Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1515630, 14 pages
https://doi.org/10.1155/2020/1515630

Research Article

SiameseDenseU-Net-based Semantic Segmentation of Urban Remote Sensing Images

Rongsheng Dong, Lulu Bai, and Fengying Li

Guangxi Key Laboratory of Trusted Software, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

Correspondence should be addressed to Fengying Li; lfy@guet.edu.cn

Received 12 August 2019; Revised 13 January 2020; Accepted 24 February 2020; Published 23 March 2020

Academic Editor: Mariko Nakano-Miyatake

Boundary pixel blur and category imbalance are common problems that occur during semantic segmentation of urban remote sensing images. Inspired by DenseU-Net, this paper proposes a new end-to-end network—SiameseDenseU-Net. First, the network simultaneously uses both true orthophoto (TOP) images and their corresponding normalized digital surface model (nDSM) as the input of the network structure. The deep image features are extracted in parallel by downsampling blocks. Information such as shallow textures and high-level abstract semantic features are fused throughout the connected channels. The features extracted by the two parallel processing chains are then fused. Finally, a softmax layer is used to perform prediction to generate dense label maps. Experiments on the Vaihingen dataset show that SiameseDenseU-Net improves the $F1$-score by 8.2% and 7.63% compared with the Hourglass-ShapeNetwork (HSN) model and with the U-Net model. Regarding the boundary pixels, when using the same focus loss function based on median frequency balance weighting, compared with the original DenseU-Net, the small-target "car" category $F1$-score of SiameseDenseU-Net improved by 0.92%. The overall accuracy and the average $F1$-score also improved to varying degrees. The proposed SiameseDenseU-Net is better at identifying small-target categories and boundary pixels, and it is numerically and visually superior to the contrast model.

1. Introduction

In the computer vision field, semantic segmentation is an important issue. In the past few decades, many classic traditional segmentation algorithms have emerged, including region-based methods, watershed algorithms, threshold methods, and cluster-based segmentation methods. In practical applications, high-resolution images are difficult to automate for two reasons: first, their spatial resolution is higher, but their spectral resolution is lower; second, the surface texture features of small targets become visible. These two factors lead to an increase in intraclass variability in the image, while the differences between classes decrease. Image semantic segmentation aims to determine the most proposed class label for each pixel in an image drawn from a set of predefined limited labels.

In 2012, the AlexNet network, proposed by Krizhevsky et al. [1], caused a new upsurge in imaging applications in the field of deep learning. Later, Tsogkas and Kokkinos [2] combined a convolutional neural network (CNN) with a fully connected conditional random field (CRF) approach to learn the lost prior information. In the data fusion competition in 2015, Lagrange et al. [3] used a pretrained CNN model as a feature extractor to classify land cover. Paisitkriangkrai et al. [4] used true orthophoto images, a corresponding digital surface model image (DSM) and a normalized digital surface model image to train a relatively small set of CNN models. Finally, the results were further optimized using CRF. Long et al. [5] proposed a fully convolutional network (FCN) to classify images at the pixel level. Unlike the classic CNN, FCN can accept an input image of any size and restore it to the same size as the input

image, thus generating a prediction for each pixel while retaining the spatial information in the original input image.

In 2016, Volpi and Tuia [6] proposed a CNN-based system CNN-FPL. This system relies on a downsample-then-upsample architecture so that CNN learns to densely label every pixel at the original resolution of the image. In 2017, Nogueira et al. [7] compared several popular neural networks and training strategies. Their experimental results on three remote sensing image datasets indicated that fine-tuning networks is the best training strategy. Liu et al. [8] used a composed inception module to replace common convolutional layers, providing a multiscale receiving area with rich context for the network. Badrinarayanan et al. [9] proposed an architecture for semantic pixelwise segmentation termed SegNet that eliminates the need to learn to upsample. The upsampled maps are convolved with trainable filters to produce dense feature maps. In 2018, Gao et al. [10] proposed a weighted equilibrium function and a neural network based on a multifeature pyramid structure. Chen et al. [11] proposed the FCN-based model structures SNFCN and SDFCN to process VHR remote sensing images. They designed the SNFCN and SDFCN frameworks with dense-shortcut connection structures and SDFCN adds three additional identity mapping shortcut connections between the symmetrical encoder-decoder pairs. This approach ensures that the gradient information can be passed directly to the upper layers of the network. Zhang et al. [12] proposed the novel supervised deep-CNN-based OBIC framework to deal with segmented superpixels and introduced two mask policies for network models. Chen et al. [13] proposed DeepLabv3+, applied depthwise separable convolution to both the decoder modules, and added Atrous Spatial Pyramid Pooling, resulting in a faster and stronger encoder-decoder network. The TreeUNet model proposed by Yu K et al. [14] in 2019 was the first to use both an adaptive hierarchy and deep neural networks in a unified deep learning structure. TreeSegNet adopts an adaptive network to increase the classification rate at the pixelwise level. The experimental results of this algorithm on the ISPRS Potsdam dataset achieved improved results.

In summary, deep learning has been widely used for image preprocessing, target recognition tasks, high-level semantic feature extraction, and remote sensing scene understanding, but how to improve the image semantic segmentation accuracy and resolve interclass imbalance are problems that remain challenging. The main contributions of this paper are as follows:

(i) As technology has developed, the types of data available in the field of image processing have also become more diverse and include as true orthophoto images, normalized digital surface models, RGB-D images containing depth information, and even three-dimensional image data. We consider data with two different statistical characteristics and use them as simultaneous model inputs, achieving parallel processing of different remote sensing image data types. Finally, the two parallel processing chains are used to fuse the features to generate dense label maps.

(ii) We adopt a suitable loss function for semantic segmentation of remote sensing images. This function introduces a factor based on the traditional cross-entropy function to suppress the dominant position of large target categories in training and focus the training process on small-target categories. This approach both guarantees the overall accuracy and improves the segmentation effect for small-target categories.

(iii) Because of complex textures and lighting, the "building" category can easily be misclassified as an "impervious surface" by other models. Based on the visual maps of local results, our model achieves excellent performance on incomplete phenomenon of the "building" category and can segment the "building" category almost completely. We consider that this result is due to the model's excellent feature fusion capabilities.

However, the SiameseDenseU-Net model uses the max pooling layer in the downsampling block to expand the receptive field of the model, which causes some information to be lost during the downsampling process. The idea of Atrous convolution [15] can be borrowed to increase the receptive field without losing information. This is also our future research work.

2. Related Works

Through extensive research on satellite remote sensing images, researchers have found that high-resolution remote sensing images have lower spectral resolution than low-resolution remote sensing images. In most cases, only the three RGB channels are available, and category information is not fully captured. Therefore, for high-resolution remote sensing images, analyzing texture and spatial context is particularly important. Many studies have focused on extracting features from pixel spatial neighborhoods [16, 17]. The semantic segmentation task for high-resolution remote sensing images is designed to predict each pixel as a category from a predefined set of semantic categories, such as buildings, low vegetation, trees, or cars. Timely access to accurate segmentation results is critical for tasks such as urban planning, environmental monitoring, and economic forecasting.

In the past few decades, a large number of statistical methods based on spectral features, including the maximum likelihood method [18] and the K-means method [19], as well as machine learning-based methods such as neural networks (NN) [20], the support vector machine (SVM) [21], object-oriented classification [22], and sparse representation [23], have been widely used in remote sensing image segmentation tasks. However, these shallow network methods often fail to adequately consider the interrelationships between global and local samples. In recent years, deep learning methods, especially convolutional neural networks, have performed well on visual learning tasks. A deep network takes the original image as

input and transforms the graph through multiple processing layers. By aggregating the features to the gradually increasing context neighborhood, the information becomes more explicit, thus achieving a distinction between different object categories [24]. The parameter set for the entire network model is learned from the original data and tags, including the underlying layer containing the original features, the middle layer containing specific task context information, and the high level that performs the actual classification.

The remote sensing image semantic segmentation task can be described as follows: given a set of labeled training data sets, the classifier learns predictive conditional probability arithmetic from the spectral features. The original pixel intensity, a simple combination of raw values, and various types of statistical information describing the local image texture [25, 26] are typical choices for input features. Another common method is to precalculate a large number of redundant feature sets for training and then let the classifier select the optimal subset [27, 28]. In this way, less relevant information can be ignored during the feature encoding process.

HSN [8] uses inception and residual modules. The inception module enables the network to extract information from multiscale receptive areas. Residual modules are employed together with the skip connection, feeding information forward from the encoder directly to the decoder to make more effective use of the spatial information. In addition, the model uses overlap inference (OI) to mitigate the boundary effects of the image. Finally, postprocessing methods based on weighted belief propagation (WBP) visually enhance the classification results. HSN is superior to the state-of-the-art FCN [5] and FPL [6] and SegNet models in terms of overall accuracy and average F-score. The core idea underlying DenseU-Net is to connect CNN features through cascade operations and use the symmetric model structure to fuse shallow information with high-level abstract semantic features. DenseU-Net has made significant progress in the segmentation accuracy for small-target categories.

However, HSN and DenseU-Net simply add more complex processing modules to the existing network structure; they do not consider the problem of processing different statistical feature images at the same time. In the field of image semantic segmentation, it has been difficult to make a large breakthrough in network structure since the emergence of U-Net. More complex network structures not only require longer training times but also lead to model overfitting. Therefore, we focus on data processing and utilization. SiameseDenseU-Net combines two parallel DenseU-Net modules to process images with different statistical characteristics simultaneously. The resulting feature information is fused by the connected channels, which improves the network's ability to extract image features.

This study was inspired by DenseU-Net [29] and makes improvements based on its work. We further explore the potential of CNNs for end-to-end semantic segmentation of high-resolution remote sensing images.

3. Proposed Methods

SiameseDenseU-Net uses two similar parallel DenseU-Nets, each of which is composed of an encoder and a decoder. The encoder consists of five consecutive sets of downsampled blocks that double the number of feature dimensions, while the decoder consists of five consecutive sets of upsampling blocks that halve the number of feature dimensions. The input feature extracts the context information through the downsampling block to obtain a hierarchical feature and then recovers the resolution of the extracted features via the upsampling block, restoring the spatial position information lost by the encoder. Simultaneously, each downsampling block has a connection with its corresponding upsampling block. The shallow texture, color, and other details are combined with the high-level abstract semantic features to form a single DenseU-Net network. SiameseDenseU-Net fuses the features extracted from the two parallel processing chains and uses a softmax layer to predict the output characteristics to generate dense label maps.

3.1. Sampling Blocks. The D-dimensional $H \times W$ feature map is the input to the downsampling block structure. The input features first pass through two convolutional layers with a padding of 1, a stride of 1×1, and a filter size of 3×3. The input x of the downsampling block and the output features y_{d1} and y_{d2} of the two convolutional layers are subjected to a cascade operation to obtain a $3D$-dimensional feature map. Finally, after the 1×1 convolution and after dimensionality reduction, the dimension z is obtained. On the one hand, it is then passed to the corresponding upsampling block; on the other hand, it forms the input to the max pooling layer. Continuous downsampling blocks can extract CNN features, providing a wider receptive field for the network and generating more accurate classifications.

Figure 1 shows that the structures of the upsampling blocks and downsampling blocks are similar. In the upsampling block, the feature map of the D-dimensional $H \times W$ is used as an input to obtain a $2H \times 2\,W$ feature map through the 2×2 transposed convolution layer. Then, feature fusion is performed using the same size feature map from the downsampling block. The dimensionality is further reduced by a 1×1 convolution. The inputs y_{u2} and y_{u3} to the layer and the output y_{u4} from the second convolutional layer are subjected to a cascade operation to obtain a $3D$ dimensional feature map. Finally, the feature map is reduced to D dimensions by a 1×1 convolution. After all the convolutional layers are complete, a batch normalization (BN) operation and a rectified linear unit (ReLU) are performed. In the extended path phase, the resolution of the image is recovered layer by layer using successive upsampling blocks, after which the model can obtain accurate positional information.

The model structure can be formalized as follows:

Model=<x, W_i, $\sigma(\cdot)$, y_i, cascade, o, W_d, z, maxpool, x_1, W_t^T>

The meaning of each variable is described below:

(1) x: the input of the downsampling block

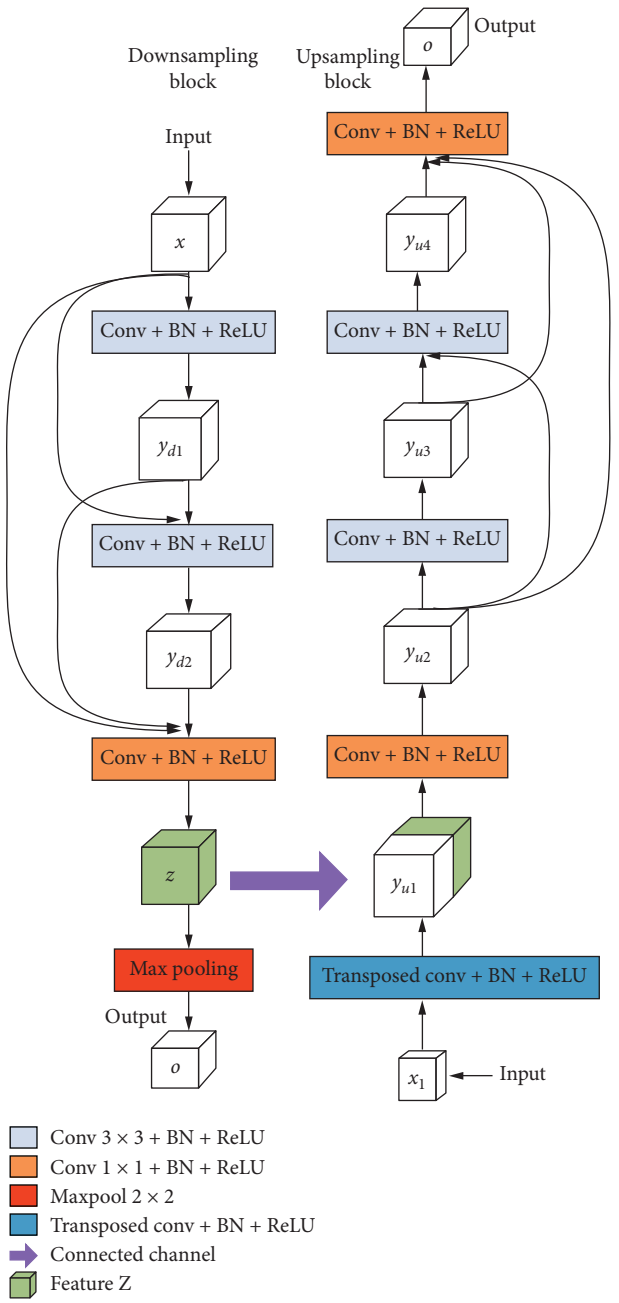


Figure 1: Details of the sampling blocks.

(2) W_i: the i-th convolution operation: the filter size is 3×3, and $i = 1, 2$

(3) $\sigma(\cdot)$: compound function, which denotes the ReLU and BN operations

(4) y_i: the output after the i-th convolution operation, where $i = 1, 2$

(5) cascade: the cascade operation

(6) o: the output

(7) W_d: the convolutional dimension reduction operation: the filter size is 1×1

(8) z: the features

(9) maxpool: the max pooling operation

(10) x_1: the input of the upsampling block

(11) W_t^T: the transposed convolution operation

The output characteristic y_{d1} of the first convolution layer is given by

$$y_{d1} = \sigma(W_1 x). \tag{1}$$

The output y_{d1} of the first convolution layer is connected to the input x of the downsampling block by a cascade operation; therefore, the output y_{d2} of the second convolution layer is given by

$$\begin{aligned} y_{d2} &= \sigma(W_2(\sigma(W_1 x) + x)) \\ &= \sigma(W_2(y_{d1} + x)). \end{aligned} \tag{2}$$

Similarly, the outputs y_{d1} and y_{d2} and the input of the downsampling block are connected to form a $3D$-dimensional feature map; then a 1×1 convolution is used for dimensionality reduction, thus reducing the dimensions of the feature map and improving the calculation efficiency. The characteristic z after dimensionality reduction is as follows:

$$\begin{aligned} z &= W_d(\sigma(W_2(\sigma(W_1 x) + x)) + \sigma(W_1 x) + x) \\ &= W_d(y_{d2} + y_{d1} + x). \end{aligned} \tag{3}$$

The dimensionally reduced feature z is passed as the input to both the max pooling layer and the corresponding upsampling block. x_1 represents the D-dimensional feature of the upsampling block input; consequently, the output characteristic y_{u1} of the transposed convolutional layer is given by

$$y_{u1} = \sigma\left(W_t^T x_1\right). \tag{4}$$

The output feature y_{u1} of the transposed convolution layer is cascaded with the feature z transmitted by the corresponding downsampling block through the connection channel, and the connected features are subjected to dimensional reduction by a 1×1 convolution:

$$\begin{aligned} y_{u2} &= W_d\left(\sigma\left(W_t^T x_1\right) + z\right) \\ &= W_d(y_{u1} + z). \end{aligned} \tag{5}$$

The dimensionally reduced feature y_{u2} is used as the input to the two layers of densely concatenated convolutional layers; thus, the output characteristic y_{u3} is given by

$$y_{u3} = \sigma(W_1 y_{u2}). \tag{6}$$

The outputs y_{u3} and y_{u2} of the convolution layer are connected by a cascade operation, and the output y_{u4} of the second layer convolution layer is as follows:

$$y_{u4} = \sigma(W_2(y_{u3} + y_{u2})). \tag{7}$$

Finally, y_{u3} and y_{u4} are connected to the input y_{u2} of the densely concatenated convolution layer through the cascade operation, and the connected feature map is subjected to dimensionality reduction using a 1×1 convolution. The output o of the final upsampling block is given by

$$o = W_d\left(y_{u4} + y_{u3} + y_{u2}\right). \tag{8}$$

The model uses a jump layer to fuse the shallow color and texture details with the high-level abstract semantic features, which can effectively improve the segmentation accuracy for relatively small classes.

3.2. Loss Function. Cross-entropy loss is a commonly applied function in image segmentation tasks. However, that loss function is calculated by summing all the pixels, which fails to consider category imbalance. Inspired by Eigen and Fergus [30], the median frequency balance is used to weight the loss of a class. The median frequency balance weights the class loss based on the ratio of the median of the sample class frequency in the training set to the target class frequency; however, this approach is insufficient to distinguish easy from difficult samples. To improve the segmentation accuracy for the small-target categories in remote sensing images, the idea of focal loss was introduced by Lin et al. [31]. By suppressing the leading role of the simple samples during training, the training process can concentrate on complex and difficult samples.

Here, N represents the number of samples in a mini-batch, C represents the number of categories, $l_c^{(n)}$ represents the true label of the one-hot encoding corresponding to sample n, and $p_c^{(n)}$ is the softmax probability of sample n being in class c. The cross-entropy loss function is defined as follows:

$$CE_{\text{loss}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} l_c^{(n)} \cdot \log\left(p_c^{(n)}\right). \tag{9}$$

The focus loss function MFB_Focal_{loss}, which is based on the median frequency balance, is defined as follows:

$$MFB\_Focal_{\text{loss}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} w_c \cdot l_c^{(n)} \cdot \left(1 - p_c^{(n)}\right)^2 \cdot \log\left(p_c^{(n)}\right). \tag{10}$$

The frequency of the category c pixel is denoted by f_c, median(f_c) is the median of the pixel frequencies of each category, and w_c represents the weight value corresponding to category c:

$$w_c = \frac{\text{median}\left(f_c\right)}{f_c}. \tag{11}$$

3.3. SiameseDenseU-Net. Inspired by DenseU-Net [29], this paper proposes a new end-to-end neural network called SiameseDenseU-Net. The network uses true orthophoto images and their corresponding normalized digital surface model images as the inputs to two DenseU-Net structures, and the downsampling blocks extract deep image features in parallel. Information is fused through the connected channels. Finally, a softmax layer is used to predict the output characteristics to generate dense label maps. The model structure is shown in Figure 2.

The dual-channel data input and the parallel model structure used for feature extraction inevitably increase model complexity, which will cause the model to spend much time on training and prediction, making it unable to quickly verify our ideas, and thus fail to improve the model. Too many parameters can also cause the model to overfit. Because the depth of the model is closely related to the feature extraction capability of the model and the size of the convolution kernel itself is already small, we cut the number of channels of the original DenseU-Net model by half when performing model clipping. Consequently, the Siamese-DenseU-Net model does not add additional parameters or calculation costs.

Table 1 gives the detailed parameters of each layer of SiameseDenseU-Net. The experimental results on the Vaihingen dataset show that SiameseDenseU-Net still performs better than does the original DenseU-Net without increasing the complexity of the model.

4. Experiments and Analysis

This experiment uses the MFB_Focal_{loss} and the cross-entropy loss function. The effectiveness of the SiameseDenseU-Net model was verified by comparing it with the original DenseU-Net and U-Net models. The HSN [8] model uses the cross-entropy loss function MFB_CE_{loss} based on the median frequency balance in this experiment; OI is used to further improve the prediction accuracy, and finally, WBP is performed during postprocessing to further improve the overall accuracy.

4.1. Dataset. The experiment used the Vaihingen dataset from the 23rd International Photogrammetry and Remote Sensing Society 2*D* Semantic Annotation Competition in 2016 [32]. The dataset contains 33 high-resolution TOP images and corresponding DSM images taken over a German town of Vaihingen. Among the 33 images in the dataset are 16 labeled images. The official ISPRS organizer also provided 33 normalized digital surface model images (nDSM) corresponding to the TOP image to limit the effects of different ground heights. There are two ground-truth versions used in the evaluation: the original version (denoted by GT) and the eroded version (indicated by erGT). Some examples are shown in Figure 3.

The experiment divided the 16 available GT images into training and testing sample sets. The training set consists of 11 images (regions 1, 3, 5, 7, 13, 17, 21, 23, 26, 32, and 37), and the test set includes 5 images (regions 11, 15, 28, 30, and 34).

The Vaihingen dataset contains six categories: impervious surfaces, low vegetation, cars, clutter/background, buildings, and trees. In the dataset, the "car" category is relatively small compared to the other categories; thus, it belongs to the small-target category, as shown in Figure 4. At the same time, in the image, the diversity of car colors also leads to large intraclass differences.

In this experiment, we cut the 11 training set images and the corresponding GT and nDSM images into 256×256 pixel images, with a 50% overlap between adjacent images. Then, each of the cut images and the corresponding GT

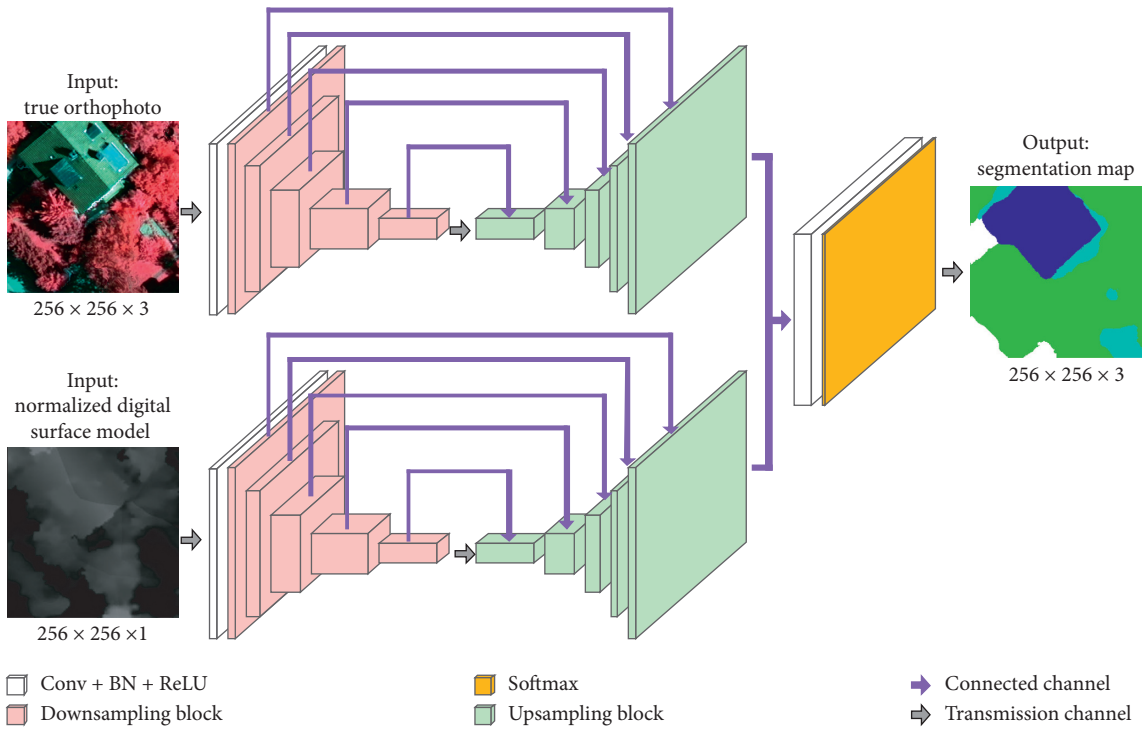


Figure 2: SiameseDenseU-Net network architecture.

image were rotated at four angles (0°, 90°, 180°, and 270°), and each rotated image was horizontally mirrored. Following this approach, each picture is represented by 8 enhanced images, including itself. These operations increased the diversity of the data.

4.2. Evaluation Index. According to the 23rd International Society for Photogrammetry and Remote Sensing 2D Semantic Annotation Competition, the overall accuracy evaluation standard is the percentage of pixels for which the correct category is predicted, and the *F*1-score is used as an evaluation criterion for measuring the segmentation accuracy of each category. The effect parameters are all between 0 and 1: the larger their values are, the higher the accuracy is. The *F*1-score formula better balances the two precision and recall parameters and thus better measures model performance. The definition of the *F*1-score is as follows:

$$F1 = \frac{2 \times \text{precision}(c) \times \text{recall}(c)}{\text{precision}(c) + \text{recall}(c)} \times 100\%. \tag{12}$$

N represents the total number of predictions, M is the number of correct prediction results, G is the sum of the predicted correct results and the unpredicted correct results, P represents the precision rate, and R represents the recall rate, and P and R are defined as follows:

$$\begin{aligned} P &= \frac{M}{N} \times 100\%, \\ R &= \frac{M}{G} \times 100\%. \end{aligned} \tag{13}$$

The percentage of the correctly predicted pixels to the total pixels is used as an evaluation criterion of the overall accuracy, in which TP represents the number of correctly predicted pixels, and AP represents the total number of all pixels. This metric is defined as follows:

$$\text{accuracy} = \frac{TP}{AP} \times 100\%. \tag{14}$$

4.3. Experimental Results. The experiment uses true orthophoto images and the normalized digital surface model as the input to SiameseDenseU-Net. These are, respectively, sent to the two parallel DenseU-Net models for training. Finally, the features extracted by the two parallel DenseU-Net models are fused, and the fused features are intensively predicted using the softmax layer to generate dense label maps. It is worth noting that the number of channels of the two parallel DenseU-Nets in the SiameseDenseU-Net model is half that of the original DenseU-Net model. Compared to the original DenseU-Net, the SiameseDenseU-Net model does not add additional parameters or computational costs.

As shown in Table 2, SiameseDenseU-Net + MFB_Focal_{loss} outperforms the original DenseU-Net + MFB_Focal_{loss} model, except on the "low vegetation" category. It also improves the *F*1-score, overall accuracy, and average *F*1-score of the other categories to varying degrees. When considering boundary pixels, the overall accuracy and average *F*1-score increased by 0.57% and 0.58%, respectively. This experiment shows that the SiameseDenseU-Net model outperforms the DenseU-Net and U-Net models without requiring additional parameters or increasing the computational cost. It is particularly

Table 1: Detailed parameters of SiameseDenseU-Net.

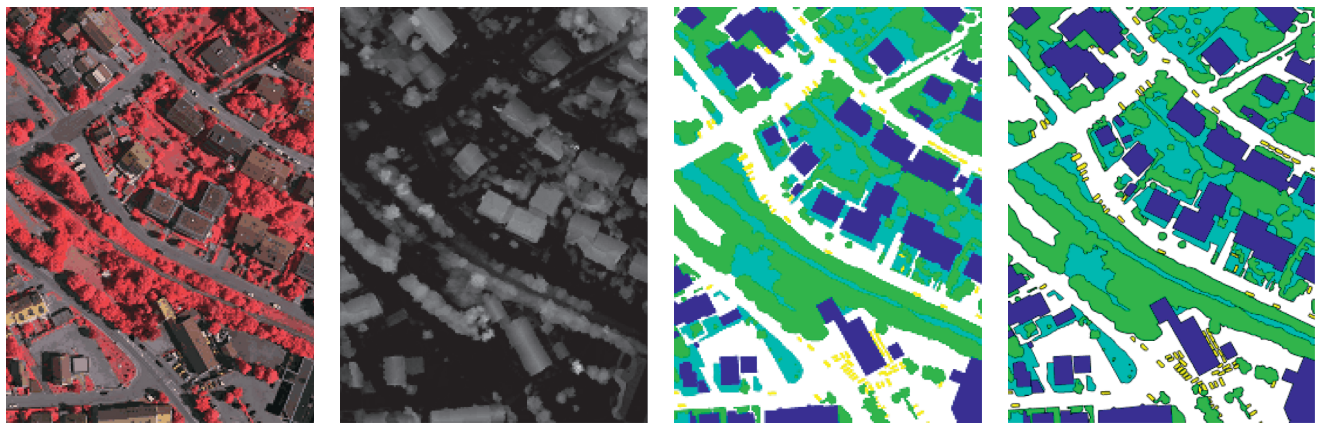
Layer name		Kernel number	Kernel size
Inconv		32	3 × 3
TOP image			
Down	Conv0_1	32	3 × 3
	Conv0_2	32	3 × 3
Block0	Conv0_3	32	1 × 1
	Maxpool0	32	2 × 2
Down	Conv1_1	64	3 × 3
	Conv1_2	64	3 × 3
Block1	Conv1_3	64	1 × 1
	Maxpool1	64	2 × 2
Down	Conv2_1	128	3 × 3
	Conv2_2	128	3 × 3
Block2	Conv2_3	128	1 × 1
	Maxpool2	128	2 × 2
Down	Conv3_1	256	3 × 3
	Conv3_2	256	3 × 3
Block3	Conv3_3	256	1 × 1
	Maxpool3	256	2 × 2
Down	Conv4_1	256	3 × 3
	Conv4_2	256	3 × 3
Block4	Conv4_3	256	1 × 1
	Maxpool4	256	2 × 2
Up	TransposedConv0	256	2 × 2
	Conv5_1	256	1 × 1
	Conv5_2	256	3 × 3
Block0	Conv5_3	256	3 × 3
	Conv5_4	256	1 × 1
Up	TransposedConv1	256	2 × 2
	Conv6_1	128	1 × 1
	Conv6_2	128	3 × 3
Block1	Conv6_3	128	3 × 3
	Conv6_4	128	1 × 1
Up	TransposedConv2	128	2 × 2
	Conv7_1	64	1 × 1
	Conv7_2	64	3 × 3
Block2	Conv7_3	64	3 × 3
	Conv7_4	64	1 × 1
Up	TransposedConv3	64	2 × 2
	Conv8_1	32	1 × 1
	Conv8_2	32	3 × 3
Block3	Conv8_3	32	3 × 3
	Conv8_4	32	1 × 1
Up	TransposedConv4	32	2 × 2
	Conv9_1	32	1 × 1
	Conv9_2	32	3 × 3
Block4	Conv9_3	32	3 × 3
	Conv9_4	32	1 × 1
nDSM			
Down	Conv0_1	32	3 × 3
	Conv0_2	32	3 × 3
Block0	Conv0_3	32	1 × 1
	Maxpool0	32	2 × 2
Down		64	3 × 3
		64	3 × 3
Block1		64	1 × 1
		64	2 × 2
Down	Conv2_1	128	3 × 3
	Conv2_2	128	3 × 3
Block2	Conv2_3	128	1 × 1
	Maxpool2	128	2 × 2
Down	Conv3_1	256	3 × 3
	Conv3_2	256	3 × 3
Block3	Conv3_3	256	1 × 1
	Maxpool3	256	2 × 2
Down	Conv4_1	256	3 × 3
	Conv4_2	256	3 × 3
Block4	Conv4_3	256	1 × 1
	Maxpool4	256	2 × 2
Up	TransposedConv0	256	2 × 2
	Conv5_1	256	1 × 1
	Conv5_2	256	3 × 3
Block0	Conv5_3	256	3 × 3
	Conv5_4	256	1 × 1
Up	TransposedConv1	256	2 × 2
	Conv6_1	128	1 × 1
	Conv6_2	128	3 × 3
Block1	Conv6_3	128	3 × 3
	Conv6_4	128	1 × 1
Up	TransposedConv2	128	2 × 2
	Conv7_1	64	1 × 1
	Conv7_2	64	3 × 3
Block2	Conv7_3	64	3 × 3
	Conv7_4	64	1 × 1
Up	TransposedConv3	64	2 × 2
	Conv8_1	32	1 × 1
	Conv8_2	32	3 × 3
Block3	Conv8_3	32	3 × 3
	Conv8_4	32	1 × 1
Up	TransposedConv4	32	2 × 2
	Conv9_1	32	1 × 1
	Conv9_2	32	3 × 3
Block4	Conv9_3	32	3 × 3
	Conv9_4	32	1 × 1
Outconv (concatenata TOP and nDSM feature)		6	3 × 3

noteworthy that when considering edge pixels, the newly proposed SiameseDenseU-Net + MFB_Focal_{loss} increases the $F1$-score of the small-target "car" category by 0.92% compared to the original DenseU-Net + MFB_Focal_{loss} model. That is, SiameseDenseU-Net + MFB_Focal_{loss} achieves excellent performance at enhancing the semantic segmentation of small-target categories.

It can also be seen from Table 2 that SiameseDenseU-Net + MFB_Focal_{loss} achieves a better overall accuracy than does HSN + OI + WBP even without postprocessing, reaching 86.2%. Moreover, its $F1$-scores on each category are better than those of HSN + OI + WBP. Especially for the small-target "car" category, SiameseDenseU-Net + MFB_Focal_{loss}'s $F1$-score increased by 8.2% over that of HSN + OI + WBP.

When ignoring the boundary pixels (erGT), the performances of all the networks are better than when the boundary pixel are considered (GT) due to object boundary ambiguity.

The experiments on the Vaihingen dataset show that the SiameseDenseU-Net model can better identify small-target



(a) (b) (c) (d)

Figure 3: Vaihingen dataset samples. (a) TOP. (b) nDSM. (c) GT. (d) erGT.

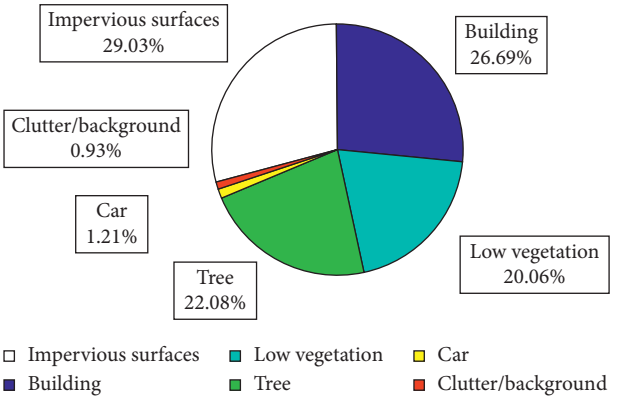


Figure 4: Pixel ratio for each category in the training sample set.

Table 2: Performances of the different models.

Models		Impervious surfaces	Building	Low vegetation	Tree	Car	Ave. $F1$	Overall. Acc
GT	HSN	87.57	92.20	75.03	84.44	75.16	82.88	84.92
	HSN + OI	88.01	92.37	75.83	84.86	76.50	83.51	85.38
	HSN + OI + WBP	88.00	92.34	75.92	84.86	75.95	83.41	85.39
	U-Net + CE_{loss}	85.82	90.51	73.62	83.33	67.24	80.10	83.46
	U-Net + MFB_Focal_{loss}	85.64	90.31	72.86	83.25	76.52	81.72	83.21
	DenseU-Net + CE_{loss}	87.77	92.42	75.89	84.36	83.21	84.73	85.28
	DenseU-Net+ MFB_Focal_{loss}	88.18	92.50	**76.23**	84.63	83.23	84.95	85.63
	SiameseDenseU-Net + CE_{loss}	88.31	92.49	76.22	84.76	82.77	84.91	85.76
	SiameseDenseU-Net + MFB_Focal_{loss}	**88.93**	**93.48**	76.08	**85.03**	**84.15**	**85.53**	**86.20**
erGT	HSN	90.89	94.51	78.83	87.84	81.87	86.79	88.32
	HSN + OI	91.32	94.66	79.73	88.30	83.60	87.52	88.79
	HSN + OI + WBP	91.34	94.67	79.83	88.31	83.59	87.55	88.82
	U-Net + CE_{loss}	88.92	92.62	77.45	86.70	75.54	84.24	86.75
	U-Net + MFB_Focal_{loss}	88.84	92.40	76.70	86.56	82.68	85.44	86.50
	DenseU-Net + CE_{loss}	90.89	94.57	79.77	87.74	90.83	88.76	88.57
	DenseU-Net + MFB_Focal_{loss}	91.30	94.64	80.17	87.99	90.96	89.01	88.92
	SiameseDenseU-Net + CE_{loss}	91.40	94.59	**80.22**	88.09	90.49	88.96	89.04
	SiameseDenseU-Net + MFB_Focal_{loss}	**92.08**	**95.57**	79.96	**88.42**	**91.33**	**89.47**	**89.49**

Note: bold font indicates the best results.

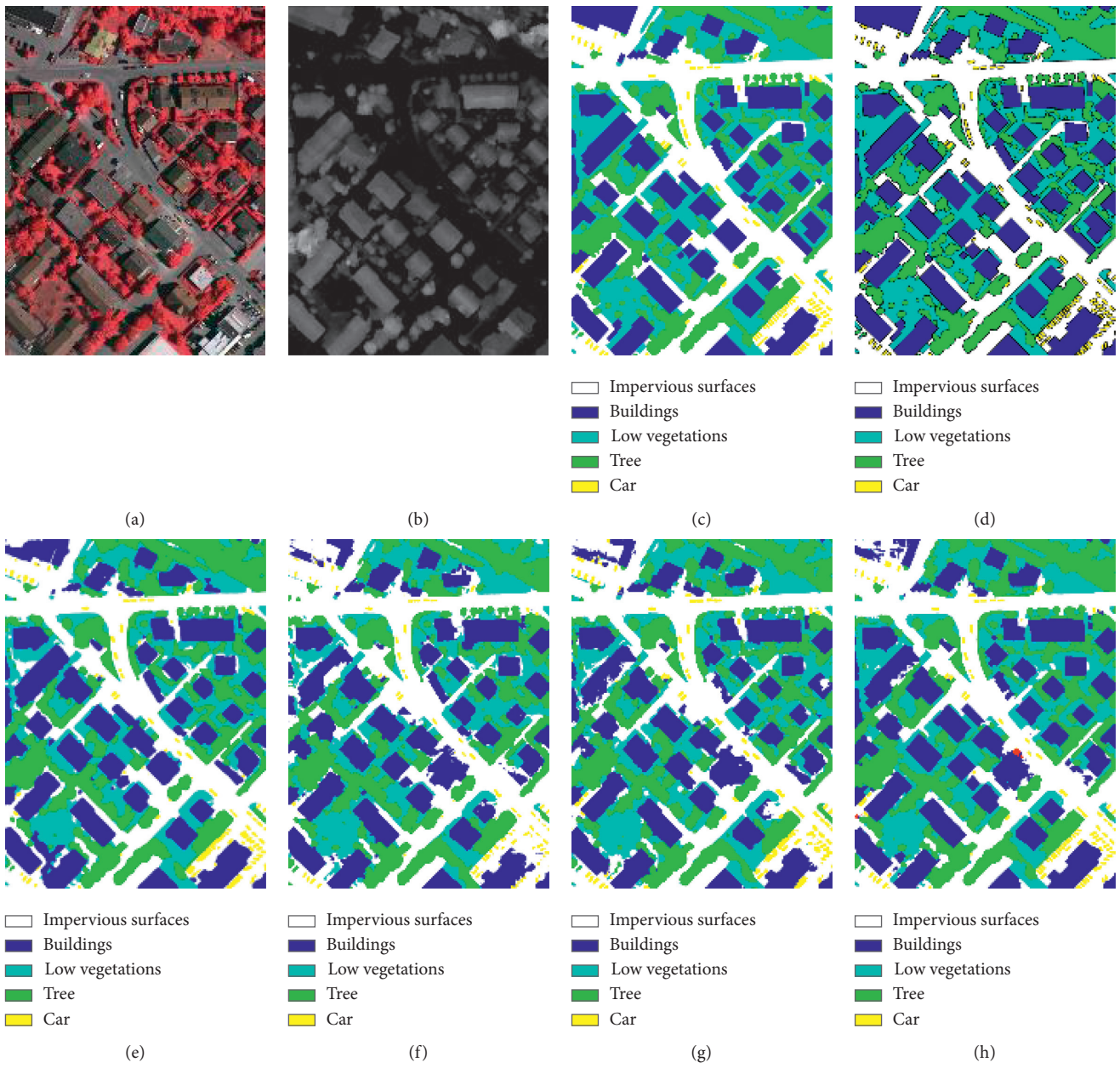


Figure 5: Continued.

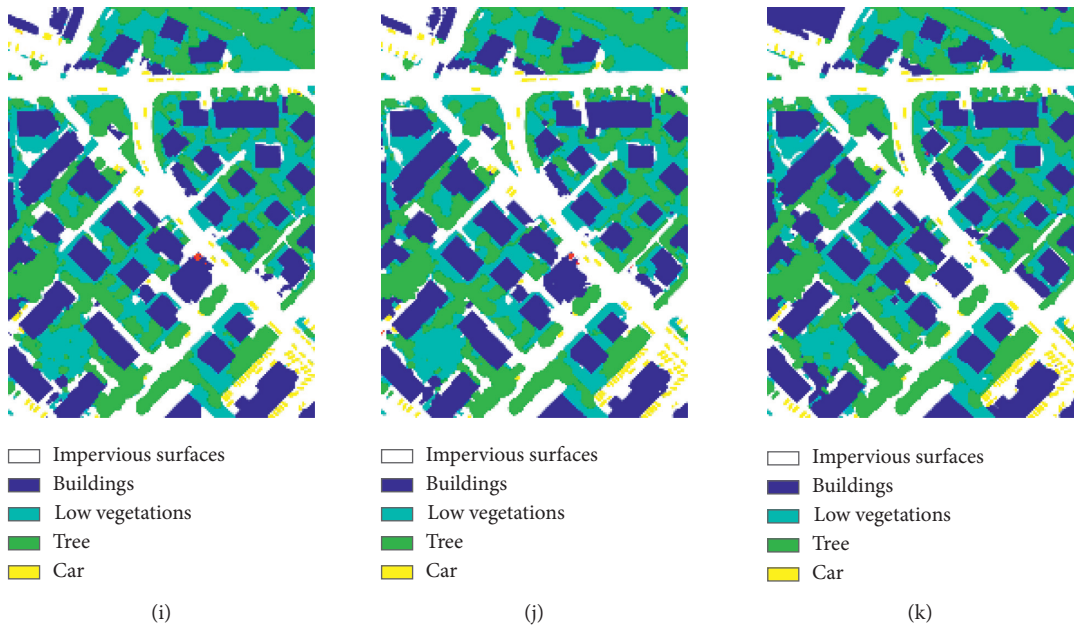


Figure 5: Visual comparison of the global results of different models. (a) TOP. (b) nDSM. (c) GT. (d) erGT. (e) HSN + OI + WBP. (f) U-Net + CE_{loss}. (g) U-Net + MFB_Focal_{loss}. (h) DenseU-Net + CE_{loss}. (i) DenseU-Net + MFB_Focal_{loss}. (j) SiameseDenseU-Net + CE_{loss}. (k) SiameseDenseU-Net + MFB_Focal_{loss}.

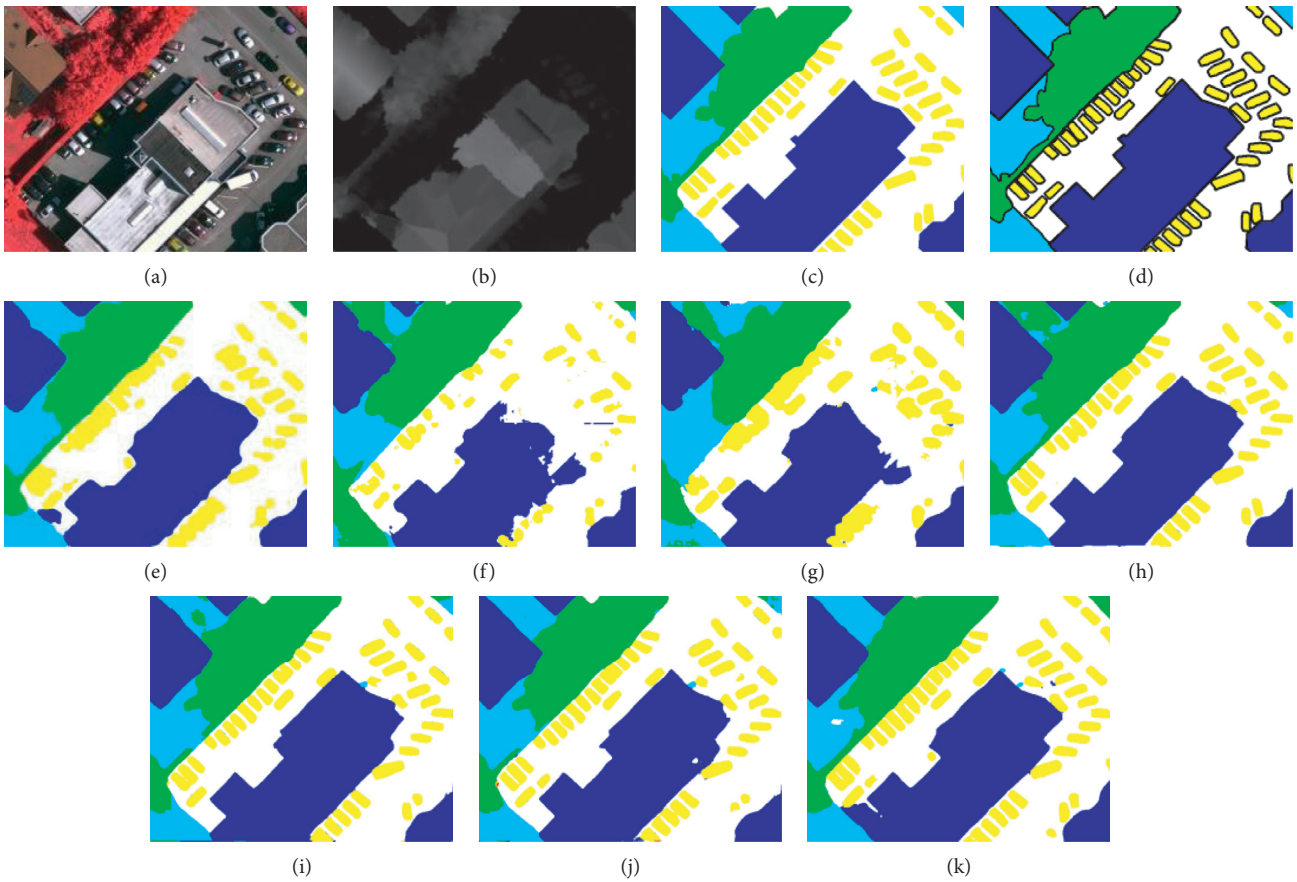


Figure 6: Local comparison of the experimental results for the "car" category. (a) TOP. (b) nDSM. (c) GT. (d) erGT. (e) HSN + OI + WBP. (f) U-Net + CE_{loss}. (g) U-Net + MFB_Focal_{loss}. (h) DenseU-Net + CE_{loss}. (i) DenseU-Net + MFB_Focal_{loss}. (j) SiameseDenseU-Net + CE_{loss}. (k) SiameseDenseU-Net + MFB_Focal_{loss}.

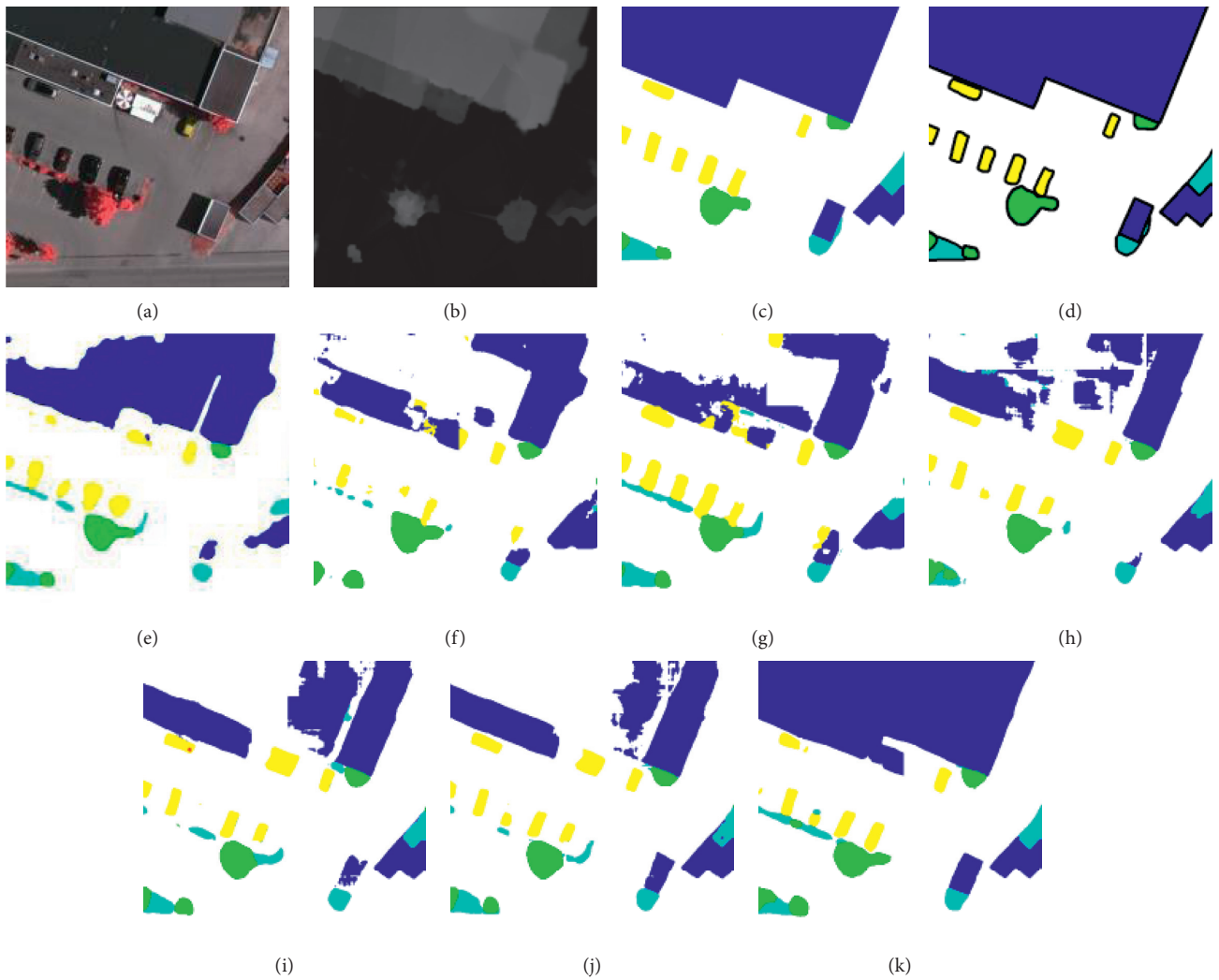


Figure 7: Visual comparison of the local results on the "buildings" category. (a) TOP. (b) nDSM. (c) GT. (d) erGT. (e) HSN + OI + WBP. (f) U-Net + CE_{loss}. (g) U-Net + MFB_Focal_{loss}. (h) DenseU-Net + CE_{loss}. (i) DenseU-Net + MFB_Focal_{loss}. (j) SiameseDenseU-Net + CE_{loss}. (k) SiameseDenseU-Net + MFB_Focal_{loss}.

"car" categories while maintaining its overall accuracy, making it numerically and visually superior to the existing DenseU-Net, U-Net, and HSN models.

Figure 5 shows the experimental results of different models on the global image. It can be seen that Siamese-DenseU-Net + MFB_Focal_{loss} outperforms the other models on the Vaihingen dataset.

Figure 6 shows a local comparison of the experimental results on the "car" category. For the small-target "car" category, the segmentation effect of the DenseU-Net + MFB_Focal_{loss} model is already excellent, but the new SiameseDenseU-Net + MFB_Focal_{loss} model performs even better on "car" boundary pixels and defective "cars".

Both Figures 7 and 8 show a partial segmentation visual comparison of the "building" categories of different models. In Figure 7, the SiameseDenseU-Net + MFB_Focal_{loss} model is optimal for semantic segmentation of the "building" category. In the "buildings" category in the upper left corner of the image, some pixels are misclassified by the other models as "impervious surfaces" due to their complex textures and lighting, which causes the "building" category in the image to be incomplete. Thus, the SiameseDenseU-Net + MFB_Focal_{loss} model also solves the problem of defective "buildings" and completely segments the "buildings" category.

As shown in Figure 8, the SiameseDenseU-Net + MFB_Focal_{loss} model is optimal for performing semantic segmentation of the "building" category boundary pixels. The boundary pixels of the "building" category in the image are jagged, and the other models fail to recognize these boundary pixels. Some models predict that the "building" category image is incomplete. In contrast, the SiameseDenseU-Net + MFB_Focal_{loss} model not only solves the problem of the incomplete "building" image but also accurately identifies the boundary pixels of the "building" category.

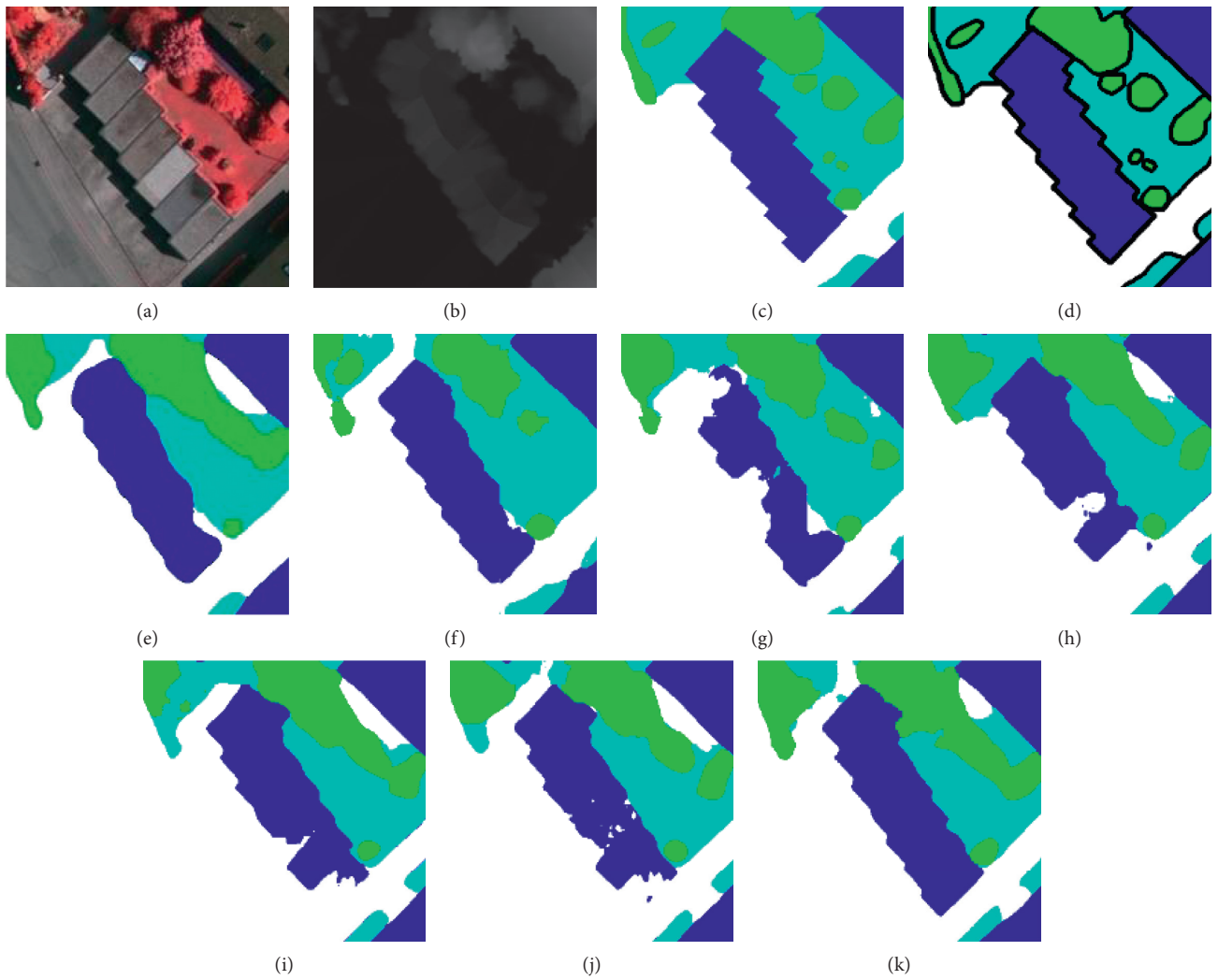


Figure 8: Visual comparison of the boundary pixels in the "building" category. (a) TOP. (b) nDSM. (c) GT. (d) erGT. (e) HSN + OI + WBP. (f) U-Net + CE_{loss}. (g) U-Net + MFB_Focal_{loss}. (h) DenseU-Net + CE_{loss}. (i) DenseU-Net + MFB_Focal_{loss}. (j) SiameseDenseU-Net + CE_{loss}. (k) SiameseDenseU-Net + MFB_Focal_{loss}.

5. Conclusions

To solve the problems of blurred boundary pixels and unbalanced categories in urban remote sensing image segmentation tasks, this paper proposed an end-to-end SiameseDenseU-Net model based on DenseU-Net. The model uses two parallel DenseU-Net networks to extract features from true orthophoto images and their corresponding normalized digital surface model images. Two parallel downsampling blocks extract image features at the same time. The features of the downsampling blocks are transmitted to the upsampling blocks for feature fusion through the connected channel. Finally, a softmax layer is used to perform prediction and generate dense label maps. The number of channels in the SiameseDenseU-Net model is half that of the original DenseU-Net model. The experimental results show that the SiameseDenseU-Net model is better at identifying the small "car" category and the "building" category without requiring additional parameters or increasing the calculation cost, and it also better solves the incomplete phenomenon of the "building" category. Simultaneously, it improves the overall accuracy and the average $F1$-score and outperforms the compared models with regard to both numerical and visual comparisons.

Data Availability

The data used to support the results of this study can be obtained by visiting http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61762024 and in

part by the Natural Science Foundation of Guangxi Province under Grant nos. 2017GXNSFDA198050 and 2016GXNSFAA380054.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.

[2] S. Tsogkas and I. Kokkinos, "Semantic part segmentation with deep learning," 2015, https://arxiv.org/pdf/1505.02438.pdf.

[3] A. Lagrange, B. Le Saux, A. Beaupère et al., "Benchmarking classification of earth-observation data: from learning explicit features to convolutional networks," in *Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE*, Milan, Italy, July 2015.

[4] S. Paisitkriangkrai, J. Sherrah, P. Janney, A. Van-Den Hengel et al., "Effective semantic pixel labelling with convolutional networks and conditional random fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Boston, MA, USA, June 2015.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, MA, USA, June 2015.

[6] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 881–893, 2016.

[7] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539–556, 2017.

[8] Y. Liu, D. Minh Nguyen, N. Deligiannis, W. Ding, and A. Munteanu, "Hourglass-shapenetwork based semantic segmentation for high resolution aerial imagery," *Remote Sensing*, vol. 9, no. 6, p. 522, 2017.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[10] X. Gao, X. Sun, Y. Zhang et al., "An end-to-end neural network for road extraction from remote sensing imagery by multiple feature pyramid network," *IEEE Access*, vol. 6, pp. 39401–39414, 2018.

[11] G. Chen, X. Zhang, Q. Wang, F. Dai, Y. Gong, and K. Zhu, "Symmetrical dense-shortcut deep fully convolutional networks for semantic segmentation of very-high-resolution remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 5, pp. 1633–1644, 2018.

[12] X. Zhang, Q. Wang, G. Chen et al., "An object-based supervised classification framework for very-high-resolution remote sensing images using convolutional neural networks," *Remote Sensing Letters*, vol. 9, no. 4, pp. 373–382, 2018.

[13] L.-C. Chen, M. Hebert, C. Sminchisescu, and Y. Weiss, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.

[14] K. Yue, L. Yang, R. Li, W. Hu, F. Zhang, and W. Li, "TreeUNet: adaptive Tree convolutional neural networks for subdecimeter aerial image segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 156, pp. 1–13, 2019.

[15] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, https://arxiv.org/abs/1706.05587.

[16] P. Tokarczyk, J. D. Wegner, S. Walk, and K. Schindler, "Features, color spaces, and boosting: new insights on semantic classification of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 280–295, 2014.

[17] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 10, pp. 3747–3762, 2010.

[18] J. Sun, J. Yang, C. Zhang, W. Yun, and J. Qu, "Automatic remotely sensed image classification in a grid environment based on the maximum likelihood method," *Mathematical and Computer Modelling*, vol. 58, no. 3-4, pp. 573–581, 2013.

[19] V. Jumb, M. Sohani, and A. Shrivas, "Color image segmentation using K-means clustering and Otsu's adaptive thresholding," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 3, no. 9, pp. 72–76, 2014.

[20] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2271–2282, 2010.

[21] H. Yu, L. Gao, J. Li, S. Li, B. Zhang, and J. Benediktsson, "Spectral-spatial hyperspectral image classification using subspace-based support vector machines and adaptive Markov random fields," *Remote Sensing*, vol. 8, no. 4, p. 355, 2016.

[22] Z. P. Sugg, T. Finke, D. C. Goodrich, M. S. Moran, and S. R. Yool, "Mapping impervious surfaces using object-oriented classification in a semiarid urban region," *Photogrammetric Engineering & Remote Sensing*, vol. 80, no. 4, pp. 343–352, 2014.

[23] B. Song, P. Li, J. Li, and A. Plaza, "One-class classification of remote sensing images using kernel sparse representation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 4, pp. 1613–1623, 2016.

[24] R. Dong, M. Liu, and F. Li, "Multilayer convolutional feature aggregation algorithm for image retrieval," *Mathematical Problems in Engineering*, vol. 2019, Article ID 9794202, 12 pages, 2019.

[25] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[26] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[27] B. Fröhlich, E. Bach, I. Walde, S. Hese, C. Schmullius, and J. Denzler, "Land cover classification of satellite images using contextual information," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3/W1, 2013.

[28] R. Dong, D. Cheng, and F. Li, "Aggregating deep convolutional features for image retrieval using multi-regional cross weighting," *Journal of Computer-Aided Design & Computer Graphics*, vol. 30, no. 4, pp. 658–665, 2018.

[29] R. Dong, X. Pan, and F. Li, "DenseU-net-based semantic segmentation of small objects in urban remote sensing images," *IEEE Access*, vol. 7, pp. 65347–65356, 2019.

[30] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015.

[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October 2017.

[32] ISPRS Vaihingen 2D Semantic Labeling Dataset, http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html.