FEW-SHOT SEGMENTATION FOR REMOTE SENSING IMAGES WITH METRIC LEARNING

Xiang Li[a,b,c], Fred Lu[a,d] , Yi Fang[a,b,c]*

[a] Multimedia and Visual Computing Lab, New York University, New York, United States.
[b] Tandon School of Engineering, New York University, New York, United States.
[c] Department of Electrical and Computer Engineering, NYU Abu Dhabi.
[d] College of Arts and Science, New York University, New York, United States.

KEY WORDS: Few-Shot Learning, Semantic Segmentation, Prototype Representation, Remote Sensing, DenseCRF

ABSTRACT:

Deep learning-based methods, especially deep convolutional neural networks (CNNs), have made significant breakthroughs in the field of computer vision and greatly advanced the performance of the semantic segmentation of remote sensing images. However, current CNN-based methods for remote sensing image segmentation require a large number of densely annotated images for model training and have limited generalization abilities for unseen object categories. In this paper, we propose probably the first few-shot learning-based method for the semantic segmentation of remote sensing images. Our method can perform semantic labeling for unseen object categories with only a few annotated samples. More specifically, our model starts by using a deep CNN to extract high-level semantic features. The prototype representation of each class is then generated by using a masked average pooling on the feature embeddings of the support images with ground truth masks. Finally, our model performs semantic labeling over the query images by matching the feature embedding of each pixel to its nearest prototypes in the embedding space. Our model is optimized with a non-parametric metric learning-based loss function with an intention to maximize the intra-class similarity of learned prototypes while minimizing the inter-class similarity. We conduct both in-domain and cross-domain experiments to demonstrate the generalization abilities of our model on unseen categories. In the in-domain experiments, our model is trained on one object class from ISPRS 2D semantic labeling dataset and the segmentation performance is evaluated on another new class of the same dataset with few annotated samples. In the cross-domain experiments, our model is trained on PASCAL VOC 2012 dataset, and the segmentation performance is evaluated on ISPRS 2D semantic labeling dataset with few annotated samples. Experiments demonstrate a satisfying in-domain and cross-domain transferring abilities of our model. We also show that our model is capable of performing few-shot segmentation of new classes with weak annotations, such as bounding boxes and scribbles.

1. INTRODUCTION

The semantic segmentation problem, which is often called image classification in the remote sensing field, is generally defined as determining the semantic class of each pixel in the input images. Recent years, deep learning, especially deep convolutional neural networks, has achieved significant breakthroughs in many applications such as scene classification (Zou et al., 2015; Cheng et al., 2018), image classification (Maggiori et al., 2016; Marmanis et al., 2018; Audebert et al., 2018), object detection (Chen et al., 2014b; Hu et al., 2019), building extraction (Mnih, 2013; Saito et al., 2016; Alshehhi et al., 2017; Li et al., 2018), land use classification (Luus et al., 2015; Castelluccio et al., 2015), point cloud classification (Yang et al., 2017; Zhao et al., 2018; Wen et al., 2019). However, training a deep neural network model usually requires large-scale annotated data. While, data labeling is a time consuming and labor-intensive task, especially for semantic segmentation which needs pixel-wise annotations. Semi- or weakly supervised learning methods, such as (Dai et al., 2015; Lin et al., 2016; Papandreou et al., 2015) are proposed to alleviate such requirements, but they still need a large amount of annotated data for training. Moreover, after training, these methods cannot generalize well to unseen classes.

In contrast to machine vision algorithms, humans can easily identify a new object class (e.g., in classification and semantic segmentation) after seeing only a few examples. In order to equip machine vision algorithms with this powerful learning abilities, recent researches started the new research topic of few-shot learning that aims to learn a model that can generalize well to new classes with few annotated data. In recent years, few-shot learning has been actively explored in computer vision field with applications to image classification (Finn et al., 2017; Oreshkin et al., 2018), semantic segmentation (Dong and Xing, 2018; Wang et al., 2019), and object detection (Kang et al., 2019; Chen et al., 2018a).

In remote sensing applications, a common situation is that one has trained a segmentation model for a certain class (e.g., building) but want to perform segmentation on a new class (e.g., road). In the previous setting, the training and testing stage are operated on the same set of pre-defined classes, so one would always need a large number of new training data in order to perform segmentation for new classes. In this paper, we propose probably the first few-shot learning-based method for the semantic segmentation of remote sensing images. More specifically, we aim to perform semantic segmentation of remote sensing images with only use a few annotated samples of new classes.

In the few-shot learning scenario, a model is designed to tackle new classes that have not been seen during model training. Instead of learning to extract feature representation for pre-defined classes, the model should have the arability to learn transferable knowledge, i.e., it should learn to extract feature patterns for both seen and unseen classes. A common way to achieve this goal is to separate the whole learning process as prototype extraction and non-parametric metric learning. The prototypes extraction part leverage the powerful feature learning abilities of CNNs to extract a feature vector representation for each semantic class. While the

*Corresponding author. Email: yfang@nyu.edu.

non-parametric metric learning part conducts pixel-wise segmentation through the nearest neighbor matching in the embedding space. After network training, each learned prototype is a compact and robust representation of the corresponding class and is at the same time sufficiently distinguishable from other classes. In the testing stage, our model performs semantic segmentation for each pixel of the query image by matching the feature embeddings to its closest prototype representation.

We list the main contributions of the proposed method as follows:

1. This paper introduces probably the first few-shot learning-based method for the semantic segmentation of remote sensing images. Our proposed model is able to perform semantic labeling on new classes using only a few annotated samples.
2. Our model is optimized by a non-parametric metric learning objective and learns to maximize in-class similarity while minimizing the inter-class similarity.
3. We investigate both in-domain and cross-domain transferring abilities of our model. We show that our model is able to learn transferable representations and perform semantic labeling on new classes in the same domain or from a different domain.
4. We investigate the segmentation performance of our model with weak annotations, including bounding boxes and scribbles. To the best of our knowledge, this is the first attempt to perform segmentation with weak annotations in the remote sensing field.
5. With only 5 annotated image patches, our model gets satisfying performance on the ISPRS 2D Semantic Labeling Challenge dataset, with an overall accuracy of 67.7% and 52.8% for Vaihingen and Potsdam dataset.

The remainder of this paper is organized as follows. In Section 2., we give a brief review of the deep learning-based semantic segmentation methods and the few-shot learning methods for semantic segmentation. The proposed few-shot learning-based segmentation method is described in detail in Section 3.. In Section 4., we conduct experiments to verify both in-domain and cross-domain transferring abilities of the proposed method for semantic labeling on unseen classes. We further discuss the effectiveness of different configurations of our method and investigate the segmentation performance of our model with weak annotations in Section 5.. Finally, the paper is concluded in Section 6..

2. RELATED WORK

2.1 Semantic Segmentation

Benefiting from the advances of convolutional neural networks (CNN), image semantic segmentation has obtained significant breakthroughs. Fully Convolutional Networks (FCN) (Long et al., 2015) stands out as probably the first approach to perform dense predictions by converting fully connected layers into convolutional layers. It thus allows the FCN model to generate segmentation maps for input images of arbitrary sizes. Later, FCN-like methods have achieved great progress in the task of semantic segmentation. In order to recover the object details and spatial information reduced by pooling layers, two typical strategies are exploited. SegNet (Badrinarayanan et al., 2017), UNet (Ronneberger et al., 2015), RefineNet (Lin et al., 2017) and Deeplabv3+ (Chen et al., 2018b) leverage encoder-decoder structure to recover detailed spatial information by fusing low-level and high-level layers step-by-step. On the other hand, dilated convolution is used in (Yu and Koltun, 2015; Chen et al., 2014a, 2017; Zhao et al., 2017) to reserve high-resolution feature maps while enlarging the receptive field of the neural network.

In the remote sensing field, semantic segmentation of remote sensing images is also called "semantic labeling" or "image classification". Mainly thanks to the progress of deep learning in the computer vision community on natural RGB images, semantic segmentation on VHR remote sensing images have achieved significant improvements. (Mnih and Hinton, 2010) is the first successful work that utilizes patch-based CNN for road extraction. Saito et al. (2016) performs road and building predictions using a single CNN network on an aerial imagery dataset. Vakalopoulou et al. (2015) extended the approach to perform building extraction from VHR multispectral images including visible and infrared bands. However, the patch-based classification approach only produces coarse maps, as an entire patch gets associated with only one label. Later, with the propose of FCN, many works ((Maggiori et al., 2016; Marmanis et al., 2016; Sherrah, 2016)) managed performing pixel-wise classification based on end-to-end trainable fully convolutional architectures. For example, (Maggiori et al., 2016) devises an end-to-end fully convolutional architecture for fine-grained pixel-wise classification on large-scale remote sensing images. Marmanis et al. (2016) designs an FCN to perform pixel-wise classification on the ISPRS semantic labeling benchmark. They also discuss different design choices of the proposed method and demonstrate an ensemble of CNNs can achieve better results. Sherrah (2016) proposes to infer a full-resolution labeling map using a deep FCN with no downsampling layers, avoiding the need for deconvolution or upsampling layers. Indeed, FCNs are well suited for semantic segmentation of remote sensing images, as they are designed to be able to produce high-resolution predictions. However, with consecutive downsampling operations, fine spatial information would lose, especially for boundary details. To recover the reduced spatial information caused by downsampling layers, recent works adopt SegNet (Badrinarayanan et al., 2017) as a baseline to perform accurate pixel-wise scene labeling of remote sensing images. For example, (Audebert et al., 2016) proposes a variant of SegNet architecture and present a multi-kernel convolutional layer to smooth the predictions. Audebert et al. (2018) proposes an multi-scale deep fully convolutional neural network build on SegNet (Badrinarayanan et al., 2017) and ResNet (He et al., 2016) and they investigate early and late fusion of multi-modal remote sensing data. Marmanis et al. (2018) explicitly adds a boundary detection branch to the SegNet (Badrinarayanan et al., 2017) architecture to reserve high-frequency details and object boundaries. They show that adding a boundary detection network can significantly improve the performance of semantic segmentation of remote sensing images.

2.2 Few-Shot Segmentation

Few-shot learning, as one of the supervised meta-learning methods, aims at learning to learn transferable knowledge that can be generalized to new classes and therefore performs image recognition (e.g., classification, segmentation) on new classes with only a few annotated examples. Few-shot segmentation is receiving growing attention recently in the computer vision field. Existing few-shot segmentation methods fall into two categories. The first category of methods employs a two-branch structure which consists of a support branch and a query branch. The support branch extracts information from support images and then the learned knowledge is used to guide the segmentation for the query branch. Shaban et al. (2017) first proposes a two-branch architecture for few-shot segmentation. The conditioning branch generates a set of parameters θ from the support set, which is then

used to guide the segmentation branch to predict the segmentation mask for the query images. Rakelly et al. (2018) proposes to generate feature embeddings from support branch, which is then fused to the query branch as additional features. Recently, (Zhang et al., 2019) proposes a dense comparison module to generate dense feature correlations between the support images and query images. They also adopt an iterative optimization module to iteratively refine the segmentation results and an attention module to fuse information from multiple support examples under the setting of k-shot learning. On the other hand, another recent work, (Wang et al., 2019), proposes to separate these two branches as prototype extraction and non-parametric metric learning. They follow Prototypical Network (Snell et al., 2017) to extract prototype representations for the foreground objects and background using a shared backbone network for both the support and the query images and then leverage a non-parametric metric learning strategy to achieve pixel-level segmentation through nearest-neighbor matching within the embedding space. Our model follows PANet (Wang et al., 2019) to perform few-shot semantic segmentation for remote sensing images. Both in-domain and cross-domain experiments on ISPRS 2D semantic labeling dataset demonstrate the effectivity and transferring abilities of our model.

3. METHODS

In this section, we introduce the proposed method for few-shot segmentation of remote sensing images. First, we introduce the background of few-shot segmentation and state our few-shot segmentation problems in Section 3.1. Then, we give an overview of the proposed method in Section 3.2. The feature extraction network and prototype learning are introduced in Section 3.3 and Section 3.4 respectively. Our model is optimized with a Non-parametric metric learning loss, which is illustrated in Section 3.5.

3.1 Problem Statement

The problem of few-shot segmentation aims at learning a segmentation model from the dataset of available classes that can perform semantic labeling for some unseen classes using the information from only a few annotated images from the same unseen classes. To achieve this goal, a model is generally trained on a dataset D_{train} with the class set C_{seen} and learns to extract transferable representations that can be applied to the new dataset D_{test} with new classes C_{unseen}, where C_{unseen} and C_{seen} have no overlap. C_{seen} C_{unseen} can come from different domains. After training, the model parameters are fixed and require no optimization in the testing stage.

To facilitate model training and evaluation, we construct several episodes from the training and testing set. Each episode E_i is constructed from a set of support images S_i (with annotations) and a set of query images Q_i. Given a C-way-K-shot segmentation task, each support set S_i consists of K annotated images per semantic class and there are in total C different classes from C_{seen} for training. We denote the support set as, $S_i = \{(I_k, M_{ck})\}$ where I_k denotes the input image and $I_k \in \mathbb{R}^{H_i \times W_i \times 3}$, and M_{ck} denotes the binary mask for class c and $M_{ck} \in \mathbb{R}^{H_i \times W_i}$, $k = 1, 2, ...K$ and $c \in \mathcal{C}_i$ with $|\mathcal{C}_i| = C$. Note that we deal with the mask of each class independently. The query set Q_i contains N_q images from the same set of class $\mathcal{C}_i$ as the support set. The support images are used for prototype learning and our model performs segmentation for the query images by matching each pixel in the query images to these learned prototypes. During training, the predicted segmentation mask of query images is compared with the ground truth ones to guide network training. As each episode contains a different set of semantic classes, our model is, therefore, able to perform class-agnostic feature learning and thus can generalize well to unseen classes. After training on the training set D_{train} until convergence, we evaluate the segmentation performance on all episodes from the test set D_{test}. More specifically, for each testing episode, our model predicts the segmentation maps for all input images in the query set Q_i given the support set S_i.

3.2 Method Overview

Different from classical semantic segmentation methods that learn to extract representative feature representations for pre-defined classes, our model is designed to perform class-agnostic feature learning and thus can generalize well to unseen classes. To achieve this goal, our model is designed to learn to extract robust and compact prototype representation for each semantic class. Then it predicts the semantic label for each pixel in the query images via non-parametric feature matching in the embedding space.

Fig 1 gives an overview of the proposed method for few-shot segmentation of remote sensing images. Our model starts with a feature extraction network to learn deep feature representations for the support and query images. We use a shared backbone network to extract feature representations for both support and query images, details will be introduced in section 3.3. Then, a masked average pooling layer is used to extract prototypes from the support sets, as described in section 3.4. Finally, our model predicts the segmentation labels for each pixel in the query images by matching its feature embedding to its nearest prototype representation. Our model is optimized with a non-parametric metric learning objective and learns to maximize in-class similarity while minimizing the inter-class similarity, as described in section 3.5.

3.3 Feature Extraction

Our feature extraction network is designed to incorporate different levels of feature representations from deep CNNs for prototype learning. We build the feature extraction network from ResNet-50 (He et al., 2016) architecture and pre-train it on ImageNet (Russakovsky et al., 2015). Previous works in the computer vision field have frequently observed that in a CNN model, the features in shallower layers usually represent low-level cues, such as edges and corners, while features in deeper layers tend to represent object-level information such as object categories. In the few-shot segmentation scenario, our model is designed to perform class-agnostic feature learning which should adapt well to unseen classes. Thus we can not use those high-level features that represent class-specific characteristics of training classes. Instead, we only focus on middle-level features that share across different classes in both seen and unseen classes.

The original ResNet-50 architecture consists of 4 residual blocks, with each block corresponding to a different level of representation. We choose the feature maps from block2 and block3 to formulate the feature embeddings for prototype learning. To maintain the spatial resolution, convolution layers in block3 are replaced by dilated convolutions (Yu and Koltun, 2015) with the dilation factor set to 2. Figure 3 gives an illustration of the dilated convolution used in our model. In this way, all feature maps after block2 have a fixed spatial resolution of 1/8 of the input image. Moreover, in a dilated convolution, the convolutional layer has a larger receptive field which enables informative inputs from a larger area. We concatenate the output of block2 and block3 and use another 3×3 convolution layer to compress the feature dimension to 256. The feature extraction network is shared by both support images and query images for feature learning.

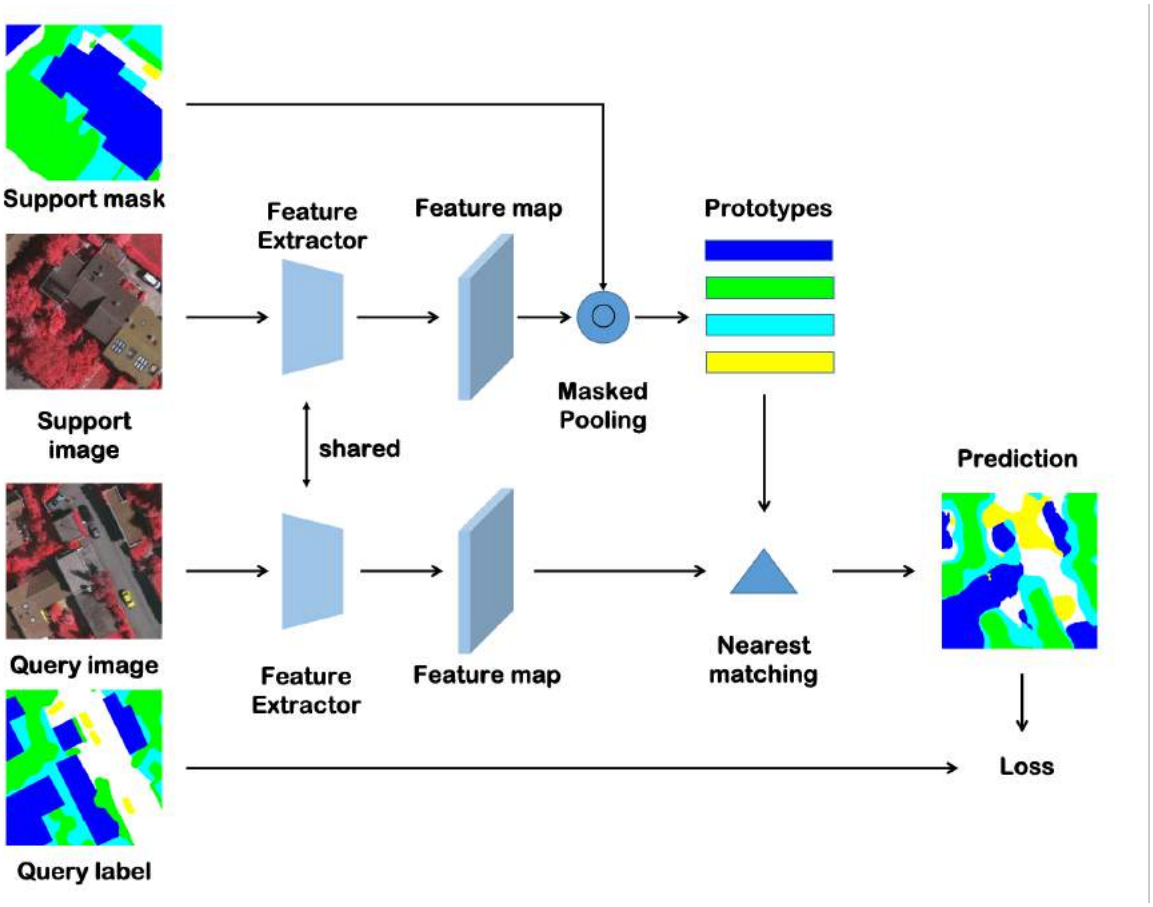

Figure 1: Overview of the proposed few-shot segmentation method for remote sensing images. Our model starts with a deep CNN to extract high-level semantic feature maps. Then, the prototype representation of each class is learned by a masked average pooling by leveraging the ground truth segmentation masks of support images. Finally, our model performs semantic labeling for the query images by matching the feature embedding of each pixel to its closest prototype representation. Our model is optimized with a non-parametric metric learning-based loss function.

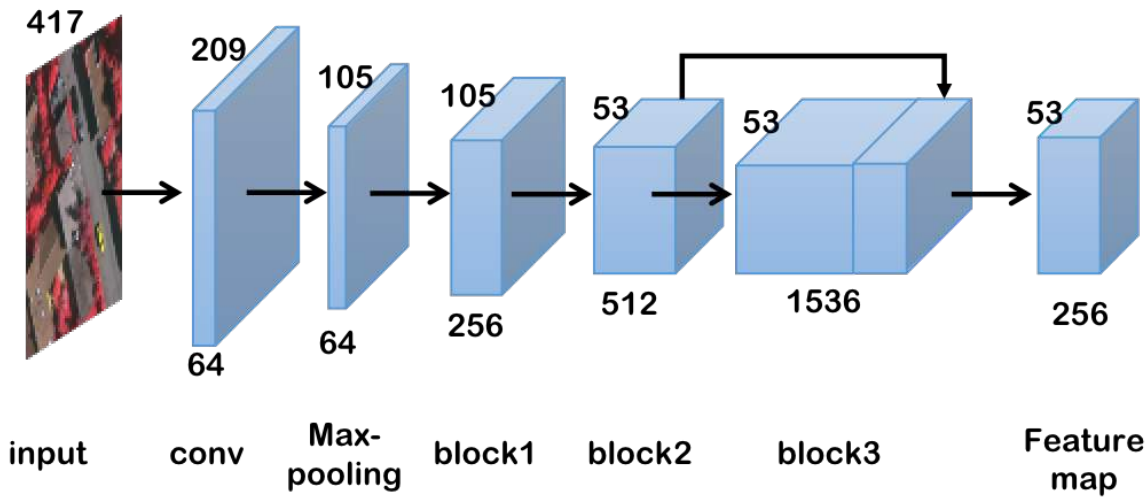

Figure 2: Illustration of the feature extraction network using ResNet-50 backbone. This backbone network consists a convolutional layer, a max-pooling layer, and 3 residual blocks. There is a residual connection between the output of block2 and block3.

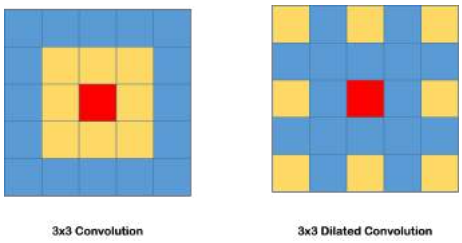

Figure 3: Illustration of the dilated convolution. Left: vanilla 3×3 convolution, right: a dilated 3×3 convolution with a dilation set to 2.

3.4 Prototype Learning

A prototype representation is generally regarded as a compressed feature vector that conveys the necessary information for distinguishing a specific class from others. In our method, the prototype representation is defined in the middle-level embedding space which ensures it to be robust towards different object categories and input variations, such as different structures, light change, occlusion, and truncation. To generate the class-wise prototype feature representation, we aggregate the deep features among different pixel locations of the same semantic class. More specifically, we exploit the ground truth annotation mask of the supported image and leverage a masked average pooling over the whole feature images to generate the prototypes for each foreground class and background class independently. In general, there are two strategies to make use of the annotation masks, i.e., early fusion and late fusion. Early fusion means we apply the semantic masks to support images before we feed them to the feature extraction network, i.e., we perform feature extraction with the masked images. While in late fusion, we directly apply the semantic masks to the feature maps. One may note that the deep feature maps and semantic masks are of different spatial resolutions. To tackle this issue, we leverage bilinear interpolation to upsample the feature maps to the same resolution as the semantic masks. In this paper, we adopt the late fusion strategy because it can maintain more information of the input image rather than abandon them and it also benefits input consistency for the shared feature extractor.

More specifically, given a support set $S_i = \{(I_k, M_{c,k})\}$ and the corresponding feature maps F_k after bilinear interpolation, where c and k denotes the class and sample index. The prototype representation of class c can be calculated through a masked average pooling following:

$$P_c = \frac{1}{K}\sum_k \frac{\sum_{(x,y)} F_k(x,y)\mathbb{I}(M_{c,k}(x,y)=1)}{\sum_{(x,y)}\mathbb{I}(M_{c,k}(x,y)=1)} \tag{1}$$

where (x, y) denotes the pixel coordinates on feature maps and on masked images and $\mathbb{I}(\cdot)$ represents an indicator function which outputs 1 if the input is true or 0 otherwise.

To facilitate network construction, we implement masked average pooling by first multiplying the mask of each class c with the upsampled feature map of the same resolution. In this way, all pixels do not belong to class c will be filled with zeros. Then we utilize a normal average pooling layer to aggregate the feature vectors over the whole masked feature map. We generate prototype representation for each class independently.

3.5 Non-parametric metric learning

Given the prototype representations, our model can predict the semantic segmentation maps for the query images by matching the deep feature vector of each pixel to its nearest prototype in the embedding space. To achieve this goal, we calculate the distance between feature vector of each pixel location in query images to all prototypes in the embedding space and search the minimal neighbor prototype and assign the corresponding class to this pixel location, see Figure 4 for illustration. To facilitate back-propagation, we also generate the probability maps by applying a softmax function over the distance values for each pixel location. We achieve this goal by the following equation:

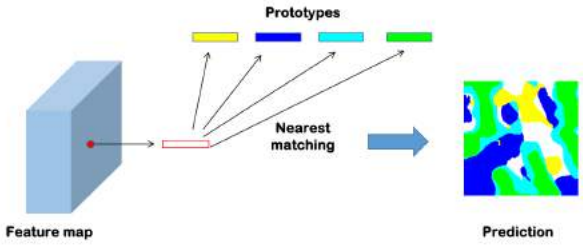

Figure 4: Illustration of pixel-wise segmentation by matching the feature vector to its nearest prototype in the embedding space.

$$P_{c,j}(x,y) = \frac{e^{-\alpha\mathcal{D}(F_j(x,y),P_c)}}{\sum_c e^{-\alpha\mathcal{D}(F_j(x,y),P_c)}} \tag{2}$$

where $F_{c,j}(x, y)$ denotes the query feature vector for the jth query image at location (x, y) and $\mathcal{D}(\cdot,\cdot)$ denotes the distance function defined in the embedding space, and α is a hyper-parameter to control probability distribution with regards to distance values. In our experiments, we set α to 20 since different values lead to similar performance. The final segmentation map for jth query image can be generated as:

$$M_j(x,y) = \arg\max_c P_{c,j}(x,y) \tag{3}$$

For the distance function $\mathcal{D}(\cdot,\cdot)$, a common choice is to use either cosine distance or Euclidean distance. According to (Wang et al., 2019), the cosine distance is generally more stable and provides better performance, probably because it has bounded output and easier to optimize. In this paper, we use cosine distance for $\mathcal{D}(\cdot,\cdot)$, calculated as,

$$\mathcal{D}(a,b) = \frac{a\cdot b}{||a||\cdot||b||} \tag{4}$$

where a, b denote two feature vectors, and $||\cdot||$ calculates vector norm.

Given the predicted probability map $P_{c,j}$ and the corresponding ground truth segmentation label map $\tilde{M}_j$, we formulate the segmentation loss as:

$$\mathcal{L}_{seg} = -\frac{1}{N}\sum_c\sum_j\sum_{(x,y)} log(P_{c,j}(x,y))\mathbb{I}(\tilde{M}_j(x,y)=c) \tag{5}$$

where N denotes the total number of pixel location in each query image. The above loss function will force the extracted feature vector of a certain class to be clustered around the learned prototypes of the corresponding class meanwhile the clustering center (i.e., prototypes) of different classes are forced to be far from each other. The above segmentation loss indeed makes the prototype learning process become a metric learning problem. After optimization, the learned prototypes are naturally well-separable for each semantic class.

To further boost the performance, we leverage the prototype alignment regularization proposed in (Wang et al., 2019) to fully exploit the input information for few-shot learning. Our hypothesis is that if the model can accurately predict the segmentation mask for query images from the prototypes extracted in support images, the prototypes extracted in query images should also be able to well segment support images. We, therefore, define an extra segmentation loss by comparing the segmentation mask predicted by matching each pixel in support images to its nearest neighbor prototypes extracted from query images and the ground truth segmentation labels of support images. One can regard this regularization loss as swapping the support and query set and use the predicted segmentation mask of the query set as its ground truth labels. After adding this prototype alignment regularization $\mathcal{L}'_{cls}$, the final loss function of our model becomes:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}'_{cls} \tag{6}$$

where λ denotes the hyper-parameter to balance these two loss terms. In our experiments, we set λ to 1 because different values have a small influence on the final segmentation performance. By using this loss function, our model naturally enforces a mutual alignment between the prototypes learned from support and query images. The whole training and testing process of our model is summarized in Algorithm 1.

Note that an important prerequisite of this prototype alignment regularization is that our segmentation loss function defined in Eq. (5) is symmetric which is naturally guaranteed by the symmetric distance function, which means $\mathcal{D}(a, b) = \mathcal{D}(b, a)$.

Algorithm 1 Training and testing process.

Construct training and testing episode
Initialize the network parameters and choose hyper-parameters
for each training episode $(S_i, Q_i) \in D_{train}$ **do**
    Extract feature maps for support and query images
    Compute prototypes from support images use masked average pooling defined in Eq. (1)
    Generate segmentation maps for the query images using Eq. (2) and Eq. (3)
    Compute loss function using Eq. (5) and Eq. (6)
    Compute the gradient and optimize via SGD
end for
for each testing episode $(S_i, Q_i) \in D_{test}$ **do**
    Extract feature maps for support and query images
    Compute prototypes from support images use masked average pooling defined in Eq. (1)
    Generate segmentation maps for the query images using Eq. (2) and Eq. (3)
end for

3.6 Post-processing

In our method, the segmentation label of each pixel in query images is assigned independently according to their distance to the prototypes. This will unavoidably cause unwanted label inconsistency in the predicted segmentation maps, see Fig. 8 for illustration. To address this issue, we leverage the widely adopted post-processing method, i.e., DenseCRF (Krähenbühl and Koltun, 2011), to refine the segmentation maps. The DenseCRF model enforces spatial consistency by penalizing the pair-wise energy among all pixel-pairs in the input image (refer to (Krähenbühl and Koltun, 2011) for more details). We employ two different strategies to refine the segmentation maps. The first strategy uses DenseCRF to refine each patch prediction and then we aggregate all refined patches to generate the final prediction for each tile. The second strategy firstly aggregates patch predictions to get a segmentation map for each tile and then we adopt DenseCRF on the whole image. We mark our methods with these two different strategies as 'Ours w DenseCRF(v1)' and 'Ours w DenseCRF(v2)' thereafter. As a default setting, we use the second version in our experiments.

4. EXPERIMENTS AND RESULTS

In this section, we conduct experiments to demonstrate the effectiveness of our model for semantic labeling of remote sensing images. We introduce the experimental dataset in section 4.1. In section 4.2, to see the cross-domain transferring ability of our proposed model, we trained our model on PASCAL Visual Object Classes Challenge 2012 (VOC 2012) dataset and evaluate the performance on ISPRS 2D semantic labeling dataset. In section 4.3, to see the in-domain transferring ability of our proposed model, we trained the model on some categories of ISPRS 2D semantic labeling dataset and evaluate the performance on other categories.

4.1 Datasets

In order to verify the effectiveness of our proposed model for semantic labeling of remote sensing images, we conduct experiments on the ISPRS 2D Semantic Labeling Challenge dataset. This dataset contains very high-resolution aerial images over two cities in Germany: Vaihingen and Potsdam. And for each aerial image, the ground truth labels are provided on six classes: buildings, impervious surfaces (e.g. roads), low vegetation, trees, cars, and clutter. The corresponding DSM information generated by dense image matching is also provided.

4.1.1 ISPRS Vaihingen The Vaihingen dataset contains 33 image tiles with a spatial resolution of 9cm/pixel, and each image has a size of around 2500×2500 pixels. Each aerial image is composed of three channels of infrared, red, and green. Following the official split, 16 tiles with provided ground truth are used for training, and the remaining 17 images are used for held-out evaluation by the challenger organizers. To enable a fair comparison with existing methods, we select 4 tiles (image numbers 5, 7, 23, 30) from the training split and use them for model evaluation. All results are reported on the validation set unless noted otherwise.

4.1.2 ISPRS Potsdam The Potsdam dataset contains 38 image tiles with a spatial resolution of 5cm/pixel, and each image has a size of 6000×6000 pixels. Each aerial image is composed of three channels of infrared, red, green, and blue. Following the official split, 24 tiles with provided ground truth are used for training, and the remaining 14 images are used for held-out evaluation by the challenger organizers. To enable a fair comparison with existing methods, we select 4 tiles (image numbers 7_8, 4_10, 2_11, 5_11) from the training split and use them for model evaluation. All results are reported on the validation set unless noted otherwise.

4.2 Cross-domain Segmentation

4.2.1 Experimental Settings We first demonstrate the segmentation abilities of our proposed model under different domains. To achieve this, we train our model on PASCAL VOC 2012 dataset and evaluate the performance on ISPRS 2D semantic labeling dataset. We follow the experimental settings used in (Wang et al., 2019) and train our model on the PASCAL-5i (Shaban et al., 2017) dataset.

In the training phase, we initialize the backbone feature extraction network with the weights pre-trained on ILSVRC (Russakovsky

et al., 2015) as in previous works (Shaban et al., 2017; Wang et al., 2019). All input images are resized to a resolution of (417, 417). For data augmentation, we randomly flip the training images both horizontally and vertically. In our experiments, we train our model on PASCAL-5i dataset with 1-way 5-shot configuration, with N_{query} set to 1. Our model is optimized using SGD algorithm with initial learning rate and momentum set to 0.001 and 0.9 respectively. We divide the learning by 10 every 10,000 iterations. The batch size is set to 1 and the weight decay is set to 0.0005.

4.2.2 Evaluation Metrics Following the benchmark settings of ISPRS 2D Semantic Labeling Challenge, we use overall accuracy and F1 score of each class to evaluate the segmentation performance. The overall accuracy is calculated as the ratio of correctly classified pixels in all test images. It measures the classification performance for all classes as a whole. While F1 score is defined as the geometric mean of precision and recall, which measures the performance for each classes separately. The calculation of overall accuracy and F1 score are defined as follows:

$$OA = \frac{\sum_i (tp_i + tn_i)}{\sum_i (tp_i + tn_i + fp_i + fn_i)} \tag{7}$$

$$precision_i = \frac{tp_i}{tp_i + fp_i} \tag{8}$$

$$recall_i = \frac{tp_i}{tp_i + fn_i} \tag{9}$$

$$F1_i = 2 * \frac{precision_i * recall_i}{precision_i + recall_i} \tag{10}$$

where i denotes the class index, choosen from $\{1, 2, .., C\}$, tp_i (true positive)/tn_i (true negative) denote the number of the positive/negative pixels of class i that were correctly classified and fp_i (false positive)/fn_i (false negative) denote the number of negative/positive pixels of class i that were incorrectly classified. According to the evaluation instructions from the challenge, the F1 score of each class is computed after eroding the borders of ground truth labels by a 3 pixels radius circle and discarding those pixels during evaluation.

4.2.3 Results on Vaihingen During the model evaluation, we divided the testing images into small patches, and each patch has a size of 417×417. For each path, we set it as the query image and randomly sample 5 patches from the training split as support images to formulate an episode. We feed each episode into our model and get the prediction result for each path. After that, the prediction result for each validation tile can be generated by collecting all patch predictions of the same tile. At last, we use DenseCRF to refine our results and get smooth predictions. We list the performance of our model and some recent works in Table 1. Note that all comparing methods use the full training set for model training, while our results are directly generated without training on the ISPRS dataset. As shown in Table 1, our model gets a satisfying overall accuracy of 67.7%, which is reasonable considering our results are generated from only 5 small patches with annotations. Also, our model gets an F1 score of above 70% on impervious surface and low vegetation classes respectively.

We also give some qualitative results in Fig. 5. As shown in this figure, our model gets satisfying results for these two images in the validation set. Moreover, our model successfully classifies most of the pixels in the validation set, especially, it detects almost all building objects and trees. The classification errors are mostly coming from the misclassification of low vegetation and trees.

4.2.4 Results on Potsdam For the Potsdam dataset, each image has a very high spatial resolution, which causes a 417×417 patch cannot cover sufficient information for semantic labeling. An extreme case is that a patch may contain only one object class. To address this issue, we downsample both the aerial images and DSM images by a factor of 2, i.e., the images are downsampled to a spatial resolution of 3000×3000 and we report our performance on the downsampled labels. Moreover, when randomly selecting support image patches, we make sure that the selected images cover all classes. Other settings are the same as we do for the Vaihingen dataset. In our experiments, we found that using VGG-Net (Simonyan and Zisserman, 2014) as the backbone for feature extraction can lead to slightly better performance compared with ResNet50 architecture, so we report our results using VGG-Net here. We list the quantitative results in Table 2 and the qualitative results in Figure 6.

As shown in Table 2, our model achieves an overall accuracy of 52.2% and 52.8% with IRRG and RGB channels as inputs respectively. From the results, we can see that our model performs slightly better with RGB inputs. By analyzing the F1 score for each class, one can observe that the improvement mainly comes from the better classification performance on low vegetation. Figure 6 shows some selected examples of our segmentation results. As can be seen in this figure, given 5 randomly selected patches with annotations, our model can still get a reasonable result. One may also note that our model gets inferior results on the Potsdam dataset compared to what we get for the Vaihingen dataset. This is probably because the object structures in Vaihingen dataset are more similar to the PASCAL VOC 2012 dataset.

4.3 In-domain Category transfer

In this section, we conduct experiments to verify the in-domain transferring abilities of our model. We train the model on selected classes from the ISPRS Vaihingen dataset and evaluate the segmentation performance on other classes with only a few annotated images. We conduct experiments with two settings: 1) we train our model on the classes including impervious surface, low vegetation, tree, and car and test the performance on building class, 2) we train our model on the classes including building, low vegetation, tree, and car and test the performance on impervious surface. During training, we randomly flip our training images horizontally and vertically and no other augmentations are used. We do not use DenseCRF to post-process the prediction results in this section. Table 3 list the performance of these two experiments.

As shown in Table 3, our proposed method gets an overall accuracy of 77.4% and 72.0% on building and impervious surface categories of the Vaihingen dataset with only 5 support label images. Moreover, after training our model on some other classes, our model is able to get significantly better performance than the cross-domain setting (73.1% and 62.6% for building and impervious surface) on a specific class with only a few annotated label images. This finding further demonstrates a satisfying property of our model, i.e., the few-shot segmentation performance can be boosted by training on some other classes of the same domain. Previous works in remote sensing domains only focus on the development of a more powerful model that can only be applied to pre-determined classes and cannot transfer to new classes. In contrast, our model can generalize well to new classes with only a few annotated label images and enjoys a performance boost from an existing in-domain dataset.

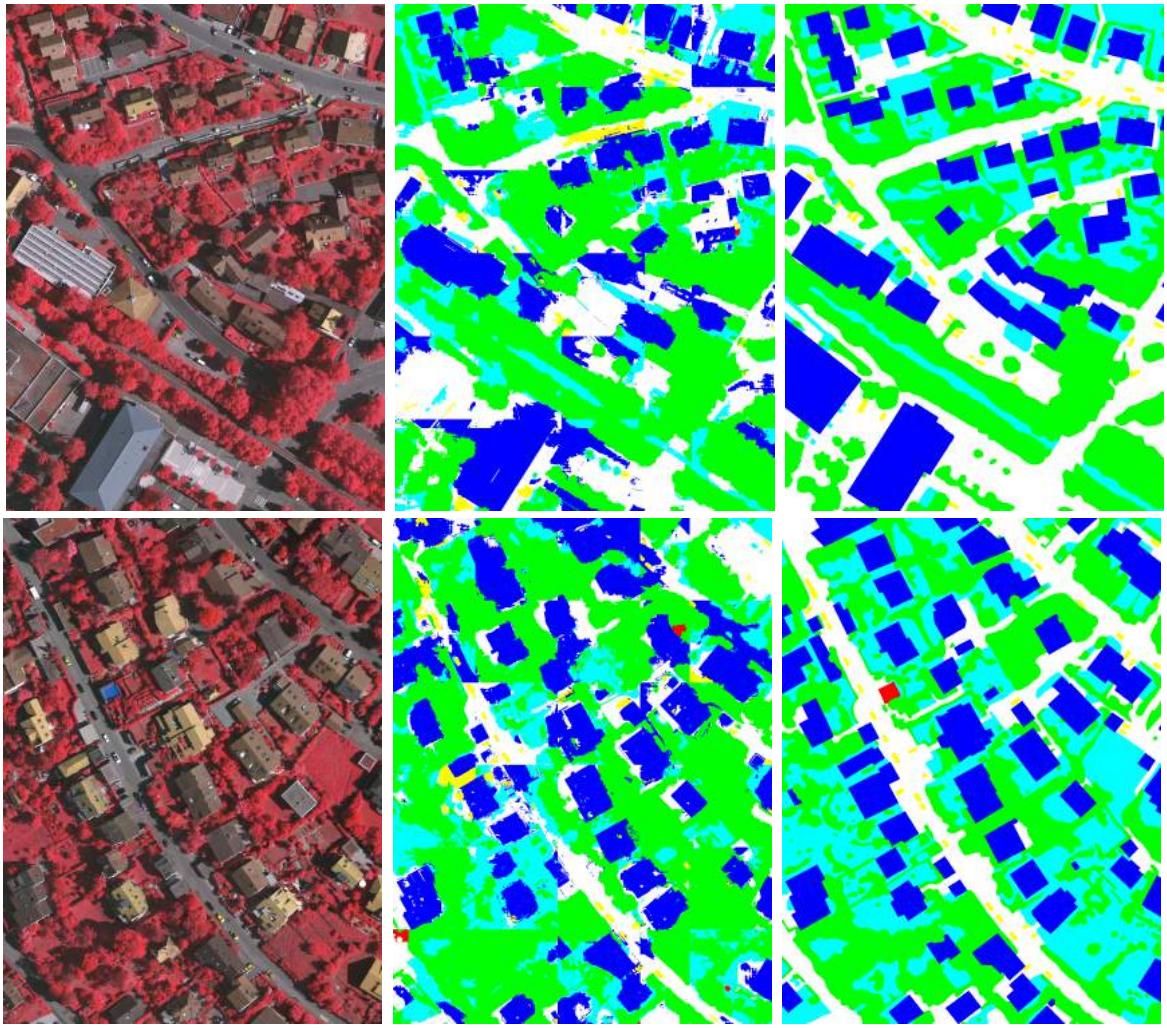

Figure 5: Segmentation results of our method on ISPRS Vaihingen dataset. From left to right: aerial images, our predictions, ground truth labels.

Method	Imp. surf.	Buildings	Low veg.	Trees	Cars	Overall
FCN (Sherrah, 2016)	90.5	93.7	83.4	89.2	72.6	89.1
FCN+fusion+boundaries (Marmanis et al., 2018)	92.3	95.2	84.1	90.0	79.3	90.3
SegNet (IRRG) (Audebert et al., 2018)	91.5	94.3	82.7	89.3	85.7	89.4
Ours	62.6	73.1	38.7	80.5	41.1	67.7

Table 1: Results on the Vaihingen dataset.

Method	Imp. surf.	Buildings	Low veg.	Trees	Cars	Overall
FCN+CRF+expert features (Liu et al., 2017)	91.2	94.6	85.1	85.1	92.8	88.4
FCN (Sherrah, 2016)	92.5	96.4	86.7	88.0	94.7	90.3
SegNet (IRRG) (Audebert et al., 2018)	92.4	95.8	86.7	87.4	95.1	90.0
Ours (IRRG)	50.0	67.8	34.3	56.8	26.0	52.2
Ours (RGB)	40.2	67.3	50.9	56.6	26.3	52.8

Table 2: Results on the Potsdam dataset.

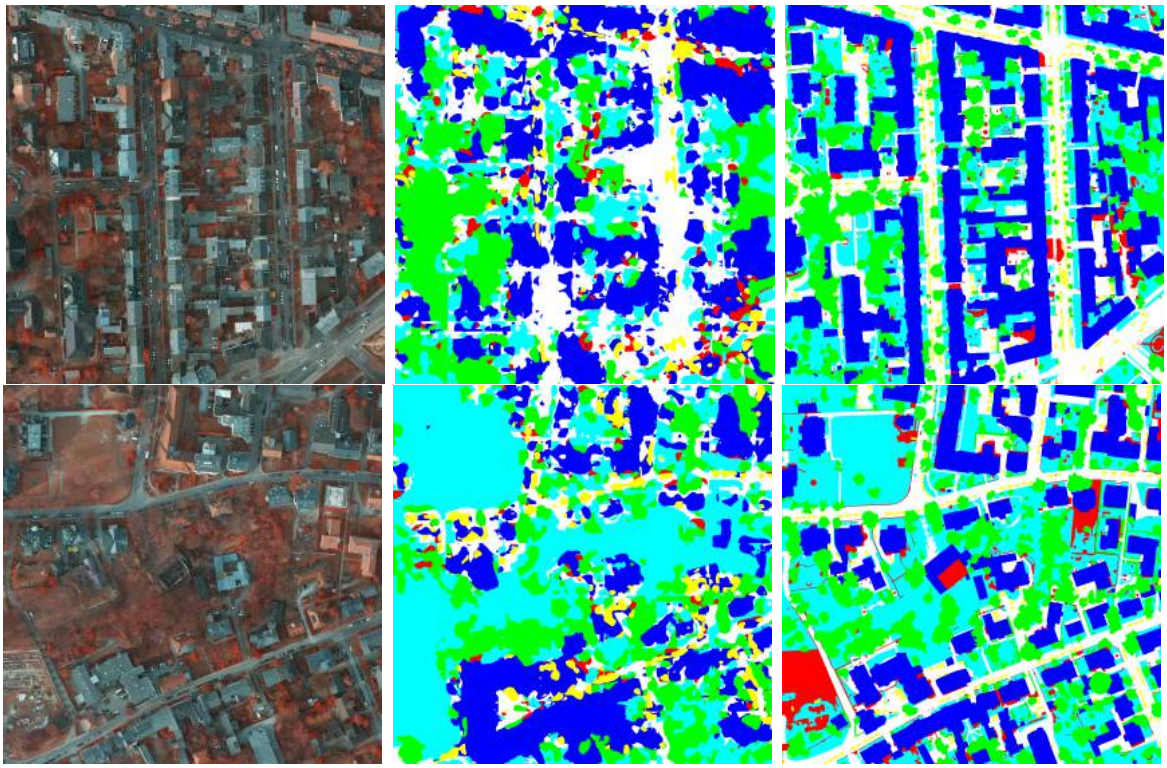

Figure 6: Selected examples of the segmentation results of our method on ISPRS Potsdam dataset.

Category	Cross-domain	In-domain
building	73.1	77.4
imp. surface	62.6	72.0

Table 3: In-domain vs. Cross-domain segmentation performance on building and impervious classes of ISPRS Vaihingen dataset.

5. DISCUSSION

5.1 Network Depth

We first explore the effect of different backbone feature extraction networks for prototype learning. We tried two different backbone networks, i.e., VGG-Net and ResNet. The VGG-Net consists of a successive of convolutional layers and fully connected layers, while the ResNet uses deeper network architecture and leverage residual connection to enable more efficient feature propagation. In various computer vision tasks, ResNet has been proved to be more powerful and robust for image feature learning. Table 4 compares the segmentation performance of our model with these two different backbone networks. To compare the performance, we report both overall accuracy and Kappa coefficient for all comparing methods. As shown in this Table 4, our model with ResNet backbone achieves better performance on the ISPRS Vaihingen dataset, with an improvement of 2.1% on the overall accuracy. While for the Potsdam dataset, using a deeper backbone network hurts the performance. The overall accuracy produced by VGG-Net (52.5%) decreases to 48.7% by using ResNet backbone. In the following experiments, we use ResNet and VGG-Net for feature extraction and prototype learning on Vainhingen and Potsdam dataset respectively.

We also give some qualitative comparison of these two backbone networks for the task of few-shot segmentation on the ISPRS Vaihingen dataset. As shown in Fig. 7, our model with the VGG-Net backbone fails to detect some building areas while our model with ResNet backbone is able to successfully label these areas.

dataset	Vaihingen		Potsdam	
Method	Acc.	Kappa	Acc.	Kappa
VGG-init	58.8	46.9	50.0	35.7
VGG-Net	60.0	47.6	52.5	39.4
ResNet50	62.1	49.6	48.7	35.2

Table 4: Segmentation performance with different network depths.

5.2 Effect of DenseCRF

In this section, we investigate the effect of DenseCRF on segmentation refinement. As we mentioned in section 3.6, we explore the influence of two different strategies of applying DenseCRF. Table 5 list the performance of our model with and without DenseCRF for segmentation refinement. We also give an example of our segmentation results with and without DenseCRF in Figure 8. As shown in Table 5, our model achieves significantly better performance on the ISPRS Vaihingen dataset by applying DenseCRF post-processing. More specifically, our method with DenseCRF(v1) and DenseCRF(v2) improves the overall accuracy by 3.8% and 6.6% respectively. By comparing our prediction with and without DenseCRF in Figure 8, one can find that our original prediction (w/o DenseCRF) leads to inconsistent predictions, e.g., the pixels of the same building instance have different predicted labels. By enforcing spatial continuity using DenseCRF, our model successfully corrects these unwanted inconsistent predictions and smooth the segmentation maps. Moreover, by comparing DenseCRF(v1) and DenseCRF(v2), one can find out that DenseCRF(v2) performs significantly better than DenseCRF(v1). This is because DenseCRF(v1) applies DenseCRF to each patch prediction independently. In this way, the segmentation maps may get broken along the patch boundaries, as illustrated in the marked areas in Fig. 8. In contrast, our model with DenseCRF(v2) works better since it first aggregates all patches into a full segmentation map and applies DenseCRF to the entire map. Thus, the labeling consistency is naturally guaranteed

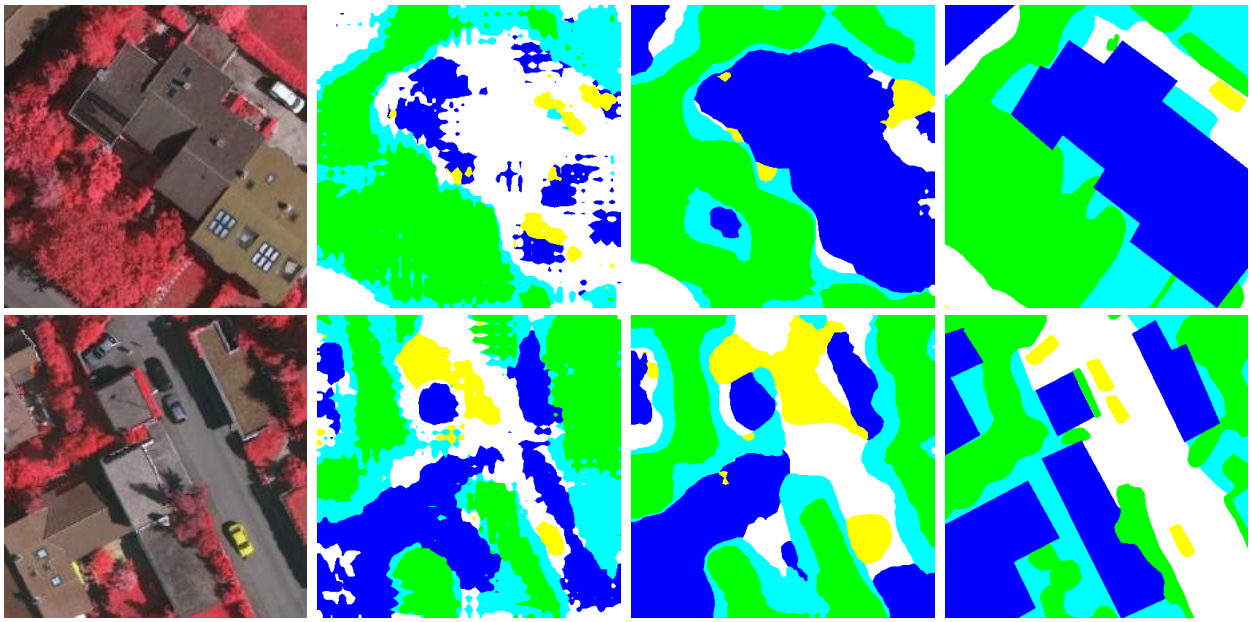

Figure 7: Segmentation results with different network depths. From left to right: aerial image, our prediction with VGG-Net, our prediction with ResNet, ground truth.

dataset	Vaihingen		Potsdam	
Method	Acc.	Kappa	Acc.	Kappa
Ours	62.1	49.6	52.5	39.4
Ours w DenseCRF(v1)	64.9	52.8	52.3	38.9
Ours w DenseCRF(v2)	67.7	56.3	52.2	38.3

Table 5: Segmentation performance with and without DenseCRF.

within the whole, aggregated segmentation map. As for the Potsdam dataset, applying DenseCRF does not improve the segmentation performance. The overall accuracy decreases slightly for both DenseCRF(v1) and DenseCRF(v2).

5.3 Number of samples

In this section, we test the effect of using different numbers of support labels for our few-shot segmentation task. We conduct experiments on both ISPRS Vaihingen and Potsdam dataset with 1/5/10 support labels. Note that when selecting the support images, we make sure that the selected images cover all classes. We list the performance under different configurations in Table 6. From Table 6 we can see that the segmentation performance of our model improves consistently as more annotated images are provided. Especially, our model achieves a satisfying segmentation accuracy of 72.4% on ISPRS Vaihingen with 10 label patches. The results on the Potsdam dataset also enjoy a performance boost when using more support labels.

Moreover, to further demonstrate the superiority of our proposed model over traditional supervised models, we conduct an experiment by training a SegNet model with only 1/5/10 randomly selected patches and evaluate the performance on the same validation set. Note that all experimental settings are the same as (Audebert et al., 2018) and we also make sure that the selected images cover all classes. The segmentation performances are listed in Table 6. From Table 6 we can see that with only a few support labels, on both the Vaihingen and Potsdam dataset, the SegNet model gets a performance far worse than our model, which use the same support labels, especially when only 1 or 5 support labels are provided. This finding is even more significant on the Potsdam dataset, where the SegNet model gets an overall accuracy 11% less than our model when only 1 support label is provided.

	Vaihingen		Potsdam	
#Samples	Acc.	Kappa	Acc.	Kappa
SegNet (1)	49.2	30.3	34.0	10.8
SegNet (5)	60.9	50.4	43.8	25.8
SegNet (10)	71.3	61.1	50.6	36.9
Ours (1)	56.9	42.2	45.0	28.5
Ours (5)	67.7	56.3	52.2	38.3
Ours (10)	72.4	62.9	52.7	40.5

Table 6: Segmentation performance with different number of support images. For a fair comparison, the SegNet (Audebert et al., 2018) here is trained with only 10 randomly selected patches.

5.4 Weak Annotations

In the above experiments, we have demonstrated the powerful generalization abilities of our model with only a few annotated labels. One direct question is that can we further reduce the annotation efforts by replacing the dense pixel-wise annotations with weak annotations. By looking deep into our model architecture, one can see that in the testing state, our model only requires a labeled mask for each semantic class. Therefore, our model can be directly applied to weak annotations by transferring them to masked label maps.

In this section, we investigate the performance of our few-shot segmentation method with two types of weak annotations, including bounding box annotations and scribble annotations. Compared to drawing pixel-level semantic segmentation masks, it would drastically reduce the annotation efforts for human annotators when drawing bounding boxes or scribbles.

In our bounding box weak annotation experiment, instead of having every shot in the support set annotated on the pixel level, the annotations will consist of a single bounding box around the largest object of a certain class. Specifically, for each semantic class, we extract the contours of all objects of this class from the original semantic maps and select the bounding box around the contour with the largest area as our support label. Figure 9 gives an example of our bounding box annotation for the class "tree".

In the scribble weak annotation experiment, we replace the ground truth semantic masks of the support images with scribble masks.

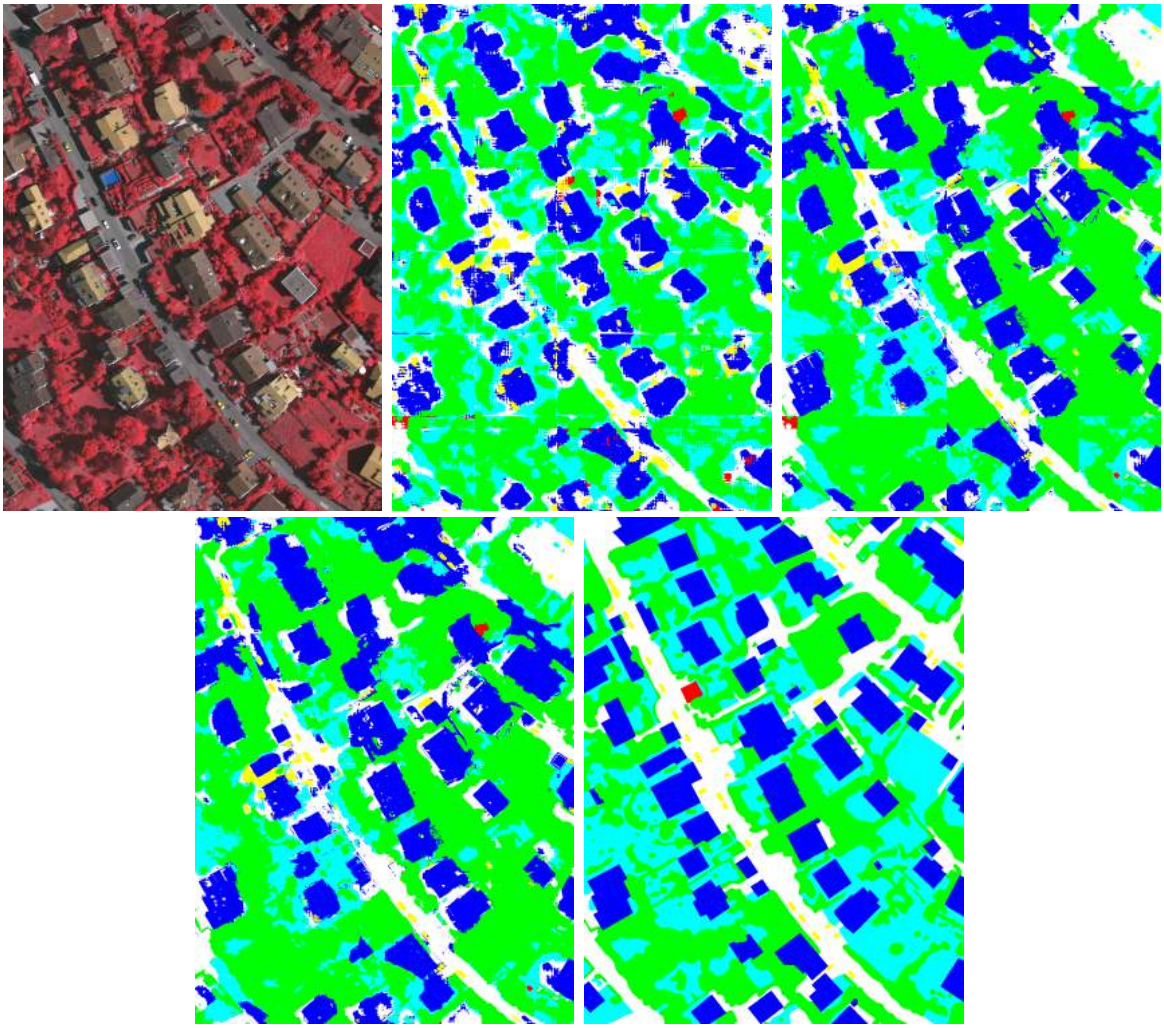

Figure 8: Segmentation results. From left to right: aerial image, our original prediction, our model with DenseCRF(v1), our model with DenseCRF(v2), ground truth.

Figure 9: An example of bounding box annotation. From left to right: aerial image, semantic map, bounding box annotation.

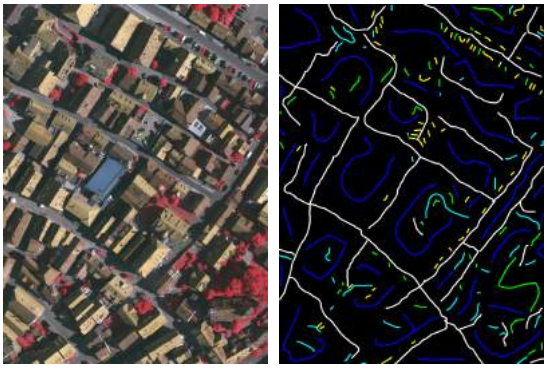

Figure 10: An example of scribble annotation. From left to right: aerial image, scribble annotation.

The scribble masks are created by drawing 10-pixel-wide lines over objects in the support set. Each line has the same semantic label as the original semantic maps they draw over. Figure 10 gives an example of our scribble annotation for the low vegetation class. Note that scribble annotations are more time-consuming compared to bounding box annotations, but still more efficient compared to dense pixel-wise annotations.

To evaluate the performance of our model under weak annotations, two additional experiments were conducted on the ISPRS Vaihingen dataset with bounding box annotations and scribble annotations. Experimental setting are the same as Section 4.2.3. The final segmentation performance are listed in Table 7.

From table 7, we can see that using bounding boxes annotations of only 5 support images, our model can still maintain a reasonable segmentation performance of 60.3% on the overall accuracy. One interesting finding is that when using "scribble" annotations, our model achieves a segmentation performance even slightly better than that of using the original pixel-wise annotations (69.3% vs. 68.6%). This is probably because when using original label maps as the support labels, the prototypes are learned over all pixels of a certain class. This may involve unwanted noise pixels and the learned prototypes are less compact. While the scribble annotations only use pixels around the center area of each object, which makes the learned prototypes more compact. This finding is important because it suggests that scribbles may be the best way to annotate support images for our and other similar approaches to few-shot segmentation because drawing scribbles takes much less effort than drawing pixel-wise labels but offers an almost identical level of segmentation accuracy.

Method	Acc.	Kappa
Pixel-wise labels	67.7	56.3
Bounding box	60.3	46.1
Scribble	68.2	58.2

Table 7: Segmentation performance on Vaihingen dataset with different type of annotations.

6. CONCLUSIONS

In this paper, we proposed probably the first few-shot learning-based method for semantic segmentation of remote sensing images. Our method can perform segmentation for unseen object categories with only a few annotated samples. More specifically, our model starts with a deep CNN to extract high-level semantic features. Then, the prototype representation of each class is generated by a masked average pooling over feature maps by leveraging the ground truth masks of support images. Finally, our model performs semantic labeling over the query images by matching the feature embedding of each pixel to its closest prototype. Our model is optimized with a non-parametric metric learning-based loss function in order to maximize the intra-class similarity of learned prototypes while minimizing the inter-class similarity. We conduct both in-domain and cross-domain experiments to demonstrate the generalization abilities of our model on unseen categories. In the in-domain experiments, our model is trained on some of the object classes in ISPRS 2D semantic labeling dataset and evaluated on a new class of the same dataset with few annotated samples. In the cross-domain experiments, our model is trained on PASCAL VOC 2012 dataset, and evaluated on ISPRS 2D semantic labeling dataset with few annotated samples. Experiments demonstrate satisfying in-domain and cross-domain transferring abilities of our model. More specifically, with only 5 annotated image patches, our model gets satisfying performance on the ISPRS 2D Semantic Labeling Challenge dataset, with an overall accuracy of 67.7% and 52.8% for Vaihingen and Potsdam dataset respectively. We also show that our model is able to achieve a similar level of few-shot segmentation accuracy of new object classes with only weak annotations, such as bounding boxes or scribbles.

ACKNOWLEDGEMENTS

We would like to gratefully acknowledge the ISPRS for providing the experimental dataset.

References

Alshehhi, R., Marpu, P. R., Woon, W. L. and Dalla Mura, M., 2017. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. ISPRS Journal of Photogrammetry and Remote Sensing 130, pp. 139–149.

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

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

Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39(12), pp. 2481–2495.

Castelluccio, M., Poggi, G., Sansone, C. and Verdoliva, L., 2015. Land use classification in remote sensing images by convolutional neural networks. arXiv preprint arXiv:1508.00092.

Chen, H., Wang, Y., Wang, G. and Qiao, Y., 2018a. Lstd: A low-shot transfer detector for object detection. In: Thirty-Second AAAI Conference on Artificial Intelligence.

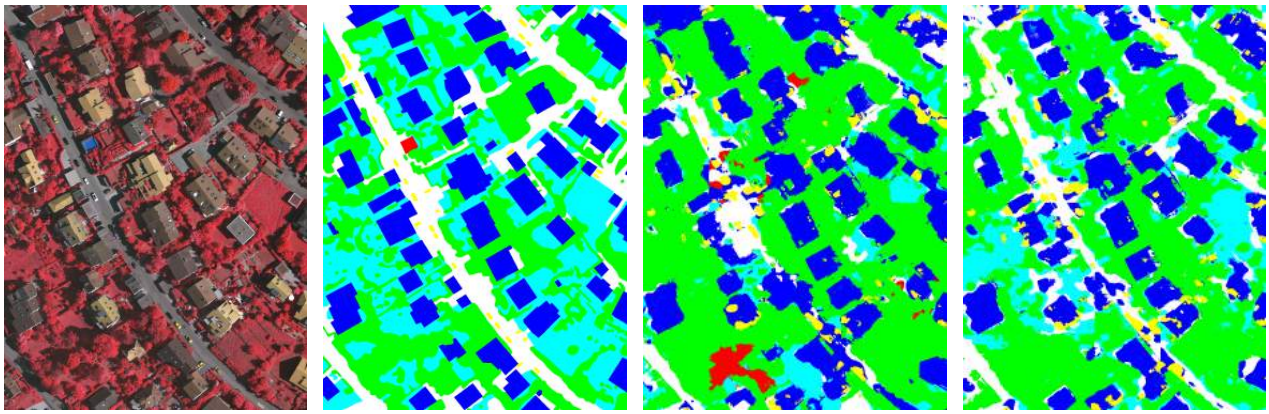

Figure 11: Few-shot segmentation results using weak annotations. From left to right: aerial image, ground truth, result using bounding box annotation, result using scribble annotation.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L., 2014a. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence 40(4), pp. 834–848.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H., 2018b. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV), pp. 801–818.

Chen, X., Xiang, S., Liu, C.-L. and Pan, C.-H., 2014b. Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geoscience and remote sensing letters 11(10), pp. 1797–1801.

Cheng, G., Yang, C., Yao, X., Guo, L. and Han, J., 2018. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. IEEE transactions on geoscience and remote sensing 56(5), pp. 2811–2821.

Dai, J., He, K. and Sun, J., 2015. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1635–1643.

Dong, N. and Xing, E., 2018. Few-shot semantic segmentation with prototype learning. In: BMVC, Vol. 1, p. 6.

Finn, C., Abbeel, P. and Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, pp. 1126–1135.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hu, Y., Li, X., Zhou, N., Yang, L., Peng, L. and Xiao, S., 2019. A sample update-based convolutional neural network framework for object detection in large-area remote sensing images. IEEE Geoscience and Remote Sensing Letters 16(6), pp. 947–951.

Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J. and Darrell, T., 2019. Few-shot object detection via feature reweighting. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 8420–8429.

Krähenbühl, P. and Koltun, V., 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In: Advances in neural information processing systems, pp. 109–117.

Li, X., Yao, X. and Fang, Y., 2018. Building-a-nets: Robust building extraction from high-resolution remote sensing images with adversarial networks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (99), pp. 1–8.

Lin, D., Dai, J., Jia, J., He, K. and Sun, J., 2016. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3159–3167.

Lin, G., Milan, A., Shen, C. and Reid, I. D., 2017. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Cvpr, Vol. 1number 2, p. 5.

Liu, Y., Piramanayagam, S., Monteiro, S. T. and Saber, E., 2017. Dense semantic labeling of very-high-resolution aerial imagery and lidar with fully-convolutional neural networks and higher-order crfs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 76–85.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

Luus, F. P., Salmon, B. P., Van den Bergh, F. and Maharaj, B. T. J., 2015. Multiview deep learning for land-use classification. IEEE Geoscience and Remote Sensing Letters 12(12), pp. 2448–2452.

Maggiori, E., Tarabalka, Y., Charpiat, G. and Alliez, P., 2016. Convolutional neural networks for large-scale remote-sensing image classification. IEEE Transactions on Geoscience and Remote Sensing 55(2), pp. 645–657.

Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S., Datcu, M. and Stilla, U., 2018. Classification with an edge: improving semantic image segmentation with boundary detection. ISPRS Journal of Photogrammetry and Remote Sensing 135, pp. 158–172.

Marmanis, D., Wegner, J. D., Galliani, S., Schindler, K., Datcu, M. and Stilla, U., 2016. Semantic segmentation of aerial images with an ensemble of cnns. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 3, pp. 473.

Mnih, V., 2013. Machine learning for aerial image labeling. PhD thesis, University of Toronto (Canada).

Mnih, V. and Hinton, G. E., 2010. Learning to detect roads in high-resolution aerial images. In: European Conference on Computer Vision, Springer, pp. 210–223.

Oreshkin, B., López, P. R. and Lacoste, A., 2018. Tadam: Task dependent adaptive metric for improved few-shot learning. In: Advances in Neural Information Processing Systems, pp. 721–731.

Papandreou, G., Chen, L.-C., Murphy, K. P. and Yuille, A. L., 2015. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: Proceedings of the IEEE international conference on computer vision, pp. 1742–1750.

Rakelly, K., Shelhamer, E., Darrell, T., Efros, A. and Levine, S., 2018. Conditional networks for few-shot semantic segmentation.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, Springer, pp. 234–241.

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

Saito, S., Yamashita, T. and Aoki, Y., 2016. Multiple object extraction from aerial imagery with convolutional neural networks. Electronic Imaging 2016(10), pp. 1–9.

Shaban, A., Bansal, S., Liu, Z., Essa, I. and Boots, B., 2017. One-shot learning for semantic segmentation. arXiv preprint arXiv:1709.03410.

Sherrah, J., 2016. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv preprint arXiv:1606.02585.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Snell, J., Swersky, K. and Zemel, R., 2017. Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, pp. 4077–4087.

Vakalopoulou, M., Karantzalos, K., Komodakis, N. and Paragios, N., 2015. Building detection in very high resolution multispectral data with deep learning features. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, pp. 1873–1876.

Wang, K., Liew, J. H., Zou, Y., Zhou, D. and Feng, J., 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 9197–9206.

Wen, C., Yang, L., Peng, L., Li, X. and Chi, T., 2019. Directionally constrained fully convolutional neural network for airborne lidar point cloud classification. arXiv preprint arXiv:1908.06673.

Yang, Z., Jiang, W., Xu, B., Zhu, Q., Jiang, S. and Huang, W., 2017. A convolutional neural network-based 3d semantic labeling method for als point clouds. Remote Sensing 9(9), pp. 936.

Yu, F. and Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Zhang, C., Lin, G., Liu, F., Yao, R. and Shen, C., 2019. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5217–5226.

Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J., 2017. Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2881–2890.

Zhao, R., Pang, M. and Wang, J., 2018. Classifying airborne lidar point clouds via deep features learned by a multi-scale convolutional neural network. International Journal of Geographical Information Science 32(5), pp. 960–979.

Zou, Q., Ni, L., Zhang, T. and Wang, Q., 2015. Deep learning based feature selection for remote sensing scene classification. IEEE Geoscience and Remote Sensing Letters 12(11), pp. 2321–2325.