remote sensing

MDPI

Article

A Stacked Fully Convolutional Networks with Feature Alignment Framework for Multi-Label Land-cover Segmentation

Guangming Wu [1,†], Yimin Guo [1,†], Xiaoya Song [1,2], Zhiling Guo [1], Haoran Zhang [1], Xiaodan Shi [1], Ryosuke Shibasaki [1] and Xiaowei Shao [1,*]

[1] Center for Spatial Information Science, University of Tokyo, Kashiwa 277-8568, Japan; huster-wgm@csis.u-tokyo.ac.jp (G.W.); guo.ym@csis.u-tokyo.ac.jp (Y.G.); song.xy@csis.u-tokyo.ac.jp (X.S.); guozhilingcc@csis.u-tokyo.ac.jp (Z.G.); zhang_ronan@csis.u-tokyo.ac.jp (H.Z.); shixiaodan@csis.u-tokyo.ac.jp (X.S.); shiba@csis.u-tokyo.ac.jp (R.S.)

[2] Key Laboratory of Cold Region Urban and Rural Human Settlement Environment Science and Technology, Ministry of Industry and Information Technology, School of Architecture, Harbin Institute of Technology, Harbin 150001, China

* Correspondence: shaoxw@csis.u-tokyo.ac.jp; Tel.: +81-04-7136-4390

† These authors contributed equally to this work.

Received: 30 March 2019; Accepted: 30 April 2019; Published: 3 May 2019

Abstract: Applying deep-learning methods, especially fully convolutional networks (FCNs), has become a popular option for land-cover classification or segmentation in remote sensing. Compared with traditional solutions, these approaches have shown promising generalization capabilities and precision levels in various datasets of different scales, resolutions, and imaging conditions. To achieve superior performance, a lot of research has focused on constructing more complex or deeper networks. However, using an ensemble of different fully convolutional models to achieve better generalization and to prevent overfitting has long been ignored. In this research, we design four stacked fully convolutional networks (SFCNs), and a feature alignment framework for multi-label land-cover segmentation. The proposed feature alignment framework introduces an alignment loss of features extracted from basic models to balance their similarity and variety. Experiments on a very high resolution(VHR) image dataset with six categories of land-covers indicates that the proposed SFCNs can gain better performance when compared to existing deep learning methods. In the 2nd variant of SFCN, the optimal feature alignment gains increments of 4.2% (0.772 vs. 0.741), 6.8% (0.629 vs. 0.589), and 5.5% (0.727 vs. 0.689) for its f1-score, jaccard index, and kappa coefficient, respectively.

Keywords: land-cover classification; image segmentation; ensemble learing; feature alignment; fully convolutional networks

1. Introduction

The distributions and changes of natural and artificial surfaces, such as grasslands, forests, buildings and roads, is fundamental information that is referenced for many applications such as urban planning [1], navigation [2], land-used management [3], and forest monitoring [4]. Traditionally, this information was obtained by labor-intensive and time-consuming field surveys [5]. The ability to achieve precise and cost-efficient updating of land cover is a long-existing demand for remote sensing. Over the last few years, with the emerging of innovative technologies, the cost as well as difficulty of capturing very high resolution(VHR) aerial imagery has significantly declined [6,7]. Thus, robust and

precise methods for the automatic classification and segmentation of land cover become the core of the whole solution.

According to the conditions of image datasets such as scale, color space, and resolution, various automatic segmentation methods have been proposed. These methods can be divided into two categories depending on whether it is necessary to have ground truth: I. unsupervised methods and II. supervised methods. Unsupervised methods can be further categorized as three groups according to their operational mechanisms: (1) threshold-based, (2) edge-based, and (3) region-based methods. Threshold-based methods separate different parts using thresholds determined by the value or histogram of the pixels [8]. Edge-based methods detect the abrupt changes using mathematical designed filters, such as Sobel [9] and Canny [10], to generate boundaries between different parts. In region-based methods, image segmentations are done by clustering or region-growing [11–13]. Because of manually adjustable parameters and the lack of need for ground truth, unsupervised methods are easier to implement, and are widely adopted for small scale datasets. However, for larger datasets, as the variety and complexity increase, the performance of unsupervised segmentation methods usually lacks generalization capability [14]. In direct contrast, supervised methods utilize the ground truth to learn segmentation patterns and then apply it to new data. For supervised methods, the segmentation problem is converted into a pixel-to-pixel image classification where pixels of different parts are classified into their corresponding categories [15]. Because the segmentation is made by classifying each pixel, these methods generally produce segmentations that are more precise.

For supervised segmentation methods, there are two fundamental procedures: feature extraction, and classification. At the early stages, these steps are done separately. The spatial or textual features are firstly extracted from the image through hand-crafted descriptors, such as haar-like, local binary pattern, and histogram of oriented gradient [16–19]. Later, various classifiers, such as support vector machines, decision trees, and neural networks are utilized for further classification using the extracted features [20–23]. Because of the separateness of the two procedures, optimizing the performance of methods requires many cycles of trial and error. Instead, convolutional neural network (CNN) methods incorporate automatic feature extraction and classification through an unified framework [24]. As these steps can directly learn from the ground truth, CNNs show superior generalization capabilities and precision in many classification and segmentation tasks [25].

Before fully convolutional networks (FCN) [26], CNN-based methods adopted patch-based approaches which classified the center pixel by using a small patch of the whole image [27]. Because of highly overlapped patches, these methods required massive memory space as well as high computational capability. To solve this problem, the FCN method utilizes fully convolutional network architectures that can directly perform pixel-to-pixel translation of the input images to ground truth. In this manner, the FCN method significantly improves training efficiency and model performance [28]. In classic FCNs (FCN32s, FCN16s and FCN-8s), the methods adopt multiple scale bilinear upsampling operations to generate segmentation output with the same height and width of input. These operations lead to information loss that affects the precision of prediction. Recently, more advanced and accurate FCN-based methods have been developed [29]. These methods improve model performance through different strategies. The U-Net and FPN methods adopt multiple skip-connections between corresponding lower and upper layers to share information between layers [30,31]. The DeconvNet replaces bilinear upsampling with deconvolution (convolution transpose) operation [32]. The MC-FCN method applies multi-constraints for various scale outputs [33]. Finally, the BR-Net method uses additional boundary information to regulate the model [34]. These methods further develop the potential of fully convolutional networks. However, with more complex architectures and stronger representation capabilities, overfitting becomes inevitable [35].

Overfitting is a long-existing problem in deep learning. This problem is more critical for smaller datasets. To compensate for the problem, several approaches are proposed. These approaches include early stopping, data augmentation, regularization, and ensemble learning. The early stopping approach stops the training model before convergence to prevent overfitting [36,37]. For the data augmentation

approach, the original images are rotated, resized, random cropped, or re-colorized to generate more training samples and increase the variety of data [38]. As for regularization, extra penalty (e.g., L1/L2) [39,40] or dropout [41] is implemented to reduce and regulate the representation capability of the model. By contrast, ensemble learning combines several models to generate a final prediction [42]. Owing to its capability of utilizing a variety of different models, biased predictions from one model can be compensated for by other models, and better results can be produced. Currently, ensemble learning is mainly applied to patch-based CNN for pixel-level classification [43]. Ensemble learning has not received any attention in FCN-based architectures. In addition, research on ensemble learning is mainly focused on various numbers or combinations of basic models. The studies on the combination approaches of different basic models are not sufficient.

To explore the capability of ensemble learning using fully convolutional networks, we design four stacked fully convolutional networks(SFCNs) using FCN-8s, U-Net, and FPN. Furthermore, we propose a feature alignment framework for efficient ensemble learning, which enhances the relations between basic models. Compared with traditional ensemble learning approaches, the proposed method implements basic segmentation loss between prediction and corresponding ground truth as well as extra alignment loss between features that are extracted separately from different basic models. The value of the alignment loss is determined by the consistency of features extracted by different models. If these features are similar, the alignment loss is zero. During iterations, the optimizer is required to update parameters to reduce the value of the weighted sum of segmentation loss and alignment loss. Thus, the optimized network is capable of generating predictions using features extracted from basic models that contain a balance of similarity and variety.

The effectiveness of the proposed feature alignment framework is demonstrated by a VHR image dataset with 2D multi-label segmantic information(refer to Section 2.1). In comparative experiments, the performances of achieved by the proposed method (SFCNv3, +FL) are 0.785($\pm$0.004) of F1-score, 0.646($\pm$0.005) of jaccard index [44], and 0.742($\pm$0.005) of kappa coefficient [45], respectively. Furthermore, sensitivity analysis indicates that the proposed feature alignment can control the balance between similarity and variety of features extracted from different basic models. By optimizing the feature alignment level, ensemble fully convolutional networks gain better model performance.

The main contributions of this study can be summarized as follows:(1) We design a stacked fully convolutinal networks architecture using multiple FCNs for efficient multi-label land-cover segmentation and (2) we further proposed a feature alignment framework to balance the similarity and variety of features extracted from basic models to gain extra performance.

The rest of the paper is organized as follows: First, the materials and methods used for this research are described in the Section 2. Then, the quantitative and qualitative comparison results of different methods are presented in the Section 3. Finally, the discussions and conclusions from this study are presented in the Sections 4 and 5, respectively.

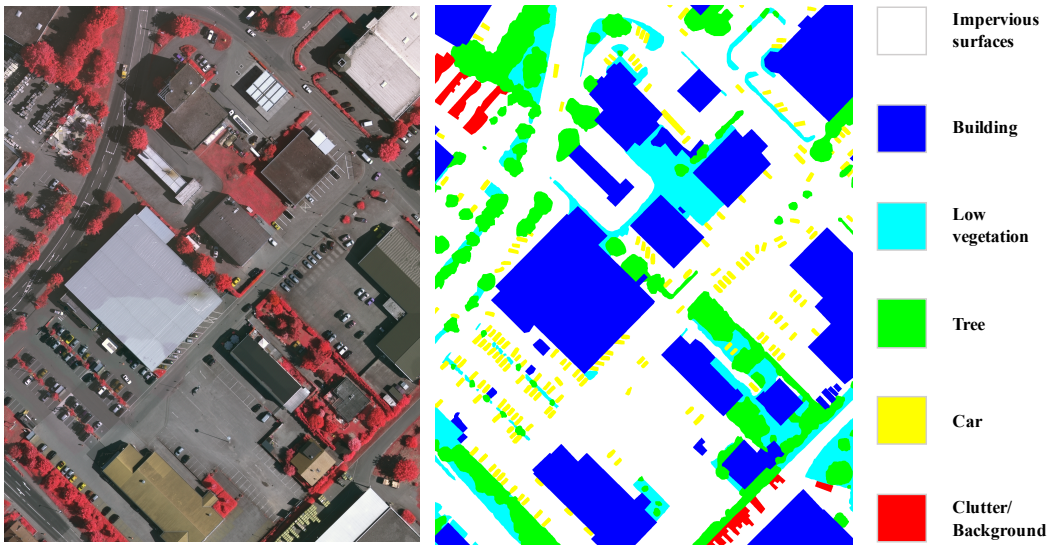
2. Materials and Methods

2.1. Dataset

For estimating the effectiveness of the designed SFCNs and proposed feature alignment framework, we conduct our experiments on ISPRS Vaihingen (Germany) 2D semantic labeling dataset. The dataset is an open benchmark, which is available online (http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html). Within the dataset, there are 33 tiles including 16 tiles for training and 17 tiles for testing. Only the tiles used for training are provided with images of annotated ground truth. The size of each tile ranges from 1388 $\times$ 2555 to 2006 $\times$ 3007 pixels. The ground sampling distance (GSD) of orthophoto is about 9 cm.

As shown in Figure 1, each tile of the dataset contains a orthophoto and its corresponding annotated ground truth. The orthophoto is an 8-bit image with three bands, which correspond to the near-infrared, red, and green bands delivered by the camera. The image of annotated ground truth

utilizes six different colors to represent land-covers of impervious surfaces, buildings, low vegetation, trees, cars, and clutter/background (see color map in Table 1).



(a) Orthophoto (b) Ground truth (c) Legend

Figure 1. Example of Vaihingen 2D semantic labeling dataset. (**a**) true orthophoto, (**b**) annotated ground truth, and (**c**) legend. The ground truth contains six types of land-covers.

Table 1. Reference of color map of Vaihingen dataset.

Land-covers	RGB Values
Impervious surfaces	[255, 255, 255]
Building	[0 , 0, 255]
Low vegetation	[0 , 255, 255]
Tree	[0 , 255, 0]
Car	[255, 255, 0]
Clutter/Background	[255, 0, 0]

2.2. Method

Figure 2 presents the workflow of this research. All 16 tiles of orthophoto, as well as their corresponding ground truth, are divided into two sets for training and testing. These sets contain 12 and 4 tiles of images, respectively. A sliding window with stride of 224 pixel is applied to each tile of the training set to generate image pathes with size of 224 $\times$ 224 pixels. After data preprocessing, the image patches are shuffled and split into two groups that include training (70%), and cross-validating (30%). The number of samples in training and cross-and validation are 744 and 312, respectively. Through several cycles of training and cross validation, the hyper-parameters are determined and optimized. Then, the predictions generated by the optimized model are further evaluated by the tiles in the test set. For performance evaluations, we choose three commonly used evaluation metrics, namely, jaccard index, f1-score, and kappa coefficient. These metrics are computed without post-processing operations [46,47] for better estimation of experimental methods.

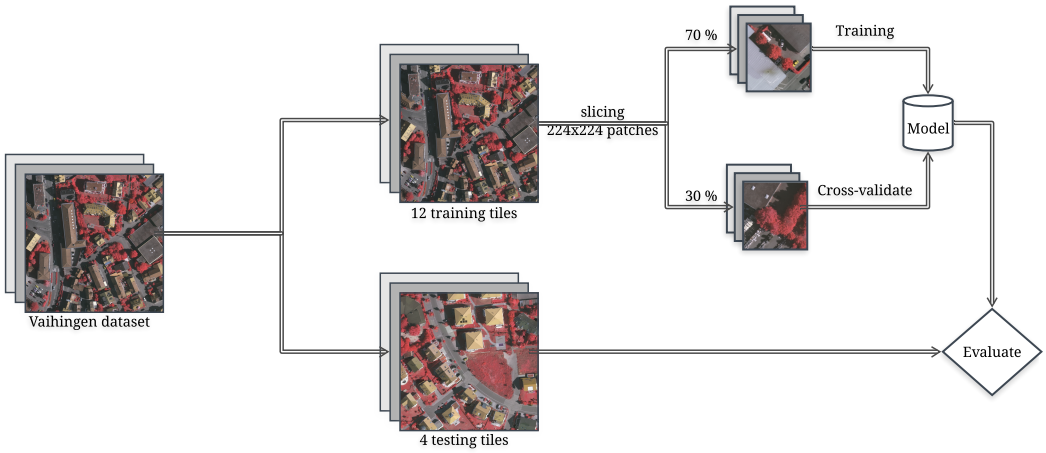


Figure 2. Experimental workflow of this research. The existing methods, as well as the proposed model, are trained and evaluated by 224 × 224 image patches extracted from original dataset.

2.2.1. Stacked Fully Convolutional Networks

After the invention of FCN in 2015, FCN and FCN-based methods have become a gold standard for many image segmentation tasks [48,49]. Compared to conventional patch-based CNN methods, FCN-based models significantly improve computational efficiency and performance. Advanced FCN-based models further enhance feature representation capabilities and improve model performance through various approaches. These approaches include various combinations of skip-connections (U-Net & FPN), replacing bilinear upsample with unpooling (SegNet) or convolution transpose (DenconvNet), multi-constraints (MC-FCN), and additional boundary information (BR-Net). However, the increased representation capability and the complexity of the models usually lead to the overfitting of training data, especially for small or biased datasets.

To avoid overfitting, approaches including early stopping, data augmentation, regularization, and ensemble learning, are widely adopted. Of them, owing to its ability to utilize the representation capability of different models, the ensemble learning approach shows better performance and generalization capability. However, ensemble learning is currently used for patch-based CNN architectures, but not for FCN-based architectures. Additionally, research on ensemble learning mainly focuses on adding numbers or trying different combinations of basic models. To our best knowledge, research on methods to discern better combinations of various FCNs in ensemble learning does not exist.

Thus, we design stacked fully convolutional networks (SFCNs), and propose a feature alignment method, which enhances the relations between basic models. For ensemble learning, if the predictions from two models are completely different (in extreme cases, one of them is all zeros and the other is all ones), the ensemble result is just an average of both biased predictions that cannot yield better performances. Therefore, to have better results, the predictions of different models should contain a certain level of variety as well as similarity. Compared with traditional ensemble learning approaches, the proposed method introduces an extra alignment loss to control similarity as well as consistency between features that are extracted separately from different basic models. In contrast to common segmentation loss, which is computed as the difference between a ground truth and its corresponding prediction, proposed alignment loss is computed among extracted features from stacked basic models. To make sure the alignment loss can be applied to a various number of basic models (e.g., 2 models of FCN and U-Net, 3 models of FCN, U-Net, and FPN), the alignment loss is computed as the mean square error(MSE) between the maximum and minimum values of the extracted features (see details in Equation (1)). The value of alignment loss becomes zero when all the extracted features are similar. The value of the alignment loss reflects the consistency of extracted features. During iterations,

the optimizer is required to update parameters to reduce the value of the weighted sum of segmentation loss and alignment loss. Thus, the optimized network is capable of generating a normalized prediction from variant basic models. Through feature alignment framework, the SFCNs can achieve a balance of similarity and variety using different basic models, and improve performance.

Figure 3 presents the design of the proposed stacked fully convolutinal networks(SFCN). The SFCN consists of two parts: (1) a framework for feature extraction using various fully convolutional networks and (2) a framework for feature alignment and output generation.

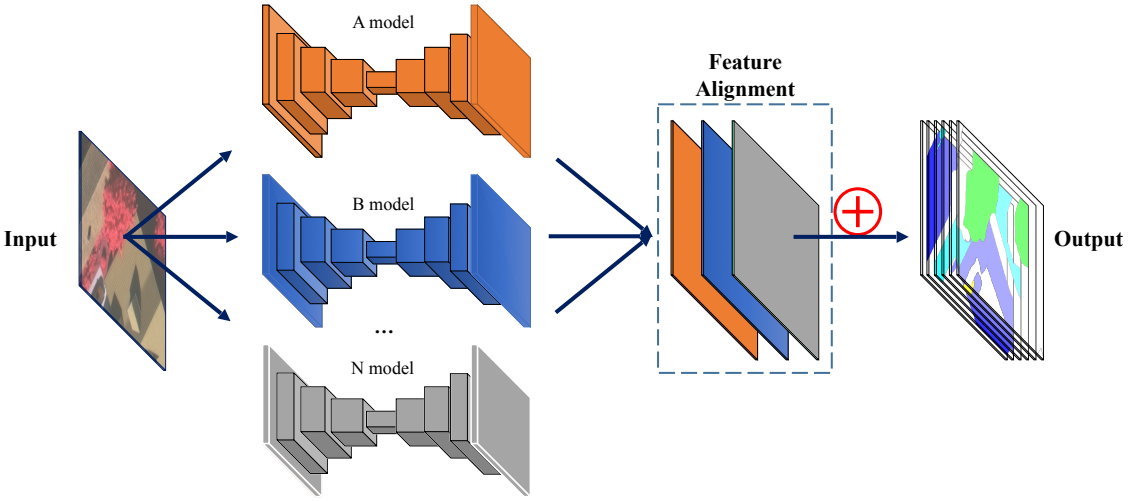


Figure 3. Proposed stacked fully convolutional networks(SFCN). The SFCN contains a framework for feature extraction using N number of fully convolutional networks(A, B, ..., N model), and a framework for feature alignment and final output generation.

In the feature extraction framework, different numbers or combinations of FCN-based models are implemented to separately extract features from the same input image. For each FCN-based model, there are several universal operations and model specific layers. For universal operations, there are convolution, nonlinear activation, and subsampling operations. For backend models, various model specific layers, such as skip-connection (U-Net & FPN) and unpooling (SegNet), are included.

For universal operations, element-wise multiplication within kernel is computed through the convolutional operation. The size of the kernel determines the receptive field and the computational efficiency of the convolution operation. Later, the output of convolution is handled by the rectified linear unit (ReLU) [50], which returns the original value if the value is larger than zero and sets values less than zero to zero. To accelerate network training, most models adopt batch normalization (BN) [51] layers before (e.g., SegNet) or after non-linear activations (e.g., FPN). To reduce the width and height of features, max-pooling [52] is chosen for subsampling in this study.

As for model specific layers, sequential bilinear upsampling [53] is commonly used to upsample the width and height of the features. By contrast, SegNet backend uses unpooling which applies corresponding pooling indices of max-pooling to achieve upsampling. In FPN and U-Net backends, skip-connection, which concatenates two layers with consistent height and width across channel axis, is applied between downward and upward layers.

In the framework for feature alignment and output generation, alignment loss that restricts the consistency of extracted features from various models and multi-class segmentation loss are computed sequentially.

- Alignment loss ($Loss_{align}$)

  Through the n^{th} FCN-based model, extracted features (denoted as Xn) with size of $W \times H \times D$ are generated. W and H is consistent with the height and width of the input. The value of D is the

same as the number of classes of land covers. The maximum and minimum value for each position from the 1^{st} to the n^{th} feature are computed. The final alignment loss ($Loss_{align}$) is calculated by the mean square error between corresponding maximum and minimum values of all positions.

$$\begin{aligned} Xmax_{i,j,k} &= max(X1_{i,j,k}, X2_{i,j,k}, ..., Xn_{i,j,k}) \\ Xmin_{i,j,k} &= min(X1_{i,j,k}, X2_{i,j,k}, ..., Xn_{i,j,k}) \\ Loss_{align} &= \frac{1}{W \times H \times D} \sum_{i=1,j=1,k=1}^{W,H,D} (Xmax_{i,j,k} - Xmin_{i,j,k})^2 \end{aligned} \tag{1}$$

- Segmentation loss ($Loss_{seg}$)

  From all extracted features ($X1, X2, ..., Xn$), the final output/prediction(Y) of the network is computed by taking the average value of all features. Then, the binary cross entropy [54], which calculates the difference between ground truth(G) and its corresponding prediction, is used as segmentation loss($Loss_{seg}$). The calculation can be formulated as

$$\begin{aligned} Y &= \frac{1}{N} \sum (X1, X2, ..., Xn) \\ Loss_{seg} &= -\frac{1}{W \times H \times D} \sum_{i=1,j=1,k=1}^{W,H,D} g_{i,j,k} \times \log(y_{i,j,k}) + (1 - g_{i,j,k}) \times \log(1 - y_{i,j,k}) \end{aligned} \tag{2}$$

  where $y_{i,j,k}$ and $g_{i,j,k}$ represent the (i,j,k) element of model output(Y) and ground truth (G). The value of $y_{i,j,k}$ is the predicted probability of the pixel category.

  Therefore, the total loss of the network can be formulated as

$$Loss_{final} = Loss_{seg} + \lambda \times Loss_{align} \tag{3}$$

where λ is the weight of the alignment loss ($Loss_{align}$). By controlling the value of λ, we are able to adjust the balance between $Loss_{align}$ and $Loss_{seg}$.

During iterations, Adam optimizer [55] will minimize $Loss_{final}$ to driven proposed network to generate pixel-to-pixel predictions for multi-label land-cover segmentation.

2.3. Experimental Set-Up

Three classic FCN-based architectures, including FCN-8s, U-Net, and FPN, are chosen as the basic models. All these models are implemented by Geoseg [56] using PyTorch (https://pytorch.org/, version = 0.3.0) as backend.

2.3.1. Network Specification

- FCN-8s. The classic FCN-8s architecture was proposed by Long et al. in 2015 [26]. This method innovatively adopts fully convolutional architecture to perform pixel-to-pixel image classification or segmentation. The FCN architecture is the first fully convolutional network used for image segmentation.
- U-Net. The U-Net architecture was proposed by Ronneberger et al. [30] for medical image segmentation. This method introduces multiple skip connections between upper and downer layers. Owing to its robustness and elegant structure, U-Net and its variants are widely adopted for many semantic segmentation tasks.
- FPN. The FPN architecture was published by Lin et al., 2017 [31]. Like U-Net, this method adopts multiple skip connections. In addition, the FPN model generates multi-scale predictions for final

output. By utilizing abundant information from the feature pyramid, the FPN method achieves state-of-the-art performance.

The number and the size of convolutional kernels have significant impact on model performance. To minimize their effect, basic models used in this research are implemented with consistent number of kernel size at corresponding layers(see details in Figure 4).

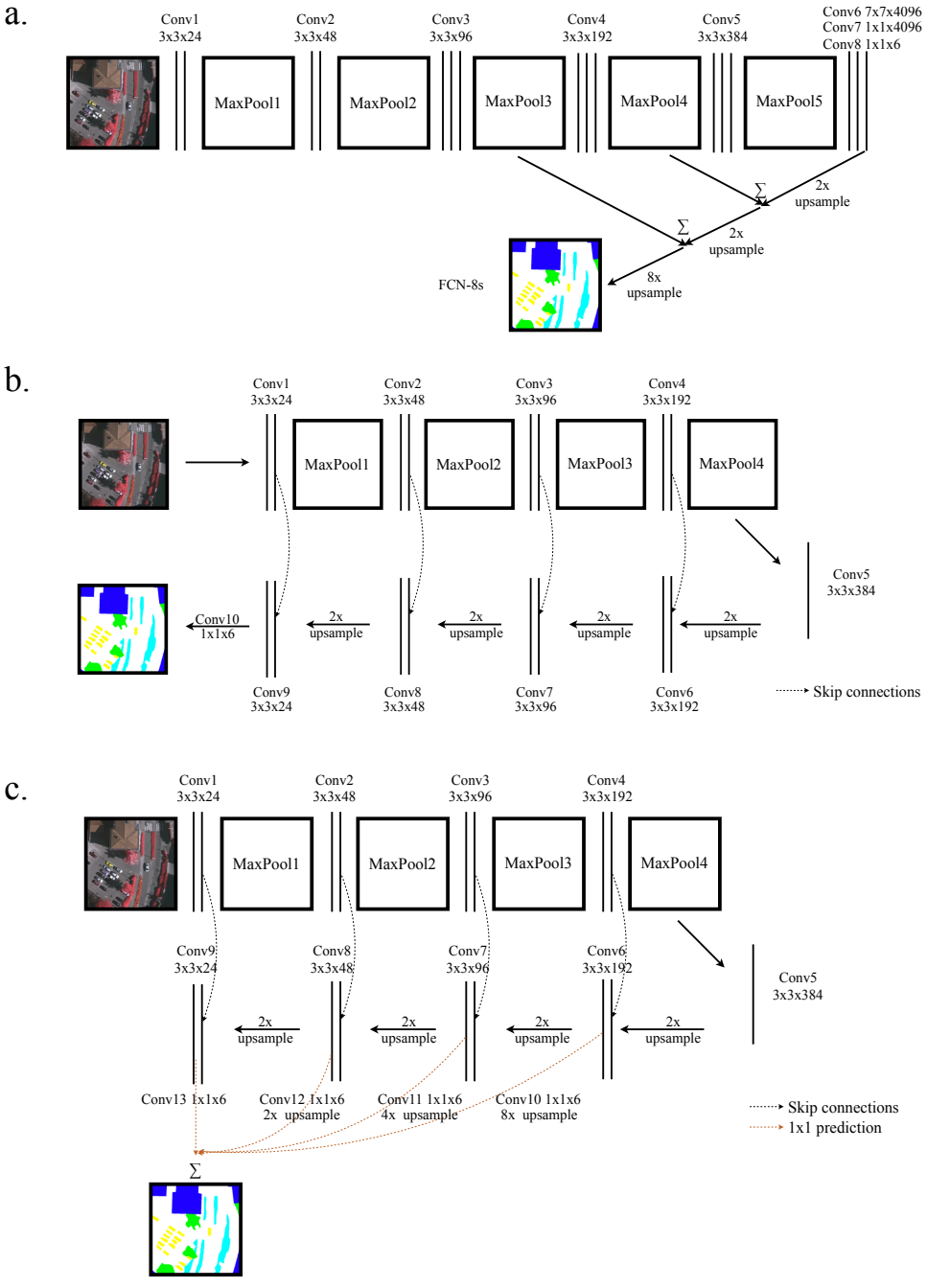


Figure 4. Specification of three basic models: (**a**) FCN-8s, (**b**) U-Net, and (**c**) FPN.

2.3.2. Model Setup

To analyze the importance or significance of the proposed alignment loss, four versions of stacked fully conventional networks (SFCNs) are setup. There are three variants utilizing different

combinations of two basic models, and a variant utilizing all three basic models. The variants using two basic models, $SFCN_{f\&p}$, $SFCN_{f\&u}$ and $SFCN_{u\&p}$ consist of FCN-8s&FPN, FCN-8s&U-Net, and U-Net&FPN, respectively (as shown in Table 2). All combinations are separately trained with different values of λ ($\lambda \in [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$). In all experiments, the models are trained, cross-validated, and tested though exactly the same dataset. To prevent random bias, each set of experiments is repeated five times. After removing the best and worst performances of each method, their average performance with the testing dataset is carefully evaluated.

Table 2. Network setting of stacked fully convolutional networks from FCN-8s, U-Net, and FPN.

Version	No. of basic models	FCN-8s	U-Net	FPN
$SFCN_{f\&p}$	2	+	−	+
$SFCN_{f\&u}$	2	+	+	−
$SFCN_{u\&p}$	2	+	+	−
$SFCN_{f\&u\&p}$	3	+	+	+

3. Results

Three well-known FCN-based methods, namely, FCN-8s, U–Net, and FPN, are chosen for basic models in this study. Four SFCN models composed from three basic models (refer to Table 2) are trained separately with various weight (λ) of alignment loss ($Loss_{align}$). All experiments are performed on the same dataset and processing platform.

Three commonly used balanced metrics, including f1-score, jaccard index, and kappa coefficient, are selected for quantitative evaluation. Figures 5 and 6 show the comparison results of experimental methods.

3.1. Sensitivity Analysis of Feature Alignment

To investigate the significance of feature alignment, four stacked fully convolutional networks (i.e., SFCNs) using sequential values of lambda ($\lambda \in [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$) are implemented and validated on the testing dataset. To prevent random bias, each set of experiments is repeated five times. After removing the best and worst performances of each method, their mean value and standard deviation (SD) of the evaluation metrics are calculated. Figure 5 and Table 3 present the trends and values of f1-score, jaccard index, and kappa coefficient over various λ of $Loss_{align}$.

Figure 5a shows the trend of performances over λ values of $Loss_{align}$ on $SFCN_{f\&p}$. As the value of λ increases, the values of three metrics improve. The best performance is achieved with maximum value of λ = 1.0. This result indicates that the introduction of feature alignment leads to better performance of the ensemble model. Figure 5b,c show the trend of performances on $SFCN_{f\&u}$ and $SFCN_{u\&p}$, respectively. When $\lambda \leq 0.8$, higher λ generally has higher value metrics. By contrast, while $\lambda \geq 0.8$, higher λ leads to weaker performances. In contract to Figure 5a–c, there is no significant change in the values of the metrics under various λ in Figure 5d, which implies that feature alignment has no significant effect on $SFCN_{f\&u\&p}$.

Table 3 reveals the values of evaluations metrics f1-score, jaccard index, and kappa coefficient of four ensemble methods using λ in [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]. For $SFCN_{f\&p}$, the best performances are achieved at λ values 0.8 and 1.0. When compared with no feature alignment(i.e.,λ = 0.0), the values of f1-score, jaccard index, and kappa coefficient increase about 3.9% (0.767 vs. 0.738), 6.7% (0.623 vs. 0.584), and 5.3% (0.721 vs. 0.685), respectively. For $SFCN_{f\&u}$, the best performance is achieved at λ value 0.8. When compared to no feature alignment, the highest values of f1-score, jaccard index, and kappa coefficient increase about 4.2% (0.772 vs. 0.741), 6.8% (0.629 vs. 0.589), and 5.5% (0.727 vs. 0.689), respectively. Like $SFCN_{f\&u}$, the best performance of $SFCN_{u\&p}$ is at λ value 0.8. With comparison to the baseline λ = 0.0, the maximum increments of f1-score, jaccard index, and kappa coefficient reach 2.6% (0.785 vs. 0.765), 4.4% (0.646 vs. 0.619), and 3.3% (0.742 vs. 0.718), respectively. By contrast to

the above methods, the values of f1-score for SFCN$_{f\&u\&p}$ are almost identical (within [0.773, 0.780]). Under optimal feature alignment condition (e.g., λ = 1.0), the values of jaccard index and kappa coefficient increase about 1.6% (0.640 vs. 0.630) and 1.2% (0.736 vs. 0.727), respectively. When compared to other methods (e.g., SFCN$_{f\&u}$), the improvement caused by feature alignment of SFCN$_{f\&u\&p}$ is not so significant. The values for the standard deviation (SD) of the three metrics from different models range from 0.001 to 0.006. When compared to the mean values, even the maximum value of SD (0.006) is not significant.

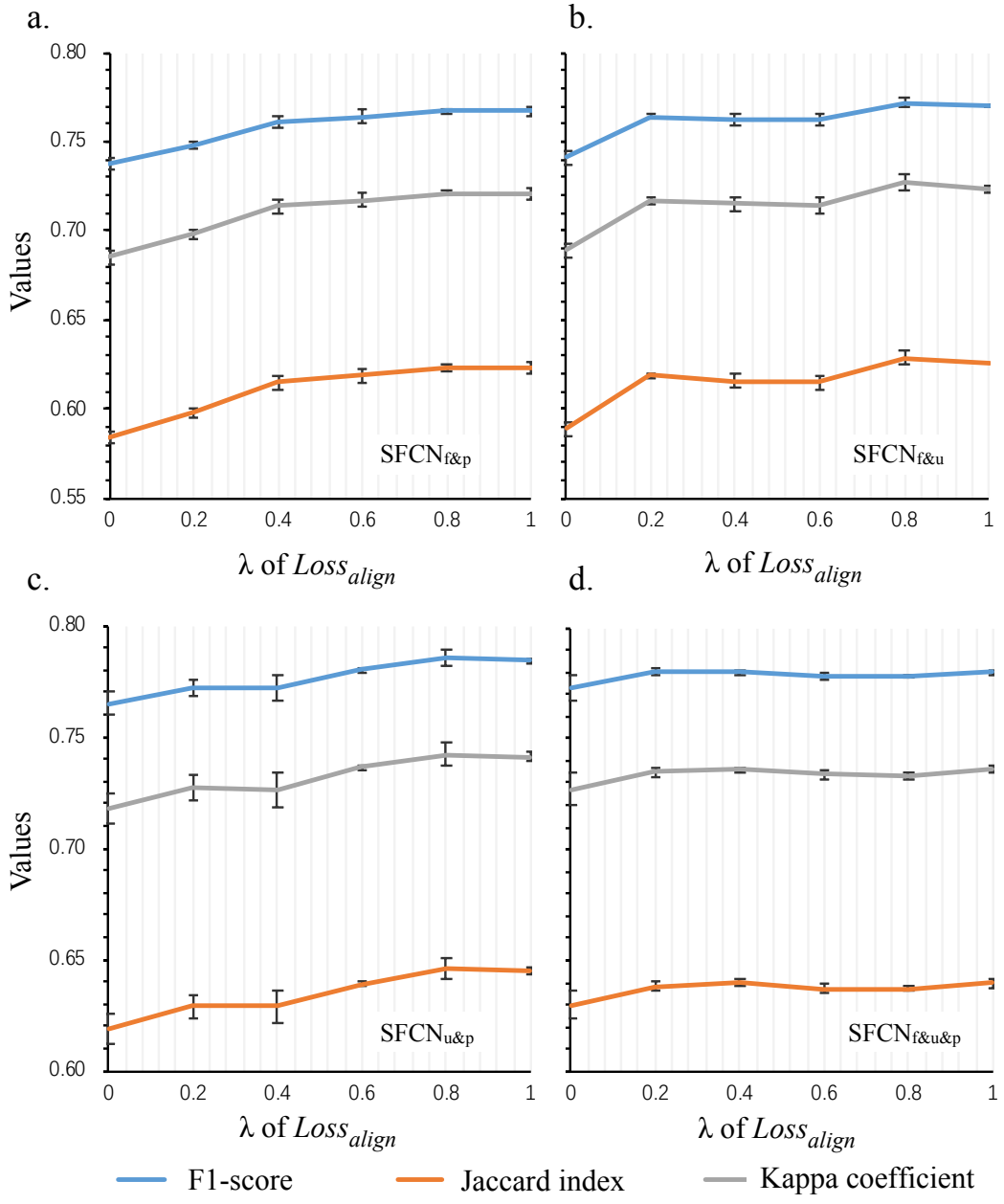


Figure 5. Trends of model performances of four SFCNs using lambda values in [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]: (**a**) performances of SFCN$_{f\&p}$ over lambda values; (**b**) performances of SFCN$_{f\&u}$ over lambda values; (**c**) performances of SFCN$_{u\&p}$ over lambda values; and (**d**) performances of SFCN$_{f\&u\&p}$ over lambda values.

Table 3. Table of model performances of four SFCNs under lambda values in [0.0, 0.2, 0.4, 0.6, 0.8, 1.0].The x and y axes represent the models and their corresponding values, respectively. (**b**) Table of mean value and standard deviation (SD) of performance comparisons among these methods. For each evaluation metric, the highest mean values and lowest SD are highlighted in **bold**.

Method		**F1-Score**		**Jaccard Index**		**Kappa Coefficient**	
Version	**λ-Value**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
$SFCN_{f\&p}$	0.0	0.738	0.003	0.584	0.004	0.685	0.004
	0.2	0.748	**0.002**	0.598	0.003	0.698	0.003
	0.4	0.761	0.004	0.615	0.004	0.714	0.004
	0.6	0.764	0.004	0.619	0.004	0.717	0.004
	0.8	**0.767**	**0.002**	**0.623**	**0.002**	**0.721**	**0.002**
	1.0	**0.767**	0.003	**0.623**	0.003	**0.721**	0.003
$SFCN_{f\&u}$	0.0	0.741	0.003	0.589	0.004	0.689	0.004
	0.2	0.764	0.002	0.619	**0.002**	0.717	**0.002**
	0.4	0.762	0.003	0.616	0.004	0.715	0.004
	0.6	0.762	0.004	0.615	0.005	0.714	0.004
	0.8	**0.772**	0.003	**0.629**	0.004	**0.727**	0.004
	1.0	0.770	**0.001**	0.626	**0.002**	0.723	**0.002**
$SFCN_{u\&p}$	0.0	0.765	0.006	0.619	0.007	0.718	0.006
	0.2	0.772	0.004	0.629	0.006	0.727	0.005
	0.4	0.772	0.006	0.629	0.008	0.726	0.007
	0.6	0.780	**0.001**	0.639	**0.002**	0.736	**0.001**
	0.8	**0.785**	0.004	**0.646**	0.005	**0.742**	0.005
	1.0	0.784	**0.001**	0.645	**0.002**	0.741	0.002
$SFCN_{f\&u\&p}$	0.0	0.773	0.006	0.630	0.007	0.727	0.006
	0.2	**0.780**	0.002	0.638	0.002	0.735	0.002
	0.4	**0.780**	**0.001**	**0.640**	**0.001**	**0.736**	**0.001**
	0.6	0.778	0.002	0.637	0.002	0.734	0.002
	0.8	0.778	**0.001**	0.637	0.002	0.733	**0.001**
	1.0	**0.780**	**0.001**	**0.640**	0.002	**0.736**	0.002

3.2. Performances Comparison

Three basic models (FCN-8s, U-Net, and FPN) and four combinations of ensemble models (i.e., SFCNs) with/without optimal feature alignment (FA) are implemented and validated by the testing dataset. To prevent random bias, each set of experiments is repeated five times. After removing the best and worst performances of each method, their mean value and standard deviation (SD) of evaluation metrics are calculated.

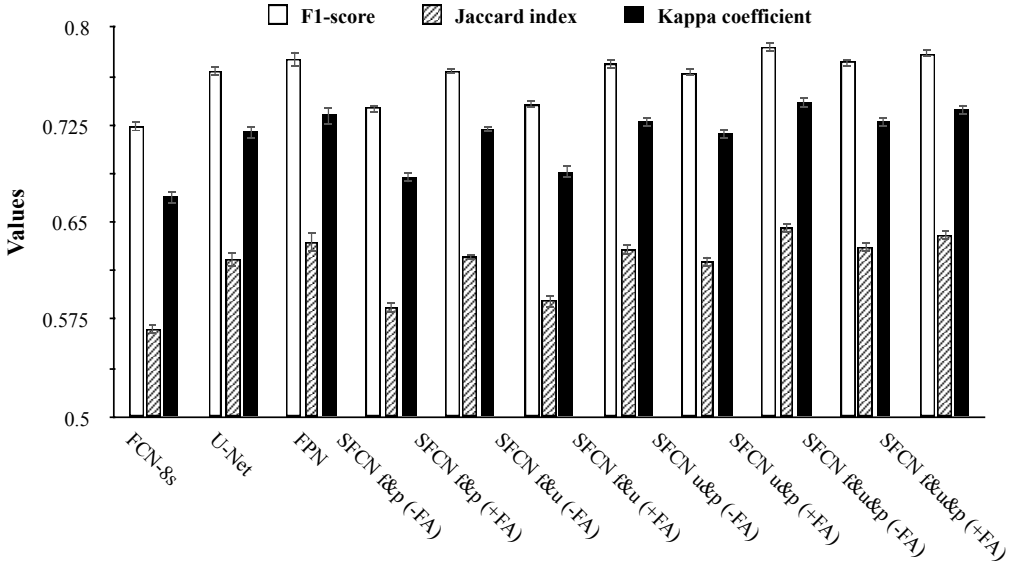
Figure 6a shows the relative performances of these models. Among three basic methods, the FPN shows the highest values for all evaluation metrics. For each combination of ensemble learning, methods with optimal feature alignment(+FA) are generally better than the corresponding methods without optimal feature alignment(−FA).

Figure 6b displays the corresponding mean and standard deviation(SD) values of evaluation metrics from different methods. Among four ensemble models without optimal feature alignment (SFCNs, −FA), $SFCN_{f\&u\&p}$(−FA) shows the higher mean values than $SFCN_{u\&p}$(−FA), $SFCN_{f\&u}$(−FA), and $SFCN_{f\&p}$(−FA) for all metrics. This observation indicates that an ensemble with more models can lead to better performance. For ensemble models using the same number of basic models ($SFCN_{f\&p}$, $SFCN_{f\&u}$, and $SFCN_{u\&p}$), a combination of U-Net and FPN ($SFCN_{u\&p}$) is better than a combination of FCN-8s and U-Net($SFCN_{f\&u}$) or FCN-8s and FPN ($SFCN_{f\&p}$). Surprisingly, the best basic model (FPN) is better than the best ensemble model without feature alignment ($SFCN_{f\&u\&p}$, −FA). This result suggests that a simple ensemble of different basic models does not assure higher performance. As for the four ensemble models with optimal feature alignment (SFCNs, +FA), $SFCN_{u\&p}$(+FA) shows the highest mean values for f1-score (0.785), jaccard index(0.646), and kappa coefficient(0.742). Ensemble

methods with feature alignment showed higher values for all three evaluation metrics Compared than their counterparts without feature alignment. Among all methods, the SFCN$_{u\&p}$(+FA) methods achieved the highest performance.

The values for the standard deviation (SD) of three metrics from different models range from 0.001 to 0.008. When compared to the mean values, even the maximum value of SD (0.008) is not significant. Through independent *t*-test, except for SFCN$_{f\&u\&p}$, methods with optimal feature alignment showed significantly different values for all three evaluation metrics Compared than their counterparts without feature alignment(see details in Table 4).

a.



b.

Method	F1-score		Jaccard index		Kappa coefficient	
	Mean	SD	Mean	SD	Mean	SD
FCN-8s	0.724	0.003	0.568	0.004	0.669	0.004
U-Net	0.766	0.004	0.621	0.006	0.719	0.005
FPN	0.776	0.006	0.635	0.008	0.732	0.007
SFCN$_{f\&p}$ (−FA)	0.738	0.003	0.584	0.004	0.685	0.004
SFCN$_{f\&u}$ (−FA)	0.741	0.003	0.589	0.004	0.689	0.004
SFCN$_{u\&p}$ (−FA)	0.765	0.006	0.619	0.007	0.718	0.006
SFCN$_{f\&u\&p}$ (−FA)	0.773	0.006	0.630	0.007	0.727	0.006
SFCN$_{f\&p}$ (+FA)	0.767	0.002	0.623	0.002	0.721	0.002
SFCN$_{f\&u}$ (+FA)	0.772	0.003	0.629	0.004	0.727	0.004
SFCN$_{u\&p}$ (+FA)	**0.785**	0.004	**0.646**	0.005	**0.742**	0.005
SFCN$_{f\&u\&p}$ (+FA)	0.780	**0.001**	0.640	**0.001**	0.736	**0.001**

Figure 6. Comparison of performances of the basic models of FCN-8s, U-Net, and FPN as well as four SFCNs with/without feature alignment. (**a**) Bar chart for comparison of relative performances. (**b**) Table of mean value and standard deviation (SD) of the performance comparison of these methods. For each evaluation metric, the highest mean values and lowest SD are highlighted in **bold**.

Table 4. Result of independent *t*-test of four SFCNs under with/without feature alignment. The *p*-value is the probability that SFCN has the same performances at both with and without feature alignment.

Group	F1-Score		Jaccard Index		Kappa Coefficient	
	t-Value	*p*-Value	t-Value	*p*-Value	t-Value	*p*-Value
$SFCN_{f\&p}$ (+FA vs.−FA)	14.438	0.0001	16.546	0.0001	15.123	0.0001
$SFCN_{f\&u}$ (+FA vs.−FA)	11.750	0.0003	11.558	0.0003	11.144	0.0004
$SFCN_{u\&p}$ (+FA vs.−FA)	5.350	0.0059	5.329	0.0060	5.210	0.0065
$SFCN_{f\&u\&p}$ (+FA vs.−FA)	2.234	0.0892	2.271	0.0856	2.415	0.0732

3.3. Qualitative Comparison

Figure 7 shows the prediction results on testing areas Tile-1, Tile-2, Tile-3, and Tile-4 of three basic models(FCN-8s, U-Net, and FPN) and optimized SFCNs. Generally, these models could correctly segment the major parts of different land-covers from the original aerial images. The FCN-8s model tends to misclassify low vegetation as trees (e.g., red rectangle in column 2, Tile 1), and the border area of buildings is usually broken (e.g., the red rectangle in column 2, Tile 3). The result generated by U-Net is unable to discriminate between roads and buildings (e.g., red rectangle in column 3, Tile 1 or row 3, Tile 2). The FPN model is generally better than FCN-8S and U-Net. However, trees and roads are misclassified as buildings (e.g., the red rectangle in column 4, Tile 3). Among SFCN models, results generated from $SFCN_{f\&p}$ and $SFCN_{f\&u}$ tend to miss the buildings in the corner area (e.g., the red rectangles in column 5, Tile 4 and column 5, Tile 4). The $SFCN_{f\&u\&p}$ model outperforms $SFCN_{f\&p}$ and $SFCN_{f\&u}$. However, there are misclassified holes within large buildings (e.g., the red rectangle in column 8, Tile 3). When compared to other methods, even with some misclassification (e.g., the red rectangle in column 7, Tile 3), $SFCN_{u\&p}$ shows better performance in major areas.

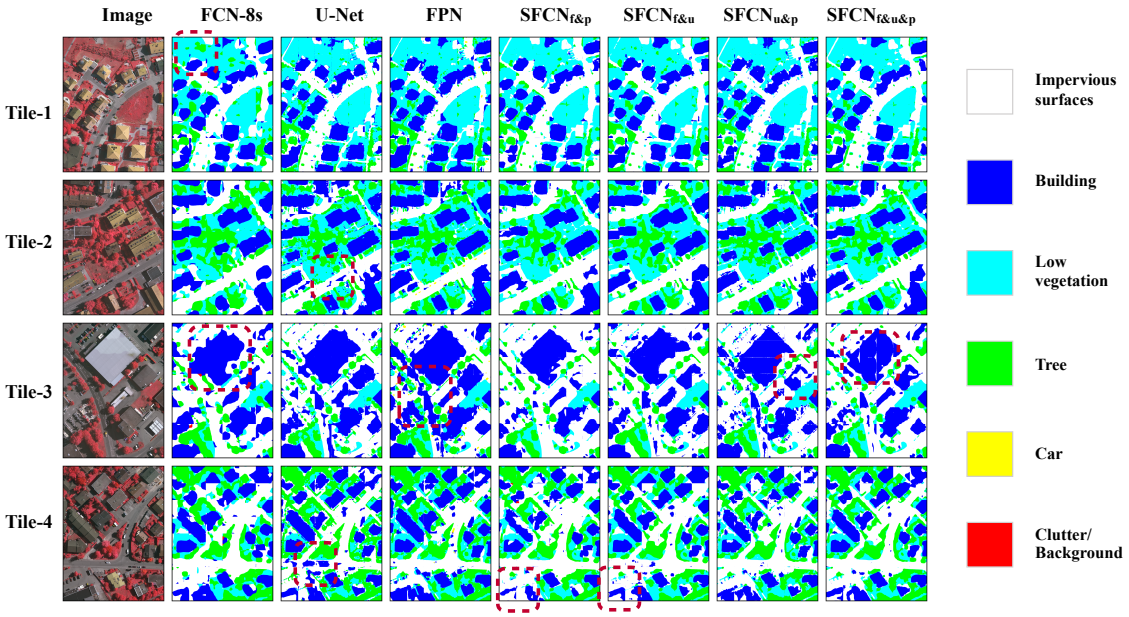


Figure 7. Segmentation results of FCN-8s, U-Net, and FPN and optimized $SFCN_{u\&p}$ for testing areas including Tile-1, -2, -3, and -4. Predicted land-covers are represented with six colors.

3.4. Computational Efficiency

All experiments are implemented and tested on a Sakura Internet Server(https://www.sakura.ad.jp/) equipped with one NVIDIA Tesla V100 GPU (https://www.nvidia.com/en-us/data-center/tesla-v100/) and installed with 64-bit Ubuntu 16.04 LTS. To eliminate the effect of some hyperparameters,

for all models, the size of batch and number of iteration are fixed to 24 and 1000, respectively. The Adam stochastic optimizer, which is running at default setting (lr = 2^{-4}, betas = [0.9, 0.999]), is used for training different models.

Table 5 shows the computating speeds in frames per second (FPS) of these methods. In training period, three basic model are processed at 41.4 FPS(FCN-8s), 59.4 FPS(U-Net), and 54.6 FPS(FPN), respectively. When compared to basic models, the ensemble methods are much slower. As the number of basic models increases (e.g., 3 in SFCN$_{f\&u\&p}$ vs. 2 in SFCN$_{f\&p}$,SFCN$_{f\&u}$, and SFCN$_{u\&p}$), the training speed decreases. Even with the same number of basic models, because of the difference in model combination, the training speeds are different. Generally, a combination of fast basic models can form a faster ensemble model (e.g., 41.8 FPS of SFCN$_{u\&p}$ vs. 31.2 FPS of SFCN$_{f\&p}$). In testing period, these methods achieved 1.3–2.1 times the processing speed. Interestingly, SFCN$_{f\&u\&p}$ has the most significant performance difference (57.6 vs. 27.2, 2.1x) between the training and testing stages.

Table 5. Comparison of the computational efficiencies of FCN-8s, U-Net, FPN and four ensemble fully conventional networks. For each column, the highest mean values and lowest SD are highlighted in **bold**.

Methods	Training FPS		Testing FPS	
	Mean	**SD**	**Mean**	**SD**
FCN-8s	41.4	**0.2**	67.1	0.3
U-Net	**59.4**	0.4	**75.4**	1.3
FPN	54.6	2.0	74.7	4.4
SFCN$_{f\&p}$	31.2	**0.2**	61.7	0.7
SFCN$_{f\&u}$	33.4	0.8	63.2	0.9
SFCN$_{u\&p}$	41.8	0.9	67.7	3.0
SFCN$_{f\&u\&p}$	27.2	**0.2**	57.6	**0.1**

4. Discussions

4.1. Regarding the Proposed Feature Alignment Framework

Deep-learning methods, especially FCN-based models, are widely adopted for automatic building extraction from large-scale aerial images [57,58]. Compared to conventional methods, the FCN-based models significantly improve segmentation performance when tested on various benchmark datasets [59,60]. Recently, more advanced FCN-based models have enhanced feature representation capabilities to achieve better model performance (e.g., FPN, MC-FCN, and BR-Net). However, the increased representation capability and as complexity of the models usually lead to overfitting. Ensemble learning, which utilizes several different networks to generate a weighted prediction, is a promising option to avoid overfitting.

In this paper, we designed four SFCNs and proposed a novel feature alignment framework to enhance the performance of the ensemble framework. In contrast to existing ensemble approaches which mainly focus on adding numbers or trying different combinations of basic models, the proposed framework introduces alignment loss to control the similarity and consistency of features extracted from different basic models. Through feature alignment, the proposed ensemble method can achieve a balance between variety and similarity so better predictions can be achieved from weaker basic models. Qualitative and quantitative results on the testing tiles demonstrated the effectiveness of our proposed stacked fully convolution networks as well as feature alignment framework. Additionally, because of its flexibility, this framework can easily extend to ensemble learning architectures using varied numbers of basic models.

4.2. Accuracies, Uncertainties, and Limitations

From the sensitivity analysis, different ensemble models show a similar trend that as the weight of alignment loss increases, the performance of the model will increase first and, after a certain level, decline (see details in Figure 5). The interpolation of this trend is: (1) When there is no feature alignment(λ=0), features extracted from different basic models are so diverse that they might have different predictions for certain locations. An ensemble of these features doesn't bring better results. (2) When feature alignment is added, at early stages, a higher value of λ forces features extracted from different basic models to be closer to each other so that they can compromise on specific locations and generate better overall predictions. However, if λ rises beyond the optimal value, the extracted features might be too similar to each other, and there will not be enough variety. Thus, the performance of the ensemble method will regress to that of a single basic model. This observation indicates that the feature alignment framework can help achieve a balance in similarity and variety of features in ensemble learning.

Among the methods, the proposed SFCNs with feature alignment ($SFCN_{u\&p}$, +FA) shows the highest values for all evaluation metrics. The values of f1-score, jaccard index, and kappa coefficient are 0.785, 0.646, and 0.742, respectively. SFCN models using two basic models ($SFCN_{f\&p}$, $SFCN_{f\&u}$, and $SFCN_{u\&p}$), with or without feature alignment (i.e., $+/-$ FA), show significantly different performances. Ensemble models with proper weights for alignment loss are generally better than their counterparts without alignment loss. Especially for $SFCN_{f\&u}$, optimal feature alignment gains increments of 4.2% (0.772 vs. 0.741) for f1-score, 6.8% (0.629 vs. 0.589) for jaccard index, and 5.5% (0.727 vs. 0.689) for kappa coefficient. These results indicate that introducing feature alignment leads to better performance of the ensemble model. However, for ensemble models using three basic models ($SFCN_{f\&u\&p}$), the values of jaccard index and kappa coefficient only increase about 1.6% (0.640 vs. 0.630) and 1.2%(0.736 vs. 0.727), respectively. The improvement caused by feature alignment for $SFCN_{f\&u\&p}$ is not significant. Additionally, when compared to the best basic model (FPN), the optimized ensemble model doesn't show big improvements (see details in Figure 6 b).

Through analysis of computating speed, we observed a significant decrease in computational efficiency at the training stage when applying ensemble learning. Of four SFCN models, the model with three basic models ($SFCN_{f\&u\&p}$) is much slower than the models with two basic models ($SFCN_{f\&p}$, $SFCN_{f\&u}$, and $SFCN_{u\&p}$). Because of the decrease in computational efficiency, even though feature alignment can be easily extended to the ensemble model with all basic models, the proposed ensemble model might not be suitable for the analysis of very large areas (e.g., automatic mapping of entire country).

5. Conclusions

In this paper, we propose a novel feature alignment framework for efficient ensemble learning of fully convolutional networks. The proposed framework can be seamlessly integrated with ensemble learning models with variant number of basic models to regulate a balance in similarity and variety of the features extracted from different branches. Their performances are verified by VHR image dataset with multi-label segmentic information. The ensemble models with proposed feature alignment show significantly better performance than existing methods. In $SFCN_{f\&u}$, optimal feature alignment gains increments of 4.2% (0.772 vs. 0.741), 6.8% (0.629 vs. 0.589), and 5.5% (0.727 vs. 0.689) for f1-score, jaccard index, and kappa coefficient, respectively. Sensitivity analysis demonstrated that feature alignment plays an important role in controlling the balance between similarity and variety of the ensemble model. In future studies, we will further optimize our feature alignment framework to achieve better performance in more complex ensemble learning architectures.

Author Contributions: G.W., Y.G., and X.S. (Xiaowei Shao) conceived and designed the experiments; G.W. performed the experiments; G.W., Y.G, and X.S. (Xiaoya Song) analyzed the data; Z.G., H.Z., X.S. (Xiaodan Shi), and R.S. contributed reagents/materials/analysis/tools; G.W. and Y.G. wrote the paper. All authors read and approved the submitted manuscript.

Acknowledgments: We want to thank International Society for Photogrammetry and Remote Sensing (ISPRS) for their kind open-souring of the data and SAKURA Internet Inc. for providing us the *koukaryoku* GPU server. We also gratefully acknowledge the financial support from the China Scholarship Council (CSC) for author G.W.(No.201808050165).

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
FCN	Fully Convolutional Networks
SFCN	Stacked Fully Convolutional Networks
FPN	Feature Pyramid Networks
FA	Feature Alignment

References

1. Yang, X.; Wu, Y.; Dang, H. Urban Land Use Efficiency and Coordination in China. *Sustainability* **2017**, *9*, 410. [CrossRef]
2. Abbott, E.; Powell, D. Land-vehicle navigation using GPS. *Proc. IEEE* **1999**, *87*, 145–162. [CrossRef]
3. Stow, D.A.; Hope, A.; McGuire, D.; Verbyla, D.; Gamon, J.; Huemmrich, F.; Houston, S.; Racine, C.; Sturm, M.; Tape, K.; et al. Remote sensing of vegetation and land-cover change in Arctic Tundra Ecosystems. *Remote Sens. Environ.* **2004**, *89*, 281–308. [CrossRef]
4. Heilman, G.E.; Strittholt, J.R.; Slosser, N.C.; Dellasala, D.A. Forest Fragmentation of the Conterminous United States: Assessing Forest Intactness through Road Density and Spatial Characteristics: Forest fragmentation can be measured and monitored in a powerful new way by combining remote sensing, geographic information systems, and analytical software. *AIBS Bull.* **2002**, *52*, 411–422.
5. Hamre, L.N.; Domaas, S.T.; Austad, I.; Rydgren, K. Land-cover and structural changes in a western Norwegian cultural landscape since 1865, based on an old cadastral map and a field survey. *Landsc. Ecol.* **2007**, *22*, 1563–1574. [CrossRef]
6. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [CrossRef]
7. Ma, L.; Li, M.; Ma, X.; Cheng, L.; Du, P.; Liu, Y. A review of supervised object-based land-cover image classification. *ISPRS J. Photogramm. Remote Sens.* **2017**, *130*, 277–293. [CrossRef]
8. Glasbey, C.A. An analysis of histogram-based thresholding algorithms. *CVGIP: Graph. Model. Image Process.* **1993**, *55*, 532–537. [CrossRef]
9. Kanopoulos, N.; Vasanthavada, N.; Baker, R.L. Design of an image edge detection filter using the Sobel operator. *IEEE J. Solid-State Circuits* **1988**, *23*, 358–367. [CrossRef]
10. Canny, J. A computational approach to edge detection. In *Readings in Computer Vision*; Elsevier: New York, NY, USA, 1987; pp. 184–203.
11. Wu, Z.; Leahy, R. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1993**, *15*, 1101–1113. [CrossRef]
12. Zhen, D.; Zhongshan, H.; Jingyu, Y.; Zhenming, T. FCM Algorithm for the Research of Intensity Image Segmentation. *Acta Electron. Sin.* **1997**, *5*, 39–43.
13. Tremeau, A.; Borel, N. A region growing and merging algorithm to color segmentation. *Pattern Recognit.* **1997**, *30*, 1191–1203. [CrossRef]
14. Ozer, S.; Langer, D.L.; Liu, X.; Haider, M.A.; van der Kwast, T.H.; Evans, A.J.; Yang, Y.; Wernick, M.N.; Yetik, I.S. Supervised and unsupervised methods for prostate cancer segmentation with multispectral MRI. *Med. Phys.* **2010**, *37*, 1873–1883. [CrossRef]
15. Li, M.; Zang, S.; Zhang, B.; Li, S.; Wu, C. A review of remote sensing image classification techniques: The role of spatio-contextual information. *Eur. J. Remote Sens.* **2014**, *47*, 389–411. [CrossRef]
16. Viola, P.; Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Kauai, HI, USA, 8–14 December 2001; IEEE: Piscataway, NJ, USA, 2001, Volume 1, p. I.

17. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; IEEE: Piscataway, NJ, USA; 1999, Volume 2, pp. 1150–1157.
18. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
19. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), San Diego, CA, USA, 20–26 June 2005; IEEE: Piscataway, NJ, USA, 2005, Volume 1, pp. 886–893.
20. Inglada, J. Automatic recognition of man-made objects in high resolution optical remote sensing images by SVM classification of geometric image features. *ISPRS J. Photogramm. Remote Sens.* **2007**, *62*, 236–248. [CrossRef]
21. Aytekin, Ö.; Zöngür, U.; Halici, U. Texture-based airport runway detection. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 471–475. [CrossRef]
22. Dong, Y.; Du, B.; Zhang, L. Target detection based on random forest metric learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 1830–1838. [CrossRef]
23. Gletsos, M.; Mougiakakou, S.G.; Matsopoulos, G.K.; Nikita, K.S.; Nikita, A.S.; Kelekis, D. A computer-aided diagnostic system to characterize CT focal liver lesions: Design and optimization of a neural network classifier. *IEEE Trans. Inf. Technol. Biomed.* **2003**, *7*, 153–162. [CrossRef] [PubMed]
24. LeCun, Y.; Bengio, Y.; others. Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Networks* **1995**, *3361*, 1995.
25. Ciresan, D.; Giusti, A.; Gambardella, L.M.; Schmidhuber, J. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems*; 2012; pp. 2843–2851. Available online: https://papers.nips.cc/paper/4741-deep-neural-networks-segment-neuronal-membranes-in-electron-microscopy-images (accessed on 1 March 2019).
26. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 3431–3440.
27. Guo, Z.; Shao, X.; Xu, Y.; Miyazaki, H.; Ohira, W.; Shibasaki, R. Identification of village building via Google Earth images and supervised machine learning methods. *Remote Sens.* **2016**, *8*, 271. [CrossRef]
28. 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 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–9.
29. Li, L.; Liang, J.; Weng, M.; Zhu, H. A Multiple-Feature Reuse Network to Extract Buildings from Remote Sensing Imagery. *Remote Sens.* **2018**, *10*, 1350. [CrossRef]
30. Ronneberger, O.; Fischer, P.; Brox, T. U–Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switherland, 2015; pp. 234–241.
31. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–15 July 2017; IEEE: Piscataway, NJ, USA, 2017; Volume 1, p. 4.
32. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; IEEE: Piscataway, NJ, USA, 2017; pp. 1520–1528.
33. Wu, G.; Shao, X.; Guo, Z.; Chen, Q.; Yuan, W.; Shi, X.; Xu, Y.; Shibasaki, R. Automatic Building Segmentation of Aerial Imagery Using Multi-Constraint Fully Convolutional Networks. *Remote Sens.* **2018**, *10*, 407. [CrossRef]
34. Wu, G.; Guo, Z.; Shi, X.; Chen, Q.; Xu, Y.; Shibasaki, R.; Shao, X. A Boundary Regulated Network for Accurate Roof Segmentation and Outline Extraction. *Remote Sens.* **2018**, *10*, 1195. [CrossRef]
35. Tetko, I.V.; Livingstone, D.J.; Luik, A.I. Neural network studies. 1. Comparison of overfitting and overtraining. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 826–833. [CrossRef]
36. Prechelt, L. Automatic early stopping using cross validation: Quantifying the criteria. *Neural Networks* **1998**, *11*, 761–767. [CrossRef]

37. Prechelt, L. Early stopping-but when? In *Neural Networks: Tricks of the Trade*; Springer: Cham, Switherland, 1998; pp. 55–69.
38. Wong, S.C.; Gatt, A.; Stamatescu, V.; McDonnell, M.D. Understanding Data Augmentation for Classification: When to Warp? In Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA), Surfers Paradise, QLD, Australia, 30 November–2 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
39. Grasmair, M.; Scherzer, O.; Haltmeier, M. Necessary and sufficient conditions for linear convergence of l1-regularization. *Commun. Pure Appl. Math.* **2011**, *64*, 161–182. [CrossRef]
40. Ng, A.Y. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In Proceedings of the 21st International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; ACM: New York, NY, USA, 2004, p. 78.
41. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
42. Rosen, B.E. Ensemble learning using decorrelated neural networks. *Connect. Sci.* **1996**, *8*, 373–384. [CrossRef]
43. Guo, Z.; Chen, Q.; Wu, G.; Xu, Y.; Shibasaki, R.; Shao, X. Village Building Identification Based on Ensemble Convolutional Neural Networks. *Sensors* **2017**, *17*, 2487. [CrossRef] [PubMed]
44. Polak, M.; Zhang, H.; Pi, M. An evaluation metric for image segmentation of multiple objects. *Image Vis. Comput.* **2009**, *27*, 1223–1227. [CrossRef]
45. Carletta, J. Assessing agreement on classification tasks: The kappa statistic. *Comput. Linguist.* **1996**, *22*, 249–254.
46. Li, E.; Femiani, J.; Xu, S.; Zhang, X.; Wonka, P. Robust rooftop extraction from visible band images using higher order CRF. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 4483–4495. [CrossRef]
47. Comer, M.L.; Delp, E.J. Morphological operations for color image processing. *J. Electron. Imaging* **1999**, *8*, 279–290. [CrossRef]
48. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the 13th European Conference on Computer Vision(ECCV 2014), Zurich, Switzerland, 6–12 September, 2014; Springer: Cham, Switherland, 2014; pp. 740–755.
49. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
50. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; Omnipress: Madison, WI, USA, 2010; pp. 807–814.
51. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; 2015; pp. 448–456.
52. Nagi, J.; Ducatelle, F.; Di Caro, G.A.; Cireşan, D.; Meier, U.; Giusti, A.; Nagi, F.; Schmidhuber, J.; Gambardella, L.M. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In Proceedings of the 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA 2011), Kuala Lumpur, Malaysia, 16–18 November 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 342–347.
53. Novak, K. Rectification of digital imagery. *Photogramm. Eng. Remote Sens.* **1992**, *58*, 339–339.
54. Shore, J.; Johnson, R. Properties of cross-entropy minimization. *IEEE Trans. Inf. Theory* **1981**, *27*, 472–482. [CrossRef]
55. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
56. Wu, G.; Guo, Z. Geoseg: A Computer Vision Package for Automatic Building Segmentation and Outline Extraction. *arXiv* **2018**, arXiv:1809.03175.
57. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 645–657. [CrossRef]
58. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Fully convolutional networkss for remote sensing image classification. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 5071–5074.

59. Demir, I.; Koperski, K.; Lindenbaum, D.; Pang, G.; Huang, J.; Basu, S.; Hughes, F.; Tuia, D.; Raskar, R. Deepglobe 2018: A challenge to parse the earth through satellite images. *arXiv* **2018**, arXiv:1805.06561.
60. Chen, Q.; Wang, L.; Wu, Y.; Wu, G.; Guo, Z.; Waslander, S.L. Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings. *ISPRS J. Photogramm. Remote Sens.* **2019**, *147*, 42–55. [CrossRef]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).