NPALOSS: NEIGHBORING PIXEL AFFINITY LOSS FOR SEMANTIC SEGMENTATION IN HIGH-RESOLUTION AERIAL IMAGERY

Yingchao Feng[1,2,3], Wenhui Diao[1,2,*], Xian Sun[1,2,3], Jihao Li[1,2,3], Kaiqiang Chen[1,2], Kun Fu[1,2,3], Xin Gao[1,2]

[1]Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China
[2]Key Laboratory of Network Information System Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China - (whdiao, sunxian, fukun, gaxi)@mail.ie.ac.cn, chenkaiqiang14@mails.ucas.ac.cn
[3]University of Chinese Academy of Sciences, Beijing, China - (fengyingchao17, lijihao17)@mails.ucas.edu.cn

KEY WORDS: Deep Learning, Semantic Segmentation, Pixel Weighting Loss, Small-sized Objects, Boundaries, Aerial Imagery

ABSTRACT:

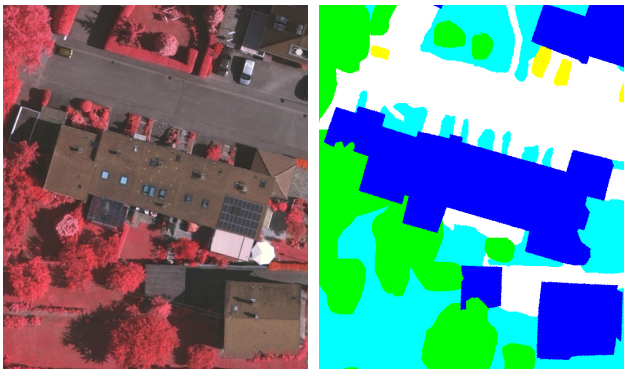
The performance of semantic segmentation in high-resolution aerial imagery has been improved rapidly through the introduction of deep fully convolutional neural network (FCN). However, due to the complexity of object shapes and sizes, the labeling accuracy of small-sized objects and object boundaries still need to be improved. In this paper, we propose a neighboring pixel affinity loss (NPALoss) to improve the segmentation performance of these hard pixels. Specifically, we address the issues of how to determine the classifying difficulty of one pixel and how to get the suitable weight margin between well-classified pixels and hard pixels. Firstly, we convert the first problem into a problem that the pixel categories in the neighborhood are the same or different. Based on this idea, we build a neighboring pixel affinity map by counting the pixel-pair relationships for each pixel in the search region. Secondly, we investigate different weight transformation strategies for the affinity map to explore the suitable weight margin and avoid gradient overflow. The logarithm compression strategy is better than the normalization strategy, especially the common logarithm. Finally, combining the affinity map and logarithm compression strategy, we build NPALoss to adaptively assign different weights for each pixel. Comparative experiments are conducted on the ISPRS Vaihingen dataset and several commonly-used state-of-the-art networks. We demonstrate that our proposed approach can achieve promising results.

1. INTRODUCTION

Semantic segmentation aims to assign each pixel to a semantic class label, which plays an essential role in many applications in the field of remote sensing, e.g., environmental modeling, land planning, and disaster assessment. Recently, with the introduction of deep fully convolution neurons network (FCN) (Shelhamer et al., 2017), the performance of semantic segmentation has been improved rapidly, and many state-of-the-art FCN based methods (Zhao et al., 2017; Chen et al., 2017b; Fu et al., 2019; Huang et al., 2019) have achieved significant segmentation quality on several benchmark datasets. However, there is still a challenge on how to improve the prediction quality of small-sized objects and object boundaries in high-resolution aerial imagery.

As shown in Figure 1, an example is taken from the ISPRS Vaihingen dataset (Cramer, 2010; Rottensteiner et al., 2012). There is a vast difference in size between objects, not only between different categories, such as the Car class and Tree class, but also between the same category, such as the Building class. Besides, the shapes of objects are extremely irregular, especially the Impervious Surfaces and Low vegetation class, which lead to complex boundaries. However, since the standard cross-entropy loss function fairly calculates the cost value of each pixel, the gradients of these small-sized objects and object boundaries are overwhelmed by a large number of well-classified pixels, which make the networks tend to be biased towards the well-classified pixels and produce poor results for such hard pixels during inference.

One way to mitigate this issue is under-sampling well-classified pixels or over-sampling small-sized objects and object boundaries. Such methods (Jeatrakul et al., 2010) may lead to sub-optimal exploitation of available data and increase the risk of over-fitting. Another widely used approach is introducing specific weights. For example, the class weighted methods, which assign different weights for each class via an inverse frequency re-weighting (Audebert et al., 2018; Mostajabi et al., 2015) or median frequency balancing (Eigen, Fergus; Badrinarayanan et al., 2017). However, these methods mainly improve the performance of the minority class. Recently, some researches utilize the focal loss (Lin et al., 2017) to reduce the weight of well-classified pixels to improve the performance of the above-



(a) True orthophoto (b) Ground truth

Figure 1. An example patch is taken from the ISPRS Vaihingen dataset. There are great differences in the size and shape of objects. The classes of interest are given by Impervious Surfaces (white), Building (blue), Low Vegetation (cyan), Tree (green) and Car (yellow).

* Corresponding author

mentioned hard pixels, but the effect is unstable and easily leads to over-fitting when applied to the multi-class semantic segmentation tasks. In addition, dice loss (Milletari et al., 2016) proposes a solution to the imbalances based on the dice coefficient. However, this loss function is generally used in binary classification tasks. We find that there is no suitable method to improve the performance of small objects and object boundaries in high-resolution aerial imagery.

In this paper, we propose a neighboring pixel affinity loss (NPALoss) to improve the performance of the small-sized objects and object boundaries. To achieve this purpose, we need a way to measure the classifying difficulty of each pixel. The values of well-classified pixels should be small and the values of small-sized objects and object boundaries should be large. Therefore, we convert the problem of how to determine the classifying difficulty of one pixel into a problem that the pixel categories in the neighborhood are the same or different. Based on this idea, we build a neighboring pixel affinity map by counting the pixel-pair relationships for each pixel in the search region. Then, we investigate different weight transformation strategies for the affinity map and find that too large or too small weight margin will affect the performance of our proposed NPALoss. Combining the affinity map and weight transformation strategy, our NPALoss helps the network pay more attention to the pixels corresponding to small-sized objects and object boundaries. Note that our NPALoss is only calculated based on the ground truth map, which can be combined with any existing architectures without adding any computational complexity. Results on the ISPRS Vaihingen dataset (Cramer, 2010; Rottensteiner et al., 2012) and various popular segmentation networks prove the effectiveness and robustness of our method.

After briefly summarizing related work in Section 2, we explain the proposed methodology in Section 3. Subsequently, we demonstrate the performance of our methodology by presenting and discussing results achieved for a standard benchmark dataset in Sections 4 and 5. Finally, we provide concluding remarks and suggestions for future work in Section 6.

2. RELATED WORK

Since the great success of the deep fully convolutional neural network (FCN) (Shelhamer et al., 2017) in semantic segmentation, various methods have been proposed to improve the segmentation performance. Some researches introduce the wider and deeper backbone networks, e.g., ResNet (He et al., 2016) and DenseNet (Huang et al., 2017), to extract richer feature map. SegNet (Badrinarayanan et al., 2017) utilizes a well-designed encode-decoder network to reduce errors caused by upsampling. Besides, dilated convolution (Yu, Koltun; Chen et al., 2017a) has been introduced to segmentation networks to reduce the output stride while preserving the resolution of the feature map. PSPNet (Zhao et al., 2017) proposes a pyramid pooling module (PPM) and DeepLab (Chen et al., 2017a,b) proposes atrous spatial pyramid pooling (ASPP) layer to exploit multi-scale context information. Recently, DANet (Fu et al., 2019) and CCNet (Huang et al., 2019) introduce the non-local (Wang et al., 2018) operation to learn long-range dependencies, which achieve state-of-the-art segmentation performance on several benchmark datasets.

However, these works cannot handle the problem of imbalanced distributions, especially for small-sized objects and object boundaries in the high-resolution aerial imagery. A relatively simple solution for many semantic segmentation networks is to introduce weighted loss functions. The purpose is to assign weak classes a higher cost value than the strong classes (usually, the criteria of weak and strong classes are determined based on the number of pixels in each category). For example, based on the original data statistics, the weights can be obtained by an inverse frequency re-weighting scheme (Audebert et al., 2018; Mostajabi et al., 2015) or median frequency balancing (Eigen, Fergus; Badrinarayanan et al., 2017). Although these methods improve the performance of the minority class, the prediction results of majority classes may be degraded. Besides, some works introduce the hard negative mining strategy (Shrivastava et al., 2016) to sample hard examples during the training process. In (Wu et al., 2016), the number of pixels to be updated during backpropagation is limited. The pixel losses are sorted and only the k highest loss positions are updated. However, this method increases the training epochs and it is difficult to find a suitable hyper-parameter k. Another recent work proposes a focal loss (Lin et al., 2017) to assign the weight for each pixel based on the prediction probability. This loss function helps detection networks efficiently train on all examples without sampling. There are also some methods (Feng et al., 2019; Wang et al., 2019) to solve the imbalance distributions in object detection in remote sensing. However, for semantic segmentation in the field of remote sensing, we find that there is no suitable method to solve the problem of imbalanced distribution, especially how to improve the performance of small-sized objects and object boundaries.

3. METHODOLOGY

In this section, we describe our proposed NPALoss for semantic segmentation in high-resolution aerial imagery. Thereby, we first describe an overview of the segmentation network and how to combine our NPALoss with the networks. (Section 3.1). Subsequently, we provide a detailed explanation of our proposed neighboring pixel affinity map, which is the basis of our NPALoss (Section 3.2). Finally, Section 3.3 introduces the weight transformation strategies in detail.

3.1 Overview

The overall architecture of our proposed method is illustrated in Figure 2. We adopt the fully convolution neurons network, which composed of the backbone network and the head network. The backbone network takes the images as the input to extract its feature maps. Then the head network aggregates higher-level information based on the output of the backbone network to produce the pixel-wise prediction. The only difference is that we replace the standard cross-entropy loss function to our proposed neighboring pixel affinity loss function. Our NPALoss introduces the neighboring pixel affinity map to adaptively assign a different weight for each pixel.

3.2 Neighboring Pixel Affinity Map

In general, the prediction results for small-sized objects and object boundaries are worse than results for the easily classified regions, because the gradients of well-classified pixels dominate the backpropagation. However, we want the loss function to shift its attention from well-classified pixels to the hard pixels. In other words, the hard pixels in the backpropagation should assign large weights while well-classified pixels should assign small weights.

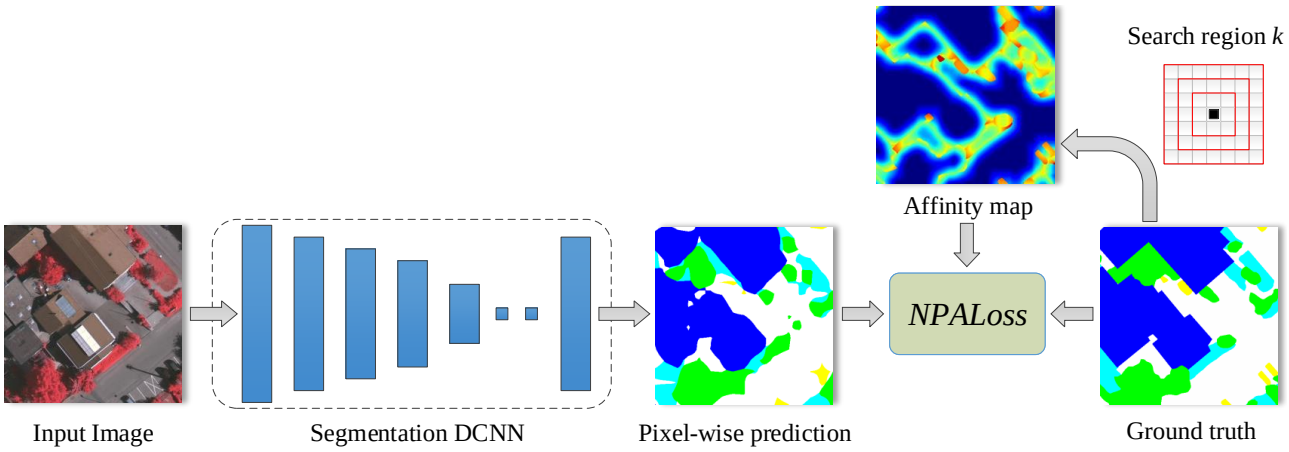


Figure 2. Overview of our framework. We replace the standard cross-entropy loss to our proposed NPALoss, which is only calculated by ground truth map and can be combined with any existing semantic segmentation networks.

To achieve this purpose, we build a neighboring pixel affinity map based on the ground truth. Specifically, we count the pixel-pair relationships in the neighborhood to measure the classifying difficulty of each pixel. As shown in Figure 3, for the top-left pixel in the ground truth map, when the search region is the neighboring eight pixels, all adjacent pixels have the same category with the top-left pixel, so the affinity value is zero, which means the top-left pixel is easy to classify. On the contrary, the neighboring pixels of small-sized objects and object boundaries always have different categories, resulting in large affinity values. When enlarging the search region, the affinity map can further judge the classifying difficulty of each pixel based on richer local context information.

We now explain the neighboring pixel affinity map more formally. Give the search region k, For each pixel $p_{x,y}$ in the ground truth, the pixel in its search region can be denoted as $q_{u,v}$, where $0 \leqslant |x-u|, |y-v| \leqslant k$. Then, the affinity map can be obtained as follow:

$$A_{x,y} = \sum_{u} \sum_{v} C(p_{x,y}) \oplus C(q_{u,v}), \tag{1}$$

where $C(\cdot)$ denotes the ground truth class of the pixel and $\oplus$ means the XOR operation, the result is 1 only when the categories of two pixels are different. As shown in Figure 4, we visualize the affinity maps with different search region k. It can be seen that the closer to the center of the hard regions (small-sized objects and object boundaries), the greater the value of the affinity maps.

3.3 Weight Transformation

According to Eq. 1, the range of value of the neighboring pixel affinity map is $[0, (2k+1)^2 - 1]$. For example, when $k = 32$, the range is $[0, 4224]$. If the affinity map is used directly as the weights of each pixel, the weight margin may be unsuitable and cause gradient overflow. Therefore, it is necessary to introduce a suitable weight transformation strategy.

In this work, we consider two transformation strategies, one is normalization operation, the other is logarithm operation. More specifically, we define the normalization operation as follow:

$$AT_{x,y} = \frac{A_{x,y} - min(A_{x,y})}{max(A_{x,y}) - min(A_{x,y})} + L, \tag{2}$$

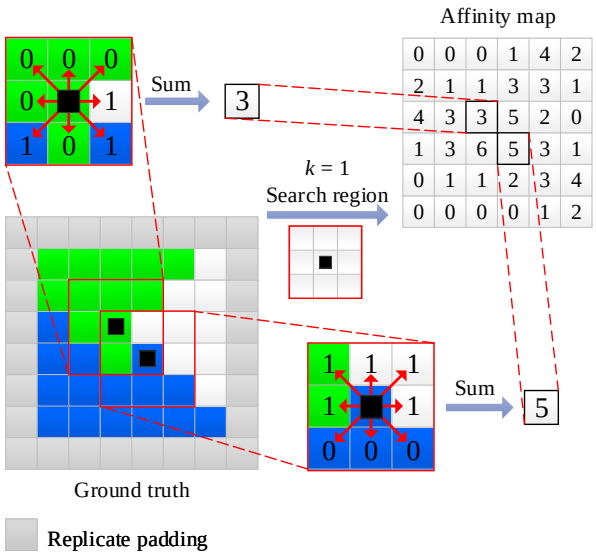


Figure 3. Illustration of how neighboring pixel affinity map calculated when search region $k = 1$. The ground truth map is padded with the values from edge pixels.

where $L(L \geqslant 0)$ is a constant, which is used to guarantee the minimum value of the affinity map. The normalization operation may result in low weight margin between the well-classified and hard pixels. Therefore, we also consider the logarithm compression operation to achieve weight transformation:

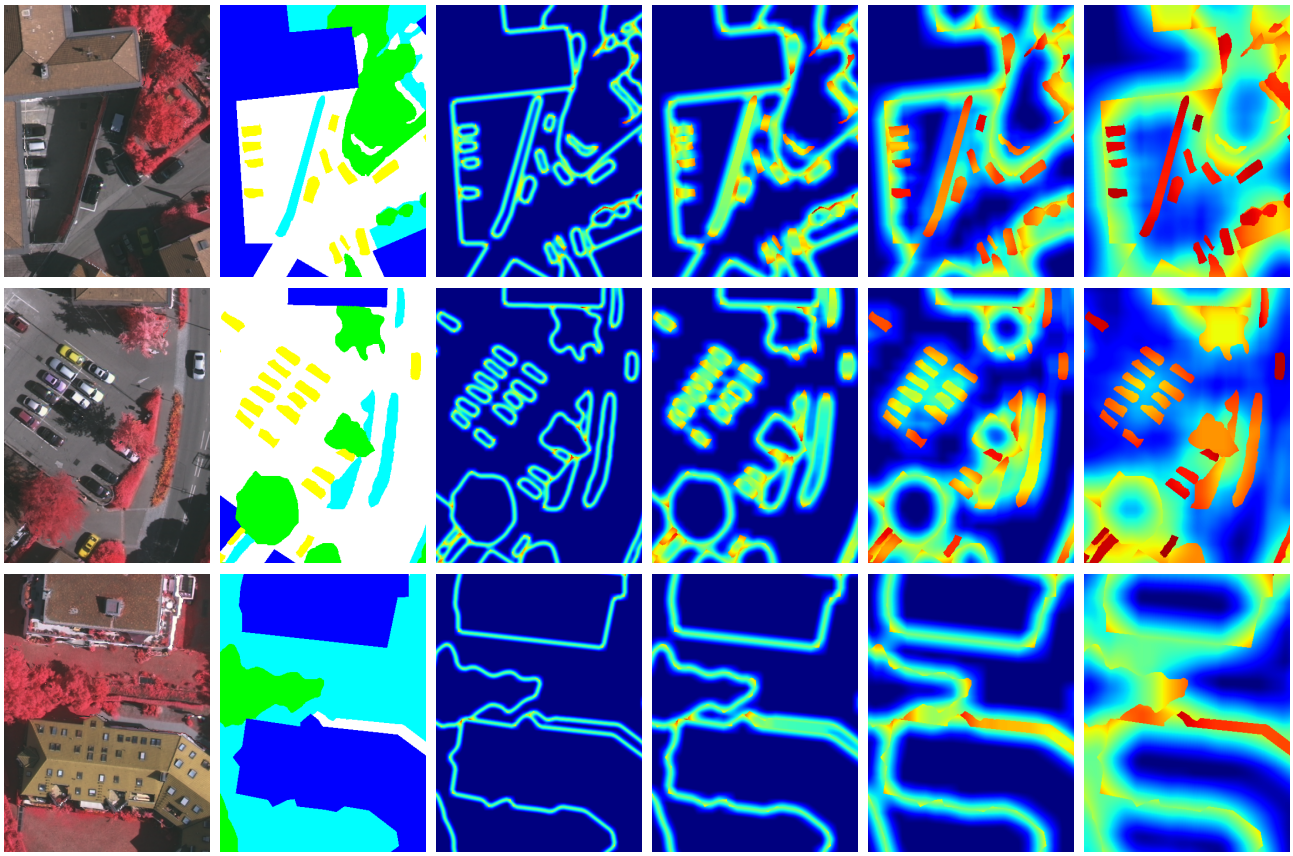
$$AT_{x,y} = log_a(A_{x,y} + a^L), \tag{3}$$

where a means the base of the logarithm. The larger the value of a, the smaller the range of the affinity map. In this work, we consider common logarithm ($a = 10$) and natural logarithm ($a = e$), respectively.

To this end, combining the neighboring pixel affinity map and the weight transformation strategy, the NPALoss can be defined as follow:

$$NPALoss(y^p, y^g) = -\frac{1}{N} \sum_{i}^{H} \sum_{j}^{W} AT_{i,j} y^g_{i,j} log(y^p_{i,j}), \tag{4}$$

where N, H, W are the mini-batch size and spatial dimension,



(a) True orthophoto (b) Ground truth (c) $k = 8$ (d) $k = 16$ (e) $k = 32$ (f) $k = 64$

Figure 4. The effect of neighboring pixel affinity maps with the search region $k = 8$, 16, 32, and 64. The higher the value of the pixel closer to the small-sized objects and object boundaries. The affinity map is only calculated based on ground truth.

respectively. y^p and y^g denote the softmax probability and the corresponding ground truth label. Note that our method not only supports cross-entropy loss function, other loss functions are also possible.

4. EXPERIMENTS

In the following, we first briefly describe the dataset used in our experiments (Section 4.1). Subsequently, we explain implementation details and experimental configurations before introducing the derived results (Section 4.2).

4.1 Dataset

We validate our proposed method on the ISPRS Vaihingen dataset (Cramer, 2010; Rottensteiner et al., 2012). The dataset contains 33 tiles, each of which is a true orthophoto (with three channels corresponding to the near-infrared, red and green domains) and corresponding Digital Surface Models (DSM) generated via dense image matching. The dataset has a resolution of 9 cm/pixel with tiles of approximately 2400 $\times$ 2000 pixels. In addition, the dataset contains 16 available tiles with ground truth labels and contains five foreground classes (Impervious Surfaces, Building, Low Vegetation, Tree, Car) and one background class (Clutter, includes water bodies and other objects such as containers, tennis courts or swimming pools). Follow the prior works (Paisitkriangkrai et al., 2015; Volpi, Tuia), we divide the labeled images into 11 training images (1, 3, 5, 7, 13, 17, 21, 23, 26, 32, 37) and 5 validation images (11, 15, 28, 30 and 34). Note that we do not use DSM or normalized DSM (nDSM) data (Gerke, 2014; Audebert et al., 2016; Chen et al., 2018a), only true orthophoto images are used as input of all networks.

4.2 Experimental Setup

Our implementations are based on the publicly available Pytorch (Paszke et al., 2017) deep learning framework and tested on a workstation with 32 GB RAM, an Intel Core i7-6700k processor, and one NVIDIA GeForce GTX 1080Ti GPU card. In this paper, we choose four popular networks, FCN (Shelhamer et al., 2017), FCN-dilated (Yu, Koltun), PSPNet (Zhao et al., 2017) and DeepLabV3 (Chen et al., 2017b), to evaluate our proposed method. The backbone network is set as ResNet101 (He et al., 2016) and pre-trained on ImageNet (Russakovsky et al., 2015). During the training process, we use mini-batch stochastic gradient descent (SGD) with batch size 6, momentum 0.9, and weight decay 0.0001. We train all models for 5000 iterations and crop the input size to 513×513. Each patch fed into the network is cropped randomly and then normalized by the subtraction of the mean value and a subsequent division by the standard deviation. We set the base learning rate to 0.01 and use the poly learning rate strategy in which the current learning rate is multiplied by $(1 - \frac{iter}{max\_iter})^{power}$ each iteration with power 0.9. Besides, the random seed is fixed for a fair comparison. For quantitative evaluation, we report the Overall Accuracy (OA) and the mean Intersection-over-Union (mIoU) using the median of 3 runs. To evaluate the performance for each

Method	Imp. Surf.	Building	Low Veg.	Tree	Car	mF_1	mIoU	OA
FCN	89.89	95.35	78.57	86.83	64.88	83.10	72.44	87.80
FCN + class weighted	89.91	95.44	78.38	86.85	68.03	83.72	73.14	87.72
FCN + focal loss	89.09	94.13	76.41	85.82	63.40	81.77	70.53	86.60
FCN + dice loss	82.47	89.04	68.22	80.48	42.72	72.58	59.33	79.58
FCN + NPALoss (ours)	**90.59**	**95.49**	**79.18**	**87.27**	**73.32**	**85.17**	**75.01**	**88.33**

Table 1. The performance of FCN with different loss functions.

Strategy	L	Imp. Surf.	Building	Low Veg.	Tree	Car	mF_1	mIoU	OA
Baseline	—	89.89	95.35	78.57	86.83	64.88	83.10	72.44	87.80
Norm	0	89.18	94.28	76.43	86.27	70.93	83.42	72.47	86.77
Norm	0.5	90.37	95.46	78.46	86.93	71.50	84.54	74.16	87.99
Ln	0.5	90.41	95.32	**79.54**	87.24	72.84	85.07	74.86	88.28
Log10	0.5	**90.59**	**95.49**	79.18	**87.27**	**73.32**	**85.17**	**75.01**	**88.33**

Table 2. The influence of different weight transformation strategies based on the FCN.

class, we additionally consider F_1 score which is defined as the harmonic mean of precision and recall.

Deep Supervision: Except for the FCN (Shelhamer et al., 2017), we utilize deep supervision (Zhao et al., 2017) on the other three networks. An auxiliary loss branch is applied apart from the main loss and we set the weight to 0.4 in all experiments.

Data Augmentation: There are only 54 million pixels in 11 training images. Therefore, it is necessary to adopt some appropriate data augmentation strategies to avoid the problem of overfitting. In this work, we adopt random mirror, random Gaussian blur and random resize between 0.5 and 2.0, and additionally add random rotation between -10 and 10 degrees on the input images to enhance the dataset in the training process.

Inference Strategy: Due to the large difference between the size of the tiles and the training size of the network, using the entire image directly as an input to the network may reduce prediction accuracy. Therefore, we choose the sliding window strategy (Chen et al., 2018b; Fu et al., 2020) when making inference. Specifically, we crop the patches from the tiles in an overlapping manner and set the overlapping stride to 1/3. The final results of the pixels in the overlapping areas are determined by the average prediction results.

4.3 Results

The prediction results derived for different loss functions are provided in Table 1, which shows the mIoU, OA and class-wise F_1 scores. The Table 2 details the results of the different weight transformation strategies. Figure 5 and Figure 6 show the effects of the different search regions. Finally, we show the performance of our proposed method on different networks in Table 3 and visualizations of the predictions in Figure 7.

5. DISCUSSION

5.1 Effectiveness of Proposed NPALoss

In this section, we compare the performance of different loss functions on FCN (Shelhamer et al., 2017). As shown in Table 1, we compare our NPALoss with class weighted loss (Badrinarayanan et al., 2017), focal loss ($\gamma = 2$) (Lin et al., 2017) and dice loss (Milletari et al., 2016). The class weighted denotes the weighted cross-entropy loss with a medium frequency balancing on the classes. Note that we do not use a weight on the background class. Instead, we apply the same weight on this class as the lowest weight on all the other classes. The results reveal that our proposed NPALoss makes further improvement on small-sized objects and object boundaries. Using our proposed NPALoss instead of the standard cross-entropy loss function, our method yields an improvement of 2.07% in mF_1 score, 2.57% in mIoU, respectively. Especially for the Car class, a typical small sized objects, the F_1 score can be improved by 8.44 points. Besides, the performance of the classes with complex boundaries, such as the Impervious Surfaces and Low vegetation class, also achieves significant improvements. On the contrary, the focal loss and the dice loss result in poor performance, which implied that these loss functions do not suitable for the multi-classes semantic segmentation tasks. Besides, applying the class weighted cross-entropy loss achieves a slight improvement on mIoU and mF_1 score while the OA shows a drop. This drop is due to the F_1 score of the Low vegetation class drops from 78.57% to 78.38%. Although the performance of minority classes are improved, the performance of other classes may be degraded. As a comparison, our method can achieve promising results in each class, especially for small-sized classes and irregularly shaped classes.

5.2 Effects of Weight Transformation

The weight transformation strategies will affect accuracy. We explored the effects of different strategies, and the results can be seen in Table 2. When using the normalization strategy to generate weights from 0 to 1, only the F_1 score of the Car class shows a significant improvement. However, the performance of the other four classes is degraded. This is because the weights of most pixels belonging to these classes are set to 0. When setting the value of L to 0.5, the mIoU is further increased by 1.69%. However, the F_1 score of the Low vegetation class is still lower than the baseline network. We think the weight margin between the well-classified and hard pixels is not enough. In order to enlarge the margin, we use the logarithm compression strategy and test the effects of the different bases of the logarithm. The results in Table 2 show that the logarithm compression strategy produces a better segmentation accuracy, and the common logarithm ($a = 10$) is better than natural logarithm ($a = e$). The OA of common logarithm reaches 75.01% and 88.33%, respectively. Therefore, it is crucial to choose a suitable weight transformation strategy, and too large or too small weight margin will affect the performance.

5.3 Effects of Search Region

The size of the search region k also produces different performance gains for the segmentation network. As shown in Fig-

Method	Imp. Surf.	Building	Low Veg.	Tree	Car	mF_1	mIoU	OA
FCN-dilated	91.98	95.68	80.18	88.25	87.94	88.80	80.70	89.35
+ NPALoss (ours)	**92.04**	**95.81**	**80.69**	**88.43**	**90.63**	**89.52**	**81.39**	**89.56**
PSPNet	91.96	95.91	80.55	88.38	88.66	89.09	80.83	89.51
+ NPALoss (ours)	**92.44**	**95.93**	**81.38**	**88.58**	**89.56**	**89.58**	**81.47**	**89.84**
DeepLabV3	92.33	96.03	80.96	88.85	88.02	89.24	80.94	89.83
+ NPALoss (ours)	**92.48**	**96.10**	**81.44**	**88.85**	**90.27**	**89.83**	**81.88**	**90.00**

Table 3. The performance of our proposed NPALoss on different segmentation networks.

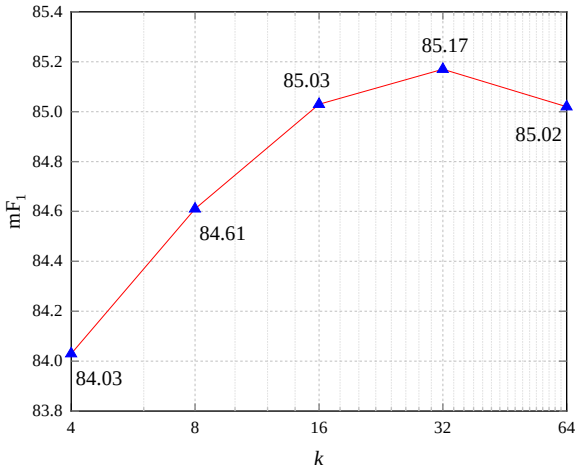


Figure 5. Evaluation on the influence of the search region k.

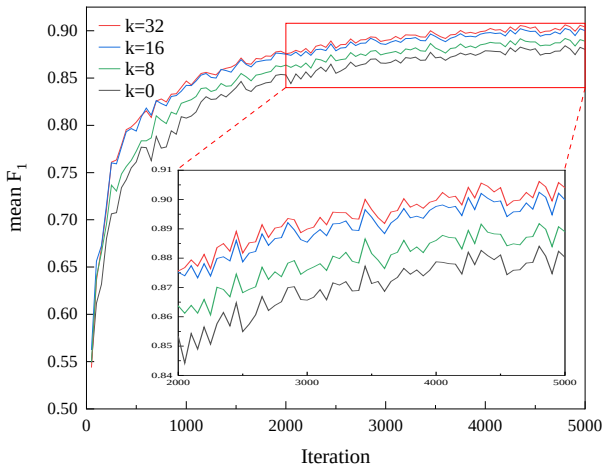


Figure 6. The mF_1 score curves for the training process with different search region k. $k = 0$ means the baseline network.

ure 4, there are huge differences between the different affinity maps. When the search region $k = 8$, only the pixels around boundaries have large weights. With the increase of the search region, the small-sized objects are also beginning to receive attention. For example, the weights of the pixels corresponding to the Car class are higher not only in the edge areas but also in the center areas when $k = 32$. Therefore, we explored the effects of the different search regions k. As shown in Figure 5, when $k = 8$, the results show an improvement from 83.10% to 84.61%, and when the search region becomes larger, the mF_1 score increased again. It is found that the mF_1 score is highest at $k = 32$, which reaches 85.17%. However, the performance will drop when the search region is further enlarged. Because the over-sized search region may begin to assign high weights to well-classified pixels, which reduce the network's attention to hard pixels. Furthermore, we show the mF_1 score curves for the training process. As shown in Figure 6, with the same number of iterations, the NPALoss achieves better performance, which proves our method can help the network converge quickly and improve the performance.

5.4 Robustness on Various Networks

To further prove the robustness of our proposed method, we use our NPALoss on FCN-dilated (Yu, Koltun), PSPNet (Zhao et al., 2017) and DeepLabV3 (Chen et al., 2017b), respectively. The FCN-dilated introduces the dilated convolution (Chen et al., 2017a) in the backbone network to enlarge the receptive field. The PSPNet and DeepLabV3 add the PPM module and ASPP module based on the FCN-dilated, respectively. The experimental results are presented in Table 3. The performance of the classes with small sized objects and complex boundaries is further improved. The mF_1 score of the three models can be improved by 0.72 points, 0.49 points, and 0.45 points, respectively. Note that the performance of FCN-dilated with our proposed NPALoss has exceeded the performance of PSPNet and DeepLabV3, which means the performance gains of our proposed NPALoss function exceeds the gains of the PPM module and ASPP module.

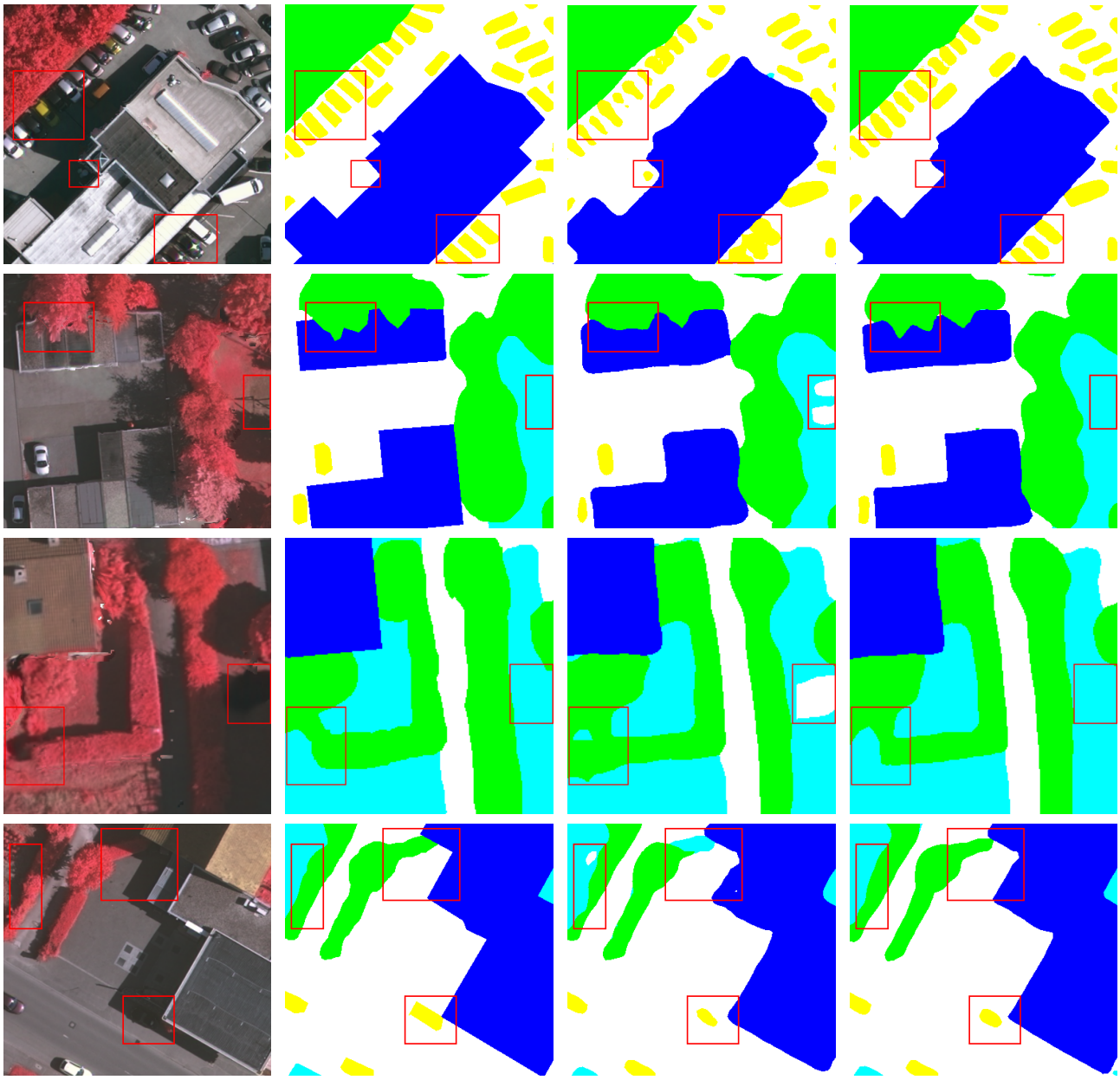
5.5 Visual Improvements

Figure 7 shows the visual results of DeepLabV3 and our proposed method. Although cars are only a few pixels and are parked densely, our method can distinguish each other. Despite being affected by shadows between adjacent classes, our method can produce more accurate boundaries. The improvement of segmentation quality of small-sized objects and object boundaries is obvious by introducing our NPALoss.

6. CONCLUSION

In this work, we propose a neighboring pixel affinity loss (NPALoss) for semantic segmentation in high-resolution aerial imagery to improve the segmentation accuracy of small-sized objects and object boundaries. Firstly, we convert the problem of how to determine the classifying difficulty of one pixel into a problem that the pixel categories in the neighborhood are the same or different. Therefore, we build an affinity map by counting the pixel-pair relationships in the search region. Then, we further investigate different weight transformation strategies to explore the suitable weight margin of the affinity map and avoid the problem of gradient overflow. We find that too large or too small weight margins will affect the performance of our proposed NPALoss. The logarithm compression strategy is better than the normalization strategy, especially the common logarithm. Combining the affinity map and the weight transformation strategy, our NPALoss helps the network pay more attention to the pixels corresponding to small-sized objects and object boundaries. Experiments on the ISPRS Vaihingen dataset and various segmentation networks prove the effectiveness

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-2-2020-475-2020 | © Authors 2020. CC BY 4.0 License.



(a) True orthophoto (b) Ground truth (c) DeepLabV3 (d) Ours

Figure 7. Visualization of the ground truth and the segmentation results achieved with the DeepLabV3 and our method. There is a significant improvement on small-sized objects and object boundaries.

and robustness of our method. In the future, we will introduce multi-scale affinity maps to build the NPALoss. Furthermore, an adaptive weight transformation strategy is also in our plan.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (No. 41701508).

References

Audebert, N., Le Saux, B., Lefèvre, S., 2016. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. *Asian conference on computer vision*, Springer, 180–196. 4

Audebert, N., Le Saux, B., Lefèvre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140, 20–32. 1, 2

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481–2495. 1, 2, 5

Chen, K., Weinmann, M., Gao, X., Yan, M., Hinz, S., Jutzi, B., Weinmann, M., 2018a. RESIDUAL SHUFFLING CONVOLUTIONAL NEURAL NETWORKS FOR DEEP SEMANTIC IMAGE SEGMENTATION USING MULTI-MODAL DATA. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(2). 4

Chen, K., Weinmann, M., Sun, X., Yan, M., Hinz, S., Jutzi, B., Weinmann, M., 2018b. SEMANTIC SEGMENTATION OF AERIAL IMAGERY VIA MULTI-SCALE SHUFFLING CONVOLUTIONAL NEURAL NETWORKS WITH DEEP SUPERVISION. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(1). 5

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., 2017a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834–848. 2, 6

Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017b. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*. 1, 2, 4, 6

Cramer, M., 2010. The DGPF-test on digital airborne camera evaluation–overview and test design. *Photogrammetrie-Fernerkundung-Geoinformation*, 2010(2), 73–82. 1, 2, 4

Eigen, D., Fergus, R., 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proceedings of the IEEE International Conference on Computer Vision*. 1, 2

Feng, Y., Diao, W., Sun, X., Yan, M., Gao, X., 2019. Towards Automated Ship Detection and Category Recognition from High-Resolution Aerial Images. *Remote Sensing*, 11(16), 1901. 2

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H., 2019. Dual attention network for scene segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1, 2

Fu, K., Chang, Z., Zhang, Y., Xu, G., Zhang, K., Sun, X., 2020. Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 161, 294–308. 5

Gerke, M., 2014. Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen). 4

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. 2, 4

Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q., 2017. Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W., 2019. Ccnet: Criss-cross attention for semantic segmentation. *Proceedings of the IEEE International Conference on Computer Vision*. 1, 2

Jeatrakul, P., Wong, K. W., Fung, C. C., 2010. Classification of imbalanced data by combining the complementary neural network and smote algorithm. *International Conference on Neural Information Processing*, Springer, 152–159. 1

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988. 1, 2, 5

Milletari, F., Navab, N., Ahmadi, S.-A., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, 565–571. 2, 5

Mostajabi, M., Yadollahpour, P., Shakhnarovich, G., 2015. Feedforward semantic segmentation with zoom-out features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1, 2

Paisitkriangkrai, S., Sherrah, J., Janney, P., Hengel, V.-D. et al., 2015. Effective semantic pixel labelling with convolutional networks and conditional random fields. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 36–43. 4

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in PyTorch. *NeurIPS Autodiff Workshop*. 4

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U., 2012. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1*, 1(1), 293–298. 1, 2, 4

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al., 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252. 4

Shelhamer, E., Long, J., Darrell, T., 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4), 640–651. 1, 2, 4, 5

Shrivastava, A., Gupta, A., Girshick, R., 2016. Training region-based object detectors with online hard example mining. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2

Volpi, M., Tuia, D., 2016. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), 881–893. 4

Wang, P., Sun, X., Diao, W., Fu, K., 2019. FMSSD: Feature-Merged Single-Shot Detection for Multiscale Objects in Large-Scale Remote Sensing Imagery. *IEEE Transactions on Geoscience and Remote Sensing*. 2

Wang, X., Girshick, R., Gupta, A., He, K., 2018. Non-local neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2

Wu, Z., Shen, C., Hengel, A. v. d., 2016. High-performance semantic segmentation using very deep fully convolutional networks. *arXiv preprint arXiv:1604.04339*. 2

Yu, F., Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. *Proceedings of the International Conference on Learning Representations*. 2, 4, 6

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2881–2890. 1, 2, 4, 5, 6