathbf{p} \in \mathbb{R}^{H\times W}$ according to $\mathbf{f}$. The two sequential steps are denoted as

$$\mathbf{f} = \mathcal{E}(\mathbf{x}) \tag{1}$$

$$\mathbf{p} = \mathcal{G}(\mathbf{f}) \tag{2}$$

where the encoder $\mathcal{E}$ and the decoder $\mathcal{G}$ are shared between the labeled data sampled from $\mathcal{L}$ and the unlabeled data sampled from $\mathcal{U}$. In this article, both the CNN and ViT architectures have been used as the segmentation model.

To scale the prediction of each pixel of $\mathbf{p}$ into the range of [0,1], the sigmoid operation is applied to each pixel along the class dimension as

$$\mathbf{p}(i,j) = \text{sigmoid}(\mathbf{p}(i,j)) = \frac{1}{1+e^{-\mathbf{p}(i,j)}} \tag{3}$$

where $[i,j]$ is the spatial location of $\mathbf{p}$ and e is the Euler's number.

C. AdaptMatch

As shown in Fig. 1, at each training iteration there are three branches trained at the same time in the AdaptMatch-based semisupervised binary segmentation framework: 1) the labeled branch, which uses labeled data for supervision training; 2) the weakly augmented unlabeled branch, which produces relatively stable predictions and high-confidence pseudo-labels of unlabeled data, which are selected by adaptive thresholds $\{\tau^F, \tau^B\}$; and 3) the strongly augmented unlabeled branch, which generates the strongly augmented predictions of unlabeled data for pseudo-supervision training with the selected pseudo-labels of the weakly augmented unlabeled branch. The adaptive thresholds $\{\tau^F, \tau^B\}$ are calculated from the historical predicted possibilities of both the labeled branch and the weakly augmented unlabeled branch, which are stored in two corresponding memory banks $\{\mathcal{M}^F, \mathcal{M}^B\}$. For clear description, each of $\{\mathcal{M}^F, \mathcal{M}^B\}$ is split into two parts, the labeled part $\mathcal{M}^l$ and the unlabeled part $\mathcal{M}^u$. In other words, $\{\mathcal{M}^F, \mathcal{M}^B\} = \{\mathcal{M}^{lF} \cup \mathcal{M}^{uF}, \mathcal{M}^{lB} \cup \mathcal{M}^{uB}\}$.

1) Labeled Branch: For an image-label pair $\{\mathbf{x}^l, \mathbf{y}^l\}$ sampled from the labeled set $\mathcal{D}^l$, following the fully-supervised setting, the image $\mathbf{x}^l \in \mathbb{R}^{H\times W\times 3}$ is weakly augmented and fed into the shared segmentation model, and then the pixel-wise prediction map $\mathbf{p}^l \in \mathbb{R}^{H\times W}$ is obtained by (1)–(3) as

$$\mathbf{p}^l = \mathcal{G}(\mathcal{E}(\mathbf{x}^l)) \tag{4}$$

$$\mathbf{p}^l(i,j) = \text{sigmoid}(\mathbf{p}^l(i,j)) = \frac{1}{1+e^{-\mathbf{p}^l(i,j)}}. \tag{5}$$

For RS binary segmentation tasks, the labeled data are used for training the model through the use of two kinds of supervised losses, binary cross-entropy (BCE) loss and the intersection of union (IoU, also called *Jaccard*) loss [4], [53], as

$$\begin{aligned}\mathcal{L}^l_{\text{BCE}} &= \frac{1}{\text{HW}}\sum_{i=1}^{H}\sum_{j=1}^{W}\text{BCE}(\mathbf{y}^l(i,j), \mathbf{p}^l(i,j))\\ &= \frac{1}{\text{HW}}\sum_{i=1}^{H}\sum_{j=1}^{W}(-\mathbf{y}^l(i,j)\log\mathbf{p}^l(i,j)\\ &\quad + (1-\mathbf{y}^l(i,j))\log(1-\mathbf{p}^l(i,j)))\end{aligned} \tag{6}$$

$$\begin{cases} \mathcal{L}^l_{\text{IoU}} = 1 - \dfrac{|\mathbf{p}^l \cap \mathbf{y}^l| + \epsilon}{|\mathbf{p}^l| + |\mathbf{y}^l| - |\mathbf{p}^l \cap \mathbf{y}^l| + \epsilon} \\ |\mathbf{p}^l \cap \mathbf{y}^l| = \displaystyle\sum_{i=1}^{H}\sum_{j=1}^{W} \mathbf{p}^l(i,j)\mathbf{y}^l(i,j) \\ |\mathbf{p}^l| = \displaystyle\sum_{i=1}^{H}\sum_{j=1}^{W} \mathbf{p}^l(i,j) \\ |\mathbf{y}^l| = \displaystyle\sum_{i=1}^{H}\sum_{j=1}^{W} \mathbf{y}^l(i,j) \end{cases} \tag{7}$$

where ϵ is a "smooth" parameter to avoid the illegal division operation when $|\mathbf{p}^l| = |\mathbf{y}^l| = 0$, with its value set to 1.

The whole supervised loss is the combination of the BCE loss and the IoU loss, as

$$\mathcal{L}_{\text{sup}} = \mathcal{L}^l_{\text{BCE}} + \mathcal{L}^l_{\text{IoU}}. \tag{8}$$

Apart from supervising model training, the labeled data plays another role: updating the labeled part of memory banks, $\{\mathcal{M}^{\text{lF}}, \mathcal{M}^{\text{lB}}\}$. At iteration b, to reduce the storage burden of the memory banks, the probability prediction map $\mathbf{p}^l_b$ and the label map $\mathbf{y}^l_b$ are resized to 64 × 64 along the dims of spatial height and width. For each pixel at location $[i, j]$, its probability prediction, $\mathbf{p}^l_b(i, j)$, is updated into either $\mathcal{M}^{\text{lF}}$ or $\mathcal{M}^{\text{lB}}$ according to its label $\mathbf{y}^l_b(i, j)$, as

$$\begin{cases} \mathcal{M}^{\text{lF}} = \mathcal{M}^{\text{lF}} \cup \{\mathbf{p}^l_b(i,j)\}, & \text{if} \quad \mathbf{y}^l_b(i,j) = 1 \\ \mathcal{M}^{\text{lB}} = \mathcal{M}^{\text{lB}} \cup \{\mathbf{p}^l_b(i,j)\}, & \text{otherwise}. \end{cases} \tag{9}$$

Here, 1 in the first equation is the class index of the foreground. There are two levels of storage units in the memory banks: pixel and set. Each set stores all the pixels of the corresponding class at each iteration during training; $\mathcal{M}^{\text{lF}}$ and $\mathcal{M}^{\text{lB}}$ have a maximum number of sets, $N^l_{\text{iter}} = 100$, to control the frequency at which they are updated. At the beginning of training when the set length is smaller than N^l_{iter}, the new set is directly stored in the associated memory bank; then when the set number reaches N^l_{iter}, the oldest set is deleted from the memory bank and the newest one is added, that is, first-in-first-out (FIFO). This updated process ensures that the stored possibility predictions always come from the latest 500 iterations, which contain class-wise global information at the dataset level. Unlike the fixed length of sets, the pixel number of each set varies with the labels of the labeled data.

2) Weakly Augmented Unlabeled Branch: In this branch, the unlabeled image $\mathbf{x}^u$ is sampled from the unlabeled set $\mathcal{D}^u$, and is weakly augmented to $\mathbf{x}^{\text{wu}}$. After being fed into the segmentation model by (1)–(3), its prediction map $\mathbf{p}^{\text{wu}}$ can be obtained as

$$\mathbf{p}^{\text{wu}} = \text{sigmoid}(\mathcal{G}(\mathcal{E}(\mathbf{x}^{\text{wu}}))). \tag{10}$$

Since that there are no labels of $\mathbf{x}^{\text{wu}}$, we use its pseudo-label map $\tilde{\mathbf{y}}^{\text{wu}} \in \mathbb{R}^{H\times W}$ for the follow-up pseudo-supervision segmentation of the strongly augmented unlabeled image. Here, $\tilde{\mathbf{y}}^{\text{wu}}$ is generated from $\mathbf{p}^{\text{wu}}$ as

$$\tilde{\mathbf{y}}^u(i,j) = \begin{cases} 1, & \text{if} \quad \mathbf{p}^{\text{wu}}(i,j) > 0.5 \\ 0, & \text{else}. \end{cases} \tag{11}$$

Similar to labeled data in (9), the prediction map of $\mathbf{x}^{\text{wu}}$, i.e., $\mathbf{p}^{\text{wu}}$, is also used to update the unlabeled part of memory banks, $\{\mathcal{M}^{\text{uF}}, \mathcal{M}^{\text{uB}}\}$. Each pixel of $\mathbf{p}^{\text{wu}}$ is updated into either $\mathcal{M}^{\text{uF}}$ or $\mathcal{M}^{\text{uF}}$ according to its pseudo-label as

$$\begin{cases} \mathcal{M}^{\text{uF}} = \mathcal{M}^{\text{uF}} \cup \{\mathbf{p}^{\text{wu}}_b(i,j)\}, & \text{if} \quad \tilde{\mathbf{y}}^u_b(i,j) = 1 \\ \mathcal{M}^{\text{uB}} = \mathcal{M}^{\text{uB}} \cup \{\mathbf{p}^{\text{wu}}_b(i,j)\}, & \text{otherwise}. \end{cases} \tag{12}$$

Here, there is also a maximum iteration number, $N^u_{\text{iter}} = 3 \times N^l_{\text{iter}} = 300$, to control the frequency with which the unlabeled memory banks $\mathcal{M}^{\text{uF}}$ and $\mathcal{M}^{\text{uB}}$ are cleared and reupdated.

3) Strongly Augmented Unlabeled Branch: The weakly augmented unlabeled image $\mathbf{x}^{\text{wu}}$ is further augmented by some strong augmentations to $\mathbf{x}^{\text{su}}$ as

$$\mathbf{x}^{\text{su}} = \mathcal{A}(\mathbf{x}^{\text{wu}}) \tag{13}$$

where $\mathcal{A}$ denotes two connected strong augmentations sampled from an intensity augmentation list whose details are provided in Section IV-A. Its prediction map $\mathbf{p}^{\text{su}}$ can also be obtained via (1)–(3) as

$$\mathbf{p}^{\text{su}} = \text{sigmoid}(\mathcal{G}(\mathcal{E}(\mathbf{x}^{su}))). \tag{14}$$

Up to now the pseudo-label map $\tilde{\mathbf{y}}^{\text{wu}}$ and the prediction map $\mathbf{p}^{\text{su}}$ of the unlabeled image $\mathbf{x}^u$ are available; there is only the mask map $\mathbf{m}^{\text{wu}} \in R^{H\times W}$ left to select the high-confidence pixels for self-training. The mask map $\mathbf{m}^{\text{wu}}$ is determined by the thresholds $\{\tau^F, \tau^B\}$ that are calculated every iteration as

$$\begin{cases} \tau^F = \dfrac{\sum_{p\in\mathcal{M}^F} p}{\sum_{p\in\mathcal{M}^F}} = \dfrac{\sum_{p\in\mathcal{M}^{\text{lF}}\cup\mathcal{M}^{\text{uF}}} p}{\sum_{p\in\mathcal{M}^{\text{lF}}\cup\mathcal{M}^{\text{uF}}}} \\ \tau^B = \dfrac{\sum_{p\in\mathcal{M}^B} p}{\sum_{p\in\mathcal{M}^B}} = \dfrac{\sum_{p\in\mathcal{M}^{\text{lB}}\cup\mathcal{M}^{\text{uB}}} p}{\sum_{p\in\mathcal{M}^{\text{lB}}\cup\mathcal{M}^{\text{uB}}}} \end{cases} \tag{15}$$

which means the threshold of each class is the average value of all the predictions of this class within previous N^l_{iter} iterations for the labeled set and previous N^u_{iter} iterations for the unlabeled set. The average prediction of each class, i.e., its mean confidence, can reveal its convergence difficulty in real-time, and therefore it has the ability to serve as the corresponding threshold.

Then, the mask map $\mathbf{m}^{\text{wu}}$ can be calculated as

$$\mathbf{m}^{\text{wu}}(i,j) = \begin{cases} 1, & \text{if} \quad \mathbf{p}^{\text{wu}}(i,j) > \tau^F \quad \text{or} \quad \mathbf{p}^{\text{wu}}(i,j) < \tau^B \\ 0, & \text{otherwise}. \end{cases} \tag{16}$$

Finally, the unsupervised loss of unlabeled data can be calculated from the pseudo-label map $\tilde{\mathbf{y}}^{\text{wu}}$, the prediction map $\mathbf{p}^{\text{su}}$, and the mask map $\mathbf{m}^{\text{wu}}$ as

$$\mathcal{L}_{\text{uns}} = \frac{1}{\text{HW}} \sum_{i=1}^{H}\sum_{j=1}^{W} \text{BCE}(\mathbf{p}^{\text{su}}(i,j), \tilde{\mathbf{y}}^{\text{wu}}(i,j)) \cdot \mathbf{m}^u(i,j) \tag{17}$$

where the detailed operation of BCE is the same as (6). To avoid unstable self-training of unlabeled data at the training beginning, at the first $N^l_{\text{iter}} = 100$ iterations $mathcalL_{\text{uns}}$ is set to zero. This equation shows that the binarized predictions of the weakly augmented image are used as the pseudo-labels to supervise the training of the strongly augmented image

TABLE I

Detailed Characteristics of Four Used Datasets

Dataset Characteristics	Building Footprints		Roads	
	Inria	Massachusetts	WHU_Roads	DeepGlobe_Roads
Sensor Source	Orthorectified Imagery	—	Gaofen-II, ZY-III, and WorldView-II	WorldView-3, WorldView-2, and GeoEye-1
Image Mode	RGB	RGB	RGB	RGB
Included Cities	Austin, Chicago, Kitsap (USA) West Tyrol, Vienna (Europe)	Boston (USA)	Liuzhou (China)	Sampled over Thailand, Indonesia, and India
Spatial Resolution	0.3 m/pixel	1 m/pixel	0.8–2 m/pixel	0.5 m/pixel
Original Size	5000×5000	1500×1500	512×512	1024×1024
Cropped Size	512×512	512×512	512×512	512×512
Number of Original Tiles	36 (each city)	151	6,828	6,226
Number of Cropped Images	2,520 (each city)	1,359	6,828	24,904
Train:Validation:Test	7:1:2	137:4:10 (Original)	7:1:2	7:1:2
Labeled: Unlabeled Training Patches	25:2,495 (1% labeled) 126:2,394 (5% labeled) 504:2,016 (20% labeled)	— 61:1,172 (5% labeled) 246:987 (20% labeled)	44:4,429 (1% labeled) 223:4,250 (5% labeled) 894:3,579 (20% labeled)	176:17,260 (1% labeled) 868:16,564 (5% labeled) 3,484:13,948 (20% labeled)

because the weakly augmented image has fewer perturbations and thereby its pseudo-labels are more trustworthy. The mask map $\mathbf{m}^{\text{wu}}$ can adaptively filter out some class-aware low-confidence pseudo-labels for the biased model and reduce their negative interference during training. It is worth noting that the gradient of the weakly augmented unlabeled branch is stopped, which is mainly used to obtain the "ground-truth" (pseudo-labels); in contrast, the gradient of the strongly augmented unlabeled branch is normally generated for self-training.

4) Overall Loss and Training Procedure: The overall loss function is the combination of the supervised loss $\mathcal{L}_{\text{sup}}$ in (8) and the unsupervised loss $\mathcal{L}_{\text{uns}}$ in (17) as

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{uns}}. \tag{18}$$

During the training stage, $\mathcal{L}_{\text{sup}}$ uses all the labeled data to train the model for binary segmentation. On top of it, $\mathcal{L}_{\text{uns}}$ can further enhance the model performance by self-training with class-wise high-confidence pseudo-labels derived from unlabeled data, which are selected by the proposed method at a balanced ratio. As a result, the combination of all the labeled data and the class-balanced high-confidence pseudo-labeled data derived from unlabeled data can effectively train the model.

To better describe the pipeline of the proposed AdaptMatch for semisupervised binary segmentation of RS images, its training procedure is summarized in Algorithm 1.

Algorithm 1 Training Procedure of AdaptMatch

```
Input: labeled training set $\mathcal{D}_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{N_l}$, unlabeled
    training set $\mathcal{D}_u = \{(\mathbf{x}_i^u)\}_{i=1}^{N_u}$, segmentation model $\mathcal{E}$-$\mathcal{G}$,
    and total iteration number $N_{iter}$, empty memory banks
    $\mathcal{M}^F = \mathcal{M}^{lF} \cup \mathcal{M}^{uF}$, $\mathcal{M}^B = \mathcal{M}^{lB} \cup \mathcal{M}^{uB}$ with labeled
    memory bank length $N_{iter}^l$ and unlabeled memory bank
    length $N_{iter}^u$
for $iter \leftarrow 1$ to $N_{iter}$ do
    sampling data: sample and weakly augment labeled pair
      $(\mathbf{x}^l, \mathbf{y}^l)$ from $\mathcal{D}_l$ and unlabeled image $\mathbf{x}_i^{wu}$ from $\mathcal{D}_u$, and
      strongly augment $\mathbf{x}^{wu}$ into $\mathbf{x}^{su}$;
    obtaining predictions and pseudo-labels: obtain labeled,
      weakly-augmented, and strongly-augmented predictions,
      $\mathbf{p}^l$, $\mathbf{p}^{wu}$, and $\mathbf{p}^{su}$, from $\mathbf{x}^l$, $\mathbf{x}^{wu}$, and $\mathbf{x}^{su}$, via segmentation
      model $\mathcal{E}$-$\mathcal{G}$; then generate pseudo-label $\tilde{\mathbf{y}}^{wu}$ from $\mathbf{p}^{wu}$;
    calculating supervised loss: calculate the supervised loss
      $\mathcal{L}_{sup}$ between $\mathbf{p}^l$ and $\mathbf{y}^l$;
    updating memory banks: use $\mathbf{p}^l$ to update $\mathcal{M}^{lF}$ and
      $\mathcal{M}^{lB}$, and use $\mathbf{p}^{wu}$ to update $\mathcal{M}^{uF}$ and $\mathcal{M}^{uB}$;
    calculating thresholds: calculate foreground and back-
      ground thresholds, $\tau^F$ and $\tau^B$, from $\mathcal{M}^F = \mathcal{M}^{lF} \cup \mathcal{M}^{uF}$
      and $\mathcal{M}^B = \mathcal{M}^{lB} \cup \mathcal{M}^{uB}$;
    calculating mask map: calculate mask map $\mathbf{m}^{wu}$ from
      $\mathbf{p}^{wu}$ based on $\tau^F$ and $\tau^B$;
    calculating unsupervised loss: calculate the unsupervised
      loss $\mathcal{L}_{uns}$ based on $\mathbf{p}^{su}$, $\tilde{\mathbf{y}}^{wu}$ and $\mathbf{m}^l$; if $iter$ is less than
      $N_{iter}^l$, $\mathcal{L}_{uns}$ is set to 0;
    optimizing model: use the combination of $\mathcal{L}_{sup}$ and $\mathcal{L}_{uns}$
      to simultaneously optimize the segmentation model;
end
Output: optimized segmentation model $\mathcal{E}$-$\mathcal{G}$
```

IV. Experiments

In this section, experimental settings, including datasets, metrics, and implementation details, are first introduced. Then, the ablation study of AdaptMatch is conducted to explore the effectiveness of each component, with the corresponding metric visualizations during training. Next, the thresholds of the foreground and background are plotted at the training stage. Model-agnostic experiments are then conducted on different segmentation models to verify the robustness and generalizability of AdaptMatch. After that, extensive comparison experiments are implemented to compare the proposed AdaptMatch and other state-of-the-art SSS methods in RS semisupervised binary segmentation. Finally, some segmentation samples and t-Distributed Stochastic Neighbor Embedding (tSNEs) of high-level features are provided for intuitive comparison.

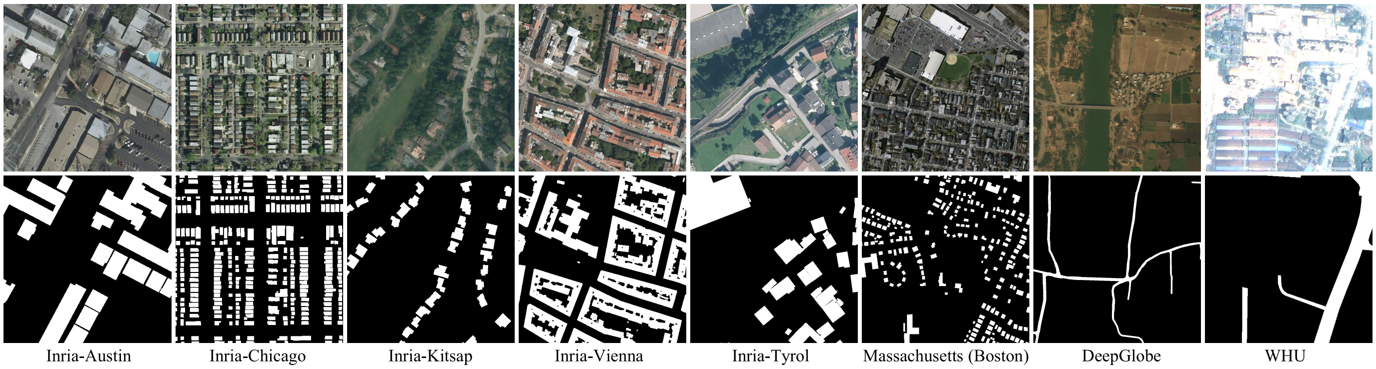


Fig. 2. Some examples of Inria, Massachusetts, DeepGlobe, and WHU.

A. Experimental Settings

1) Datasets and Metrics: To comprehensively evaluate the proposed method in RS semisupervised binary segmentation, we conduct experiments on two widely studied tasks, building footprint extraction and road extraction. Correspondingly, two building footprint datasets, Inria [39] and Massachusetts [54], and two road datasets, WHU_Roads [55] and DeepGlobe_Roads [56], have been used. Inria contains the five cities of Austin, Chicago, Kitsap, Tyrol, and Vienna, and we conduct the comparison experiments on each of them for individual and stable evaluation; because DeepGlobe's test ground-truth is publicly available, we split its training part into training, validation, and test. As a result, there is a total of eight subdatasets in comparison experiments: Austin, Chicago, Kitsap, Tyrol, Vienna, Massachusetts, WHU, and DeepGlobe. All of them are cropped into the same size of 512 × 512 and then are randomly split into training, validation, and test sets. The details of these datasets are summarized in Table I. Some examples of these datasets are shown in Fig. 2. Four classical metrics of the foreground class are used to evaluate binary segmentation performance comprehensively: Recall, Precision, IoU, and $F1$-score. For each of these metrics, the higher their values are, the better their performance is.

2) Implementation Details: Experiments are employed based on three advanced ImageNet-pretrained [57] semantic segmentation backbones, including two CNNs, EfficientUNet-B1 [58], and Deeplab_v3+ [59], and one state-of-the-art ViT backbone, SegFormer-B2. The weak augmentations for the labeled data and weakly augmented unlabeled data include random vertical flip, random horizontal flip, random rescaling between 0.5 and 2.0, and random crop. In addition to the weak augmentations, nine kinds of strong augmentation strategies referred from RandAugment [48], [60] are applied to the strongly augmented unlabeled data: equalize, identity, contrast, sharpness, autocontrast, brightness, color, posterize, and solarize. After being cleared, they accumulate from scratch again. Comparison methods are reproduced based on their official codes, including CCT [20],[1] CutMix [46], [47],[2] CPS [21],[3] CCVC [22],[4] FixMatch [34],[5] UniMatch [37].[6] For the actual model training, Adam [61] is used as the optimizer to train the models. The learning rate is initialized at 2.5e-4 and decreases with iterations as $lr = lr_{\text{init}} \cdot (1 - (\text{iter}/N_{\text{iter}}))^{0.9}$, where N_{iter} is the total number of training iterations. The mini-batch size M is set to 4 for Only-sup, Fully-sup, FixMatch, and our AdaptMatch, and to 2 for CCT, CutMix, CPS, CCVC, ICNet, and UniMatch because of their high GPU memory occupation. In our AdaptMatch, the labeled, weakly augmented unlabeled, and strongly augmented unlabeled branches share the same mini-batch size of 4 at each iteration. All the methods are trained for 10K iterations for all the building footprint datasets and 20K iterations for the road datasets, and they are evaluated every 500 iterations. During training, the models with the best validation performance are saved and tested by the test sets after training. The experiments are implemented based on PyTorch 1.9.1[7] on one Tesla V-100 GPU with 32 GB memory.

B. Ablation Study

In general, the whole objective function of AdaptMatch consists of the supervised loss $\mathcal{L}_{\text{sup}}$ for the labeled data and the unsupervised loss $\mathcal{L}_{\text{uns}}$ for the unlabeled data. Here, it is worth noting that $\mathcal{L}_{\text{uns}}$ is significantly affected by the thresholds $\{\tau^F, \tau^B\}$. To verify the respective effect of the two sets, $\mathcal{L}_{\text{uns}}$ is split into two corresponding parts, of which one is to only use the labeled set for calculating $\{\tau^F, \tau^B\}$ (denoted as $\mathcal{L}_{\text{uns}}^{l}$) and the other to also use the unlabeled set for calculation (denoted as $\mathcal{L}_{\text{uns}}^{u}$). Here, the ablation experiments are conducted in Inria-Austin, Inria-Kitsap, WHU, and DeepGlobe, with SegFormer-B2 as the binary segmentation model at three labeled ratios of 1%, 5%, and 20%. Experimental results are provided in Table II, where overall accuracy (OA) is provided to evaluate the overall performance of both foreground and background.

The first lines show the baseline performances trained only on the labeled sets by the BCE + IoU loss, that is, $\mathcal{L}_{\text{sup}}$. Beyond that, when the labeled set is used for calculating τ ($\mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{uns}}^{l}$), the Recall metric improves significantly,

[1] https://github.com/yassouali/CCT
[2] https://github.com/Britefury/cutmix-semisup-seg
[3] https://github.com/charlesCXK/TorchSemiSeg
[4] https://github.com/xiaoyao3302/CCVC
[5] https://github.com/google-research/fixmatch
[6] https://github.com/LiheYoung/UniMatch
[7] https://pytorch.org/

TABLE II

ABLATION STUDY OF THE PROPOSED ADAPTMATCH

Datasets	$\mathcal{L}_{sup}$	$\mathcal{L}_{uns}^{l}$	$\mathcal{L}_{uns}^{u}$	1%				5%				20%			
				Rec.	Pre.	IoU	F1	Rec.	Pre.	IoU	F1	Rec.	Pre.	IoU	F1
Inria-Austin	✓			77.86	85.24	68.61	81.38	85.18	87.09	75.63	86.13	87.55	87.38	77.84	87.46
	✓	✓		84.51	83.87	**72.70**	**84.19**	86.84	86.86	76.75	86.85	88.39	87.16	**78.21**	**87.77**
	✓	✓	✓	84.19	84.12	72.65	84.16	86.54	87.62	**77.05**	**87.07**	88.31	87.17	78.16	87.74
Inria-Kitsap	✓			62.72	76.23	52.46	68.82	70.20	80.57	60.04	75.03	78.07	79.48	64.98	78.77
	✓	✓		73.99	75.77	**59.84**	**74.87**	76.80	79.79	**64.29**	**78.26**	79.21	79.65	65.88	79.43
	✓	✓	✓	70.73	77.99	58.96	74.18	76.02	80.58	64.25	78.23	78.70	80.51	**66.10**	**79.59**
WHU	✓			63.31	72.62	51.11	67.65	71.98	76.50	58.95	74.17	77.17	78.96	64.00	78.05
	✓	✓		72.07	66.91	53.13	69.40	73.21	76.13	59.54	74.64	76.65	79.76	64.17	78.18
	✓	✓	✓	70.00	70.80	**54.32**	**70.40**	75.42	75.55	**60.62**	**75.48**	76.77	80.25	**64.57**	**78.47**
DeepGlobe	✓			56.56	78.03	48.79	65.58	72.51	74.95	58.37	73.71	75.67	77.17	61.83	76.41
	✓	✓		70.71	69.95	54.23	70.33	73.19	76.13	**59.53**	**74.63**	75.90	76.26	61.39	76.08
	✓	✓	✓	67.97	74.19	**54.97**	**70.94**	74.01	75.26	**59.53**	**74.63**	76.09	77.87	**62.56**	**76.97**

TABLE III

MODEL-AGNOSTIC EXPERIMENTS OF THE PROPOSED ADAPTMATCH

Datasets	Model: Method	1%				5%				20%			
		Rec.	Pre.	IoU	F1	Rec.	Pre.	IoU	F1	Rec.	Pre.	IoU	F1
Inria-Austin	Deeplab_V3+: Only-Sup	74.74	84.28	65.60	79.23	81.62	83.82	70.51	82.71	80.69	87.92	72.63	84.15
	Deeplab_V3+: AdaptMatch	77.97	84.60	68.28	81.15	82.22	85.52	72.17	83.84	82.50	85.95	72.70	84.19
	EfficientUNet_B1: Only-Sup	74.92	83.86	65.48	79.14	81.86	86.52	72.61	84.13	84.61	87.73	75.65	86.14
	EfficientUNet_B1: AdaptMatch	82.96	83.20	71.06	83.08	85.14	86.01	74.78	85.57	85.35	87.45	76.04	86.39
	SegFormer_B2: Only-Sup	77.86	85.24	68.61	81.38	85.18	87.09	75.63	86.13	87.55	87.38	77.84	87.46
	SegFormer_B2: AdaptMatch	84.19	84.12	**72.65**	**84.16**	86.54	87.62	**77.05**	**87.07**	88.31	87.17	**78.16**	**87.74**
Inria-Kitsap	Deeplab_V3+: Only-Sup	63.88	61.48	45.62	62.66	69.55	76.83	57.49	73.01	74.37	79.17	62.20	76.70
	Deeplab_V3+: AdaptMatch	57.68	73.80	47.87	64.75	66.89	80.69	57.66	73.14	74.03	81.84	62.94	77.73
	EfficientUNet_B1: Only-Sup	73.07	68.33	54.58	70.62	74.57	80.80	63.35	77.56	77.71	81.37	65.97	79.50
	EfficientUNet_B1: AdaptMatch	73.01	75.90	**59.26**	**74.42**	75.09	81.02	63.86	77.94	77.81	82.05	**66.49**	**79.87**
	SegFormer_B2: Only-Sup	62.72	76.23	52.46	68.82	70.20	80.57	60.04	75.03	78.07	79.48	64.98	78.77
	SegFormer_B2: AdaptMatch	70.73	77.99	58.96	74.18	76.02	80.58	**64.25**	**78.23**	78.70	80.51	66.10	79.59
WHU	Deeplab_V3+: Only-Sup	58.00	60.55	42.10	59.25	69.33	72.06	54.64	70.67	70.88	74.74	57.19	72.76
	Deeplab_V3+: AdaptMatch	58.65	74.47	48.83	65.62	69.88	72.90	55.47	71.36	75.65	67.04	55.14	71.09
	EfficientUNet_B1: Only-Sup	58.79	77.44	50.19	66.84	71.00	77.07	58.61	73.91	73.27	80.71	62.35	76.81
	EfficientUNet_B1: AdaptMatch	74.07	68.14	**55.01**	**70.98**	76.45	72.74	59.42	74.55	74.79	79.88	62.93	77.25
	SegFormer_B2: Only-Sup	63.31	72.62	51.11	67.65	71.98	76.50	58.95	74.17	77.17	78.96	64.00	78.05
	SegFormer_B2: AdaptMatch	70.00	70.80	54.32	70.40	75.42	75.55	**60.62**	**75.48**	76.77	80.25	**64.57**	**78.47**
DeepGlobe	Deeplab_V3+: Only-Sup	53.85	69.90	43.71	60.83	61.54	77.32	52.13	68.53	69.81	74.27	56.33	71.91
	Deeplab_V3+: AdaptMatch	55.37	79.35	48.40	65.22	65.74	74.34	53.58	69.78	73.14	71.10	56.38	72.10
	EfficientUNet_B1: Only-Sup	60.20	71.03	48.33	65.17	71.97	72.38	56.46	72.17	73.94	76.63	60.34	75.26
	EfficientUNet_B1: AdaptMatch	69.92	69.51	53.51	69.70	73.14	73.48	57.87	73.31	74.19	76.84	60.63	75.49
	SegFormer_B2: Only-Sup	56.56	78.03	48.79	65.58	72.51	74.95	58.37	73.71	75.67	77.17	61.83	76.41
	SegFormer_B2: AdaptMatch	67.97	74.19	**54.97**	**70.94**	74.01	75.26	**59.53**	**74.63**	76.09	77.87	**62.56**	**76.97**

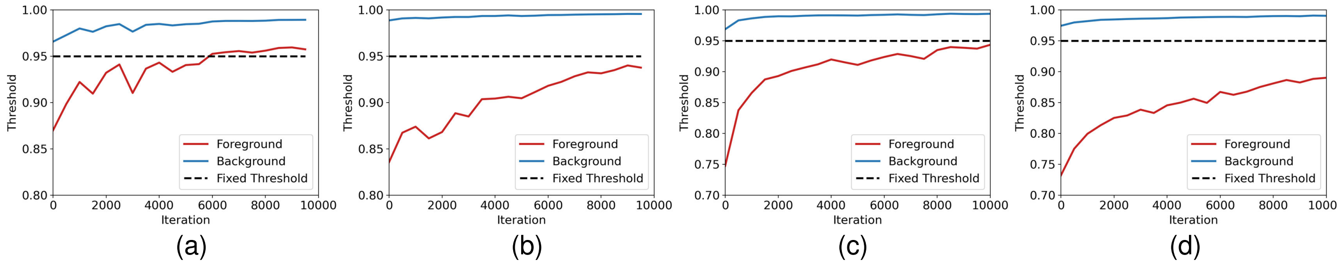


Fig. 3. Adaptive foreground and background thresholds with training iterations under the 1% labeled ratio. For better visualization, red: τ^B, blue: $1-\tau^F$. (a) 1% Inria-Austin. (b) 1% Inria-Kitsap. (c) 1% WHU_Roads. (d) 1% DeepGlobe_Roads.

especially at the small ratio. For example, $\mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{uns}}^{l}$ boost the Recall of 1% labeled Inria-Kitsap from 62.73% to 73.99%, with an improvement of 11.26% points. After further utilizing the unlabeled set for the calculation of $\{\tau^F, \tau^B\}$, there is not always performance improvement, and there is even a small drop of Recall at the labeled ratio of 1%. The main reason is that when there is not enough data for model training, the pseudo-labels of the unlabeled data are biased to the

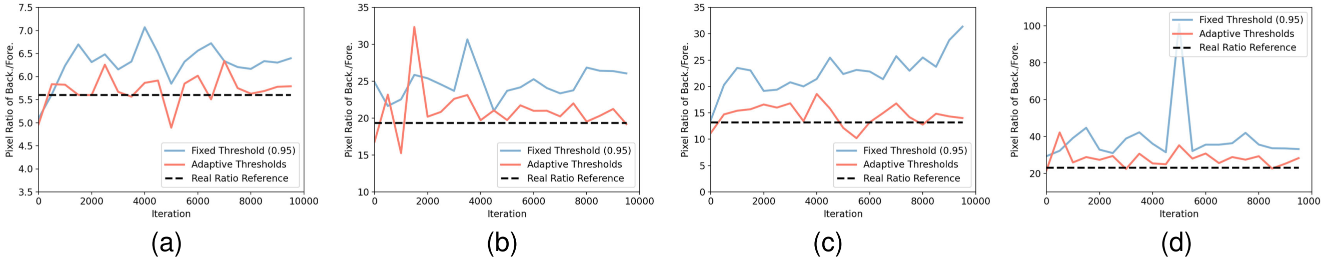


Fig. 4. Pixel number ratios of predicted foreground/background of validation sets by a fixed threshold (FixMatch) and adaptive thresholds (our AdaptMatch), respectively. The dashed lines are the real pixel ratios of ground-truth foreground/background. (a) 1% Inria-Austin. (b) 1% Inria-Kitsap. (c) 1% WHU_Roads. (d) 1% DeepGlobe_Roads.

TABLE IV

COMPARISON RESULTS (%) OF ADAPTMATCH AND SOME STATE-OF-THE-ART METHODS ON THE INRIA-AUSTIN DATASET BASED ON SEGFORMER_B2. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND ONES ARE UNDERLINED

Ratio	Method	Inria-Austin			
		Rec.	Pre.	IoU	F1
1%	Only-Sup	77.86	85.24	68.61	81.38
	CutMix [47] [BMVC'20]	78.43	86.18	69.66	82.12
	CCT [20], [51] [CVPR'20, Arxiv'22]	76.79	86.07	68.30	81.16
	CPS [21] [CVPR'21]	82,24	77.33	66.26	79.71
	CCVC [22] [CVPR'23]	83.90	75.48	65.94	79.47
	ICNet [44] [GRSL'22]	83.24	78.75	67.97	80.93
	UniMatch [37] [CVPR'23]	82.03	84.76	71.48	83.37
	FixMatch [34] [NeurIPS'20]	80.11	87.16	71.66	83.49
	Our AdaptMatch	84.19	84.12	**72.65**	**84.16**
5%	Only-Sup	85.18	87.09	75.63	86.13
	CutMix [47] [BMVC'20]	83.68	87.07	74.43	85.34
	CCT [20], [51] [CVPR'20, Arxiv'22]	84.33	86.22	74.32	85.27
	CPS [21] [CVPR'21]	83.43	86.10	73.53	84.75
	CCVC [22] [CVPR'23]	86.37	82.34	72.88	84.31
	ICNet [44] [GRSL'22]	85.05	84.01	73.20	84.53
	UniMatch [37] [CVPR'23]	85.30	86.71	75.44	86.01
	FixMatch [34] [NeurIPS'20]	85.38	87.27	75.93	86.32
	Our AdaptMatch	86.54	87.62	**77.05**	**87.07**
20%	Only-Sup	87.55	87.38	77.84	87.46
	CutMix [47] [BMVC'20]	86.40	86.84	76.40	86.62
	CCT [20], [51] [CVPR'20, Arxiv'22]	87.57	85.83	76.50	86.69
	CPS [21] [CVPR'21]	84.87	88.02	76.08	86.42
	CCVC [22] [CVPR'23]	87.23	85.48	75.97	86.35
	ICNet [44] [GRSL'22]	85.60	86.88	75.80	86.24
	UniMatch [37] [CVPR'23]	86.29	87.84	77.08	87.06
	FixMatch [34] [NeurIPS'20]	87.87	87.92	**78.41**	**87.90**
	Our AdaptMatch	88.31	87.17	78.16	87.74
100%	Fully-Sup	88.41	87.08	78.17	87.75

TABLE V

COMPARISON RESULTS (%) OF ADAPTMATCH AND SOME STATE-OF-THE-ART METHODS ON THE INRIA-CHICAGO DATASET BASED ON SEGFORMER_B2. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND ONES ARE UNDERLINED

Ratio	Method	Inria-Chicago			
		Rec.	Pre.	IoU	F1
1%	Only-Sup	81.51	81.55	69.26	81.43
	CutMix [47] [BMVC'20]	80.83	82.98	69.34	81.89
	CCT [20], [51] [CVPR'20, Arxiv'22]	81.86	82.09	69.45	81.97
	CPS [21] [CVPR'21]	79.38	71.86	60.56	75.43
	CCVC [22] [CVPR'23]	81.84	74.18	63.70	77.82
	ICNet [44] [GRSL'22]	81.23	82.95	69.61	82.08
	UniMatch [37] [CVPR'23]	85.93	81.38	71.81	83.59
	FixMatch [34] [NeurIPS'20]	83.13	85.38	72.77	84.24
	Our AdaptMatch	86.83	82.60	**73.41**	**84.66**
5%	Only-Sup	85.90	84.54	74.24	85.22
	CutMix [47] [BMVC'20]	86.02	84.29	74.13	85.15
	CCT [20], [51] [CVPR'20, Arxiv'22]	84.97	82.40	71.92	83.67
	CPS [21] [CVPR'21]	84.34	84.41	72.98	84.38
	CCVC [22] [CVPR'23]	84.54	83.09	72.12	83.80
	ICNet [44] [GRSL'22]	82.06	86.75	72.92	84.34
	UniMatch [37] [CVPR'23]	86.23	84.37	74.36	85.29
	FixMatch [34] [NeurIPS'20]	84.57	86.75	74.90	85.65
	Our AdaptMatch	87.75	84.34	**75.46**	**86.01**
20%	Only-Sup	87.87	86.11	76.96	86.98
	CutMix [47] [BMVC'20]	86.02	84.29	74.13	85.15
	CCT [20], [51] [CVPR'20, Arxiv'22]	88.27	84.86	76.26	86.53
	CPS [21] [CVPR'21]	84.10	87.42	75.02	85.73
	CCVC [22] [CVPR'23]	87.33	84.86	75.56	86.08
	ICNet [44] [GRSL'22]	83.46	89.02	75.67	86.15
	UniMatch [37] [CVPR'23]	85.78	87.33	76.29	86.55
	FixMatch [34] [NeurIPS'20]	88.04	86.33	77.27	87.18
	Our AdaptMatch	88.80	85.70	**77.34**	**87.22**
100%	Fully-Sup	90.09	84.87	77.62	87.40

background. In this case, these low-confidence foreground samples are predicted as background and do not contribute to the threshold of the foreground. Therefore, the foreground threshold τ^F of $\mathcal{L}_{\text{sup}} + \mathcal{L}^l_{\text{uns}} + \mathcal{L}^u_{\text{uns}}$ becomes relatively higher, resulting in lower Recall or even a marginal drop in some 1% and 5% cases compared with $\mathcal{L}_{\text{sup}} + \mathcal{L}^l_{\text{uns}}$. Nonetheless, $\mathcal{L}_{\text{sup}}+\mathcal{L}^l_{\text{uns}}+\mathcal{L}^u_{\text{uns}}$ yields the more stable performance of all the datasets under different labeled ratios, which is more friendly to practical RS semisupervised binary segmentation tasks with unclear labeled ratios.

In general, the introduction of the adaptive thresholds $\{\tau^F, \tau^B\}$ significantly improves the recall rate of the foreground while maintaining relatively stable precision, and thus improves the overall performance, as observed from IoU and $F1$. The improvement of OA is not as significant as IoU and $F1$ because the improvement mainly comes from the foreground, which only accounts for a small part of all pixels but attracts our main interest. It is worth mentioning that the performance gain of the proposed decreases with the increase of the labeled ratio, especially for 20%. The main reason is that although there is a big gap in the absolute number of labeled data between 20% and 100%, the increase in effective samples is not significant. In other words, there are redundant data in 100%, most of which is repetitive from feature representation. As a result, the performance of Only-Sup quickly reaches saturation with an increase in the labeled ratio, which can be observed from the rapid decrease in the gap between it and Fully-Sup in Section IV-B.

TABLE VI

COMPARISON RESULTS (%) OF ADAPTMATCH AND SOME STATE-OF-THE-ART METHODS ON THE INRIA-VIENNA DATASET BASED ON SEGFORMER_B2. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND ONES ARE UNDERLINED

Ratio	Method	Inria-Vienna			
		Rec.	Pre.	IoU	F1
1%	Only-Sup	83.78	87.28	74.67	85.50
	CutMix [47] [BMVC'20]	85.63	87.27	76.12	86.44
	CCT [20], [51] [CVPR'20, Arxiv'22]	85.30	86.46	75.24	85.87
	CPS [21] [CVPR'21]	84.87	86.24	74.75	85.55
	CCVC [22] [CVPR'23]	82.58	85.15	72.19	83.85
	ICNet [44] [GRSL'22]	86.44	84.04	74.25	85.23
	UniMatch [37] [CVPR'23]	86.55	88.61	77.88	87.57
	FixMatch [34] [NeurIPS'20]	85.67	88.71	77.25	87.16
	Our AdaptMatch	88.84	87.25	**78.63**	**88.04**
5%	Only-Sup	88.76	90.11	80.88	89.43
	CutMix [47] [BMVC'20]	88.93	89.24	80.32	89.09
	CCT [20], [51] [CVPR'20, Arxiv'22]	89.34	89.23	80.64	89.28
	CPS [21] [CVPR'21]	88.08	88.37	78.93	88.23
	CCVC [22] [CVPR'23]	87.78	89.21	79.35	88.49
	ICNet [44] [GRSL'22]	88.83	87.85	79.11	88.34
	UniMatch [37] [CVPR'23]	89.15	89.29	80.54	89.22
	FixMatch [34] [NeurIPS'20]	89.12	90.21	**81.26**	**89.66**
	Our AdaptMatch	89.69	89.62	81.25	89.65
20%	Only-Sup	90.38	89.93	82.07	90.15
	CutMix [47] [BMVC'20]	88.93	89.24	80.32	89.09
	CCT [20], [51] [CVPR'20, Arxiv'22]	90.15	88.89	81.02	89.52
	CPS [21] [CVPR'21]	90.10	89.01	81.08	89.55
	CCVC [22] [CVPR'23]	88.58	89.78	80.46	89.17
	ICNet [44] [GRSL'22]	88.52	90.73	81.18	89.61
	UniMatch [37] [CVPR'23]	89.51	89.98	81.39	89.74
	FixMatch [34] [NeurIPS'20]	90.26	90.40	82.36	90.33
	Our AdaptMatch	91.25	89.89	**82.76**	**90.56**
100%	Fully-Sup	91.24	90.17	82.99	90.70

TABLE VII

COMPARISON RESULTS (%) OF ADAPTMATCH AND SOME STATE-OF-THE-ART METHODS ON THE INRIA-KITSAP DATASET BASED ON SEGFORMER_B2. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND ONES ARE UNDERLINED

Ratio	Method	Inria-Kitsap			
		Rec.	Pre.	IoU	F1
1%	Only-Sup	62.72	76.23	52.46	68.82
	CutMix [47] [BMVC'20]	64.08	74.19	52.40	68.77
	CCT [20], [51] [CVPR'20, Arxiv'22]	63.74	70.31	50.22	66.86
	CPS [21] [CVPR'21]	69.65	70.71	54.05	70.18
	CCVC [22] [CVPR'23]	77.56	63.75	53.82	69.98
	ICNet [44] [GRSL'22]	70.76	54.43	44.44	61.53
	UniMatch [37] [CVPR'23]	65.95	75.92	54.54	70.58
	FixMatch [34] [NeurIPS'20]	60.58	81.03	53.06	69.33
	Our AdaptMatch	70.73	77.99	**58.96**	**74.18**
5%	Only-Sup	70.20	80.57	60.04	75.03
	CutMix [47] [BMVC'20]	71.34	78.69	59.78	74.83
	CCT [20], [51] [CVPR'20, Arxiv'22]	72.14	79.88	61.05	75.81
	CPS [21] [CVPR'21]	70.73	80.25	60.24	75.19
	CCVC [22] [CVPR'23]	79.57	71.98	60.75	75.58
	ICNet [44] [GRSL'22]	74.42	72.41	57.98	73.40
	UniMatch [37] [CVPR'23]	70.36	79.49	59.55	74.64
	FixMatch [34] [NeurIPS'20]	72.79	80.66	61.97	76.52
	Our AdaptMatch	76.02	80.58	**64.25**	**78.23**
20%	Only-Sup	78.07	79.48	64.98	78.77
	CutMix [47] [BMVC'20]	71.34	78.69	59.78	74.83
	CCT [20], [51] [CVPR'20, Arxiv'22]	71.65	78.54	59.92	74.94
	CPS [21] [CVPR'21]	76.45	78.45	63.18	77.44
	CCVC [22] [CVPR'23]	77.32	73.46	60.43	75.34
	ICNet [44] [GRSL'22]	77.38	75.26	61.69	76.31
	UniMatch [37] [CVPR'23]	80.09	76.62	64.36	78.31
	FixMatch [34] [NeurIPS'20]	77.53	81.02	65.61	79.24
	Our AdaptMatch	78.70	80.51	**66.10**	**79.59**
100%	Fully-Sup	80.44	81.30	67.88	80.87

C. Model-Agnostic Experiments

To verify the generalizability of our AdaptMatch on segmentation models, in this section, we conduct model-agnostic experiments based on Deeplab_V3, EfficientUNet_B1, and SegFormer_B2, of which the first two are classical CNNs and the third is ViT architecture. Similar to Section IV-B, the experimental results are based on Inria-Austin, Inria-Kitsap, WHU_Roads, and DeepGlobe_Roads as the model at three labeled ratios of 1%, 5%, and 20%. The results are shown in Table III.

The proposed AdaptMatch can boost the performance of all the models at different ratios to different extents, demonstrating its robustness and generalization to models. When only considering models, we find that Deeplab_V3+ shows the worst performance in comparison with EfficientUNet_B1 and SegFormer_B2 for both Only-Sup and AdaptMatch. For example, at the ratio of 1%, the smallest IoU gain of Deeplab_V3+ is 2.25% of Inria-Kitsap; in contrast, at the same ratio, the smallest IoU improvements of EfficientUNet_B1 and SegFormer_B2 are 4.68% of Inria-Kitsap and 3.21% of WHU, respectively. The relatively poor segmentation ability of Deeplab_V3+ leads to unstable feature representation and probability predictions, limiting the capacity of the proposed AdaptMatch to utilize high-quality unlabeled data for self-training. When it comes to EfficientUNet_B1 and SegFormer_B2, the gains obtained from AdaptMatch are greater and more stable. This phenomenon demonstrates that an excellent segmentation model can not only provide high base performance in the only supervised setting but also produce high-quality and stable predictions of unlabeled data for further SSL.

D. Visualizations of Adaptive Thresholds and Rebalanced Pseudo-Labels

To intuitively show the effect of our AdaptMatch, we visualize the thresholds of the foreground and the background when training. The used datasets and labeled ratios are set the same as in Section IV-B and IV-C. The thresholds are plotted in Fig. 3. In Fig. 3, the dashed line represents the fixed threshold of 0.95 that remains consistent with FixMatch [34]. The red lines are the thresholds calculated by AdaptMatch of the foreground at the 1% labeled ratio, and the blue lines are those of the background.

When we pay attention to the difference between classes, it can be observed that, compared with the foreground, the background has a much higher threshold that rapidly converges to quite close to 1; in contrast, the threshold of the foreground does not reach 0.95 in most cases. This indicates that it is harder for models to detect the foreground than the background, which is consistent with our motivation to decrease the threshold of the foreground and increase that of the background for balanced training. From the perspective of tasks, we find that WHU_Roads and DeepGlobe_Roads have lower thresholds of the foreground (road) than Inria-Austin and

TABLE VIII

Comparison Results (%) of AdaptMatch and Some State-of-the-Art Methods on the Inria-Tyrol Dataset Based on SegFormer_B2. The Best Results Are in **Bold** and the Second Ones Are Underlined

Ratio	Method	Inria-Tyrol			
		Rec.	Pre.	IoU	F1
1%	Only-Sup	78.34	79.12	64.92	78.73
	CutMix [47] [BMVC'20]	77.48	78.00	63.59	77.74
	CCT [20], [51] [CVPR'20, Arxiv'22]	67.89	72.31	53.88	70.03
	CPS [21] [CVPR'21]	69.94	80.51	59.81	74.86
	CCVC [22] [CVPR'23]	77.41	67.31	56.25	72.00
	ICNet [44] [GRSL'22]	66.83	72.20	53.16	69.41
	UniMatch [37] [CVPR'23]	75.90	85.81	67.43	80.55
	FixMatch [34] [NeurIPS'20]	74.73	84.60	65.78	79.36
	Our AdaptMatch	85.59	78.02	**68.97**	**81.63**
5%	Only-Sup	83.95	86.36	74.12	85.14
	CutMix [47] [BMVC'20]	81.71	83.05	70.03	82.37
	CCT [20], [51] [CVPR'20, Arxiv'22]	79.79	84.04	69.29	81.86
	CPS [21] [CVPR'21]	81.40	83.85	70.36	82.60
	CCVC [22] [CVPR'23]	84.94	80.09	70.13	82.44
	ICNet [44] [GRSL'22]	76.49	81.71	65.31	79.01
	UniMatch [37] [CVPR'23]	82.38	82.52	70.14	82.45
	FixMatch [34] [NeurIPS'20]	84.04	86.04	73.95	85.02
	Our AdaptMatch	85.10	86.39	**75.04**	**85.74**
20%	Only-Sup	86.66	85.87	75.84	86.26
	CutMix [47] [BMVC'20]	81.71	83.05	70.03	82.37
	CCT [20], [51] [CVPR'20, Arxiv'22]	82.49	79.48	68.00	80.95
	CPS [21] [CVPR'21]	85.45	81.61	71.66	83.49
	CCVC [22] [CVPR'23]	81.28	82.54	69.35	81.90
	ICNet [44] [GRSL'22]	76.79	84.95	67.59	80.66
	UniMatch [37] [CVPR'23]	82.32	83.69	70.94	83.00
	FixMatch [34] [NeurIPS'20]	86.71	87.00	76.77	86.86
	Our AdaptMatch	86.88	87.32	**77.15**	**87.10**
100%	Fully-Sup	87.91	86.21	77.07	87.05

TABLE X

Comparison Results (%) of AdaptMatch and Some State-of-the-Art Methods on the WHU_Roads Dataset Based on SegFormer_B2. The Best Results Are in **Bold** and the Second Ones Are Underlined

Ratio	Method	WHU_Roads			
		Rec.	Pre.	IoU	F1
1%	Only-Sup	63.31	72.62	51.11	67.65
	CutMix [47] [BMVC'20]	55.15	74.69	46.47	63.45
	CCT [20], [51] [CVPR'20, Arxiv'22]	57.77	73.12	47.65	64.55
	CPS [21] [CVPR'21]	40.05	74.92	35.32	52.18
	CCVC [22] [CVPR'23]	65.72	16.18	14.92	25.97
	ICNet [44] [GRSL'22]	67.14	66.31	50.06	66.72
	UniMatch [37] [CVPR'23]	62.87	71.21	50.13	66.78
	FixMatch [34] [NeurIPS'20]	61.60	71.99	49.69	66.39
	Our AdaptMatch	70.00	70.80	**54.32**	**70.40**
5%	Only-Sup	71.98	76.50	58.95	74.17
	CutMix [47] [BMVC'20]	73.08	74.96	58.74	74.01
	CCT [20], [51] [CVPR'20, Arxiv'22]	69.33	77.28	57.59	73.09
	CPS [21] [CVPR'21]	67.94	78.38	57.22	72.79
	CCVC [22] [CVPR'23]	77.28	69.54	57.74	73.21
	ICNet [44] [GRSL'22]	71.50	75.25	57.89	73.33
	UniMatch [37] [CVPR'23]	73.93	75.26	59.47	74.59
	FixMatch [34] [NeurIPS'20]	75.10	75.69	60.50	75.39
	Our AdaptMatch	75.42	75.55	**60.62**	**75.48**
20%	Only-Sup	77.17	78.96	64.00	78.05
	CutMix [47] [BMVC'20]	66.40	72.31	52.94	69.23
	CCT [20], [51] [CVPR'20, Arxiv'22]	76.39	77.94	62.81	77.16
	CPS [21] [CVPR'21]	75.96	78.01	62.57	76.97
	CCVC [22] [CVPR'23]	79.14	75.38	62.89	77.22
	ICNet [44] [GRSL'22]	73.33	78.07	60.80	75.62
	UniMatch [37] [CVPR'23]	79.46	75.39	63.09	77.37
	FixMatch [34] [NeurIPS'20]	76.60	80.18	64.41	78.35
	Our AdaptMatch	76.77	80.25	**64.57**	**78.47**
100%	Fully-Sup	78.88	80.84	66.46	79.85

TABLE IX

Comparison Results (%) of AdaptMatch and Some State-of-the-Art Methods on the Massachusetts Buildings Dataset Based on SegFormer_B2. The Best Results Are in **Bold** and the Second Ones Are Underlined

Ratio	Method	Massachusetts Buildings			
		Rec.	Pre.	IoU	F1
5%	Only-Sup	78.69	81.33	66.65	79.99
	CutMix [47] [BMVC'20]	78.20	81.68	66.53	79.90
	CCT [20], [51] [CVPR'20, Arxiv'22]	79.59	79.76	66.21	79.67
	CPS [21] [CVPR'21]	72.14	84.68	63.81	77.91
	CCVC [22] [CVPR'23]	79.93	78.48	65.56	79.20
	ICNet [44] [GRSL'22]	78.13	79.35	64.93	78.74
	UniMatch [37] [CVPR'23]	79.34	82.53	67.94	80.91
	FixMatch [34] [NeurIPS'20]	77.50	82.98	66.87	80.14
	Our AdaptMatch	83.84	78.58	**68.24**	**81.12**
20%	Only-Sup	81.05	81.19	68.23	81.12
	CutMix [47] [BMVC'20]	78.41	77.56	63.91	77.98
	CCT [20], [51] [CVPR'20, Arxiv'22]	81.41	79.06	66.97	80.22
	CPS [21] [CVPR'21]	79.00	82.70	67.79	80.81
	CCVC [22] [CVPR'23]	81.92	81.00	68.72	81.46
	ICNet [44] [GRSL'22]	77.52	83.14	66.99	80.23
	UniMatch [37] [CVPR'23]	81.99	80.99	68.75	81.48
	FixMatch [34] [NeurIPS'20]	81.31	82.10	69.07	81.71
	Our AdaptMatch	81.36	82.69	**69.52**	**82.02**
100%	Fully-Sup	83.14	79.74	68.64	81.40

Inria-Kitsap (building footprint), revealing that road extraction is harder than building footprint extraction. There are two possible reasons for this: 1) the image resolution of WHU_Roads and DeepGlobe_Roads is lower than that of Inria, which increases their difficulty in obtaining fine-grained semantic features; and 2) there are many occlusions on roads, such as trees and buildings, which is more severe than that of buildings.

To further demonstrate the effect of AdaptMatch in rebalancing pseudo-labels, in Fig. 4, we plot the pixel ratios of the predicted foreground to the background of validation sets during training based on a fixed threshold of 0.95 (i.e., FixMatch) and adaptive thresholds calculated by our AdaptMatch. Fig. 4 shows that the adaptive thresholds of our AdaptMatch have similar converging trends of pixel number ratios to the fixed threshold of FixMatch due to their similar strong-to-weak self-training mechanism. However, the ratios of FixMatch remain higher than the real ratios in the middle and later stages of training; in contrast, the ratios of the adaptive thresholds are closer to the real ratios (shown by dashed lines in Fig. 4). Such results verify the positive effects of adaptive thresholds on rebalancing foreground and background distributions.

E. Comparison Experiments

To objectively evaluate the effectiveness of the proposed AdaptMatch in RS semisupervised binary segmentation, it is compared with some SSS methods based on the same model of SegFormer_B2 with the same training iterations. There are eight comparison methods in total, two supervised baselines, and seven diverse state-of-the-art methods, as follows.

1) Only-Sup, which only uses the limited labeled data for model training.

TABLE XI

COMPARISON RESULTS (%) OF ADAPTMATCH AND SOME STATE-OF-THE-ART METHODS ON THE DEEPGLOBE_ROADS BUILDINGS DATASET BASED ON SEGFORMER_B2. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND ONES ARE UNDERLINED

Ratio	Method	DeepGlobe_Roads			
		Rec.	Pre.	IoU	F1
1%	Only-Sup	56.56	78.03	48.79	65.58
	CutMix [47] [BMVC'20]	58.03	72.23	47.44	64.36
	CCT [20], [51] [CVPR'20, Arxiv'22]	55.96	74.47	46.95	63.90
	CPS [21] [CVPR'21]	58.01	73.16	47.83	64.71
	CCVC [22] [CVPR'23]	72.38	55.90	46.07	63.08
	ICNet [44] [GRSL'22]	72.84	59.41	48.63	65.44
	UniMatch [37] [CVPR'23]	72.55	52.92	44.10	61.20
	FixMatch [34] [NeurIPS'20]	59.94	73.90	49.47	66.19
	Our AdaptMatch	67.97	74.19	**54.97**	**70.94**
5%	Only-Sup	72.51	74.95	58.37	73.71
	CutMix [47] [BMVC'20]	69.22	75.49	56.52	72.22
	CCT [20], [51] [CVPR'20, Arxiv'22]	68.01	76.88	56.46	72.17
	CPS [21] [CVPR'21]	65.85	78.64	55.86	71.68
	CCVC [22] [CVPR'23]	76.72	67.20	55.82	71.65
	ICNet [44] [GRSL'22]	73.13	68.47	54.71	70.72
	UniMatch [37] [CVPR'23]	76.65	69.33	57.24	72.80
	FixMatch [34] [NeurIPS'20]	70.54	78.71	59.24	74.40
	Our AdaptMatch	74.01	75.26	**59.53**	**74.63**
20%	Only-Sup	75.67	77.17	61.83	76.41
	CutMix [47] [BMVC'20]	78.87	70.14	59.04	74.25
	CCT [20], [51] [CVPR'20, Arxiv'22]	71.07	76.55	58.36	73.71
	CPS [21] [CVPR'21]	79.60	70.59	59.78	74.83
	CCVC [22] [CVPR'23]	78.07	69.01	57.81	73.26
	ICNet [44] [GRSL'22]	75.10	71.71	57.94	73.37
	UniMatch [37] [CVPR'23]	76.09	73.58	59.76	74.81
	FixMatch [34] [NeurIPS'20]	74.94	78.39	62.11	76.63
	Our AdaptMatch	76.09	77.87	**62.56**	**76.97**
100%	Fully-Sup	75.19	78.06	62.07	76.60

2) Fully-Sup, which utilizes all the labeled data for training.
3) CCT [20], which keeps consistent predictions when adding various perturbations at the feature level, and is applied to semisupervised change detection in [51].
4) CutMix [46], [47], which applies the data augmentation technique of CutMix to the self-training of the SSS model as a regularization.
5) CPS [21], which introduces a classical cross-pseudo supervision regularization based on the predictions from two homogeneous but differently initialized models.
6) CCVC [22], which further extends the cross-supervision predictions to heterogeneous forms and proposes a distance loss to decrease their feature distance on the basis of CPS.
7) ICNet [44], which designs an iterative contrastive network for the semisupervised segmentation method of RS images.
8) FixMatch [34], which utilizes the high-confidence pseudo-labels from a weakly augmented instance selected by a fixed threshold to supervise the classification of a strongly augmented counterpart.
9) UniMatch [37], which further introduces a feature perturbation branch and another strong-augmentation branch on top of FixMatch.

Here, the proposed AdaptMatch follows a classical strong-to-weak self-training paradigm for the unlabeled data of FixMatch and UniMatch. However, different from the fixed threshold of FixMatch and UniMatch, the proposed method further designs an adaptive threshold mechanism for selecting class-wise high-confidence pseudo-labels based on class-aware convergence difficulties.

TABLE XII

TRAINING TIME OF AN ITERATION OF DIFFERENT METHODS ON 1% LABELED INRIA-KITSAP, WHICH IS OBTAINED BY AVERAGING 100 ITERATIONS FOR STABLE RESULTS

Method	Training Time (Second)	IoU
Only-Sup	0.62	52.46
UniMatch	1.31 (+0.69)	54.54
FixMatch	1.46 (+0.84)	53.06
AdaptMatch	1.53 (+0.91)	58.96

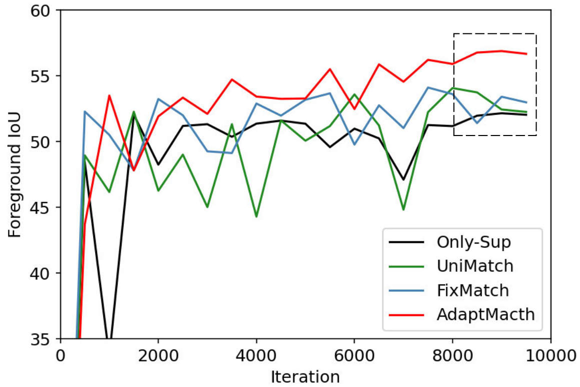


Fig. 5. Foreground IoU of the validation set during training.

The comparison experiments are conducted on Inria-Austin, Inria-Chicago, Inria-Vienna, Inria-Kitsap, and Inria-Tyrol, Massachusetts_Buildings, WHU_Roads, and DeepGlobe_Roads at the labeled ratios of 1%, 5%, and 20%. Since Massachusetts_Buildings contains relatively high noise labels, there is no 1% labeled Massachusetts_Buildings. The results are provided in Tables IV–XI.

From the perspective of methods, it can be seen that: in comparison with Only-Sup, the baseline, the CutMix, CCT, CPS, and CCVC methods show negative effects of the utilization of unlabeled data on model training; in contrast, UniMatch, FixMatch, and the proposed AdaptMatch have positive influences on performance improvement for the comprehensive metrics of IoU and $F1$-score, which consider Recall and Precision together. The significant difference among them is that the first group of methods uses all the unlabeled data for training, while the second group of methods only uses the high-confidence pseudo-labeled data for training via fixed/adaptive thresholds. Their performance gaps verify the necessity of the selection of pseudo-labels in RS semisupervised binary segmentation, especially with the serious imbalanced distribution of foreground and background. Although FixMatch outperforms AdaptMatch in a few scenarios, our AdaptMatch still achieves the majority of the best results, which are more robust across datasets in different ratios.

From the perspective of metrics, it could be observed that when maintaining relatively stable Precision, AdaptMatch has a high Recall ratio of the foreground. The Recall of AdaptMatch is higher than UniMatch and FixMatch while its

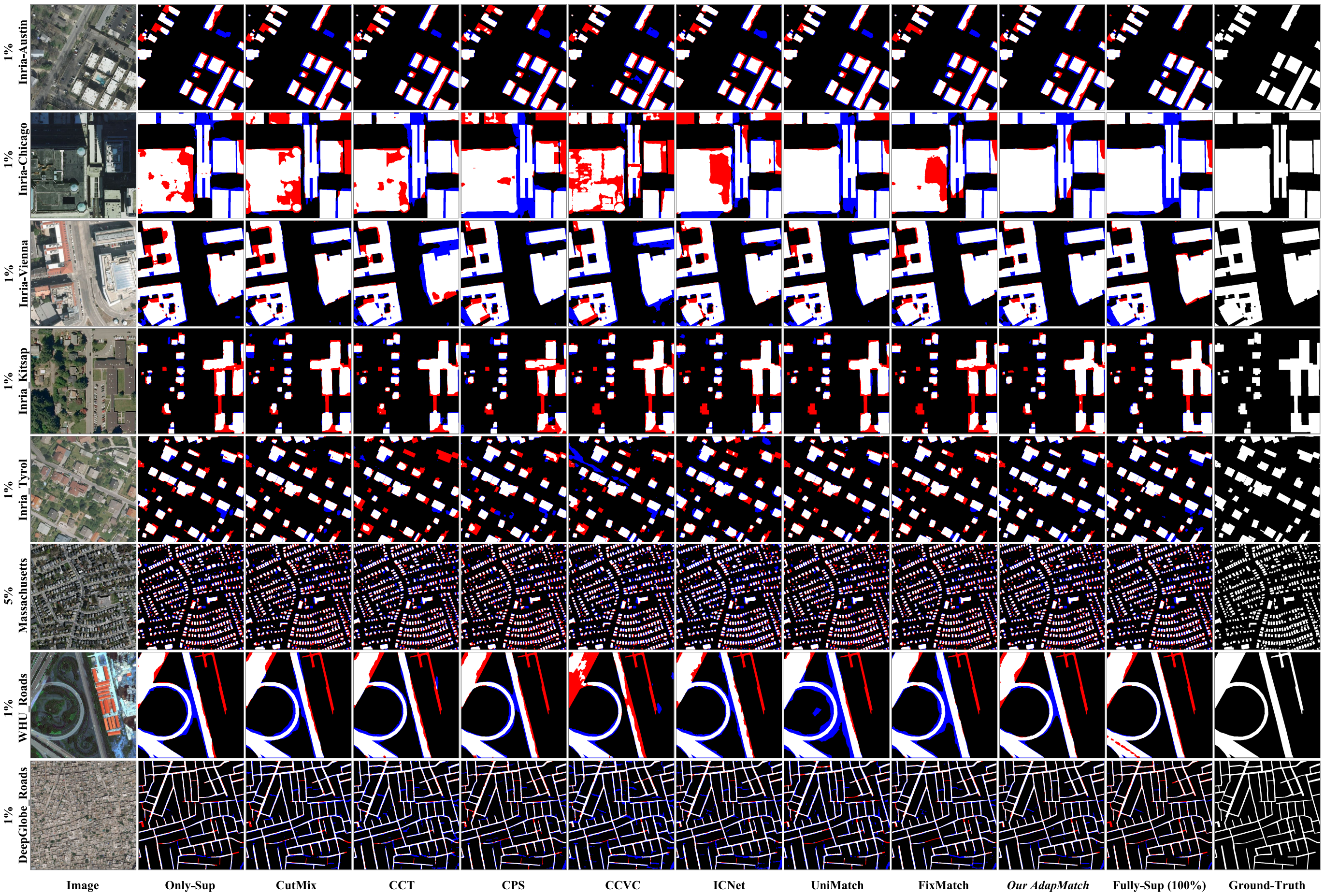


Fig. 6. Sample visualization of our AdaptMatch and comparison methods. Here, red represents the areas that are foreground in ground-truth but are predicted as background, blue represents the areas that are background in ground-truth but are predicted as foreground, and white represents the areas that are correctly predicted as foreground. In other words, "red + white" is the ground-truth, while "blue + white" is the predicted area. The less area red + blue covers, the better the performance is.

Precision is higher than others in general. This indicates that our method has a more balanced performance. As a result, AdaptMatch achieves almost all the best IoU and $F1$-score values.

Another interesting phenomenon is that our AdaptMatch with 20% labeled data can obtain performances that approach or even exceed Fully-Sup with 100% labeled data in some cases, such as Massachusetts_Buildings and DeepGlobe_Roads. The main reason is that the proposed AdaptMatch focuses on selecting balanced pseudo-labels for self-training to avoid the model bias to the dominant foreground, which is not sufficiently considered in the fully-supervised setting. In addition, AdaptMatch applies widely used strong augmentations to unlabeled data that are helpful for model generalizability. Moreover, these datasets contain more noisy data increasing as the labeled ratio rises, which reduces the gain of the labeled data; by contrast, the adaptive thresholding mechanism of AdaptMatch can filter out numerous noise.

In general, the proposed AdaptMatch can achieve superior and more robust performance across different RS semisupervised binary tasks at various labeled ratios, in comparison with other SSS methods.

F. Training Complexity and Convergence

Training complexity and convergence are important in practice and are therefore discussed in this section. A comparison of training complexity and convergence is made among the basic method Only-Sup, the two best comparison methods, UniMatch and FixMatch, and the proposed AdaptMatch.

Running on the same hardware Tesla V-100 GPU, Table XII shows the time cost of an iteration of the methods with their best IoUs. Overall, UniMatch, FixMatch, and AdaptMatch have a similar training complexity between 1.31 and 1.53 s for each iteration, which is around 2.3× that of Only-Sup. The key difference between FixMatch and AdaptMatch is that AdaptMach utilizes an extra memory mechanism to store predictions and calculate thresholds. Nevertheless, the increase in training time from FixMatch to AdaptMatch is only 0.07 s by 4.8%, which verifies the efficiency of the proposed memory bank-based threshold strategy.

The foreground IoU of the validation set of the methods during training is shown in Fig. 5. It can be found that the proposed AdaptMatch outperforms other comparison methods with higher and more stable results, especially in the second half stage. As shown in the dashed rectangle, AdaptMatch convergences to stable status roughly from 8000 iterations which is similar to Only-Sup; by contrast, FixMatch and UniMatch have fluctuating performance until 9000 iterations. The results demonstrate the superior training convergence of the proposed method.

G. Visualization of Some Samples

To intuitively compare the performance of the proposed AdaptMatch and the comparison methods, some example visualizations are provided in Fig. 6.

It can be seen that our method can achieve the most balanced performance between Recall and Precision. What's more, we can intuitively observe some characteristics of wrongly predicted areas, which we hope will inspire future work in RS semisupervised binary segmentation. First, we observe that many wrongly predicted pixels are located at the edges of buildings and roads (foreground) as a result of the occlusion from high objects (such as trees), the shadows caused by slanting sunlight, and inaccurate ground-truth. The second error source is the insufficient feature learning of some confusing objects and areas, such as the rectangular square in 1% labeled Inria in Fig. 6, which is similar to buildings.

V. Conclusion

In this article, we rethink RS semisupervised binary segmentation tasks from the perspective of the imbalanced distribution of the foreground and background, which severely degrades the self-training of unlabeled data on segmentation models. To alleviate this problem, we propose a novel AdaptMatch-based segmentation framework for RS semisupervised binary segmentation. The proposed AdaptMatch can select high-quality pseudo-labels and thus create a relatively balanced self-training through the use of adaptive thresholds for foreground and background. In addition, AdaptMatch is a plug-and-play module that can be easily combined with various CNN and ViT backbones. In comparison with some other state-of-the-art SSS methods, our AdaptMatch achieves superior and more robust performance on different RS binary segmentation tasks at various labeled ratios, demonstrating its effectiveness and generalizability.

In future work, some more fine-grained thresholds related to semantic information could be further designed to obtain more accurate pseudo-labels. Taking building footprint extraction as an example, the buildings could be separated into several types, each with its own respective thresholds.

References

[1] F. Zhang, Y. Shi, and X. X. Zhu, "Domain-agnostic domain adaption for building footprint extraction," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2022, pp. 318–321.

[2] Q. Li, L. Mou, Y. Hua, Y. Shi, and X. X. Zhu, "CrossGeoNet: A framework for building footprint generation of label-scarce geographical regions," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 111, Jul. 2022, Art. no. 102824.

[3] X. Lu et al., "Multi-scale and multi-task deep learning framework for automatic road extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9362–9377, Nov. 2019.

[4] H. Xu, H. He, Y. Zhang, L. Ma, and J. Li, "A comparative study of loss functions for road segmentation in remotely sensed road datasets," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 116, Feb. 2023, Art. no. 103159.

[5] Z. Yuan, L. Mou, Z. Xiong, and X. X. Zhu, "Change detection meets visual question answering," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5630613.

[6] Q. Wang, Z. Yuan, Q. Du, and X. Li, "GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 3–13, Jan. 2019.

[7] Z. Lv, H. Huang, W. Sun, M. Jia, J. A. Benediktsson, and F. Chen, "Iterative training sample augmentation for enhancing land cover change detection performance with deep learning neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 21, 2023, doi: 10.1109/TNNLS.2023.3282935.

[8] Z. Lv, P. Zhong, W. Wang, Z. You, and N. Falco, "Multiscale attention network guided with change gradient image for land cover change detection using remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[9] Z. Lv, P. Zhong, W. Wang, Z. You, J. A. Benediktsson, and C. Shi, "Novel piecewise distance based on adaptive region key-points extraction for LCCD with VHR remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5607709.

[10] O. Ghorbanzadeh et al., "The outcome of the 2022 Landslide4Sense competition: Advanced landslide detection from multisource satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 9927–9942, 2022.

[11] X. Liu et al., "Feature-fusion segmentation network for landslide detection using high-resolution remote sensing images and digital elevation model data," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4500314.

[12] R. M. Mateos et al., "Integration of landslide hazard into urban planning across Europe," *Landscape Urban Planning*, vol. 196, Apr. 2020, Art. no. 103740.

[13] M. R. Guarini, P. Morano, and F. Sica, "Eco-system services and integrated urban planning. a multi-criteria assessment framework for ecosystem urban forestry projects," in *Values and Functions for Future Cities*, 2020, pp. 201–216.

[14] W. Cheng et al., "Ecosystem health assessment of desert nature reserve with entropy weight and fuzzy mathematics methods: A case study of Badain Jaran desert," *Ecol. Indicators*, vol. 119, Dec. 2020, Art. no. 106843.

[15] M. Lourenço, D. Estima, H. Oliveira, L. Oliveira, and A. Mora, "Automatic rural road centerline detection and extraction from aerial images for a forest fire decision support system," *Remote Sens.*, vol. 15, no. 1, p. 271, Jan. 2023.

[16] K. Heidler, L. Mou, C. Baumhoer, A. Dietz, and X. X. Zhu, "HED-UNet: Combined segmentation and edge detection for monitoring the Antarctic coastline," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4300514.

[17] C. M. Albrecht, C. Liu, Y. Wang, L. Klein, and X. X. Zhu, "Monitoring urban forests from auto-generated segmentation MAPS," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2022, pp. 5977–5980.

[18] R. Chen et al., "Monitoring rainfall events in desert areas using the spectral response of biological soil crusts to hydration: Evidence from the Gurbantunggut Desert, China," *Remote Sens. Environ.*, vol. 286, Mar. 2023, Art. no. 113448.

[19] Y. Zou, Z. Zhang, H. Zhang, C.-L. Li, X. Bian, J.-B. Huang, and T. Pfister, "PseudoSeg: Designing pseudo labels for semantic segmentation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=-TwO99rbVRu

[20] Y. Ouali, C. Hudelot, and M. Tami, "Semi-supervised semantic segmentation with cross-consistency training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12671–12681.

[21] X. Chen, Y. Yuan, G. Zeng, and J. Wang, "Semi-supervised semantic segmentation with cross pseudo supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2613–2622.

[22] Z. Wang, Z. Zhao, X. Xing, D. Xu, X. Kong, and L. Zhou, "Conflict-based cross-view consistency for semi-supervised semantic segmentation," 2023, *arXiv:2303.01276*.

[23] Y. Liu, Y. Tian, Y. Chen, F. Liu, V. Belagiannis, and G. Carneiro, "Perturbed and strict mean teachers for semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4248–4257.

[24] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[25] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 17, 2004, pp. 529–536.

[26] D.-H. Lee et al., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, vol. 3, no. 2, p. 896.

[27] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "MixMatch: A holistic approach to semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 5049–5059.

[28] H. Pham, Z. Dai, Q. Xie, and Q. V. Le, "Meta pseudo labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11552–11563.

[29] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10684–10695.

[30] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "ST++: Make self-training work better for semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4258–4267.

[31] Y. Wang et al., "Semi-supervised semantic segmentation using unreliable pseudo-labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4238–4247.

[32] R. He, J. Yang, and X. Qi, "Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6910–6920.

[33] D. Guan, J. Huang, A. Xiao, and S. Lu, "Unbiased subclass regularization for semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9958–9968.

[34] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 596–608.

[35] H. Chen et al., "SoftMatch: Addressing the quantity-quality trade-off in semi-supervised learning," 2023, *arXiv:2301.10921*.

[36] B. Zhang et al., "FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 18408–18419.

[37] L. Yang, L. Qi, L. Feng, W. Zhang, and Y. Shi, "Revisiting weak-to-strong consistency in semi-supervised semantic segmentation," 2022, *arXiv:2208.09910*.

[38] W. Huang, Y. Shi, Z. Xiong, Q. Wang, and X. X. Zhu, "Semi-supervised bidirectional alignment for remote sensing cross-domain scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 195, pp. 192–203, Jan. 2023.

[39] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? The inria aerial image labeling benchmark," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 3226–3229.

[40] Y. Zhang, W. Li, M. Zhang, S. Wang, R. Tao, and Q. Du, "Graph information aggregation cross-domain few-shot learning for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 30, 2022, doi: 10.1109/TNNLS.2022.3185795.

[41] Y. Zhang, W. Li, W. Sun, R. Tao, and Q. Du, "Single-source domain expansion network for cross-scene hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 32, pp. 1498–1512, 2023.

[42] X. Sun, A. Shi, H. Huang, and H. Mayer, "BAS^{4}Net: Boundary-aware semi-supervised semantic segmentation network for very high resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5398–5413, 2020.

[43] J. Wang, C. H. Q. Ding, S. Chen, C. He, and B. Luo, "Semi-supervised remote sensing image semantic segmentation via consistency regularization and average update of pseudo-label," *Remote Sens.*, vol. 12, no. 21, p. 3603, Nov. 2020.

[44] J.-X. Wang, S.-B. Chen, C. H. Q. Ding, J. Tang, and B. Luo, "Semi-supervised semantic segmentation of remote sensing images with iterative contrastive network," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[45] B. Zhang et al., "Semi-supervised deep learning via transformation consistency regularization for remote sensing image semantic segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, pp. 1–15, 2022.

[46] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.

[47] G. French, S. Laine, T. Aila, M. Mackiewicz, and G. Finlayson, "Semi-supervised semantic segmentation needs strong, varied perturbations," 2019, *arXiv:1906.01916*.

[48] X. Zhang, X. Huang, and J. Li, "Semisupervised change detection with feature-prediction alignment," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5401016.

[49] X. Zhang, X. Huang, and J. Li, "Joint self-training and rebalanced consistency learning for semi-supervised change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5406613.

[50] S. Desai and D. Ghose, "Active learning for improved semi-supervised semantic segmentation in satellite images," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1485–1495.

[51] W. G. C. Bandara and V. M. Patel, "Revisiting consistency regularization for semi-supervised change detection in remote sensing images," 2022, *arXiv:2204.08454*.

[52] J. Li, B. Sun, S. Li, and X. Kang, "Semisupervised semantic segmentation of remote sensing images with consistency self-training," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5615811.

[53] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *Proc. 12th Int. Symp. Vis. Comput. (ISVC)*, Las Vegas, NV, USA, 2016, pp. 234–244.

[54] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, Graduate Dept. Comput. Sci., Univ. of Toronto, Toronto, ON, Canada, 2013.

[55] M. Zhou, H. Sui, S. Chen, J. Wang, and X. Chen, "BT-RoadNet: A boundary and topologically-aware neural network for road extraction from high-resolution remote sensing imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 168, pp. 288–306, Oct. 2020.

[56] I. Demir et al., "DeepGlobe 2018: A challenge to parse the Earth through satellite images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 172–17209.

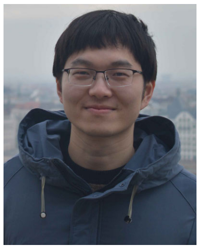
[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[58] B. Baheti, S. Innani, S. Gajre, and S. Talbar, "Eff-UNet: A novel architecture for semantic segmentation in unstructured environment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1473–1481.

[59] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

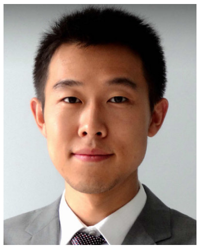
[60] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 3008–3017.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.



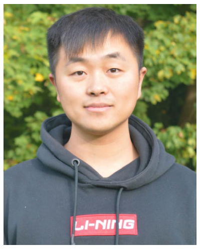
Wei Huang received the B.E. degree in control theory and engineering and the M.E. degree in computer science and technology from the School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with the Technical University of Munich, Munich, Germany.

His research interests include transfer learning, deep learning, and remote sensing.



Yilei Shi (Member, IEEE) received the Dipl.-Ing. degree in mechanical engineering and the Dr.-Ing. degree in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2010 and 2019, respectively.

He is currently a Senior Scientist with the Chair of Remote Sensing Technology, TUM. His research interests include fast solver and parallel computing for large-scale problems, high-performance computing, and computational intelligence, advanced methods on synthetic aperture radar (SAR) and InSAR processing, machine learning, deep learning for a variety of data sources, such as SAR, optical images, and medical images, and partial differential equation (PDE)-related numerical modeling and computing.



Zhitong Xiong (Member, IEEE) received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China, in 2021.

He is currently a Post-Doctoral Researcher with the Chair of Data Science in Earth Observation, Technical University of Munich (TUM), Munich, Germany. His research interests include computer vision, machine learning, and remote sensing.

Xiao Xiang Zhu (Fellow, IEEE) received the M.Sc., Dr.-Ing., and Habilitation degrees in signal processing from the Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

She is the Chair Professor for Data Science in Earth Observation at Technical University of Munich (TUM) and was the founding Head of the Department "EO Data Science" at the Remote Sensing Technology Institute, German Aerospace Center (DLR). From 2019 to 2022, she has co-coordinator of the Munich Data Science Research School (www.mu-ds.de) and the head of the Helmholtz Artificial Intelligence—Research Field "Aeronautics, Space and Transport". Since May 2020, she has been the PI and Director of the International Future AI Laboratory "AI4EO—Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond," Munich. Since October 2020, she has been serving as a Director of the Munich Data Science Institute (MDSI), TUM. She was a Guest Scientist or Visiting Professor at the Italian National Research Council (CNR-IREA), Naples, Italy, Fudan University, Shanghai, China, The University of Tokyo, Tokyo, Japan, and the University of California at Los Angeles, Los Angeles, CA, USA, in 2009, 2014, 2015, and 2016, respectively. She is also a Visiting AI Professor at ESA's Phi-Laboratory. Her main research interests are remote sensing and Earth observation, signal processing, machine learning, and data science, with their applications in tackling societal grand challenges, e.g., global urbanization, UN's SDGs, and climate change.

Dr. Zhu is a member of the Young Academy (Junge Akademie/Junges Kolleg) at the Berlin–Brandenburg Academy of Sciences and Humanities, the German National Academy of Sciences Leopoldina, and the Bavarian Academy of Sciences and Humanities. She serves on the scientific advisory board in several research organizations, among others the German Research Center for Geosciences (GFZ, 2020 to 2023) and the Potsdam Institute for Climate Impact Research (PIK). She is an Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING and serves as the Area Editor for the Special Issue on *IEEE Signal Processing Magazine*.