Calibrated Focal Loss for Semantic Labeling of High-Resolution Remote Sensing Images

Haiwei Bai, Jian Cheng, Yanzhou Su, Siyu Liu, Graduate Student Member, IEEE, and Xin Liu

Abstract—Currently, the most advanced high-resolution remote sensing image (HRRSI) semantic labeling methods rely on deep neural networks. However, HRRSIs naturally have a serious class imbalance problem, which is not yet well solved by the current method. The cross-entropy loss is often used to guide the training of semantic labeling neural networks for HRRSIs, but it is essentially dominated by the major classes in the image, resulting in poor predictions for the minority class. Based on the prediction results, focal loss (FL) effectively suppresses the negative impact of class imbalance in dense object detection by redistributing the loss of each sample. In this article, we thoroughly analyze the inadequacy of FL for semantic labeling, which inevitably introduces confusing-classified examples that are more difficult to classify while suppressing the loss of well-classified examples. Therefore, following the core idea of FL, we redefine the hard examples in semantic labeling of HRRSIs and propose the prediction confusion map to measure the classification difficulty. Based on this, we further propose the calibrated focal loss (CFL) for the semantic labeling of HRRSIs. Finally, we conduct complete experiments on the International Society for Photogrammetry and Remote Sensing Vaihingen and Potsdam datasets to analyze the semantic labeling performance, model uncertainty, and confidence calibration of different loss functions. Experimental results show that CFL can achieve outstanding results compared with other commonly used loss functions without increasing model parameters and training iterations, demonstrating the effectiveness of our method. In the end, combined with our previously proposed HCANet, we further verify the effectiveness of CFL on state-of-the-art network structures.

Index Terms—Calibrated focal loss (CFL), class imbalance, focal loss (FL), high-resolution remote sensing images (HRRSIs), prediction confusion map (PCM), semantic labeling.

I. INTRODUCTION

Semantic labeling (or semantic segmentation in computer vision) of high-resolution remote sensing images (HRRSIs) aims to assign a certain semantic category to each pixel in the image, which is a crucial step in the practical application of remote sensing (RS) [1], [2], [3]. With the development of aerospace and sensor technology, more and more HRRSIs are obtained, and their resolutions are gradually increasing. These images are widely used in land resource [4], [5] and ecological environment monitoring [6], [7], natural disaster detection [8], [9], [10] and urban development planning [11], [12].

At present, the most advanced semantic labeling methods of HRRSIs rely on deep neural networks [13], [14], [15], [16], [17]. These networks are all derived from fully convolutional networks (FCN), which was first proposed by Long et al. [18]. It uses a fully convolutional layer to replace the fully connected layer in the classification network so that the network maintains its 2-D high-level semantic features and then restores the resolution of the image through upsampling to obtain a semantic mask. FCN has made a milestone breakthrough in the task of image semantic segmentation. Since then, the semantic segmentation networks based on deep learning (DL) have been flourishing [19], [20], [21], [22]. The deep neural networks learn data distribution characteristics from a large number of annotated images and update the model parameters through the backpropagation algorithm [23] to make it fit the data distribution of the training dataset as much as possible. Therefore, data are the core driving force of deep neural networks, and data with a balanced distribution are more conducive to the representation of deep neural networks.

However, HRRSIs reflect the object information of different categories on the earth’s surface, and these objects inherently have the problem of class imbalance. For example, roads in cities occupy only a small part of the urban land area, while small objects, such as cars, take up even smaller space. However, the remaining large categories (impervious surfaces, buildings, low vegetation, and trees) occupy the vast majority of the ground. In HRRSIs, the specific manifestation is that these small objects occupy fewer pixels in the image, leading to class imbalance. In the 1990 s, Anand et al. [24] explored the impact of class imbalance in the classification neural networks on the backpropagation algorithm. They show that in class imbalanced scenarios, the length of the minority class’s gradient component is much smaller than that of the majority class. In other words, the majority class dominates the gradient that guides the model parameter update. This makes the error of the majority class decrease rapidly in the early training iterations, while the error of the minority category often increases. In the subsequent training, the error reduction rate of the minority class is very small, resulting in a slower net error convergence speed. This makes the semantic labeling of HRRSIs with imbalanced data very difficult.
In DL, there are mainly two approaches to deal with class imbalance. One is to process the training data by resampling to balance the number of samples of different classes in the training set. Such methods mainly include using random undersampling (RUS) to reduce the samples in large categories or using random oversampling (ROS) to increase the samples in small groups [25]. The other is to improve the algorithm mainly including the design of the network structure and loss function. However, semantic labeling networks designed for class imbalance problems are extremely rare. In contrast, the purposeful design of reasonable loss functions is more common. The representative method is to increase the weight of the loss for samples that are difficult to classify and assign larger gradients to them in backpropagation, forcing the model to pay more attention to them. These loss functions can usually effectively deal with the problem of class imbalance, and ultimately achieve excellent performance.

However, from the perspective of the specific implementation, resampling the training data for semantic labeling, which is essentially a pixel-level classification task, is very difficult. For example, small objects, such as cars, have a relatively small number of pixels and are often submerged in the background in HRRSIs. Performing ROS on such objects will inevitably increase background samples, such as roads, buildings, and trees, and cannot effectively increase the proportion of minority groups. Furthermore, the RUS for majority groups will reduce the overall training data, which may cause the model to be more prone to overfitting. Therefore, a more reasonable way is to design excellent algorithms to deal with the class imbalance in the semantic labeling of HRRSIs. Therefore, in this article, we design a novel loss function suitable for the semantic labeling of HRRSIs, which is called calibrated focal loss (CFL).

The CFL is designed based on focal loss (FL) [26], which forces the network to pay more attention to hard examples by suppressing the loss of easy examples and has achieved promising results in dense object detection. In this article, we found that when dealing with semantic labeling, FL will inevitably introduce confusing-classified examples while suppressing the loss of well-classified examples. This phenomenon is shown in Fig. 1. Due to the nonlinear suppression of FL, the range of loss value for these samples is very small. Therefore, the difference between the losses of positive and negative samples is very small, which makes the loss value insensitive to whether the sample is correctly classified. In this manuscript, we call them confusing-classified examples. The classification of these confusing-classified examples is often entangled between some two extremely hard-to-classify classes of all classes. More seriously, these confusing-classified examples widely exist in HRRSIs. They mainly include edges where different objects touch each other or objects of different classes with very small interclass distances.

To address this problem, we first propose a prediction confusion map (PCM) based on the prediction results to measure the classification difficulty of each pixel in the HRRSIs. Then, according to the classification difficulty of image pixels, we construct a calibration term for FL. This item is essentially a weighted cross-entropy (CE) loss whose weights are the inverse of PCM. The purpose of adding this item is to make the network pay more attention to the confusing-classified examples introduced by FL, thereby improving the performance of the model. We conducted sufficient experiments on two HRRSI semantic labeling datasets of International Society for Photogrammetry and Remote Sensing (ISPRS) to verify the effectiveness of the method. Furthermore, we analyzed the uncertainty of the network model based on the Monte Carlo Dropout method [27]. At last, since the addition of the calibration item may cause the model to be overconfident, we use expected calibration error (ECE) [28] to measure the confidence calibration of different loss functions. We found that compared with the common semantic labeling loss functions, the proposed CFL can help the models achieve better prediction results with lower model uncertainty and well confidence calibration.

To summarize, the main contributions of this article are threefold, which are as follows:

1) Proposing PCM to measure the difficulty of classification of each pixel in the HRRSIs. With the help of PCM, we further studied the defects of FL in the semantic labeling of HRRSIs.
2) Based on PCM, we propose the CFL for semantic labeling of HRRSIs, which can force the network to pay more attention to examples that are difficult to classify, thereby improving the performance of the network.
3) Performing extensive experiments to verify the effectiveness of our method on the ISPRS Potsdam and Vaihingen HRRSI semantic labeling datasets. Further, we use the Monte Carlo Dropout method and the ECEs to analyze the model uncertainty and confidence calibration of the models guided by different loss functions. Finally, we further validate the effectiveness of CFL on our previously proposed HCANet [29].

The rest of this article is organized as follows. Section II introduces the related work involved in this contribution. Then, we will describe our CFL in detail in Section III. In Sections IV and V, we present a detailed experimental evaluation and discussion.
to verify the effectiveness of the method and evaluate the model uncertainty and confidence calibration of the network guided by different loss functions. Furthermore, using CFL as the loss function, we retrain the previously proposed HCANet, and the final results are compared with other state-of-the-art methods. Finally, Section VI concludes this article.

II. RELATED WORK

A. Semantic Labeling of HRRSIs

In recent years, with the development of DL theory, a large number of semantic labeling models based on deep neural networks have emerged in RS. Most of them focus on designing sophisticated network structures to improve network performance. Many works focus on the edge information of objects, which is generally considered to be difficult to distinguish perfectly but very important for semantic labeling.

Marmains et al. [30] combined semantic labeling with semantic information edge detection, making class boundaries explicit in the model, thereby significantly improving the performance of the network. Liu et al. [31] added multiple weighted edge supervisions to the network to preserve the spatial boundary information, which can effectively reduce semantic ambiguity. Furthermore, Bokhovkin and Burnaev [32] proposed a boundary loss function to encourage the neural network to better take into account the boundaries. In addition, some works improved the performance of the network by introducing additional data. Kaiser et al. [33] and Audebert et al. [34] introduced open street maps to increase the amount of training data. Some other works learned additional feature representations from digital surface models [35], [36], [37], [38]. These works either focused on the edge information or introduced additional data to improve network performance.

However, additional data acquisition and processing will greatly increase research costs and workload. Moreover, samples that are difficult to classify are not only the edges of the objects but also the categories with small interclass distances. Yue et al. [39] stacked a tree convolutional neural networks (CNN) block adaptively constructed based on the confusion matrix and TreeCutting algorithm after the conventional semantic segmentation network, which is dedicated to improving the discrimination of easily confused classes. While this approach can improve network performance, the tree-CNN block brings additional model parameters, increasing the risk of overfitting. In addition, the network uses multistage training, which increases the difficulty of network training. All of these studies increase model parameters or training iterations. On the contrary, the novel semantic labeling loss function proposed in this article pays more attention to hard examples that ignore category and location differences.

B. Class Imbalance

HRRSIs reflect the information on the geophysical surface, whether natural or man-made, surface objects naturally have the characteristics of class imbalance. In DL, many scholars have paid a lot of effort to solve the problem of class imbalance, mainly including data resampling and loss function design.

1) Data Resampling: Data resampling is an effective way to address the negative effects of class imbalance. Hensman and Masko [40] made the CIFAR-10 dataset into several training sets with different data distributions to research the impact of class imbalance on the performance of CNN. The results show that imbalanced training data have a severely negative impact on the model performance of CNN. In other experiments in their work, oversampling the imbalanced training dataset improved performance to that of the balanced set. Therefore, it is concluded that oversampling is a viable way to counter the impact of class imbalances. Lee et al. [41] adopted a two-phase learning procedure, where a model is first pretrained with thresholded data with balanced data distribution, and then fine-tuned it using all data. They further proved that pretraining models with semibalanced data generated through RUS of majority groups or data augmentation for minority classes can significantly improve the performance of minority groups. In addition to RUS and ROS, Pouyanfar et al. [42] proposed a dynamic sampling technique to perform the classification of imbalanced data with a deep CNN. The basic idea is to oversample the low-performing classes and undersample the high-performing ones, which effectively suppresses the negative impact of class imbalance.

In semantic segmentation, Havaei et al. [43] also proposed a two-phase learning procedure to manage class imbalance when performing brain tumor image segmentation. They pointed out that the two-phase learning process was essential in dealing with the imbalanced distribution in their data. In their first training stage, small patches with a balanced number of positive and negative samples were used as the training set. This method is more convenient to implement for binary classification tasks, such as brain tumor segmentation, but it is not suitable for multiclass semantic labeling of HRRSIs. In summary, the resampling strategy can fundamentally change the data distribution of the training set to guide the model to learn more distinctive features and they are easy to implement for image classification. However, these methods are difficult to deploy for semantic labeling of HRRSIs, which uses pixels as basic samples.

2) Loss Function: In DL, a reasonable loss function designed for a specific task is usually independent of the construction of the model, and will not change the network structure or affect the training process. Therefore, it is a very promising and important academic research aspect for the development of DL. The most widely used loss function in semantic labeling of HRRSIs is CE loss, which is defined as a measure of the difference between two probability distributions for a given random variable or set of events. The weighted cross-entropy (WCE) loss is a variant of CE loss designed to deal with class imbalance, where samples from different classes are weighted by the reciprocal of their proportion in the total data.

Kampffmeyer et al. [44] reweighted the CE loss with median frequency balancing (MFB) [45], which effectively improved the classification accuracy for small classes in urban RS. While such a weighting strategy yields more balanced classification results by boosting small classes, larger groups tend to not improve after a certain threshold. Bischke et al. [46] proposed a new WCE loss based on the model uncertainty of the network guided by the MFB. For a given pixel, this method uses Monte Carlo Dropout [27] to draw ten Monte Carlo samples from the model uncertainty distribution and use their average as the prediction result. The weights of each sample are then calculated according to the model uncertainty, with higher uncertainty corresponding to higher weights. This method effectively balances the weights of different classes, ensuring that small classes are given more attention during training.

Furthermore, the network uses multistage training, which increases the risk of overfitting. In addition, the network uses multistage training, which increases the difficulty of network training. All of these studies increase model parameters or training iterations. On the contrary, the novel semantic labeling loss function proposed in this article pays more attention to hard examples that ignore category and location differences.
the prediction results to calculate the model uncertainty, which increases the forward calculation by ten times and, thus, greatly increases the model training time.

Li et al. [47] proposed a loss function that combines weighted binary CE loss and dice coefficient loss and effectively addresses the class imbalance problem in the change detection task. Yuan and Xu [48] made full use of the spatial correlation between a pixel and its neighboring pixels and proposed a NeighborLoss function, which achieved outstanding performance in the task of building segmentation. Although these two novel loss functions have achieved excellent performance in their respective binary segmentation tasks, they are not suitable for the multiclass RS image semantic segmentation task in this article.

Lin et al. [26] proposed FL that effectively addresses the extreme class imbalance commonly encountered in dense object detection problems, where positive foreground samples are heavily outnumbered by negative background samples. Because of its excellent performance, Doi and Iwasaki [49] used FL instead of CE loss to train the model and explore its impact on the model performance. However, the experimental results of this article show that FL does not always improve the performance of semantic segmentation of HRRSIs, and its effect is unclear. In Section III, we will analyze in detail the defects of FL when dealing with semantic labeling of HRRSIs. Based on this analysis, we propose the CFL, which is suitable for semantic labeling of HRRSIs.

When the label is a one-hot vector, the expressions of different loss functions for each pixel is

\[ FL = -\left(1 - p_i\right)^2 \cdot \log(p_i) \]
\[ CE = -\log(p_i) \]
\[ MFB = -\omega_c \cdot \log(p_i) \]
\[ \omega_c = \frac{\text{median}(f_c | c \in C)}{f_c} \]

where \(\omega_c\) is the weight for class \(c\), \(f_c\) is the proportion of samples in class \(c\) to the total data, \(p_i\) is the softmax probability, and \(C\) is the set of all classes.

C. Model Uncertainty and Confidence Calibration

The DL model cannot be widely applied in real applications if we do not know whether the model is certain about its decision or not. Therefore, the measurement of model uncertainty has received extensive attention in recent years [50], [51], [52]. However, there are relatively few studies on the uncertainty estimation of semantic labeling of HRRSIs based on DL. Gal and Ghahramani’s [27] proved that dropout training [53] in neural networks can be used as approximate Bayesian inference in Gaussian processes, which can represent model uncertainty in DL. Inspired by this, in this article, we use the Monte Carlo Dropout method to evaluate the uncertainty of the models guided by different loss functions to enhance the practical applicability of deep neural networks. This article obtains the uncertainty map by retrieving 16 Monte Carlo samples from the model predictions and then computing the standard deviation of the softmax probabilities of the samples.

Confidence calibration is the problem of predicting probability estimates representative of the true correctness likelihood, which is usually measured by ECE [28]. Guo et al. [54] found that deep neural networks are generally poorly calibrated indicating that the models are overconfident in their predictions. Mukhoti et al. [55] pointed out that the network models guided by FL in the classification task have been well calibrated. Inspired by these works, considering that the design idea of CFL may lead to the risk that the models are overconfident in their predictions, this article adopts the ECE to measure the confidence calibration of different loss functions. In the end, we use these results to investigate the impact of different loss functions on the semantic labeling of HRRSIs.

III. METHODOLOGY

A. Defects of FL for Semantic Labeling

When dealing with classification issues, FL reduces the loss weights of the well-classified examples so that the training of the network is concentrated on a set of sparse hard examples, and prevents a large number of easy negative numbers from overwhelming the classifier during training, which can effectively suppress the negative impact of class imbalance. The core idea of FL can perfectly counter the thorny problem of data imbalance between positive and negative samples.

However, our experimental results show that the performance of FL is relatively poor when dealing with the dense prediction task of semantic labeling. Through careful thinking and analysis, we found that this may be because FL suppresses the loss of well-classified examples while making it more difficult to classify those samples that are wandering between some two classes when dealing with semantic labeling of HRRSIs. In this article, we name them as confusing-classified examples instead of hard examples. This is because hard examples have relatively large loss values, which can gain more attention from the model. However, the losses of these confusing-classified examples are relatively small, resulting in a lack of attention from the model.

As shown in Fig. 1, compared with the CE loss, FL suppresses the loss value of the confusing-classified examples to a smaller value range, making the loss function curve flatter, which may reduce the representativeness of the loss value to the correctness of the classification result. Unfortunately, these confusing-classified examples widely exist in HRRSIs and are mainly composed of two categories. One is the samples with a very small interclass distance, and the other is the boundary part where objects of different classes touch each other. The second category samples are particularly prominent due to the intricacies of scenes in HRRSIs. The insufficient ability of FL to distinguish these two types of samples leads to its poor performance in the semantic labeling of HRRSIs.

B. Prediction Confusion Map

The concepts of well-classified examples and confusing-classified examples are essentially trying to reflect the difficulty of classification, which is based on intuitive understanding without a clear definition. In order to quantitatively measure the difficulty of classification of each sample in semantic labeling
of HRRSIs, this article proposes a PCM based on the prediction results of the network, where the lower the confusion, the simpler the classification of the sample. In multiclass semantic labeling, each sample is determined to have a unique class label; however, the sample may have a certain similarity with other classes. Therefore, the prediction result of the network for each pixel is a probability vector, namely \((p_1, p_2, \ldots, p_n)\), where \(n\) is the number of classes. Generally, the class corresponding to the maximum value \(\max_{i \in [1, n]} p_i\) is taken as the final classification result.

Conventional loss functions only use the predicted value \(p_i\) corresponding to the nonzero position of the one-hot vector label when calculating the loss, for example, FL, CE loss, and its variants, weighted WCE and MFB cross-entropy loss function, etc. During the training process, the network uses the loss information to optimize the model parameters through the back-propagation algorithm [23]. Obviously, for a given sample, in the entire training process of the model, only a certain \(p_i\) in the prediction probability vector is used, thus ignoring the association with other categories. This is equivalent to treating multiclass semantic labeling as a binary classification task that only contains positive and negative classes.

However, we believe that the likelihood that a sample belongs to other classes is useful for the correctness of its classification. Therefore, in this article, we introduce the probability information of the sample belonging to other classes in the prediction result to measure the difficulty of its classification, and based on this, we further propose the CFL. Here, we assume an example to express our idea.

For example, in the case of six-class classification, suppose there are two samples whose prediction results are

\[
s_1 = (0.5, 0.4, 0.1, 0, 0, 0) \\
s_2 = (0.5, 0.1, 0.1, 0.1, 0.1, 0.1)
\]

\(\max s_1 = 0.5 = \max s_2 = 0.5\)

but, intuitively, \(s_1\) has a higher prediction confusion than \(s_2\). This is because the gap between \(p_1 = 0.5\) and \(p_2 = 0.4\) in its prediction probability vector is relatively small. So that, the correct classification of \(s_1\) sample is more difficult than that of \(s_2\).

Based on this observation, we normalize the logits output through the Softmax function to get the predictive probability maps. For each pixel, we sort each \(p_i\) of the predictive probability vector in descending order. Then, take the difference value between the maximum and the second value in the ordinal sequence as the confusion level of the prediction result, which is called prediction confusion in this article. This process is shown in Fig. 2. The smaller the value, the higher the degree of confusion. As shown in Fig. 3, the PCM of the image can be obtained by extending this operation on the entire image.

\[
\text{prediction confusion} = r_1 - r_2 \\
r_i = \text{rank}(p_i), i \in [1, \text{number class}]
\]

where \(p_i\) represents the predicted probability that the sample belongs to each class, and \(\text{rank}(p_i)\) represents sorting these probability values in descending order.

Based on this definition, we still use the examples mentioned above to calculate their prediction confusion separately

\[
\text{pc}_{s_1} = 0.5 - 0.4 = 0.1 \\
\text{pc}_{s_2} = 0.5 - 0.1 = 0.4
\]
where \( p_{c,s_1} \) represents \( s_1 \)'s prediction confusion. The results show that using our measurement method, the \( s_1 \) sample has a higher prediction confusion, so it is harder to classify.

In the experiments of this article, we use PCM to measure the confusion of the predictions of different loss functions, which will be described in detail in the experimental part. The experimental results prove that compared with CE, FL will lead to higher prediction confusion and worse labeling result. This is completely consistent with our analysis that FL suppresses the loss of well-classified examples while making it more difficult to classify samples with small interclass distances or the edge regions of the objects that touch each other. Thus, the semantic labeling performance of the model decreases.

C. Calibrated FL

Class imbalance is a common problem in real-world classification tasks. In this article, we consider that class imbalance can be analyzed from two perspectives. One is the data distribution, which is manifested in the imbalance in the number of samples belonging to different classes or the number of positive and negative samples. At present, the common method to solve the problem of class imbalance from this perspective is to use WCE loss, which redistributes the weights of the losses according to the statistical results of samples of different classes. However, the weights are a set of nonlearnable hyperparameters defined by classes, which essentially represent strong prior knowledge of the data distribution and have no correlation with the prediction results of the model.

The other view is to divide the training data into easy and hard examples according to the degree of classification difficulty, and the famous FL is designed from this perspective. It directly measures the classification difficulty of the sample according to the \( p_i \) in the prediction result corresponding to the positive category in the label and suppresses the loss of easy samples, allowing the network to focus on training on a set of sparse and hard samples. From this perspective, the training process of the model completely ignores the category information of the training data but is closely related to the prediction results, which allows the model to continuously focus on hard examples during training, thereby achieving better classification results.

Generally in practical applications, the two perspectives of class imbalance are unified, that is, it is more difficult to classify examples with a small sample size. However, this is not the case in the semantic labeling of HRRSIs. For example, in the ISPRS Potsdam dataset, the prediction results of the minority class of cars are usually better than trees and low vegetation, which have large data volumes. This may be because, in the orthographic projection, the shape and structure of the cars are relatively fixed and, therefore, easier to learn and distinguish.

Since being proposed, FL and its variants have been widely used in various classification and detection tasks in DL, and have achieved excellent performance \([56, 57, 58, 59, 60, 61, 62, 63]\). Therefore, this article inherits the concept of FL, ignoring the class information of the samples and only consider the classification difficulty to construct CFL, trying to solve the problem of class imbalance in the semantic labeling of HRRSIs.

In the abovementioned, we analyze the defects of FL in dealing with semantic labeling tasks. It introduces confusing-classified examples when suppressing easy examples and highlighting hard examples. Furthermore, we propose a quantitative measure of the difficulty of sample classification based on the prediction results, namely PCM. Based on these analyses, we propose the CFL, which adds a calibration item to FL to increase the loss of the confusing-classified examples, forcing the network to pay more attention to these samples. The calibration term is essentially a pixel-by-pixel WCE loss, and its weight is established based on the PCM. The specific formula is expressed as

\[
\text{CFL} = -\alpha \cdot w_c \cdot \log(p_i) + \text{FL}
\]

\[
w_c = 1 - (r_1 - r_2)
\]

\[
r_i = \text{rank}(p_i), i \in [1, \text{number class}]
\]

\[
\text{FL} = -(1 - p_i)^2 \cdot \log(p_i).
\]

Consistent with PCM, \( \text{rank}(p_i) \) represents the descending order of the model prediction probability vector, \( \alpha \) is the reconciliation parameter, which is used to adjust the ratio of the loss of the confusing-classified example to the overall loss (we found \( \alpha = 2.5 \) to work best in our experiments).

IV. EXPERIMENT

In this section, we use two classic semantic segmentation networks FCN \([18]\) and DeepLabV3+ \([64]\) to conduct experimental analysis on two HRRSI semantic labeling datasets of ISPRS. This article uses FL as the baseline to compare the network performance, model uncertainty, and confidence calibration of different loss functions, including CFL, FL, CE loss, and a variant of WCE loss, that is, MFB cross-entropy loss (MFB). Furthermore, in the end, we combine our previously proposed state-of-the-art HCANet \([29]\) to verify the effectiveness of CFL in this article. We also compare the final results with other recent state-of-the-art methods.

A. Dataset

This article uses the ISPRS Vaihingen and the Potsdam dataset for experimental analysis. Both of these datasets are HRRSIs covering urban scenes. Vaihingen is a relatively small town with many independent buildings and small multistory buildings, while Potsdam is a typical historic city with huge buildings, narrow streets, and dense settlement structures. Therefore, the ground object characteristics of the two datasets are quite different.

1) Vaihingen Dataset: The ISPRS Vaihingen dataset contains a top view of Vaihingen, Germany. The dataset contains 33 images, and each image contains six classes (including background/clutter). The average size of each image is 2494 \( \times \) 2064 pixels, and each has three bands, which are near-infrared (NIR), red (R), and green (G) wavelengths. The ground sampling distance is 9 cm. We follow the official division method, using 16 pictures for training and 17 pictures for testing. Since the number of background samples in this dataset is very small, just
like the official benchmark, we exclude the background when calculating the scores of each class.

2) Potsdam Dataset: The Potsdam dataset consists of 38 images, each with four channels [NIR, R, G, and blue (B)]. In order to use the same type of images as the Vaihingen dataset, we use three bands of NIR, R, and G. The size of each image is 6000 × 6000 pixels, and it is annotated with six classes like the Vaihingen dataset. The ground sampling distance is 5 cm. Similarly, according to the official division method, we use 24 pictures for training and 14 pictures for testing.

B. Experiment Setup

In order to verify the effectiveness of our method, we use FCN8s with Vgg-16 as the backbone and DeepLabV3+ with ResNet-101 as the backbone for experiments. Both Vgg-16 and ResNet-101 are initialized with the pretrained model on the ImageNet dataset. We use atrous convolution in the fourth block of ResNet-101 to preserve the resolution of the feature map and set output stride = 8. For a detailed description of the HCANet network structure, refer to [29]. In terms of the data, since the resolution of the RS images in the ISPRS datasets is too large to be directly used for training, we adopt the sliding window method to crop all training images to a size of 512 × 512 with an overlap ratio of one-third. In the end, we got 705 training images on the Vaihingen dataset and 7776 training images on the Potsdam dataset. The test pictures were cropped in a noncovered order to use the same type of images as the Vaihingen dataset, we use the three bands of NIR, R, and G. We conduct experiments in the Pytorch-1.7 DL framework, use the stochastic gradient descent algorithm to optimize the model, setting momentum = 0.9, and use the poly learning strategy [65] for training, setting \( lr_{\text{init}} = 0.01 \) and batch size = 8

\[
\text{lr} = lr_{\text{init}} \times \left(1 - \frac{\text{current\_iterations}}{\text{max\_iterations}}\right)^{0.9}
\]

The data augmentation operations include random flipping and random cropping. In the uncertainty analysis experiment, we use the Monte Carlo dropout method, set dropout=0.1, and calculate the standard deviation of the 16 prediction results as the final result.

C. Evaluation Metrics

In this article, we use three common semantic labeling metrics to evaluate the performance of different losses, including overall accuracy (OA), mean IoU (mIoU), and mean \( F_1 \). In addition, the Intersection over Union (IoU) scores and \( F_1 \) scores of each class are also analyzed and compared as follows.

1) OA: The overall pixelwise accuracy for test dataset is given as follows:

\[
\text{OA} = \frac{TP}{TP + FP + TN + FN}.
\]

2) \( F_1 \): The \( F_1 \) score for each class is given as follows:

\[
F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.
\]

3) Mean \( F_1 \): The mean \( F_1 \) score for all classes.

4) IoU: The IoU for each class is given as follows:

\[
\text{IoU} = \frac{TP}{FP + FN + TP}.
\]

5) mIoU: The mIoU score for all classes.

V. Results

A. Ablation Study for Reconciliation Parameter \( \alpha \)

In order to determine the optimal weight of the calibration term in CFL, that is, the value of \( \alpha \), we use FCN8s to perform ablation experiments on the ISPRS Vaihingen and Potsdam datasets. The experimental results are given in Table I. It can be seen from the table that when \( \alpha = 0 \), CFL is equivalent to FL, when \( \alpha \neq 0 \), the calibration item is added to FL, and its performance is improved. In the experiment of this article, when \( \alpha = 2.5 \), CFL got the best results on both datasets. Therefore, all experiments in this article have \( \alpha \) set as 2.5.

B. Visualization Curve of Training Process

In the training phase, we use Tensorboard to observe and record our training process. We separately recorded the training loss and the test indicators after each training epoch, including OA, mean \( F_1 \), and mIoU scores. Fig. 4 shows the training process of DeepLabV3+ on the Vaihingen dataset. We set smoothing = 0.6 to facilitate visual observation. It can be seen from Fig. 4 that no matter which loss function is used, the network can always converge to a certain local optimal solution through training, and the performance on the test set has been steadily improved. More specifically, CFL always performed best on the test set.

C. Experimental Results of FCN8s

Table II tabulates the numerical experiment results of FCN on the Vaihingen dataset. Compared with FL, CFL has made a great improvement and achieved the best experimental results except for trees. The scores of OA, mean \( F_1 \), and mIoU are increased by 1.28%, 1.87%, and 2.68%, respectively. In addition, three scores of CFL exceeded CE by 0.35%, 0.9%, and 0.74%, respectively. An interesting conclusion is that MFB does not help the improvement of CE’s performance. Fig. 5 shows the visualized results of FCN on the Vaihingen dataset. It can be seen from Fig. 5 that PCMs with different loss functions have similarities, that is, areas with higher confusion are mostly concentrated near the edges of the objects. Comparing columns c and d, it can be seen that the PCM, errormap, and visual prediction results of CE and MFB are not distinguishable, which is consistent with the numerical results. In contrast, the PCM of FL has the largest area of the dim part and spreads outward along the edges, which is caused by a large number of confusing-classified examples in this area. However, the PCM of CFL is the brightest and the edges of the objects are clear, which shows that the addition of the calibration item greatly reduces the confusion of the confusing-classified examples. Finally, comparing their errormaps and prediction results, it can be seen that CFL has better performance on the edges of the objects, and has the most
Table I
Comparisons of Different $\alpha$ for CFL.

| $\alpha$ | Vaihingen OA | mIoU | mean $F_1$ | Potsdam OA | mIoU | mean $F_1$ |
|----------|---------|------|---------|---------|------|---------|
| 0        | 87.96   | 76.35 | 85.66   | 87.95   | 75.34 | 84.68   |
| 0.5      | 88.79   | 78.21 | 86.88   | 88.51   | 75.52 | 84.70   |
| 1.0      | 88.81   | 78.47 | 86.98   | 88.91   | 75.85 | 85.08   |
| 1.5      | 88.97   | 78.55 | 87.27   | 89.05   | 75.89 | 84.98   |
| 2.0      | 89.22   | 78.86 | 87.29   | 89.23   | 76.35 | 85.36   |
| 2.5      | **89.24** | **79.03** | **87.53** | **89.38** | **76.82** | **85.46** |
| 3.0      | 89.13   | 78.67 | 87.21   | 89.26   | 76.17 | 85.20   |
| 3.5      | 89.07   | 78.54 | 87.1    | 89.21   | 76.16 | 85.20   |
| 4.0      | 88.90   | 77.84 | 86.73   | 89.19   | 76.13 | 85.16   |

The bold values in tables represent the best results.

Table II
Experimental Results on the Vaihingen Dataset of FCN

| $F_1$ | OA      | Imp. surf. | Build | Low veg | Tree | Car | mean $F_1$ |
|-------|---------|------------|-------|---------|------|-----|------------|
| CE    | 88.89   | 91.09      | 93.61 | 82.68   | 88.54 | 77.25 | 86.63      |
| MFB   | 88.53   | 90.58      | 93.02 | 82.50   | 88.52 | 76.32 | 86.19      |
| FL    | 87.96   | 90.39      | 92.33 | 81.15   | 87.91 | 76.47 | 83.66      |
| CFL   | **89.24** | **91.26** | **94.31** | **83.15** | **88.45** | **80.46** | **87.53** |
| IoU   | 88.89   | 83.64      | 87.98 | 70.47   | **79.44** | 62.93 | 78.29      |
| MFB   | 88.53   | 82.78      | 86.96 | 70.22   | 79.40 | 61.70 | 77.33      |
| FL    | 87.96   | 82.46      | 85.75 | 68.28   | 78.52 | 61.90 | 76.35      |
| CFL   | **89.24** | **83.93** | **89.23** | **71.17** | **79.34** | **67.31** | **79.03** |

The bold values in tables represent the best results.

Fig. 4. Training process of DeepLabV3+ on Vaihingen dataset. (a) Train loss. (b) Test mean $F_1$. (c) Test mIoU. (d) Test OA.

Accurate result. The most prominent is the refined predictions of building edges.

Table III tabulates the numerical experimental results of FCN on the Potsdam dataset. As shown in the table, CFL has achieved the best results in all indicators. Compared with FL, CFL increased OA, mean $F_1$, and mIoU scores by 0.8%, 0.78%, and 1.18%, respectively. In addition, compared with CE, the scores of OA, mean $F_1$, and mIoU increased by 0.49%, 1.24%, and 1.45%, respectively. In this experiment, although MFB did not help improve CE’s OA score, it did improve mean $F_1$ and mIoU. Fig. 6 shows the visualized results of FCN on the Potsdam dataset. Consistent with the performance on the Vaihingen dataset, FL has the highest confusion of prediction, while CFL has the lowest. It is obvious that the wrong predictions are concentrated in the shadow area of the trees in the lower right corner of the picture, where CE, MFB, and FL all have different degrees of prediction errors. Interestingly, the higher confusion part of CE is concentrated in the boundary part of the error area, while the confusion of the central area is lower. However, CE still predicts the central region incorrectly, which indicates that CE is overconfident in its prediction results. In contrast, CFL has the lowest degree of confusion, whose PCM and prediction results are consistent, and has obtained almost perfect prediction.

D. Experimental Results of DeepLabV3+

Table IV tabulates the numerical experiment results of DeepLabV3+ on the Vaihingen dataset. Obviously, CFL achieved the best results on all indicators. Compared with FL, CFL increased by 0.88%, 1.86%, 2.84% in OA, mean $F_1$, and mIoU scores respectively. In addition, CFL exceeded CE 0.43%, 0.89%, 1.42% in OA, mean $F_1$ and mIoU scores respectively. The results of MFB are slightly better than that of CE. Fig. 7
Fig. 5. Visualization experimental results for FCN on the Vaihingen dataset. The first row is the input image and label, and the PCMs corresponding to different loss functions. The second row is the errormaps, in which the black part is the correct labeling result, and the rest are wrong. The third row is the labeling result corresponding to each loss function. (a) Image. (b) Label. (c) CE. (d) MFB. (e) FL. (f) CFL.

TABLE III

| $F_1$  | OA   | Imp. surf. | Build | Low veg | Tree | Car | Clutter | mean $F_1$ |
|--------|------|------------|-------|---------|------|-----|---------|------------|
| CE loss    | 88.89 | 91.03      | 94.58 | 85.61   | 87.22 | 94.42 | 52.49   | 84.22      |
| MFB loss   | 88.70 | 91.30      | 94.38 | 85.52   | 87.43 | 92.82 | 56.61   | 84.68      |
| FL         | 88.58 | 90.78      | 94.04 | 85.47   | 86.79 | 94.43 | 56.57   | 84.68      |
| CFL        | 89.38 | 91.72      | 94.72 | 86.30   | 87.49 | 94.89 | 57.64   | 85.46      |

The bold values in tables represent the best results.

TABLE IV

| $F_1$  | OA   | Imp. surf. | Build | Low veg | Tree | Car | Clutter | mean $F_1$ |
|--------|------|------------|-------|---------|------|-----|---------|------------|
| CE loss    | 89.75 | 92.01      | 94.65 | 83.07   | 88.87 | 83.6  | 8.44    |
| MFB loss   | 89.84 | 92.16      | 94.69 | 83.38   | 88.76 | 84.22 | 8.64    |
| FL         | 89.30 | 91.49      | 94.14 | 82.91   | 88.41 | 80.41 | 8.47    |
| CFL        | 90.18 | 92.46      | 95.10 | 83.42   | 88.99 | 86.66 | 8.33    |

The bold values in tables represent the best results.

shows the visualized results of DeepLabV3+ on the Vaihingen dataset. Consistent with FCN, the PCM of FL has the lowest brightness corresponding to higher confusion, while CFL has greatly improved it. The four PCMs consistently show high confusion on the road surface in the middle part of the picture, and they all have different degrees of prediction errors in this area. Comparing the prediction outputs and the errormaps, it can be seen that CFL has the best prediction result for the middle part, and MFB has also improved the performance of CE to a certain extent. It is necessary to note that, consistent with FCN’s
Fig. 6. Visualization experimental results for FCN on the Potsdam dataset. (a) Image. (b) Label. (c) CE. (d) MFB. (e) FL. (f) CFL.

Fig. 7. Visualization experimental results for DeepLabV3+ on the Vaihingen dataset. (a) Image. (b) Label. (c) CE. (d) MFB. (e) FL. (f) CFL.

prediction results on the Potsdam dataset, the high confusion part of CE still tends to concentrate on the edge of the wrong prediction area, which once again shows that CE is overconfident in its prediction results. Finally, unfortunately, for the narrow road on the left-hand side of the picture, these four losses are all predicted incorrectly, and consistently predicted the road as low vegetation. This requires the neural network model to extract more distinctive features.

Table V tabulates the numerical experiment results of DeepLabV3+ on the Potsdam dataset. It can be seen from the table that CFL achieved the best results except for low vegetation and trees. Compared with FL, CFL has increased by 0.11%, 0.38%, and 0.46% in OA, mean $F_1$, and mIoU indicators. Compared with CE, OA has not exceeded much, while mean $F_1$ and mIoU have exceeded by 0.47% and 0.52%, respectively, and MFB also slightly improves mean $F_1$ and mIoU. Fig. 8 shows the visualized results of DeepLabV3+ on the Potsdam dataset. Consistent with other experimental results, FL has the highest prediction confusion, while CFL greatly reduces it and finally achieves the best prediction results. The performance improvement of MFB is still not significant.

In general, the addition of the calibration item can always reduce the prediction confusion of FL, thereby improving its performance. However, MFB does not always improve the performance of CE. In addition, comparing PCM and prediction results, we find that CE tends to be overconfident in its prediction results. In the following experiments, we will analyze the impact of different losses on model uncertainty, and finally, calculate the ECEs of the predicted results to analyze their prediction calibration.
TABLE V
EXPERIMENTAL RESULTS ON THE POTSDAM DATASET OF DEEPLABV3+

|  | OA | Imp. surf. | Build | Low veg | Tree | Car | Clutter | mean F1 |
|---|---|---|---|---|---|---|---|---|
| CE loss | 90.67 | 93.14 | 96.48 | 87.07 | 88.01 | 95.45 | 59.80 | 86.66 |
| MFB loss | 90.64 | 93.07 | 96.40 | 87.17 | 88.14 | 95.46 | 61.66 | 86.98 |
| FL | 90.63 | 93.1 | 96.54 | 86.95 | 87.69 | 95.46 | 60.74 | 86.75 |
| CFL | 90.74 | 93.16 | 96.63 | 87.02 | 87.77 | 95.70 | 62.49 | 87.13 |
| IoU mean | mIoU |
| CE loss | 90.67 | 87.15 | 91.21 | 77.10 | 78.59 | 91.30 | 42.65 | 78.33 |
| MFB loss | 90.64 | 87.03 | 93.05 | 77.25 | 78.79 | 91.31 | 44.58 | 78.67 |
| FL | 90.65 | 87.09 | 93.31 | 76.91 | 78.08 | 91.32 | 43.62 | 78.39 |
| CFL | 90.74 | 87.20 | 93.49 | 77.03 | 78.20 | 91.75 | 45.44 | 78.85 |

The bold values in tables represent the best results.

Fig. 8. Visualization experimental results for DeepLabV3+ on the Potsdam dataset. (a) Image. (b) Label. (c) CE. (d) MFB. (e) FL. (f) CFL.

E. Uncertainty Analysis

The risk of the actual application of a DL model depends on the model’s confidence in its prediction results. Therefore, this article adopts the Monte Carlo dropout method to measure the model uncertainty of different loss functions.

Figs. 9 and 10 are the uncertainty maps of FCN and DeepLabV3+ with different loss functions on the Vaihingen dataset. In order to compare the effects of different network structures and loss functions on model uncertainty, we conducted experiments on the same test image. We display the errormap and the uncertainty map of each class to analyze the relationship between model performance and uncertainty. The first to fourth columns are the results of CE, MFB, FL, and CFL, respectively.

It can be seen from Fig. 9 that regardless of the loss function, the model uncertainty of the wrong prediction area is consistently higher. Moreover, CE and CFL express this consistency more accurately. Furthermore, comparing the uncertainty maps of different network structures, it can be found that the uncertainty maps of DeepLabV3+ have higher contrast, while that of FCN are lower. The prediction results of CE have a small area of error in the middle of the image, and its uncertainty is also scattered around the area which mainly hovers between buildings and low vegetation. The geographic structure of the lower part of the image is relatively complex. Most of the wrong predictions surround the boundaries, and the uncertainty in these areas is relatively high.

By observing the results of MFB in the second row, it can be found that the model uncertainty does not decrease compared with CE. More seriously, in the upper part of the image, the main uncertainty of the MFB model increases from the two classes of CE to the three classes of impervious roads, buildings, and low vegetation.

Comparing the third and fourth rows, it can be found that CFL can not only improve the performance of FL but also greatly reduce model uncertainty. In the upper half of the image, FL has a large area of prediction errors, and its uncertainty is mainly distributed in four classes, while the uncertainty of the cars in the middle of the image is also high. However, CFL compressed the uncertainty of these areas to a very low level, mainly between the two classes of low vegetation and trees that are difficult to distinguish.

In addition, by observing Fig. 10, we find that the DeepLabV3+ model shows almost the same characteristics as FCN. One difference from FCN is that the uncertainty of CFL in DeepLabV3+ is mainly between low vegetation and buildings.
Fig. 9. Monte Carlo dropout uncertainty maps of each class of FCN on the Vaihingen dataset. The first to fourth rows represent the prediction error maps and uncertainty maps of each class of CE, MFB, FL, and CFL. The higher the brightness, the stronger the uncertainty. (a) Error map. (b) Imp. surf. (c) Build. (d) Low veg. (e) Tree. (f) Car. (g) Clutter.

Fig. 10. Monte Carlo dropout uncertainty maps of each class of DeepLabV3+ on the Vaihingen dataset. (a) Error map. (b) Imp. surf. (c) Build. (d) Low veg. (e) Tree. (f) Car. (g) Clutter.
However, a detailed analysis of this phenomenon is beyond the scope of this study and remains to be researched. Figs. 11 and 12 are the uncertainty maps of FCN and DeepLabV3+ on the Potsdam dataset. Just like the performance on the Vaihingen dataset, the uncertainty of network models and their prediction errors show consistency. Similarly, the model uncertainty of MFB is not improved compared with CE. Observing Figs. 11 and 12, we find that FL also has a higher uncertainty in the area where its prediction is correct, which is manifested in the upper half of the input image. However, CFL significantly alleviates this phenomenon, so that the uncertainty of each class in the corresponding position in the picture is greatly reduced. It can be seen that on the Potsdam dataset, although the improvement of the performance of FL by CFL is relatively small from the experimental numerical results, it can greatly reduce the model uncertainty.

### F. Confidence Calibration Analysis

Table VI records the ECEs of the two networks guided by different loss functions on the Vaihingen and Potsdam test datasets. It can be seen from the table that, just as in [55], FL achieved the smallest ECEs, representing the best confidence calibration. However, the ECEs of CE are always the highest, which is consistent with the conclusion that CE has the risk of overconfidence in its prediction results drawn from the experimental results. However, comparing the ECE of CE and MFB and the experimental numerical results, it can be seen that although MFB has little help in improving the performance of CE, it can always slightly reduce its ECE. In contrast, CFL always increases ECE, leading to poor calibration. Despite the flaws of CFL, compared with CE and MFB, it still achieves smaller ECEs and obtains the best results. This makes CFL qualified to be a new general loss function in the semantic labeling of HRRSIs.

Here is a very interesting discovery. FL achieved the lowest ECEs, but its model uncertainty is the worst, while CFL can reduce model uncertainty, but it will increase ECE. It seems difficult to achieve outstanding results in all aspects if simply improving the loss function. Therefore, in future work, we will further study the semantic labeling methods of HRRSIs with excellent performance, model uncertainty, and low ECE.

### G. Comparison With Other State-of-The-Art Methods

In this section, we use our previously proposed HCANet to verify the computational efficiency and effectiveness of CFL. Table VII records the performance and training time of HCANet guided by CE loss and CFL on the ISPRS Vaihingen and Potsdam test sets. As can be seen from the table, compared
Fig. 12. Monte Carlo dropout uncertainty maps of each class of DeepLabV3+ on the Potsdam dataset. (a) Errormap. (b) Imp. surf. (c) Build. (d) Low veg. (e) Tree. (f) Car. (g) Clutter.

### TABLE VII

**Comparison of CE and CFL Performance and Training Efficiency on HCANet**

| Dataset   | Imp. surf. | Build | Low veg | Tree | Car | Clutter | mean $F_1$ | OA  | Training Time |
|-----------|------------|-------|---------|------|-----|---------|-------------|-----|---------------|
| Vaihingen |            |       |         |      |     |         |             |     |               |
| CE        | 92.51      | 95.15 | 83.82   | 89.07| 84.41| -       | 88.99       | 90.25| 2h 6min       |
| CFL       | 92.56      | 95.56 | 84.17   | 89.19| 85.42| -       | 89.38       | 90.46| 2h 9min       |
| Potsdam   |            |       |         |      |     |         |             |     |               |
| CE        | 93.14      | 96.45 | 86.90   | 88.25| 95.58| 60.46   | 86.80       | 90.69| 18h 28min     |
| CFL       | 93.19      | 96.67 | 87.15   | 88.22| 95.98| 60.43   | 86.94       | 90.82| 18h 51min     |

The numbers represent $F_1$ scores. The bold values in tables represent the best results.

### TABLE VIII

**Ablation Study of the TTA Strategy**

| Dataset   | SW | Flp | MS | Imp. surf. | Build | Low veg | Tree | Car | Clutter | mean $F_1$ | OA  |               |
|-----------|----|-----|----|------------|-------|---------|------|-----|---------|-------------|-----|---------------|
| Vaihingen |    |     |    |            |       |         |      |     |         |             |     |               |
| HCANet    | √  |     |    | 93.36      | 95.56 | 84.17   | 89.19| 85.42| -       | 89.38       | 90.46|               |
| HCANet    | √  |     | √  | 93.20      | 95.91 | 85.22   | 90.16| 87.63| -       | 90.42       | 91.27|               |
| HCANet    | √  | √   |    | 93.31      | 96.09 | 85.84   | 90.52| 87.80| -       | 90.71       | 91.55|               |
| Potsdam   |    |     |    |            |       |         |      |     |         |             |     |               |
| HCANet    | √  |     |    | 93.19      | 96.67 | 87.15   | 88.22| 95.98| -       | 86.94       | 90.82|               |
| HCANet    | √  |     | √  | 93.40      | 96.95 | 87.66   | 88.35| 96.30| -       | 92.59       | 91.20|               |
| HCANet    | √  | √   |    | 93.77      | 97.07 | 88.25   | 89.07| 96.39| -       | 92.95       | 91.58|               |
| HCANet    | √  | √   | √  | 93.33      | 97.18 | 88.44   | 89.04| 96.69| -       | 93.04       | 91.68|               |

In order to compare with other state-of-the-art methods, our metrics only calculate the five categories in the dataset except the clutter. The numbers represent $F_1$ scores. The bold values in tables represent the best results.
TABLE IX

| Dataset       | UZ1_1[3] | USTG_4[67] | SRA-FCN [1] | V-PureNet [37] | TSMTA [72] | AMAM [69] | PSNet [20] | BKG [111] | CASIA2 [69] | S-Net [70] | BSC-Net [71] | HCANet-CFL |
|---------------|----------|------------|-------------|----------------|------------|-----------|-----------|-----------|-------------|------------|----------------|-------------|
| Imp. surf.     | 89.20    | 93.50      | 81.60       | 86.90          | 57.30      | 81.50     | 87.30     | 92.40     | 89.20       | 83.25      | 90.20          | 90.71       |
| Build         | 89.40    | 94.80      | 77.50       | 86.50          | 72.60      | 84.18     | 87.50     | 92.40     | 89.20       | 83.25      | 90.20          | 90.71       |
| Low veg       | 91.40    | 94.90      | 80.63       | 88.37          | 87.05      | 88.54     | 92.23     | 92.04     | 89.20       | 83.25      | 90.20          | 90.71       |
| Tree          | 91.10    | 94.30      | 82.90       | 88.80          | 81.30      | 87.72     | 92.40     | 92.04     | 89.20       | 83.25      | 90.20          | 90.71       |
| Car           | 92.00    | 94.40      | 84.50       | 89.90          | 86.30      | 89.42     | 90.00     | 92.40     | 89.20       | 83.25      | 90.20          | 90.71       |
| mean F1       | 91.63    | 95.02      | 83.25       | 88.87          | 87.16      | 89.19     | 90.44     | 92.40     | 89.20       | 83.25      | 90.20          | 90.71       |
| OA            | 93.29    | 95.53      | 85.06       | 90.34          | 88.70      | 90.55     | 91.11     | 92.40     | 89.20       | 83.25      | 90.20          | 90.71       |

The numbers represent F1 scores. The bold values in tables represent the best results.

with CE loss, CFL can make HCANet achieve better results, while increasing the training time very little. Furthermore, following [74], we employ a test–time augmentation (TTA) strategy to validate the model’s performance on very large-scale high-resolution RS images. The TTA strategy includes sliding-window testing with overlap (SW), random flipping (Flip), and multiscale testing (MS). Finally, we compare the final results with other state-of-the-art methods on the ISPRS Vaihingen and Potsdam datasets.

Table VIII tabulates the results of the ablation study of the TTA strategy. It is obvious that each strategy can increase the performance of the HCANet. Finally, we compare the final results with other state-of-the-art methods, and the numerical results are given in Table IX. It can be seen that our HCANet achieves state-of-the-art performance guided by CFL.

VI. CONCLUSION

This article proposes a CFL for semantic labeling of HRRSIs. First of all, this article analyzes the deficiencies of FL in semantic labeling, that is, it will introduce confusing-classified examples while suppressing the loss of well-classified examples. However, the insufficient classification ability of FL for these samples leads to poor semantic labeling performance. In order to measure the difficulty of classification, this article proposes a PCM based on the prediction results of the network. Based on PCM, a calibration item is added to FL to construct CFL, forcing the network to pay more attention to those confusing-classified examples. A large number of experiments conducted on two challenging benchmark datasets show that the proposed CFL can make up for the deficiencies of FL in semantic labeling of HRRSIs, and achieve outstanding results on different datasets, exceeding the general CE loss. Furthermore, this article also uses the Monte Carlo dropout method to analyze the model uncertainty of different loss functions. Experimental results prove that CFL can not only improve the performance of the network but also reduce the model uncertainty. In addition, the ECEs of the prediction results of the networks guided by different loss functions are further calculated to measure their confidence calibration. Although the ECEs of CFL are slightly worse than FL, they are still better than that of CE. Finally, combined with our previously proposed HCANet, we investigate the computational efficiency of CFL and its effectiveness on HCANet. Experimental results demonstrate that HCANet guided by CFL can achieve state-of-the-art results on the ISPRS Vaihingen and Potsdam test sets. The research in this article shows that CFL qualifies as a promising general loss function for HRRSI semantic labeling.

REFERENCES

[1] L. Mou, Y. Hua, and X. X. Zhu, “Relation matters: Relational context-aware fully convolutional network for semantic segmentation of high-resolution aerial images,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 11, pp. 7557–7569, Nov. 2020.
[2] W. Chen, Z. Jiang, Z. Wang, K. Cui, and X. Qian, “Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 8916–8925.
[3] M. Volpi and D. Tuia, “Dense semantic labeling of subdecimeter resolution images with convolutional neural networks,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 2, pp. 881–893, Feb. 2017.
[4] N. Bengana and J. Heinikila, “Improving land cover segmentation across satellites using domain adaptation,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 14, pp. 1399–1410, 2020.
[35] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, “Classification with an edge: Improving semantic image segmentation of high-resolution remotely sensed images,” ISPRS J. Photogrammetry Remote Sens., vol. 155, pp. 158–172, 2018.

[36] N. Audebert, B. Le Saux, and S. Lefèvre, “Joint learning from earth observation and openstreetmap data to get faster better semantic maps,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2017, pp. 1552–1560.

[37] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1050–1059.

[38] M. P. Naeini, G. Cooper, and M. Hauskrecht, “Obtaining well calibrated probabilities using Bayesian binning,” in Proc. 29th AAAI Conf. Artif. Intell., 2015, pp. 2901–2907.

[39] H. Bai, J. Cheng, X. Huang, S. Liu, and C. Deng, “HCA: A hierarchical context aggregation network for semantic segmentation of high-resolution remote sensing images,” IEEE Geosci. Remote Sens. Lett., vol. 19, pp. 1–5, 2021.

[40] P. Marmans, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, “Classification with an edge: Improving semantic image segmentation with boundary detection,” ISPRS J. Photogrammetry Remote Sens., vol. 135, pp. 158–172, 2018.

[41] X. Zheng, X. Wu, L. Huan, W. He, and H. Zhang, “A gather-to-guide network for remote sensing semantic segmentation of RGB and auxiliary image,” IEEE Trans. Geosci. Remote Sens., vol. 60, pp. 1–15, 2022.

[42] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, “Learning aerial image segmentation from online maps,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 11, pp. 6054–6068, Nov. 2017.

[43] M. Havaei et al., “Brain tumor segmentation with deep neural networks,” Medical Image Analysis, vol. 34, pp. 62–75, 2017.

[44] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, “Semantic image segmentation of deep convolutional networks and multimodal architectures for earthquake damage detection based on satellite imagery and digital urban data,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 14, pp. 8606–8613, 2021.

[45] S. Bernabé, C. González, A. Fernández, and U. Bhangale, “Portability and acceleration of deep learning inferences to detect rapid earthquake damage from vhr remote sensing images using intel openvino toolkit,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 13, pp. 3870–3879, 2020.

[46] T. Miyanoto and Y. Yamamoto, “Using 3-D convolution and multimodal representations by back-propagating errors,” ISPRS J. Photogrammetry Remote Sens., vol. 135, pp. 158–172, 2018.

[47] C. Liang-Chieh, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, “Fully convolutional networks,” in CVPR, 2015, pp. 2480–2488.

[48] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in Proc. Int. Conf. Mach. Learn., 2016, pp. 1050–1059.

[49] M. P. Naeini, G. Cooper, and M. Hauskrecht, “Obtaining well calibrated probabilities using Bayesian binning,” in Proc. 29th AAAI Conf. Artif. Intell., 2015, pp. 2901–2907.
[50] W. J. Maddox, P. Izmilov, T. Garipov, D. P. Vetrov, and A. G. Wilson, “A simple baseline for Bayesian uncertainty in deep learning,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 13153–13164.

[51] N. Tagasova and D. Lopez-Paz, “Single-model uncertainties for deep learning,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 6417–6428.

[52] A. Loquercio, M. Segu, and D. Scaramuzza, “A general framework for uncertainty estimation in deep learning,” IEEE Robot. Automat. Lett., vol. 5, no. 2, pp. 3153–3160, Apr. 2020.

[53] S. van den Berg, S. Joos, P. W. Verbeek, and A. van den Hengel, “Data-efficient uncertainty estimation,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 5, pp. 2358–2369, May 2020.

[54] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. Int. Conf. Mach. Learn., 2017, pp. 1312–1320.

[55] D. Marcos, M. Völpi, B. Kellenberger, and D. Tuia, “Land cover mapping of VHR remote sensing images using a two-stage multiscale training architecture,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 8, pp. 5367–5376, Aug. 2020.

[56] Y. Sun, X. Zhang, Q. Xin, and J. Huang, “Developing a multi-filter convolutional neural network for semantic segmentation using high-resolution aerial imagery and lidar data,” ISPRS J. Photogrammetry Remote Sens., vol. 143, pp. 3–14, 2018.