International Journal of Geo-Information

MDPI

Article

A Dual-Path and Lightweight Convolutional Neural Network for High-Resolution Aerial Image Segmentation

Gang Zhang [1,2], Tao Lei [1,*], Yi Cui [1] and Ping Jiang [1]

[1] Institute of Optics and Electronics, Chinese Academy of Sciences, P.O. Box 350, No.1 Guangdian Avenue, Chengdu 610209, China; zhanggang@ioe.ac.cn (G.Z.); cuiyi@ioe.ac.cn (Y.C.); jiangping@ioe.ac.cn (P.J.)

[2] University of Chinese Academy of Sciences, No. 19 (A) Yuquan Road, Beijing 100049, China

* Correspondence: taoleiyan@ioe.ac.cn

Received: 7 October 2019; Accepted: 9 December 2019; Published: 12 December 2019

Abstract: Semantic segmentation on high-resolution aerial images plays a significant role in many remote sensing applications. Although the Deep Convolutional Neural Network (DCNN) has shown great performance in this task, it still faces the following two challenges: intra-class heterogeneity and inter-class homogeneity. To overcome these two problems, a novel dual-path DCNN, which contains a spatial path and an edge path, is proposed for high-resolution aerial image segmentation. The spatial path, which combines the multi-level and global context features to encode the local and global information, is used to address the intra-class heterogeneity challenge. For inter-class homogeneity problem, a Holistically-nested Edge Detection (HED)-like edge path is employed to detect the semantic boundaries for the guidance of feature learning. Furthermore, we improve the computational efficiency of the network by employing the backbone of MobileNetV2. We enhance the performance of MobileNetV2 with two modifications: (1) replacing the standard convolution in the last four Bottleneck Residual Blocks (BRBs) with atrous convolution; and (2) removing the convolution stride of 2 in the first layer of BRBs 4 and 6. Experimental results on the ISPRS Vaihingen and Potsdam 2D labeling dataset show that the proposed DCNN achieved real-time inference speed on a single GPU card with better performance, compared with the state-of-the-art baselines.

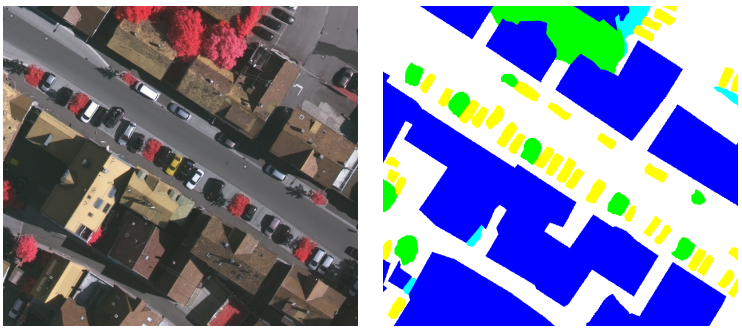
Keywords: remote sensing; convolutional neural networks; high-resolution aerial images; semantic segmentation; semantic boundaries; lightweight network

1. Introduction

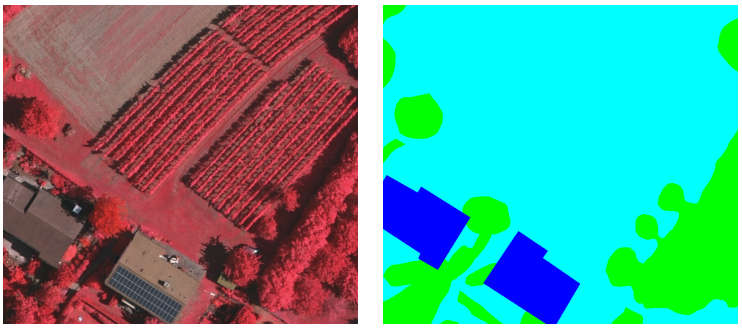
With the rapid development of remote sensing technologies, more and more high-resolution aerial images are available for us to obtain information in various domains, such as urban planning, environmental monitoring, landscape classification, disaster relief, navigation, etc. As a result, accurate and real-time semantic segmentation of high-resolution aerial images is of great significance and receives more attention. Some traditional image segmentation methods, such as watershed algorithm [1], graph cuts [2], and random forest [3], are used to classify high-resolution aerial images. They usually need artificially setting thresholds and interaction controls and are sensitive to noises, thus they cannot provide accurate semantic segmentation results.

In the past few years, deep learning methods, especially the Fully Convolutional Network (FCN) [4], have significantly promoted the development of semantic segmentation. Some deep learning based semantic segmentation methods [4–10] developed for natural images have been applied to high-resolution aerial images and achieved good performance. However, the features extracted

by these methods are not good at discriminating: (1) two objects which are classified into the same semantic label but with different appearances, named intra-class heterogeneity, as shown in Figure 1a, where the houses (or cars) have different shapes, sizes, and colors, but they belong to the same semantic label; and (2) two adjacent objects which are categorized into two different semantic labels but with similar appearances, named inter-class homogeneity, as shown in Figure 1b, where the low vegetation and trees are similar in colors, but their semantic labels are distinct. To tackle these two challenges, we need to consider each category of pixels as a whole, instead of assigning semantic label to each single pixel independently. To address the intra-class heterogeneity issue, we need to combine the multi-level and global context features to encode the local and global information, which can learn the discriminative and effective features to correctly categorize variant objects belonged to the same semantic label. Semantic boundaries can detect the feature variations on adjacent objects with similar appearance but different semantic labels. We can integrate it into the training process to help the network to learn the discriminative features to enlarge the inter-class differences. Based on the above two points, we propose a novel Deep Convolutional Neural Network (DCNN) that contains a spatial path and an edge path to tackle the problems of intra-class heterogeneity and inter-class homogeneity in high-resolution aerial images simultaneously.



(**a**) intra-class heterogeneity



(**b**) inter-class homogeneity

Figure 1. Examples of intra-class heterogeneity and inter-class homogeneity in high-resolution aerial images: (**a**) houses (or cars) have different shapes and colors, but they belong to the same semantic label; and (**b**) low vegetation and trees are similar in appearance, but they belong to two different semantic labels.

In remote sensing applications, one of the major challenges is automatically extracting urban objects from data acquired in real-time. To the best of our knowledge, most of the proposed semantic segmentation networks for high-resolution aerial images are focused on improving the accuracy with little attention paid to computational efficiency. These networks often have huge number of parameters and long inference time. In this work, we also take the computational efficiency into consideration for semantic segmentation of high-resolution aerial images. The feature extractor of the proposed DCNN is inspired from MobileNetV2 [11], which provides an efficient classification network. We modify

it to improve the prediction accuracy by introducing atrous convolution and discarding the strided convolution in the deeper convolutional layers.

The remainder of this paper is arranged as follows. Section 2 gives an overview of related approaches for high-resolution aerial images segmentation. Section 3 describes the proposed method in detail. Section 4 presents the experimental results of our proposed method and comparisons with other methods. The discussion of obtained results is presented in Section 5. Finally, conclusions are drawn in Section 6.

2. Related Work

In computer vision, while convolutional networks have been used for a long time, their success was limited by the amount of available training images and high-performance computing resources [12]. Since AlexNet [13] won the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012, DCNN has become the mainstream research method in the field of computer vision and achieved great success in various applications, such as image classification, object detection, semantic segmentation, object tracking, face recognition, etc.

The goal of semantic segmentation is to assign each pixel in an image with a semantic label [14]. FCN [4] is considered a milestone in deep learning techniques for semantic segmentation, since it demonstrates how DCNNs could be trained end-to-end to solve this problem, efficiently learning how to produce dense pixel-level predictions for input images of arbitrary sizes. SegNet [5] introduces an encoder–decoder architecture for semantic segmentation. The encoder extracts features via convolution, max pooling, and activation layers, while storing the index of each max pooling window. The decoder is similar to the encoder, upsampling the input, using indices stored from the encoding stage. U-Net [6] is a U-shaped architecture, which is a symmetric DCNN and uses skip connections between the downsampling path and the upsampling path. It combines different levels of context information to predict a good segmentation map. DeepLab [7,8] introduces atrous convolution in DCNN to effectively enlarge the receptive fields without increasing the number of network parameters. Atrous Spatial Pyramid Pooling (ASPP) employs multiple parallel atrous convolutional layers with different dilation rates to exploit multi-scale features, thus capturing objects as well as image context at multiple scales. In DenseNet [9], each layer receives feature maps from all preceding layers and passes on its output feature maps to all subsequent layers. Therefore, the loss could be propagated to earlier layers directly, and the vanishing-gradient problem is alleviated. GCN [10] employs large convolutional kernels and effective receptive fields to address the classification and localization issues for semantic segmentation. PSPNet [15] is a DCNN to exploit the global context information of an image by different-region-based context aggregation through the pyramid pooling module. It concatenates the feature extraction layers and the upsampled pyramid pooling layers, combining local and global context information together. These state-of-the-art models employ the following technologies that are widely used in semantic segmentation algorithms: (1) skip connections between lower convolutional layers and higher convolutional layers to fuse features of different levels for better pixel-level labeling; (2) atrous convolution to enlarge receptive fields without increasing computational parameters; and (3) global pooling convolutional layer to guide the location of objects. These technologies are integrated into our method to tackle the challenge of intra-class heterogeneity problem. Holistically-nested Edge Detection (HED) [16] is an edge detection DCNN that adopts FCN architecture with multiple side-outputs for deeply supervised learning. In this paper, we use a HED-structured sub-network to extract semantic boundaries for deep supervision of our network in the learning process, which helps to deal with the problem of inter-class homogeneity.

In remote sensing research, DCNNs have been recently employed for high-resolution aerial images segmentation [17]. Kampffmeyer et al. [18] focused on small object segmentation through measuring the uncertainty of DCNNs. This method achieved high overall accuracy as well as good performance for small objects segmentation. Guo et al. [19] exploited FCN with atrous convolution to perform semantic segmentation for high-resolution remote sensing images. They

used graph-based segmentation and selective search method to augment the training data and conditional random fields(CRF) to refine the segmentation results. Chen et al. [20] proposed a DCNN based on DeepLabv3 [8], which adopted modified ASPP, a fully connected fusion path and pre-trained encoder for high-resolution remote sensing images segmentation. Liu et al. [21] introduced an effective method to detect manhole cover objects in remote sensing images. They designed two sub-networks: a multi-scale output network for manhole cover object-like edge generation, and a multi-level convolution matching network for object detection based on fused feature maps. Schuegraf and Bittner [22] proposed two parallel U-Net-like [6] DCNNs, which merged depth and spectral information. The output of the two DCNNs were combined together for binary building mask generation. Panboonyuen et al. [23] presented a DCNN based on GCN [10], which adopted more convolutional layers, channel attention module, and domain specific transfer learning. Liu et al. [24] proposed a FCN based DCNN, in which a spatial residual inception (SRI) module was employed to capture and aggregate multi-scale contexts for semantic segmentation by fusing muti-level features. Pan et al. [25] presented a DCNN for building extraction from high-resolution aerial images, which composed of a U-Net, channel attention mechanisms, and an adversarial network. Benjdira et al. [26] designed an unsupervised algorithm using Generative Adversarial Networks (GANs), which demonstrated improved performance when passing from the ISPRS Potsdam 2D labeling dataset to the ISPRS Vaihingen 2D labeling dataset. Pan et al. [27] presented a novel Dense Pyramid Network (DPN) based on DenseNet to extract and take full advantage of features. They used group convolutions to extract feature maps of each channel of multi-sensor data and channel shuffle to enhance the representation ability of the network. To deal with the class imbalance problem, they adopted the median frequency balanced focal loss. Yao et al. [28] proposed the dense-coordconv network (DCCN) to reduce the loss of spatial features and strengthen object boundaries. This method adopted DenseNet as backbone, putting coordinate information into feature maps. Liu et al. [29] designed ScasNet to improve the accuracy of manmade objects and intricate fine-structured objects by sequential global-to-local context aggregation in a self-cascaded manner. Wu et al. [30] presented four stacked fully convolutional networks (SFCNs) and feature alignment framework for multi-label land-cover segmentation. However, training the entire network was time-consuming due to huge network architecture. Marmanis et al. [31] trained a DCNN to extract scale-dependent class boundaries, and then used it with color and DSM information as input to the FCN to obtain the semantic labels. Although it achieved high accuracy, it was computationally complex and time-consuming.

3. Proposed Method

In this section, we introduce the details of our proposed DCNN for high-resolution aerial image segmentation. Instead of regarding the high-resolution aerial image segmentation task as a single and independent problem, we formulate it as a multi-task learning framework by exploring the complementary information, which can predict the results of semantic labels and boundaries simultaneously. A semantic label prediction path is designed to tackle the intra-class heterogeneity problem. It combines the multi-level and global context features to encode the local and global information to learn the discriminative and effective features for the two objects with different appearances but same semantic label. A boundary prediction path is designed for guiding the process of feature learning to differentiate the adjacent objects with similar appearance but different semantic labels. The proposed DCNN jointly trains and refines the semantic and boundary information in a unified network. Basically, the predictions of semantic labels and boundaries are both pixel-wise classification tasks, which need to extract the feature maps first. Our DCNN is thus constructed on a common feature extraction network, which first learns common representations using shared convolutional layers and then appends two parallel paths with respect to multi-level spatial features fusion and semantic boundaries detection. At the same time, we need to consider the compute efficiency of the DCNN for real-time applications. Therefore, we adopt a high-performance lightweight network architecture—MobileNetV2—as our basic feature extraction network.

3.1. Network Architecture

The overall architecture of our proposed DCNN is shown in Figure 2. It is an encoder–decoder network structure. The backbone of the encoder is based on the MobileNetV2 with two aspects of improvement: (1) replacing the standard convolution in the last four Bottleneck Residual Blocks (BRBs) with atrous convolution; and (2) removing the convolution stride of 2 in the the first layer of BRBs 4 and 6. The decoder contains a spatial path and an edge path. The spatial path combines the multi-level features and the global context information stage-by-stage to refine the semantic information. The edge path is a HED-like [16] network, which employs deep supervision at each side-output layer. The parameters of MobileNetV2 are shared and updated for the spatial path and the edge path jointly, while the parameters of the two individual paths are updated independently for inferring the probability of semantic labels and boundaries, respectively. Specifically, the feature maps predicted from BRBs 3–7 in MobileNetV2 are fed into two different paths (green and blue arrows shown in the figure) in order to acquire the segmentation masks of semantic objects and boundaries at the same time.

The detail descriptions of the encoder and decoder are given in the following subsections.

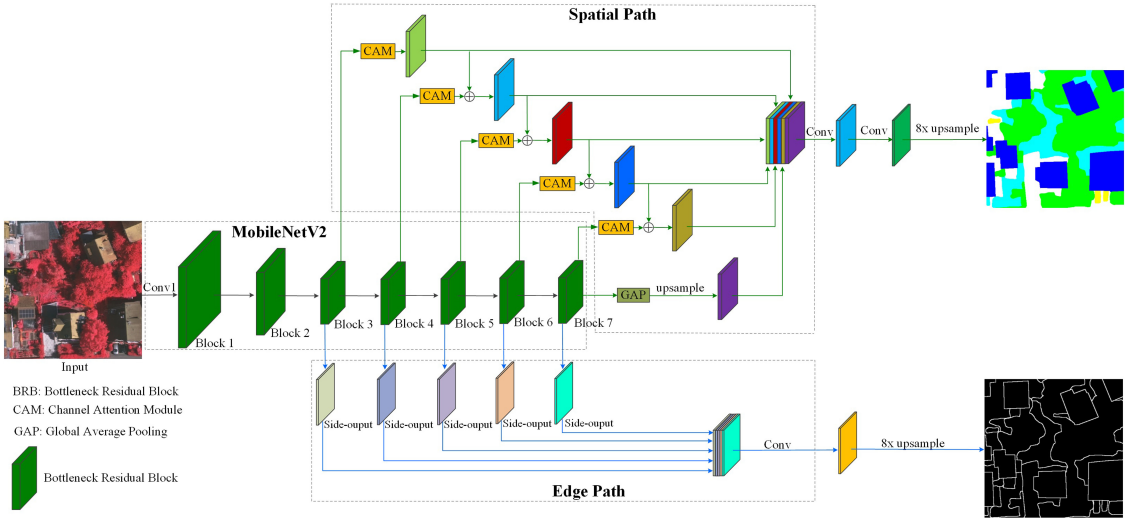


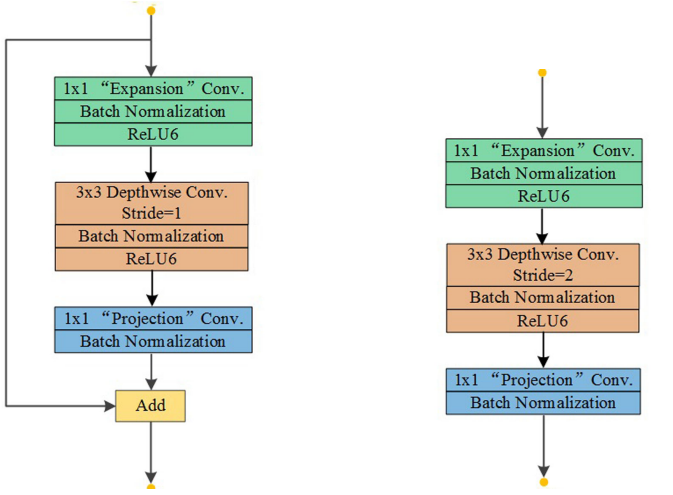
Figure 2. The architecture of the proposed network. Given an input image, we use the modified MobilenetV2 to extract the shared feature maps. Then, two paths are appended to capture semantic context and boundary context while simultaneously generating semantic segmentation maps and edge score maps.

3.2. Encoder

The encoder extracts features from the input image. The basic structure of the encoder is similar to the original MobileNetV2, except we remove its fully-connected layers, and make two modifications to improve the network performance. Firstly, to effectively enlarge the receptive fields, we replace the standard convolution in BRBs 4–7 with atrous convolution applying strides (holes) of 2, 4, 8, and 16, respectively. Secondly, to acquire more detailed context information, we change the stride of the first layer of each block sequence in BRBs 4 and 6 from 2 to 1.

3.2.1. MobileNetV2 with Multi-Level Contextual Features

MobileNetV2 is an efficient and lightweight DCNN architecture that has demonstrated the state-of-the-art performance on multiple tasks and benchmarks in real-time applications. The basic building block is BRB, which is a bottleneck depthwise separable convolution with residuals. The structure of a typical BRB is shown in Figure 3. It is composed of three sublayers: a 1×1 "Expansion" layer with ReLU6, a 3×3 depthwise layer with ReLU6, and a 1×1 "Projection" layer without any non-linearity.



(**a**) BRB with Stride=1 (**b**) BRB with Stride=2

Figure 3. The basic structure of BRB. There are two types of blocks: (**a**) residual block with stride of 1; and (**b**) block with stride of 2 for downsampling.

The BRB architecture applies a non-linear function (ReLU6 [32]) that converts the input to the output by expanding and projecting channels. The 1×1 "Expansion" layer is a 1×1 convolution to expand the number of channels input to the 3×3 depthwise convolution. The "Expansion" layer always has more output channels than its input channels. The ratio between the number of output channels and the number of input channels is given by expansion factor. The default expansion factor is 6. The 3×3 depthwise layer performs lightweight depthwise convolution [33] by applying a single convolution operation per input channel. The 1×1 "Projection" layer makes the number of output channels the same as the input ones. There is residual connection [34] between input channels and output channels if the convolution stride equals 1, which improves the ability of gradient propagation across multiplier layers. Each layer has batch normalization and the activation function ReLU6. However, the output of the 1×1 "Projection" layer does not have an activation function applied to it, because appending a non-linearity after it destroys useful feature information.

The MobileNetV2 used for feature extraction in our proposed DCNN contains a fully convolution layer with 32 channels, followed by seven BRBs described in Table 1. Each BRB contains n basic blocks, as shown in Figure 3. We apply atrous convolution through BRBs 4–7 with strides 2, 4, 8, and 16 to enlarge the receptive fields and capture context information at different levels. To get more detailed context information, we modify the stride of the first layer in BRBs 4 and 6 from 2 to 1 in the original MobileNetV2.

Table 1. The architecture of MobileNetV2 used for feature extraction in our proposed DCNN. Each line describes a sequence of 1 or more identical layers. *t*: expansion factor; *c*: the number of output channels. *n*: the number of repeated layers; *s*: stride of the first layer of each sequence, all others use stride 1; *as*: stride of the atrous convolution.

Block Index	Operator	*t*	*c*	*n*	*s*	*as*
0	Conv2D	-	32	1	2	1
1	BRB	1	16	1	1	1
2	BRB	6	24	2	2	1
3	BRB	6	32	3	2	1
4	BRB	6	64	4	1	2
5	BRB	6	96	3	1	4
6	BRB	6	160	3	1	8
7	BRB	6	320	1	1	16

3.2.2. Atrous Convolution

In the application of DCNNs for semantic segmentation, max-pooling and striding convolution are employed to reduce the memory occupancy and enlarge the receptive fields. As a result, the resolution of output feature maps reduces significantly. Although "deconvolution" [4] layers or upsampling operations could be used, the loss of the spatial information, especially the boundary information, is too large. Atrous convolution, also called dilated convolution, has been shown that it can enlarge the receptive fields without reducing the image resolution [7]. In the case of 1D atrous convolution, given an input signal $x(i)$ with a filter $w(k)$ of length K, the output of $y(i)$ is defined as:

$$y(i) = \sum_{k=1}^{K} x(i + r \cdot k)w(k) \tag{1}$$

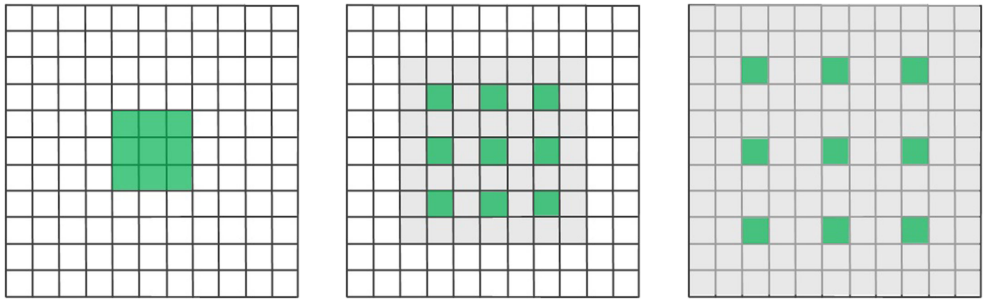
where r is the dilation rate that indicates the stride at which we sample the input signal. When $r = 1$, it is standard convolution. In atrous convolution, the convolution kernel is expanded by the dilation rate, and $r - 1$ zeros are inserted along the space dimension between the adjacent weights to create a sparse filter. The size of the receptive field can be calculated as:

$$rf = ((k + 1) \cdot (r - 1) + k)^2 \tag{2}$$

Figure 4 gives a simple example of 2D atrous convolution. Figure 4a shows a standard 3×3 convolution, a special case for dilation rate = 1, covering a 3×3 receptive field. Figure 4b demonstrates a 3×3 atrous convolution with dilation rate = 2. While the convolution kernel size is still 3×3, the receptive field is increased to 7×7. Figure 4c illustrates a 3×3 atrous convolution with dilation rate = 3. Its receptive field is 11×11, but the actual number of parameters is still 3×3.

3.3. Decoder

The function of the decoder is to predict the semantic label of each pixel at the same resolution of the input image. It constructs the pixel-wise semantic label from the feature maps extracted by the encoder. The feature maps output from the encoder are in low resolution and has many channels, with each channel representing a particular feature. As shown in Figure 2, the spatial path is used to merge these feature maps by a series of convolutional layers and Channel Attention Module (CAM). The edge path is employed to detect semantic boundaries by deep supervision. Finally, $8\times$ bilinear upsampling are used on feature maps output from the spatial path to recover the resolution.



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

Figure 4. Example of atrous convolution of 3 × 3 kernel size with different dilation rates: (**a**) atrous convolution with dilation rate = 1, also known as standard convolution, which has a receptive field of 3 × 3; (**b**) atrous convolution with dilation rate = 2, which has a receptive field of 7 × 7; and (**c**) atrous convolution with dilation rate = 3, which has a receptive field of 11 × 11. While the receptive field grows exponentially, the number of parameters associated with each filter is identical.

3.3.1. Spatial Path

The intra-class heterogeneity problem is mainly because of the lack of context information. Therefore, we need the multi-level receptive fields and context information to refine the spatial information. The outputs of BRBs 3–7 in MobileNetV2 have different receptive fields. In the lower block, the network output features with fine spatial information, but it has poor semantic information due to its small receptive fields and without guidance of spatial context. While in the upper block, it has good semantic information because of its large receptive fields, but the feature maps are coarse. To sum up, the lower block provides finer spatial predictions, while the upper block generates more accurate semantic predictions. Therefore, we introduce the spatial path to take advantage of these blocks for better predictions. In our spatial path, we sum up the features of adjacent blocks stage-by-stage, as shown in Figure 2. Then, these layers are concatenated together and fed to a convolution layer to further fuse features of different receptive fields. However, different scale of receptive fields provide features with different discrimination, resulting in inconsistent semantic segmentation results. Therefore, to generate identical semantic label for one certain class, we need to use more discriminative features. Here, we adopt the high semantic information generated by global average pooling from BRB 7. With this global context information, we introduce the strongest consistency constraint into the network as a guidance.

Furthermore, to refine the features of each BRB, we propose a specific CAM inspired by SENet [35]. As shown in Figure 5, CAM is designed to assign a weight factor for each feature channel, which could guide the feature learning adaptively and assign important channels with higher weights. It employs the global average pooling on each feature channel to encode an attention vector, which is used to re-weight the original features.

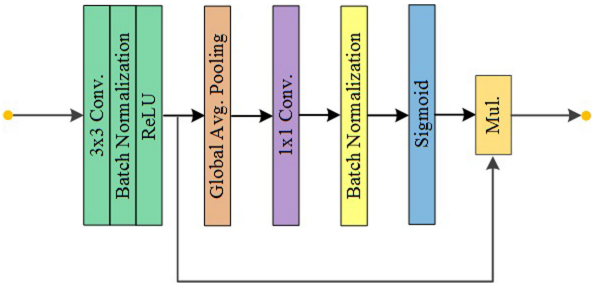


Figure 5. The structure of the Channel Attention Module (CAM).

3.3.2. Edge Path

In the task of high-resolution aerial image segmentation, it is hard to discriminate two classes with similar appearance when they are spatially adjacent. To improve the discriminative ability of the network on this problem, we introduce the edge path (as shown in Figure 2) to guide the feature learning for semantic segmentation task. To extract the semantic boundaries accurately, we adopt the network architecture proposed in HED [16]. In our proposed edge path, we attach side-outputs to the last five blocks of MobileNetV2 for semantic boundary detection. We apply deep supervision at each side-output block to learn multi-level representations for semantic boundary predictions. In detail, 1×1 convolutional layers with one channel are appended to each of the last five blocks of MobileNetV2 to generate semantic boundary score maps. Then, these score maps are concatenated together and fed to a 1×1 convolutional layer with one channel to output the final score map. This semantic boundary detection network could distinguish the semantic boundaries between two adjacent objects that belong to different classes, making the inter-class features distinction as great as possible.

3.4. Lost Function

In this paper, we use the Softmax loss to supervise the training of the spatial path, and adopt the binary cross entropy loss to supervise the training of the edge path. The training of the network is formulated as a per-pixel classification problem regarding the groundtruth segmentation masks including semantic objects and their boundaries. Therefore, the loss function of our DCNN can be written as:

$$L_{total} = \alpha \cdot L_{spatial} + \beta \cdot (L_{edge} + \sum_{k=1}^{K} L_{side}^{k}) \tag{3}$$

where $L_{spatial}$ is defined as the Softmax loss of the spatial path, L_{edge} and L_{side}^{k} denote the binary cross entropy loss of the fused edge and the side-output edges in the edge path, respectively. The number of the edge side-output, K, is 5. α and β are the balance weights.

3.5. Network Training

In this section, we introduce our training details of the proposed network.

3.5.1. Transfer Learning

In remote sensing domain, due to the expensive cost and complicated acquisition process, there is insufficient training data with accurate annotations for semantic segmentation task. Compared with the limited data in remote sensing, much more training data of natural images are available. Studies (e.g., [36]) have proven that transfer learning in DCNNs can alleviate the problem of insufficient training data. The parameters learned from lower layers in a DCNN can be shared across tasks, while those in higher layers are specific to different tasks. Therefore, transferring the parameters learned from other domains could help reduce overfitting and make the network converge quickly.

We utilized a two-step training procedure to train our DCNN. Firstly, the original MobileNetV2, which is developed for image classification task, was trained on the ImageNet dataset [37]. Then, the encoder was loaded with the pre-trained parameters in the first step, while the rest layers randomly initialized with Kaiming initialization [38]. Finally, we fine-tuned the whole DCNN on the ISPRS 2D semantic labeling dataset [39] in an end-to-end manner.

3.5.2. Implementation Details

We trained the proposed DCNN using stochastic gradient descent (SGD) [13] with batch-size 16, base learning rate 0.01, momentum 0.9 and weight decay 0.0005. The learning rate of the pre-trained weights was set as half of the base learning rate. We trained the DCNN for 50 epochs and divided

the learning rate by 10 after 25, 35, and 45 epochs. As for α and β in Equation (3), we finally used the values of 50 and 0.0025, respectively, after a series of experiments.

As the images of ISPRS 2D semantic labeling dataset are very high-resolution, we could not feed them directly into our DCNN. We randomly cropped all the images into 256 × 256 patches as inputs of each epoch. To avoid overfitting, data augmentation was employed. We used mean subtraction and random cropping on the input image patches to augment the dataset in the training process. For the label of semantic boundaries, we extracted the boundaries from the semantic segmentation's groundtruth with the MATLAB *imgradient* function.

Our DCNN was implemented under the pytorch [40] framework. All experiments were executed on a Linux PC with 64 bit Ubuntu 18.04, CPU i7-5930K with 64 GB memory, and a Nvidia Geforce GTX TITAN X GPU with 12 GB memory.

4. Experiments and Evaluations

4.1. Dataset

We evaluated the proposed network on the benchmark dataset of the ISPRS 2D semantic labeling dataset [39]. It is comprised of very high-resolution aerial images over two cities, Vaihingen and Potsdam, in Germany. The semantic labels of the dataset contain six classes: impervious surfaces (e.g., roads), buildings, low vegetation, trees, cars, and clutters.

4.1.1. ISPRS Vaihingen

The Vaihingen dataset consists of 33 images with an average size of approximately 2100 × 2100 at a spatial resolution of 9 cm. Sixteen tiles were used for training, and the other 17 tiles were used for testing. All tiles have near infrared (IR), red (R), and green (G) color channels, together with Digital Surface Model (DSM) extracted from the Lidar point cloud and normalized DSM (nDSM). In the 16 training tiles, 12 tiles (1, 3, 5, 7, 13, 17, 21, 23, 26, 28, 32, and 37) were chosen for training and the remaining 4 tiles (11, 15, 30, and 34) for validation. The other 17 tiles (2, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 27, 29, 31, 33, 35, and 38) were used for testing. Note that DSM and nDSM data in this dataset were not used.

4.1.2. ISPRS Potsdam

The Potsdam dataset is comprised of 38 images with size of 6000 × 6000 at a spatial resolution of 5 cm. Twenty-four tiles composed the training set, and the other 14 tiles were preserved as test set. Each tile has the following bands: IR, R, G, and blue (B) color channels; DSM; and nDSM. We selected 18 tiles (2_10, 2_11, 3_10, 3_11, 4_10, 4_11, 5_10, 5_11, 6_7, 6_8, 6_9, 6_10, 6_11, 7_7, 7_8, 7_9, 7_10, and 7_11) for training and 6 tiles (2_12, 3_12, 4_12, 5_12, 6_12, and 7_12) for validation in the training set. The other 14 tiles were reserved for testing. Note that only the three-band IRRG images extracted from raw four-band IRRGB data were used, and DSM and nDSM data on this dataset were not used.

4.2. Evaluating Metrics

To measure the performance of different DCNNs, we used the following two metrics: Overall accuracy and $F1$. Let TP denote the number of true positives, TN denote the number of true negatives, FP denote the number of false positives, and FN denote the number of false negatives. Overall accuracy is a metric that takes into account all correctly classified pixels indistinctly. It can be written as [41]:

$$OA = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$F1$ is considered as the harmonic mean of precision and recall. It is defined as [41]:

$$F1 = 2 \times \frac{precision \cdot recall}{precision + recall} \tag{5}$$

where $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$.

4.3. Ablation Study

In this subsection, we decompose our method to study how each component affects the segmentation performance. We used the unmodified MobileNetV2 (denoted as MNetV2) and spatial path (described in Section 3.3.1) as our base semantic segmentation network. Then, we evaluated whether the modified MobileNetV2 (denoted as MNetV2*, described in Section 3.2.1), and edge path (described in Section 3.3.2) can bring benefit to the final segmentation performance. As shown in Table 2, we can observe that the performance of the MNetV2* achieved higher accuracy than the MNetV2, which demonstrates that the MNetV2* can preserve more useful information and provide contextual detail information. This is due to that removing the stride of the first layer in BRBs 4 and 6 and employing atrous convolution in BRBs 4–7 provide more detailed information and enlarge the receptive fields. They improved the overall accuracy from 86.09% to 88.72%. Especially, the $F1$ score value of the car category was improved from 59.43% to 82.98% by a large margin. The edge path is employed to address the inter-class homogeneity problem. Under the guidance of deep supervisory signal from the edge path, the network could discriminate the semantic boundaries between two adjacent objects. Finally, this improved the overall accuracy from 88.72% to 89.61%.

Table 2. Experimental results on the ISPRS Vaihingen test tiles. MNetV2: feature extraction backbone of unmodified MobileNetV2; MNetV2*: modified MNetV2 described in Section 3.2.1; SP: spatial path; EP: edge path.

Methods	Imp. Surf.	Building	Low Veg.	Tree	Car	Avg. F1	OA
MNetV2+SP	88.10	90.87	79.30	86.98	59.43	80.94	86.09
MNetV2*+SP	91.06	93.32	81.47	88.36	82.98	87.44	88.72
MNetV2*+SP+EP	92.15	94.44	82.27	88.70	84.19	88.35	89.61

Figure 6 presents the visual comparisons of the segmentation results on the ISPRS Vaihingen test tiles. The first row is an image patch with building roofs of different shapes. We can observe that the MNetV2+SP confused similar manmade objects, such as the building roofs and roads, and it obtains inaccurate localization for buildings. The MNetV2*+SP could predict the shapes of the building roofs more accurately and distinguish the building roofs and roads with similar colors. With the help of the edge path, the MNetV2*+SP+EP could label the contours of building roofs more clearly. The second row of Figure 6 is an image patch with highly inconsistent cars. The MNetV2+SP labeled all the cars together, while the MNetV2*+SP and the MNetV2*+SP+EP could discriminate almost all of the cars clearly. For the four cars in the top-right corner in the image patch, the MNetV2*+SP+EP could detect their contours by the help of the edge path and label them one by one, while the MNetV2*+SP could not separate them completely. Low vegetation and trees are prone to be confused by DCNNs due to their similar colors, as shown in the third and fourth rows of Figure 6. The MNetV2+SP and the MNetV2*+SP mislabeled some low vegetation areas as trees, while the MNetV2*+SP+EP could provide a relatively proper segmentation results for these plants under the guidance of semantic boundaries detected by the edge path. The clutter category is hard to be properly labeled by DCNNs because it contains a variety of different categories of objects, as shown in the fifth row of Figure 6. We can observe that the MNetV2+SP could not give a good prediction. The results of the MNetV2*+SP are relatively better, while they are still less accurate. The MNetV2*+SP+EP produced more accurate and robust segmentation results. Figure 6g is the semantic boundaries generated by the edge path of our proposed DCNN. We can observe that our DCNN could predict accurate and clear object contours while suppressing most of the scattered and minor edge responses inside the objects.

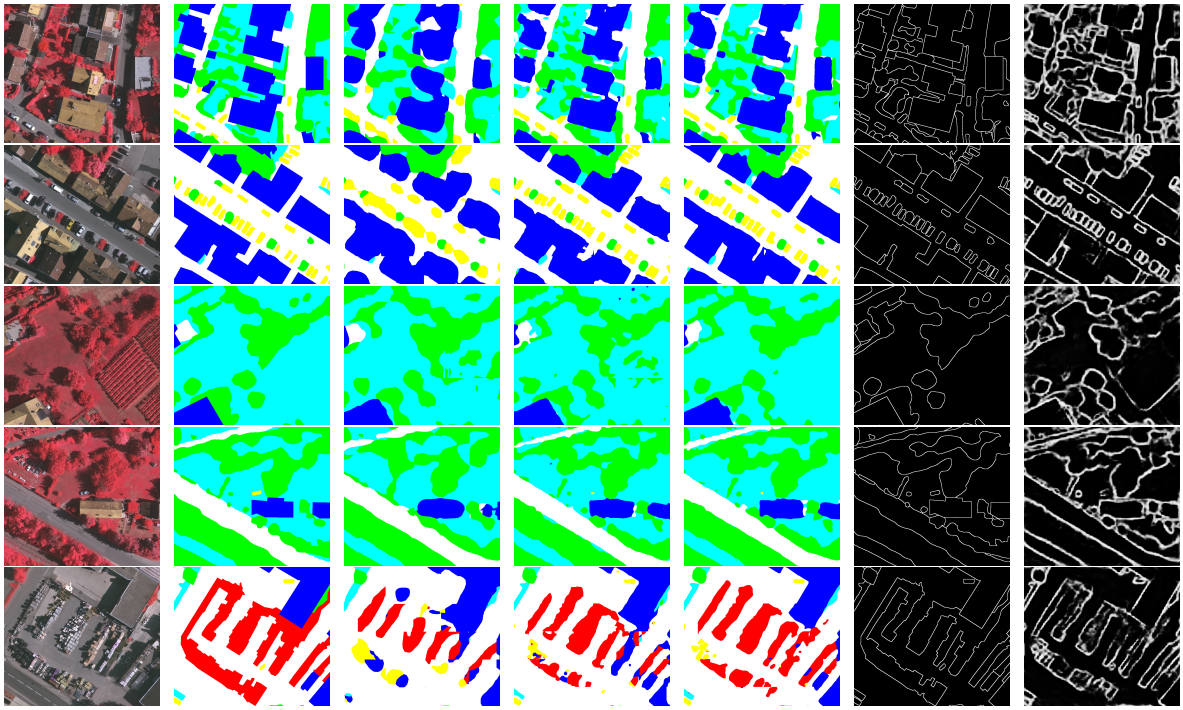


Figure 6. Examples of semantic segmentation results on the ISPRS Vaihingen test tiles: (**a**) raw images; (**b**) the groundtruths; (**c**–**e**) the segmentation results of MNetV2+SP, MNetV2*+SP, and MNetV2*+SP+EP, respectively; (**f**) the semantic boundaries extracted from groundtruths by MATLAB *imgradient* function; and (**g**) the predicted semantic boundaries of our proposed network.

4.4. Comparing with Other Methods

To verify the performance, we evaluated the proposed DCNN on the test tiles of ISPRS 2D semantic labeling dataset, and compared it with other widely-used lightweight models listed below:

(1) **ICNet**: Zhao et al. [42] introduced an Image Cascade Network (ICNet) that incorporates multi-resolution branches under proper label guidance to reduce computations, and further fuses these branches to generate the final results.

(2) **ESPNet**: Mehta et al. [43] proposed ESPNet for semantic segmentation of high-resolution images. It is based on the Efficient Spatial Pyramid (ESP) module, which is computationally efficient.

(3) **BiSeNet**: BiSeNet [44] designs a spatial path with small stride to preserve the spatial information and generate high-resolution feature maps, and a context path with fast downsampling to obtain large receptive fields parallelly. In the pursuit of better accuracy without loss of speed, a Feature Fusion Module (FFM) is employed to fuse the two paths and refine the final prediction. We used ResNet18 as the backbone of BiSeNet in the experiment.

(4) **LW_RefineNet**: LW_RefineNet [45] is a lightweight version of RefineNet [46]. It reduces the number of parameters and floating point operations in the original RefineNet by replacing the 3×3 convolutional layers with 1×1 convolutional layers and removing the Residual Convolutional Unit (RCU). We used LW_RefineNet-50 as the comparing mode.

Table 3 shows the number of parameters of the compared models. Our model has 2.3M parameters, which is bigger than ESPNet, and smaller than the others.

Table 3. The number of parameters of the compared models.

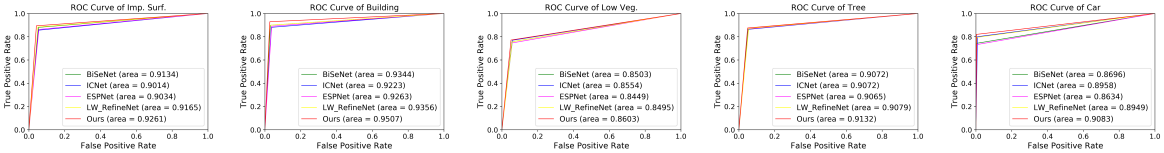
Model	ICNet	ESPNet	BiSeNet	LW_RefineNet	Ours
Parameters	26.5M	0.37M	49M	27M	2.3M

4.4.1. Comparison on the ISPRS Vaihingen Dataset

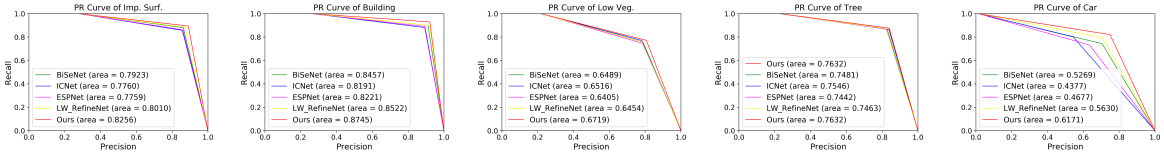
The quantitative results of the compared models are exhibited in Table 4. As shown in the table, our proposed method outperformed the others on each category $F1$ score, average $F1$ score, and overall accuracy, especially for the low vegetation category and the car category. Moreover, as shown by the ROC and PR curves in Figure 7, our method provided better performance on all categories.

Table 4. Quantitative comparison with the state-of-the-art models on the ISPRS Vaihingen test tiles (the values in bold are the best).

Methods	Imp. Surf.	Building	Low Veg.	Tree	Car	Avg. F1	OA
ICNet	88.78	90.76	81.01	88.09	67.91	83.31	87.04
ESPNet	88.84	91.02	79.98	87.51	72.35	83.94	86.85
BiSeNet	90.19	92.59	80.75	87.77	78.43	85.95	87.95
LW_RefineNet	90.59	92.82	80.34	87.56	80.32	86.33	88.01
Ours	**92.15**	**94.44**	**82.27**	**88.70**	**84.19**	**88.35**	**89.61**



(**a**) ROC Curve



(**b**) PR Curve

Figure 7. ROC and PR curves of all the comparing models on the ISPRS Vaihingen test tiles: (**a**) the ROC curve; and (**b**) the PR curve. Classes from left to right: impervious surface (Imp. Surf.), buildings, low vegetation (Low Veg.), trees, and cars.

Figure 8 gives comparisons on qualitative performance of the ISPRS Vaihingen test tiles. In the first row of fine-structured buildings, ICNet and ESPNet provided inaccurate and incomplete labeling, while BiSeNet and LW_RefineNet were relatively better. Our proposed DCNN generated more coherent segmentation results. The second row of Figure 8 is an image patch with cars of different shapes and colors, which is a representative intra-class heterogeneity problem. ICNet, ESPNet, BiSeNet, and LW_RefineNet were less effective at labeling these confusing cars separately. In contrast, our proposed method could generate good segmentation results with precise semantic boundaries. The third and fourth rows are low vegetation and trees that are similar in color, which represents the challenge of inter-class homogeneity. ICNet, ESPNet, BiSeNet, and LW_RefineNet confused these two classes and mislabeled some low vegetation areas as trees. Our network presented more accurate and robust labeling due to the employment of the edge path. The clutter category contains confusing manmade objects and is hard to label, as shown in the fifth row of Figure 8. Deep models often mislabel it into the car category due to their similar shapes and colors, and confuse it with buildings because of their similar colors. We can observe that ESPNet mislabeled more than half of the clutters into cars and buildings. The results of ICNet, BiSeNet, and LW_RefineNet were relatively good, but they still mislabeled about half of the clutters. Our DCNN presented better labeling than all the above methods. Therefore, our proposed method gave better visual quality on the ISPRS Vaihingen test tiles.

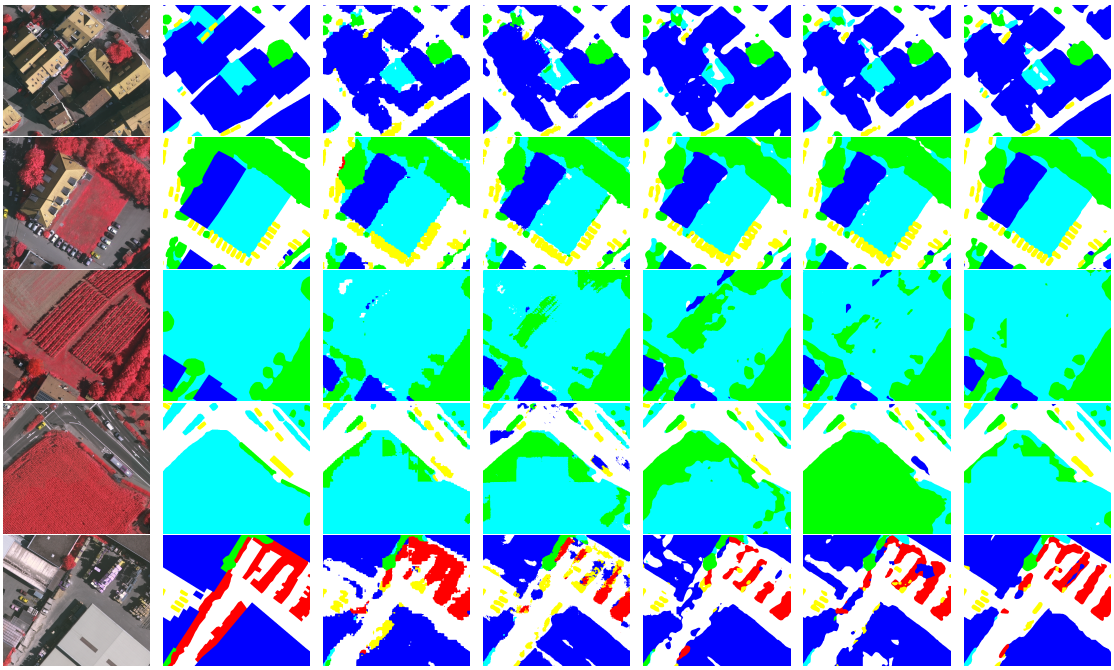


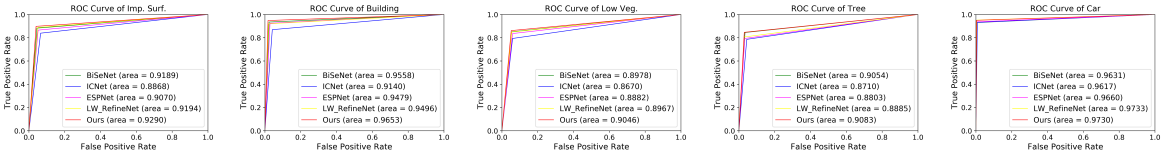
Figure 8. Examples of semantic segmentation results on the ISPRS Vaihingen test tiles: (**a**) raw images; (**b**) the groundtruths; and (**c**–**g**) the segmentation results of ICNet, ESPNet, BiSeNet, LW_RefineNet, and our network, respectively.

4.4.2. Comparison on the ISPRS Potsdam Dataset

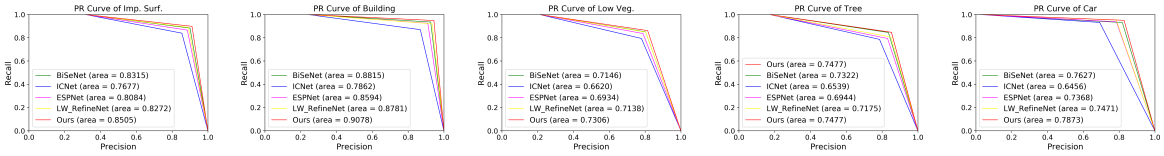
The numerical results of the compared deep models are listed in Table 5. As shown in the table, our model achieved the best performance in terms of category $F1$ score, average $F1$ score, and overall accuracy. Furthermore, the ROC and PR curves shown in Figure 9 also verify the advantage of our proposed DCNN.

Table 5. Quantitative comparison with the state-of-the-art models on the ISPRS Potsdam test tiles (the values in bold are the best).

Methods	Imp. Surf.	Building	Low Veg.	Tree	Car	Avg. F1	OA
ICNet	86.98	88.28	81.32	81.11	84.84	84.51	83.58
ESPNet	89.73	93.00	83.71	83.80	90.64	88.18	86.84
BiSeNet	91.43	94.35	85.30	86.71	92.90	90.14	88.71
LW_RefineNet	91.18	93.89	85.06	85.35	91.49	89.39	88.10
Ours	**92.63**	**95.82**	**86.21**	**87.58**	**94.12**	**91.27**	**89.93**



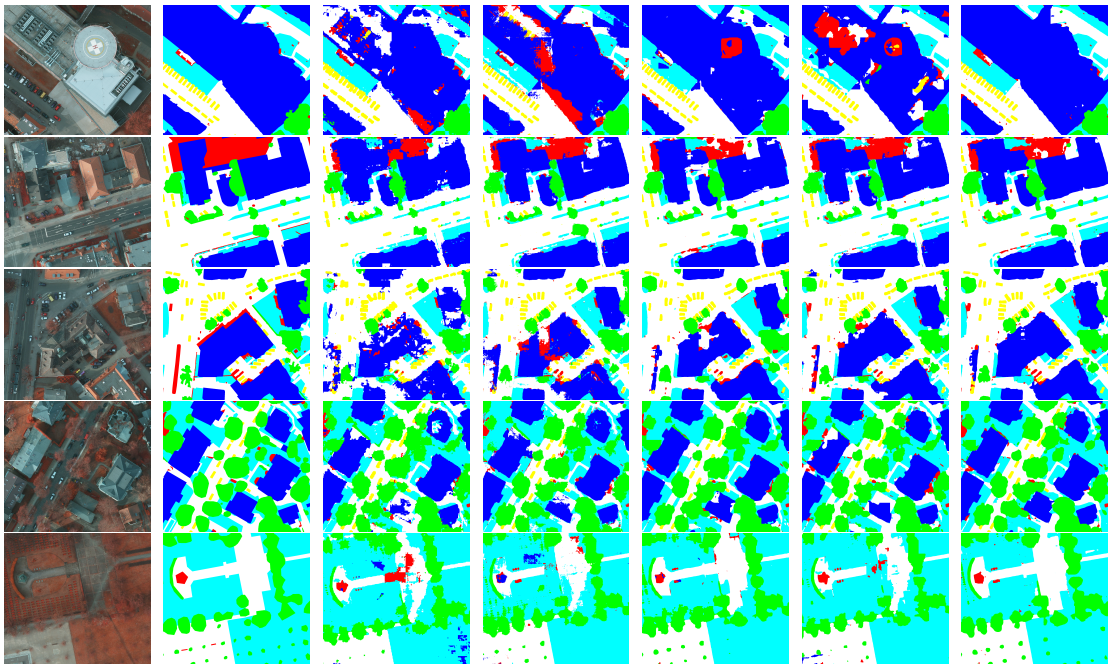
(**a**) ROC Curve



(**b**) PR Curve

Figure 9. ROC and PR curves of all the comparing models on the ISPRS Potsdam test tiles: (**a**) the ROC curve; (**b**) the PR curve. Classes from left to right: impervious surface (Imp. Surf.), buildings, low vegetation (Low Veg.), trees, cars.

Figure 10 exhibits the visual comparisons between the deep models on the ISPRS Potsdam test tiles. We can observe that all four comparison models were less good at discriminating manmade objects, such as buildings and roads, while our model could generate more precise segmentation results. For the confusing categories of low vegetation and trees, our proposed method also performed better than the others. Although there are a few flaws in the segmentation results of our model, it can provide relatively more coherent labeling and more accurate semantic boundaries.



(a) **(b)** **(c)** **(d)** **(e)** **(f)** **(g)**

Figure 10. Examples of semantic segmentation results on the ISPRS Potsdam test tiles: (**a**) raw images; (**b**) the groundtruths; and (**c**–**g**) the segmentation results of ICNet, ESPNet, BiSeNet, LW_RefineNet, and Our network, respectively.

4.4.3. Running Time

For the comparison of running time, we used the same image patch size of 1024 × 1024. Table 6 shows the number of frames per second (FPS) that can be processed by all the comparing models on a single NVIDIA Titan X GPU. Our network runs at a speed that is as competitively fast as ESPNet

while achieving a better accuracy. The inference speed indicates that it is possible to run our network for high-resolution aerial image segmentation in real-time.

Table 6. Inference speed comparison of our proposed method against other state-of-the-art methods.

Model	ICNet	ESPNet	BiSeNet	LW_RefineNet	Ours
Speed (FPS)	29.4	108.2	51.2	8.5	91.9

5. Discussions

5.1. Performance Discussion on the Benchmarks

Table 7 shows the quantitative results on the ISPRS Vaihingen and Potsdam test tiles. As shown, the F1 scores of all the categories are above 82.00%. For the categories impervious surface and buildings, the F1 scores are even greater than 92.00%, which is as accurate as manual annotations by human being. This demonstrates the effectiveness of the stage-by-stage multi-scale contexts aggregation strategy adopted in the spatial path. For the confusing categories of low vegetation and trees, the results are also competitive, which is mainly due to the employment of the edge path in our network. For the fine-structured cars, our model can provide robust segmentation results, especially on the ISPRS Potsdam dataset, achieving 94.12% of F1 score. This great performance is mainly derived from the employment of the modified MobileNetV2 and the spatial path. Overall, our method achieved 88.35% of average F1 score and 89.61% of overall accuracy on the ISPRS Vaihingen dataset and 91.27% of average F1 score and 89.93% of overall accuracy on the ISPRS Potsdam dataset. This verified the effectiveness of our proposed method on improving the segmentation accuracy of high-resolution aerial images. The quantitative results on the ISPRS Potsdam dataset is slightly better than that of the ISPRS Vaihingen dataset. The possible reasons are that the spatial resolution the ISPRS Potsdam dataset is higher than that of the the ISPRS Vaihingen dataset and the ISPRS Potsdam dataset provides more training samples.

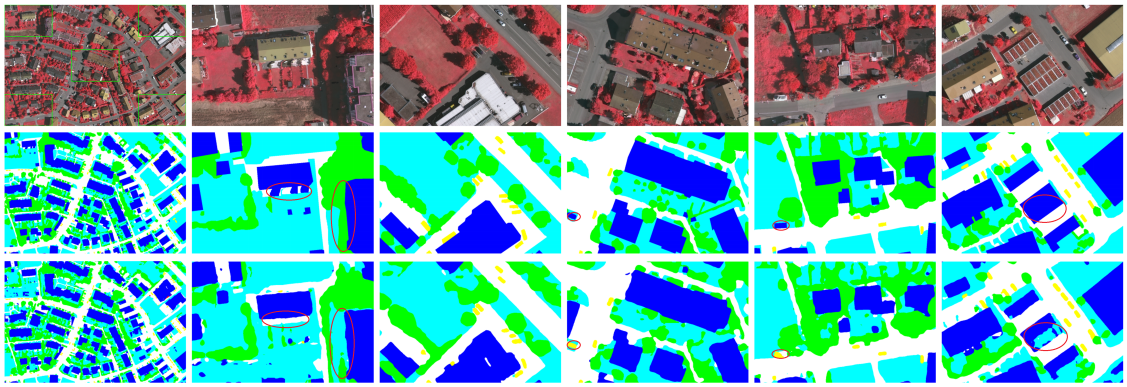
Table 7. Quantitative results on the ISPRS Vaihingen and Potsdam test tiles.

Datasets	**Imp. Surf.**	**Building**	**Low Veg.**	**Tree**	**Car**	**Avg. F1**	**OA**
ISPRS-Vaihingen	92.15	94.44	82.27	88.70	84.19	88.35	89.61
ISPRS-Potsdam	92.63	95.82	86.21	87.58	94.12	91.27	89.93

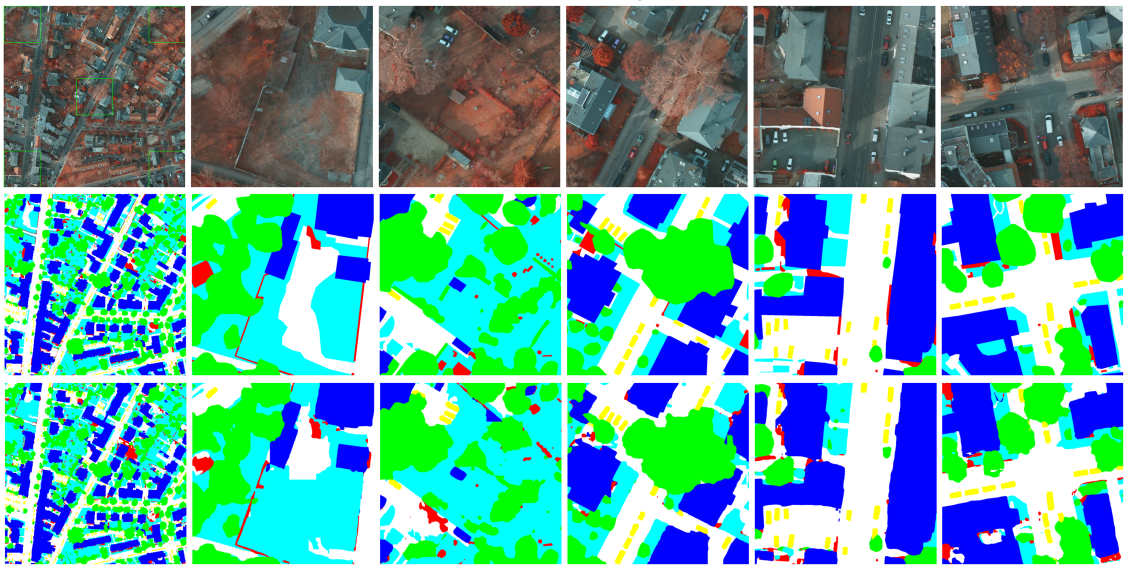
The visual performance of our proposed method on the two datasets is shown in Figure 11. We can observe that our network could obtain coherent and robust labeling results. Moreover, our method could provide labeling with smooth boundary and accurate localization, especially for the confused low vegetation and trees. For the segmentation of fine-structured buildings and cars, it can label most of them precisely with coherent contours.

Although our method achieved competitive results on the two public benchmarks, it still has limitations in dealing with high-resolution aerial images with complex backgrounds. In the ISPRS Vaihingen dataset, our network confused some parts of the buildings as impervious surface, as shown in the second column of Figure 11a. We can see that the white parts of the buildings are very difficult to identify, even by human being. While our network could distinguish low vegetation and trees preferably by employing the edge path, it still confused them in some challenge situations. As shown in the second column of Figure 11a, it mislabeled low vegetation in shadow as trees. Incorporating elevation information (such as DSM) may further improve the discriminative ability of our method on these two categories. As shown in the fourth and fifth column of Figure 11a, our network mislabeled tiny houses as cars, which are very similar in shapes and colors. As shown in the sixth column of Figure 11a, our method mislabeled some parts of the buildings as low vegetation, because the color of the rooftops are similar to the color of low vegetation. In the ISPRS Potsdam dataset, our method

could not perform well in labeling clutters, which contain variant categories of objects. The above limitations show that, with only the IRRG channels in the image, the DCNN can merely learn features based on the color and context information of the objects. When objects are similar in color and context, DCNN cannot distinguish them in some difficult situations. In remote sensing domain, since other helpful information (such as DSM) is available in some cases, we can use them to help improving the performance of our DCNN in the future.



(a) ISPRS Vaihingen



(b) ISPRS Potsdam

Figure 11. Example results of semantic segmentation on the ISPRS Vaihingen and Potsdam test tiles: (**a**) results on the ISPRS Vaihingen dataset; and (**b**) results on the ISPRS Potsdam dataset. The first column is the image tiles, the second to sixth columns are the image patches in top-left, top-right, center, bottom-left, and bottom-right, respectively. In (**a**,**b**), the first to third rows show the raw images, the corresponding groundtruths, and segmentation results, respectively.

5.2. Influence of Semantic Boundary on Segmentation Results

Figure 12 shows some examples of the predicted semantic boundaries generated by the edge path. We can observe that the edge path can provide preferable semantic boundaries between different semantic objects, which provide important guiding information for discriminating them. The examples shown in Figure 12 demonstrate that, when the semantic boundary maps in column Figure 12e are as accurate as the corresponding groudtruths in Figure 12d, the final semantic segmentation results are of high performance. However, when the semantic boundary maps fail to generate strong edge responses, or suppress disturbing responses inside the objects, on some semantic boundaries in the image (such as the areas marked by the red circles in Figure 12), the network can hardly provide

precise segmentation results, even producing incorrect ones. These results demonstrate that the edge path provides significant information for accurate semantic segmentation in our network architecture.

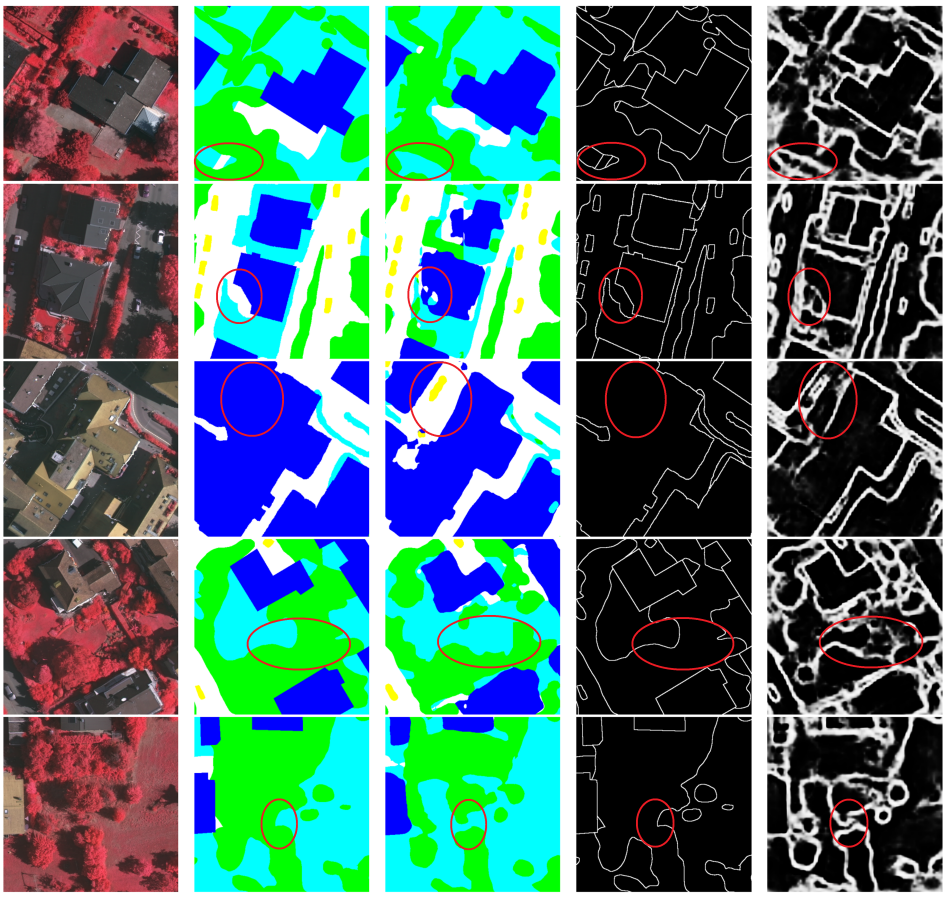


Figure 12. Visualization of the semantic boundary maps generated by the edge path: (**a**) raw images; (**b**) the groundtruths of semantic segmentation; (**c**) the segmentation results generated by our network; (**d**) the groundtruths of semantic boundary; and (**e**) the predicted semantic boundaries generated by the edge path.

6. Conclusions

In this work, a novel dual-path and lightweight DCNN is proposed for semantic segmentation in high-resolution aerial images. We design the spatial path and the edge path to address the challenges of intra-class heterogeneity and inter-class homogeneity existing in high-resolution aerial image segmentation. The spatial path makes full use of multi-level features and eliminate the loss of spatial information. The edge path is a HED-like network used to predict the semantic boundaries for deep supervision. Moreover, we enhance the computational efficiency of the proposed DCNN by employing the backbone of MobileNetV2. We modify the base MobileNetV2 in the following two aspects: (1) replacing the standard convolution in the last four BRBs with atrous convolution; and (2) removing the convolution stride of the first layer in BRBs 4 and 6. Experimental results on the ISPRS 2D semantic labeling dataset illustrate the advantages of our proposed DCNN. The proposed network was compared with other lightweight DCNNs, such as ICNet, ESPNet, BiSeNet, and LW_RefineNet, and achieved the best segmentation results, both quantitatively and qualitatively, while yielding real-time inference speed.

Author Contributions: G.Z. wrote the manuscript, designed the network, and conducted the experiments. T.L., Y.C. and P.J. contributed to the conceptual design of the experiments and reviewed and revised the paper.

Funding: This work was supported by Youth Innovation Promotion Association, Chinese Academy of Sciences (Grant No. 2016336).

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

Abbreviations

The following abbreviations are used in this manuscript:

DCNN	Deep Convolutional Neural Networks
BRB	Bottleneck Residual Block
CAM	Channel Attention Module
HED	Holistically-nested Edge Detection

References

1. Meyer, F.; Beucher, S. Morphological segmentation. *J. Vis. Commun. Image R.* **1990**, *1*, 21–46. [CrossRef]
2. Boykov, Y.Y.; Jolly, M.P. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Vancouver, BC, Canada, 7–14 July 2001; pp. 105–112.
3. Pal, M. Random forest classifier for remote sensing classification. *Int. J. Remote Sens.* **2005**, *26*, 217–222. [CrossRef]
4. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
5. Vijay, B.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495.
6. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
7. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv* **2016**, arXiv:1606.00915v1.
8. Chen, L.C.; Papandreou, G.; Schroff, F.; Hartwig, A. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
9. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
10. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network. *arXiv* **2017**, arXiv:1703.02719.
11. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. *arXiv* **2018**, arXiv:1801.04381.
12. Egmont-Petersen, M.; de Ridder, D.; Handels, H. Image processing with neural networks—A review. *Pattern Recogn.* **2002**, *35*, 2279–2301. [CrossRef]
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
14. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv* **2017**, arXiv:1704.06857.
15. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. *arXiv* **2016**, arXiv:1612.01105.
16. Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3–18.
17. Zhu, X.; Tuia, D.; Mou, L.; Xia, G.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Trans. Geosci. Remote Sens.* **2017**, *5*, 8–36. [CrossRef]

18. Kampffmeyer, M.; Salberg, A.B.; Jenssen, R. Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1–9.
19. Guo, R.; Liu, J.; Li, N.; Liu, S.; Chen, F.; Cheng, B.; Ma, C.Pixel-wise classification method for high resolution remote sensing imagery using deep neural networks. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 110. [CrossRef]
20. Chen, G.; Li, C.; Wei, W.; Jing, W.; Woźniak, M.; Blažauskas, T.; Damaševičius, R. Fully Convolutional Neural Network with Augmented Atrous Spatial Pyramid Pool and Fully Connected Fusion Path for High Resolution Remote Sensing Image Segmentation. *Appl. Sci.* **2019**, *9*, 1816. [CrossRef]
21. Liu, W.; Cheng, D.; Yin, P.; Yang, M.; Li, E.; Xie, M.; Zhang, L. Small Manhole Cover Detection in Remote Sensing Imagery with Deep Convolutional Neural Networks. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 49. [CrossRef]
22. Schuegraf, P.; Bittner, K. Automatic Building Footprint Extraction from Multi-Resolution Remote Sensing Images Using a Hybrid FCN. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 191. [CrossRef]
23. Panboonyuen, T.; Jitkajornwanich, K.; Lawawirojwong, S.; Srestasathiern, P.; Vateekul, P. Semantic Segmentation on Remotely Sensed Images Using an Enhanced Global Convolutional Network with Channel Attention and Domain Specific Transfer Learning. *Remote Sens.* **2019**, *11*, 83. [CrossRef]
24. Liu, P.; Liu, X.; Liu, M.; Shi, Q.; Yang, J.; Xu, X.; Zhang, Y. Building Footprint Extraction from High-Resolution Images via Spatial Residual Inception Convolutional Neural Network. *Remote Sens.* **2019**, *11*, 830. [CrossRef]
25. Pan, X.; Yang, F.; Gao, L.; Chen, Z.; Zhang, B.; Fan, H.; Ren, J. Building Extraction from High-Resolution Aerial Imagery Using a Generative Adversarial Network with Spatial and Channel Attention Mechanisms. *Remote Sens.* **2019**, *11*, 917. [CrossRef]
26. Benjdira, B.; Bazi, Y.; Koubaa, A.; Ouni, K. Unsupervised Domain Adaptation Using Generative Adversarial Networks for Semantic Segmentation of Aerial Images. *Remote Sens.* **2019**, *11*, 1369. [CrossRef]
27. Pan, X.; Gao, L.; Zhang, B.; Yang, F.; Liao, W. High-Resolution Aerial Imagery Semantic Labeling with Dense Pyramid Network. *Sensors* **2018**, *18*, 3774. [CrossRef] [PubMed]
28. Yao, X.; Yang, H.; Wu, Y.; Wu, P.; Wang, B.; Zhou, X.; Wang, S. Land Use Classification of the Deep Convolutional Neural Network Method Reducing the Loss of Spatial Features. *Sensors* **2019**, *19*, 2792. [CrossRef] [PubMed]
29. Liu, Y.; Fan, B.; Wang, L.; Bai, J.; Xiang, S.; Pan, C. Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 78–95. [CrossRef]
30. Wu, G.; Guo, Y.; Song, X.; Guo, Z.; Zhang, H.; Shi, X.; Shibasaki, R.; Shao, X. A Stacked Fully Convolutional Networks with Feature Alignment Framework for Multi-Label Land-cover Segmentation. *Remote Sens.* **2019**, *11*, 1051. [CrossRef]
31. Marmanis, D.; Schindler, K.; Wegner, J.D.; Galliani, S.; Datcu, M.; Stilla, U. Classification with an edge: Improving semantic image segmentation with boundary detection. *arXiv* **2016**, arXiv:1612.01337v1
32. Krizhevsky, A.; Hinton, G. Convolutional deep belief networks on cifar-10. *Unpubl. Manuscr.* **2010**, *40*, 1–9.
33. Chollet, F. Xception: Deep learning with depthwise separable convolutions. *arXiv* **2017**, arXiv:1610.02357.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv* **2015**, arXiv:1512.03385.
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. *arXiv* **2017**, arXiv:1709.01507.
36. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 8–13 December 2014; pp. 3320–3328.
37. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* **2015**, *115*, 211–252. [CrossRef]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv* **2015**, arXiv:1502.01852.
39. ISPRS (International Society for Photogrammetry and Remote Sensing). 2D Semantic Labeling Challenge. Available online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (accessed on 10 November 2018).
40. Facebook. Available online: http://pytorch.org (accessed on 9 September 2017).
41. Duda, R.; Hart, P.; Stork, D. *Pattern Classification*; Wiley Press: Hoboken, NJ, USA, 2000.

42. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. ICNet for Real-Time Semantic Segmentation on High-Resolution Images. *arXiv* **2018**, arXiv:1704.08545.
43. Mehta, S.; Rastegari, M.; Caspi, A.; Shapiro, L.; Hajishirzi, H. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. *arXiv* **2018**, arXiv:1803.06815v2.
44. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 325–341.
45. Nekrasov, V.; Shen, C.; Reid, I.; Light-Weight RefineNet for Real-Time Semantic Segmentation. In Proceedings of the 29th British Machine Vision Conference (BMVC), Newcastle, UK, 3–6 September 2018.
46. Li, G.; Milan, A.; Shen, C.; Reid, I. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. *arXiv* **2016**, arXiv:1611.06612.