# Deep Bilateral Filtering Network for Point-Supervised Semantic Segmentation in Remote Sensing Images

Linshan Wu, Leyuan Fang, *Senior Member, IEEE*, Jun Yue, Bob Zhang, *Senior Member, IEEE*, Pedram Ghamisi, *Senior Member, IEEE*, and Min He

***Abstract*—Semantic segmentation methods based on deep neural networks have achieved great success in recent years. However, training such deep neural networks relies heavily on a large number of images with accurate pixel-level labels, which requires a huge amount of human effort, especially for large-scale remote sensing images. In this paper, we propose a point-based weakly supervised learning framework called the deep bilateral filtering network (DBFNet) for the semantic segmentation of remote sensing images. Compared with pixel-level labels, point annotations are usually sparse and cannot reveal the complete structure of the objects; they also lack boundary information, thus resulting in incomplete prediction within the object and the loss of object boundaries. To address these problems, we incorporate the bilateral filtering technique into deeply learned representations in two respects. First, since a target object contains smooth regions that always belong to the same category, we perform deep bilateral filtering (DBF) to filter the deep features by a nonlinear combination of nearby feature values, which encourages the nearby and similar features to become closer, thus achieving a consistent prediction in the smooth region. In addition, the DBF can distinguish the boundary by enlarging the distance between the features on different sides of the edge, thus preserving the boundary information well. Experimental results on two widely used datasets, the ISPRS 2-D semantic labeling Potsdam and Vaihingen datasets, demonstrate that our proposed DBFNet can achieve a highly competitive performance compared with state-of-the-art fully-supervised methods. Code is available at https://github.com/Luffy03/DBFNet.**

***Index Terms*—Bilateral filtering, point annotations, remote sensing, semantic segmentation, weakly-supervised learning.**

This work was supported in part by the National Natural Science Fund of China under Grant 61922029, in part by the Science and Technology Plan Project Fund of Hunan Province under Grant 2019RS2016, and in part by the Key Research and Development Project of Science and Technology Plan of Hunan Province under Grant 2021SK2039. (Corresponding author: Leyuan Fang.)

Linshan Wu is with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China (e-mail: linshanwu@hnu.edu.cn).

Leyuan Fang is with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China, and also with the Peng Cheng Laboratory, Shenzhen 518000, China (e-mail: fangleyuan@gmail.com).

Jun Yue is with the Department of Geomatics Engineering, Changsha University of Science and Technology, Changsha 410114, China (e-mail: jyue@pku.edu.cn).

Bob Zhang is with the PAMI Research Group, Department of Computer and Information Science, University of Macau, Taipa 999078, Macau (e-mail:bobzhang@um.edu.mo).

Pedram Ghamisi is with the Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Helmholtz Institute Freiberg for Resource Technology, 09599 Freiberg, Germany, and also with the Institute of Advanced Research in Artificial Intelligence (IARAI), 1030 Vienna, Austria (e-mail: p.ghamisi@gmail.com).

Min He is with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China (e-mail: hemin@hnu.edu.cn).

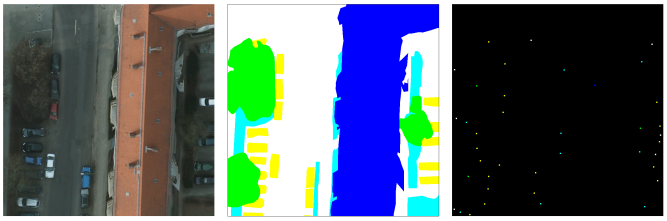



Fig. 1. Illustration of the proposed problem. (a) is a target image, (b) and (c) are the corresponding pixel-level labels and point-level labels, respectively. In this paper, we train the semantic segmentation network using the point annotations (c) instead of the pixel-level labels (b).

## I. Introduction

SEMANTIC segmentation is one of the most fundamental and challenging tasks in remote sensing image (RSI) interpretation. The goal is to assign the corresponding pixel-wise semantic label to each pixel for a given RSI. In particular, semantic segmentation in very-high-resolution (VHR) and large-scale RSIs plays an increasingly significant role in many applications, such as environmental monitoring [1], [2], agriculture [3], [4], urban planning [5], [6], and land cover classification [7].

In RSI segmentation, the typical methods can be generally divided into two groups: handcrafted feature-based methods and deep neural network (DNN) methods. The handcrafted feature-based methods manually design features and classifiers for RSI segmentation. The simple linear iterative cluster (SLIC) algorithm [8] is the most widely used of these methods, and adapts a k-means clustering approach to efficiently generate superpixel features, segmenting the images into small regions grouped by neighboring pixels. Liu [9] proposes a new objective function and a novel graph construction for superpixel segmentation. Radman [10] proposes a robust SVM-HOG model, using GrowCut segmentation to extract handcrafted features. After extracting the handcrafted features, the Markov model [11] and region adjacency graph [12] model are used to merge the most similar adjacent regions. However, spectral features are rather complicated in VHR RSIs, and thus these methods may lead to larger intra-class variance and smaller inter-class variance [13].

With the success of deep learning [14]–[17], semantic segmentation methods based on DNN [18]–[23] have made great progress recently, both in natural scenes and remote

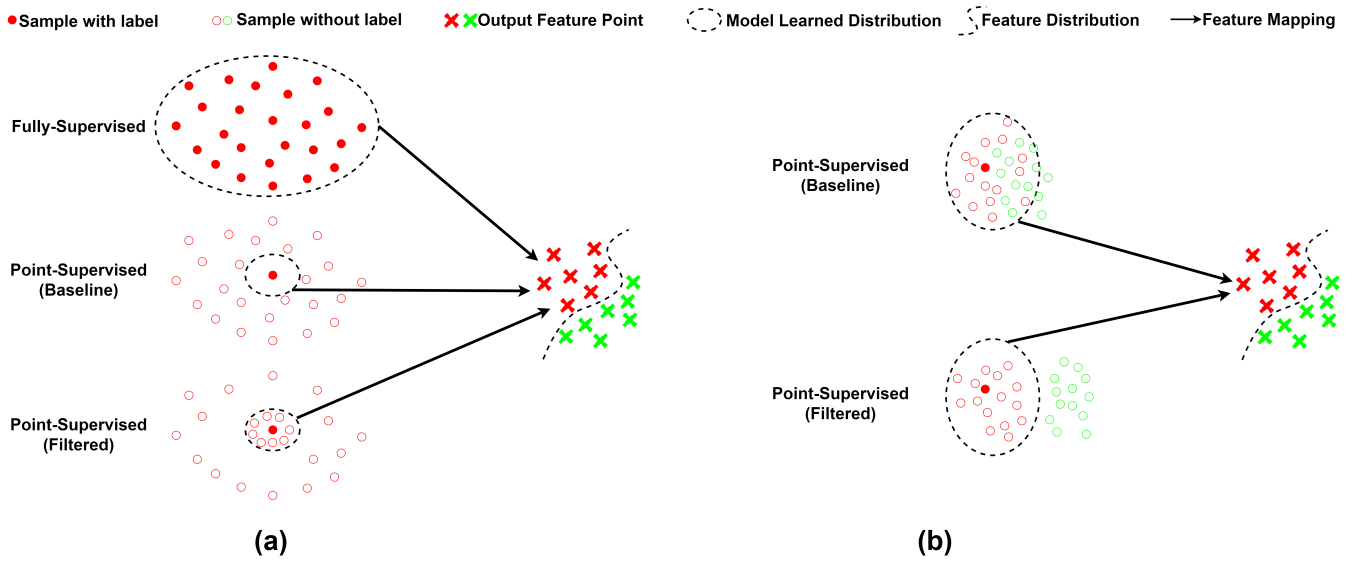



Fig. 2. Issues addressed by the proposed method: (a) DBFNet filters the features so they become closer, aiming to not only map the samples with a label, but also map the nearby and similar samples to specific categories; (b) DBFNet aims to enlarge the distance between the features at the edge, and then distinguishes the features near the boundary without any prior.

sensing scenes. However, unlike the natural images taken at a close range, the scale of RSIs is inconsistent. Especially for small-scale land covers, features may be lost with the decrease of spatial resolution, leading to poor segmentation results. Considering this characteristic of RSIs, some research works construct networks based on multi-scale feature aggregation. Nogueira [24] employs dilated convolution [17] to enhance context information for feature aggregation. Maggiori [25] merges multi-scale features from distinct (shallow or deep) layers. To better refine the boundaries in RSIs, Li [26] introduces a boundary attention module (BA-module) to capture land-cover boundary information from hierarchical features aggregation. In general, the DNN-based methods can surpass the traditional approaches by a large margin.

Most DNN-based semantic segmentation methods rely on a huge number of pixel-level labels for fully-supervised training. However, for large-scale RSIs, collecting accurate annotations of pixel-level labels is very difficult, requiring considerable financial resources and human effort. Recently, to decrease the amount of human intervention needed for training models, weakly-supervised learning (WSL) methods [27]–[29] have been proposed for semantic segmentation. These methods use other kinds of convenient labels, *i.e.*, image-level labels, point annotations, scribbles, and bounding-box. In general, most of the WSL methods are designed for natural images, and few of them are available for RSIs. This is because unlike a target object, which always covers most of the area in a natural image, different categories of objects are often distributed in different regions of the large-scale RSIs. Therefore, it is very difficult to only employ image-level labels to retrieve the location of the objects in RSIs. In addition, creating scribble labels or bounding-box labels is difficult and laborious for large-scale RSIs.

Based on the above considerations, in this paper, we investigate the possibility of training a WSL model with point annotations for semantic segmentation in RSIs. As shown in Fig. 1, we assigned only one point for one object in an RSI for supervision in the training process, which is more convenient to create than other weak annotations, *i.e.*, scribbles and bounding-box. The point annotations can tell us not only the categories of objects the image contains but also the locations of these objects. However, the point annotations are not sufficient for providing strong supervision. Specifically, the weak point annotations may cause two main problems. First, with only one point for one object, the sparse point annotations cannot reveal the complete structure of the objects, which may lead to blurred estimations. In addition, the point annotations lack boundary information, resulting in the wrong segmentation in the boundary area.

To address these two problems, we propose a point-based WSL framework called the deep bilateral filtering network (DBFNet) for the semantic segmentation of RSIs. Specifically, deep bilateral filtering (DBF) is inspired by the classical bilateral filtering (BF) algorithm in [30], which aims to smooth images and preserve edges. Our DBFNet extracts deep features with a CNN-based encoder and then performs DBF to transform the deep features. The illustration of our issues is shown in Fig. 2. First, since the target object always contains a smooth internal, and thus a smooth region always belongs to the same category. It is optimal for the distribution of the features in the smooth region to be closer. Therefore, our DBF is employed to filter the deep features using a nonlinear combination of nearby feature values, aiming to encourage the nearby and similar features to become closer aiming to make consistent and high-confidence predictions within the smooth region. In addition, similar to the traditional BF, which can smooth images while preserving the edges, our DBF is employed to enhance the boundary distinction in high-dimension feature

levels. Our DBF filters the features and enlarges the distance between the features on different sides of the edge, aiming to distinguish the features near the boundary, thus achieving better estimations in the boundary area.

The remainder of this paper is organized as follows: Section II reviews related works. Section III describes the details of our proposed method. Section IV conducts experiments to verify the effectiveness of the proposed method and compare its performance with that of other methods. Finally, we discuss the conclusions of the paper in Section V.

## II. Related Work

### A. Weakly-Supervised Semantic Segmentation

Weakly-supervised semantic segmentation has also achieved great progress, with several methods proposed recently. Most of these methods focus on the most challenging problem by only using image-level labels for semantic segmentation. The main difficulty is recovering the precise spatial information from only image-level labels. To this end, most existing works usually rely on a class activation map (CAM) [31]: the CAM is used to identify the most discriminative regions of objects while cross-image semantic similarities and differences are exploited to constrain the predicted CAMs. Based on CAM, SEC [27] introduces three loss functions called seeding, expansion, and boundary constraint losses to expand the initial seeds and train the segmentation model. Some other methods [28], [29], [32], [33] have further developed the region growing strategy to improve the quality of segmentation labels.

Furthermore, works in [34], [35] utilize point annotations to train segmentation models that can achieve object segmentation for images and videos. Introducing extra information for well-defined loss functions and backpropagation can be another potential way for weakly-supervised learning. Works in [36], [37] propose to introduce the low-level affinity of the original images to supervise the segmentation predictions, which have achieved promising results in natural scenes. However, since different categories of objects are often distributed across large-scale RSIs, this low-level affinity is not obvious in large-scale RSIs. Thus, for point-supervised semantic segmentation in RSIs, it is very difficult to utilize this kind of prior information to design well-defined loss functions for effective backpropagation.

More recently, several works [38]–[40] have adopted a multi-stage approach, *i.e.*, first generating coarse pseudo segmentation labels, then using the pseudo labels to train existing segmentation models. These methods tend to be recursive-style, employing multiple models, training cycles, and off-the-shelf saliency methods. The typical AffinityNet [29] learns semantic affinities among adjacent pixels and then generates high-quality pseudo labels by transferring the semantics of known pixels to their adjacent unknown pixels. In addition, works in [41]–[44] have introduced the single-stage semantic segmentation model, but the results did not surpass the state-of-the-art multi-stage methods.

WSL methods are also used in RSI interpretation. Li [45] proposes a weakly-supervised building extraction framework that utilizes image-level tags to extract CAM maps for supervision. Lian [46] proposes a patch-based DNN supervised by point annotations for road segmentation. However, these previous works are proposed for simple two-class segmentation tasks, which cannot be applied to multi-class semantic segmentation tasks. Hua [47] further utilizes scribble and polygon annotations to train segmentation models, but obtaining scribble and polygon annotations is very laborious. Semi-supervised learning methods [48]–[51] are also widely applied in RSI semantic segmentation, which leverage partial fully-annotated labeled images and adequate unlabeled images for training. However, it is very expensive to obtain the fully-annotated labeled images for training a model. In this work, we explore the potential of weak point annotations for semantic segmentation in RSIs.

Note that, the accuracy of the aforementioned methods is much lower than that of the fully-supervised learning (FSL) methods, and may not be sufficient for practical applications.

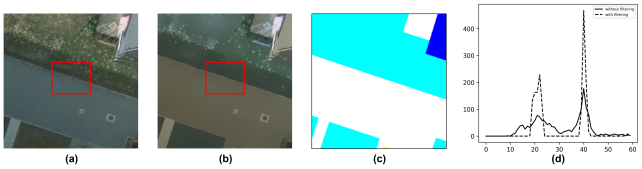



Fig. 3. Images before bilateral filtering (a) and after bilateral filtering (b). (c) is the corresponding ground truth segmentation label. Please see the boundary between the road and the low vegetation, shown in the red box. (d): Histogram (solid) of image intensities is without filtering for (a), and histogram (dash) represents (b) with filtering. As a result, we conclude that the BF can smooth similar regions while preserving the edges.

### B. Bilateral filtering

Bilateral filtering (BF) was first proposed in [30] as a method to smooth images and preserve edges through a non-linear combination of nearby pixel values. The BF combines gray levels or colors based on both their geometric closeness and their photometric similarity and prefers near values to distant values in both domain and range. A simple case of bilateral filtering is shift-invariant Gaussian filtering, in which both the closeness function $c(\varepsilon, x)$ and similarity function $s(\phi, f)$ are Gaussian functions of Euclidean distance between their arguments. More specifically, $c$ is radially symmetric as:

$$c(\varepsilon, x) = e^{-\frac{1}{2}(\frac{d(\varepsilon,x)}{\sigma_d})^2} \tag{1}$$

where

$$d(\varepsilon, x) = d(\varepsilon - x) = \| \varepsilon - x \| \tag{2}$$

is the Euclidean distance between $\varepsilon$ and $x$. The similarity function $s$ is perfectly analogous to $c$:

$$s(\phi, f) = e^{-\frac{1}{2}(\frac{\delta(\phi,f)}{\sigma_r})^2} \tag{3}$$

where

$$\delta(\phi, f) = \delta(\phi - f) = \| \phi - f \| \tag{4}$$

is a suitable measure of distance between the two intensity values $\phi$ and $f$. The terms $\sigma_d$ and $\sigma_r$ represent the geometric spread and photometric spread, respectively, and can be

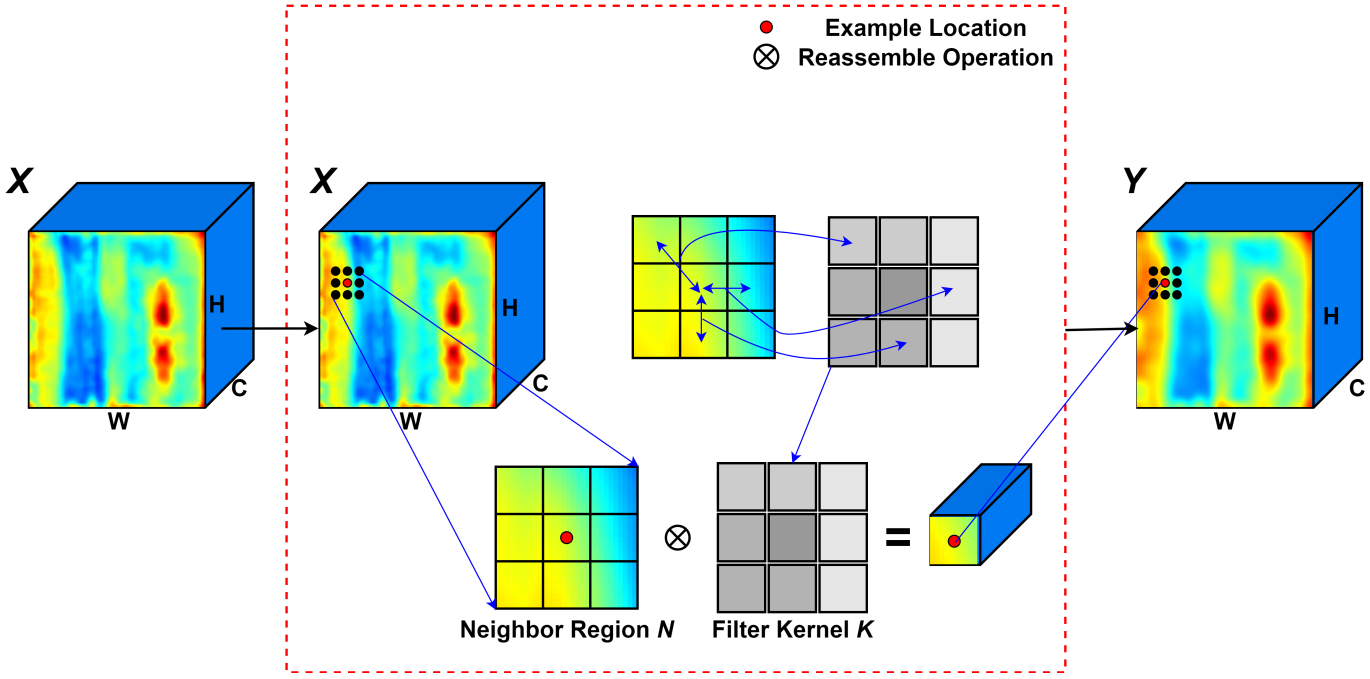



Fig. 4. The framework of the DBF module. A feature map with size $C \times H \times W$ is transformed in this figure.

calculated or defined manually. Fig. 3 shows how BF affects an image. As can be seen, BF encourages the values of the pixels in the smooth region to get closer, while preserving the sharp edge by enlarging the distance between the pixels from different sides of the edges.

In this paper, we propose a Deep Bilateral Filtering Network (DBFNet) for point-supervised semantic segmentation in RSIs, which incorporates the traditional BF technique into DNN. However, it is inappropriate to combine the traditional BF with the DNN in a straightforward manner. Specifically, the main problems for such a straightforward combination are as follows:

1) The traditional BF employs a Gaussian filter with a closeness function $c$ to measure the closeness between different pixels in the whole image. However, this function needs to calculate all of the pixels across the whole image, which requires a huge amount of costly computation to train a large number of images with DNN. Thus, a straightforward combination of traditional BF into DNN is infeasible and inappropriate.
2) The traditional BF method filters images with original sizes. However, DNNs always extract multi-scale features with different sizes. Thus, the straightforward combination of traditional BF and DNN cannot perform effective filtering for these multi-scale features. It is important to design specific filtering methods to adapt these multi-scale features.
3) The traditional BF is a time-consuming iterative process, which would entail a huge amount of costly computation to train a large number of images with DNN. This is another reason why it is inappropriate to incorporate traditional BF into DNN using a straightforward approach. As in the development of many algorithms in this field, it is important to consider the balance between time consumption and filtering iterations.

Thus, instead of utilizing straightforward combination, we have designed several specific mechanisms in our proposed DBFNet to address the above three issues, as details in Section III.

## III. Proposed DBFNet Method

In this section, we first introduce our observations and the overall ideas of our proposed DBFNet. Then, we give the details of the proposed DBFNet for point-supervised semantic segmentation. Finally, we introduce how to train the proposed models. The framework of the DBFNet is shown in Fig. 4.

### A. Underlying Concept of DBFNet

In this work, we focus on the problem of using only point annotations to train a semantic segmentation model. As shown in Fig. 1, the training dataset contains input images with corresponding point annotations. When assigning one point for one object, there are only a few points in the labels that contain information useful for supervision; the rest are considered to be background without any information. Therefore, the key challenge lies in exploiting the extremely small amount of information from the point annotations. Inspired by the BF algorithm [30], we try to address the problem in two ways. The illustration of the issues that we propose to solve is shown in Fig. 2.

First, as mentioned in section I, sparse point annotations cannot reveal the complete structure of objects. However, we find that in RSIs, the internal structure of a complete object is almost smooth, which means that within a smooth region, these similar pixels can be considered to be in the

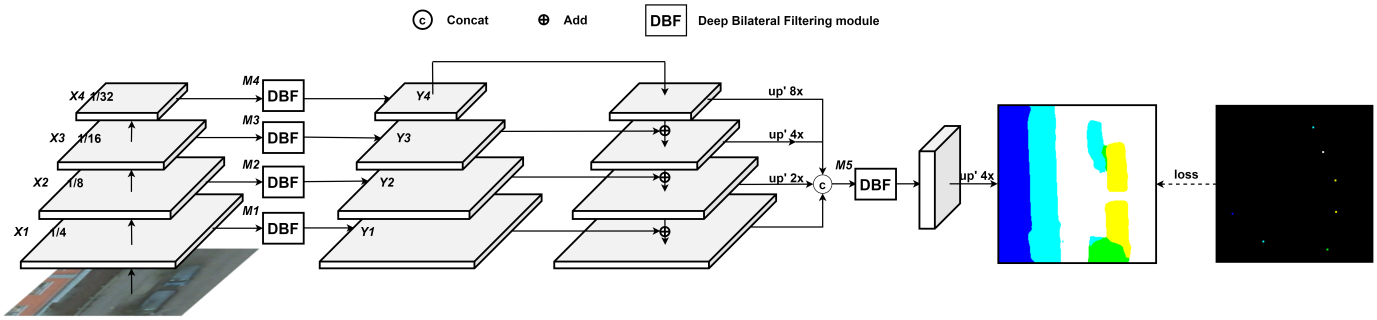



Fig. 5. The overall framework of our proposed DBFNet.

same category. Therefore, we encourage the distributions of the learned deep features in the smooth region to be closer, enabling a consistent and high-confidence prediction with the spatial-closed and value-similar pixels. The key point is, unlike the fully-supervised learning methods in which all samples with corresponding labels can be mapped to a specific category, in point-supervised learning, only a single sample with point annotation can be assigned to a particular category. Our DBFNet can smooth the model-learned feature while simultaneously discarding the nuisance values. As shown in Fig. 2(a), we perform our proposed DBFNet to aggregate the nearby and similar features using a nonlinear combination of nearby feature values, achieving a closer model learned distribution. By enforcing closed distributions, both the sample with point annotation and also the nearby and similar samples can be mapped to the semantic category represented by the point annotation.

Second, without any boundary information from our point annotations, it is difficult to refine the complete and sharp boundaries of objects. The classical BF algorithm [30] can preserve the edges by considering geometric closeness and similarity of pixels. Inspired by this observation, our proposed DBFNet applies the BF to the deeply learned representation with the intention of enlarging the distance between the transformed features on different sides of the edge. As shown in Fig. 2(b), DBFNet is applied to filter and differentiate the features at the edge, aiming to distinguish the boundary. In addition, compared with the conventional BF, DBFNet conducts bilateral filtering at a high-dimensional feature level, which can obtain more robust representations. In this case, we decrease the influence of the lack of boundary information and improve the boundary refinement in the segmentation map.

### B. Architecture of DBFNet

*1) Deep Bilateral Filtering Module:* Given an input image, we first extract deep CNN features $\boldsymbol{X} \in R^{C\times H\times W}$ with the encoder network, where $C$ indicates the number of channels and $H$ and $W$ represent the height and width, respectively. The DBF module then converts $\boldsymbol{X}$ to filtered features $\boldsymbol{Y} \in R^{C\times H\times W}$, as shown in Fig. 4. Note that the size of the transformed feature $\boldsymbol{Y}$ is the same as $\boldsymbol{X}$.

Unlike the classical BF method [30], we do not compute the closeness function for all the positions in $\boldsymbol{X}$, which consumes a great deal of computation and GPU memory. Instead, we adopt a dilated $3 \times 3$ convolutional operation as the filter to capture the contextual information for the objects. Similar to the work in [23], we utilize different rates of convolution, where the rates relate to the distance $d$ between the nearby positions and the target neighborhood center $p$. The filters with different $d$ allow us to arbitrarily enlarge the view and easily consider multi-scale objects.

After collecting the values from nearby positions, we compute the similarity $s$ between the target neighborhood center and its neighbor positions $\xi$. More specifically,

$$s = e^{(-d(\xi,p))}, \tag{5}$$

where

$$d(\xi,p) = \| \xi - p \| \tag{6}$$

where $d$ represents the sum of $C$ channels' Euclidean distance between the nearby positions $\xi$ and the target neighborhood center $p$. Note that we do not calculate the variance of the neighborhood values, since it is time-consuming in high-dimensional space. Finally, the values of $s$ are used as the weights to build the filter kernel $\boldsymbol{K}$.

With the filter kernel $\boldsymbol{K}$, the DBF module reassembles the features within a local neighbor region $\boldsymbol{N}$ via the function $\phi$. We adopt a simple form of $\phi$, which is just a weighted sum operator. For the target neighborhood center $p$ and the corresponding neighbor region $\boldsymbol{N}$ with $3 \times 3$ positions, the reassembly is shown as follows:

$$p^{'} = \frac{1}{n}\sum_{i=1}^{n} s_i \cdot \xi_i, \tag{7}$$

where in a $3 \times 3$ size region, $n$ is set as eight. The terms $s_i$ and $\xi_i$ represent the weight and the value of the $i_{th}$ nearby position, respectively. The $p^{'}$ is the new target neighborhood center transformed from $p$.

With the filter kernel $\boldsymbol{K}$, each position in the region $\boldsymbol{N}$ contributes to the target center $p^{'}$ differently, based on the different similarity $s$. The feature $\boldsymbol{Y}$ filtered by our DBF can be more discriminative than the original feature $\boldsymbol{X}$, since the information from relevant points in a neighbor region $\boldsymbol{N}$ can be effectively utilized. In addition, as mentioned in Section III-A, our DBF module can cluster the nearby and similar features while preserving the edges of the objects in high-dimension space, thus obtaining more coherent segmentation predictions.

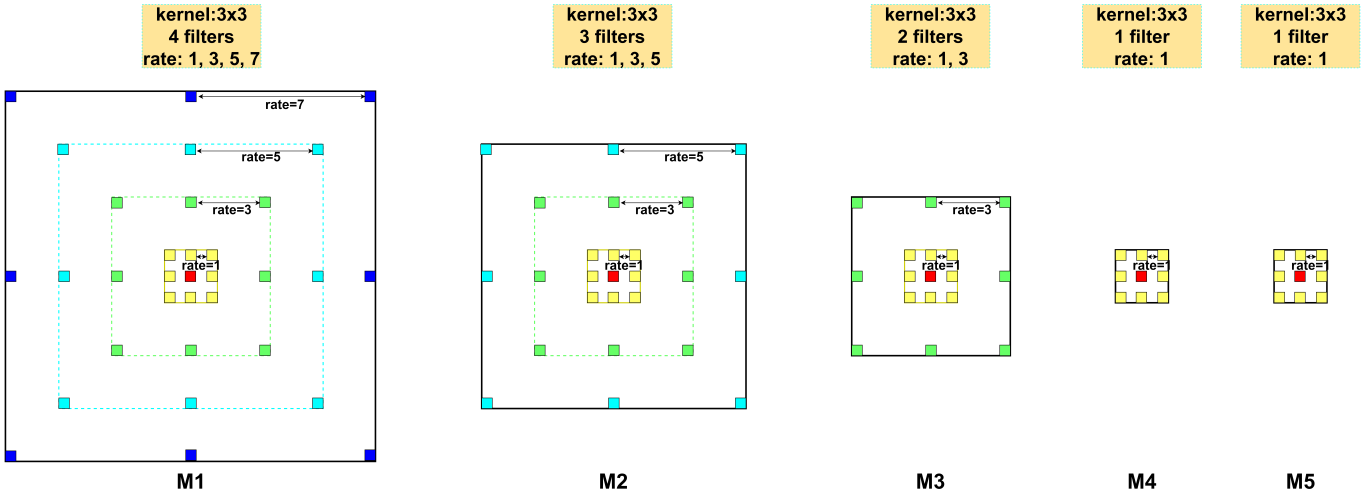



Fig. 6. The settings of the DBF modules in the training process.

*2) Multi-scale DBFNet with FPN:* To adapt to different scale objects in remote sensing images, we propose to utilize multi-scale DBFNet with the feature pyramid network (FPN) [52]. As our encoder, the FPN can extract multi-scale and rich semantic features.

Specifically, the FPN extracts four feature maps $\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3}$, and $\boldsymbol{X_4}$ from the input image; their respective resolutions are $1/4, 1/8, 1/16, 1/32$ of the input resolution. Next, each of the four feature maps is transformed by our DBF module independently, as shown in Fig. 5. Since these feature maps are in different resolutions, the transformed features $\boldsymbol{Y_1}, \boldsymbol{Y_2}, \boldsymbol{Y_3}$, and $\boldsymbol{Y_4}$ also have different scales. Next, we interpolate $\boldsymbol{Y_2}, \boldsymbol{Y_3}$,, and $\boldsymbol{Y_4}$ to the size of $\boldsymbol{Y_1}$, and fuse these features by concatenation. After concatenation, we employ another DBF module to refine the final segmentation map.

Since these feature maps have different scales, in the training process, we apply different settings of DBF module to different-scales feature maps. Fig. 6 shows the training settings for the DBF modules $\boldsymbol{M_1}, \boldsymbol{M_2}, \boldsymbol{M_3}, \boldsymbol{M_4}$,and $\boldsymbol{M_5}$, where $\boldsymbol{M_1}, \boldsymbol{M_2}, \boldsymbol{M_3}$, and $\boldsymbol{M_4}$ are the DBF modules applied to $\boldsymbol{X_1}, \boldsymbol{X_2}, \boldsymbol{X_3}, \boldsymbol{X_4}$ and $\boldsymbol{M_5}$ represents the last DBF module. For the higher-resolution feature map (*e.g.*, $\boldsymbol{X_1}$), we utilize more filters of multi-rates to capture more information. For the low-resolution feature map (*e.g.*, $\boldsymbol{X_4}$), we employ a small-rate filter to build the corresponding DBF module. Note that to accelerate the prediction, we set only one filter with one rate for all of the DBF modules in the testing process. Hence, the settings for the DBF modules are different in the training process are different from those in the testing process.

In addition, the proposed DBF can be iterative as the BF, which means we can set different iterations for the DBF. However, we should note that more iterations will consume more time, so it is vital to balance the trade-off between iterations and time. In the training process, iteration is set to 1 time for quicker training, while 3 times is the best parameter in the testing process.

### C. Training of DBFNet

*1) Loss function:* For semantic segmentation, we adopt a standard cross-entropy loss function on the final predictions. The segmentation loss $L_{seg}$ is denoted as:

$$\begin{aligned} L_{seg} &= \sum_{i,j} CE(y, y^{'}) \\ &= -\frac{1}{n}\sum_{j}^{num}\sum_{i}^{cls}[y_{ji}log(y^{'}_{ji}) + (1-y_{ji})log(1-y^{'}_{ji})], \end{aligned} \tag{8}$$

where $num$ denotes the total number of samples with labels, and $cls$ represents the total number of categories. In addition, $y$ and $y^{'}$ represent the segmentation ground truth (GT) and the segmentation predictions, respectively. Note that our GT is the point annotations; most of the samples are non-labeled backgrounds. Therefore, we only calculate the loss with the labeled pixels and ignore the background, which means $num$ is a small number.

In addition, the point annotations can tell us all of the categories in the input image. We then set a penalty $L_{penal}$ to enforce the network not to predict non-existent categories, which means the model should not predict a building when the image does not contain the building category. The penalty $L_{penal}$ is described as:

$$L_{penal} = -\sum_{i}^{cls}(1-y_i)\sum_{j}^{num} log(1-y_{ji}). \tag{9}$$

Therefore, the total loss $L_{total}$ is combined by $L_{seg}$ and $L_{penal}$ as

$$L_{total} = L_{seg} + L_{penal} \tag{10}$$

*2) Recursive learning:* Recursive training (RL) is a common training strategy in weakly-supervised semantic segmentation, in which the trained model is applied to the training dataset recursively; the output of the network is closer to the real label than the original weakly-supervised model's

predictions. As described in Section II-C, most WSL methods for semantic segmentation employ multi-stage training by using multiple training rounds.

As an option, in this paper, we use RL to refine the segmentation predictions of our DBFNet. The first step is to use point annotations to train the DBFNet. The DBFNet can generate pseudo labels for the training dataset. The second step is to train a standard semantic segmentation network utilizing the generated pseudo labels, since the pseudo labels can provide more information than the original point annotations. The newly trained model from the second step can be used to generate new pseudo labels, which means the RL can work recursively. However, the generated pseudo labels inevitably have a great deal of noise. Therefore, the RL strategy can only provide limited improvement in accuracy. We will discuss the RL's impact in detail in the following section.

## IV. Results of Experiments

To verify the effectiveness of the proposed method, we conducted extensive experiments on two well-known RSI semantic segmentation benchmarks: the Potsdam and Vaihingen semantic labeling datasets released by ISPRS 2D Semantic Labeling Challenge. In this section, we first introduce the datasets, the evaluation metrics settings, and the implementation details. Then we perform detailed extensive ablation experiments on the Potsdam dataset. Finally, we report our results on the two datasets.

### A. Datasets

*1) Potsdam:* The Potsdam dataset consists of 38 tile aerial images ($6000 \times 6000$ pixels) with NIR-R-G-B channels together with DSM and normalized DSM. Note that we only use the R-G-B bands for training and testing. There are 24 images in the training set and 14 in the test set. This dataset includes five foreground object classes and one background class.

*2) Vaihingen:* The Vaihingen dataset contains 33 orthorectified image tile mosaics with three spectral bands (red, green, and near-infrared). Each image has a corresponding DSM with the same spatial resolution of 9cm. Among them, 16 tiles with GT labels are used for training and the remaining 17 images for testing; this dataset has the same six categories as the Potsdam dataset.

### B. Evaluation Metrics

To evaluate the performance of the proposed method, we employed the overall accuracy OA and the F1 scores for the five foreground object classes with the following formulas:

$$OA = \frac{TP + TN}{Num} \tag{11}$$

$$OA = \frac{TP}{TP + FP} \tag{12}$$

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

TABLE I
Abalation Study for DBFNET on the Potsdam Test Dataset

| Method | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | mF1 (%) | OA (%) |
|---|---|---|---|---|---|---|---|
| Baseline (FPN) | | | | | | 69.27 | 69.54 |
| DBFNet | ✓ | | | | | 78.32 | 78.73 |
| DBFNet | ✓ | ✓ | | | | 79.85 | 80.03 |
| DBFNet | ✓ | ✓ | ✓ | | | 82.13 | 82.98 |
| DBFNet | ✓ | ✓ | ✓ | ✓ | | 83.47 | 84.28 |
| DBFNet | ✓ | ✓ | ✓ | ✓ | ✓ | **85.53** | **86.71** |

[1] Note: $M_1, M_2, M_3, M_4, M_5$ are the 5 DBF modules shown in Fig. 5.

TABLE II
Performance Comparisons by Backbone for the Results on the Test Dataset of Potsdam

| Backbone | mF1 (%) | OA (%) |
|---|---|---|
| ResNet-18 | **85.53** | **86.71** |
| ResNet-34 | 84.61 | 85.88 |
| ResNet-50 | 83.85 | 85.25 |
| ResNet-101 | 82.91 | 84.04 |

$$F_1 = (1 + \beta^2) \times \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}, \tag{14}$$

where $TP, TN, FP$, and $FN$ are the number of true positives, true negatives, false positives, and false negatives, respectively. $Num$ represents the total number of pixels in the dataset. The equivalence factor between precision and recall is denoted by $\beta$, which is set to 1. Precision reports the number of positive category predictions that belong to the positive category, and recall reports the number of positive category predictions made out of all positive samples in the datasets. The mean F1 score (m-F1) denotes the average of F1 scores over all categories.

### C. Experimental Settings

*1) Experimental environment:* All experiments were processed on a server computer with an NVIDIA GeForce RTX 2080Ti GPU (11 GB). The implementation of the framework was based on the open-source toolbox Pytorch [53].

*2) Data preprocessing:* Due to the memory limitation of the computer and graphics processing unit (GPU), we cut the patch of the original training samples. For the Potsdam dataset, the sliding window method was used to cut the original images from the top left to the bottom right without overlap, thereby obtaining a final training image size of $256 \times 256$ pixels. For the Vaihingen dataset, to obtain sufficient data, we cut the original images with an overlap rate of 50%. In the test phase, we adopted two methods of data augmentation: random flipping and random rotating. To speed up the convergence of the network and increase the generalization ability of the

TABLE III
Performance Comparisons with Recursive Learning and the Results from the Potsdam Test Dataset

| Method | mF1 (%) | OA (%) |
|---|---|---|
| DBFNet | 85.53 | 86.71 |
| + RL (round 1) | **87.19** | **88.30** |
| + RL (round 2) | 87.04 | 88.12 |
| + RL (round 3) | 86.91 | 87.95 |

TABLE IV
EFFICIENCY COMPARISONS WITH ASPP, PPM, SA, RCCA, OCR, ISA, CAA, RSA, AND DBF MODULE WHEN PROCESSING INPUT FEATURE MAP OF SIZE ($1 \times 2048 \times 128 \times 128$) DURING INFERENCE STAGE

| Method | Par(M△) | Mem(MB△) | GFLOPS(△) |
|---|---|---|---|
| ASPP [54] | 15.1 | 284 | 503 |
| PPM [55] | 22.0 | 792 | 619 |
| SA [56] | 10.5 | 2168 | 619 |
| RCCA [57] | 10.6 | 427 | 804 |
| OCR [58] | 10.5 | 202 | 354 |
| ISA [59] | 11.8 | 252 | 386 |
| CAA [60] | 9.3 | 283 | 148 |
| RSA [60] | 3.8 | 110 | 144 |
| **DBFNet (Ours)** | **4.2** | **84** | **105(train) 152(test)** |

Note: For DBFNet, training time and testing time are different.

model, we conducted mean normalization and standardization on the input data as in previous works.

*3) Implementation details:* We used ResNet [14] pre-trained on ImageNet [16] as our backbone. We trained the models with mini-batch stochastic gradient descent and employed a learning rate of 0.001 with the momentum set to 0.9 and weight decay to 0.0005 based on the Adam optimizer [64]. For the training phase only, we further adopted horizontal and vertical flipping to overcome overfitting.

### *D. Experiments on Potsdam Dataset*

*1) Ablation study for the DBFNet:* In the proposed DBFNet, five DBF modules are employed to transform the feature maps learned by our deep neural network. To further verify the performance of the DBF modules, we conducted extensive experiments with different settings, as shown in Table I. We used ResNet-18 as the backbone of our DBFNet in this experiment.

As shown in Table I, the proposed DBF modules can achieve a remarkable improvement compared with the baseline model FPN (ResNet-18). We observed that the first DBF module M1 yields a result of 78.32% in m-f1 and 78.73% in OA, which yields a 9.05 percentage point and 9.19 percentage point improvement in m-F1 and OA, respectively. Furthermore, when more DBF modules are adopted, the performance of our network is further enhanced. Finally, integration of all five DBF modules outperforms the other settings and obtains the improvements of 16.26 and 17.17 percentage points in m-F1 and OA, respectively, compared with the baseline. This demonstrates that our approach can greatly improve semantic segmentation by filtering the deep features.

Traditionally, a powerful backbone can extract more effective features for a specific visual task. In our proposed DBFNet, we use ResNet-18, ResNet-34, ResNet-50, and ResNet-101 in [14] to investigate the backbone's effect on the segmentation predictions, as shown in Table II.

As can be seen, the deeper backbones achieved lower accuracy. The main reason for this might be the weakness of the point annotations, as in our WSL framework, the point annotations can only provide limited supervision, which the model with more parameters may overfit easily. A well-generalized DNN depends on the ratio of the model parameters to the size of the training samples, where a lower ratio always means better generalization properties [65], [66]. The ratio can be calculated as follows:

$$ratio = \frac{parameters\ of\ models}{training\ samples} \tag{15}$$

The design of DNN in FSL methods, where the training samples are fully-annotated, is discussed in [65], [66]. In our WSL task, only a few training samples are annotated. Since we do not utilize other loss functions for supervision, with a standard CE loss used in our DBFNet, only a few annotated training samples are incorporated into the backpropagation, which is presented in Section III-C. Thus, in our WSL method, deeper backbones with myriad parameters will make the ratio higher, thereby reducing the generalizability of the DBFNet. However, in FSL methods, all training samples are annotated, and the size of annotated training samples is much larger than the parameters of the model, which makes the ratio very low. Thus, with adequate annotated training samples, the model will not overfit easily. In this case, FSL methods with deeper backbones can learn more robust features and achieve better performance. Therefore, for our proposed DBFNet, we selected ResNet-18 as the backbone.

As described in Section III-D, we employ RL to refine the segmentation predictions from our DBFNet. First, we train the DBFNet to generate pseudo labels for the training dataset. As an optional item, we utilize the dense CRF [23] to refine the pseudo labels. For the second step, we train a standard semantic segmentation network utilizing the generated pseudo labels. In addition, in the second step we conducted extensive experiments to investigate how many rounds recursive learning yield the best result. Note that the standard network is built with ResNet-18.

The effect of recursive learning is shown in Table III. We can see that as a common strategy for WSL, recursive learning can improve the accuracy of our framework slightly. However, we find that adding more training rounds does not lead to higher accuracy. The reason is that the deep networks may easily overfit to the noisy pseudo labels. As a result, we only took one training round for the RL in our experiment.

*2) Comparison with context aggregation modules and attention modules:* To validate the efficiency of our DBF module, we first conducted a fair comparison with several well-verified context aggregation approaches: ASPP in DeeplabV3+ [54], PPM in PSPNet [55], SA in nonlocal network [56], RCCA in CCNet [57], OCR in OCRNet [58], ISA in [59], and CAA, RSA in HMANet [60]. All the experiments above were conducted under the same settings for fair comparisons. We investigated these modules in terms of efficiency, including model parameters, GPU memory, and computation cost. Note that computation cost is measured with GFLOPs. The implementation settings of our DBF module are described in section IV.C(3). Specifically, for the time consumed by computation, we take both the training process and the testing process into consideration, since the settings in our DBF modules are different, as described in Section III-C3.

As shown in Table IV, the DBF module increases only a few model parameters, and the very low cost of GPU memory and time consumption are acceptable. Specifically, as in the

TABLE V
PERFORMANCE COMPARISONS WITH WEAKLY-SUPERVISED LEARNING METHODS AND THE RESULTS OF THE POTSDAM TEST DATASET

| Method | Imp. surf. | Building | Low veg. | Tree | Car | mF1 (%) | OA (%) |
|---|---|---|---|---|---|---|---|
| DeeplabV3 + RL | 78.34 | 84.43 | 66.60 | 59.67 | 71.44 | 72.10 | 73.70 |
| PSPNet + RL | 78.80 | 84.22 | 66.53 | 59.98 | 71.67 | 72.24 | 73.77 |
| FPN + RL | 79.45 | 84.54 | 67.01 | 58.39 | 70.84 | 72.05 | 73.68 |
| DeeplabV3 + NormCut-Loss [36] + RL | 88.72 | 91.43 | 77.23 | 74.51 | 76.44 | 81.67 | 82.52 |
| DeeplabV3 + DenseCRF-Loss [37] + RL | 90.55 | 94.14 | 79.60 | 75.67 | 78.21 | 83.63 | 83.85 |
| **DBFNet** | 89.29 | 94.06 | 81.25 | 78.16 | 84.88 | 85.53 | 86.71 |
| **DBFNet + RL** | **90.80** | **95.34** | **83.16** | **79.98** | **86.67** | **87.19** | **88.30** |

TABLE VI
PERFORMANCE COMPARISONS WITH STATE-OF-THE-ART FULLY-SUPERVISED LEARNING METHODS ON THE TEST DATASET OF POTSDAM

| Method | Supervision | Backbone | Imp. surf. | Building | Low veg. | Tree | Car | mF1 (%) | OA (%) |
|---|---|---|---|---|---|---|---|---|---|
| FCN [18] | full | ResNet-18 | 87.61 | 90.93 | 81.09 | 80.72 | 84.93 | 85.06 | 86.16 |
| UNet [19] | full | ResNet-18 | 88.03 | 91.87 | 81.99 | 82.01 | 87.71 | 86.32 | 86.58 |
| SegNet [21] | full | ResNet-18 | 86.29 | 89.75 | 84.89 | 82.71 | 87.60 | 86.25 | 86.91 |
| PSPNet [55] | full | ResNet-18 | 89.72 | 94.08 | 84.25 | 84.79 | 88.17 | 88.20 | 88.73 |
| RefineNet [61] | full | ResNet-18 | 87.67 | 93.95 | 87.16 | 85.39 | 89.58 | 88.75 | 87.01 |
| FPN [52] | full | ResNet-18 | 89.01 | 93.22 | 86.15 | 84.74 | 90.23 | 88.67 | 88.99 |
| DeepLabV3+ [54] | full | ResNet-18 | 90.81 | 95.01 | 85.23 | 84.36 | 89.26 | 88.93 | 89.62 |
| FCN [18] | full | ResNet-101 | 89.27 | 93.07 | 82.25 | 82.29 | 85.58 | 86.50 | 87.37 |
| UNet [19] | full | ResNet-101 | 89.64 | 93.08 | 82.44 | 82.35 | 88.31 | 87.17 | 87.61 |
| SegNet [21] | full | ResNet-101 | 87.96 | 92.72 | 85.25 | 83.91 | 88.17 | 87.60 | 87.93 |
| PSPNet [55] | full | ResNet-101 | 90.61 | 95.30 | 85.41 | 85.84 | 90.38 | 89.53 | 89.45 |
| RefineNet [61] | full | ResNet-101 | 88.12 | 94.28 | 88.73 | 86.54 | 89.94 | 89.52 | 87.24 |
| FPN [52] | full | ResNet-101 | 89.24 | 94.02 | 86.38 | 85.14 | 91.45 | 89.25 | 89.87 |
| DeepLabV3+ [54] | full | ResNet101 | 91.98 | 95.60 | 85.34 | 84.81 | 90.17 | 89.58 | 90.11 |
| HRNet [62] | full | W48 | 93.30 | 96.57 | 88.04 | 87.78 | 91.36 | 91.41 | 91.52 |
| **DBFNet** | weak | ResNet-18 | 89.29 | 94.06 | 81.25 | 78.16 | 84.88 | 85.53 | 86.71 |
| **DBFNet + RL** | weak | ResNet-18 | 90.80 | 95.34 | 83.16 | 79.98 | 86.67 | 87.19 | 88.30 |

TABLE VII
PERFORMANCE COMPARISONS WITH WEAKLY-SUPERVISED LEARNING METHODS AND THE RESULTS OF THE VAIHINGEN TEST DATASET

| Method | Imp. surf. | Building | Low veg. | Tree | Car | mF1 (%) | OA (%) |
|---|---|---|---|---|---|---|---|
| DeeplabV3 + RL | 80.40 | 81.64 | 68.97 | 73.60 | 57.45 | 72.42 | 76.06 |
| PSPNet + RL | 80.02 | 82.15 | 69.22 | 71.37 | 58.02 | 72.15 | 75.84 |
| FPN + RL | 78.30 | 79.21 | 68.55 | 74.26 | 56.21 | 71.30 | 75.03 |
| DeeplabV3 + NormCut-Loss [36] + RL | 86.12 | 87.51 | 70.03 | 79.37 | 68.52 | 78.31 | 79.04 |
| DeeplabV3 + DenseCRF-Loss [37] + RL | 89.71 | 90.11 | 71.38 | 82.65 | 75.49 | 81.87 | 82.18 |
| **DBFNet** | 90.27 | 92.84 | 78.81 | 86.81 | 81.09 | 85.97 | 86.32 |
| **DBFNet + RL** | **90.88** | **93.04** | **79.35** | **87.19** | **81.36** | **86.36** | **86.92** |

TABLE VIII
PERFORMANCE COMPARISONS WITH STATE-OF-THE-ART FULLY-SUPERVISED LEARNING METHODS ON THE TEST DATASET OF VAIHINGEN

| Method | Supervision | Backbone | Imp. surf. | Building | Low veg. | Tree | Car | mF1(%) | OA(%) |
|---|---|---|---|---|---|---|---|---|---|
| FCN [18] | full | ResNet-18 | 87.15 | 88.07 | 71.32 | 85.65 | 72.93 | 81.02 | 83.98 |
| UNet [19] | full | ResNet-18 | 88.70 | 89.59 | 77.90 | 86.54 | 75.71 | 83.69 | 86.37 |
| SegNet [21] | full | ResNet-18 | 85.51 | 89.20 | 78.99 | 83.39 | 85.79 | 84.58 | 85.12 |
| PSPNet [55] | full | ResNet-18 | 91.08 | 89.63 | 79.26 | 85.12 | 86.90 | 86.40 | 87.19 |
| RefineNet [61] | full | ResNet-18 | 88.17 | 92.59 | 82.41 | 85.87 | 86.33 | 87.07 | 86.68 |
| FPN [52] | full | ResNet-18 | 90.01 | 89.95 | 81.18 | 85.71 | 87.46 | 86.86 | 87.04 |
| DeepLabV3+ [54] | full | ResNet-18 | 93.88 | 92.75 | 80.76 | 86.09 | 87.70 | 88.24 | 88.31 |
| FCN [18] | full | ResNet-101 | 88.67 | 90.83 | 76.32 | 86.67 | 74.21 | 83.34 | 86.51 |
| UNet [19] | full | ResNet-101 | 89.54 | 91.06 | 78.81 | 89.08 | 78.73 | 85.44 | 87.03 |
| SegNet [21] | full | ResNet-101 | 86.97 | 91.21 | 80.24 | 84.36 | 89.17 | 86.51 | 85.87 |
| PSPNet [55] | full | ResNet-101 | 93.32 | 90.42 | 80.53 | 86.91 | 87.02 | 87.65 | 88.69 |
| RefineNet [61] | full | ResNet-101 | 88.80 | 91.84 | 83.57 | 86.62 | 87.04 | 87.57 | 86.93 |
| FPN [52] | full | ResNet-101 | 91.03 | 90.22 | 82.01 | 86.89 | 87.78 | 87.59 | 86.32 |
| DeepLabV3+ [54] | full | ResNet101 | 94.34 | 91.35 | 81.32 | 87.84 | 88.14 | 88.60 | 88.91 |
| HRNet [62] | full | W48 | 94.51 | 92.22 | 83.19 | 87.58 | 90.93 | 89.69 | 90.03 |
| **DBFNet** | weak | ResNet-18 | 90.27 | 92.84 | 78.81 | 86.81 | 81.09 | 85.97 | 86.32 |
| **DBFNet + RL** | weak | ResNet-18 | 90.88 | 93.04 | 79.35 | 87.19 | 81.36 | 86.36 | 86.92 |

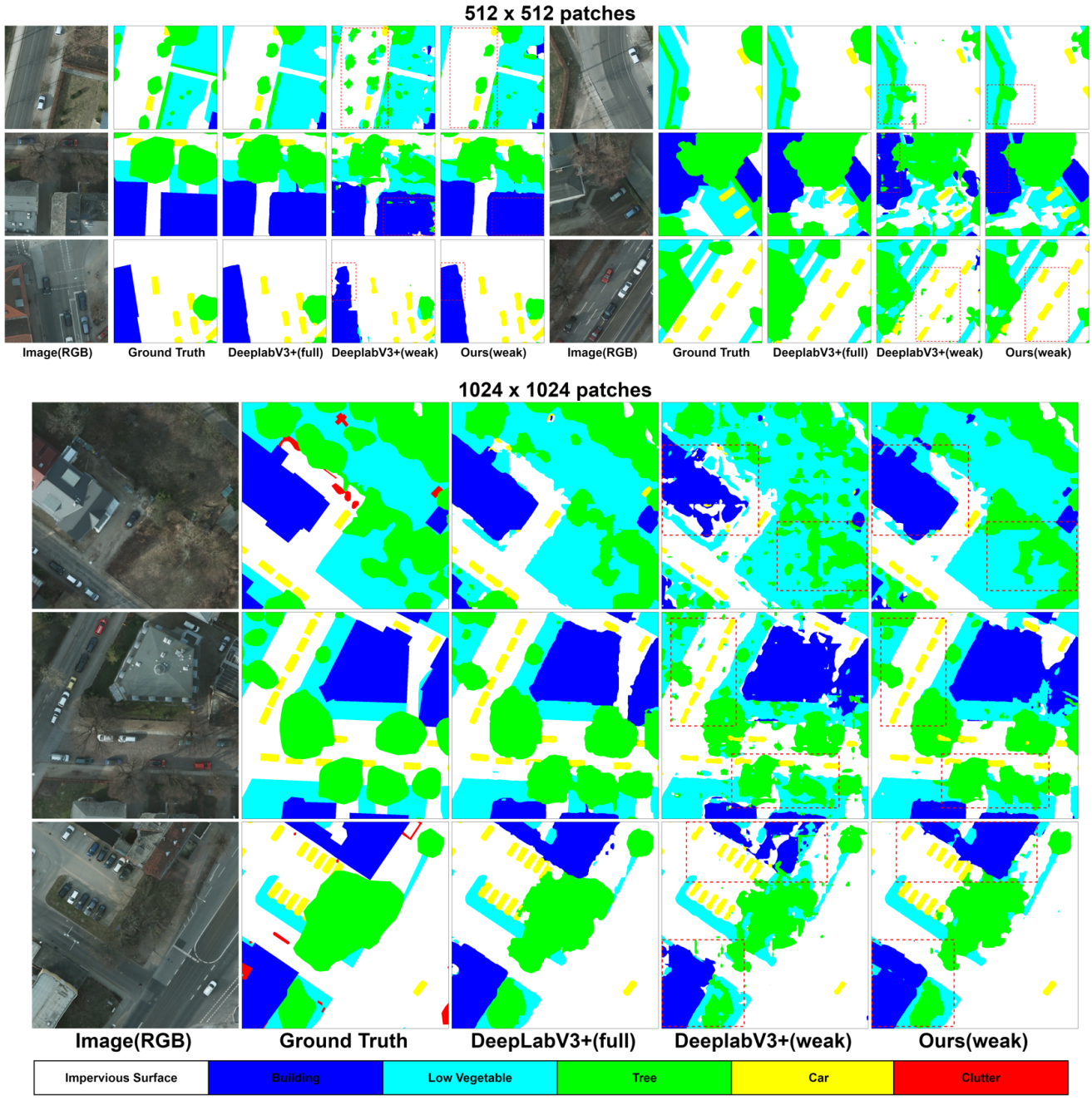



Fig. 7. Qualitative comparisons between the proposed method and other methods, using the Potsdam test dataset.

training process the DBF modules have higher rates and fewer iteration times, so the training process costs less time than the testing process.

*3) Comparison with weakly-supervised learning methods:* To further verify our proposed DBFNet, it is necessary to compare it with other WSL methods. Therefore, we define a baseline method for comparisons: we use point annotations to train a standard segmentation network and then conduct the same recursive learning process to the baseline methods for fairness. We take the state of-the-art DeepLabV3+ [54], FPN [52], and PSPNet [55] with ResNet-18 as the standard models; the detailed results are shown in Table V. It can be seen that our DBFNet outperforms the baseline methods by a large margin. In particular, our F1 score for Car is much higher, which demonstrates the effectiveness of our proposed DBF module for refining the boundary of small objects.

We further conducted experiments to evaluate the performance of NormCut Loss [36] and DenseCRF Loss [37] on point-supervised semantic segmentation for RSIs. For fair comparisons, we incorporated these two loss functions into the baseline method (DeeplabV3+ and RL) to evaluate the results. The details of the results are shown in Table V. It can be seen that these two loss functions can also bring an improvement to the baseline method. The segmentation model achieved 9.57

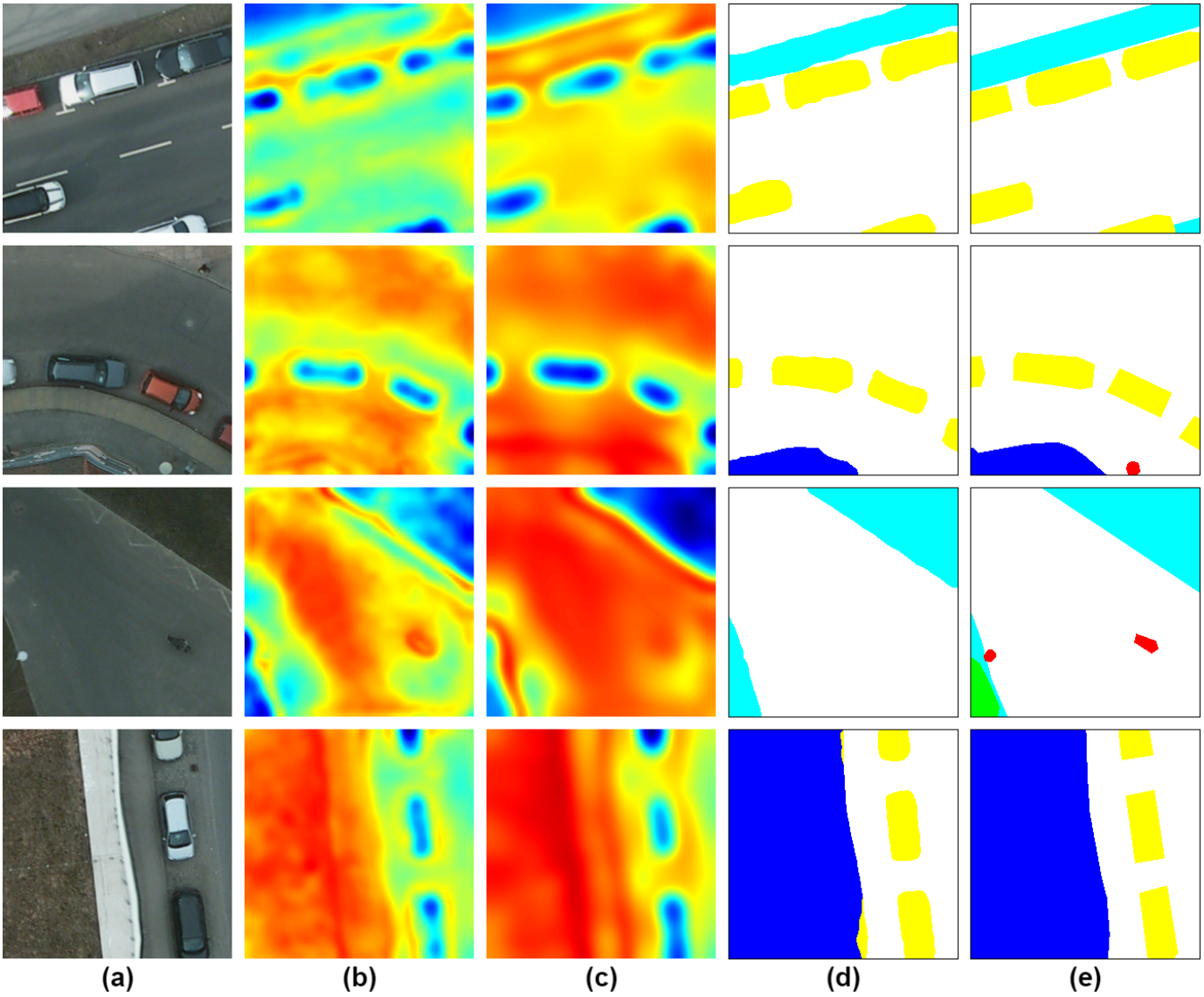



Fig. 8. Visualization of the intermediate outputs before and after the DBF modules on images from the Potsdam dataset: (a) Input images; (b) Feature before the last DBF module (after backbone); (c) Feature after the last DBF module; (d) Segmentation results; (e) Ground truth.

percentage points of m-F1 improvement and 11.53 percentage points of m-F1 improvement with NormCut Loss and DenseCRF Loss, respectively. However, our proposed DBFNet can achieve more competitive results. The reason is that the low-level affinity used in these two loss functions is not obvious in this remote sensing dataset, especially for the categories such as Low vegetable, Tree, and Car. Trees and low vegetables always contain various characteristics and scales, while cars always have a small scale in RSIs. Thus, it is very difficult to utilize the low-level affinity in RSIs.

Instead of designing loss functions for backpropagation, our DBFNet is a forward-propagation filtering method. It does not require any other extra information for supervision, which is simpler and more effective. Our DBFNet can achieve more competitive results compared with other weakly-supervised methods. It is worth noting that, to evaluate the pure effectiveness of our DBFNet, we do not incorporate extra loss functions into our framework.

*4) Comparison with fully-supervised learning methods:* We compare the segmentation performance of our proposed WSL method DBFNet with some state-of-the-art fully-supervised learning (FSL) methods. Note that all the FSL methods use the same backbone, ResNet-101, since the backbone ResNet-101 performs better in the FSL methods, while the WSL methods are built with ResNet-18 as discussed above.

The results are shown in Table VI. It can be seen that in FSL methods, deeper backbones like ResNet-101 can achieve better performance than light-weight backbones like ResNet-18. In FSL methods, HRNet [62] achieves the highest performance with 91.41% m-F1 on the Potsdam dataset. It can be observed that our proposed WSL approach can achieve results that are very close to those obtained from the state-of-the-art FSL method. Moreover, our proposed WSL method can even outperform the classical FSL methods FCN and UNet by 0.69 percentage points and 0.02 percentage points in m-F1 score. Specifically, compared with the FSL methods, the gap

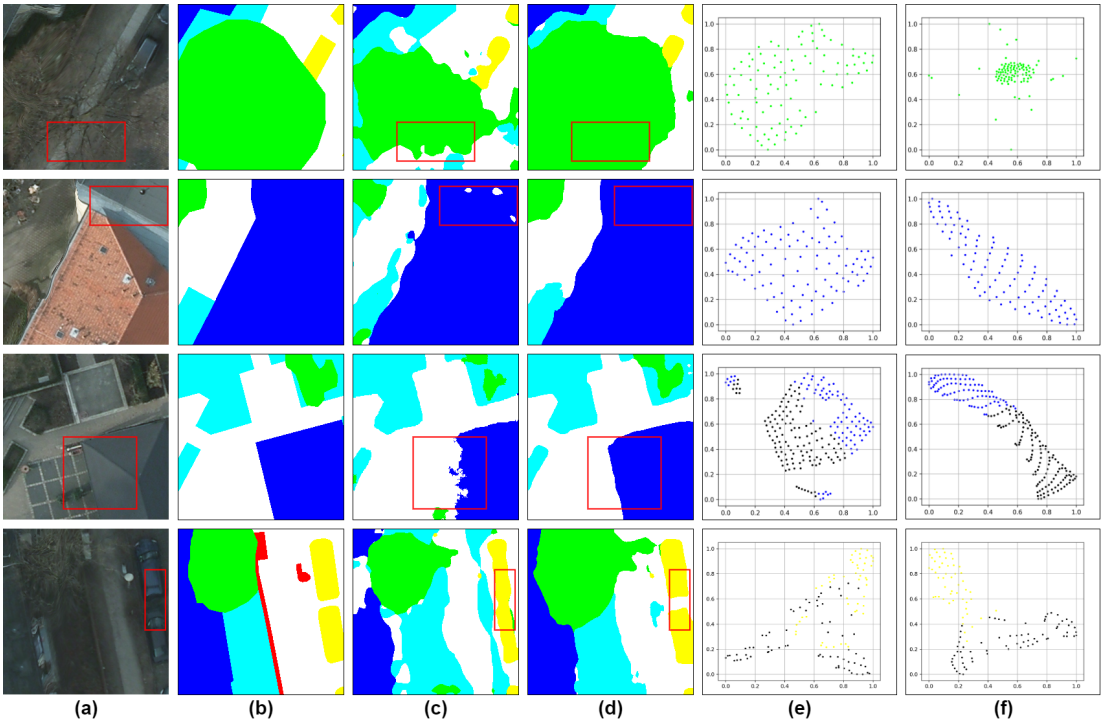



Fig. 9. A contrastive analysis of the proposed DBF and the baseline method for 4 different categories: (a) Input images, in which we focus on the objects in the red boxes; (b) Ground truth; (c) Baseline segmentation results; (d) Our segmentation results. We then map the high-dimension features of (c) and (d) to a 2D space with t-SNE [63], shown in (e) and (f). (For a clear visualization, we changed the impervious surface category's color from white to black.) Here, the images are from the Potsdam dataset.

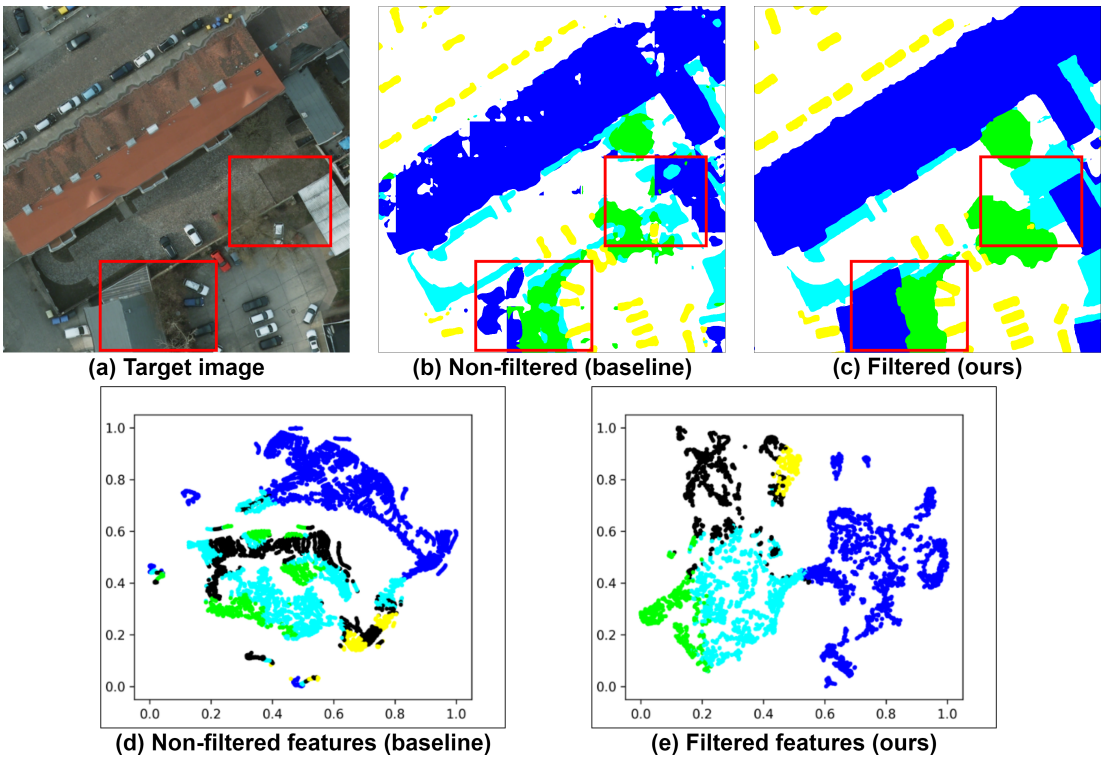



Fig. 10. A contrastive analysis of the proposed DBF and the baseline method: (a) Input images from the Potsdam dataset with a size of 1200 × 1200, in which we also focus on the objects in the red boxes; (b) A non-filtered (baseline) segmentation result; (c) A filtered (ours) segmentation result. We then map the high-dimension features of (b) and (c) to a 2D space with t-SNE [63], shown in (d) and (e). The comparison of feature distributions further proves that our DBF can not only cluster the nearby and similar features but also enhance boundary information.

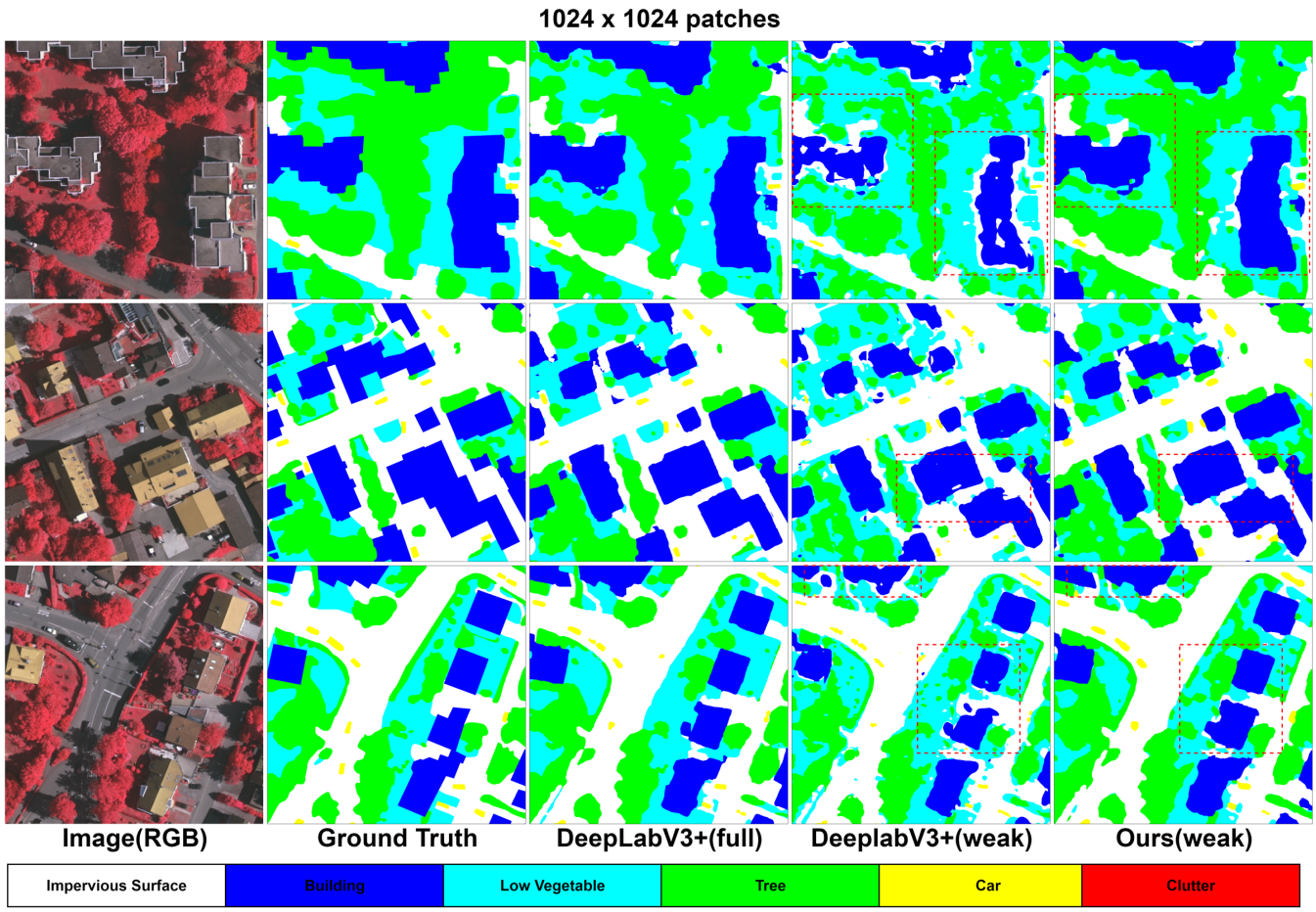



Fig. 11. Qualitative comparisons between the proposed method and other methods on the Vaihingen test dataset.

in accuracy is mainly caused by the Tree and Car categories, while we predict reliable segmentation maps for the other three categories.

*5) Visualization results:* We provide qualitative comparisons between our method and other methods in Fig. 7, including $512 \times 512$ and $1024 \times 1024$ patches. In particular, we leverage the red dashed box to mark those challenging regions that are easy to misclassify. Our method surpasses the baseline WSL method by a large margin. Compared with the baseline WSL method using DeeplabV3+ [54], our method can maintain object coherence and obtain finer boundary information. In particular, for our method, the results of the Building category achieve more complete construction and precise boundaries. For small objects (*e.g.*, cars), our method performs high-confidence predictions of sharp boundaries. In addition, compared with the FSL method DeeplabV3+ [54], our WSL method shows better visualizations of the segmentation map, which demonstrates the effectiveness of our proposed framework.

*6) Visualization results of DBF module and feature distribution:* To get a deeper understanding of our proposed DBF module, we visualize the intermediate outputs of the important stage of the network. We normalize the value in the channel dimension for visualization. In Fig. 8, we visualize the learned feature maps in the DBF module. Note that we take the last DBF module for visualization, and the resolution of the feature map before and after the DBF module is 1/4 of the input image. As shown in Fig. 8(b), due to the weak supervision of our point annotations, the feature map extracted by the baseline backbone (FPN [52] with ResNet [14]) is fragmented and incomplete, and is not clear enough for accurate predictions. Fig. 8(c) shows the output feature map through our DBF module mentioned above. It can be observed that objects of the same category have a similar response and maintain the coherence of the internal features.

Furthermore, we visualize the all-category deep feature distributions in latent space, aiming to verify the DBF module's effect on the deep feature. To this end, we first use the DBF module to filter the feature extracted from the backbone, and then we map its high-dimensional deep feature to a 2D space with t-SNE [63], as shown in Fig. 9. Here, we also take the last DBF module for visualization, and focus on the distributions of the objects in the red box. The comparisons of the feature distributions further prove that our DBF module can encourage nearby and similar deep features to get closer, while preserving the sharpness of the edges. In the first and second rows, we label the t-SNE maps of the Tree category and the Building category. As can be seen, the similar features get closer and achieve coherent segmentation predictions. In the third and the fourth row, we focus on the edges of the objects, *e.g.*,

Impervious Surface and Building, Impervious Surface and Car. From the t-SNE maps, we can observe that the DBF module is applied to distinguish the features near the boundary, which preserves the sharp boundaries of the objects and yields better predictions. These observations are consistent with our t-SNE [63] analysis in Fig. 10, which further verifies the effectiveness of our DBF.

### *E. Experiments on Vaihingen Dataset*

We conducted experiments on the ISPRS Vaihingen benchmark to further evaluate the effectiveness of our DBFNet. We adopted the same training and testing settings that were employed for the Potsdam dataset. Numerical comparisons with the WSL methods and FSL methods are shown in Tables VII and VIII. On the Vaihingen dataset, HRNet [62] also achieves the highest performance with 89.69% m-F1. It can be seen that for the Vaihingen dataset, our DBFNet also outperforms the baseline WSL methods by a large margin. It is worth noting that our DBFNet outperforms the NormCut Loss [36] and DenseCRF Loss [37] with a large margin. This is because, in the Vaihingen dataset, the low-level affinity in the original images is not obvious. Remarkably, our method achieved 86.92% in OA and 86.36% in m-F1. The results show that our proposed WSL method can also work on the Vaihingen dataset, and the m-F1 also surpasses the FSL method FCN and UNet by 3.02 percentage points and 0.92 percentage points, respectively.

Finally, qualitative results are shown in Fig. 10. It can be seen that our method can also produce better segmentation predictions than the baseline. In particular, it can be seen in Fig. 11 that for the Building category, our method predicts more complete construction. We also mark the improved regions with red dashed boxes, which demonstrates the effectiveness of our proposed framework.

## V. Conclusion

In this paper, we presented a point-based weakly supervised learning framework for semantic segmentation in remote sensing images. Our proposed Deep Bilateral Filtering Network (DBFNet) clustered the nearby and similar deep features to make consistent and high-confidence predictions with the smooth region. In addition, the DBFNet employed the DBF to transform the deep features to distinguish the objects' edges, which alleviated the point annotations' lack of boundary information. Extensive experiments on the ISPRS Potsdam benchmark and Vaihingen benchmark demonstrate the effectiveness and efficiency of the proposed DBFNet. The proposed weakly-supervised method can achieve very competitive results compared with state-of-the-art fully-supervised methods.

In the future, we will explore the development of quantitative experiments to analyze the visualization and interpretation of the results performed by our DBFNet and compare them with other methods. Moreover, we will investigate the proficiency of the point-supervised semantic segmentation task in remote sensing images. In addition, we will consider combining prior information from images to design well-defined backpropagation to further improve the segmentation performance

## VI. Acknowledgment

This work was supported in part by the National Natural Science Fund of China under Grant 61922029 and Grant 62101072, in part by the Science and Technology Plan Project Fund of Hunan Province under Grant 2019RS2016, in part by the Key Research and Development Project of Science and Technology Plan of Hunan Province under Grant 2021SK2039, and in Part by the Natural Science Fund of Hunan Province under Grant 2021JJ30003 and Grant 2021JJ40570.

Our deepest gratitude goes to the editors and reviewers for their careful work and thoughtful suggestions that have helped improve this paper substantially.

## References

[1] J. C. Tilton, W. T. Lawrence, and A. J. Plaza, "Utilizing hierarchical segmentation to generate water and snow masks to facilitate monitoring change with remotely sensed image data," *GISci. Remote Sens.*, vol. 43, no. 1, pp. 39–66, 2006.

[2] D. Li, G. Zhang, Z. Wu, and L. Yi, "An edge embedded marker-based watershed algorithm for high spatial resolution remote sensing image segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 10, pp. 2781–2787, 2010.

[3] H. Sheng, X. Chen, J. Su, R. Rajagopal, and A. Ng, "Effective data fusion with generalized vegetation index: Evidence from land cover segmentation in agriculture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2020, pp. 267–276.

[4] L. Yang, R. Huang, J. Huang, T. Lin, L. Wang, R. Mijiti, P. Wei, C. Tang, J. Shao, Q. Li, and X. Du, "Semantic segmentation based on temporal features: Learning of temporal-spatial information from time-series SAR images for paddy rice mapping," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–16, 2021.

[5] Q. Wang, J. Gao, and X. Li, "Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes," *IEEE Trans. on Image Process.*, pp. 1–1, 2019.

[6] Q. Zhang and K. C. Seto, "Mapping urbanization dynamics at regional and global scales using multi-temporal DMSP and OLS nighttime light data," *Remote Sens. Environ.*, vol. 115, no. 9, pp. 2320–2329, 2011.

[7] A. Sarkar, A. Banerjee, N. Banerjee, S. Brahma, B. Kartikeyan, M. Chakraborty, and K. L. Majumder, "Landcover classification in MRF context using Dempster-Shafer fusion for multisensor imagery," *IEEE Trans. Image Process.*, vol. 14, no. 5, pp. 634–645, 2005.

[8] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. SuSstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[9] M. Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011.

[10] A. Radman, N. Zainal, and S. A. Suandi, "Automated segmentation of iris images acquired in an unconstrained environment using HOG-SVM and GrowCut," *Dig. Signal Process.*, vol. 64, pp. 60–70, 2017.

[11] L. Feng, X. Jia, D. Fraser, and A. Lambert, "Super resolution for remote sensing images based on a universal hidden Markov tree model," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1270–1278, 2010.

[12] Tupin, Florence, Koux, and Michel, "Markov random field on region adjacency graph for the fusion of SAR and optical data in radargrammetric applications." *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 8, pp. 1920–1928, 2005.

[13] J. C. Ton, J. Sticklen, and A. K. Jain, "Knowledge-based segmentation of Landsat images," *IEEE Trans. Geosci. Remote Sens.*, vol. 29, no. 2, pp. 222–232, 2002.

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

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2261–2269.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, pp. 1–42, 2014.

[17] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *Proc. Int. Conf. Learn. Represent. (ICLR)*, p. 563–575, 2016.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2015.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015.

[20] Z. Zhou, M. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, 2020.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2017.

[22] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Proc. Deep Learn. Data Labeling Med. Appl. (DLMIA)*, 2016, p. 179–187.

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

[24] K. Nogueira, M. Mura, J. Chanussot, W. R. Schwartz, and J. Santos, "Dynamic multicontext segmentation of remote sensing images based on convolutional networks," *IEEE Trans. Geosci. Remote Sens.*, vol. PP, no. 99, pp. 1–18, 2019.

[25] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "High-resolution aerial image labeling with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. PP, no. 12, pp. 1–12, 2017.

[26] A. Li, L. Jiao, H. Zhu, L. Li, and F. Liu, "Multitask semantic boundary awareness network for remote sensing image segmentation," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–14, 2021.

[27] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *Eur. Conf. Comput. Vis.*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2016, pp. 695–711.

[28] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

[29] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

[30] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Int. Conf. Comput. Vis.*, 2002.

[31] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2921–2929.

[32] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. Huang, "Revisiting dilated convolution: A simple approach for weakly- and semi- supervised semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[33] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[34] A. Bearman, O. Russakovsky, V. Ferrari, and F. F. Li, "What's the point: Semantic segmentation with point supervision," in *Eur. Conf. Comput. Vis.*, 2016.

[35] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 616–625.

[36] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized cut loss for weakly-supervised cnn segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1818–1827.

[37] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov, "On regularized losses for weakly-supervised cnn segmentation," in *Proc. Euro. Conf. Comput. Vis. (ECCV)*, 2018, pp. 507–522.

[38] G. Sun, W. Wang, J. Dai, and L. Van Gool, "Mining cross-image semantics for weakly supervised semantic segmentation," in *Eur. Conf. Comput. Vis.*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., 2020, pp. 347–365.

[39] Y. T. Chang, Q. Wang, W. C. Hung, R. Piramuthu, and M. H. Yang, "Weakly-supervised semantic segmentation via sub-category exploration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.

[40] W. Shimoda and K. Yanai, "Self-supervised difference detection for weakly-supervised semantic segmentation," in *Int. Conf. Comput. Vis.*, 2020.

[41] N. Araslanov and S. Roth, "Single-stage semantic segmentation from image labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 4252–4261.

[42] S. Hong, J. Oh, H. Lee, and B. Han, "Learning transferrable knowledge for semantic segmentation with deep convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 3204–3212.

[43] G. Papandreou, L. C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Int. Conf. Comput. Vis.*, 2016.

[44] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional multi-class multiple instance learning," 2015.

[45] Z. Li, X. Zhang, P. Xiao, and Z. Zheng, "On the effectiveness of weakly supervised semantic segmentation for building extraction from high-resolution remote sensing imagery," *IEEE Jour. Sel. Topics Appl. Earth Obser. Remote Sens.*, vol. 14, pp. 3266–3281, 2021.

[46] R. Lian and L. Huang, "Weakly supervised road segmentation in high-resolution remote sensing images using point annotations," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2021.

[47] Y. Hua, D. Marcos, L. Mou, X. X. Zhu, and D. Tuia, "Semantic segmentation of remote sensing images with sparse annotations," *IEEE Geosci. Remote Sens. Letters*, vol. 19, pp. 1–5, 2021.

[48] 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 Jour. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5398–5413, 2020.

[49] Y. He, J. Wang, C. Liao, B. Shan, and X. Zhou, "Classhyper: Classmix-based hybrid perturbations for deep semi-supervised semantic segmentation of remote sensing imagery," *Remote Sens.*, vol. 14, no. 4, p. 879, 2022.

[50] J. Castillo-Navarro, B. Le Saux, A. Boulch, N. Audebert, and S. Lefèvre, "Semi-supervised semantic segmentation in earth observation: The minifrance suite, dataset analysis and multi-task network study," *Machine Learning*, pp. 1–36, 2021.

[51] B. Zhang, Y. Zhang, Y. Li, Y. Wan, and F. Wen, "Semi-supervised semantic segmentation network via learning consistency for remote sensing land-cover classification," *ISPRS Annal. Photog., Remote Sens. Spatial Inform. Sciences*, vol. 2, pp. 609–615, 2020.

[52] T. Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[53] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. Devito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[54] 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)*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018, pp. 833–851.

[55] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 6230–6239.

[56] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[57] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*.

[58] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., 2020, pp. 173–190.

[59] L. Huang, Y. Yuan, J. Guo, C. Zhang, X. Chen, and J. Wang, "Interlaced sparse self-attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019.

[60] R. Niu, X. Sun, Y. Tian, W. Diao, K. Chen, and K. Fu, "Hybrid multiple attention network for semantic segmentation in aerial images," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–18, 2021.

[61] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit (CVPR)*, 2017, pp. 5168–5177.
[62] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, 2020.
[63] V. D. M. Laurens and G. Hinton, "Visualizing data using t-sne," *Journal of Mach. Learn. Res.*, vol. 9, no. 2605, pp. 2579–2605, 2008.
[64] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Comput. Science*, 2014.
[65] V. Turchenko, E. Chalmers, and A. Luczak, "A deep convolutional auto-encoder with pooling-unpooling layers in caffe," *arXiv preprint arXiv:1701.04949*, 2017.
[66] M. Zhang and W. Shi, "A feature difference convolutional neural network-based change detection method," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 7232–7246, 2020.

**Linshan Wu** received the B.S. degree from the College of Electrical and Information Engineering, Hunan University, Changsha, China, in 2020, where he is currently pursuing the master degree. His research interests focus on computer vision and label-efficient learning.

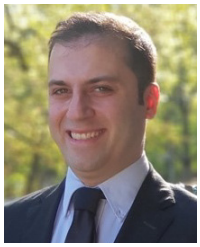
**Leyuan Fang** (S'10-M'14-SM'17) received the Ph.D. degree from the College of Electrical and Information Engineering, Hunan University, Changsha, China, in 2015.

From September 2011 to September 2012, he was a Visiting Ph.D. Student with the Department of Ophthalmology, Duke University, Durham, NC, USA, supported by the China Scholarship Council. From August 2016 to September 2017, he was a Post-Doctoral Researcher with the Department of Biomedical Engineering, Duke University, Durham, NC, USA. He is a Professor with the College of Electrical and Information Engineering, Hunan University, and an Adjunct Researcher with the Peng Cheng Laboratory, Shenzhen, China. His research interests include sparse representation and multiresolution analysis in remote sensing and medical image processing. Dr. Fang was a recipient of one 2nd-Grade National Award at the Nature and Science Progress of China in 2019. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and Neurocomputing.

**Jun Yue** received the B.Eng. degree in geodesy from Wuhan University, Wuhan, China, in 2013 and the Ph.D. degree in GIS from Peking University, Beijing, China, in 2018.

He is currently an Assistant Professor with the Department of Geomatics Engineering, Changsha University of Science and Technology. His research interests include satellite image understanding, pattern recognition, and few-shot learning. Dr. Yue serves as a reviewer for IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Geoscience and Remote Sensing, ISPRS Journal of Photogrammetry and Remote Sensing, IEEE Geoscience and Remote Sensing Letters, IEEE Transactions on Biomedical Engineering, Information Fusion, Information Sciences, etc.

**Bob Zhang** (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2011. He was with Center for Pattern Recognition and Machine Intelligence and later was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently an Associate Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His current research interests include biometrics, pattern recognition, and image processing. In addition, he is also a Technical Committee Member of the IEEE Systems, Man, and Cybernetics Society, and an Associate Editor of IET Computer Vision.



**Pedram Ghamisi** (Senior Member, IEEE) graduated with a Ph.D. in electrical and computer engineering at the University of Iceland in 2015. He works as (1) the head of the machine learning group at Helmholtz-Zentrum Dresden-Rossendorf (HZDR), Germany and (2) visiting professor and group leader of AI4RS at the Institute of Advanced Research in Artificial Intelligence (IARAI), Austria. He is a co-founder of VasoGnosis Inc. with two branches in San Jose and Milwaukee, the USA.

He was the co-chair of IEEE Image Analysis and Data Fusion Committee (IEEE IADF) between 2019 and 2021. Dr. Ghamisi was a recipient of the IEEE Mikio Takagi Prize for winning the Student Paper Competition at IEEE International Geoscience and Remote Sensing Symposium (IGARSS) in 2013, the first prize of the data fusion contest organized by the IEEE IADF in 2017, the Best Reviewer Prize of IEEE Geoscience and Remote Sensing Letters in 2017, and the IEEE Geoscience and Remote Sensing Society 2020 Highest-Impact Paper Award. His research interests include interdisciplinary research on machine (deep) learning, image and signal processing, and multisensor data fusion. He is also a co-founder of VasoGnosis Inc., with two branches in San, USA. For detailed info, please see http://pedram-ghamisi.com/.

**Min He** received the B.Eng. degree from Xiangtan University, Xiangtan, China, in 1999. She received the M.Sc. degree in circuits and systems and the Ph.D. degree in automatic control science and engineering from Hunan University, Changsha, China, in 2003 and 2011, respectively.

Since 2003, she has worked with the College of Electrical and Information Engineering, Hunan University. Her research interests include AI algorithms, especially in image analysis and pattern recognition.