IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 10, 2020, accepted July 21, 2020, date of publication July 29, 2020, date of current version September 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012701

Mask-R-FCN: A Deep Fusion Network for Semantic Segmentation

YUNFENG ZHANG AND MINGMIN CHI, (Member, IEEE)
Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai 200438, China

Corresponding author: Mingmin Chi (mmchi@fudan.edu.cn)

This work was supported in part by the Shanghai Fabric Eyes AI Technology Company Ltd., through Fabric Eyes Science Funds, and in part by the National Key Research and Development Program under Contract 2017YFA0402600.

ABSTRACT Remote sensing image classification plays a significant role in urban applications, precision agriculture, water resource management. The task of classification in the field of remote sensing is to map raw images to semantic maps. Typically, fully convolutional network (FCN) is one of the most effective deep neural networks for semantic segmentation. However, small objects in remote sensing images can be easily overlooked and misclassified as the majority label, which is often the background of the image. Although many works have attempted to deal with this problem, making a trade-off between background semantics and edge details is still a problem. This is mainly because they are based on a single neural network model. To deal with this problem, a convolutional deep network with regions (R-CNN), which is highly effective for object detection is leveraged as a complementary component in our work. A learning-based and decision-level strategy is applied to fuse both semantic maps from a semantic model and an object detection model. The proposed network is referred to as Mask-R-FCN. Experimental results on real remote sensing images from the Zurich dataset, Gaofen Image Dataset (GID), and DataFountain2017 show that the proposed network can obtain higher accuracy than single deep neural networks and other machine learning algorithms. The proposed network achieved better average accuracies, which are approximately 2% higher than those of any other single deep neural networks on the Zurich, GID, and DataFoundation2017 datasets.

INDEX TERMS Deep fusion, deep semantic segmentation, fully convolutional network, object detection, remote sensing.

I. INTRODUCTION

With the fast development of sensor technologies, it is easier to obtain remote sensing images, which play an increasingly important role in urban applications, such as rapid urban mapping [1], hazard and health monitoring [2], and earth observation [3], [4]. To achieve these tasks, it is necessary to design an effective model with high accuracy and robustness for the classification of remote sensing images. Owing to the fastest-growing deep learning techniques [5], it is possible to build automatic classifiers on big data processing platforms, such as Apache Spark [6], along with powerful computing capacity, which is distributed and in parallel for the analysis of remote sensing images [7], [8]. Recent studies indicate that convolution neural networks (CNNs) can effectively learn feature representations and help with several tasks, such as large-scale image recognition [9]–[13], object detection [14]–[17], and semantic segmentation [18]–[21]. Recent advances have proved that CNN is an effective method for extracting mid/high-level features from remote sensing images [22].

The semantic segmentation tasks of remote sensing are usually based on hyperspectral [23]–[26] and multispectral [27]–[29] remote sensing images. Hyperspectral images generally have tens or hundreds of bits of spectral information per pixel. Shallow supervised algorithms based on single pixels in remote sensing such as SVM [20], decision tree [30], and random forest [31] work well on these datasets. Unlike hyperspectral images, multispectral images usually have higher resolution but only contain certain channels such as RGB-NIR (near infrared), SWIR (short wave infrared), and TIR (thermal infrared); they may seem similar to natural images. Shallow supervised algorithms based on a single pixel do not consider contextual information around the

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

given pixel. They can classify hyperspectral RS images well but find it difficult to classify multispectral remote sensing images via a single pixel. Many works have tried to design using contextual information and have resorted to shallow supervised algorithms in order to classify single pixels [32], [33]. They have been proven to classify multispectral images efficiently but certain problems remain, such as being too dependent on expert knowledge or providing extracted features inferior to those available using deep learning.

In recent years, CNNs have been widely exploited for segmentation of remote sensing data [34], [35]. A former popular segmentation method based on deep learning is called patch classification [36], [37], in which features are usually extracted from the surrounding pixels to classify the central pixel. Usually the deep neural networks are first used to extract features and then the shallow supervised methods are applied to classify the central pixel. However, such methods have the problem of high computation effort. The main cause for this is the existence of the fully connected layers. In 2014, the fully convolutional network (FCN) [38] was proposed which allows for the elimination of the fully connected layers at the end of network. This structure has been widely applied in remote sensing [39], [40]. The subsequent semantic segmentation method adopted this structure, which can also be called end-to-end classification. There are two different architectures of this end-to-end classification method. The first is the encoder-decoder architecture [41], [42]. The encoding process gradually reduces the location information and extracts the abstract features through the pooling layer, and the decoding process gradually restores the location information by upsampling or deconvolution. Usually, there is a direct link between the encoder and decoder [43]. The second architecture is the dilated convolution, which does not consider the pooling layer. Typically, in [44], the dilated convolution uses expanded convolution to aggregate multi-scale information, called ''context module''. However, dilated convolution architecture requires a large memory and has a heavy computational burden.

Although deep learning methods have been successfully applied in the segmentation of remote sensing images, certain new problems are still associated with it. A significant problem is that small objects usually cannot be well classified by these methods. Few works have tried to improve the segmentation of small objects, such as SegNet [41], DeepUNet [45], and DeepLab [46]; however, the effect of the improvement is limited. Although these algorithms can solve the problem of small object segmentation to some extent, they still suffer the problem of making a trade-off between background semantics and edge details [37], [47]. This is mainly because they are based on a single neural network model. Deep fusion has become one of the fastest growing research directions in the field of remote sensing. Generally, there are two forms of the deep learning-based fusion method: the decision-level fusion and the feature-level fusion. The simplest way to achieve feature-level fusion using deep learning is to stack multiple channels from different imaging methods on a single data cube, followed by the application of the regular training strategy [36]. Another way to fuse features of spatial information in deep neural networks is to extract different types of spatial filters and concentrate them as the joint features to make the final prediction [48], [49]. However, these methods suffer from lack of additional data and inability to utilize large-scale pretrained model. Compared to feature-level fusion methods, decision-level fusion methods use an explicit way to learn the final fusion layer using the land maps generated by several single networks. From the perspective of the fusion type, the decision-level fusion can be divided into two types: post-processing and learning-based. In the first case, different networks infer different land maps; the outputs are fused in the subsequent step for post-processing [50]. In the second case, before learning the land maps from different networks, a few randomly initiated layers or pretrained layers are connected to the backend of all outputs to learn an optimal result. Experiments in [51] showed that learning-based methods can obtain better fusion results than the direct-fusion methods. Our method is based on decision-level and learning-based fusion. We use object detection model as a complementary descriptor since it is a powerful tool to retrieve these small objects that were lost. Object detection in remote sensing usually refers to the identification of bounding boxes of interest (such as tanks, airplanes, and harbors) in a satellite image and their classification by the correct labels [52], [53]. The current object detection methods can be divided into two categories: the two-stage and one-stage methods. In the two-stage method, extraction is first applied to generate region proposals in the entire image, and then the regions are cropped to extract features for prediction. The region-based CNN (R-CNN) [54] is the first of its kind that proves that CNNs can greatly improve the performance of object detection using PASCAL [55]. The subsequently proposed R-CNNs such as fast-RCNN [56] and faster-RCNN [57] continue to utilize the same structure as R-CNN. Unlike the two-stage approach, the YOLO [58]–[60] series methods model object detection as a regression problem with an end-to-end training network. During prediction, only one inference is required to obtain the bounding boxes and category labels. Because there is only one inference, one-stage methods run faster than two-stage methods but have lower detection accuracy.

Considering the advantages of both segmentation and object detection, we propose here a deep neural network fused architecture to better classify the remote sensing images by combining the pixel-based FCN and object-based Mask-RCNN [61]. Mask-RCNN is able to automatically detect objects in the images and assign a semantic label to each object (a predefined region). In addition, for each object, an instance segmentation strategy is applied in Mask-RCNN to make a pixel-level prediction. Here, the proposed network is referred to as Mask-R-FCN. The classification results are fused in the decision level by two deep neural networks (DNNs) for the proposed Mask-R-FCN. Experimental results on real remote sensing images from Zurich [20], GID [21], and DataFountain2017 show that the proposed network can

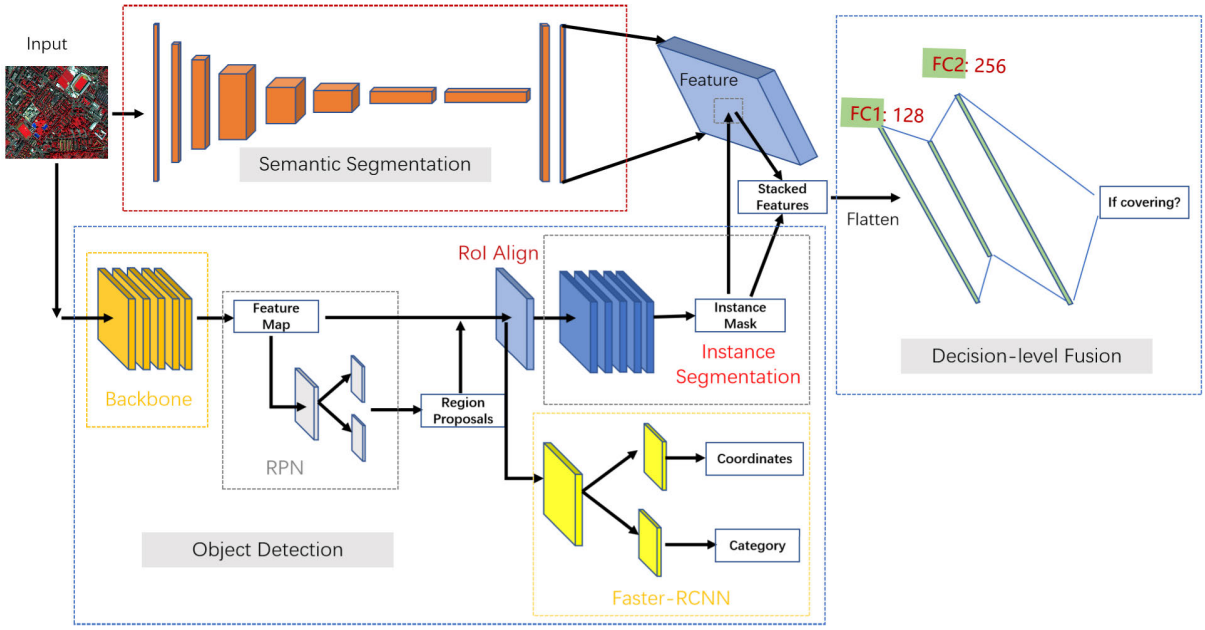

FIGURE 1. Architecture of the proposed Mask-R-FCN network.

obtain higher accuracy than a single DNN, such as FCN, as well as many machine learning algorithms.

This article is organized as follows. Section II describes the proposed Mask-R-FCN for the classification of remote sensing images by combining the pixel-based FCN and object-based Mask-RCNN. Section III presents the experimental results for the comparison of different methods. Discussion and conclusion are given in Section IV.

II. METHODOLOGY

To better classify small objects in urban remote sensing images, a deep fused network is designed by combining the pixel-based FCN [38] and the object-based Mask-RCNN [61]. The classification results are fused at the decision level. The architecture of the proposed model is shown in Fig. 1. In the following sections, the architectures of FCN and Mask-RCNN are firstly described, following which the proposed Mask-R-FCN network is introduced.

A. SEMANTIC SEGMENTATION

FCN is an end-to-end classifier and the architecture can be adapted by fine-tuning to other deep networks, such as VGGNet [11]. The general architecture of an FCN is depicted in Fig. 2. For convenience, we reduce the number of convolutions and skip connections. Each layer of data in a convolutional layer is a three-dimensional array of size $h \times w \times d$, where h and w are the spatial dimensions, and d is the feature or channel dimension. The input is the image, with pixel size $h \times w$, and d spectral channels, where d is usually equal to 3 for RGB remote sensing images or 4 for NIR-RGB remote sensing images.

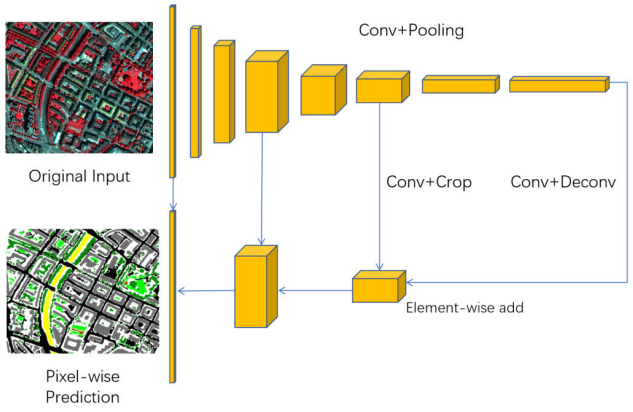

FIGURE 2. A typical architecture of FCN.

FCN updates network parameters according to a softmax loss. The loss function is the sum of the pixel-level softmax loss on the spatial map of the last layer. The loss function is described as

$$l(x;\ \Theta) = \sum_{ij} l'(x_{ij},\ \Theta_{ij}), \tag{1}$$

where x denotes the predicted map while θ denotes the real semantic map. l' is the softmax loss for each pixel.

Usually, large amounts of raw images cannot be trained simultaneously due to memory limitations and the time consumption of the process. In our experimental dataset, the average size of the images is 1100×1150 pixels, so a considerable amount of memory and time-cost are required to train even a small batch using FCN. In addition, if an entire image is taken as a sample during training, the number of samples

can be too small (less than 20), leading to an overfitting problem. Accordingly, raw images are usually cropped to smaller fixed-size patches for the training data by using sliding windows. Overlapping is allowed to increase the number of training data and enhance the robustness of the model.

FCN can provide a pixel-level prediction. Its output layer has the same size as the input, i.e., $h \times w$. FCN first downsamples the input to a smaller size and then upsamples the activations to the same size as that of the inputs, as shown in Fig. 2. Deconvolution is usually used to implement the upsampling operation, which directly conducts a transposed convolution on the feature map. But in FCN, a variant of the deconvolution is used. This variant first performs an uppooling operation on the feature map and then conducts a convolution.

In addition, skip connections are introduced into the network, which allows the process of decoding to fuse abstract semantic information from different depths of encoding layers and preserve the location information, which is usually lost in the step-by-step process of the pooling operations.

B. OBJECT DETECTION

R-CNN is designed for object detection (detecting localization of the object and distributing a label to each region proposal) within an image. Because our experimental datasets are meant for the task of semantic segmentation, and each image of this dataset is annotated by pixel-level labels, preprocessing is required to construct a new dataset to cater the task of object detection. We follow the common architecture of Mask-RCNN. First, a global feature map is generated by several convolutional layers and subsequently, the map is fed into three different branches. The first branch inherits the region proposal network (RPN). The RPN generates candidate region proposals and refines the bounding boxes for the first time. Next, an RoI align operation is applied on the global feature map according to the predicted coordinates from the RPN. The aligned features finally go through two parallel branches, one for predicting pixel-level classification on each bounding box and the other for predicting categories and confidences for each candidate. The former adds a ''head'' module behind RoI align, which scales up the output of RoI align to enhance the accuracy of the mask prediction.

In addition, unlike the traditional FCN's softmax loss for each category, the mask branch uses average binary cross-entropy loss for the k-th predicted Mask. Here, k denotes the number of categories. In general, Mask-RCNN uses a multi-task loss. The loss consists of three parts: classification loss, boundary loss, and mask loss:

$$\begin{aligned} L(\{p_i, t_i, m_i\}) = & \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\ & + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \\ & + \frac{1}{N_{mask}} \sum_i L_{mask}(m_i), \end{aligned} \tag{2}$$

where N_{cls} denotes the number of categories, L_{cls} represents the category loss function. The category loss function is described as

$$L_{cls}(p_i, p_i^*) = -log[p_i p_i^* + (1 - p_i^*)(1 - p_i)], \tag{3}$$

where p_i denotes the confidence with which the target is recognized as an object and p_i^* is a binary function, the output of which is 1 if the target is a real object, otherwise it is 0. In other words, only when the object in the i-th bound is a positive example will it contribute to the category loss.

The bounding-box regression loss function is represented as

$$\begin{aligned} L_{reg}(t_i, t_i^*) &= smooth_{L1} \\ &= \left\{ \begin{array}{l} 0.5x^2, \ if \ |x| < 1 \\ |x| - 0.5, \ otherwise \end{array} \right\}. \end{aligned} \tag{4}$$

In addition to the above two types of loss functions, the mask branch in Mask-RCNN uses an average binary cross-entropy loss function, which is described as

$$L_{mask} = -[m_i \times log(m_i^*) + (1 - m) log(1 - m_i^*)], \tag{5}$$

where m_i represents the confidence that the object is predicted as the target, and m_i^* represents the i-th ground truth mask. Using the binary cross-entropy loss function avoids competition between classes. The classification task is only assigned to the classification loss function, and the mask layer only distinguishes specific small classes in the mask.

C. INTEGRATING MODULE

A strategy for integration is applied to both the pixel-level and object-level semantic results. This is called ''decision-level fusion''. Our approach is primarily based on learning-based fusion.

The definition of the training samples in the fusion module can be seen in Fig. 3. For each bounding box predicted from Mask-RCNN, the relevant features will be acquired in the penultimate layer of the mask branch, which is a feature map with the size of $14 \times 14 \times 256$. Then the corresponding features are obtained from the penultimate layer of FCN which are cropped by the coordinates of the bounding box. Because the sizes of both feature maps are mismatched, an ROI pooling step will be used for both to generate feature blocks of the same size. Then, both feature maps can be stacked into a single cube along the axis of the channel. After the stacking operation, we flatten the cube into a single vector and fed them into two fully connected layers.

The network used in fusion module is a fully connected network. Because there training samples are few, a simple two-layer fully connected network with two hidden layers is used. The number of unions of both hidden layers are 128 and 256. The output layer is a binary value activated by the sigmoid function for determining if the corresponding regions should cover the output maps of FCN. The fusion module updates network parameters according to binary cross-entropy loss.

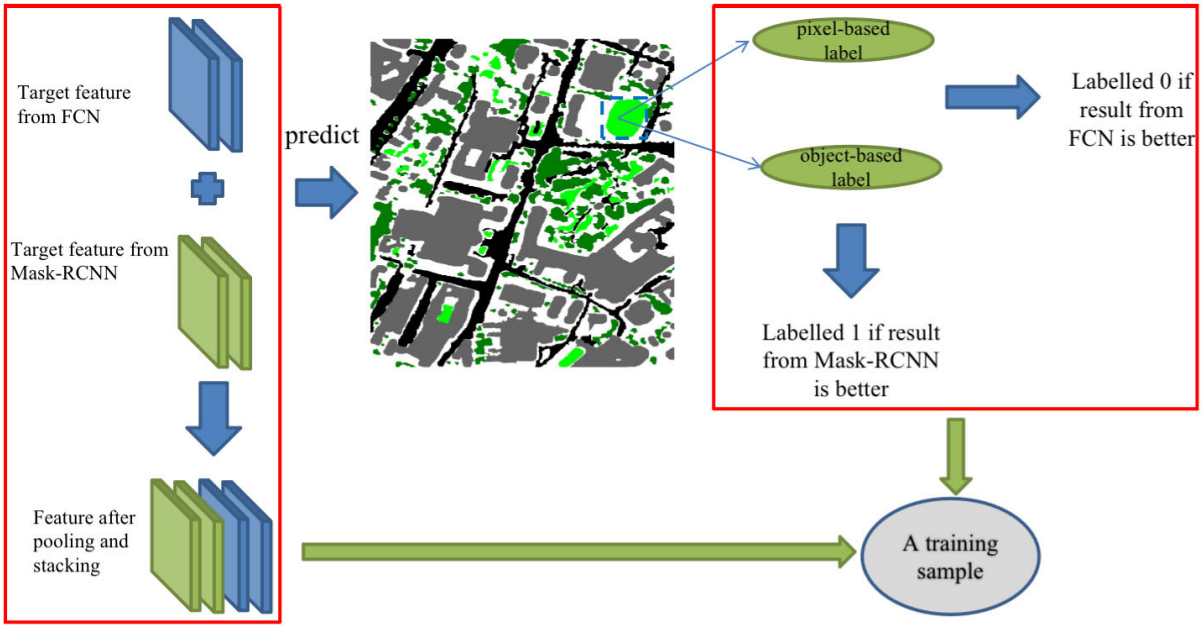

FIGURE 3. The definition of training samples in fusion module. The fusion feature is generated by features from both the semantic and object detection modules. The label is defined by comparison of both modules.

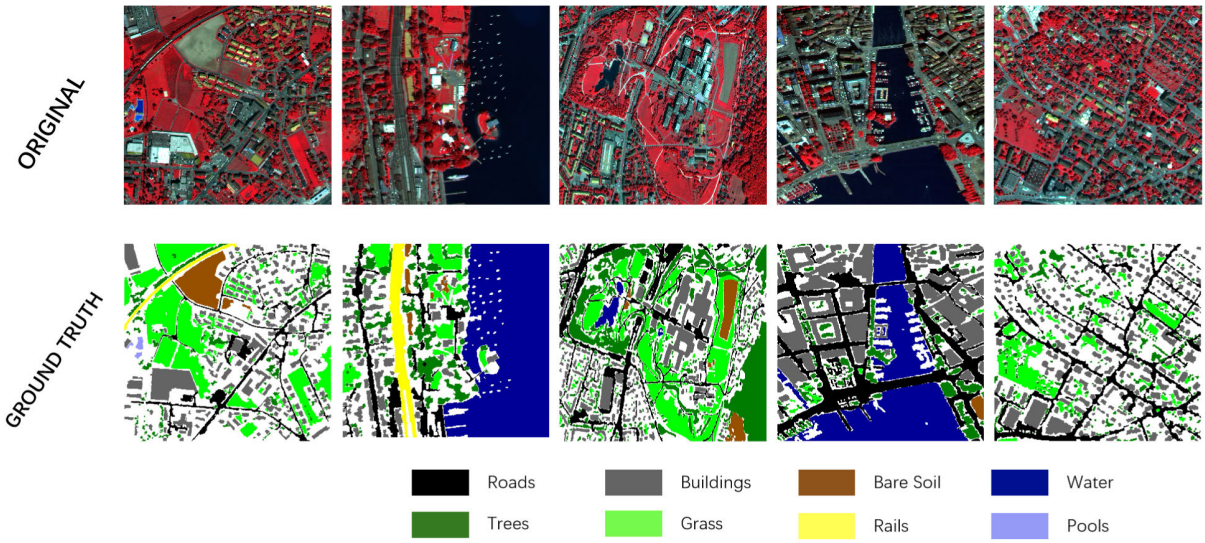

FIGURE 4. Image patch examples from Zurich dataset. The above row shows the original RGB-NIR remote sensing images, and the corresponding row below shows the ground truth of the semantic label. The class legend is given at the bottom.

III. EXPERIMENTS

To validate the proposed method, we conduct experiments on three different remote sensing datasets. These experiments share the same setting.

A. DATA DESCRIPTION

To verify the semantic segmentation performance of the proposed Mask-R-FCN model, we choose the Zurich dataset [20] that provides land-cover maps of very high-resolution (VHR) satellite images. There are 20 image chips from a single large QuickBird image acquired in 2002 over Zurich, Switzerland, and each image chip is pansharpened to 0.6 m resolution. This dataset contains 8 land-use categories covering roads, buildings, trees, grass, bare soil, water, railways, and swimming pools. The average size of each image is 1000 × 1150 pixels and each image has 4 channels of RGB and NIR. An example along with the dataset is shown in Fig. 4.

B. DATA PREPROCESSING

1) PREPARING DATASETS FOR SEMANTIC SEGMENTATION

We first convert the original RGB-like labels to one-hot coding for each image in the Zurich dataset. We use the

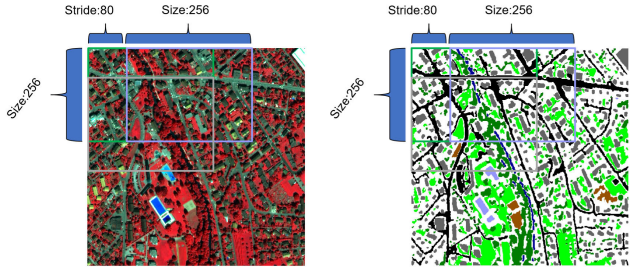

FIGURE 5. A demonstration for sliding windows.

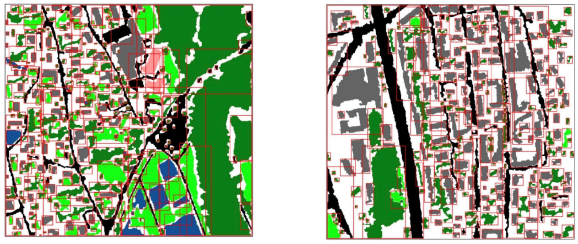

FIGURE 6. Examples generated by the algorithm. In the algorithm, the horizontal bounding boxes are used while the aim of this task is to accurately localize the instance in terms of a horizontal bounding box in the (xmin, ymin, xmax, ymax) format. The ground truths for object detection task are generated by calculating the axis-aligned bounding boxes. Some examples of horizontal bounding boxes generated by the algorithm can be seen in the two semantic maps.

data augmentation strategy on the original dataset. In general, when training FCN, considering the memory and training speeds, the common size of the input image is 256 × 256. However, the average size of our image is 1000 × 1150 for Zurich dataset, 7200 × 6800 for GID, and 4000 × 4000 for DataFountain2017. Therefore we use the sliding window approach to segment the original image. To further increase the number of data samples and improve robustness of the model, we narrow the stride while using the sliding window for segmentation, which causes window overlap in the samples. The aforementioned strategy is illustrated in Fig. 5. Among the segmented images, we randomly choose the ones from a single chip for testing and the rest of the images are considered as training data. In order to take advantage of the existing powerful network trained on large-scale data, we only use the RGB channels of the original images for segmentation. The reason is that the natural images usually have only three channels. Another method is to use RGB-NIR images to train the network from scratch; however, this method brings a drop in the classification accuracy. The possible reason is that the network often has millions of parameters, and the small-scale remote sensing dataset cannot train effectively; this usually leads to an overfitting problem.

2) PREPARING DATASETS FOR OBJECT DETECTION

The dataset used in our study is annotated at pixel level. However, because of the object detection task, the dataset should be annotated at the object level. To meet this requirement of using a Mask-RCNN, each of the object regions should be annotated at the pixel-level. Therefore, the algorithm proposed in our article uses a traversal method to generate all the bounding boxes from the semantic map. Because the bounding boxes are derived from the pixel-level map, in addition to being assigned object-level labels, they are naturally annotated at the pixel-level. Some examples derived by our algorithm are illustrated in Fig. 6. In the process of preparing small object dataset, a threshold value of δ is given. If the number of valid pixels within the labeled object (the valid pixels include pixels which are non-background) exceeds δ, this object will be removed from the candidate training dataset; otherwise, the object will be retained. After the above operations for each candidate box generated by the algorithm, the final training dataset includes only a list of candidate boxes containing small objects. By doing so, the input size of training samples is limited, so that the model can focus more on the detection of small objects; we set δ to 4000.

C. EXPERIMENTAL DETAILS

1) BASELINE FCN MODEL

We build our semantic segmentation model on an FCN framework that first downsamples the input, upsamples the encoding output, and subsequently classifies the output. The input images are processed to a fixed size of 256 × 256. We refer to FCN-8s, which uses the VGG network as the base network for extracting deep features, when training FCN. FCN-8s was proved most effective in the original article among FCN-8s, FCN-16s, and FCN-32s. We use the pretrained models for which the input has three channels, even though the Zurich images have four NIR-RGB channels. This is because the general pretrained model is always trained on natural images, which have three RGB channels. An attempt is also made to train a model from scratch by using RGB-NIR images. The network is trained end-to-end for 100 epochs on the training splits from the Zurich dataset. The input size of all the deep learning algorithms is 256 × 256. The models are trained by the stochastic gradient descent (SGD) with a mini-batch size of 16 and a fixed learning rate of 0.0001. Each layer uses leaky rectified linear functions as their activation functions. We adopt a standard data augmentation scheme including random rotation and random scaling. We did not use a color transform because this augmentation is used for RGB images. RGB in Zurich data actually refers to the digital number (DN) value, which is the brightness value of a pixel in a remote sensing image. Using a color transform may break the relationship between categories and pixel bands.

2) OBJECT DETECTION MODEL

Here we use Mask-RCNN as our baseline object detection model. To be consistent with FCN, we use 256 × 256 as input size, which is helpful for the final fusion step. Performing preprocessing in a way similar to FCN, we use a sliding window of 256×256 and a stride of 80 to segment the object-annotated maps. For a bounding box that spans two patches, we simply

cut it into two parts along the boundary. This is done for most remote sensing images because object classification results of land use are less affected by the integrity of the target object, for example, water, grass, trees and so on. In our initial experiment, overfitting may easily take place when training the model from scratch with only thousands of training examples, for which the channel is of the NIR-RGB type. We removed the infrared channel and used only the RGB channels to train the model, in order to use the pretrained model verified on the existing PASCAL VOC [55] and MS COCO datasets [62]. We adopt the same data augmentation scheme in this article [63] which is intended for small object detection. In addition, we apply several strategies, including "strict in" and "strict out," so that Mask-RCNN focuses on small object detection.

"Strict in": the first strategy mentioned above limits the size of input data and makes the model focus on the detection of small objects. Moreover, we modify the anchor parameter of Mask-RCNN. The anchor parameter is used to set the size of predefined boxes, which adds a priori to the model. This priori assumes the rough size of the candidate box that may appear on each cell. The training difficulty can be reduced by a priori to speed up the convergence of the model. Another advantage of the predefined boxes is that the model can be more inclined to detect objects with the same size as that of the predefined boxes. To make the model be more inclined to detect small objects, we changed the original anchor parameter, which was (32, 64, 128, 256, 512), to (8, 16, 32, 64, 128).

"Strict out": the loss function of Mask-RCNN contains l_{mask} loss. Only when the intersection over union (IoU) between the generated candidate box and the real target box exceeds a certain threshold (the original threshold is 0.7), will the candidate box contribute to L_{mask} loss. Because the purpose of Mask-R-FCN is to improve the accuracy of semantic segmentation with small object pixels, we pay more attention to the accuracy and confidence of small object pixels rather than the recall of small objects. Here, we raise the threshold value of IoU to 0.8.

During prediction, we also apply some strategies to filter candidate boxes. Usually, Mask-RCNN produces a large amount of candidate boxes, and the same object may be crossed or covered by multiple candidate boxes. The purpose of Non Maximum Suppression (NMS) is to reserve an optimal candidate box for each target. There are two parameters: minimum confidence and NMS threshold. When the confidence of a candidate box is lower than the minimum confidence, it will be removed. The NMS threshold is a judgment threshold for NMS algorithm. If the rest of the candidate boxes have IoU over this threshold, the candidate boxes will be removed. The original minimum confidence and NMS threshold are set to 0.7 and 0.3, respectively. Here, because our goal is to obtain a candidate box with better confidence, we set the minimum confidence to 0.8 and the NMS threshold to 0.2. When the minimum confidence increases, the reserved candidate boxes have higher probabilities to contain objects. When the NMS threshold decreases, the candidate boxes with large overlap will be removed.

3) MODEL FUSION

Before training the fusion layers, the features and the semantic maps of all the split patches need to be generated through FCN and Mask-RCNN. The generated output is subsequently used to construct our training and testing samples for the fusion model. The procedure is as follows: First, for each bounding box generated by Mask-RCNN, on its semantic segmentation branch, the corresponding features extracted by the network are selected. For FCN, according to the coordinates of the bounding box of the former network, the relative coordinates of FCN's feature map are computed, and the corresponding feature area is selected. We then apply RoI pooling to both features, to pool them into the same fixed-size cubes. Then, we stack the two cubes and flatten them into a single vector, which is a training example. Additionally, each training example has a binary label value indicating whether the semantic map of the bounding box generated by Mask-RCNN should cover the corresponding area of FCN's semantic map. This binary value is given as follows: it is set to one if the overwriting can boost the accuracy. Otherwise, it is set to zero. An example can be seen in Fig. 7. The fusion model is simply composed of two fully connected layers, of which the parameters are randomly initialized. We use the Adam optimizer to train the network and the learning rate is set to 0.0001.

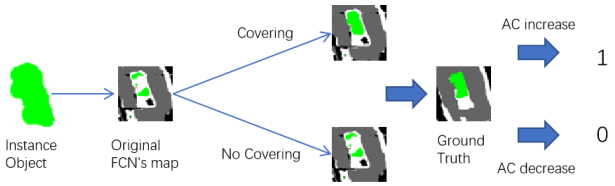

FIGURE 7. Definition of fusion data label. Here, AC stands for accuracy.

During prediction, we also apply an NMS algorithm to filter candidate boxes to cover the baseline semantic maps. The algorithm here is slightly different from the original one. There are also two parameters: minimum coverage confidence and pixel-level NMS threshold (pNMS threshold). When the output probability from the fusion network of a candidate box is lower than the minimum coverage confidence, it will not be used to cover the result of semantic segmentation. Instead of judging whether the boundaries of the remaining candidate boxes overlap with the current candidate box, we can directly judge whether the segmentation instances overlap. The pNMS threshold is set to 0, i.e., as long as there are duplicate pixels between the reserved candidate boxes and the current candidate box, these reserved candidate boxes will be removed.

D. EVALUATION PROTOCOL

During evaluation, we use different metrics to evaluate FCN, Mask-RCNN and fusion model. In the case of FCN,

we adopted the leave-one-out strategy, wherein we trained FCN on 19 images and tested on the held-out image. Each time we selected an image as a test set, we conducted 20 groups of experiments. For each experiment, we computed the accuracy by employing the following metrics: overall accuracy (OA), average accuracy (AA) and the F1 score (F1). The first metric describes the overall accuracy based on all the chips and is computed as the percentage of correctly assigned pixels over the total number of all pixels. The average accuracy represents the average metric of per-class accuracy. For each class, we add the number of correct predictions and compute the percentage of the total number of correct predictions made per-class. Next, we compute the average per-class accuracy. The third metric considers the precision and recall of the classification model. The F1 score represents the weighted average of the model precision and recall with a maximum of 1 and a minimum of 0. For model fusion during training, we use the common precision to evaluate the fusion of accuracy and testing. For model evaluation, we compute both the precision and the same metrics as for FCN. To evaluate the model fusion strategy, we compared the fusion results with some baseline networks experiments conducted in our study and some obtained from other studies. The comparative experiments were performed with the following models: the proposed Mask-R-FCN fusion model, FCN baseline model, learned RP, and CNN-KNN.

E. QUANTITATIVE RESULTS

1) STUDY ON ZURICH DATA

The proposed Mask-R-FCN network was validated by real remote sensing images from Zurich, Switzerland, obtained in 2002 [20]. The spatial resolution of the images was submeter, and there were 4 channel features, i.e., NIR-RGB. The semantic images contained eight categories, i.e., roads, buildings, trees, grass, bare soil, water, railways, and swimming pools.

We compared our results to segmented maps from raw images with several methods including CNN-KNN [64], learned RP [20], and FCN baseline. For the comparison with learned RP [20], the authors used data-specific regularities to learn a discriminative conditional random field (CRF) using structured support vector machines (SSVM); the CRF is leveraged to model the segmentation problem. However, the learned RP method continues to suffer from low classification accuracy and requires appreciable effort to set up feature extraction rules. The author of CNN-KNN [64] designed a custom CNN as the feature extractor. The input image is first cut into patches with a size of 64×64 and then fed into the CNN extractor. To this end, a supervised method is used as a classifier. A significant improvement can be seen in this experiment; however, since it is a two-stage method and needs to extract and classify the features of each patch, the efficiency of this method may be relatively reduced as it is time consuming. Moreover, since it is a custom network, it does not leverage the advantage of pretrained model trained on large-scale data. We also compared our results with those of a few other previous works based on the Zurich dataset [20].

In the experiments, the pretrained FCN in the context of the VGG16 was adopted, and training data were exploited to fine-tune all layers of FCN-VGG via back-propagation algorithm. We also conducted an experiment using RGB-NIR data to train FCN network from scratch. The experimental results are shown in Tab. 2. We denote FT-FCN as the fine-tuned FCN and FS-FCN as FCN trained from scratch. It can be seen from the results that the average accuracy of the network trained from scratch is worse than that of the fine-tuned network. The reason may be that a large-scale network often has millions of parameters, and using only a small-scale remote sensing image dataset to train such a large network often leads to the problems of overfitting and local minimum. Although there is a decrease in average accuracy, we can see that in some categories the accuracies have been improved, such as water and grass. The possible reason is that the near-infrared channel always has high reflectivity for vegetation, water, and soil. To leverage the advantages of fine-tuned network, we use the pretrained FCN as the fusion submodule.

TABLE 1. Average accuracies for different image sizes from Zurich dataset.

Image Size	Average Accuracy (%)
1491*1066	77.76
1068*1001	78.3
1364*1295	81.85
782*622	83.6
871*1146	85.7
782*622	87.37

We adopted the same leave-one-out strategy as in [20]. In each round of experiments, we used 19 pictures as the training dataset and the remaining 1 as the validation dataset. Part of the detailed results can be seen in Tab. 1. In each round of experiments, the experimental results vary in AA. Referring to the previous work, we average the AAs of all results in each round. The final average accuracy is 80.11%. The classification maps of some rounds are shown in Fig. 8.

In the object-based R-CNN, a fine-tuning ResNet is adopted as the CNN network and the fully connected output layer of 1000 cells is replaced by a 9-cell output layer for the remote sensing images used in the experiments. The overall accuracies are detailed in Tab. 2. The classification results are significantly improved by using deep neural networks compared to shallow machine learning algorithms and the proposed Mask-R-FCN obtains the best results compared to single deep networks.

To investigate the improvement of the after-fusion strategy in the classification of small objects, the AAs based on different categories are shown in Tab. 2. The accuracy of the proposed method is significantly improved in some categories with high probability of the existence of small objects, such as roads, soil, trees, pools, and especially buildings

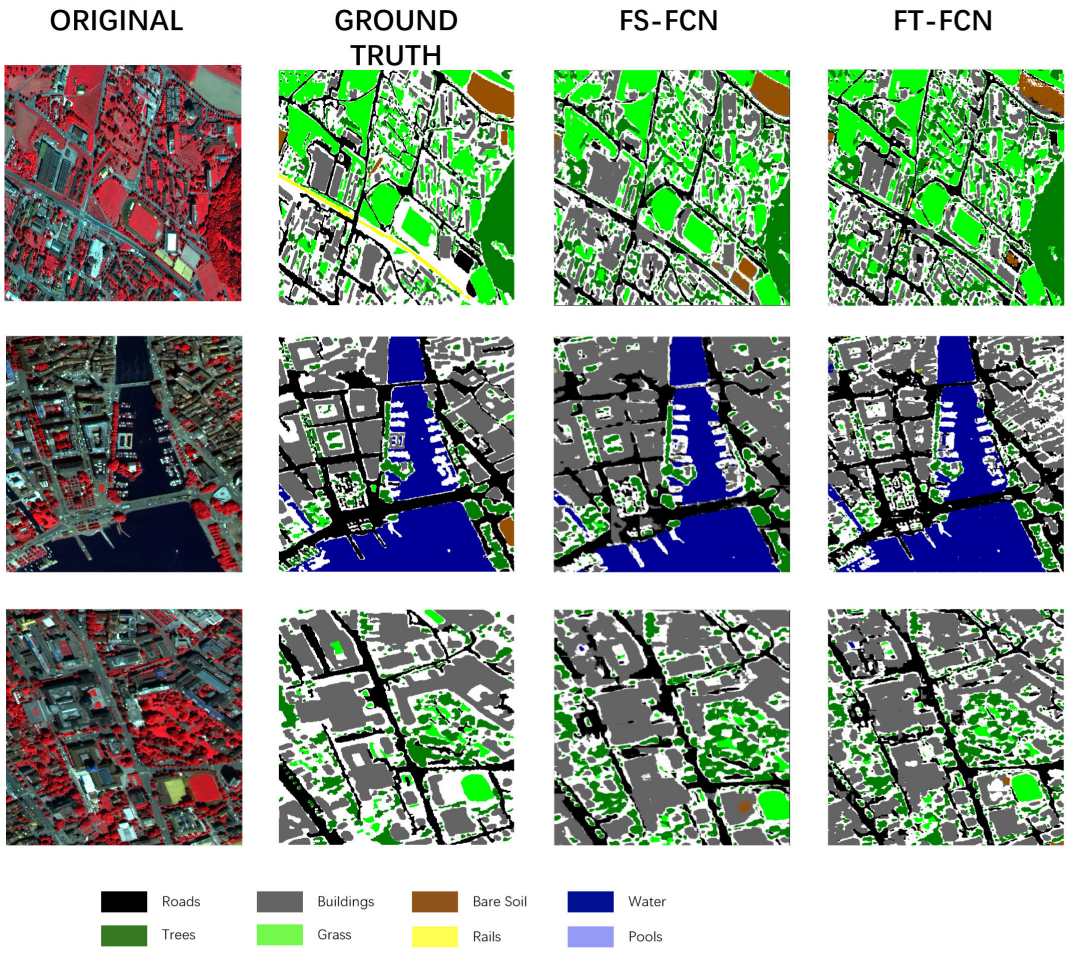

FIGURE 8. Classification maps generated by fine-tuned FCN and FCN trained from scratch.

TABLE 2. Classification accuracies for each class by using different methods from Zurich dataset.

Methods	AA (%)	Roads	Build.	Trees	Grass	Soil	Water	Rails	Pools
Mask-R-FCN	**82.28**	**89.12**	**92.74**	**98.52**	82.55	**79.17**	93.95	27.44	**94.74**
FT-FCN	80.11	88.53	87.34	96.12	83.68	78.87	97.42	17.80	91.17
FS-FCN	78.83	79.34	85.64	95.71	84.69	74.90	**98.53**	21.35	90.44
CNN-KNN	80.83	73.77	81.98	86.00	85.76	50.05	95.78	**41.62**	94.06
LEARNED RP	76.82	83.83	86.39	94.04	**86.58**	71.32	93.55	16.99	81.87
RF UNARY	72.19	81.80	87.04	94.14	84.38	64.25	91.08	2.11	72.72
MCSVM UNARY	74.55	80.77	84.88	92.68	84.79	69.99	93.54	9.73	80.03
CS POTTS	73.71	83.86	86.41	95.07	82.88	67.10	91.65	3.46	79.22
PASSIVE RP	73.72	83.47	86.68	95.10	82.49	67.58	91.62	3.32	79.50

and rails, which are increased by 5.4% and 9.64% respectively. However, the accuracy of grass and water decreases, especially for water, for which the accuracy decreases by 3.47%. This is probably because the proposed Mask-R-FCN method introduces some noise, which leads to false detection of small objects. In general, compared to FCN, the AA increases by 2.17% by using the fusion strategy, and also exceeds CNN-KNN and LEARNED RP 20M. The classification maps before and after fusion are shown in Fig. 10. Our Mask-R-FCN conducts the fusion strategy on the FT-FCN. One can see that the result of FCN model may easily overlook some small objects such as trees, buildings, and some small regions of grass. The after-fusion result, which can generally recall these small objects, exhibits better accuracy than the before-fusion result. However, there are also several errors in coverage that lead to decreased accuracy. The recall, precision, and F1 values for each class are given in Tab. 3. From the results, it can be seen that after fusion, recalls of most categories increased, but precisions increased and decreased half-by-half. F1 values increased slightly. The inference time

TABLE 3. Recalls, Precisions, and F1 scores for each class before and after fusion on Zurich data.

	Before Fusion			After Fusion		
Category	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)
Roads	88.59	88.53	88.56	89.17	89.12	89.14
Build.	91.43	87.34	89.34	93.28	92.74	93.01
Trees	95.10	96.12	95.61	96.74	98.52	97.62
Grass	95.18	83.68	89.06	93.94	82.55	87.88
Soil	87.53	78.87	82.98	86.51	79.17	82.68
Water	98.46	97.42	97.94	92.77	93.95	93.36
Rails	89.50	17.80	29.69	90.67	27.44	42.13
Pool	87.94	91.17	89.53	91.94	94.74	93.32

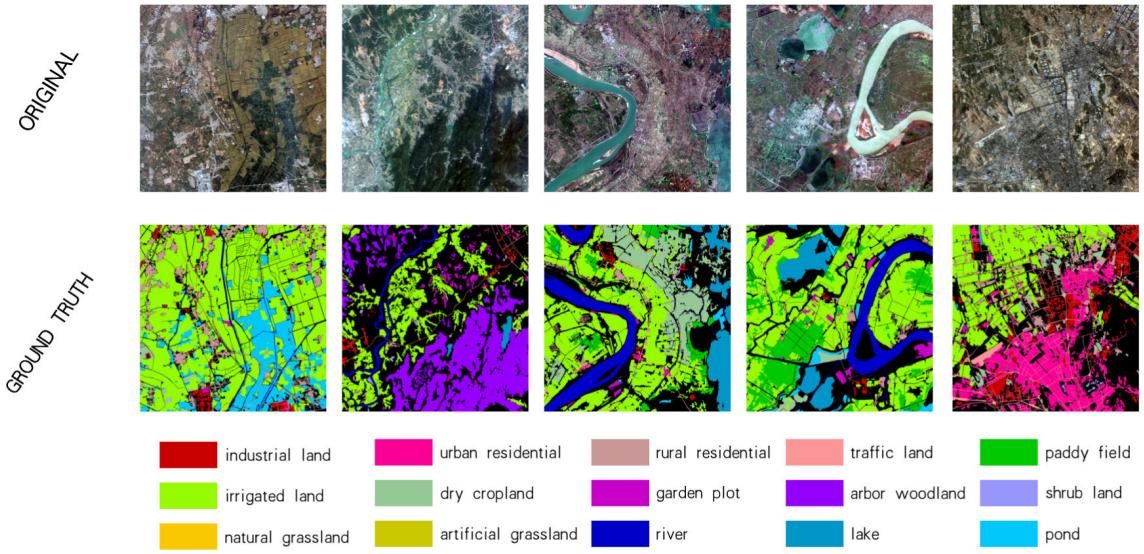

FIGURE 9. Image patch examples from GID. The upper row shows the original RGB-NIR remote sensing images and the corresponding row below shows the ground truth of the semantic label. The class legend is given at the bottom.

values for FCN and Mask-RCNN on a single patch are 170ms and 30ms, respectively.

2) STUDY ON GAOFEN IMAGE DATASET

In this section, we evaluate our method on the Gaofen Image Dataset (GID) [21]. GID is a high-resolution dataset for land cover classification tasks. This dataset consists of two parts: a five-class large-scale land classification dataset and a 15-class fine-grained land classification dataset. In total, there are 150 high-resolution Gaofen-2 (GF-2) images acquired from more than 60 different cities in China [21] from 5 December 2014 to 13 October 2016. Fig. 9 shows some examples of the original images and the corresponding ground truths. We chose the second part as our experimental dataset. This part contains ten, each with a size of 6800 × 7200. We randomly selected eight images as the training set and two images as the test set. We adopted the same data preprocessing as for the Zurich dataset.

Tab. 4 lists the recall, precision, and F1 score for each class. In the experiment, the last two images were selected as test sets. When counting the number of categories, it was found that the last two images did not contain industrial land, paddy fields, dry cropland, or lake categories. These categories were excluded when calculating recall, precision, and F1 scores.

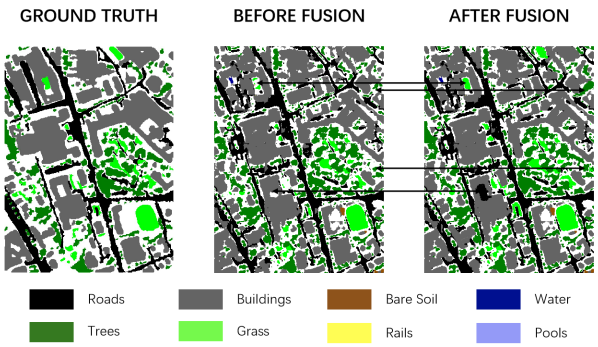

FIGURE 10. Classification maps generated by before- and after-fusion stages on Zurich dataset. The black double arrows point to the examples of the before- and after-fusion results.

From the results, it can be seen that after fusion, recalls of most categories increased. In particular, urban residential and shrub land increased by approximately 2%. The results after fusion also showed significant precision improvement in most categories, including arbor woodland, which gained

TABLE 4. Recalls, Precisions, and F1 scores for each class before and after fusion on GID.

	Before Fusion			After Fusion		
Category	**Precision(%)**	**Recall(%)**	**F1(%)**	**Precision(%)**	**Recall(%)**	**F1(%)**
urban residential	88.52	82.40	85.35	88.56	84.51	86.49
rural residential	0.00	0.00	0.00	0.00	0.00	0.00
traffic land	53.91	0.45	0.89	49.45	0.43	0.86
irrigated land	0.00	0.00	0.00	0.00	0.00	0.00
garden plot	0.00	0.00	0.00	0.17	0.94	0.29
arbor woodland	75.14	42.72	54.47	78.15	42.92	55.41
shrub land	92.37	47.20	62.48	92.13	49.62	64.50
natural grassland	53.52	65.75	59.01	55.59	66.14	60.41
artificial grassland	93.19	21.61	35.08	94.50	22.41	36.23
river	0.00	0.00	0.00	0.00	0.00	0.00
pond	21.73	79.86	34.17	21.72	80.13	34.18

a 3% improvement. For F1 scores, all categories showed slight increases. The categories of rural residential, irrigated land, garden plots, and rivers were poorly predicted. However, after fusion, some results were still retrieved in garden plots.

3) STUDY ON DataFountain 2017

To validate the generalization performance of the model further, the same experiment was conducted on a remote sensing image dataset with lower resolution. This data set is a competition data set relative to DataFountain from 2017 taken in southern China, with a resolution of 8 meters. Detailed information can be obtained at https://www.datafountain.cn/competitions/270.

The classification results were obtained by shallow machine learning algorithms, i.e., random forest, SVM, and other deep learning models, such as the fine-tuned FCN by the remote sensing images, a patch-based CNN with five layers owing to the shortage of labeled remote sensing data, and our Mask-R-FCN method.

TABLE 5. Average accuracies on DATAFOUNTAIN2017 dataset by using different methods.

Methods	Average Accuracy (%)
Pixel-based SVM with RBF Kernel	60.70
Random Forest	48.57
Patched-based CNN	72.01
FCN	76.69
Mask-R-FCN	79.35

The results are shown in Tab. 5. Since Datafountain2017 is a competition dataset, no groundtruth for each class is provided available and only an average accuracy was returned based on the results we provided. The upper results come from shallow models based on single pixels. These results are generally poor. The lower results are all based on deep models. The proposed Mask-R-FCN network can obtain higher accuracy than using a single DNN, such as FCN and other shallow machine learning algorithms. This proves that the effective fusion mechanism is also good at improving the accuracy of classification in remote sensing images with lower spatial resolution.

IV. CONCLUSION

Single deep neural networks usually suffer from making a trade-off between background semantics and detailed semantics. In attempting to better classify background pixels, semantic segmentation models usually exhibit poor performance for pixels within small objects. An object detection method could avoid the problem of insensitivity to small objects. In our work, a Mask-R-FCN deep fusion network was proposed by combining the pixel-based FCN and the object-based Mask-R-CNN to reduce the difficulty of small object segmentation. The semantic labels provided by FCN were modified at the decision level by the results provided by Mask-R-CNN. A strategy of model fusion was proposed to learn whether the instance segmentation results should be used to cover the map of the interest of FCN.

Experiments were performed on real remote sensing images from Zurich and China. The results demonstrated that the proposed Mask-R-FCN network outperformed compared to other deep learning networks, such as single FCNs and other machine learning algorithms. The average accuracy obtained by our method on the Zurich dataset was 82.28%, which is approximately 2% higher than that of other single DNNs. On the GID dataset, there were significant improvements in most categories. According to the experimental results, our proposed method performs well on datasets with lower spatial resolution such as DataFountain2017.

Other domains also have the same problem of misclassifying the pixels belonging to small objects. Further studies can investigate the effectiveness of our proposed method on natural image datasets. Although the proposed method can improve performance in categories of small-scale regions, it brings slight drops on categories of large-scale regions. Further study will be done to search for optimal parameters to improve the accuracies of both small-scale and large-scale classes.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their thoughtful comments and efforts towards improving our manuscript and all the data providers for their hard work.

REFERENCES

[1] C. Corbane, J.-F. Faure, N. Baghdadi, N. Villeneuve, and M. Petit, "Rapid urban mapping using SAR/Optical imagery synergy," *Sensors*, vol. 8, no. 11, pp. 7125–7143, Nov. 2008.

[2] T. Kutser, L. Metsamaa, N. Strömbeck, and E. Vahtmäe, "Monitoring cyanobacterial Blooms by satellite remote sensing," *Estuarine, Coastal Shelf Sci.*, vol. 67, nos. 1–2, pp. 303–312, Mar. 2006.

[3] C. Pohl and J. L. Van Genderen, "Review article multisensor image fusion in remote sensing: Concepts, methods and applications," *Int. J. Remote Sens.*, vol. 19, no. 5, pp. 823–854, Jan. 1998.

[4] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning Earth observation classification using ImageNet pretrained networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, Jan. 2016.

[5] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[6] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, and X. Meng, "Apache spark: A unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[7] W. Huang, L. Meng, D. Zhang, and W. Zhang, "In-memory parallel processing of massive remotely sensed data using an apache spark on Hadoop YARN model," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 1, pp. 3–19, Jan. 2017.

[8] Z. Sun, F. Chen, M. Chi, and Y. Zhu, "A spark-based big data platform for massive remote sensing data processing," in *Proc. Int. Conf. Data Sci.* Cham, Switzerland: Springer, 2015, pp. 120–126.

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

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Processing Syst.*, 2012, pp. 1097–1105.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[12] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat: A learning framework for satellite imagery," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2015, pp. 1–10.

[13] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[14] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3735–3739.

[15] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.

[16] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.

[17] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[18] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," 2017, *arXiv:1704.06857*. [Online]. Available: http://arxiv.org/abs/1704.06857

[19] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.

[20] M. Volpi and V. Ferrari, "Semantic segmentation of urban scenes by learning local class interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 1–9.

[21] X.-Y. Tong, G.-S. Xia, Q. Lu, H. Shen, S. Li, S. You, and L. Zhang, "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sens. Environ.*, vol. 237, Feb. 2020, Art. no. 111322.

[22] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[23] G. Bilgin, S. Erturk, and T. Yildirim, "Segmentation of hyperspectral images via subtractive clustering and cluster validation using one-class support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 2936–2944, Aug. 2011.

[24] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM-and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.

[25] S. Van der Linden, "Classifying segmented hyperspectral data from a heterogeneous urban environment using support vector machines," *J. Appl. Remote Sens.*, vol. 1, no. 1, Oct. 2007, Art. no. 013543.

[26] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, "Hyperspectral image unsupervised classification by robust manifold matrix factorization," *Inf. Sci.*, vol. 485, pp. 154–169, Jun. 2019.

[27] S. K. Pal and P. Mitra, "Multispectral image segmentation using the rough-set-initialized EM algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2495–2501, Nov. 2002.

[28] P. Mitra, B. Uma Shankar, and S. K. Pal, "Segmentation of multispectral remote sensing images using active support vector machines," *Pattern Recognit. Lett.*, vol. 25, no. 9, pp. 1067–1074, Jul. 2004.

[29] R. Kemker, C. Salvaggio, and C. Kanan, "Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 60–77, Nov. 2018.

[30] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 399–409, Sep. 1997.

[31] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognit. Lett.*, vol. 27, no. 4, pp. 294–300, Mar. 2006.

[32] J. Li, H. Zhang, and L. Zhang, "Supervised segmentation of very high resolution images by the use of extended morphological attribute profiles and a sparse transform," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 8, pp. 1409–1413, Aug. 2014.

[33] J. Yuan, D. Wang, and R. Li, "Remote sensing image segmentation by combining spectral and texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 16–24, Jan. 2014.

[34] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[35] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[36] A. Lagrange, B. Le Saux, A. Beaupere, A. Boulch, A. Chan-Hon-Tong, S. Herbin, H. Randrianarivo, and M. Ferecatu, "Benchmarking classification of Earth-observation data: From learning explicit features to convolutional networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 4173–4176.

[37] Y. Liu, Q. Ren, J. Geng, M. Ding, and J. Li, "Efficient patch-wise semantic segmentation for large-scale remote sensing images," *Sensors*, vol. 18, no. 10, p. 3232, Sep. 2018.

[38] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[39] Y. Xu, B. Du, and L. Zhang, "Beyond the patchwise classification: Spectral-spatial fully convolutional networks for hyperspectral image classification," *IEEE Trans. Big Data*, early access, Jun. 17, 2019, doi: 10.1109/TBDATA.2019.2923243.

[40] G. Chen, X. Zhang, Q. Wang, F. Dai, Y. Gong, and K. Zhu, "Symmetrical dense-shortcut deep fully convolutional networks for semantic segmentation of Very-High-Resolution remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 5, pp. 1633–1644, May 2018.

[41] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

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

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[44] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: http://arxiv.org/abs/1511.07122

[45] R. Li, W. Liu, L. Yang, S. Sun, W. Hu, F. Zhang, and W. Li, "DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 3954–3962, Nov. 2018.
[46] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[47] Z. Niu, W. Liu, J. Zhao, and G. Jiang, "DeepLab-based spatial feature extraction for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 251–255, Feb. 2019.
[48] D. Tuia, R. Flamary, and N. Courty, "Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 272–285, Jul. 2015.
[49] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery," 2016, *arXiv:1606.02585*. [Online]. Available: http://arxiv.org/abs/1606.02585
[50] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. van den Hengel, "Semantic labeling of aerial and satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 2868–2881, Jul. 2016.
[51] N. Audebert, B. Le Saux, and S. Lefèvre, "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 180–196.
[52] M. Kang, K. Ji, X. Leng, and Z. Lin, "Contextual region-based convolutional neural network with multilayer fusion for SAR ship detection," *Remote Sens.*, vol. 9, no. 8, p. 860, Aug. 2017.
[53] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
[54] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[55] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2011). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
[56] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[57] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[58] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[59] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
[60] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767
[61] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[62] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
[63] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," 2019, *arXiv:1902.07296*. [Online]. Available: http://arxiv.org/abs/1902.07296
[64] K. Djerriri and M. S. Karoui, "Classification of quickbird imagery over urban area using convolutional neural network," in *Proc. Joint Urban Remote Sens. Event (JURSE)*, Mar. 2017, pp. 1–4.

YUNFENG ZHANG received the B.S. degree in computer science from Hubei University, Wuhan, China, in 2017. He is currently pursuing the M.S. degree with the Fudan University of Computer Science. His research interests include deep learning, model fusion, remote sensing, and face recognition.

MINGMIN CHI (Member, IEEE) received the B.S. degree in electrical engineering from the Changchun University of Science and Technology, Changchun, China, in 1998, the M.S. degree in electrical engineering from Xiamen University, Xiamen, China, in 2002, and the Ph.D. degree in computer science from the University of Trento, Trento, Italy, in 2006. She was a Student Visitor with the Department of Empirical Inference, Max-Planck Institute for Intelligent Systems, Tübingen, Germany, from May 2005 to March 2006. She is currently an Associate Professor with the School of Computer Science, Fudan University, Shanghai, China. Her research interests include data science, big data, and machine learning with applications to remote sensing, intelligent manufacturing, oil and gas exploration, computer vision, and astronomy. She has been a Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING.