Point-Based Weakly Supervised Deep Learning for Water Extraction from High-Resolution Remote Sensing Imagery

Ming Lu[1], Leyuan Fang[1] and Yi Zhang[2]

[1]the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
[2]the College of Computer Science, Sichuan University, Chengdu 610065, China

Abstract

The use of deep learning for water extraction requires precise pixel-level labels. However, it is very difficult to label high-resolution remote sensing images at the pixel level. Therefore, we study how to utilize point labels to extract water bodies and propose a novel method called the neighbor feature aggregation network (NFANet). Compared with pixel-level labels, point labels are much easier to obtain, but they will lose a lot of information. In this paper, we take advantage of the similarity between the adjacent pixels of a local water body, and propose a neighbor sampler to resample remote sensing images. Then, the sampled images are sent to the network for feature aggregation. Our method uses neighboring features instead of global or local features to learn more representative features. The experimental results show that the proposed NFANet method not only outperforms other weakly supervised approaches, but also obtains similar results as the state-of-the-art ones.

Keywords

Deep learning, weak supervision, semantic segmentation, water extraction

1. Introduction

Water-body extraction from high-resolution remote sensing images is an important research topic in the field of remote sensing. Although the traditional algorithms have made some progress in water-body extraction, there are still problems such as low automation, cumbersome manual feature extraction, and insufficient extraction accuracy. In recent years, deep learning has become an emerging research hot spot in the field of artificial intelligence. The rapid development of deep learning technology and the improvement of computer hardware performance have made deep learning, especially the CNN-based techniques, successful in many important tasks, such as image classification, target detection, and semantic segmentation, and their performance has surpassed many traditional algorithms. The work [1] in proposes a method that combines graph convolutional network (GCN) and CNN to fuse different Hyperspectral features to improve the performance of hyperspectral classification. Work in [2] studys the multi-modal models and proposes a variety of plug-and-play fusion modules to fuse the features of remote sensing images of different modalities. Work in [3] discusses the importance of nonconvex modeling in interpretable AI models from multiple topics.Therefore, it is necessary to apply deep learning to extract water bodies [4, 5, 6].

Unfortunately, the success of deep learning for feature extraction is highly dependent on the availability of sufficient pixel-level labels for training. However, high-resolution remote sensing images are large in scale and data volume, which makes pixel-level labeling extremely laborious. The pixel-level annotation usually requires a lot of time and labor costs, as well as professional knowledge to accurately mark uncertain boundaries between different classes of interest, which hinders the extraction of informative features from high-resolution remote sensing images to a certain extent. Training models using weak labels have received more and more attention in the field of computer vision. Compared with fully-supervised semantic segmentation, weak-supervised learning does not require pixel-level labels, and has the characteristics of fast labeling and low cost. However, the use of weak annotations makes the supervision information seriously insufficient and, thus, key information such as shape, texture, and edges are usually lost, which makes it difficult to extract water from high-resolution remote sensing images with complex scenes.

Some researchers try to use traditional methods combined with deep learning to solve weak supervision problems. The work in [7] combines super-pixels and a local map to obtain rough pseudo-labels to train a water extraction model. Work in [8] combines super-pixel pooling with multi-scale feature fusion to detect buildings. Other researchers attempt to obtain better results by using the extraction capabilities of neural networks. Work in [9] learns from the principle of CAM [10] and extracts feature maps from UNet [11] for hard-threshold processing to obtain segmentation predictions. These methods have achieved promising results in the field of weak-supervised learning but do not consider the characteristics of the image itself.

CDCEO 2021: 1st Workshop on Complex Data Challenges in Earth Observation, November 1, 2021, Virtual Event, QLD, Australia.
1148462196@qq.com (M. Lu); leyuanfang@hnu.edu.cn (L. Fang); yzhang@scu.edu.cn (Y. Zhang)

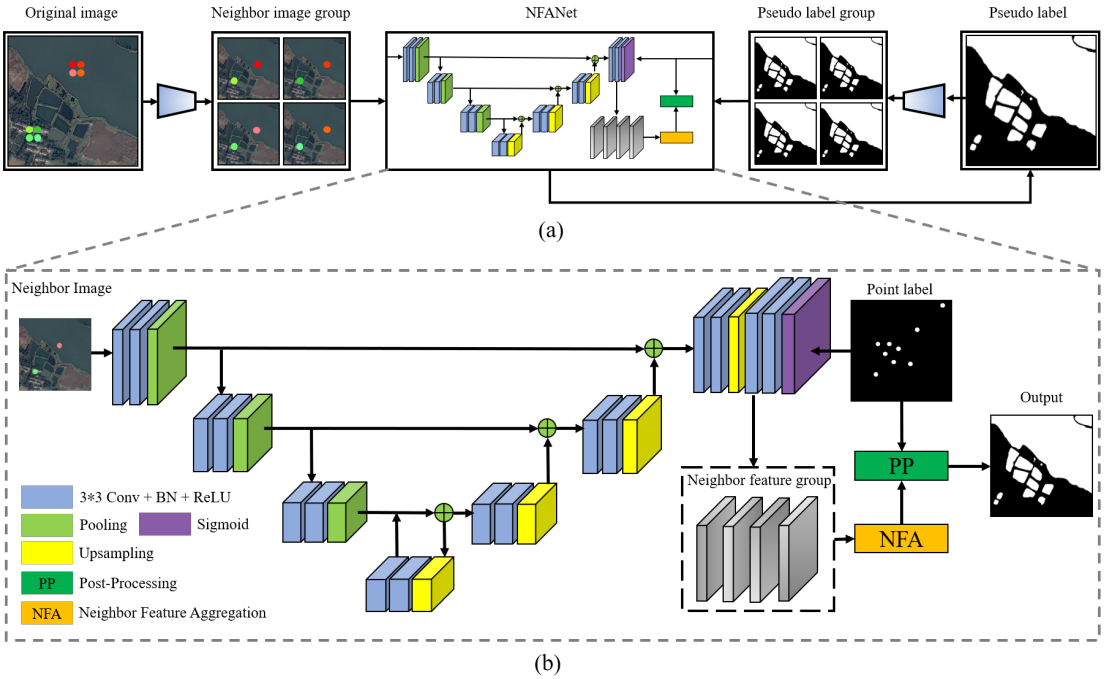

Figure 1: Proposed neighbor sampler. When k is set to 2, a cell contains neighbor pixels of the 2×2 size and is re-allocated to a neighbor image group. Best viewed in color.

Unlike other natural objects, water bodies are usually liquid, the colors and textures of local water bodies are very similar. Therefore, there is a high degree of similarity between neighbor pixels in water bodies, which makes the inherent difference of the features contained between the neighbor pixels of the water-body generally smaller than that of the non-water-body. We hope to map the neighbor pixels of the remote sensing image to the same location in space, and then extract neighbor features from multiple neighbor pixels, and use the neighbor features to jointly decide whether the pixel at this location belongs to the water bodies. Based on the above motivation, we propose the neighbor feature aggregation network (NFANet) to make full use of this property. Specifically, we utilize a sampling method called neighbor sampler to generate a set of neighbor images from high-resolution remote sensing images. The neighbor pixels of the original image are separately allocated to each neighbor image, so that the pixel values of any two neighbor images at the same position are similar but the pixel values are different. On the whole, neighbor image groups have similar but different characteristics. Then, we use an end-to-end model to perform feature extraction on each image of the neighbor image groups, and aggregate the features by using the feature aggregation module. Compared with other methods that only use the local information or global information of an image, the neighbor feature aggregation effectively utilizes the neighbor information and, therefore, more representative features can be learned.

2. Method

Figure 1 illustrates the proposed weakly supervised water extraction framework. Figure 1.a shows the entire recursive training process. We will describe it in the third section. The acquisition of pseudo-labels is shown in Figure 1.b. We input neighbor images into the network and use point labels for supervision to obtain neighbor features. Then the feature aggregation module is used to aggregate the features extracted from the previous step. Finally, post-processing is performed to obtain pseudo-labels. We will describe the details of each of the above steps in the following sections.

2.1. Neighbor Sampler

First, we introduce a neighbor sampler to obtain a neighbor images group $(n_1(x), n_2(x), \ldots, n_L(x))$ from a single optical remote sensing image x. L represents the number of neighbor images. Figure 2 shows the schematic diagram of generating a group of neighbor

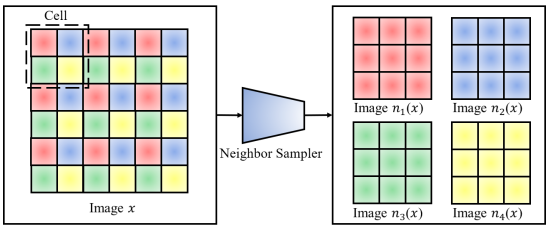

Figure 2: Proposed neighbor sampler. When k is set to 2, a cell contains neighbor pixels of the 2×2 size and is re-allocated to a neighbor image group. Best viewed in color.

images using the neighbor sampler. Let us assume that the width, height, and channel of the input image x are W, H, C, respectively. The implementation of the neighbor sampler $N = (n_1, n_2, \ldots, n_L)$ is described as follows:

1. The image x is divided into $\frac{W}{K} \times \frac{H}{K}$ cells, where the size of each cell is $K \times K \times C$. K is experimentally set to 2 and, therefore, $L = K \times K = 4$.

2. For the $i-th$ row and $j-th$ column of the cell, the pixels in the adjacent positions of each cell are selected in the order from top to bottom and from left to right, which are regarded as the $(i,j)-th$ elements of $N = (n_1(x), n_2(x), \ldots, n_l(x))$. When K is set to 2, the pixels at the upper left, upper right, lower left, and lower right adjacent positions are selected, respectively.

3. For all $\frac{W}{K} \times \frac{H}{K}$ cells being divided in step 1, step 2 will be repeated until all the cells are resampled, and a neighbor sampler $N = (n_1, n_2, \ldots, n_L)$ is generated. Given an optical remote sensing image x, neighbor images group $(n_1(x), n_2(x), \ldots, n_L(x))$ is generated, where the size of each neighbor image is $\frac{W}{K} \times \frac{H}{K} \times C$.

In this way, the neighbor image dataset can be generated from the original dataset. Neighbor images are similar but not identical, because for any two neighbor images, $(i,j)-th$ pixel comes from the neighboring location of the original remote sensing image.

2.2. Neighbor feature aggregation and post-processing

We input the neighbor images group to an end-to-end network to extract features, and obtain the corresponding neighbor feature group $(f_1(X), f_2(X), \ldots, f_L(X)) \in \mathbb{R}^{H \times W \times C \times L}$, where $f_l(X) \in \mathbb{R}^{H \times W \times C}$ represents the feature maps extracted from the $l-th$ image in the neighbor images group. We use the encoder-decoder structure as the feature extraction network. Specifically, the feature maps are extracted from the penultimate convolutional layer. The network structure is shown in Figure 1.b. It is worth noting that the network is replaceable (in the experimental part, a variety of network structures are used for feature extraction). CMax pooling is adopted to reduce the number of channels of each neighbor feature to one. CMax pooling is defined mathematically in detail as follows: Given a three-dimensional feature maps tensor group $F = (f_1(x), f_2(x), \ldots, f_L(x)) \in \mathbb{R}^{H \times W \times C \times L}$, The operation of CMax pooling is as follows:

$$z_{i,j,l}(x) = max(f_{i,j,1,l}(x), f_{i,j,2,l}(x), \ldots, f_{i,j,3,l}(x)), \quad i = 1, 2, \ldots, H, j = 1, 2, \ldots, w, l = 1, 2, \ldots, L. \tag{1}$$

As a result, the feature maps group $Z = (z_1(x), z_2(x), \ldots, z_L(x)) \in \mathbb{R}^{H \times W \times L}$ is obtained. Then, the OTSU algorithm is used to binarize each feature in Z to obtain the result $O = (o_1(x), o_2(x), \ldots, o_L(x)) \in \mathbb{R}^{H \times W \times L}$. The formula is as follows:

$$o_l = Otsu(z_l), l = 1, 2, \ldots, L. \tag{2}$$

Finally, we vote for all binarized neighbor features of the neighbor feature group to obtain the aggregated result V. V is calculated using the following equation:

$$V_{i,j} = \begin{cases} 1, \sum_{l=1}^{L} o_{i,j,l} \geq \frac{2}{L} \\ 0, \sum_{l=1}^{L} o_{i,j,l} < \frac{2}{L} \end{cases} \tag{3}$$

To sum up, the mathematical definition of the feature aggregation module is detailed as follows:

$$V = Vote(Otsu(CMax(F))) \tag{4}$$

where $F \in \mathbb{R}^{H \times W \times C \times L}$ represents the neighbor features group and $V \in \mathbb{R}^{H \times W}$ is the output. Next, the aggregated result V is input to the post-processing module. The specific operations include filling small holes in the closed area by using area filling and removing noise by using morphological operations. Then, we apply a point-label constraint to the processed results. If the area in the result contains point labels, the entire area is retained, otherwise it is not retained. The generated results are used as pseudo-labels and input into the recursive training as supervision information.

2.3. Recursive training

Recursive training is a weakly supervised strategy. When applying the resulting model over the training set, the network outputs capture the shape of objects significantly better than that of just pseudo-labels [12]. We have observed through experiments that when the training set is input to the network again, the obtained network output will become smoother than the coarse-grained pseudo-label, which improves the accuracy of the prediction result to a certain extent.

We embed the neighbor sampler into the recursive training so that the network can learn neighbor features (the flowchart is shown in Figure 1.a). Recursive training

Table 1
Water extraction results (%). The full supervision uses 70% of the training set

Method	Supervision	BgIoU	FgIoU	MIoU	BgDice	FgDice	MDice
FCN[13]	full	89.83 ± 0.26	63.89 ± 0.79	76.86 ± 0.52	94.64 ± 0.14	77.96 ± 0.58	86.30 ± 0.36
Ours(FCN)	weak	89.30 ± 0.31	58.52 ± 0.78	73.91 ± 0.55	94.35 ± 0.17	73.83 ± 0.62	84.09 ± 0.40
UNet	full	90.46 ± 0.13	65.72 ± 0.67	78.09 ± 0.40	94.99 ± 0.07	79.31 ± 0.48	87.15 ± 0.28
Ours(UNet)	weak	89.83 ± 0.40	59.93 ± 1.01	74.88 ± 0.71	94.63 ± 0.21	74.94 ± 0.79	84.79 ± 0.50
ResUNet[14]	full	90.39 ± 0.25	65.72 ± 0.51	78.05 ± 0.37	94.93 ± 0.14	79.30 ± 0.37	87.12 ± 0.25
Ours(ResUNet)	weak	89.68 ± 0.59	59.52 ± 1.41	74.60 ± 1.00	94.55 ± 0.33	74.60 ± 1.11	84.58 ± 0.72
NestedUNet[15]	full	90.43 ± 0.20	65.83 ± 0.51	78.13 ± 0.36	94.97 ± 0.11	79.39 ± 0.37	87.18 ± 0.24
Ours(NestedUNet)	weak	90.03 ± 0.16	60.41 ± 0.46	75.22 ± 0.31	94.75 ± 0.10	75.32 ± 0.35	85.04 ± 0.23
DLinkNet[16]	full	90.23 ± 0.25	65.27 ± 0.67	77.75 ± 0.46	94.86 ± 0.14	78.98 ± 0.49	86.92 ± 0.31
Ours(DLinkNet)	weak	89.86 ± 0.36	59.95 ± 0.91	74.90 ± 0.63	94.66 ± 0.20	74.95 ± 0.71	84.80 ± 0.45
DeepLabV3+[17]	full	90.47 ± 0.17	66.03 ± 0.50	78.25 ± 0.33	95.00 ± 0.09	79.54 ± 0.36	87.27 ± 0.23
Ours(DeepLabV3+)	weak	89.96 ± 0.35	60.23 ± 0.89	5.09 ± 0.62	94.71 ± 0.19	75.17 ± 0.69	84.94 ± 0.44

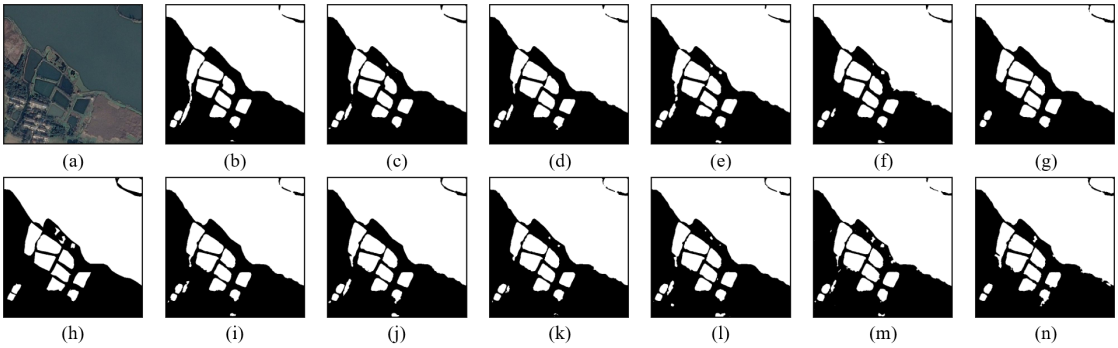

Figure 3: prediction results of the fully supervised methods and our methods. (a) and (h) represent the original image and the ground truth. (b)-(g) represent the prediction results of FCN, UNet, ResUNet, NestedUNet, DLinkNet, and DeepLab V3+, respectively. (i)-(n) represent the prediction results of our method based on FCN, UNet, ResUNet, NestedUNet, DLinkNet and DeepLab V3+, respectively.

consists of three steps. First, the remote sensing image is used to generate neighbor images group. We apply the neighbor images group and point-label to train the network to obtain pseudo-label. Second, the pseudo-label is used to generate pseudo-labels groups. It is worth noting that the $i-th$ image of the neighbor images group and the $i-th$ image of the pseudo-labels group are resampled in the same way. Third, input the $i-th$ image into the network and utilize the $i-th$ label as the supervision information for training. After training the model with all training sets, the neighbor images group are input again to obtain the results group. When $k=2$, the number of results is 4. We perform a weighted average on the results group to obtain a new pseudo-label.

3. Experimental results

3.1. Datasets and evaluation

To prove the effectiveness of the proposed method, we applied the method to high-resolution visible spectrum images for water extraction. This water-body dataset comes from the Gaofen Challenge [18], which contains RGB pan-sharpened images with a resolution of 0.5 m and does not contain infrared bands or digital elevation models. All images are taken from Wuhan and Suzhou, China, mainly in rural areas supplemented by urban areas. The positive labels in the dataset include rivers, reservoirs, rice fields, ditches, ponds, and lakes, while all other non-water pixels are considered negative. The data set is cropped into 1000 images with the size of 492×492 without any overlap. We re-annotated the dataset. The rule is that each independent water body is randomly labeled with a point label of the size 5×5.

In the experiment, the weak supervision models use

Table 2
Water extraction results (%). The full supervision uses 70% of the training set

Method	BgIoU	FgIoU	MIoU	BgDice	FgDice	MDice
Baseline	45.14 ± 5.41	14.26 ± 2.40	29.70 ± 1.69	62.08 ± 5.15	24.90 ± 3.66	43.49 ± 1.17
U-CAM [9]	85.30 ± 0.59	46.79 ± 1.34	66.04 ± 0.93	92.06 ± 0.34	63.74 ± 1.25	77.90 ± 0.78
Local Map [7]	86.07 ± 0.48	48.33 ± 1.56	67.20 ± 1.02	92.51 ± 0.28	65.15 ± 1.43	78.83 ± 0.85
Ours	89.83 ± 0.40	59.93 ± 1.01	74.88 ± 0.71	94.63 ± 0.21	74.94 ± 0.79	84.79 ± 0.50

point labels as the initial supervision information, while the full supervision models use pixel-level labels. Because the remote sensing image segmentation/classification evaluation index of overall accuracy or Kappa coefficient cannot effectively describe the real structure of image segmentation geometry, we choose to use fgIoU (foreground IoU), bgIoU (background IoU), mIoU (mean IoU), fgDice (foreground Dice), bgDice (background Dice) and mDice (mean Dice) to comprehensively evaluate the results. For each model, we performed five independent runs to calculate the aforementioned evaluation indicators and standard deviations.

3.2. Comparison with Fully Supervised Approaches

In Table 1, we report the water extraction performance of our proposed approach and compare it with the fully supervised approaches. Figure 3 also provides the visual performance of all approaches. These approaches randomly use 70% of the samples as the training set, and the remaining data as the test set. Experiments demonstrate that our method achieves the best score using the NestedUNet-based model, and the visual performance shows that the prediction results obtained by our method are very close to the ground truth. The mIoU of our method reached 75.22%, and the mDice reached 85.04%. Compared with the best fully-supervised model DeepLab V3+, the mIoU of our method is only reduced by 3.03%, and mDice is only reduced by 2.23%. But the labeling cost of our method is much less than that of the fully supervised method. Nevertheless, it is difficult to achieve fully supervised performance using only point labels.

3.3. Comparison with Weakly Supervised Approaches

We compare our method with several other weakly supervised remote sensing approaches. The experimental results are shown in Table 2. To be fair, all methods are based on UNet. It can be seen that the mIoU of our method is 8.84%, which is higher than that of the U-CAM-based method with the mDice of 6.89%. In addition, Figure 4 shows the prediction results of other weakly supervised methods and our method. Although other weak supervision methods can predict the local area of the water body, there are errors in the detection of the water body boundary, while our method is relatively more accurate. The studied weak supervised methods cannot detect small objects appropriately while this issue is solved to a great extent by the proposed method.

3.4. Effectiveness of neighbor sampling

In the ablation experiment, the other settings are unchanged, and only the value of K is changed. We set the neighbor sampling parameter k of our proposed network from 1 to 4 and only use cross entropy and dice loss to train the model. For different K values of NFANet, in order to avoid interference from other modules, we only select UNet as the feature extraction network for comparative experiments. In particular, when the value of K is set to 1, the neighbor image group degenerates into the input image. As shown in Figure 5, with the gradual increase of K, mIoU first increases and then decreases. With the increase of neighborhood sampling parameter K, the number of adjacent pixels to be considered increase geometrically, resulting in information redundancy. The size of each reconstructed neighbor image is gradually reduced, and the boundary of the water body will also become unclear. Therefore, we set K equal to 2, because the neighbor features require less computation and achieves better performance.

3.5. Effectiveness of feature aggregation

As shown in Table 3, when K is set to 2 ($L = 4$), assuming that the features of the $i-th$ neighboring image is f_i, $\sum_{l=1}^{L} f_l$ means feature aggregation is used. Compared with the best method that does not use feature aggregation, the mIoU and mDice of our method are improved by 4.8% and 3.7% respectively. To a certain extent, the greater the number of neighboring images, the more features are available, and these features can complement each other. Therefore, after the feature aggregation, the performance of the prediction results can be improved.

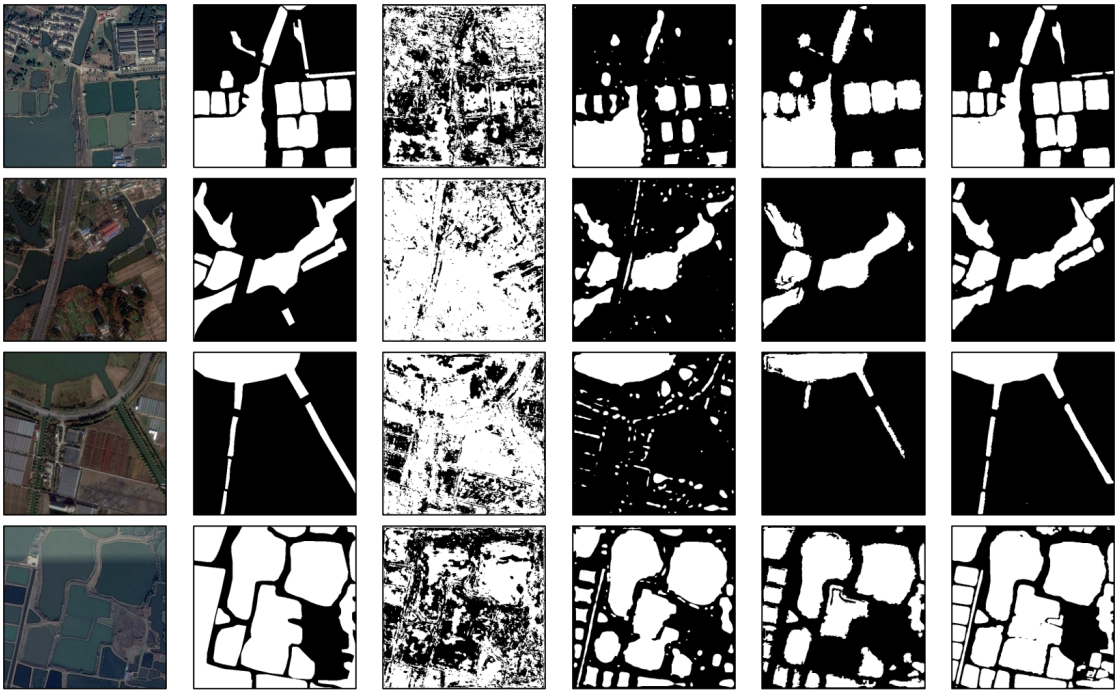

Figure 4: Prediction results of the investigated weakly supervised methods. (a) and (b) represent the original image and the ground truth. (c) represents the baseline. (d) represents U-CAM. (e) represents the Local Maps method. (f) represents our method.

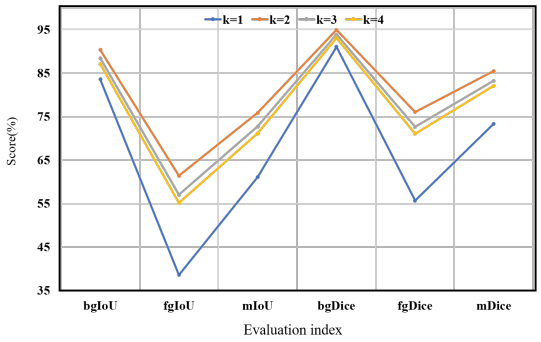

Figure 5: Effectiveness of neighbor sampling.

Table 3
Effectiveness of feature aggregation

NF	BgIoU	FgIoU	MIoU	BgDice	FgDice	MDice
f_1	86.6	38.3	62.5	92.8	55.4	74.1
f_2	87.2	51.6	69.4	93.2	68.1	80.6
f_3	87.7	52.4	70.1	93.5	68.8	81.1
f_4	86.2	48.5	67.3	92.6	65.3	79.0
$\sum_{l=1}^{L} f_l$	89.8	59.9	74.9	94.6	74.9	84.8

3.6. Time consumption

The hardware configurations for the experiments in this paper consisted of Intel Core i7-9700k 3.60 GHz CPU, GeForce RTX 2080Ti GPU, and 16GB RAM. The results of the GPU inference time are shown in Table 4. The results in the table are the average GPU inference time of the data set. After using recursive training to improve the quality of pseudo-labels, we input the pseudo-labels and original images into the models consistent with the fully-supervised methods for training and inference. Therefore, the GPU inference time of the proposed method is the same as that of the fully-supervised methods. It can be observed from the table that the inference time of DLinkNet is the shortest. because DLinkNet compresses the feature channel in the decoder to reduce the computational cost. NestedUNet embeds U-Nets of different depths in its architecture, which requires more convolution calculations, thus increasing the consumption of inference time.

Table 4
GPU inference time of different mothods

Method	FCN	UNet	ResUNet	NestedUNet	DLinkNet	DeepLabV3+
Time (ms)	17.32	15.14	17.90	31.71	11.84	23.97

4. Conclusion

In this paper, we proposed a network entitled NFANet. Unlike traditional convolutional neural networks that only use global or local features for discrimination, NFANet uses neighbor features, which allows more representative features to be learned. We fuse these neighbor features to obtain pseudo-labels, and improve the label quality by recursive training. We tested it on water data sets and compared it with advanced fully supervised and weakly supervised methods. By using only point labels, the proposed method obtains comparable results with that of full supervision. As a possible future work, we will conduct research on weakly supervised or semi-supervised methods of self-correction. Remote sensing images collected from satellites are usually affected by spectral variability.

Work in [19] uses endmember dictionary and spectral variability dictionary to model different spectral variability respectively. In addition, this method provides reasonable prior knowledge for the spectral variability dictionary. Our proposed method considers local neighbor pixels. Therefore, when encountering various degeneration, noise influences and other variability factors, it is necessary to analyze whether these variability factors cause greater interference between neighbor pixels. If the variability factors bring different effects to different local areas, it is very likely that the prediction results will lose the water bodies. In future work, we will consider introducing cross-local features to improve the network's ability to learn features between different local water bodies.

References

[1] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, J. Chanussot, Graph convolutional networks for hyperspectral image classification, IEEE Transactions on Geoscience and Remote Sensing 59 (2021) 5966–5978. doi:10.1109/TGRS.2020.3015157.

[2] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, B. Zhang, More diverse means better: Multimodal deep learning meets remote-sensing imagery classification, IEEE Transactions on Geoscience and Remote Sensing 59 (2021) 4340–4354. doi:10.1109/TGRS.2020.3016820.

[3] D. Hong, W. He, N. Yokoya, J. Yao, L. Gao, L. Zhang, J. Chanussot, X. Zhu, Interpretable hyperspectral artificial intelligence: When nonconvex modeling meets hyperspectral remote sensing, IEEE Geoscience and Remote Sensing Magazine 9 (2021) 52–87. doi:10.1109/MGRS.2021.3064051.

[4] Y. Ren, H. Xu, B. Liu, X. Li, Sea ice and open water classification of sar images using a deep learning model, in: IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 3051–3054. doi:10.1109/IGARSS39084.2020.9323990.

[5] V. Poliyapram, N. Imamoglu, R. Nakamura, Deep learning model for water/ice/land classification using large-scale medium resolution satellite images, in: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, 2019, pp. 3884–3887. doi:10.1109/IGARSS.2019.8900323.

[6] R. Yan, S. Dong, Optical remote sensing image waters extraction technology based on deep learning context-unet, in: 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019, pp. 1–4. doi:10.1109/ICSIDP47821.2019.9173433.

[7] K. Fu, W. Lu, W. Diao, M. Yan, H. Sun, Y. Zhang, X. Sun, Wsf-net: Weakly supervised feature-fusion network for binary segmentation in remote sensing image, Remote Sensing 10 (2018).

[8] J. Chen, F. He, Y. Zhang, G. Sun, M. Deng, Spmf-net: Weakly supervised building segmentation by combining superpixel pooling and multi-scale feature fusion, Remote Sensing 12 (2020) 1049.

[9] S. Wang, W. Chen, S. M. Xie, G. Azzari, D. B. Lobell, Weakly supervised deep learning for segmentation of remote sensing imagery, Remote Sensing 12 (2020) 207.

[10] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, CVPR (2016).

[11] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention (2015).

[12] A. Khoreva, R. Benenson, J. Hosang, M. Hein, B. Schiele, Simple does it: Weakly supervised instance and semantic segmentation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[13] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[14] Z. Chu, T. Tian, R. Feng, L. Wang, Sea-land segmentation with res-unet and fully connected crf, in: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, 2019, pp. 3840–3843. doi:10.1109/IGARSS.2019.8900625.

[15] Z. Zhou, M. Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested u-net architecture for medical image segmentation, 4th Deep Learning in Medical Image Analysis (DLMIA) Workshop (2018).

[16] L. Zhou, C. Zhang, M. Wu, D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, pp. 192–1924. doi:10.1109/CVPRW.2018.00034.

[17] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, Springer, Cham (2018).

[18] 2020 gaofen challenge on automated high-resolution earth observation image interpretation, http://en.sw.chreos.org/, 2020.

[19] D. Hong, N. Yokoya, J. Chanussot, X. X. Zhu, An augmented linear mixing model to address spectral variability for hyperspectral unmixing, IEEE Transactions on Image Processing 28 (2019) 1923–1938. doi:10.1109/TIP.2018.2878958.