remote sensing

MDPI

Article

DSSM: A Deep Neural Network with Spectrum Separable Module for Multi-Spectral Remote Sensing Image Segmentation

Hongming Zhu [1], Rui Tan [1], Letong Han [1], Hongfei Fan [1], Zeju Wang [1], Bowen Du [1,2,*] and Sicong Liu [3] and Qin Liu [1]

[1] School of Software Engineering, Tongji University, 4800 Caoan Road Jiading District, Shanghai 201804, China; zhu_hongming@tongji.edu.cn (H.Z.); 2031570@tongji.edu.cn (R.T.); 2031543@tongji.edu.cn (L.H.); fanhongfei@tongji.edu.cn (H.F.); 1751926@tongji.edu.cn (Z.W.); qin.liu@tongji.edu.cn (Q.L.)

[2] Department of Computer Science, University of Warwick, Gibbet Hill Road, Coventry CV4 7AL, UK

[3] School of Geodesy and Geomatics, Tongji University, 1239 Siping Road Yangpu District, Shanghai 200082, China; sicong.liu@tongji.edu.cn

* Correspondence: B.Du@Warwick.ac.uk

Abstract: Over the past few years, deep learning algorithms have held immense promise for better multi-spectral (MS) optical remote sensing image (RSI) analysis. Most of the proposed models, based on convolutional neural network (CNN) and fully convolutional network (FCN), have been applied successfully on computer vision images (CVIs). However, there is still a lack of exploration of spectra correlation in MS RSIs. In this study, a deep neural network with a spectrum separable module (DSSM) is proposed for semantic segmentation, which enables the utilization of MS characteristics of RSIs. The experimental results obtained on Zurich and Potsdam datasets prove that the spectrum-separable module (SSM) extracts more informative spectral features, and the proposed approach improves the segmentation accuracy without increasing GPU consumption.

Keywords: deep neural network; image segmentation; multi-spectral images; spectrum separable

check for updates

Citation: Zhu, H.; Tan, R.; Han, L.; Fan, H.; Wang, Z.; Du, B.; Liu, S.; Liu, Q. DSSM: A Deep Neural Network with Spectrum Separable Module for Multi-Spectral Remote Sensing Image Segmentation. *Remote Sens.* **2022**, *14*, 818. https://doi.org/10.3390/rs14040818

Academic Editor: Giovanni Poggi

Received: 7 December 2021
Accepted: 31 January 2022
Published: 9 February 2022

1. Introduction

With the rapid development of remote sensing (RS) technology, multi-spectral (MS) images are able to provide increasingly complicated and effective information. As one of the most significant steps in the interpretation of RSIs, segmentation is a comprehensive research topic that includes computer vision (CV), neural networks and RS fields. Segmentation tasks in RS generally focus on extracting a specific category, e.g., water, buildings or cars, or multiple categories all together [1,2]. Today, segmentation for RSIs plays a significant role in disaster prevention and control, land-use planning, urban sprawl detection, etc. [3–6].

As deep learning (DL) technology has grown rapidly in recent years, CNN[7]- and FCN[8]-based methods have performed competitively in computer vision image (CVI) segmentation tasks [9–13], and have outperformed traditional methods in RSI segmentation [14–19]. The major differences between remote sensing images (RSIs) and CVIs are:

1. As compared with CVIs, RSIs have two major feature dimensions: spatial features and spectral features.
2. In the spatial dimension, CVIs generally have a lower resolution and a lower variety of objects. Correspondingly, the resolution in RSIs is generally hundreds of times higher than CVIs. Moreover, RSIs have a more complicated spatial distribution, more diverse object textures, and boundary patterns, and extremely unbalanced object categories.
3. In the spectral dimension, CVIs consist of red, green, and blue spectra (RGB), which indicate unitary spectrum characteristics. However, aside from visible spectra, such as

RGB, RSIs contain certain invisible spectra, such as near infrared (NIR), which make it possible to record a wide variety of object spectrum characteristics.

In recent years, many CNN- and FCN-based RSI segmentation methods have focused on spatial features by improving the effectiveness of object textures and boundary patterns and reducing the impact of unbalanced object categories in order to promote segmentation performance. For example, [14] fused semantic and spatial information and alleviated the boundary blur by introducing a channel-weighted multi-scale feature module and boundary attention module into ResNet [20]. On the basis of the infrastructure of SegNet [11], with the help of an attention mechanism, Ref. [15] proposed a lightweight end-to-end network to automatically enhance the spatial and channel features. The authors of [16] designed a multi-scale context aggregation network with adaptive spatial pooling, reduced the spatial information loss during the process of convolution and pooling process, and promoted the semantic representation capability of feature maps. Moreover, Ref. [17] added a 1 × 1 convolution and full connection (FC) layer into the atrous spatial pyramid pooling (ASPP) module, which improved the capacity of fusing multi-scale features in Deeplab. In addition, Ref. [21] proposed balanced cross entropy (BCE) loss to optimize the training of the segmentation network.

These methods, however, lose sight of the spectral features between RSIs and CVIs. As regards CVIs, spectra represent simple color characteristics, so it is reasonable that these spectra share equivalent weights in the convolution units. Correspondingly, the spectrum is extremely significant and complicated for RSIs. For example, there are various spectrum-sensitive objects in RSIs, such as water and trees; therefore, it is important to extract the correlation between spectra. Considering this factor, certain traditional RSI segmentation algorithms leverage the divergence of objects on a specific spectrum or a combination of several spectra. For instance, NDVI [22] and NDWI [23] are two typical object extraction approaches based on spectrum characteristics, and the local spectral histogram method [18,24] calculates a spectral histogram of each spectrum and obtains the qualitative discrimination results using the synthesis stage.

Recently, researchers have taken into account spectral features in CNN-based methods. One-dimensional (1D), two-dimensional (2D), and three-dimensional (3D) CNN-based methods are proposed to model spectral features [25–27]. 1D CNN-based methods exploit the spectral features by convolving spectrum-dimension vectors for each pixel, which sacrifice the spatial representation capability [28]. 2D CNN-based methods extract spectral features in a two-stage manner. Firstly, low-dimensional representations of the spectra are obtained using dimensionality reduction methods. Then, general CNNs are leveraged to explore the spatial features [29–31]. Obviously, spectral and spatial features are utilized in a dissociated manner in 2D CNN-based methods. The convolution kernels in 3D CNN-based methods are cubic rather than flat, which can be used to easily extract and fuse the spectral and spatial features [32,33]. On the one hand, only local and low-level spectral features can be explored in this way. On the other, the order of spectra can limit the feature extraction capability of 3D kernels. In summary, when tackling RSI segmentation with DL-based methods, we expect to achieve the high-dimension, abstract, and robust feature representation wherein spectral and spatial features are effectively integrated, which is crucial for subsequent classification.

As discussed above, it is worth exploring the integration of traditional methods and DL-based methods of modeling spectra. Therefore, we proposed a deep neural network with a spectrum separable module (DSSM) to explore the possibility of enhancing the capability of extracting spectral features and to improve the segmentation accuracy in MS RSIs. The source codes are available at https://github.com/RuiTan/DSSM (accessed on 18 January 2022). The main contributions of our work are as follows:

1. A model was designed to realize the self-learning fusion mechanism of MS features through a depth-separable convolution and attention mechanism, which takes dissimilar contributions of different spectra and features into account and reduces the misclassification errors.

2. On the basis of the above model, an end-to-end MS image segmentation framework called DSSM was proposed to improve the segmentation ability of all surface elements in an experimental dataset and to improve the comprehensive segmentation accuracy.
3. A series of experiments were conducted to verify the effectiveness and superiority of our proposed method. The experimental results provide a new baseline for further research.

The rest of this article is organized as follows: Section 2 illustrates related work. In Section 3, the proposed method is described in detail. The effectiveness is analyzed and compared with various state-of-the-art (SOTA) methods using the Zurich [34] and Potsdam[35] datasets in Section 4. Finally, Section 5 presents the conclusion.

2. Related Work

In this section, we firstly review the development of image segmentation approaches in the CV field, including the design of the infrastructure, the principle of various SOTA methods, and fine-tuning in terms of specific scenarios. Then, the image segmentation methods in the RS field are reviewed and compared with the proposed algorithms.

Image segmentation, which aims to automatically assign a category to each pixel, is an active research topic in the field of CV and RS. In the CV field, traditional solutions utilize the knowledge of digital image processing tools, topology and mathematics to segment images, which makes it difficult to adequately leverage the color, spatiality, shape, texture, and boundary features [36,37]. To efficiently integrate these diverse features, certain methods based on CNNs provide a new strategy with which to analyze and interpret images [38]. However, the convolution operation, which is the most significant component of CNNs, is an irreversible feature extraction process, and, thus, it is hard to obtain the pixel-level classification results [36]. To solve this problem, with the encoder–decoder architecture, FCN imposes an upsampling and skip-connection module to retrieve the feature map in its original size [8]. Consequently, FCN has become the benchmark in the image segmentation task, and, thus, correlative algorithms are discussed in the following.

Compared with FCN, UNet has a more graceful architecture and a more complicated skip-connection module, which is able to efficiently fuse multi-scale features [9]. On the basis of UNet, UNet++ optimizes the skip-connection module with a pruned, deep supervised subnetwork [10]. PSPNet fuses the multi-scale features through the pyramid pooling module and combines the structural information using CRF [39], which improves the segmentation performance [12]. On the basis of the pyramid pooling module, combined with dilated convolution, Google proposes an ASPP module, which extracts and fuses multi-scale features more effectively [13,40–42].

In addition to the optimization of network structure, researchers also improved the convolution units. As regards enhancing the feature extraction capability, the attention mechanism was introduced to tackle the image processing task and to learn the significant correlation of multi-channel feature maps [43,44]. Inspired by this idea, we optimized the feature extraction module to reassign the weight of the feature maps for each spectrum. Furthermore, to decrease the convolution operation parameters, XCeption proposed depthwise separable convolution (DS-CNN), which reduces the dimension of 3D feature maps and accelerates the speed of training models [45]. On this basis, we conducted the spectrum separable module (SSM) to decouple the spectral information, which makes it possible to reassign the weight of spectral feature maps.

Most of the methods mentioned above obtained SOTA results on various image data sets. However, as mentioned before, considering the characteristics and differences between RSIs and CVIs, fine-tuning and embedding of RS expertise are still needed in RSI segmentation tasks [25,26,46]. Various researchers have made innovations and optimized the framework or structure, obtaining good results on certain RS datasets. More specifically, inspired by atrous convolution, an FCN-based method without downsampling is proposed to obtain and fuse features of different scales [47]. The holistically nested edge detection method, based on SegNet (HNED-SegNet), realizes RSI segmentation through an end-to-end edge detection scheme [48]. Thereafter, Pan proposed a dense pyramid network to enhance the low information flow between dimensional features by independently processing the digital surface model of the image using grouped convolution and connecting an effective data fusion method [49]. ScasNet, which adopted a coarse-to-fine refinement strategy, consists of a pre-trained encoder, various self-cascading convolution units, and a decoder component. It uses a pre-trained encoder to obtain more effective low-dimensional features, utilizes residual correction for multi-scale feature fusion, and obtained SOTA results on multiple challenging benchmark data sets [2]. Various researchers improved the loss function used in network training for the uneven distribution of ground objects of RSIs, such as focal loss [50] and balanced or weighted cross-entropy loss [21,51].

Furthermore, other researchers focus on fusing spectral and spatial features. More specifically, spectral features of each pixel are exploited by 1D CNN, which contains five layers, i.e., an input layer, a 1D convolutional layer, a max-pooling layer, a fully connected layer, and an output layer [28]. In [29,31,52,53], diversiform strategies are utilized to explore the effective representation of hyperspectral data in a lower dimension, including principal components analysis (PCA), 1D CNN, local discriminant embedding (LDE) and fractional order darwinian particle swarm optimization (FODPSO). A lightweight framework, with bag-of-features learning, that integrates 2D and 3D CNNs is proposed to learn the joint spatial-spectral features in [54]. In [55], a mini-graph convolutional network (miniGCN) is integrated with a CNN to extract fused spectral and spatial features in an efficient approach. In contrast, in our work, we focus on exploring the abstract correlation among spectra rather than finding a specific representation, and we model the spectral information in a global manner rather than in a local manner as a 3D CNN does.

In summary, beginning with the idea of decoupling spectral features, we built a spectral separable convolution module, and leveraged the attention mechanism to select effective features in a self-learning manner. Extensive experiments demonstrated that our module effectively improved the baseline segmentation accuracy, and the overall performance achieved SOTA results.

3. Methodology

In this section, we describe the proposed network structure in detail. First, we explain how the proposed spectral feature extraction strategy module explores the links between channels in MS images. Secondly, the detailed network structure of the DSSM is illustrated herein. Finally, the loss function formulation is elaborated.

3.1. Spectrum Separable Module

In this section, we describe the structure and principle of the SSM in detail, focusing on how SSM can improve the representation of spectral features.

In the sixth paragraph in Section 2, we discussed the shortcomings of CNNs in processing RSIs, because the inherent channel correlation inherent to CNNs ignores the different influence capabilities between spectra. Therefore, inspired by the depth-separable concept of DS-CNN, we proposed the spectrum separable module, as shown in Figure 1.

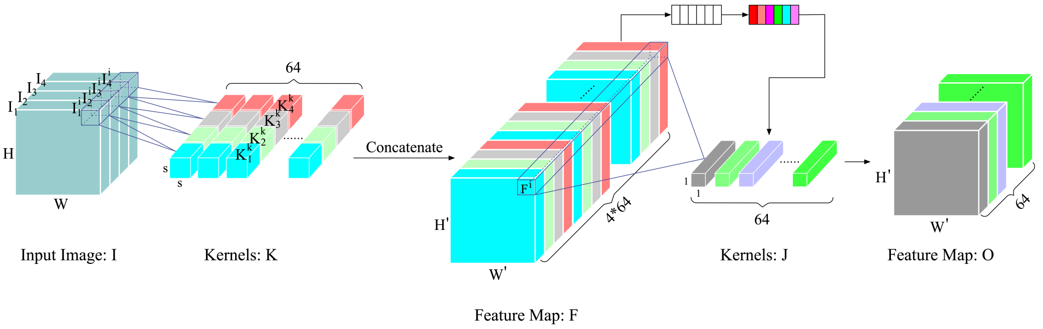


Figure 1. Overview of the proposed SSM. The network structure of the SSM is composed of spectrum-wise convolution, depth-wise attention and point-wise convolution.

The network structure of the SSM can be divided into three steps: spectrum-wise convolution, depth-wise attention and point-wise convolution.

3.1.1. Spectrum-Wise Convolution

In the spectrum-wise convolution step, $I \in \mathbb{R}^{H \times W \times 4}$ refers to the input image. I_j^i represents the i-th convolutional block on the j-th spectrum where $j \in \{1,2,3,4\}$ and $i \in \{1,2,\ldots,n\}$, and where n is the number of convolutional blocks. Then, for each convolutional block, we adopt four groups of kernels with 64 kernels in each group. $K_j^k \in \mathbb{R}^{s \times s}$ denotes the k-th kernel in the j-th group where $k \in \{1,2,\ldots,64\}$. Moreover, feature map F can be obtained by concatenating all the convolutional results and $\boldsymbol{F_{co}^x}$ is defined to conduct the concatenation operation upon dimension x. Therefore, the i-th value in $F \in \mathbb{R}^{H' \times W' \times 256}$ can be indicated as:

$$F_i = \boldsymbol{F}_{co}^k(\boldsymbol{F}_{co}^j(I_j^i \otimes K_j^k)) \tag{1}$$

Different from DS-CNNs, we removed the constraint on the number of output feature graphs. We isolated each spectrum, treated them as separate $H \times W \times 1$ images, and then carried out the convolution operation. Since the input has only one channel, there is no redundant channel correlation. Here, we set the number of convolution kernels per channel to 64 for comparison with the baseline. By separating the spectrum, we convert one input into four inputs. Moreover, for each input, the network can learn the feature type required by each input through backpropagation, which is a capability that CNNs and DS-CNNs do not have. This approach allows us to obtain the feature maps for each spectrum, and fuse them together using a concatenation operation to produce the final feature maps.

Since the concatenation operation is conducted for feature fusion, the spectrum-based feature maps are equally dealt with in the following processes, which can cause a loss of accuracy. For example, the nearshore water may have a different color from the ordinary water in the RGB image due to the shallow depth. However, the corresponding pixel values for the same nearshore water in NIR may not be very different from ordinary water. To achieve better segmentation accuracy, a trade-off should be made to capture both the texture features in RGB and the near-infrared features in NIR for the water area. We expected the proposed network to learn the features based on certain points of focus, and thus an attention mechanism was utilized in the proposed model.

3.1.2. Depth-Wise Attention

In the first step, we obtained feature map F, which contains a number of channels representing multiple types of features. By applying the attention mechanism on F, we enhanced the important features and weakened the unimportant ones. In brief, we gave each channel the ability to self-learn its weights.

In order to obtain the global feature of a channel, the global average pooling method (GAP) was adopted to integrate the global information and obtain the compressed feature graph $G \in 1 \times 1 \times 256$. This process is represented as $\boldsymbol{F_{sq}}$, and then G can be calculated by:

$$G_j = \boldsymbol{F_{sq}}(F) = \frac{1}{W' \times H'} \sum_{x=1}^{W'} \sum_{y=1}^{H'} F_j(x,y) \tag{2}$$

Since we want to obtain the weights of a complete channel, GAP is used to integrate the channel's internal information. Note that GAP is a relatively simple, but effective, pooling method to integrate global information. In addition to this, for example, maximum pooling and random pooling both cause more or less pixel point loss inside the channel, so we chose GAP.

Thereafter, two FC layers were adopted to integrate the multi-channel information, followed by the sigmoid function as the active function with which to generate the weight map S that can be applied to the original feature map. $\boldsymbol{F_{ex}}$ is used to denote this operation, and S can be calculated by:

$$S = \boldsymbol{F_{ex}}(G) = \boldsymbol{\sigma}(\boldsymbol{FC_1}(\boldsymbol{FC_2}(G))) \tag{3}$$

It is necessary to use the FC to integrate the channel information here, because each channel value of G is the result of a GAP based entirely on a single channel. If it is directly applied to F, the process of obtaining the following T is completely without backpropagation and training, which makes it impossible to learn the law of channel enhancement and attenuation through all channel information.

Finally, the obtained weight map was multiplied directly by the channel corresponding to the original feature map, which is represented by $\boldsymbol{F_{scale}}$, with which we obtain the weighted feature map T. At this point, the feature map has acquired the ability to express the strength and weaknesses of multiple features. T can be expressed as:

$$T = \boldsymbol{F_{scale}}(S, F) = \boldsymbol{F_{co}^{j}}(S_j \times F_j) \tag{4}$$

3.1.3. Point-Wise Convolution

We applied the attention mechanism to a series of features corresponding to each spectrum, and the convolution kernel of 1×1 to integrate the channel correlations. On the one hand, this enabled it to learn nonlinear relationships between channels, and on the other hand, it allows it to configure subsequent input dimensions by specifying the number of convolution kernels.

In Equation (5), $J \in \mathbb{R}^{1\times1\times256}$ refers to the kernels we adopted here, and the final feature map $O \in \mathbb{R}^{H' \times W' \times 64}$ can be obtained by:

$$O = J \otimes T \tag{5}$$

3.1.4. Computational Cost

To compare the computational cost between standard convolutional layers and DSSM when the input and output has the same shape, we used the following assumptions:

1. Both the standard convolutional layer and DSSM take as input a $D_i \times D_i \times M$ image I as input and produce a $D_i \times D_i \times N$ feature map F, where D_i is the spatial width and height of a square input image, M is the number of spectra in the input image, and N is the number of channels in the output feature map.
2. The padding type is the same, which means the spatial size of output feature map should be the same as that of the input image.
3. During the execution process of the model, the GPU consumption is mainly determined by the model size under the same conditions. In turn, the model size is proportional to the number of parameters. Therefore, in this section, the number of parameters of the module is used to evaluate the computational cost.

Thereafter, in the standard convolutional layer, N convolution kernels of size J are needed to obtain the feature map, where D_k is the spatial length of the convolution kernel, and M is the number of the input image spectrum, and N is the number of output channels as defined previously. Thus, the standard convolution layer has the the computational cost of :

$$Cost(CNN) = D_i \times D_i \times M \times N \times D_k \times D_k \tag{6}$$

where the cost is greatly influenced by the shape of the input and output. As we all know, the coupling between each spectrum leads to the inevitable equality in the standard convolution.

To solve this problem, SSM was proposed to break the coupling and adopt a 1×1 convolution to integrate the mixed feature map. Here, we need M kernel groups in SSM, then in each group, N' convolution kernels K of size $D_k \times D_k \times 1$ are designed to obtain specific feature map group for each spectrum. Therefore, the computational cost of the first SSM step is:

$$Cost_1(SSM) = D_i \times D_i \times M \times N^{'} \times D_k \times D_k \tag{7}$$

These groups are concatenated together, rather than added, to obtain a mixed feature map. Then, N kernels of size $D_k \times D_k \times 1$ are adopted to apply a linear combination of the mixed feature map to obtain the final feature map. Considering the cost of the above step together, SSM has the total cost of:

$$Cost(SSM) = Cost_1(SSM) + M \times N \times N^{'} \times D_k \times D_k \tag{8}$$

Therefore, we calculate the following ratio by comparing SSM with the standard convolution:

$$\frac{Cost(SSM)}{Cost(CNN)} = \frac{N^{'}}{N} + \frac{N^{'}}{D_i^2} \tag{9}$$

It is noteworthy that when we set $N' = 1$ then the cost is the same as that in DS-CNN. Moreover, when N' is set to be the same as N, we maintain the dimension of the feature map. Now, we solely need to focus on the component of $\frac{N}{D_i^2}$, which is equal to $\frac{1}{64}$ in our implementation, where N and D_i are both 64. Hence, the computational cost was only 1.56% more than standard convolutions, but a much improved spectral feature extraction capability was successfully achieved.

3.2. Network Structure

In order to solve the semantic segmentation problems of MS images, we developed an end-to-end network framework based on Deeplab. The structure of the framework is shown in Figure 2.

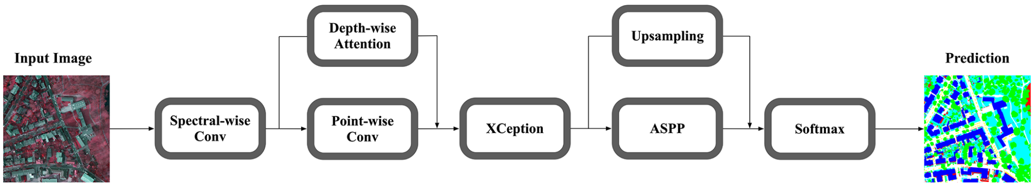


Figure 2. Network Structure.

The DSSM is composed of the following steps:

Step 1: Derive a feature map with spectral information

The MS image is used as the model input and can be processed using the SSM, which results in a feature map with a strong expression ability for spectral information.

Step 2: Feature extraction using XCeption

The feature map is transferred into XCeption for low-dimensional feature extraction.

Step 3: Multi-scale feature extraction using ASPP

The extracted feature map is input into the ASPP module to extract the multi-scale feature, and the result is concatenated to obtain the high-dimensional feature.

Step 4: Upsampling and concatenating

The high-dimensional feature is firstly upsampled and then concatenated with the low-dimensional feature. Finally, a simple convolution and upsampling operation is adopted to return the size of the feature to the input size.

3.3. Loss Function

In our proposed framework, the overall loss function can be indicated as:

$$\mathcal{L} = \mathcal{L}_{bce} + \mathcal{L}_{dice} \tag{10}$$

3.3.1. Balanced Sparse Softmax Cross Entropy

In image segmentation, there is usually a large deviation in the proportion of pixels of different categories. To cope with the biased sampling class, we defined a balanced sparse softmax cross-entropy-based tradeoff strategy.

Assuming that there are S classes of surface object in our dataset, we adopt $\hat{\mathcal{Y}} \in \mathbb{R}^{H \times W \times S}$ to denote the true label, and $\hat{\mathcal{Y}} = \{\hat{y}_i, i = 1, 2, \ldots, S\}$ which means i is the object category. Moreover, each $\hat{y}_i \in \mathbb{R}^{H \times W}$ is a binary map that only contains 1 and 0, which represents if the current pixel belongs to the i-th category or not. Similarly, $\mathcal{Y} \in \mathbb{R}^{H \times W \times S}$ is adopted to refer to the prediction map, where $\mathcal{Y} = \{y_i, i = 1, 2, \ldots, S\}$. The difference is that the value of a specific pixel in y_i is the probability of the pixel belonging to i-th category. Following a simple balanced strategy, and balanced sparse softmax cross entropy can be defined as:

$$\mathcal{L}_{bce} = -\sum_{i=1}^{S} \frac{\|\hat{y}_i log(y_i)\|_1}{\beta_i} \tag{11}$$

where $\|A\|_1$ denotes the $L1$ norm of matrix A and β_i can be defined as:

$$\beta_i = \frac{\|y_i = i\|_1}{\sum_{j=1}^{S} \|y_j = j\|_1} \tag{12}$$

3.3.2. Dice Coefficient

The dice coefficient is also a commonly used loss function applied in image segmentation tasks [21], which can be defined as:

$$\mathcal{L}_{dice} = 1 - \frac{2\|\mathcal{Y}\hat{\mathcal{Y}}\|_1}{\|\mathcal{Y}\|_1 + \|\hat{\mathcal{Y}}\|_1} \tag{13}$$

4. Experiment

We designed and conducted three groups of experiments with the following three objectives:

1. In order to verify the effectiveness of the proposed DSSM in an MS information fusion, firstly, we compared the segmentation performance of the baseline framework based on RGB, NIR, and RGB-NIR inputs to verify whether RGB-NIR contains more valuable information and improves the performance. Secondly, we compared the proposed DSSM with the baseline framework based on the same set of RGB-NIR inputs to evaluate its effectiveness.
2. In order to further verify the validity of the proposed SSM, features from different levels were visualized and compared to demonstrate that more abstract and effective features can be extracted from the SSM.
3. In order to verify the overall performance of the proposed framework, comparison experiments were carried out to compare with other SOTA methods.

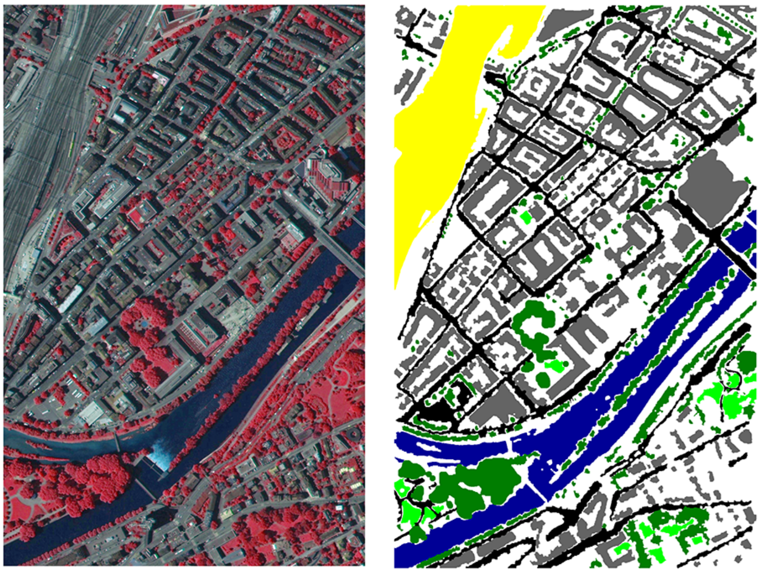
4.1. Dataset

4.1.1. Data Introduction

In image segmentation tasks, supervised learn-based frameworks generally require large amounts of data with high quality labels, which are often difficult to obtain. Fortunately, Michele released his self-labeled Zurich dataset in 2015, which includes 20 high-resolution photos of Zurich, Switzerland with a 0.62-m GSD obtained from a QuickBird satellite in August 200. Surface objects in this dataset are classified into eight different categories: road, building, tree, grass, bare soil, water, railway, and swimming pool, and each object is annotated by a specific color. In order to reflect the real-world distribution, the number of pixels in various samples is unbalanced, as shown in Figure 3. In addition, since it was marked manually, these labels are not completely consistent with the real-world objects, and there are some errors or missing labels. The data set contains the original image and the corresponding labels. The original image is an MS image containing four channels of NIR and RGB, with a 16-bit depth. The 16-bit depth image contains a lot of colors and subtle differences among those colors that humans cannot recognize, but that neural networks can easily pick up.

ID	Color	Label	Proportion	ID	Color	Label	Proportion
1	⬛	Roads	14.61%	2	◼	Buildings	20.64%
3	🟩	Trees	9.72%	4	🟩	Grass	7.16%
5	🟫	Bare Soil	0.90%	6	🟦	Water	4.87%
7	🟨	Rails	1.31%	8	🟪	Pools	0.13%

(a)



(b) (c)

Figure 3. Example from the Zurich dataset. Class legend with colors and related labels (**a**); note that a white background is not considered to be a separate class. Original image (RGB) (**b**) and its ground truth (**c**).

Another dataset we adopted is the Potsdam dataset provided by ISPRS as shown in Figure 4, which contain 38 images of Potsdam, Germany.

4.1.2. Sampling Strategy

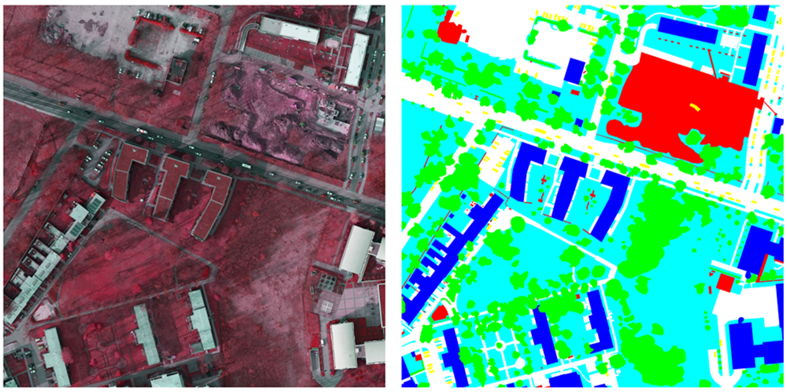
For the Zurich dataset, as the resolution of the source photos ranges from 650 × 650 to 1700 × 1700, random sampling was conducted to take into account the impact of resource constraints, training efficiency, training effects, and other factors. The 20 photos were randomly sampled to produce 20,000 slices, 256 × 256 in size, preserving the maximum amount of detail and edge information in the original photos. Then, 80% of these participated in training and validation, and the remaining 20% served as test sets.

4.1.3. Data Augmentation

In order to improve the expression ability of the data and the generalization ability of the model, we applied a variety of data enhancement methods to the data, including rotation, flipping, slicing, Gaussian filtering, bilateral filtering, gamma transformation, and so on. In fact, the random slice method mentioned above, which increases the number of samples, was also used for data augmentation.

ID	Color	Label	Proportion	ID	Color	Label	Proportion
1	🟫	Bare Soil	44.48%	2	🟦	Buildings	15.98%
3	🟩	Trees	5.33%	4	🟨	Cars	0.99%
5	■	Grass	10.64%	6	□	22.59	4.87%

(a)



(b) (c)

Figure 4. Example from the Potsdam dataset. Class legend with colors and related labels (**a**). Original image (RGB) (**b**) and its ground truth (**c**).

4.2. Evaluation Methods

In order to evaluate the effectiveness of the proposed framework, we adopted intersection over union (IoU) as an evaluation indicator for a single category of ground objects. IoU defines the similarity between the predicted area and the ground reality area of the objects in this set of images. The calculation formula is as follows:

$$IoU_i = \frac{TP_i}{FP_i + TP_i + FN_i} \tag{14}$$

where TP, FP, and FN represent the counts of true positive, false positive, and false negative, respectively, and i refers to the category number. The result of the count is generally obtained by the confusion matrix between the predicted value and the ground reality.

Moreover, FW IoU was used to evaluate the overall segmentation performance, which can be calculated as:

$$FWIoU = \sum_{i=1}^{N} p_i \times IoU_i \tag{15}$$

where p_i refers to the percentage of pixels of the i-th ground object. When calculating the average IoU, $FWIoU$ also took into account the occurrence probability of ground objects, thus making the evaluation of the segmentation result more accurate.

4.3. Experimental Environments

The proposed architecture is implemented using the Tensorflow library. The hardware device is a Ubuntu server equipped with four GeForce RTX 2080 Ti GPUs (each has 12 GB of memory), one Intel i9-9960X CPU and 64 GB of RAM. The detailed hardware configuration and software requirements are shown in Table 1.

Table 1. Hardware configuration and software requirements in our implementation.

Item		**Version**
Hardware	GPU	Four GeForce RTX 2080 Ti, 12 GB
	CPU	Intel i9-9960X CPU
	RAM	64 GB
	Operating System	Ubuntu 18.04
Software	CUDA	10.2
	Python	3.6
	Tensorflow	1.14.0

4.4. Comparison of Various Processing Strategies

In the introduction to SSM, we stated that the SSM actually provides a strategy for fusing different spectra from the input image. The purpose of this section of the experiment was to verify the effectiveness of the fusion strategy in the SSM. This group of experiments were conducted on the Zurich and Potsdam datasets.

Before we proposed the SSM, we put forward a hypothesis that different spectra would show different values on different ground objects. On the basis of this assumption, we treated each spectrum of the input image as an independent single spectral image. We convolved them separately, and then we established the nonlinear relationship among the feature maps generated by all the spectra through 1×1 convolution. This enabled the network to learn which features needed to be learned for each spectrum independently. In order to assess whether more valuable information is contained in RGB-NIR and the effectiveness of the SSM, we designed several processing strategies for comparison, which are as follows:

1. We fed three spectra of red, green, and blue into the baseline (baseline with RGB).
2. We fed the NIR into the baseline (baseline with NIR).
3. We fed four spectra into baseline (baseline with RGB-NIR).
4. We fed all four spectra into the proposed framework (DSSM with RGB-NIR).

In this group of experiments, the Deeplabv3+ network, which is fine-tuned in terms of cost function and the number of convolutional layers, was adopted as the baseline with which to compare the DSSM.

Figure 5 shows the segmentation results of the baseline with RGB, the baseline with NIR, the baseline with RGB-NIR, and the DSSM with RGB-NIR. Figure 5a,b are the original images and their ground truths. Figure 5c–e refer to the baseline with RGB, the baseline with NIR and the baseline with RGB-NIR, respectively. Figure 5f represents the DSSM with RGB-NIR inputs.

As a result of the large receptive field of the original image, the contrast effect of the whole image is not obvious. Therefore, various small image blocks with a size of 200×200 were cut from the original image. These image blocks were selected from certain parts with extreme differences in segmentation results in typical ground objects.

Firstly, we compared the segmentation differences of different inputs on small isolated trees, as shown in the yellow box in the first line of Figure 5, where three small isolated trees are shown. The baseline with RGB and the baseline with NIR strategies lost the ability to identify small trees, and they roughly identified the area as the background. The RGB-NIR strategy identified a tree, but its outline is quite different from the ground truth. The DSSM with the RGB-NIR strategy identified three complete trees. Although the boundary between the first tree and the second tree is connected, the outline of the two trees can be seen, and the shape of the identified trees is closer to the ground truth than that of the other three strategies.

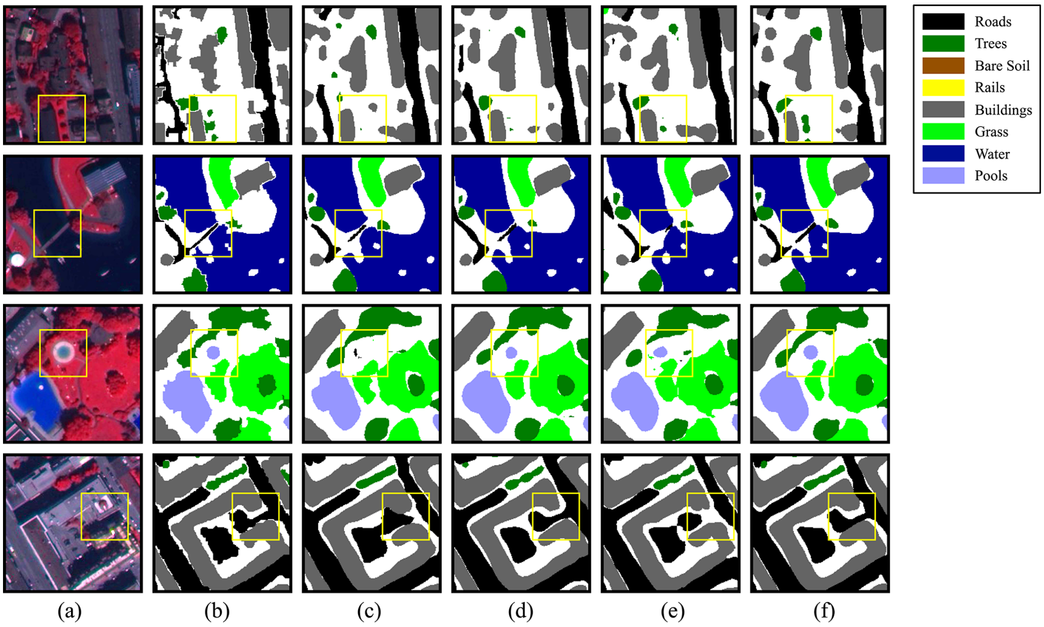


Figure 5. Comparison of various processing strategies in the Zurich dataset. (**a**) Original image. (**b**) Ground truth. (**c**) Baseline with RGB. (**d**) Baseline with NIR. (**e**) Baseline with RGB-NIR. (**f**) DSSM with RGB-NIR.

In the yellow box in the second line of Figure 5, we selected small roads, which are bridges marked as roads. Curiously, the two strategies, RGB and NIR, were able to coarsely divide the bridge. However, when they were mixed, i.e., when the RGB-NIR strategy was used, the recognition ability actually reduced. Their simultaneous use may inhibit the other's recognition ability. Of course, the RGB-NIR with DSSM strategy also correctly divided the bridge.

In the yellow box in the third line of Figure 5, there is a small and shallow swimming pool. The RGB strategy completely lost the ability to recognize this object, and even identified the shore of the swimming pool as a road. The baseline with NIR, the baseline with RGB-NIR and the DSSM with RGB-NIR were all able to identify the swimming pool, but the predicted contour of the DSSM RGB-NIR was closest to the ground truth.

In the yellow box in the fourth row of Figure 5, there is a road with a shaded area. Both the baseline with NIR and the baseline with RGB-NIR accurately identified the road with a shaded area, while the baseline with RGB and the baseline with RGB-NIR consider that to be the background.

In a previous analysis, we compared four classes in which obvious differences can be observed with the human eye. Since each image block was only 200 × 200 in size, the visual effects in most areas using different processing strategies were similar. In Table 2, IoU and FW IoU of different strategies on each ground object are quantitatively given. It can be clearly seen from Table 2 that the fusion strategy of the DSSM proposed by us greatly improved the segmentation effect of each type of ground object. Firstly, the baseline with RGB-NIR performed better than the baseline with RGB and the baseline with NIR by 1.18% in terms of FW IoU using the Zurich dataset. However, as compared with the first three strategies, the DSSM with RGB-NIR consistently outperformed them, registering increased FW IoUs of approximately 3.47%. Furthermore, the confusion matrix for the prediction is shown in Figure 6. The wrong classification frequently occurred for the background; this does not represent an inter-object mistake but a difficulty in distinguishing between background and non-background objects.

Table 2. The quantitative results of different processing strategies using the Zurich dataset.

Strategy	Background	Road	Bare Soil	Tree	Water	Buildings	Grass	Rails	Pool	FW IoU
Baseline with RGB	75.94	79.87	94.99	71.86	91.65	84.36	84.22	89.34	90.96	84.80
Baseline with NIR	75.60	79.70	95.21	72.06	91.11	83.92	84.90	89.85	90.88	84.80
Baseline with RGB-NIR	78.09	80.80	95.46	75.65	91.05	85.33	86.24	89.21	91.98	85.98
DSSM with RGB-NIR	**83.18**	**85.76**	**96.24**	**81.27**	**94.05**	**88.71**	**89.95**	**91.61**	**94.31**	**89.45**

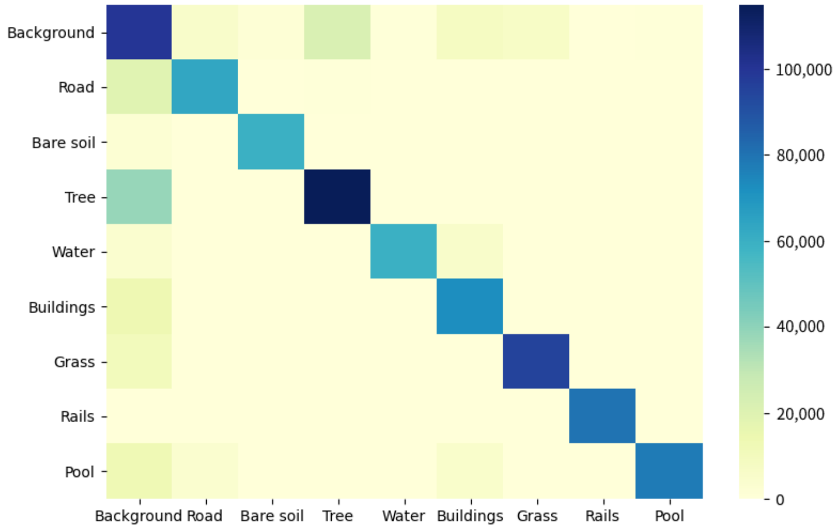


Figure 6. Confusion matrix for the prediction using the Zurich dataset.

To further verify the efficiency of the DSSM, we also conducted other experiments using the Potsdam dataset. Figure 7 and Table 3 show the effects of the four strategies using the Potsdam dataset. The DSSM with RGB-NIR continued to demonstrate a better segmentation capability than the other three strategies, which increased by 1.92% in terms of FW IoU.

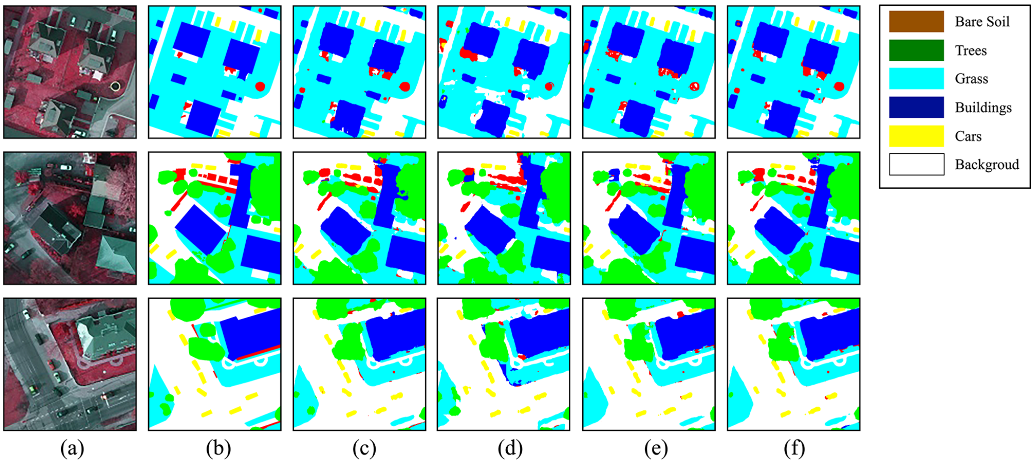


Figure 7. Comparison of various processing strategies in the Potsdam dataset. (**a**) Original image. (**b**) Ground Truth. (**c**) Baseline with RGB. (**d**) Baseline with NIR. (**e**) Baseline with RGB-NIR. (**f**) DSSM with RGB-NIR.

Table 3. The quantitative results of different processing strategies using the Potsdam dataset.

Strategy	Bare Soil	Building	Tree	Car	Grass	Road	FW IoU
Baseline with RGB	89.92	64.58	73.01	80.63	78.80	83.21	78.36
Baseline with NIR	85.91	60.53	68.29	75.65	77.72	77.19	74.22
Baseline with RGB-NIR	90.13	67.72	73.09	81.26	78.99	83.90	79.18
DSSM with RGB-NIR	**90.24**	**71.29**	**75.63**	**83.66**	**81.49**	**84.31**	**81.10**

In addition to comparing the segmentation accuracy of different strategies, we also compared the consumption cost of each strategy. As can be seen from Table 4, for the first three schemes, we maintained the same size and number of convolution kernels in the baseline. At this point, the number of spectra of the input image only affects the thickness of the convolution kernel, and we already know from the previous analysis that the thickness of the convolution kernel in the CNN does not affect the parameters of the model; thus, their consumption cost remained the same. As compared with the aforementioned three strategies, the consumption cost of the DSSM with RGB-NIR increased by about 20%. Considering that this strategy provides a great improvement in segmentation accuracy, we believe that the increase in consumption is acceptable.

Table 4. Consumption cost of different processing strategies using the Zurich dataset.

Strategy	Consumption Cost
Baseline with RGB	8785 MiB
Baseline with NIR	8785 MiB
Baseline with RGB-NIR	8785 MiB
DSSM with RGB-NIR	**10,835 MiB**

In conclusion, the fusion strategy in our proposed SSM is more effective than that in those strategies that use the pure CNN to obtain the image feature, and it provides an acceptable consumption cost. Another interesting fact is that using the RGB-NIR strategy is sometimes less effective than directly using the NIR strategy, which can be seen in both the qualitative and quantitative results. In the next section, we discuss why the SSM improves the segmentation accuracy in more detail.

4.5. Comparison of Features Extracted from Different Levels

In this group of experiments, we assessed the validity of the SSM from another perspective. Different levels of features extracted from the baseline and DSSM were visualized and contrasted. This group of experiments was conducted using the Zurich dataset.

Figure 8 shows the low-level features extracted from the baseline and DSSM, respectively. Figure 8a,b are the original images and their ground truth. Figure 8c shows the low-level features extracted in the baseline where spectra are convolved in a weight-sharing manner. Figure 8d–g represent the low-level features explored in DSSM in a spectrum-separable way. Concentrating on the yellow boxes in Figure 8d–g, different features are revealed in different spectra. In other words, diversiform characteristics under different spectra can be captured by the SSM. However, as shown in Figure 8c, differentiation among objects becomes less distinct due to the weight-sharing strategy.

In Figure 9, we visualize different level of features. Figure 9a,b are the original images and their ground truth. Figure 9c,d show the low-level features. They are extracted by one convolutional layer in the baseline and the SSM, respectively, in the proposed method. Obviously, as shown in the yellow boxes, fused low-level features in the DSSM provide a sharper and more legible pattern and information. Figure 9e,f illustrate the high-level features obtained from the ASPP in the baseline and DSSM, respectively. The higher the level is, the more abstract and unrecognizable the pattern will be. As shown in Figure 9e,f, although the features are abstract and hard to recognize, high-level features in the DSSM contain more complicated manifestations.

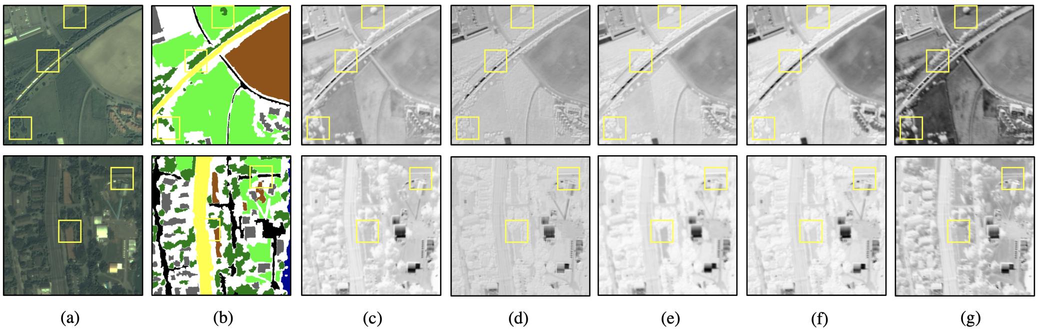


Figure 8. Comparison of spectral features extracted from the baseline and proposed method using the Zurich dataset. (**a**) Original image. (**b**) Ground truth. (**c**) Low-level features in the baseline. (**d**) Low-level features of the red band in DSSM. (**e**) Low-level features of the green band in DSSM. (**f**) Low-level features of the blue band in DSSM. (**g**) Low-level features of the NIR band in DSSM.

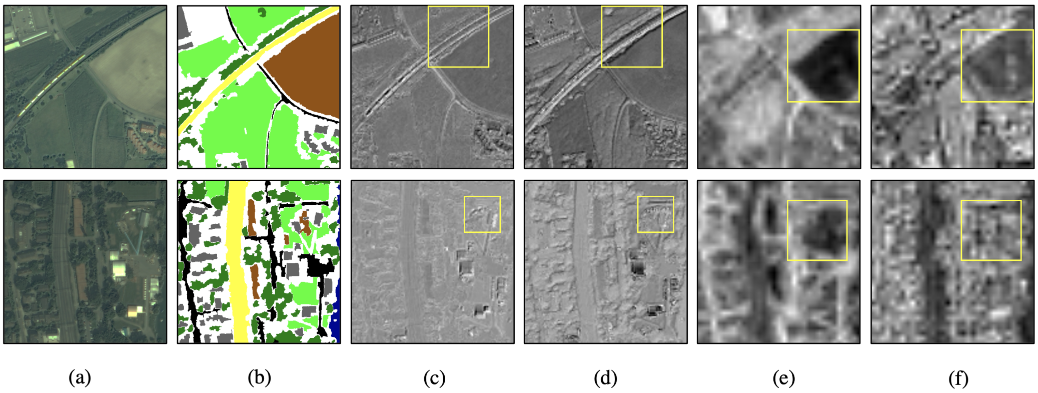


Figure 9. Comparison of spectral features extracted from the baseline and proposed method using the Zurich dataset. (**a**) Original image. (**b**) Ground truth. (**c**) Low-level features in the baseline. (**d**) Fused low-level features in DSSM. (**e**) High-level features from ASPP in the baseline. (**f**) High-level features from ASPP in DSSM.

4.6. Comparison of Other Popular Segmentation Methods

In order to verify the overall performance of the DSSM, we compared the DSSM with UNet++, DeeplabV3+, HNED-SegNet, ScasNet, and miniGCN using the Zurich and Potsdam datasets. As discussed in Section 2, UNet++ and DeeplabV3+ are CV-field methods and HNED-SegNet, ScasNet, and miniGCN are RS-field methods. In order to obtain more effective results, only the network structure was customized in the experiment using each method.

As shown in Table 5, the DSSM outperformed the other methods in most surface elements, and provided a satisfying improvement of 5.78% for trees in terms of IoU. Vegetation areas, such as trees, have a similar pattern as other surface elements in a single spectrum. Through establishing the correlations among spectra, we improved the segmentation efficiency of these areas. However, as a result of the extremely irregular shapes and the inaccurate labels of trees, the IoU remained lower than the FW IoU. Moreover, the segmentation accuracy improved by 2.99% for water and pools for the same reason. Other surface elements improved by 1.53% overall when background and rails were excluded.

Furthermore, the IoU of the DSSM for rails was 1% lower than that of UNet++, because the multi-level skip-connection structure of UNet++ is more effective on these narrow and slender surface elements. Overall, the FW IoU is increased by 2.19%, which demonstrates that the segmentation accuracy of the DSSM in terms of segmenting MS RSIs is generally better than that of other SOTA methods.

Table 5. The quantitative results of popular methods using the Zurich dataset.

Strategy	Background	Road	Bare Soil	Tree	Water	Buildings	Grass	Rails	Pool	FW IoU
UNet++	**85.97**	82.84	95.71	74.47	90.99	86.03	84.25	**92.60**	91.53	87.15
DeeplabV3+	83.01	83.09	95.96	75.49	91.04	86.72	87.43	91.23	91.33	87.26
ScasNet	83.04	81.21	95.53	72.73	90.31	85.12	83.45	88.89	90.18	85.61
HNED-SegNet	85.43	82.81	95.65	74.55	90.98	86.04	84.55	90.60	91.79	86.93
miniGCN	82.11	80.21	93.53	71.76	90.33	85.66	84.00	87.20	91.32	85.12
Proposed method	83.18	**85.76**	**96.24**	**81.27**	**94.05**	**88.71**	**89.95**	91.61	**94.31**	**89.45**

The quantitative results using the Potsdam dataset are shown in Table 6, which are similar to those using the Zurich dataset. It is worth noting that our method still has a lot of room for improvement in the results of the buildings. Overall, the FW IoU is increased by 0.19% compared with the best method, which is HNED-SegNet.

Table 6. The quantitative results of popular methods using the Potsdam dataset.

Strategy	Bare Soil	Building	Tree	Car	Grass	Road	FW IoU
UNet++	79	84.82	74.91	**87.47**	78.87	80.04	80.85
DeeplabV3+	90.13	67.72	73.09	81.26	78.99	83.90	79.18
ScasNet	90.04	**85.76**	72.49	80.28	72.49	83.58	80.77
HNED-SegNet	81.07	83.41	73.85	85.72	77.89	83.65	80.93
miniGCN	82.01	83.27	65.41	80.58	73.62	80.03	77.49
Proposed method	**90.24**	71.29	**75.63**	83.66	**81.49**	**84.31**	**81.10**

Table 7. The comparison of prediction time for a batch of popular methods.

Strategy	Prediction Time (ms/batch)
UNet++	**9.31**
DeeplabV3+	10.63
ScasNet	11.63
HNED-SegNet	10.15
miniGCN	12.38
Proposed method	10.85

We also compared the execution time with other methods as shown in Table 7. We define the unit of execution time as the time it takes for the model to predict each batch. Model and data loading times have been excluded from the results. In the actual prediction, each batch contains 64 images of 256 × 256 size. UNet++ takes the shortest time, thanks to its streamlined network structure. Our method takes about 10.85 ms, which is 1.54 ms slower than UNet++. The extra time consumption of the proposed method is acceptable when taking its performance improvement into account.

5. Conclusions

In this paper, we propose a deep-learning based, end-to-end network structure DSSM for the semantic segmentation of MS optical RSIs. The framework is mainly composed of an SSM module and a deep neural network.

The SSM is based on a DS-CNN and optimizes the spectral feature extraction strategy. First, features are independently extracted through spectrum-wise convolution, and then the importance of each feature is studied using a depth-wise attention module. Finally, a nonlinear relationship between features is established through point-wise convolution to generate the final feature map. These extracted features not only contain spatial information, but also have a stronger ability to express spectral correlation. We applied the SSM as a prefeature extraction module in deep neural network. The experimental results show that the DSSM has a better segmentation capability than other SOTA methods, and provides an improvement of 2.19% in terms of FW IoU. Moreover, our proposed SSM can be easily grafted onto other deep-learning-based networks.

In future work, we will focus on transforming the processing techniques proposed in the SSM into hyper-spectral RSIs, reducing the complexity of the network and consumption cost, and applying the strategy to other RS tasks, such as change detection.

Author Contributions: Conceptualization, H.Z. and R.T.; methodology, Q.L.; software, L.H.; validation, B.D.; formal analysis, H.F.; investigation, Z.W.; resources, B.D.; data curation, L.H.; writing—original draft preparation, R.T.; writing—review and editing, B.D. and H.Z.; visualization, L.H.; supervision, Q.L.; project administration, H.Z.; funding acquisition, H.F., H.Z. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been supported by the National Key R&D Program of China (No. 2018YFB0505001), the National Natural Science Foundation of China (No. 61702374), the Natural Science Foundation of Shanghai (No.21ZR1465100).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. 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]
2. Liu, Y.; Fan, B.; Wang, L.; Bai, J.; Xiang, S.; Pan, C. Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 78–95. [CrossRef]
3. Matikainen, L.; Karila, K. Segment-based land cover mapping of a suburban area—Comparison of high-resolution remotely sensed datasets using classification trees and test field points. *Remote Sens.* **2011**, *3*, 1777–1804. [CrossRef]
4. Zhang, Q.; Seto, K.C. Mapping urbanization dynamics at regional and global scales using multi-temporal DMSP/OLS nighttime light data. *Remote Sens. Environ.* **2011**, *115*, 2320–2329. [CrossRef]
5. Lu, X.; Yuan, Y.; Zheng, X. Joint dictionary learning for multispectral change detection. *IEEE Trans. Cybern.* **2016**, *47*, 884–897. [CrossRef] [PubMed]
6. Goldblatt, R.; Stuhlmacher, M.F.; Tellman, B.; Clinton, N.; Hanson, G.; Georgescu, M.; Wang, C.; Serrano-Candela, F.; Khandelwal, A.K.; Cheng, W.H.; et al. Using Landsat and nighttime lights for supervised pixel-based image classification of urban land cover. *Remote Sens. Environ.* **2018**, *205*, 253–275. [CrossRef]
7. O'Shea, K.; Nash, R. An introduction to convolutional neural networks. *arXiv* **2015**, arXiv:1511.08458.
8. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 640–651.
9. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland, 2015; pp. 234–241.
10. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Cham, Switzerland, 2018; pp. 3–11.
11. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
12. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
13. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.

14. Sun, X.; Shi, A.; Huang, H.; Mayer, H. BAS Net: Boundary-Aware Semi-Supervised Semantic Segmentation Network for Very High Resolution Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 5398–5413. [CrossRef]
15. Li, H.; Qiu, K.; Chen, L.; Mei, X.; Hong, L.; Tao, C. SCAttNet: Semantic segmentation network with spatial and channel attention mechanism for high-resolution remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 905–909. [CrossRef]
16. Zhang, J.; Lin, S.; Ding, L.; Bruzzone, L. Multi-scale context aggregation for semantic segmentation of remote sensing images. *Remote Sens.* **2020**, *12*, 701. [CrossRef]
17. Chen, G.; Li, C.; Wei, W.; Jing, W.; Woźniak, M.; Blažauskas, T.; Damaševičius, R. Fully convolutional neural network with augmented atrous spatial pyramid pool and fully connected fusion path for high resolution remote sensing image segmentation. *Appl. Sci.* **2019**, *9*, 1816. [CrossRef]
18. Radoux, J.; Bourdouxhe, A.; Coos, W.; Dufrêne, M.; Defourny, P. Improving ecotope segmentation by combining topographic and spectral data. *Remote Sens.* **2019**, *11*, 354. [CrossRef]
19. Kemker, R.; Salvaggio, C.; Kanan, C. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS J. Photogramm. Remote Sens.* **2018**, *145*, 60–77. [CrossRef]
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
21. Peng, D.; Zhang, Y.; Guan, H. End-to-end change detection for high resolution satellite images using improved unet++. *Remote Sens.* **2019**, *11*, 1382. [CrossRef]
22. Goward, S.N.; Markham, B.; Dye, D.G.; Dulaney, W.; Yang, J. Normalized difference vegetation index measurements from the Advanced Very High Resolution Radiometer. *Remote Sens. Environ.* **1991**, *35*, 257–277. [CrossRef]
23. Gao, B.C. NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sens. Environ.* **1996**, *58*, 257–266. [CrossRef]
24. Liu, X.; Wang, D. A spectral histogram model for texton modeling and texture discrimination. *Vis. Res.* **2002**, *42*, 2617–2634. [CrossRef]
25. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [CrossRef]
26. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
27. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 166–177. [CrossRef]
28. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
29. Makantasis, K.; Karantzalos, K.; Doulamis, A.; Doulamis, N. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 4959–4962.
30. Santara, A.; Mani, K.; Hatwar, P.; Singh, A.; Garg, A.; Padia, K.; Mitra, P. BASS net: Band-adaptive spectral-spatial feature learning neural network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5293–5301. [CrossRef]
31. Ghamisi, P.; Chen, Y.; Zhu, X.X. A self-improving convolution neural network for the classification of hyperspectral data. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1537–1541. [CrossRef]
32. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
33. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]
34. Volpi, M.; Ferrari, V. Semantic segmentation of urban scenes by learning local class interactions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
35. Rottensteiner, F.; Sohn, G.; Jung, J.; Gerke, M.; Baillard, C.; Benitez, S.; Breitkopf, U. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *I-3*, 293–298.
36. Minaee, S.; Boykov, Y.Y.; Porikli, F.; Plaza, A.J.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [CrossRef]
37. Sultana, F.; Sufian, A.; Dutta, P. Evolution of image segmentation using deep convolutional neural network: A survey. *Knowl.-Based Syst.* **2020**, *201*, 106062. [CrossRef]
38. Wu, J. *Introduction to Convolutional Neural Networks*; National Key Lab for Novel Software Technology, Nanjing University: Nanjing, China, 2017; Volume 5, p. 495.
39. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H. Conditional random fields as recurrent neural networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1529–1537.
40. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
41. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]

42. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
43. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
44. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
45. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
46. Cheng, G.; Yang, C.; Yao, X.; Guo, L.; Han, J. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2811–2821. [CrossRef]
47. Sherrah, J. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *arXiv* **2016**, arXiv:1606.02585.
48. Marmanis, D.; Schindler, K.; Wegner, J.D.; Galliani, S.; Datcu, M.; Stilla, U. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS J. Photogramm. Remote Sens.* **2018**, *135*, 158–172. [CrossRef]
49. Pan, X.; Gao, L.; Zhang, B.; Yang, F.; Liao, W. High-resolution aerial imagery semantic labeling with dense pyramid network. *Sensors* **2018**, *18*, 3774. [CrossRef]
50. Nemoto, K.; Hamaguchi, R.; Imaizumi, T.; Hikosaka, S. Classification of rare building change using cnn with multi-class focal loss. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 4663–4666.
51. Gao, X.; Sun, X.; Zhang, Y.; Yan, M.; Xu, G.; Sun, H.; Jiao, J.; Fu, K. An end-to-end neural network for road extraction from remote sensing imagery by multiple feature pyramid network. *IEEE Access* **2018**, *6*, 39401–39414. [CrossRef]
52. Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 778–782. [CrossRef]
53. Zhao, W.; Du, S. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
54. Roy, S.K.; Chatterjee, S.; Bhattacharyya, S.; Chaudhuri, B.B.; Platoš, J. Lightweight spectral–spatial squeeze-and-excitation residual bag-of-features learning for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5277–5290. [CrossRef]
55. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5966–5978. [CrossRef]