remote sensing

MDPI

Article

Semantic Labeling of High Resolution Aerial Imagery and LiDAR Data with Fine Segmentation Network

Xuran Pan [1,2], Lianru Gao [2,*], Andrea Marinoni [3], Bing Zhang [2,4], Fan Yang [1] and Paolo Gamba [3]

1 Tianjin Key Laboratory of Electronic Materials and Devices, School of Electronics and Information Engineering, Hebei University of Technology, Tianjin 300401, China; 201611901006@stu.hebut.edu.cn (X.P.); 201621901026@stu.hebut.edu.cn (F.Y.)
2 Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China; zb@radi.ac.cn
3 Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy; andrea.marinoni@unipv.it (A.M.); paolo.gamba@unipv.it (P.G.)
4 College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: gaolr@radi.ac.cn; Tel.: +86-10-8217-8172; Fax: +86-10-8217-8009

Received: 8 April 2018; Accepted: 9 May 2018; Published: 11 May 2018

Abstract: In this paper, a novel convolutional neural network (CNN)-based architecture, named fine segmentation network (FSN), is proposed for semantic segmentation of high resolution aerial images and light detection and ranging (LiDAR) data. The proposed architecture follows the encoder–decoder paradigm and the multi-sensor fusion is accomplished in the feature-level using multi-layer perceptron (MLP). The encoder consists of two parts: the main encoder based on the convolutional layers of Vgg-16 network for color-infrared images and a lightweight branch for LiDAR data. In the decoder stage, to adaptively upscale the coarse outputs from encoder, the Sub-Pixel convolution layers replace the transposed convolutional layers or other common up-sampling layers. Based on this design, the features from different stages and sensors are integrated for a MLP-based high-level learning. In the training phase, transfer learning is employed to infer the features learned from generic dataset to remote sensing data. The proposed FSN is evaluated by using the International Society for Photogrammetry and Remote Sensing (ISPRS) Potsdam and Vaihingen 2D Semantic Labeling datasets. Experimental results demonstrate that the proposed framework can bring considerable improvement to other related networks.

Keywords: high resolution aerial imagery; LiDAR; spectral image; semantic segmentation; deep learning; convolutional neural network (CNN)

1. Introduction

Semantic segmentation of high resolution remote sensing images aims at assigning each pixel a certain semantic class, for instance, building, car, tree, or low vegetation. Accurate and timely acquisition of segmentation results is fundamental for precise urban planning, environmental monitoring and economic forecasting. With the development of aerospace remote sensing technology, the spatial resolution of remote sensing images has notably increased. Higher spatial resolution brings a lot of tiny objects and fine details, but also causes large intra-class variance and small inter-class differences, which often leads to segmentation ambiguities [1].

Several approaches based on spectral statistical features have been proposed for high resolution remote sensing image classification, including maximum likelihood method [2], minimum distance method [3] and K-means [4]. Moreover, methods based on machine learning such as neural networks

(NN) [5] and support vector machines (SVM) [6] have been developed for this task as well. Finally, new models based on object-oriented classification [7] and sparse representation [8] have been developed too. Although these frameworks might obtain satisfactory classification performance, they typically suffer of major drawbacks which can jeopardize the processing outcomes. Indeed, when scenarios characterized by high spectral complexity are taken into account, these architectures are not able to accurately track the interplay among samples on global and local scale. Furthermore, these shallow learning networks cannot satisfy the requirements for the complexity and diversity of functions and training samples because they usually have only one hidden layer.

Over the last few years, deep learning has received widespread attention in the image analysis field [9]. Convolutional neural networks (CNNs) play a key-role within deep learning techniques. They have achieved remarkable results in several applications, especially for recognition and bounding box object detection [10,11]. Semantic segmentation is a task that has higher requirements than object detection. Naturally, semantic segmentation attracts more research attention in the deep learning field as a progression from coarse to fine inference. There are several available CNNs for semantic segmentation which can be divided into two categories: patch-based methods and pixel-based methods. When patch-based methods are considered, the image patches around each pixel from the input image are extracted, and a single label for each patch with CNN for whole-image classification is predicted [12–16]. This class of algorithms provides a remarkable enhancement in image segmentation: however, it also shows drawbacks, such as huge RAM cost, low computational efficiency, and limited receptive field.

On the other hand, pixel-based methods can predict labels for all the pixels of whole image at a time. The most classic architecture is fully convolutional network (FCN) proposed by Long et al. [17], which replaces fully connected layers with convolutional layers, and implements transposed convolutional layers to upscale the coarse outputs into fine segmentation results. Following the encoder–decoder paradigm of FCN, multiple CNN architectures have been developed and achieved better results. To mitigate the classification ambiguities caused by large factor upsample in FCN, Chen et al. [18] proposed "DeepLab" with a fully connected Conditional Random Field (DeepLab-CRF), which introduced "atrous" convolutions to avoid the effect of removing pooling layer and integrated the responses at the final CNN layer with a fully connected CRF to smooth the raw segmentation results. Noh et al. [19] proposed to replace bilinear interpolation by a multi-layer deconvolution network which is composed of deconvolution and unpooling layers in the upscaling stage (DeconvNet). Badrinarayanan et al. [20] presented a novel CNN architecture named SegNet, which shares a similar architecture as DeconvNet but with much smaller parameterization, and is easier to be trained end-to-end.

In recent years, CNNs were gradually applied into semantic segmentation for remote sensing images. In the patch-based methods category, Paisitkriangkrai et al. [21] suggested to integrate CNN features with handcraft features, and utilized CRFs to further improve the classification performance. Nogueira et al. [22] compared several existing powerful CNNs with three train strategies, which are training from scratch, fine-tuning, and using CNNs as feature extractor. Experimental results on three remote sensing datasets indicated that the fine-tuning is the best strategy. In [1], a FCN architecture without downsampling was proposed to mitigate feature detail loss. The architecture had a pre-trained network for color-infrared (CIR) images and a network trained from scratch for digital surface model data. The features of these two networks were then combined by concatenation. Although this architecture achieved state-of-the-art semantic labeling accuracy for high-resolution aerial imagery, it soon was outperformed by downsample- then-upsample architectures for its limited receptive field. Moreover, its decision-level fusion strategy is bound to train two separate neural networks for CIR images and LiDAR data respectively, which means the number of trainable parameters is doubled. Audebert et al. [23] transferred a powerful semantic segmentation architecture from generic images (SegNet [20]) to remote sensing images. They compared both decision-level and feature-level fusion methods, and proved their proposed dual stream SegNet to fuse multi-sensor data

by residual correction in the feature level performed slightly better. Our FSN also applied feature-level fusion, but ours differs from their method in two ways. First, we processed LiDAR data by a lighter weighed branch, which guarantees the reduction of the computational overhead. Moreover, since CIR data contain more information than LiDAR data, we concatenate features from main encoder and lightweight branch in different depth while they concatenate the features of heterogeneous data in the same depth. In [24], Liu et al. fused a FCN trained on CIR images and a logistic regression trained on light detection and ranging (LiDAR) data in the decision level by a higher-order CRF framework, which outperformed the original counterparts. In [25], Volpi et al. applied learnable transpose convolutional layers in the decoder stage to decrease the spatial information loss, but the segmentation accuracy was still limited. In [26], Maggiori et al. analyzed some dense semantic labeling CNNs of high-resolution remote sensing imagery deeply, and derived a CNN framework with a multi-layer perceptron to learn to combine features at different resolutions. In [27], an hourglass-shape network was designed followed by a downsample-then-upsample paradigm, which introduced inception module to take advantage of multi-scale receptive fields, together with residual units to feed information from encoder to decoder directly. However, the methods in [25–27] did not pay much attention to multi-sensor data fusion problem. They fused CIR images and the LiDAR data in the first layer, which usually cannot exploit the features of each data and cannot obtain satisfying results.

In summary, it can be concluded that pixel-based methods achieve better performance than patch-based methods, and training from scratch is often outperformed by fine-tuning a pre-trained network. Although deep learning has achieved a solid success in semantic segmentation for remote sensing images, the well-known trade-off between recognition and localization [18,19] remains a challenging endeavor. Down-sampling operations give the network wider vision to produce more accurate recognition, but the small sized feature maps may lead to inaccurate location. Down-sampling operation can offer the network wider receptive field to obtain more accurate recognition, but the smaller feature maps may result in inaccurate localization. Besides, the existing upscale methods such as transposed convolutional layer and unpooling layer tend to bring artificial values in the low resolution feature maps and then upscale these low resolution feature maps. This can cause an information loss, and therefore decrease the segmentation accuracy. Moreover, multi-sensor data often are fused at the decision-level or are stacked together as input. The former approach suffers from the large number of parameters, and the latter makes the network unable to be initialized by pre-trained weights which is proved to be superior to random initialization [22–24].

In this paper, a new CNN-based architecture named fine segmentation network (FSN) is proposed for semantic segmentation of high resolution aerial imagery. The FSN belongs to pixel-based methods category and follows the encoder–decoder paradigm as FCN; moreover, it fuses the multi-sensor data in feature-level by multi-layer perceptron (MLP). The main contributions of the proposed network can be summarized as follows:

- The encoder is structured into two parts: a main encoder and a lightweight branch. The main encoder is based on the Vgg16 [28] for CIR images. The lightweight branch is designed to deal with its corresponding LiDAR images: the digital surface models (DSMs) and the normalized DSMs (nDSMs) independently. This design accomplishes the feature extraction of multi-sensor data with a relatively few parameters.
- Sub-pixel convolution layers proposed for image and video super-resolution [29] are implemented to replace the traditional deconvolution layers in the proposed FSN. Without adding any artificial value, sub-pixel convolution layer calculates convolutions in low resolution feature maps and upscales them in a single step. Thus, the contextual area can be expanded by a filter of the same size as that of common up-sampling layer.
- MLP is used to accomplish effective feature-level fusion of multi-sensor remote sensing data at the back end of the structure. Moreover, multi-resolution feature maps are also fed into MLP to mitigate the recognition/localization trade-off.

The proposed FSN is evaluated on the International Society for Photogrammetry and Remote Sensing (ISPRS) Potsdam and Vaihingen 2D Semantic Labeling datasets. Experimental results demonstrate that it can bring considerable improvement to other related networks.

The reminder of this paper is organized as follows. In Section 2, a brief introduction for the main component of convolutional neural network is presented. In Section 3, the proposed FSN is detailed at the beginning, followed by a presentation for the post-processing method. Section 4 presents data and experiment settings, and experimental results. Section 5 draws the conclusions.

2. Convolutional Neural Network

Convolutional neural network [30] is a special version of deep neural network which is characterized by sparse connectivity and parameter sharing. Sparse connectivity means the neurons of CNNs are not fully connected, as shown in Figure 1a; indeed, each two layers are partially connected to make better use of local spatial characteristics. Parameter sharing means neurons in one feature map share a same parameters matrix. As shown in Figure 1b, layer n has four neurons belonging to one feature mapping, and connection lines of the same color identify the same weights. These two characteristics reduce the complexity of the network structure and the total amount of parameters.

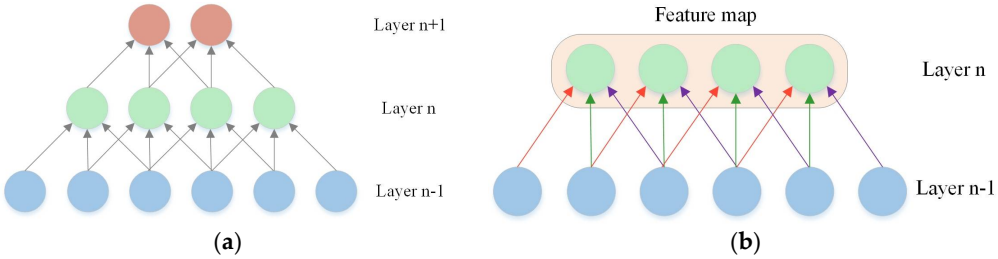


Figure 1. Illustration of sparse connectivity and parameter sharing for CNN: (**a**) sparse connectivity; and (**b**) parameter sharing.

CNNs can also be considered as a complex non-linear function to turn the inputs into target variables. In this section, we briefly present an introduction of CNNs' main composition elements. The basic type of CNN layers in semantic segmentation include: convolutional layer, non-linear activation layer, spatial pooling layer, transposed convolutional layer, and unpooling layer.

2.1. Convolutional Layer

The convolutional layers are the main component of CNNs. A convolutional layer can be considered as a set of neurons or filters. Each filter has a series of learnable parameters which are arranged as a convolution kernel with size $P \times Q \times D$, where P, Q and D represent length, width and depth of convolution kernel, respectively. The conversion of input image with size $N_i \times L_i \times D$ to output of convoltional layer with size $N_o \times L_o \times D'$ is performed by convolutional layer (a set of D' filters). When the filter is centered on the spatial position (i,j) of the input, and the response for the d'-th filter can be written as:

$$y_{ijd'} = \sum_{d=1}^{D} \sum_{q=1}^{Q} \sum_{p=1}^{P} W_{pqd} \cdot x_{pqd} + b \tag{1}$$

where $y_{ijd'}$ is the response for d'-th filter, x_{pqd} is tnhe window surrounding spatial position (i,j) of the input, W_{pqd} is the learnable weights of convolution kernel, and b is a learnable bias. The spatial dimensions of the output can be calculated as:

$$N_o = \frac{N_i - P + 2Z}{S} + 1 \tag{2}$$

$$L_o = \frac{N_i - Q + 2Z}{S} + 1 \tag{3}$$

where Z is the number of rows and columns padded on the borders of input, and S is the stride of convolution kernel sliding. Figure 2 reports an example where input image is of size $7 \times 7 \times 3$, and the size of convolution kernel is $3 \times 3 \times 3$. Moreover, the convolution is performed with stride 1 without padding, and only one filter is employed. Then, the size of output is $5 \times 5 \times 1$.

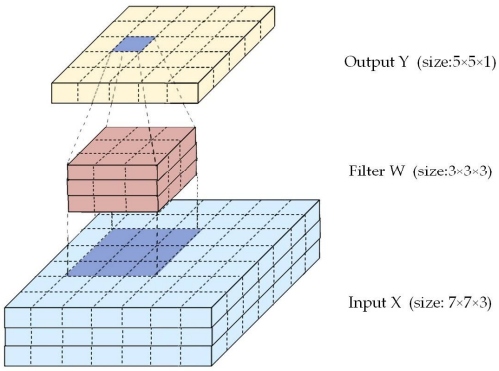


Figure 2. Illustration of convolutional layer.

2.2. Nonlinear Activation Layer

Generally, neural network for image process aims to perform convolutions to images, but this operation is obviously linear. Thus, nonlinear activation layer is introduced into CNNs to increase the network's ability to express any complex non-linear mapping. The most common nonlinear activation applied in CNNs is the Rectified Linear Unit (ReLU) function [31], formulated as:

$$f(x) = \max(0, x) \tag{4}$$

Other common activation functions include Sigmoid function, Tanh function, and leaky ReLU function [32].

2.3. Spatial Pooling Layer

The role of spatial pooling layer is to reduce the dimensionality of the representation and create an invariance of small shifts and distortions by pooling over small windows into single values [33]. Spatial pooling operation often utilizes small windows (e.g., 2×2 or 3×3), to slide over the feature maps, and convert the information within the window into a single value. The existing pooling methods include max pooling, mean pooling, and sum pooling functions. In this paper, the max pooling function is taken into consideration due to its stability and wide application in the literature. Given the size of window as $P \times P$, P_{ij} denotes the window centered on the spatial location of (i,j). Then, max pooling function returns the maximum value of the window area as:

$$y_{ij} = \max_{a \in P_{ij}} x_a \tag{5}$$

2.4. Transposed Convolutional Layer

Transposed convolution is also called deconvolution [34]. In semantic segmentation area, transpose convolution is a popular approach to recover the lost feature details caused by pooling operations or other downsampling operations. The low resolution input is first up-scaled by using bilinear interpolation or adding zeros, and then the convolutional operations are employed on the raw up-scaled results to fit in sensible values.

2.5. Unpooling Layer

Unpooling, is the reverse operation of pooling. Pooling is an irreversible operation; therefore, the position of the max value is recorded in the pooling stage (take max pooling as example). In the unpooling stage, only the value of the recorded position is activated, whilst the value of other position is set to 0.

3. Proposed FSN Method

In this section, we detail the proposed FSN method for semantic segmentation of high resolution aerial imagery. We first present the general network design of FSN in Section 3.1, and then the post-processing method is introduced in Section 3.2.

3.1. Network Architecture of FSN

The proposed FSN belongs to pixel-based methods, and it is designed in the encoder–decoder structure, as shown in Figure 3. The encoder consists of two parts: the main encoder part for color-infrared images and the lightweight branch for DSMs data. The main encoder is based on the first 13 layers of Vgg16 net, and each convolutional layer is followed by a ReLU activation. Five pooling layers are employed to downsample the feature maps to achieve wider receptive fields.

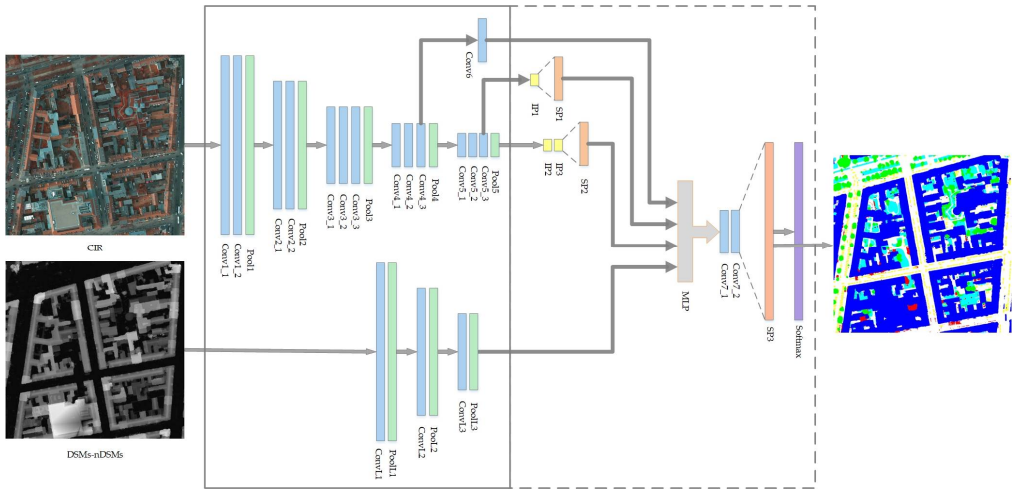


Figure 3. Network architecture of FSN, where structure depicted in the solid-line box is encoder and structure depicted in the dashed-line box is decoder.

To obtain accurate fine resolution segmentation, we also consider DSM and nDSM data. These additional records characterize the height of the ground objects, thus can help the network to recognize buildings and trees. When these additional data are concatenated with CIR images as input of the main encoder, the pre-trained weights cannot be utilized to initialize the network anymore, i.e., the network cannot be trained by fine-tuning. Furthermore, although the recognition of buildings and trees can take advantage of DSM/nDSM data, the analysis based on other objects can be eventually degraded by their signal distribution for LiDAR data have no useful information about these objects. Therefore, an extra branch is designed to extract features from these LiDAR data independently, and the combination with the features of CIR images is done by MLP in an efficient and flexible manner [26]. Since LiDAR data have no spectral information and the value of each pixel represents the height degree of current pixel, these additional data to some extent can be regarded as the probability graph of buildings and trees, whilst each pixel of heat maps represents the probability that the pixel belongs to the specific class, which makes LiDAR data close to the score maps from upper layer of the decoder. Therefore, a lightweight convolutional network can generate the appropriate features to complement with the preliminarily upscaled features from the upper layer of the main

encoder. Hence, we designed a structure composed by three convolutional layers (the first two are followed by ReLU activation). Moreover, three maxpooling layers are employed to downsample the feature maps. The detailed configurations of encoder are listed in Table 1.

Table 1. Configurations of lightweight branch.

Layer	Filter Size	Number of Filters	Stride	Padding	Layer	Filter Size	Number of Filters	Stride	Padding
Conv1_1	3	64	1	1	Conv4_3	3	512	1	1
Conv1_2	3	64	1	1	Pool4	2		2	
Pool1	2		2		Conv5_1	3	512	1	1
Conv2_1	3	128	1	1	Conv5_2	3	512	1	1
Conv2_2	3	128	1	1	Conv5_3	3	512	1	1
Pool2	2		2		Pool5	2		2	
Conv3_1	3	256	1	1	ConvL1	3	64	1	1
Conv3_2	3	256	1	1	PoolL1	2		2	
Conv3_3	3	256	1	1	ConvL2	3	128	1	1
Pool3	2		2		PoolL2	2		2	
Conv4_1	3	512	1	1	ConvL3	3	16	1	1
Conv4_2	3	512	1	1	PoolL3	2		2	

In the decoder stage, to enrich the receptive fields and contextual information of FSN, we introduce into our architecture the inception modules (the yellow blocks in Figure 3), which consist of convolution layers with multiple sized kernels [27,35] (see Figure 4).

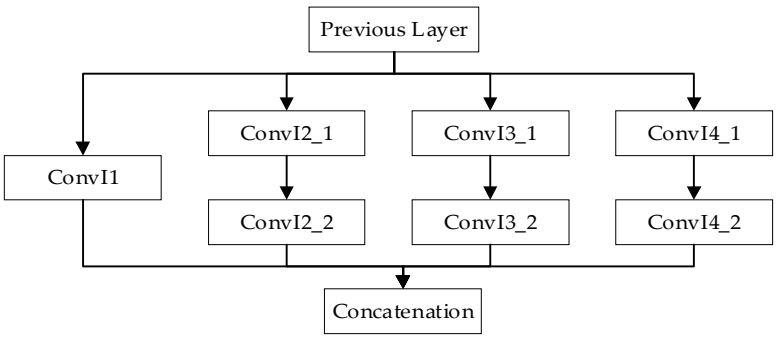


Figure 4. Architecture of inception module.

We also introduce a powerful super resolution module "sub-pixel convolution layer" [29] (orange blocks in Figure 3) into our design to perform upsampling in the decoder stage. Contrary to previous upscaling methods, sub-pixel convolution increases the resolution after convolutional operations, therefore convolutions with a smaller kernel size can integrate the same information while maintaining a given contextual area. Figure 5 shows an example of transposed convolutional layer and sub-pixel convolution layer. It is worth noting that transposed convolutional layer increases resolution by using interpolation operation or adding zero values in the first place, and the raw results are then filled in with sensible values by employing convolutions on it. Without filling any artificial value in space between pixels, sub-pixel convolution layer simply employs regular convolutions on the low resolution feature map, and reshapes it to high resolution by phase shifting in a single step.

Instead of only using upper layer's output, feature maps of different resolutions are combined to improve the segmentation performance. This combination is set to address the trade-off between recognition and localization. Indeed, the high resolution feature maps from lower layers are precise but show a small receptive field. On the other hand, low resolution feature maps from upper layers deliver low spatial details, although on wider samples. Hence, upper layer can detect some objects that lower layer cannot. Therefore, it is not a wise choice to reduce the depth of the network blindly or to discard the high resolution feature maps.

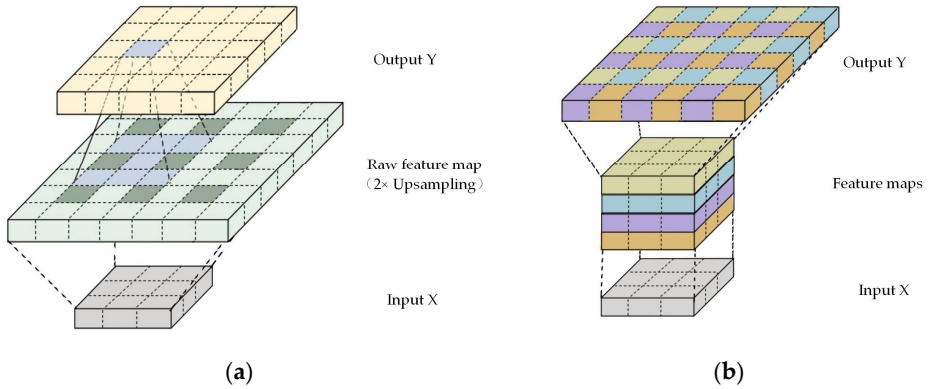


Figure 5. Example of: (**a**) transpose convolutional layer; and (**b**) sub-pixel convolution layer.

The common approach to tackle this issue is delivered on an element-wise addition. For instance, the skip connections of FCN-8s that first upscale the lower resolution feature maps can fit the higher resolution ones and then add them element-wise. However, this linear combination cannot provide accurate characterization of practical scenarios, and a nonlinear combination is required. Here, we propose to utilize multi-layer perceptron (MLP) [26] to learn how to combine feature maps at different resolutions. MLP is a minimal system with one hidden layer, as shown in Figure 3. Specifically, multiple scale feature maps including features of lightweight branch are concatenated in depth, and a hidden layer with 1×1 convolutional kernels is employed on the pool of features to approximate the combining function. The detail configurations of decoder are shown in Table 2.

Table 2. Configurations of layers in decoder.

Layer	Filter Size	Number of Filters	Stride	Padding
Conv6	3	32	1	1
ConvI1	1	IP1: 64; IP2: 64; IP3: 256	1	1
ConvI2_1	1	IP1: 128; IP2: 128; IP3: 128	1	1
ConvI2_2	3	IP1: 128; IP2: 128; IP3: 512	1	1
ConvI3_1	1	IP1: 64; IP2: 64; IP3: 64	1	1
ConvI3_2	5	IP1: 32; IP2: 32; IP3: 128	1	1
ConvI4_1	1	IP1: 32; IP2: 32; IP3: 64	1	1
ConvI4_2	7	IP1: 32; IP2: 32; IP3: 128	1	1
Conv7_1	1	256	1	1
Conv7_2	3	384	1	1
Layer	**Scale**	**Input Channel**	**Output Channel**	
SP1	2	256	64	
SP2	4	1024	64	
SP3	8	384	6	

3.2. Post-Processing Method for FSN-Based Segmentation

The proposed FSN network provides relatively fine segmentation result. However, it still shows some drawbacks, such as slight inaccuracy in determining the border of objects. These effects might be caused by the max valued label assignment probability criterion, which does not consider the occurrence of classes with a lower probability. To further improve the accuracy of the segmentation results, we adopt fully connected conditional random fields (CRFs) as post-processing method in our segmentation task.

Several works have already applied CRFs to refine the segmentation results, and improved the segmentation performance [18,21]. In this work, the output of softmax layer (heat maps) is inputted into the fully connected CRFs as unary potential, and the CIR image is inputted as the pairwise potential with color and position information. Thus, CIR image is served as the pairwise potential to

describe the "distance" between each pixel, and it is also related to the color information. The general segmentation pipeline is shown in Figure 6.

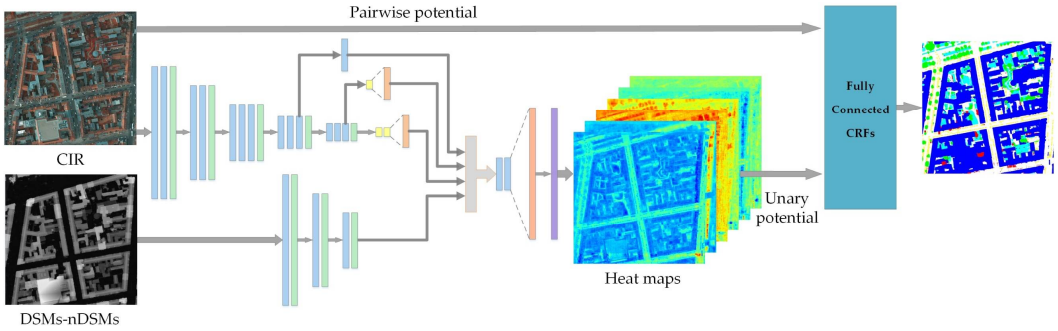


Figure 6. General procedure of the image segmentation.

4. Experiments and Results Analysis

4.1. Experimental Design

To compare the proposed network with the state-of-the-art methods, we tested all the considered algorithms on two open benchmarks of aerial image labeling provided by Commission III of ISPRS [36,37], namely Potsdam and Vaihingen datasets. In this section, we first briefly introduce the datasets, and then the competing scheme is presented. Finally, an introduction of the evaluation metrics for the test results is provided.

(1) Dataset description

The datasets used in this work are two famous open airborne datasets provided by Commission III of ISPRS [36,37]. These datasets include very high resolution true orthophoto (TOP) tiles, DSMs, and corresponding ground truth maps of two German regions. Both regions cover urban scenes. Specifically, Potsdam is a historic city with dense settlement structure, whilst Vaihingen is a small village with detached buildings.

The Potsdam dataset [36] consists of 38 TOP tiles: each tile has six channels, i.e., near-infrared, red, green, blue, DSMs, and nDSMs. The spatial resolution of image tiles is 5 cm, and they are all of size 6000×6000 pixels. Six classes (impervious surfaces, building, low vegetation, tree, car, and clutter) have been pixel-wise labeled on 24 tiles. The considered segmentation architectures use five channels, i.e., near-infrared, red, green, DSMs, and nDSMs. We randomly choose four tiles to be validation set, namely 5_10, 6_7, 6_12, and 7_10, and four tiles are chosen to be test set, namely 2_12, 4_10, 5_11, and 7_11. The other 16 tiles are employed for training.

The Vaihingen dataset [37] includes 33 TOP tiles with a spatial resolution of 9 cm. For each tile, there are five channels including near-infrared, red, green, DSMs, and nDSMs, as also provided by Gerke et al. [38]. The average size of the TOP tiles is 2494×2064 pixels. Only 16 tiles have ground truths, which also contain the same six classes as the Potsdam dataset. Here, we also employ the same five channels. Number 17, 34, and 37 are selected for validation, and number 3, 11, and 32 are selected for testing, while the remaining 10 tiles are chosen for training.

(2) Training and Inference Strategy

The proposed FSN network is trained with sparse softmax cross-entropy loss function. Adam Optimizer [39] is employed to optimize the loss function. We utilize parameters of pre-trained Vgg16 net [40] to initialize the main encoder to employ parameter-transfer learning. The lightweight branch and decoder part are initialized by normally distributed random variables. For Potsdam dataset, we set a low learning rate 10^{-5} and step down five times every five epochs, the batch size is set to 10, and image patch size of this dataset is 512×512 pixels; for Vaihingen dataset, the initial

learning rate is 10^{-5} and step down five times every five epochs, the batch size is set to 20 and image patch size is 256 × 256 pixels, the further details of these two datasets are present in Section 4.1. Data augmentation is adopted to mitigate the over-fitting phenomenon caused by constraints of the labeling data. The image patch is extracted with 50% overlap, and each image patch is flipped vertically and horizontally, and then rotated 90°, 180° and 270° [27].

In the inference stage, sliding window overlap is employed to mitigate the border effect. The full tile test images of two datasets we used in this work are larger than 2000 × 2000 pixels, and we need to clip the image into small patches to fit the memory constrain. This processing leads to a problem that the segmentation results of patch border are bound to suffer from inconsistent phenomenon. Thus, we set 75% overlapping size in the inference procedure, as the size is proven to achieve the best accuracy in previous works [23,27].

(3) Competing methods

To prove the rationality of the lightweight design, the proposed lightweight (3 layers) branch for DSMs and nDSMs is compared with middleweight (6 layers) and heavyweight (9 layers) branches first. The proposed network without sub-pixel convolution layer version and without MLP version are then evaluated to compare with the proposed version. The former one is removing all sub-pixel convolution layers (SP1–3) and replacing it with transposed convolutional layers to check if sub-pixel convolution layer can bring benefits to the segmentation task. The latter one is set to study whether MLP can achieve better feature combination performance than common element-wise addition. Hence, we remove MLP in our design and combine the feature maps at different resolutions by element-wise adding. In this case, to equalize the number of feature maps to the addition, the filter number of Conv6 and ConvL3 are set to 64.

We further evaluate the performance of the proposed FSN and FSN-noL by comparing with FCN-8s, SegNet, and HSN, in which FSN-noL represent FSN without lightweight branch version. Since FCN-8s and SegNet are acting as strong baselines for CIR images only, we use near-infrared, red and green channels to train these architectures. HSN is designed for five channels input, so it served as the CIR + LiDAR data baseline. The details of these baselines can be found in Appendix A. Overlap inference with 75% overlapping size is employed in the inference stage of all experiments.

(4) Evaluation metrics

To guarantee a fair comparison, we evaluated the performance of the considered frameworks in terms of overall accuracy (OA), per-class F_1 score, average F_1 score. Moreover, trainable weights for each testing result are computed. OA is the ratio of the number of correct labeled pixels and the total number of the whole image pixels. F_1 score can be expressed as:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision \ + \ recall} \tag{6}$$

where precision and recall can be calculated based on the confusion matrices. Precision is the true positive pixels divided by the sum of true positive and false positive pixels. Recall is the ratio of true positive pixels and the sum of true positive and false negative pixels.

4.2. Validation of Lightweight Branches

We firstly present the comparison results of the middleweight, the heavyweight and the proposed lightweight branches. Specifically, light-, medium- and heavy-weight branches are characterized by 3, 6, and 9 convolutional layers, respectively. Figure 7 shows the structures of these branches, in which all max pooling layers have size 2 and stride 2. Table 3 shows the configurations of the middleweight and heavyweight branches, whereas the configurations of lightweight branch are presented in Section 3.1 (see ConvL1~PoolL3 in Table 1).

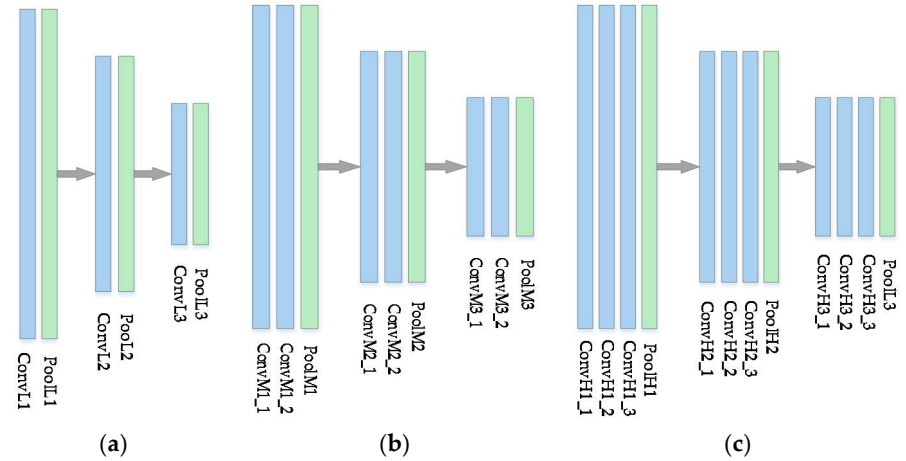


Figure 7. Structure of multi-scale extra branches: (**a**) lightweight branch; (**b**) middleweight branch; and (**c**) heavyweight branch.

Table 3. Configurations of convolutional layers in middleweight branch and heavyweight branch. ConvM is convolutional layer of middleweight branch; ConvH is convolutional layer of heavyweight branch.

Layer	Filter Size	Number of Filters	Layer	Filter Size	Number of Filters
ConvM1_1	3	64	ConvH1_3	3	64
ConvM1_2	3	64	ConvH2_1	3	128
ConvM2_1	3	128	ConvH2_2	3	128
ConvM2_2	3	128	ConvH2_3	3	128
ConvM2_2	3	64	ConvH3_1	3	64
ConvM2_2	3	16	ConvH3_2	3	64
ConvH1_1	3	64	ConvH3_3	3	16
ConvH1_2	3	64			

Table 4 shows the segmentation performances of three extra branches on Potsdam dataset, and the results are assessed by considering the original ground truth (GT) and its erode version. The erode ground truth (E-GT) aims to exclude the impact of uncertain border definitions, so the boundaries of objects are eroded by a circular disc of 3 pixel radius. The lightweight branch provides the best results in most cases. Moreover, the accuracy decreases as the depth of the extra branch increases. This proves that the lightweight convolutional network can generate the appropriate features to complement with the preliminarily upscaled features from the upper layer of the main encoder.

Table 4. Experimental results on the multi-scale extra branch (Potsdam validation set). HW is FSN with heavyweight branch; MW is FSN with middleweight branch; LW is FSN with lightweight branch (the proposed version); GT is Ground Truth; E-GT is Eroded Ground Truth.

	Methods	Imp. Surf.	Build	Low Veg.	Tree	Car	Clutter	Aver. F_1	OA
GT	HW	91.03	90.83	83.84	80.79	90.71	**85.76**	87.16	87.43
	MW	91.01	**91.30**	84.27	80.59	90.85	84.79	87.14	87.54
	LW	**91.41**	91.16	**84.41**	**81.14**	**91.51**	83.50	**87.19**	**87.59**
E-GT	HW	92.90	91.60	85.70	83.71	96.13	**88.05**	89.68	89.26
	MW	92.89	**92.90**	86.01	83.67	96.20	87.10	**89.80**	89.30
	LW	**93.23**	91.89	**86.26**	**84.01**	**96.53**	85.73	89.61	**89.44**

4.3. Validation of Sub-Pixel Convolution Layers and Multi-Layer Perceptron

The results of the proposed FSN, FSN without sub-pixel convolution layers (FSN-noSC), and FSN without multi-layer perceptron (FSN-noMLP) versions are reported in this subsection. Table 5 shows the numerical results of these three models evaluated on the test set of Potsdam dataset. Apparently, FSN achieve the best performance under all the accuracy metrics. When compared with FSN-noSC (which is without sub-pixel convolution version), the improvement in terms of overall accuracy of the proposed FSN reaches up to 0.6%. The same scale of improvement can be found in the comparison between the proposed FSN and the FSN without MLP version (FSN-noMLP). This outcome proves that sub-pixel convolution layer and multi-layer perceptron provide a strong contribution to the segmentation performance of the FSN design.

Table 5. Experimental results on the effect of sub-pixel convolution and multi-layer perceptron (Potsdam validation set). FSN-noSC is FSN without sub-pixel convolution version; FSN-noMLP is FSN without multi-layer perceptron version.

	Methods	Imp. Surf.	Build	Low Veg.	Tree	Car	Clutter	Aver. F_1	OA
GT	FSN-noSC	90.84	90.44	83.51	79.18	90.92	**85.80**	86.78	86.99
	FSN-noMLP	90.21	90.02	83.21	80.11	89.81	85.56	86.49	86.72
	FSN	**91.41**	**91.16**	**84.41**	**81.14**	**91.51**	83.50	**87.19**	**87.59**
E-GT	FSN-noSC	92.70	91.19	85.23	82.27	96.43	**88.02**	89.31	88.79
	FSN-noMLP	92.13	90.87	85.08	83.16	95.59	87.85	89.11	88.61
	FSN	**93.23**	**91.89**	**86.26**	**84.01**	**96.53**	85.73	**89.61**	**89.44**

Visual comparisons (Figure 8) show that the segmentation results of FSN are more accurate and coherent compared to the results of FSN-noSC and FSN-noMLP. For instance, the building with roof lawn in Figure 8a–e is easily confused by low vegetation. When removing multi-layer perceptron or sub-pixel convolution layer the building is mislabeled with low vegetation in different degree; on the other hand, the proposed FSN can label it correctly. In Figure 8f–j, we can also observe the mislabeled car in the middle of the segmentation result of FSN-noSC and FSN-noMLP. In contrast, the segmentation results of the proposed FSN are more precise. Through the above analysis, we can state that sub-pixel convolution layer has strong ability to obtain richer representations and MLP can learn to combine features in an appropriate manner.

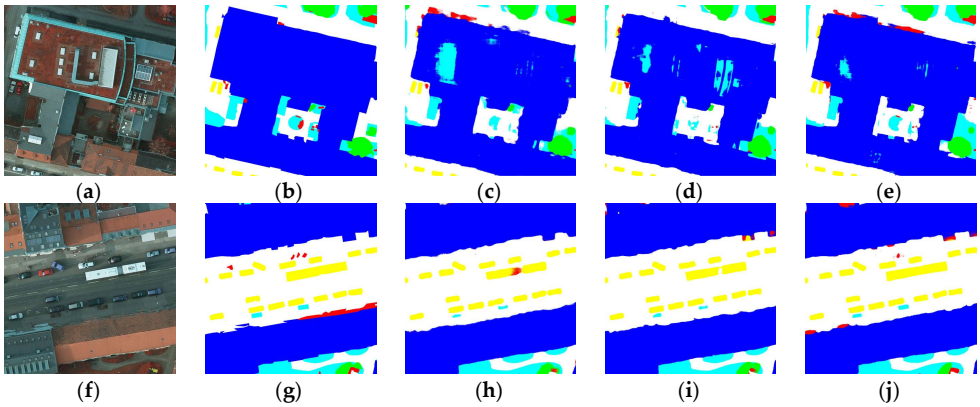


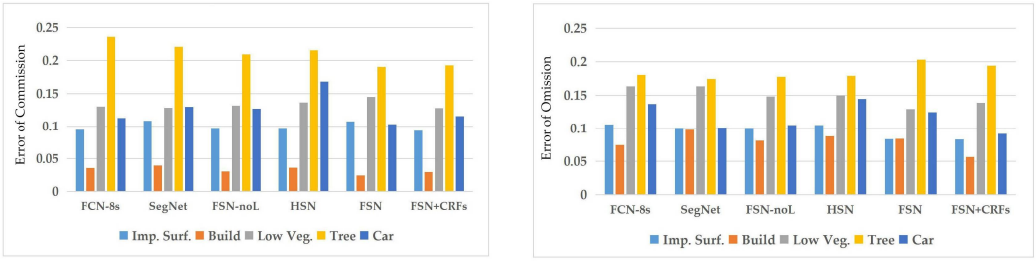
Figure 8. Semantic Labeling results for two patches of Potsdam validation set. Classes: impervious surface (white); buildings (blue); low vegetation (cyan); tree (green); car (yellow); clutter (red). In the first row, (**a**) is true orthophoto, (**b**) is ground truth, (**c**–**e**) are inference results of FSN-noMLP, FSN-noSC and the proposed FSN for image patch with building with roof lawn; in the second row, (**f**) is true orthophoto, (**g**) is ground truth, (**h**–**j**) are inference results of FSN-noMLP, FSN-noSC and the proposed FSN for image patch with a street between buildings.

4.4. Potsdam Dataset Results

In this subsection, we evaluate FCN-8s, SegNet, FSN-noL, HSN, FSN, and FSN with post-process (FSN + CRFs) by using Potsdam test set. The comparisons are reported in Table 6. As expected, the proposed FSN outperforms the other methods in almost every evaluation metrics. For CIR images only, FSN-noL shows better performance than that of FCN-8s and SegNet. The class clutter of Potsdam dataset is hard to be correctly labeled in most previous works, for its high intra-class variance caused by the diversified components. The proposed FSN can achieve an over 5% increased F_1 score of this class compared with other networks. It is worth noting that HSN integrated with Markov random field can smooth the raw segmentation results and further improve the accuracy. In this case, we employ fully connected CRFs as post-processing method to further improve the accuracy of our work. With the help of color and position information of original images, we achieve about 0.5% increase in overall accuracy. Figure 9a,b shows the errors of commission and omission for each method per classes. Then, we can observe that FSN-noL achieved lower errors compared with FCN-8s and SegNet: specifically, it made fewer mistakes in class impervious surface without LiDAR data. FSN and FSN + CRFs show lowest errors of commission and omission. By comparison, HSN suffer from higher errors in class tree and car, for its deficiencies on multi-sensor fusion.

Table 6. Experimental results on Potsdam test set.

	Methods	Imp. Surf.	Build	Low Veg.	Tree	Car	Clutter	Aver. F_1	OA
GT	FCN-8s	90.02	94.59	85.59	78.59	87.68	46.14	80.44	87.36
	SegNet	89.52	93.33	85.68	79.78	88.28	44.69	80.21	87.08
	FSN-noL	90.11	94.54	86.12	80.34	88.29	44.91	80.72	87.74
	HSN	89.92	93.96	85.80	79.90	84.20	44.24	79.67	87.30
	FSN	90.34	94.74	86.19	80.46	88.75	51.43	81.99	87.91
	FSN + CRFs	**91.14**	**95.73**	**86.75**	**80.66**	**89.52**	**51.79**	**82.60**	**88.57**
E-GT	FCN-8s	92.34	95.84	87.95	81.63	94.01	50.87	83.77	89.80
	SegNet	91.76	94.58	88.08	82.87	94.76	49.30	83.56	89.51
	FSN-noL	92.41	95.81	88.58	83.44	94.63	49.50	84.06	90.21
	HSN	92.23	95.25	88.24	82.99	89.82	48.83	82.89	89.77
	FSN	92.95	95.85	88.61	**83.76**	95.19	55.92	85.38	90.51
	FSN + CRFs	**93.32**	**96.89**	**89.08**	83.69	**95.66**	**56.41**	**85.84**	**90.94**



(**a**) (**b**)

Figure 9. Errors of commission and omission of each model per classes (Potsdam dataset): (**a**) error of commission; and (**b**) error of omission. Lower values indicate the better segmentation performance.

Visually, we can observe in Figure 10a–h that the impervious surface with scattering lawn is hard to be recognized. FCN-8s, SegNet, and FSN-noL fail to label this kind of impervious surface correctly. Indeed, FSN-noL achieves a better performance on it. Thanks to the LiDAR data, HSN, FSN and FSN + CRFs can label the building and impervious surface more accurately; especially, the proposed FSN and FSN + CRFs prove that FSN can properly fuse multi-sensor features. The sparse low vegetation shares a similar color to impervious surface and deciduous trees in this dataset: in fact, as shown in Figure 10i–p, part of impervious surface is labeled as low vegetation in segmentation results of FCN-8s, SegNet and HSN. Additionally, class clutter consists of different kinds of objects

(e.g., water bodies, playgrounds, and containers), so the networks can hardly label this class. In Figure 10q–x, FSN and FSN-noL can correctly segment the class clutter and have a better detail characterization ability. Hence, thanks to the inception layers, which provide FSN multi-scale receptive field, it can achieve a relatively complete building segmentation result. In addition, the implementation of MLP brings more appropriate features to mitigate the recognition/localization trade-off, so that the segmentation results of small objects such as cars and small scale clutters have accurate outlines. Furthermore, the post-process (i.e., fully connected CRFs) smooths the raw segmentation results and amends some tiny mislabeled blocks. Figure A4 in Appendix B shows the full tile predictions of Potsdam dataset. To highlight the difference between methods, each tile is followed by its red/green images that marks mislabeled pixels in red and correctly labeled pixels in green.

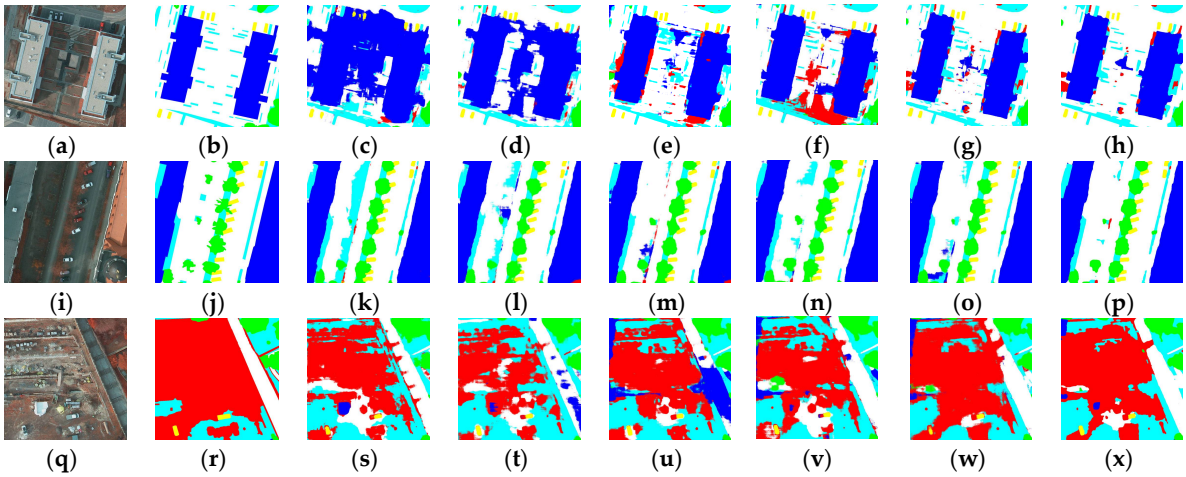


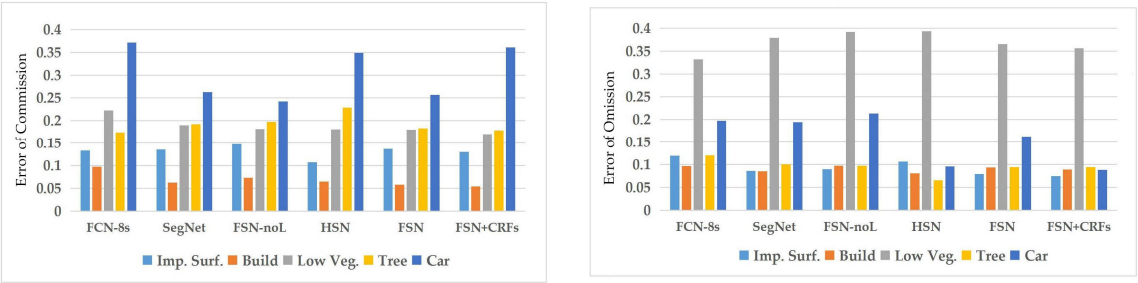
Figure 10. Semantic Labeling results for three patches of Potsdam test set. Classes: impervious surface (white); buildings (blue); low vegetation (cyan); tree (green); car (yellow); clutter (red). In the first row, (**a**) is true orthophoto, (**b**) is ground truth, (**c**–**h**) are inference results of FCN-8s, SegNet, FSN-noL, HSN, FSN and FSN+CRFs for image patch with buildings; in the second row, (**i**) is true orthophoto, (**j**) is ground truth, (**k**–**p**) are inference results of FCN-8s, SegNet, FSN-noL, HSN, FSN and FSN+CRFs for image patch with a street between buildings; in the third row, (**q**) is true orthophoto, (**r**) is ground truth, (**s**–**x**) are inference results of FCN-8s, SegNet, FSN-noL, HSN, FSN and FSN+CRFs for image patch with clutters.

4.5. Vaihingen Dataset Results

The lower spatial resolution and the shortage of the labeled data of Vaihingen dataset make the segmentation performance worse with respect to the Potsdam images. The results are listed in Table 7. We can see that FSN exhibits the best performance in both average F_1 score and overall accuracy compared with other methods. Moreover, FSN-noL achieves higher overall accuracy than that of FCN-8s and SegNet. As for per-class F_1 score, FSN + CRFs achieve highest score in most classes. Figure 11a,b shows the errors of commission and omission for each method per classes of Vaihingen test set. Thanks to the wider receptive field and appropriate feature fusion approach of FSN, errors of FSN-noL were lower than those delivered by FCN-8s and SegNet for the majority of the classes. Further, FSN and FSN + CRFs outperform HSN and other networks.

Table 7. Experimental results on Vaihingen test set.

	Methods	Imp. Surf.	Build	Low Veg.	Tree	Car	Aver. F_1	OA
GT	FCN-8s	87.28	90.28	73.70	84.91	68.84	81.00	84.65
	SegNet	87.79	91.59	74.02	84.49	76.87	82.95	85.07
	FSN-noL	88.63	92.68	73.98	84.67	76.30	83.25	85.66
	HSN	89.28	92.80	74.04	83.96	74.56	82.93	85.75
	FSN	88.89	92.55	75.04	85.50	**78.01**	**84.00**	86.13
	FSN + CRFs	**89.49**	**92.95**	**75.93**	**85.78**	74.01	83.63	**86.63**
E-GT	FCN-8s	90.58	92.70	78.66	89.43	77.02	85.68	88.59
	SegNet	91.13	93.89	78.80	89.09	84.60	87.50	88.98
	FSN-noL	91.99	95.03	78.92	89.32	84.78	88.01	89.64
	HSN	92.40	95.15	78.92	88.76	82.42	87.53	89.67
	FSN	92.27	94.89	80.19	90.14	**86.31**	**88.76**	90.14
	FSN + CRFs	**92.78**	**95.18**	**80.99**	**90.33**	83.26	88.51	**90.55**



(**a**) (**b**)

Figure 11. Errors of commission and omission of each model per classes (Vaihingen dataset): (**a**) error of commission; and (**b**) error of omission. Lower values indicate the better segmentation performance.

Figure 12 illustrates some predictions of closeups, and Figure A5 in Appendix B shows the full tile prediction and its red/green images of Vaihingen dataset. In Figure 12a–h, the shadows of trees pose difficulties for the segmentation task, whilst trees and low vegetation are similar in color, so most networks fail to distinguish them correctly. Cement road and Low-rise building with cement roof are also prone to confused by networks, as shown in Figure 12i–p. The main reason for these ambiguities is the insufficient contextual and spatial information of networks. As expected, these classes with small inter-class differences can be well labeled by FSN. Moreover, FSN-noL outperforms FCN-8s and SegNet in terms of accuracy on those classes, since inception modules and sub-pixel convolution layers provide more diverse and wider vision for the proposed network. The lower spatial resolution makes the small-scale objects such as the cars in this dataset hard to be segmented correctly when compared with Potsdam dataset, as illustrated in Figure 12q–x, FCN-8s, SegNet and HSN mislabel part of the cars as impervious surface, while FSN-noL and the proposed FSN can accurately label the cars.

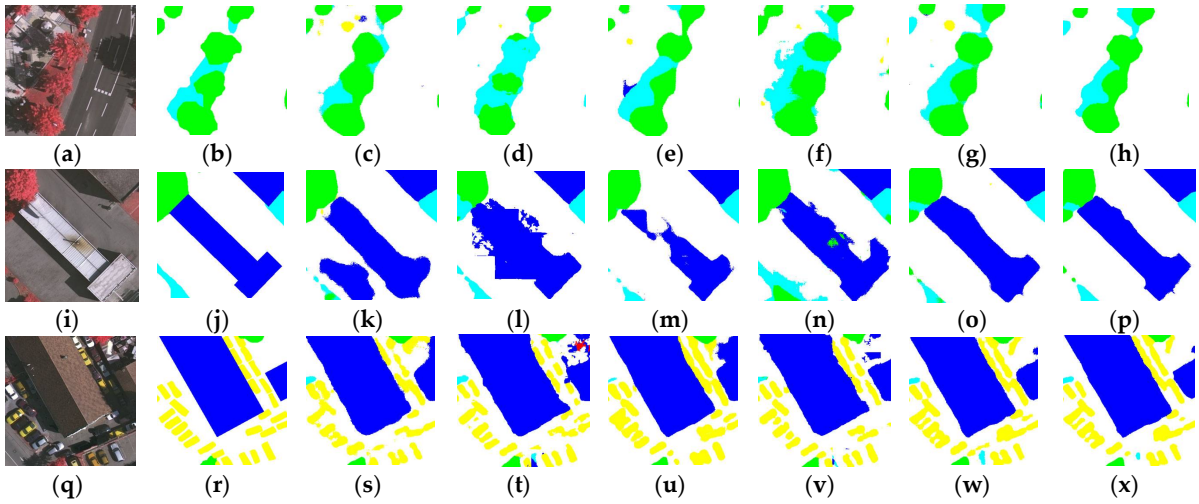


Figure 12. Semantic Labeling results for three patches of Vaihingen test set. Classes: impervious surface (white); buildings (blue); low vegetation (cyan); tree (green); car (yellow); clutter (red). In the first row, (**a**) is true orthophoto, (**b**) is ground truth, (**c**–**h**) are inference results of FCN-8s, SegNet, FSN-noL, HSN, FSN and FSN+CRFs for image patch with trees and low vegetation areas; in the second row, (**i**) is true orthophoto, (**j**) is ground truth, (**k**–**p**) are inference results of FCN-8s, SegNet, FSN-noL, HSN, FSN and FSN+CRFs for image patch with a low-rise building; in the third row, (**q**) is true orthophoto, (**r**) is ground truth, (**s**–**x**) are inference results of FCN-8s, SegNet, FSN-noL, HSN, FSN and FSN+CRFs for image patch with cars parked around buildings.

4.6. Submission to the ISPRS Challenge

We submitted the test results on the hidden test sets of Potsdam dataset (ID: "CASDE") and Vaihingen dataset (ID: "CASRS") to the ISPRS Challenge, which can be accessed on line [41,42]. We scored 90.0% and 89.5% in overall accuracy for Potsdam and Vaihingen test set, respectively. Our score belongs to upper middle class in the leaderboard. Indeed, we believe that the results we achieved provide a significant point for discussion and enhancement for the application of deep learning techniques in the remote sensing community. In fact, when compared to the other architectures in the competition, our framework is characterized by a smaller computational cost. Hence, the trade-off between accuracy and computational complexity of the proposed approach is higher than that of several models in the ISPRS challenge. Further, this effect makes the proposed FSN a valid option for semantic labeling of remote sensing data by means of deep learning methods, especially in terms of system efficiency. Specifically, some architectures have mainly focused on improving accuracy by using multimodel feature fusion, which reasonably leads to a major effort in terms of required computational complexity. For instance, in the Potsdam 2D Labelling challenge, a recent submission (ID: "BKHN2") [41] employed all channels and ensemble five FCN-8s (VGG) models to improve the accuracy. When compared with the proposed FSN, it achieves 0.6% increase in overall accuracy. On the other hand, the amount of trainable weights is strongly increased, since a single FCN-8s has many more trainable weights than FSN. Moreover, when we compare our approach with the similar-scale networks such as "RITL7" [24], it is worth noting that it achieves an overall accuracy of 88.4% by fusing multisensor features in decision-level by means of higher order CRFs. Hence, the FSN (ID: "CASDE2") we introduce outperforms "RITL7" both in terms of overall accuracy and per class F_1-score. Additionally, we also submitted the results of the FSN-noL (ID: "CASDE1"), and scored 89.7% points in overall accuracy. When compared to FSN, the F_1-scores of the class building and tree dropped by 0.8% and 0.3%, respectively. These effects mean that the fusion with LiDAR data delivers a valuable enhancement to the recognition of building and tree.

In the Vaihingen 2D Labelling challenge, the SegNet with multi-kernel convolutional layer and dual-stream fusion strategy (ID: "ONE_7") [23] achieved 0.3% increase in overall accuracy when

compared with the proposed FSN. However, it also suffers from more than twofold increase in trainable weights. Moreover, the proposed FSN (ID: "CASRS1") outperforms many other networks in this dataset, such as ID: "UZ_1" (FCN + nDSM) [25] with an 87.3% overall accuracy and ID: "DST_2" (FCN-noDS + RF + CRFs) [1] with an 89.1% overall accuracy. The results of FSN-noL on this dataset (ID: "CASRS2") have been submitted to the ISPRS challenge evaluation as well. This test scored 88.7% overall accuracy. Analogously to Potsdam dataset, the F_1-scores of impervious surface, building and tree are lower than FSN, which further shows the benefits of fusion with LiDAR Data.

4.7. Trainable Weights and Receptive Fields

Table 8 reports the trainable weight counts in each model considered in this paper. Since FCN-8s, SegNet and FSN share the same structure of encoder and at least 13 layers are employed in each encoder stage, the trainable weights of these three models are more than those employed in HSN, which has only nine layers in the corresponding stage. However, as the experimental results show, the decrease of the encoder layers and downsampling scale may cause the segmentation accuracy loss. Especially when the spatial resolution increased, for instance, the HSN has less encoder layer and its evaluation results on Potsdam dataset (which has a higher spatial resolution of 0.5 m than Vaihingen dataset) are worse than the other three models. However, the large amount of trainable weights makes the training phase more difficult, and causes the over-fitting phenomenon, which becomes more apparent when the number of training samples is limited. For example, FCN's evaluation results on Vaihingen dataset are worse than its performance on Potsdam dataset. Finally, the proposed FSN can balance the trade-off between performance and computational complexity, and achieve a higher accuracy with relatively fewer parameters.

Table 8. Trainable weight and receptive field of the FCN-8s, SegNet, HSN and the proposed FSN.

Network	FCN-8s	SegNet	HSN	FSN
Trainable Weights	140.5 M	29.4 M	5.56 M	18.0 M
Receptive Field	404	212	212	596

The largest receptive field of each model is also included in the Table 8. We can observe from the table that FSN has wider receptive field than other models. It is worth noting that both HSN and FSN have multi-scale receptive areas: this effect is caused by the use of inception layers, which allows them to achieve higher overall accuracy by enriching the contextual information.

5. Conclusions

A novel fine segmentation network (FSN) for semantic segmentation of multi-sensor remote sensing data is proposed in this paper. The architecture follows an encoder–decoder structure with a feature-level fusion approach. The encoder includes a main encoder for CIR data and a lightweight branch designed for LiDAR data. In the decoder stage, inception modules are introduced to enrich the receptive field and contextual information of the network. Sub-pixel convolution layers are employed to allow the network adaptively upscale the feature maps. Furthermore, the multi-sensor and multi-resolution feature maps are combined by multi-layer perceptron in an efficient and flexible manner. Transfer learning is used to tackle the shortage of training sample. Overlapping inference is used to mitigate the border effects, and fully connected CRFs serves as post processing method to further improve the accuracy. Experimental results based on ISPRS 2D Potsdam and Vaihingen datasets show that the proposed FSN outperforms the other related networks and provides better segmentation results with a relatively moderate computational complexity. Next steps in this research field will consider applying K-fold cross-validation for hyperparameter search.

Author Contributions: X.P. was mainly responsible for network modeling and experimental designing. L.G. provided the original idea for the proposed methods and to the experimental analysis. A.M. offered valuable

suggestions and comments for the network design and experiments. B.Z. completed the theoretical framework. F.Y. provided support regarding the application of post-processing method. P.G. provided important suggestions for improving quality of the paper.

Acknowledgments: This research was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDA19080302, and by the National Natural Science Foundation of China under Grant No. 91638201.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Related Works of CNN Architectures

In this section, we introduce several existing CNN architectures as the baseline to compare with our proposed fine segmentation network (FSN), which are fully convolutional network (FCN) [17], SegNet [20], and hourglass-shape network HSN [27].

Appendix A.1. Architecture 1: FCN

First, we present FCN as it makes a significant contribution to semantic segmentation by using CNN. The architecture of FCN is shown in Figure A1. In the encoder stage (within the solid-line box), we choose the Vgg16 net [28], which could achieve brilliant performance in [17] compared to other deep CNNs. On the other hand, in the decoder part (within the dashed-line box), we employed skip connections and built the net FCN-8s.

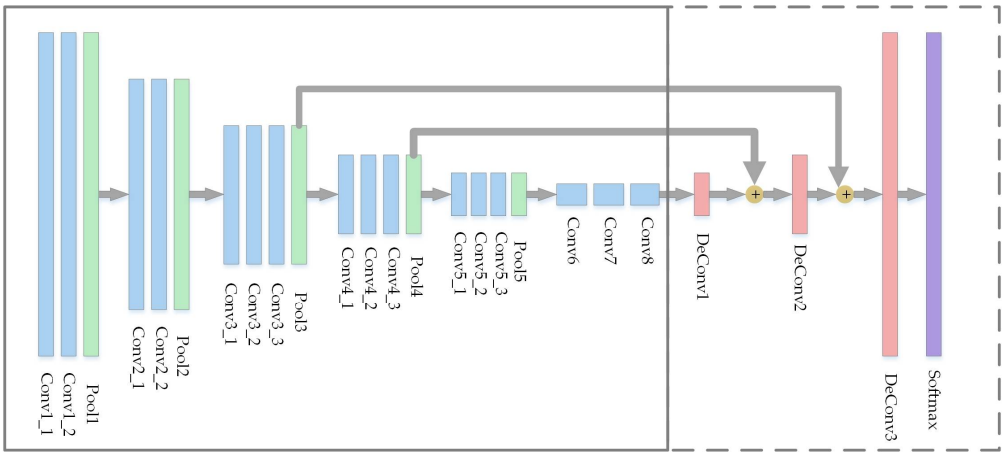


Figure A1. Architecture of FCN-8s, where structure depicted in the solid-line box is encoder and structure depicted in the dashed-line box is decoder.

Each blue block represents convolution layer followed by a ReLU activation, green blocks indicate pooling layers, and red blocks are deconvolutional layers, also called transpose convolutional layers. Instead of upscaling feature maps to the original image size by a large factor, FCN-8s upscale the feature maps in three steps to mitigate classification ambiguities in the up-sampled result. Although it achieves better performance than FCN-32s which upsample feature maps by a large factor 32, it still suffers from spatial-information loss problem [18].

Appendix A.2. Architecture 2: SegNet

The second architecture is SegNet [20]. The architecture is shown in Figure A2. SegNet's encoder part is very similar to the one proposed in FCN-8s, except for the Conv6-8 use and the decoder part is the mirror version of encoder part. The blue and green blocks are the same as with FCN-8s, but each convolution layer is followed by a batch normalization and a ReLU layers. The pink blocks represent unpooling layers.

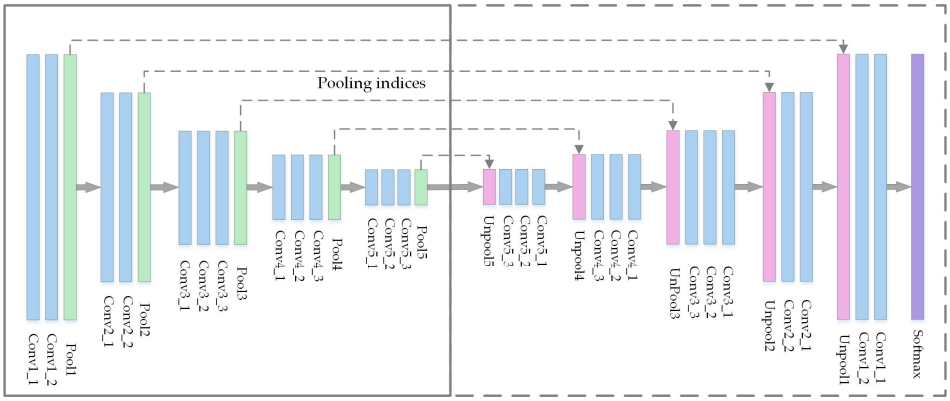


Figure A2. Architecture of SegNet, where structure depicted in the solid-line box is encoder and structure depicted in the dashed-line box is decoder.

Appendix A.3. Architecture 3: HSN

Hourglass-shaped network (HSN) for Earth observation data classification [27] is inspired by HSN for pose estimation [43] and image depth estimation [44]. It imports two powerful modules: residual module and inception module. The architecture is shown in Figure A3: the blue and green blocks represent convolution layers and pooling layers, respectively, and batch normalization is employed after each convolution layers. The brown and yellow blocks represent residual module and inception module, respectively, the structure of which is detailed in [27].

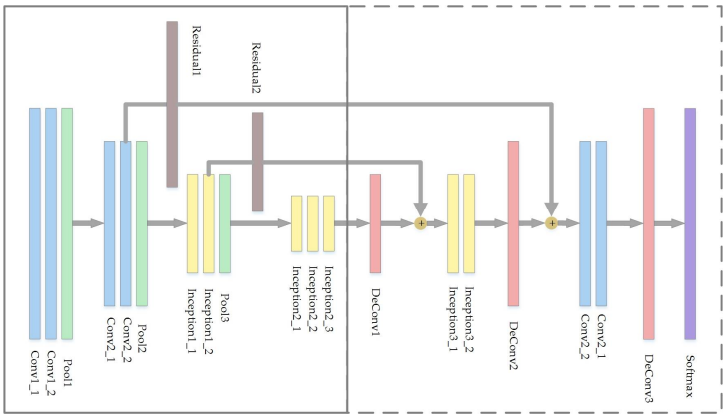
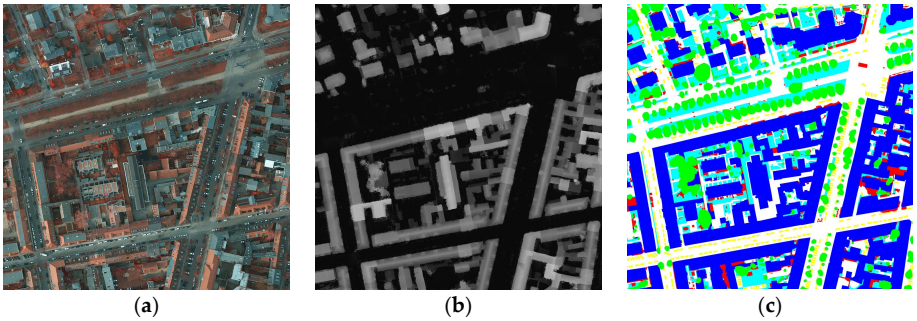


Figure A3. Architecture of HSN, where structure depicted in the solid-line box is encoder and structure depicted in the dashed-line box is decoder.

Appendix B. Full Tile Prediction



(**a**) (**b**) (**c**)

Figure A4. *Cont.*

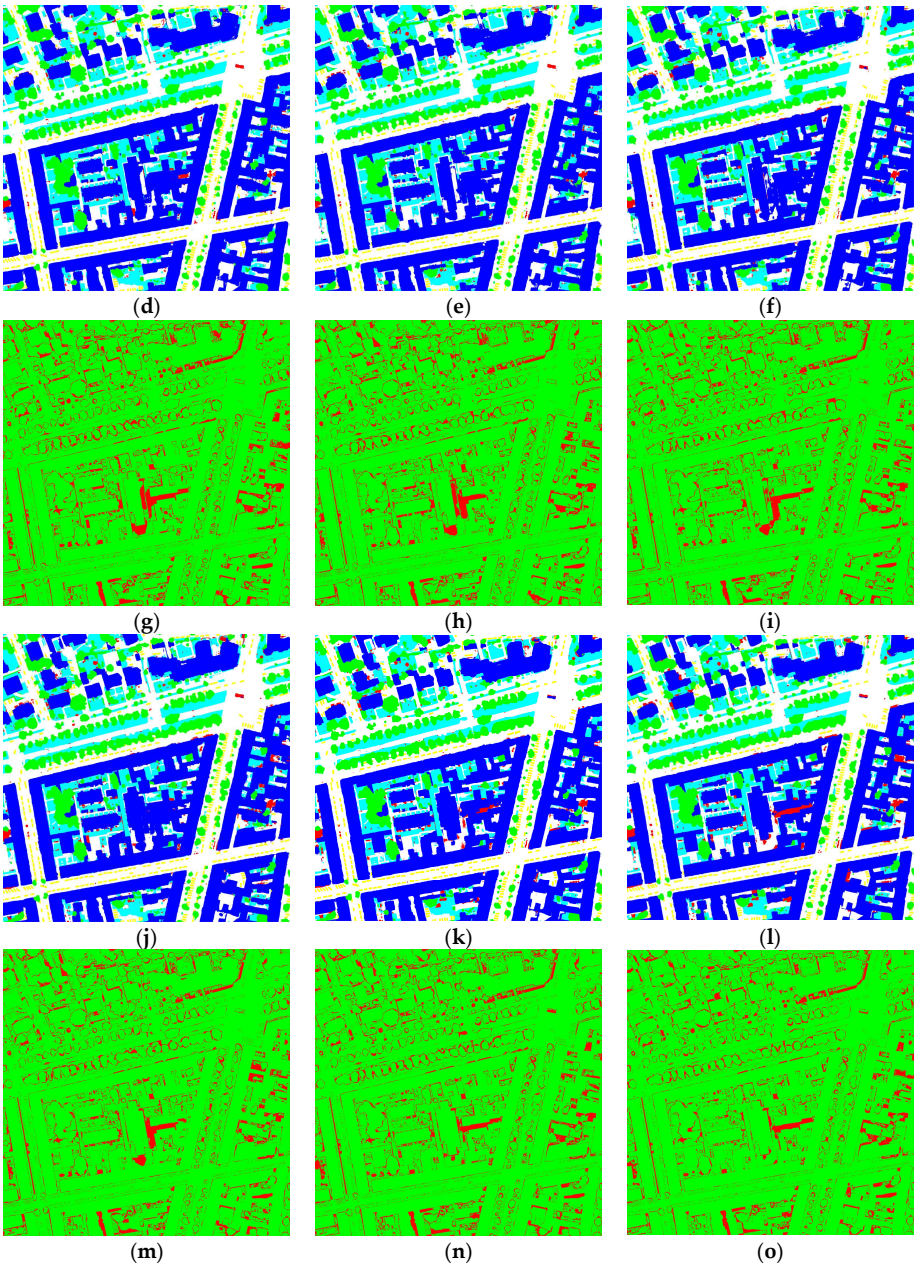


Figure A4. Full tile prediction for NO. 5_11 of Potsdam dataset. Classes: impervious surface (white); buildings (blue); low vegetation (cyan); tree (green); car (yellow); clutter (red). (**a**) TOP, true orthophoto; (**b**) nDSM, normalized DSM; (**c**) GT, ground truth; (**d**–**f**) inference result of FCN-8s, SegNet, and FSN-noL, respectively; (**g**–**i**) Red/Green Images of FCN-8s, SegNet, and FSN-noL, respectively; (**j**–**l**) inference result of HSN, FSN and FSN + CRFs respectively; and (**m**–**o**) Red/Green Images of HSN, FSN and FSN + CRFs, respectively.

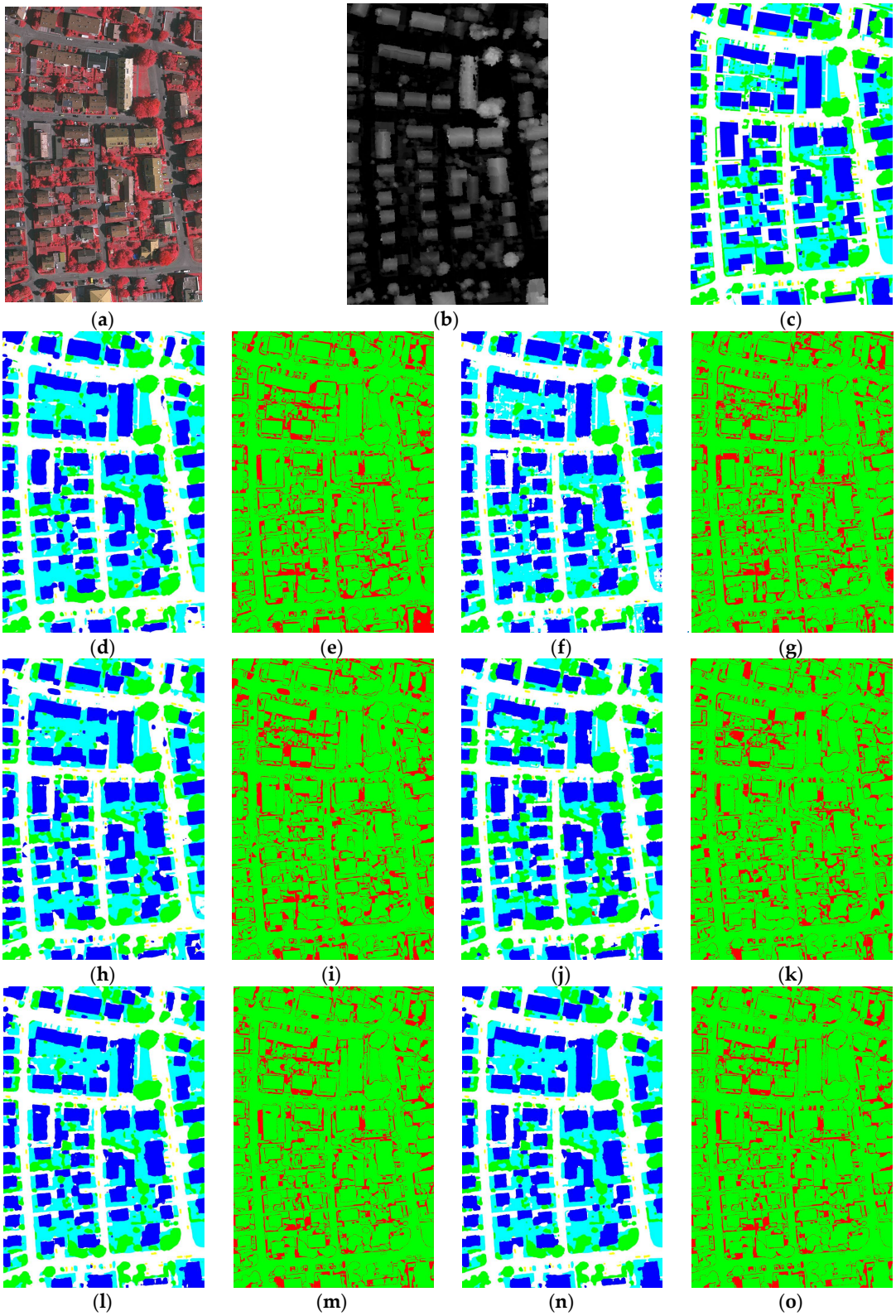


Figure A5. Full tile prediction for NO. 3 of Vaihingen dataset. Classes: impervious surface (white); buildings (blue); low vegetation (cyan); tree (green); car (yellow); clutter (red). (**a**) TOP, true orthophoto; (**b**) nDSM, normalized DSM; (**c**) GT, ground truth; (**d**,**e**) inference result and Red/Green Image of FCN-8s; (**f**,**g**) inference result and Red/Green Image of SegNet; (**h**,**i**) inference result and Red/Green Image of FSN-noL; (**j**,**k**) inference result and Red/Green Image of HSN; (**l**,**m**) inference result and Red/Green Image of FSN; and (**n**,**o**) inference result and Red/Green Image of FSN + CRFs.

References

1. Sherrah, J. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *arXiv*, 2016.
2. Sun, J.; Yang, J.; Zhang, C.; Yun, W.; Qu, J. Automatic remotely sensed image classification in a grid environment based on the maximum likelihood method. *Math. Comput. Model.* **2013**, *58*, 573–581. [CrossRef]
3. Toth, D.; Aach, T. Improved minimum distance classification with gaussian outlier detection for industrial inspection. In Proceedings of the 11th International Conference on Image Analysis and Processing, Palermo, Italy, 26–28 September 2001; pp. 584–588.
4. Jumb, V.; Sohani, M.; Shrivas, A. Color image segmentation using k-means clustering and otsus adaptive thresholding. *Int. J. Innov. Technol. Explor. Eng.* **2014**, *3*, 71–76.
5. Ratle, F.; Camps-Valls, G.; Weston, J. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2010**, *48*, 2271–2282. [CrossRef]
6. Yu, H.; Gao, L.; Li, J.; Li, S.S.; Zhang, B.; Benediktsson, J. Spectral-spatial hyperspectral image classification using subspace-based support vector machines and adaptive markov random fields. *Remote Sens.* **2016**, *8*, 355. [CrossRef]
7. Sugg, Z.; Finke, T.; Goodrich, D.; Susan Moran, M.; Yool, S. Mapping impervious surfaces using object-oriented classification in a semiarid urban region. *Photogramm. Eng. Remote Sens.* **2014**, *80*, 343–352. [CrossRef]
8. Song, B.; Li, P.; Li, J.; Plaza, A. One-class classification of remote sensing images using kernel sparse representation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 1613–1623. [CrossRef]
9. Wang, Q.; Lin, J.; Yuan, Y. Salient band selection for hyperspectral image classification via manifold ranking. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1–11. [CrossRef] [PubMed]
10. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
11. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; Lecun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
12. Cirean, D.C.; Giusti, A.; Gambardella, L.M.; Schmidhuber, J. Deep neural networks segment neuronal membranes in electron microscopy images. In Proceedings of the Advances in Neural Information Processing System, Nevada, NV, USA, 3–6 December 2012; pp. 2852–2860.
13. Pinheiro, P.; Collobert, R. Recurrent convolutional neural networks for scene labeling. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 82–90.
14. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous detection and segmentation. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 297–312.
15. Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision—ECCV*; Springer: Cham, Switzerland, 2014; pp. 345–360.
16. Ganin, Y.; Lempitsky, V. N^4-fields: Neural network nearest neighbor fields for image transforms. In Proceedings of the Asian Conference on Computer Vision (ACCV), Kent Ridge, Singapore, 1–5 November 2014; pp. 536–551.
17. Long, J.; Shelhamer, E.; Darrell, T. In Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
18. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.L.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. In Proceedings of the 3rd International Conference on Learning Representations (ICLR2015), San Diego, CA, USA, 7–9 May 2015.
19. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 1520–1528.
20. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]

21. Paisitkriangkrai, S.; Sherrah, J.; Janney, P.; Hengel, A. Effective semantic pixel labelling with convolutional networks and conditional random fields. In Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 36–43.
22. Nogueira, K.; Penatti, O.A.B.; Santos, J.D. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recogn.* **2016**, *61*, 539–556. [CrossRef]
23. Audebert, N.; Le Saux, B.; Lefèvre, S. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In Proceedings of the Asian Conference on Computer Vision (ACCV), Taipei, Taiwan, 21–23 November 2016; pp. 180–196.
24. Liu, Y.; Piramanayagam, S.; Monteiro, S.; Saber, E. 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 (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1561–1570.
25. Volpi, M.; Tuia, D. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 881–893. [CrossRef]
26. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. High-resolution aerial image labeling with convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 7092–7103. [CrossRef]
27. Liu, Y.; Minh Nguyen, D.; Deligiannis, N.; Ding, W.; Munteanu, A. Hourglass-shapenetwork based semantic segmentation for high resolution aerial imagery. *Remote Sens.* **2017**, *9*, 522. [CrossRef]
28. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Machine Learning (ICML), San Diego, CA, USA, 7–9 May 2015.
29. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1874–1883.
30. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Nevada, NV, USA, 3–6 December 2012; pp. 1106–1114.
31. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceeding of International Conference on Machine Leaning (ICML), Haifa, Israel, 21–25 June 2010; pp. 807–814.
32. Maas, A.Y.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the International Conference on Machine Learning (ICML), Atlanta, GA, USA, 16–21 June 2013; pp. 16–21.
33. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
34. Zeiler, M.D.; Taylor, G.W.; Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2018–2025.
35. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
36. ISPRS Potsdam 2D Semantic Labeling Dataset. Available online: http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html (accessed on 10 December 2017).
37. ISPRS Vaihingen 2D Semantic Labeling Dataset. Available online: http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html (accessed on 10 December 2017).
38. Gerke, M. *Use of the Stair Vision Library within the ISPRS 2d Semantic Labeling Benchmark (Vaihingen)*; University of Twente: Enschede, The Netherlands, 2015.
39. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
40. Pretrained Models. Available online: http://www.vlfeat.org/matconvnet/pretrained/ (accessed on 10 December 2017).
41. ISPRS Semantic Labeling Contest (2D): Results (Potsdam). Available online: http://www2.isprs.org/potsdam-2d-semantic-labeling.html (accessed on 30 March 2018).

42. ISPRS Semantic Labeling Contest (2D): Results (Vaihingen). Available online: http://www2.isprs.org/vaihingen-2d-semantic-labeling-contest.html (accessed on 30 March 2018).
43. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 483–499.
44. Chen, W.; Fu, Z.; Yang, D.; Deng, J. Single-image depth perception in the wild. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016; pp. 730–738.