remote sensing

MDPI

Article

Classification for High Resolution Remote Sensing Imagery Using a Fully Convolutional Network

Gang Fu [1,2,*], Changjun Liu [2], Rong Zhou [3], Tao Sun [2] and Qijian Zhang [4]

1 Department Of Engineering Physics, Tsinghua University, Beijing 100084, China
2 China Institute of Water Resources and Hydropower Research (IWHR), Beijing 100038, China; lcj2005@iwhr.com (C.L.); sunt@iwhr.com (T.S.)
3 Beijing Soil and Water Conservation Center, Beijing 100036, China; zhour@bjwater.gov.cn
4 Water Resources Information Center of Henan Province, Zhengzhou 450003, China; zqj@hnsl.gov.cn
* Correspondence: gangfu2008@hotmail.com; Tel.: +86-10-6279-4967

Academic Editors: Qi Wang, Nicolas H. Younan, Carlos López-Martínez, Lenio Soares Galvao and Prasad S. Thenkabail
Received: 21 March 2017; Accepted: 16 May 2017; Published: 18 May 2017

Abstract: As a variant of Convolutional Neural Networks (CNNs) in Deep Learning, the Fully Convolutional Network (FCN) model achieved state-of-the-art performance for natural image semantic segmentation. In this paper, an accurate classification approach for high resolution remote sensing imagery based on the improved FCN model is proposed. Firstly, we improve the density of output class maps by introducing Atrous convolution, and secondly, we design a multi-scale network architecture by adding a skip-layer structure to make it capable for multi-resolution image classification. Finally, we further refine the output class map using Conditional Random Fields (CRFs) post-processing. Our classification model is trained on 70 GF-2 true color images, and tested on the other 4 GF-2 images and 3 IKONOS true color images. We also employ object-oriented classification, patch-based CNN classification, and the FCN-8s approach on the same images for comparison. The experiments show that compared with the existing approaches, our approach has an obvious improvement in accuracy. The average precision, recall, and Kappa coefficient of our approach are 0.81, 0.78, and 0.83, respectively. The experiments also prove that our approach has strong applicability for multi-resolution image classification.

Keywords: deep learning; convolutional neural network (CNN); fully convolutional network (FCN); classification; remote sensing; high resolution

1. Introduction

Classification is a fundamental task for remote sensing imagery analysis. Applying intelligent methods, such as pattern recognition and statistical learning, is an effective way to obtain class information of ground objects. It is always the main focus of research and commercial development. Early classification was mainly for low spatial resolution (10–30 m) images and pixel-leveled images, including unsupervised classification (also known as clustering, such as K-means [1]) and supervised classification (such as Neural Networks [2,3] and Support Vector Machines [4,5]). These methods often use only spectral information of the images, and have formed general modules in commercial software, and have been successfully applied in land resources, environment, agriculture, and other fields. In recent years, some new approaches have appeared that are much superior to the traditional approaches. For example, Yuan Yuan et al. [6] and Qi Wang et al. [7] applied the latest achievements in the machine learning field, such as Manifold Ranking and Sparse Representation, to hyperspectral image classification.

High resolution (2 m spatial resolution and higher) remote sensing images contain more ground details. Many applications tend to obtain attributes of a ground object (such as a single building) rather than pixels. However, the pixel-level classification methods are sensitive to noise, and lack semantic meaning of the objects, and are difficult for obtaining object-level information. Therefore, object-oriented classification [8] is proposed, and it has made great achievements in high resolution image classification. At present, eCognition [9], ENVI [10], and other commercial software have developed object-oriented classification modules. Most of the object-oriented approaches perform a "segmentation-classification" mode. In the segmentation stage, Multi-Resolution (MR) [11], Full-Lambda Schedule (FLS) [12], Mean-Shift [13], Quadtree-Seg [14], and other image segmentation approaches are used to generate image segments, which we called image objects. In the classification stage, object features (color, texture, and geometric features) are calculated, which are taken as inputs of supervised or unsupervised classification, or a manually designed rule set for feature filtering, to achieve the final class discrimination.

Land-cover has various types, and is affected by noise, illumination, season, and many other factors, and brings great difficulties to classification using high resolution images. Even using the object-oriented approaches, accurate classification is still very difficult. From the pattern recognition perspective, selection/extraction of representative features is the bottleneck to improving accuracy. That is, the use of a specific set of features cannot be achieved on the classification for all kinds of ground objects. Therefore, learning features automatically from a remote sensing data set rather than using manually designed features, and then performing classification on the learned features, is an effective way to improve the accuracy of classification.

Deep learning theory was explicitly proposed by Hinton et al. [15] in 2006. It is a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data [16]. The basic motivation of deep learning is to establish a deep neural network to simulate the leaning and analysis mechanism of the human brain. Compared with the traditional machine learning theories, the most significant difference of deep learning is emphasizing automatic feature learning from a huge data set through the organization of multi-layer neurons. In recent years, various deep learning architectures such as Deep Belief Networks (DBN) [17], Convolutional Neural Networks (CNN) [18], and Recurrent Neural Networks (RNN) [19] have been applied to fields like computer vision [20,21], speech recognition, natural language processing, audio recognition, and bioinformatics, and they have been shown to produce state-of-the-art results in these domains.

In deep learning techniques, CNN has achieved remarkable results in image classification, recognition, and other vision tasks, and has the highest score on many visual databases such as ImageNet, Pattern Analysis, Statistical Modeling and Computational Learning Visual Object Classes (PASCAL VOC), and Microsoft Common Objects in Context (MS-COCO). For image classification, the basic structure of the standard CNN is stacks of "convolutional-pooling" layers as multi-scale feature extractors, and subsequent numbers of fully connected layers as classifiers. Many works on CNN-based remote sensing image analysis emerged in recent years. Nguyen et al. [22] presented an approach for satellite image classification using a five-layered network and achieved classification accuracy higher than 75%. Wang et al. [23] used a CNN structure with three layers and Finite State Machine (FSM) for road network extraction for long-term path planning. Marco Castelluccio et al. [24] explored the use of CNNs for the semantic classification of remote sensing scenes. Similarly, Hu et al. [25] also classified different scenes from high resolution remote sensing imagery using a pre-trained CNN model. Weixun Zhou et al. [26] employed CNN architecture as a deep feature extractor for high-resolution remote sensing image retrieval (HRRSIR). Volodymyr Mnih [27] proposed a CNN-based architecture to learn large scale contextual features for aerial image labeling. The model produces a dense classification patch, instead of outputting a single value image category. Martin Lagkvist et al. [28] presented a novel remote sensing imagery classification method based on CNNs for five classes (vegetation, ground, road, building, and water), outperforming the existing classification approaches. Besides the CNN family approaches, Yuan Yuan et al. [6] used a Stacked

AutoEncoder classifier for a classification experiment after using the Manifold Ranking based salient band selection.

The standard CNN is in an "image-label" manner and its output is the probability distribution over different classes. However, most of the remote sensing image classification expects a dense class map as the output, which has the same dimensions as the original image. A class map is a 2-D distribution of class labels with pixel correspondence, which is in a "pixel-label" mode. In the study of Martin Lagkvist et al. [28], a "per-pixel" classification is considered using overlapped patches and average post-processing. However, the use of the overlapped patches introduces too much redundant computations, and the averaging processing may easily lose useful edge information. Based on the standard CNN, Jonathan Long et al. [29] proposed the Fully Convolutional Network (FCN) model in 2015. By replacing fully connected (FC) layers in the standard CNN with convolutional layers, the FCN model maintains the 2-D structure of images, and firstly carries out CNN-based image semantic segmentation. In order to obtain a dense class map, Liang-Chieh Chen et al. [30] used the "atrous" convolution instead of the ordinary convolution, increasing the density of the predicted class labels, and then performed the Conditional Random Fields (CRFs) as post-processing to refine the region boundaries. The CRFs-based boundary refinement is also used in the works of Sakrapee et al. [31]. In order to integrate the CRFs procedure into the training stage, Shuai Zheng et al. [32] applied the idea of RNN to image segmentation, implementing an "end-to-end" training procedure. In the remote sensing society, several studies employ FCN-based approaches for dense class map generation. Jamie Sherrah [33] analyzed the down-sampling and up-sampling mechanism in CNNs, and adopted an FCN architecture for aerial image semantic labelling. The down-sampling mechanism of standard FCN is removed by involving deconvolution. D. Marmanis et al. [34] also used FCN and subsequent deconvolution architecture to perform a semantic segmentation for aerial images. Emmanuel Maggiori et al. [35–37] addressed the dense classification problem, and compared the patch-based CNN dense classification using CNN with FCN. With the advantages of FCN, the author proposed an end-to-end framework for large-scale remote sensing classification. A multi-scale mechanism was also considered by designing a specific neuron module that processes its input at multiple scales.

In this paper, we perform a FCN-based classification on high spatial resolution remote sensing imagery with 12 classes (bare land, grass, tree, water, building, cement ground, parking lot, playground, city road, trail, shadow, and others). These classes are typical ground objectives in city areas, and some of them (such as building, cement ground, road, and parking lot) are easily confused in traditional classification tasks. The class configurations were arranged to test the effectiveness of our approach in a complex environment. We fine-tuned the model parameters of the ImageNet-pretrained VGG-16 [37] network using GF-2 satellite images, to adapt it to our remote sensing imagery classification task. The VGG network has a more compact structure of convolutional and pooling layers, and achieved the highest classification accuracy for ImageNet ILSVRC-2014. To overcome the noise caused by pixel-level classification, we refine the region boundaries using fully connected CRFs, following the procedure of Liang-Chieh Chen et al. [30] and Sakrapee et al. [31]. The refined output is more readily applied to an object-oriented analysis.

We compare our approach with the object-oriented approach with MR segmentation [11] and SVM classification, patch-based CNN classification proposed in [27], and the FCN-8s approach proposed in [29], which achieved success for high resolution imagery classification or natural image segmentation. The result shows that our approach achieves higher accuracy in the classification. For those objectives which are difficult to be classified, our approach has lower confusion rates.

2. Methods

Similar to other supervised classification, our approach generally has two stages: the training stage and the classification stage, which is illustrated in Figure 1. In the training stage (the upper part of Figure 1), image-label pairs, with pixel-class correspondence, are input into the FCN network as training samples. The error between predicted class labels and ground truth (GT) labels is calculated

and back-propagated through the network using the chain rule, and then the parameters of the FCN network are updated using the gradient descent method. The above iteration will be stopped when the error is less than a given threshold. In the classification stage (the lower part of Figure 1), the trained FCN network is performed on an input image to generate a rough class prediction. The rough class prediction, with the input image, is then input into the CRFs post-processing module to generate the final refined classification. The details of the training stage and classification stage are presented in Sections 2.2 and 2.3, respectively.

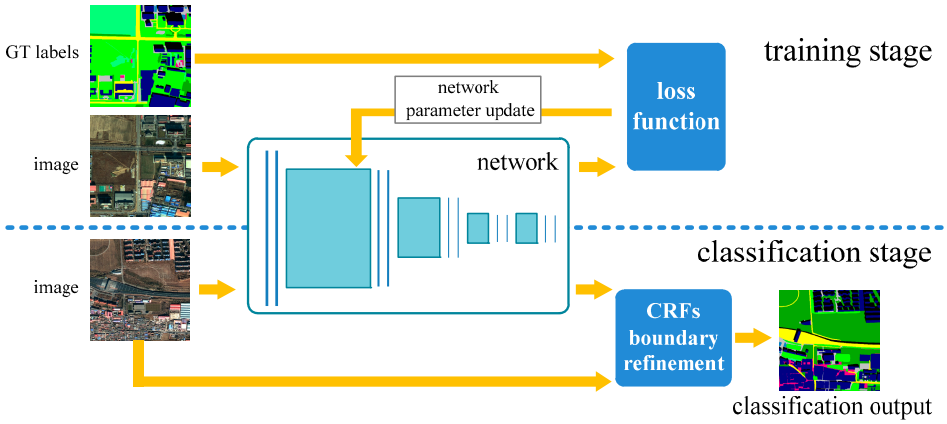


Figure 1. The general pipeline of our approach: The training stage and the classification stage are illustrated in the upper and lower parts, respectively.

2.1. Network Architecture

CNN currently is the state-of-the-art in visual recognition such as classification and detection. Simonyan et al. [38] developed the very deep CNN networks (VGG) by increasing the depth to 16–19 weight layers. To reduce the number of parameters in the networks, small 3×3 filters are used in all the convolutional layers. VGG models won the runner-up in ImageNet ILSVRC-2014. Although the subsequently emerged deeper models, such as ResNet [39] and Inception-V4 [40], achieved a higher score in many vision tasks, VGG networks have clear structures and compact memory requirements, which can be easily extended and applied, so we chose the 16-layered VGG network as our basic network architecture. Based on the VGG network, we constructed the FCN model by replacing the last three fully connected layers (two layers with 4096 neurons and one with 1000 neurons) with convolutional layers. Then following the idea of Liang-Chieh Chen et al. [30], we use "atrous" convolution (also known as "dilation" convolution in other studies) instead of the ordinary convolution to increase the feature density, and build the multi-scale classification model by adding the skip-layer network architecture.

2.1.1. Fully Convolutional Network

In classification tasks, the last structures in standard CNN are always several Fully Connected (FC) layers (see Figure 2a for illustration). These layers play the role of classifier like standard BP neural networks (For example, in Figure 2a, the 3 FC layers are similar to a 3-layered BP network with one hidden layer). From the first FC layer, the 2-D structure of the input image maintained by the convolutional-pooling layers is lost. The output of standard CNN is a 1-D distribution over classes (for a Softmax regression). It works in an "image-label" manner. In other words, given an image, it predicts one class label (a scalar) for it. The "image-label" mode has great advantages in single scene classification. The effectiveness has been presented in studies of Marco Castelluccio et al. [24] and Hu et al. [25].

However, in most remote sensing applications, a 2-D dense class map is required as an output. To maintain the 2-D structure, some approaches were presented based on the common CNN structures. The most typical one is the patch-based CNN approach [27,28]. The basic idea of patch-based CNN is: separate the large image into small patches, and apply the common CNN model on each patch to predict the class label(s) centered at the corresponding patch. Finally, the class labels will be arranged in a 2-D layout as the output. Jonathan Long et al. [29] proposed the FCN model, which is a convolutionalized version of CNN. FCN replaces all the FC layers with convolutional layers. Thus, the important 2-D structure of the image is maintained. Figure 2b is the illustration of the FCN model.

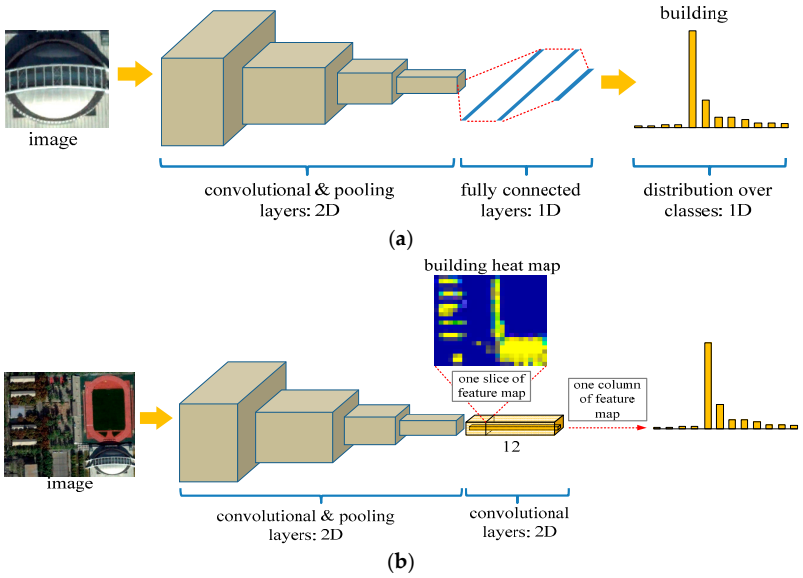


Figure 2. Network architectures for standard Convolutional Neural Network (CNN) and Fully Convolutional Network (FCN). (**a**) Architecture of standard CNN: stacks of convolutional-pooling layers and fully connected (FC) layers. Given an image, the distribution over classes is predicted. The class with the largest distribution value is considered as the class of a given image; (**b**) Architecture of FCN: FC layers are replaced by convolutional layers. FCN maintains the 2-D structure of the image.

Compared with patch-based CNN, the advantages of the FCN model are obvious for

- *Easy implementation*: The FCN architecture is designed brilliantly by replacing the FC layers by convolutional layers, which enables us to take arbitrary sized images as inputs. Additionally, by training entire images at a time instead of patch cropping, FCN does not have to rearrange the output labels together to obtain the label predictions and thus reduces the implementation complexity.
- *Higher accuracy*: Under the patch-based CNN learning framework, only the "intra-patch" context information is taken into account. Nevertheless, correlations among patches are ignored, which might lead to obvious gaps between patches. Unlike the patch-based CNN, FCN performs the classification in a single-loop manner, and considers the context information overall and seamlessly. Please refer to Section 4.2 for more details.
- *Less expensive computation*: In patch-based CNN, when using overlapped patches for dense class label generation, such as the study of Martin Lagkvist et al. [28], it introduces too much redundant computations (especially convolutions) on the overlapped regions. By performing a single loop operation, the FCN model makes remarkable progress and allows the large image classification to be implemented in a more effective way.

We adopt the FCN model for remote sensing imagery classification. The output number (channels) of the last convolutional layer (also called feature maps) is equal to the class number of our task (so in

this paper, it is 12 for 12-class classifications). The feature maps can be seen as a stack of heat maps for all classes. A 2-D slice along the channel axis represents the heap map (score distribution) of the corresponding class (For example in Figure 2b and in Figure 3c, we extract the heap map for the building).

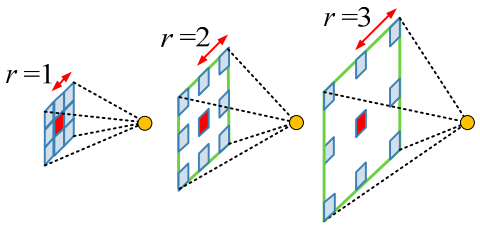


Figure 3. "Atrous" convolutions with $r = 1,\ 2,$ and 3. The first convolution ($r = 1$) is actually the ordinary convolution.

2.1.2. Atrous Convolution for Dense Feature Extraction

The repeated combination of pooling and striding at consecutive layers significantly reduces the spatial resolution of the resulting feature map. Typically in our VGG-16 model, 5 max-pooling layers with $1/2$ down-sampling cause $1/32$ total factor reduction in spatial resolution. For high resolution remote sensing image classification tasks, such operations lead to a serious loss of spatial information. Liang-Chieh Chen et al. [30], inspired by the Wavelet Transform, proposed the "atrous" convolution for generating dense feature maps. In the 1-D case, given the input signal $x[i]$, and the convolutional kernel w, the output of "atrous" convolution $\mathrm{y}[i]$ is calculated as:

$$\mathrm{y}[i] = \sum_{k=1}^{K} x[i + r{\cdot}k]w[k] \tag{1}$$

where r denotes the *rate* parameter corresponding to the stride. In the 2-D cases, "atrous" convolutions (use 3×3 kernel) with rate $r = 1,\ 2,$ and 3 are demonstrated in Figure 3.

In order to further illustrate the effect of "atrous" convolution, we compare it with standard convolution using a simple example in Figure 4. Firstly, represented by the red route, we take an image patch (300×300) as an input, and perform $1/2$ down-sampling and 10×10 standard convolution (horizontal Gaussian derivative kernel) on it, which is used to simulate a pooling-convolution combination in standard CNNs. The receptive field corresponding to the original image is 20×20, and only $1/4$ of the image positions are involved in calculating the feature map. The obtained low resolution feature map is then enlarged by an up-sampling operation with a factor of 2. Secondly, as a comparison, we perform "atrous" convolution with rate $r = 2$ on the original image. The size of the receptive field is unchanged, but the density of the feature map is increased by two times, which means half of the image positions are considered for generating the feature map. Compared with the standard convolution, the "atrous" convolution generates a high resolution feature map, while keeping the size of receptive field. Besides, there is no extra parameter involved. The "atrous" convolution for dense feature map generation is illustrated by the blue route in Figure 4.

The "atrous" convolution is generally applicable and allows us to efficiently compute dense CNN feature maps at any target subsampling rate without introducing any approximations and extra parameters. Theoretically, the "atrous" convolution can be applied to each convolutional layer of the network to maintain the resolution, but this ends up being too costly, and the advantage for translation invariant brought by the down-sampling operation could also be weakened. So we modify the basic VGG-16 network to adapt it to our classification task. We take this modified network as our primary architecture (we add multi-scale functionality, which is described in Section 2.1.3).

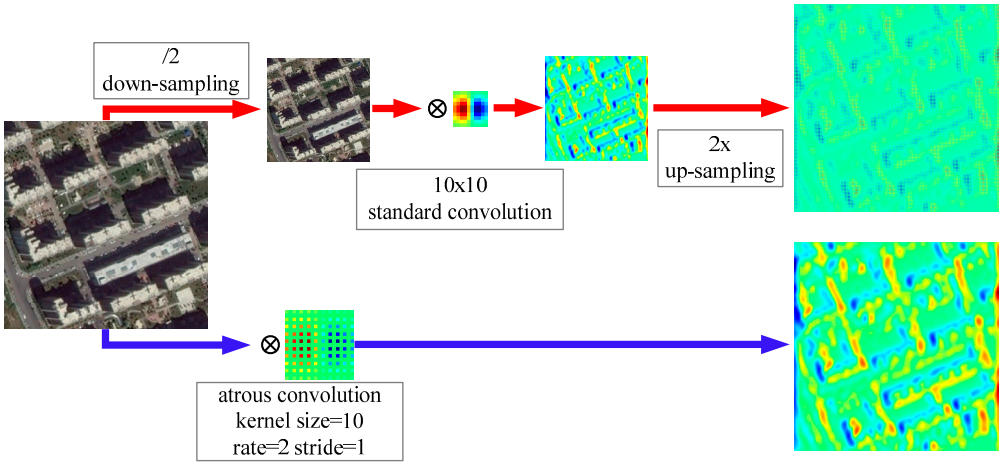


Figure 4. Illustration of atrous convolution for dense feature map generation. Red route: standard convolution performed on a low resolution feature map. Blue route: dense feature map generated using atrous convolution with rate $r = 2$ on a high resolution input feature map.

2.1.3. Network Architecture for Multi-Scale Classification

The variant of resolution will affect the classification accuracy. Single-scale classification has great limitation in its applicability. Therefore, many works considered multi-scale classification in their approaches [29–32]. A simple method for a multi-scale classification is training the model on datasets that contain objects of varying sizes. However, this approach needs the times of sample storage and training time (more iteration to traverse all the samples). A good idea for CNN-based multi-scale segmentation and detection is using the skip-layer network architecture [29,41]. In this architecture, links are added to incorporate the feature responses from different levels of the primary network stream, and these responses are then combined in a shared output layer [42]. Our multi-scale network architecture is illustrated in Figure 5.

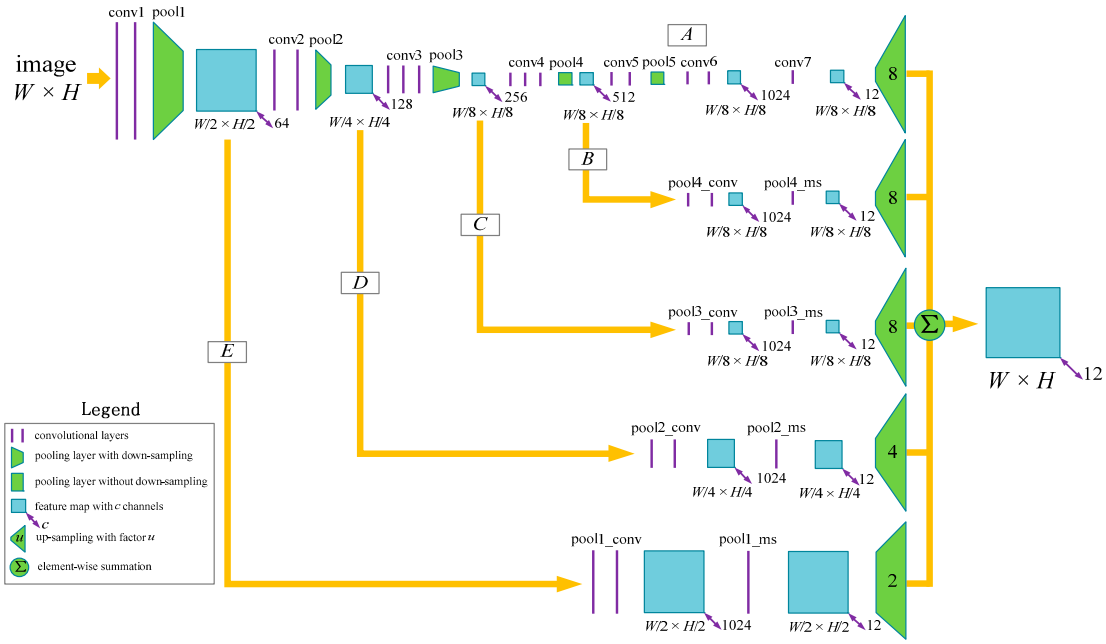


Figure 5. Multi-scale network architecture.

As presented in Figure 5, feature maps are generated along five streams. The stream A is our primary network, generating a feature map with dimension $W/8 \times H/8 \times 12$, which is described in Section 2.1.2. Branch streams B to E are the added skip-layer architecture for the multi-scale classification. These streams begin from the feature map generated by layers pool4 to pool1, respectively. For each branch stream, the subsequent architecture is the layer group with two convolutional layers, generating a feature map with 1024 channels, and then a convolutional layer (kernel $1 \times 1 \times 12$) outputs a 12-channeled feature map. Each stream, including the primary stream and the branch streams, introduce down-sampling effects caused by the max-pooling operation (the factor is $1/8$ for stream A to C, $1/4$ for stream D, and $1/2$ for stream E). However, in the applications of remote sensing classification, we need the class map to have the same size with the input image. So we perform the up-sampling operation after the feature maps are generated by these streams to recover the feature maps at the original image resolution. In this paper, we adopt Liang-Chieh Chen et al.'s [30] approach, and use simple bilinear interpolation to increase the resolution by a factor of 8, 4, and 2 at negligible computational cost. The up-sampled feature maps are then combined using summation in an element-wise manner. The output of this network architecture is a feature map with dimension $W \times H \times 12$. Our multi-scale network architecture captures three levels of resolution, represented by stream A to C, stream B, and stream E.

2.2. Network Training

Our training dataset is collected from two GF-2 high resolution remote sensing images (true color fusion images with 0.8 meter resolution) of northeastern Beijing, China.

The images were taken in 5 December 2014 and 2 September 2015, respectively. The reason why we chose images with different imaging times is to increase the anti-interference abilities of our model, such as the change of seasons, to enhance its applicability. In our training dataset, there are a total of 74 images (size 1024×1024). We manually labeled all images at the pixel level as ground truth (GT) label data. In other words, for each image, there exists a 1024×1024 label map, having a pixel-class (row-col indexed) correspondence with it. We used 70 images for training, and the remaining 4 images for testing. Three image-GT label pair examples are illustrated in Figure 6.

The general procedure of our training stage is: Image-GT label pairs are input into the multi-scale classification network as training samples. The *Softmax* function is performed on the output feature map generated by the network to predict the class distribution. Then the cross entropy loss is calculated and back-propagated, and finally the network parameters are updated using Stochastic Gradient Descent (SGD) with momentum. The general procedure is shown in Figure 7.

The softmax function is used to *probabilize* the output feature map of our multi-scale network. However, the mode of softmax here is different from that in the standard CNNs: it is performed on each location with row-column coordinate $(i,\ j), 0 \leq i < H\ and\ 0 \leq j < W$, and it outputs a dense distribution over the classes. Figure 8 illustrates this function.

Figure 8 shows that the output of our multi-scale network is a $H \times W \times 12$ feature map, which has the same width and height as the original image. A "drill hole" along the channel axis at location $(i,\ j)$ is the feature vector with 12 elements corresponding to the pixel at the same location. The softmax function is adopted on this feature vector to generate a 12-D probabilized vector, which is the discrete distribution over 12 classes at location $(i,\ j)$. The softmax function will traverse each location to obtain the dense class distribution.

The SGD method with momentum is used for parameter updates in our training, which is described by the following:

$$W^{(n+1)} = W^{(n)} - \Delta W^{(n+1)} \tag{2}$$

where $W^{(n)}$ and $W^{(n+1)}$ denote the old parameters and new parameters, respectively, and $\Delta W^{(n+1)}$ is the increment for the current iteration, which is a combination of old parameters, gradient, and historical increment:

$$\Delta W^{(n+1)} = \eta \cdot \left(d_w \cdot W^{(n)} + \frac{\partial J(W)}{\partial W^{(n)}} \right) + m \cdot \Delta W^{(n)} \tag{3}$$

where $J(W)$ is the loss function, η is the learning rate for step length control, and d_w and m denote the weight decay and momentum, respectively.

We employ the VGG-16 network which has been pre-trained on ImageNet for fast convergence. We use a "step" policy for learning rate adjustment ($gamma = 0.1$, $step\_size = 15,000$) so that closer to the error minimum, the smaller the step length is. The base learning rate is 0.0001. The basic parameters for calculating increments are: $m = 0.9$, and $d_w = 0.0005$. The max iteration in our training is 60,000. In the training procedure, we first randomly shuffle the samples, and then feed them into the network in batches. Each batch contains 10 images. We also crop and rotate samples randomly in each batch to increase the diversity and variability of the samples.

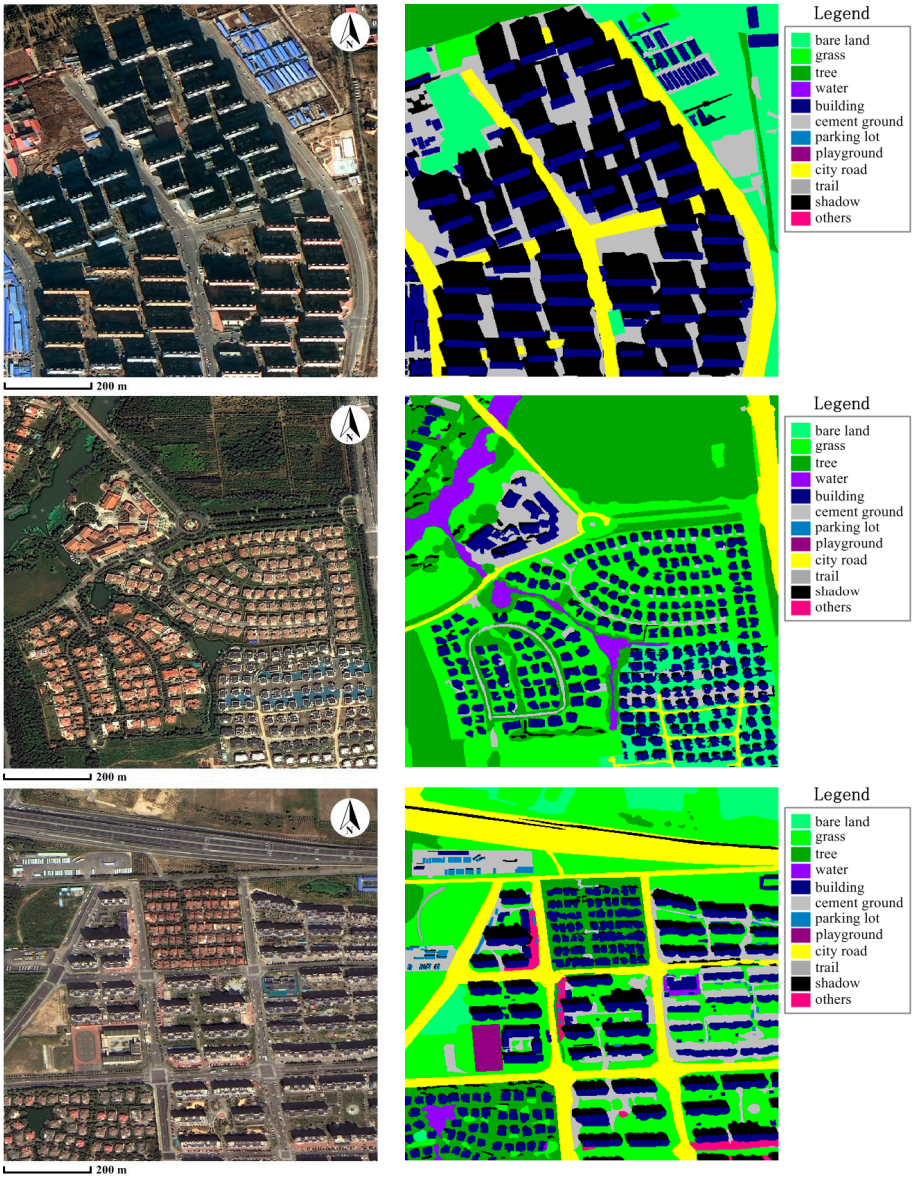


Figure 6. Three sample examples for our classification training. (**a**) Original images; (**b**) Ground truth (GT) labels corresponding to the images in (**a**).

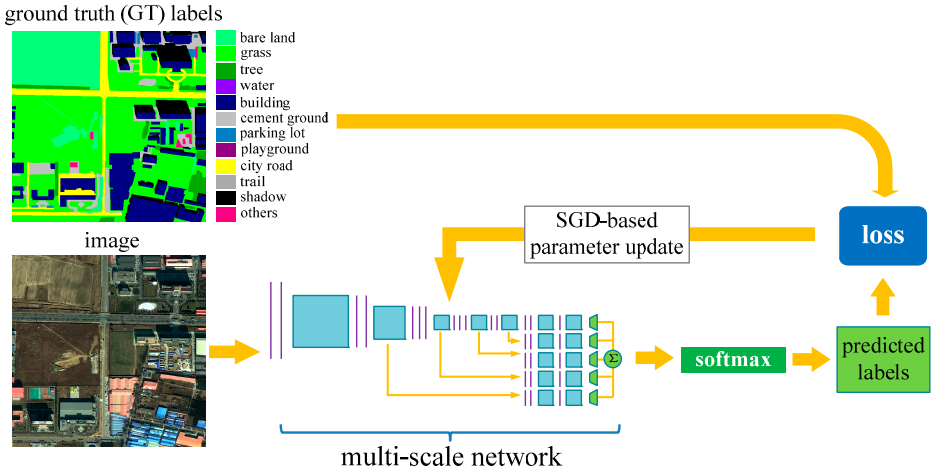


Figure 7. General procedure of network training.

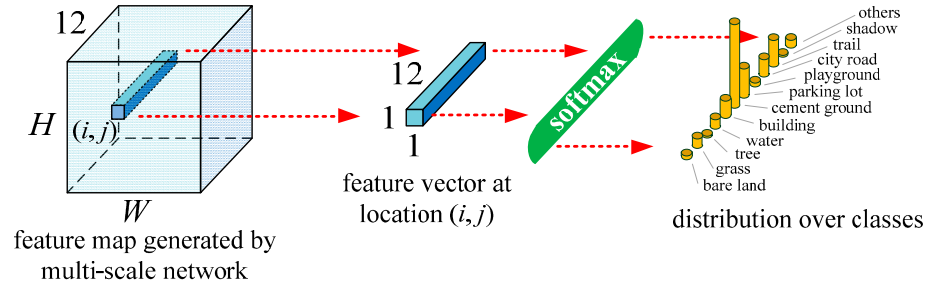


Figure 8. Softmax function performed on the output feature map.

2.3. Classification Using the Trained Network

The trained network is adopted on an image for classification. However, our multi-scale network involves up-sampling operations, leading to the blurring of classification boundaries. Several works [29–32] use CRFs as post-processing to refine the image segmentation results. So following their idea, we adopt the fully connected CRFs for our rough class prediction. The model employs the energy function:

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \tag{4}$$

where x is the label assignment for pixels. $\theta_i(x_i) = -\log P(x_i)$ is the *unary potential*, where $P(x_i)$ is the label assignment probability at pixel i as the output of our multi-scale network after the softmax function. $\theta_{ij}(x_i, x_j)$ is the *pairwise potential* represented by a fully connected graph, connecting all pairs of image pixels i and j. We use the following definition of the pairwise potential [43]

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w_m \cdot k^m\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right) \tag{5}$$

where $\mu(x_i, x_j)$ is the sign function, and $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$, and is zero otherwise. $\mu(x_i, x_j)$ removes the self-connected links from the graph. k^m is a Gaussian kernel function that takes feature as input (denoted by $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ extracted for pixel i and j). Each Gaussian kernel is weighted by w_m. In our study, the bilateral position and color terms is adopted as the kernel function

$$w_1 \cdot \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right) + w_2 \cdot \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right) \tag{6}$$

where p_i, p_j denote the locations, and p_i, p_j denote the color of pixel i, j. So the first kernel depends on both pixel positions and color, and the second kernel only depends on pixel positions. σ_α, σ_β, and σ_γ are the hyper parameters that control the scale of the Gaussian kernels. The classification pipeline is illustrated in Figure 9.

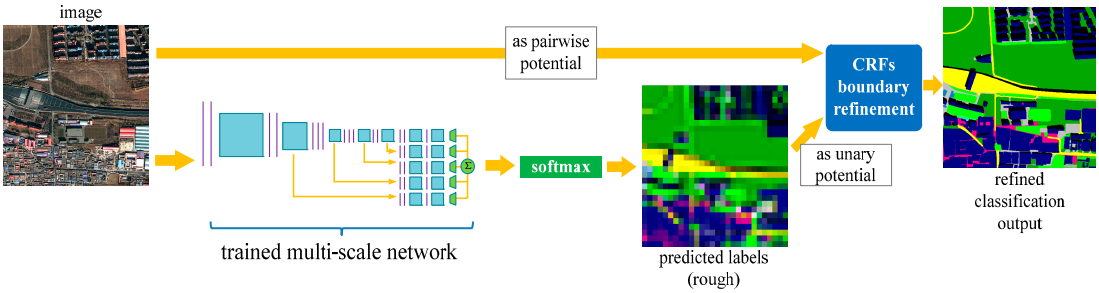


Figure 9. General procedure of image classification using the trained network.

In CRFs post-processing, the rough class distribution predicted by the multi-scale network is input as the unary potential, and the original image provides the pairwise potential with position and color information. The CRFs is solved using mean field approximation [43]. The class labels are adjusted and refined under the position-color constraints. The weight parameters we adopt in this paper are $w_1 = 4, w_2 = 3$, which are the default configuration of [30]. Following the idea of [43], we use $\sigma_\alpha = 54, \sigma_\beta = 5,$ and $\sigma_\gamma = 4$ through a cross-validation on the training set. We employ 10 mean field iterations for solving CRFs.

3. Experiment and Comparison

In the following section, the experiment and comparison will be presented to evaluate our classification approach. Our algorithm is implemented using Microsoft Visual C++ 11, and is performed on the Windows 7 operating system installed NVIDIA GeForce GTX980M graphic device with 8G byte graphic memory.

3.1. Comparison Setup

We conduct two groups of experiment (denoted as Experiment A and B) on GF-2 and IKONOS true color images, respectively. We compare our approach with object-oriented classification using MR segmentation [11], SVM classification (MR-SVM), patch-based CNN classification proposed in [27], and the FCN-8s approach proposed in [29].

3.1.1. MR-SVM

For Multi-Resolution and Support Vector Machine (MR-SVM) object-oriented classification, the first step is MR segmentation [11] to generate image objects. The quality of image objects directly affects the classification results. We believe that the high quality image objects are neither over-covered nor over-segmented. Ideally, each image object contains only a single-class ground object. The MR segmentation is controlled by the scale, shape, and compactness parameters. In order to obtain high-quality image objects, we determine the parameters through the times of experiments by different settings, to achieve the ideal segmentation as much as possible. The parameters we used in MR segmentation are listed in Table 1.

Once the image objects are obtained, we construct the initial feature space using 60 common features involving spectral, geometric, and texture aspects:

- *Spectral features*: mean, standard deviation, brightness, and max difference for each band.
- *Geometric features*: area, length, width, length-width ratio, border length, compactness, elliptic fit, rectangular fit, density, shape index, main direction, and symmetry.

- *Texture features*: Features calculated from the Gray Level Co-occurrence Matrix (GLCM) and the Gray Level Difference Vector (GLDV) with all directions, etc.

Table 1. Scale, shape, and compactness parameters used in the Multi-Resolution (MR) segmentation.

Experiment	Scale	Shape	Compact
Exp.A-(1)	115	0.5	0.5
Exp.A-(2)	140	0.3	0.8
Exp.A-(3)	105	0.4	0.5
Exp.A-(4)	100	0.4	0.7
Exp.B-(1)	120	0.3	0.5
Exp.B-(2)	80	0.5	0.4
Exp.B-(3)	85	0.5	0.7

To select the most representative features for the following classification, we seek significant features for optimal class separation using the Separability and Thresholds (SEaTH) method [44]. According to the SEaTH method, we optimize the 60-D initial feature space, and obtain a 10-D sub feature space including: mean value and brightness for each band; density and length-width ratio of the image object; GLCM-mean value for each band; GLDV-mean for the first band. In the classification stage, we select almost 25% of the image objects from each image as training samples, and input their features to the SVM classifier implemented using the LibSVM library [45]. The kernel function we used in SVM is the Radial Basis Function (RBF), and the objective function type is the C-Support Vector Classification (C-SVC). To determine the optimal penalty factor C and kernel function parameter γ, we employ a simple grid search for all training samples on the $C - \gamma$ domain that minimize the classification error. The search range of C and γ are $[0.4, 1.6]$ and $[0.02, 0.14]$ according to the experience [45]. The step lengths are 0.2 and 0.01, respectively. According to the grid search, the optimal parameters we used for the SVM classifier are $C = 1.2$ and $\gamma = 0.08$.

3.1.2. Patch-Based CNN

In the patch-based classification experiment, the general procedure is illustrated in Figure 10.

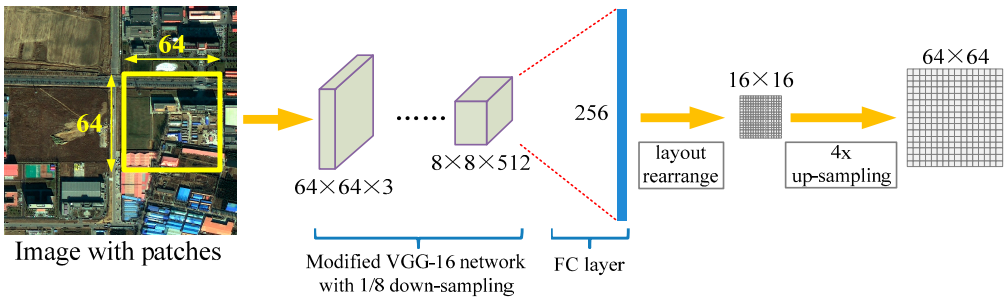


Figure 10. General procedure of our patch-based CNN classification experiment.

Different from the architecture used in [27], we employ the VGG-16 network as the main structure for its high performance in the previous vision tasks. In order to prevent excessive reduction of the resolution, we modified the stride and padding values of the last two pooling layers (the stride and padding values we used are all 1) so that the architecture has a $1/8$ down-sampling effect. Following the idea of Volodymyr Mnih [27], the last 3 FC layers are modified to a single FC layer with output number 256 representing a 16×16 prediction area. So for 64×64 input patches, the overall architecture causes a $1/4$ down-sampling. Finally, we perform an up-sampling post-processing with a factor of 2 to increase the resolution.

3.1.3. FCN-8s

For the FCN model, we directly employ the FCN-8s model proposed by Jonathan Long et al. [29]. The architecture of the model is also the VGG-16 network with skip-layer structure. The final prediction is fused from the output of three branches (from the primary network, the pool4 layer, and the pool3 layer, respectively) after the up-sampling operation. In the training phase, by modifying the number of outputs from 21 to 12, we fine-tuned the network based on the ImageNet pre-trained model. The training parameters for FCN-8s in the experiment are the same as ours. In the testing stage, except for the CRF-based post-processing, we use the same classification parameters as our approach. Please refer to [29] for detailed information.

3.2. Experiments and Comparison

In Experiment A, we adopt our trained model on four GF-2 true color images (0.8 m resolution) for the classification (In the following section, they will be abbreviated as Exp.A-(1) to Exp.A-(4)). All the image sizes are 1024 × 1024. These images are the testing images that are not involved in training. Figure 11 is the illustration of the results and the comparison. In Experiment B, we adopt the same trained model on three IKONOS true color images (1.0 m resolution) for the classification (Abbreviated as Exp.B-(1) and Exp.B-(3) in the following section) to test the applicability. All the image sizes are also 1024 × 1024. Figure 12 illustrates the classification results and comparison.

We employ precision, recall, and Kappa coefficient as the indicators to evaluate our approach. These indexes are calculated from the confusion matrix C, where the precision is calculated as $\frac{1}{12}\sum_i C_{ii} / \sum_j C_{ij}$ that denotes the average proportion of pixels being classified to one class that are correct, and the recall is computed as $\frac{1}{12}\sum_i C_{ii} / \sum_i C_{ij}$ that represents the average proportion of pixels that are correctly classified, and the Kappa coefficient measures the consistency of the predicted classes with the GT classes. The comparisons are listed in Table 2.

Table 2. Comparison between approaches using MR-SVM, patch-based CNN, FCN-8s, and our approach.

Approach	Index	Exp.A-(1)	Exp.A-(2)	Exp.A-(3)	Exp.A-(4)	Exp.B-(1)	Exp.B-(2)	Exp.B-(3)	Mean
MR-SVM	Precision	0.67	0.72	0.67	0.66	0.65	0.73	0.64	0.68
	Recall	0.52	0.59	0.52	0.63	0.39	0.51	0.74	0.56
	Kappa	0.55	0.66	0.62	0.65	0.54	0.64	0.64	0.61
Patch-based CNN	Precision	0.68	0.64	0.71	0.55	0.73	0.76	0.70	0.68
	Recall	0.61	0.61	0.70	0.73	0.47	0.58	0.74	0.63
	Kappa	0.64	0.69	0.62	0.70	0.63	0.71	0.75	0.68
FCN-8s	Precision	0.83	0.84	0.68	0.66	0.81	0.78	0.83	0.78
	Recall	0.71	0.79	0.80	0.80	0.66	0.66	0.79	0.74
	Kappa	0.73	0.80	0.81	0.80	0.76	0.81	0.82	0.79
Ours	Precision	**0.86**	**0.87**	**0.74**	**0.68**	**0.84**	**0.78**	**0.92**	**0.81**
	Recall	**0.83**	**0.78**	**0.81**	**0.82**	**0.70**	**0.68**	**0.84**	**0.78**
	Kappa	**0.79**	**0.85**	**0.84**	**0.83**	**0.78**	**0.84**	**0.89**	**0.83**

The above statistics show our approach obtains the best performance compared with the others. Approaches using carefully-designed MR-SVM and patch-based CNN achieve similar accuracy levels, and the FCN-8s approach performs much better than those two. Some ground objects such as building, city road, and cement ground, have similar spectral and geometrical features, which are hard to distinguish. For example, in Exp.A-(2), when using MR-SVM, the recall for "cement ground" is 0.41. That means that more than half of the pixels are wrongly classified. The proportions that are incorrectly classified as "building" and "road" are 0.26 and 0.19. It means that in that case, the object-oriented classification has almost no effect on distinguishing these classes.

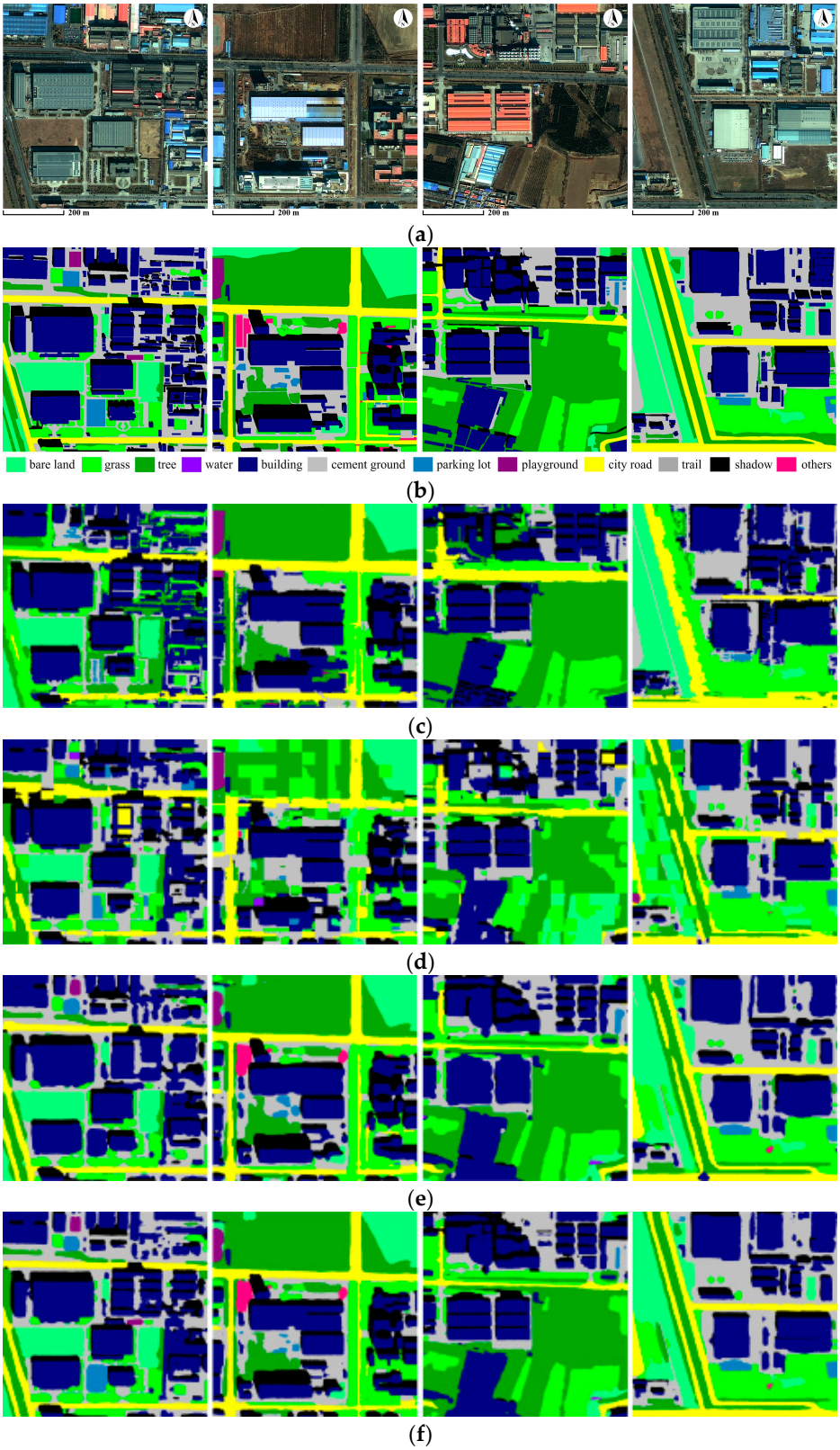


Figure 11. Classification results on GF-2 images (Experiment A). (**a**) Original images; (**b**) GT labels corresponding to the images in (**a**); (**c**–**e**) Results of the MR-SVM object-oriented classification, patch-based CNN classification, and FCN-8s classification corresponding to the images in (**a**), respectively; (**f**) Our classification results corresponding to the images in (**a**).

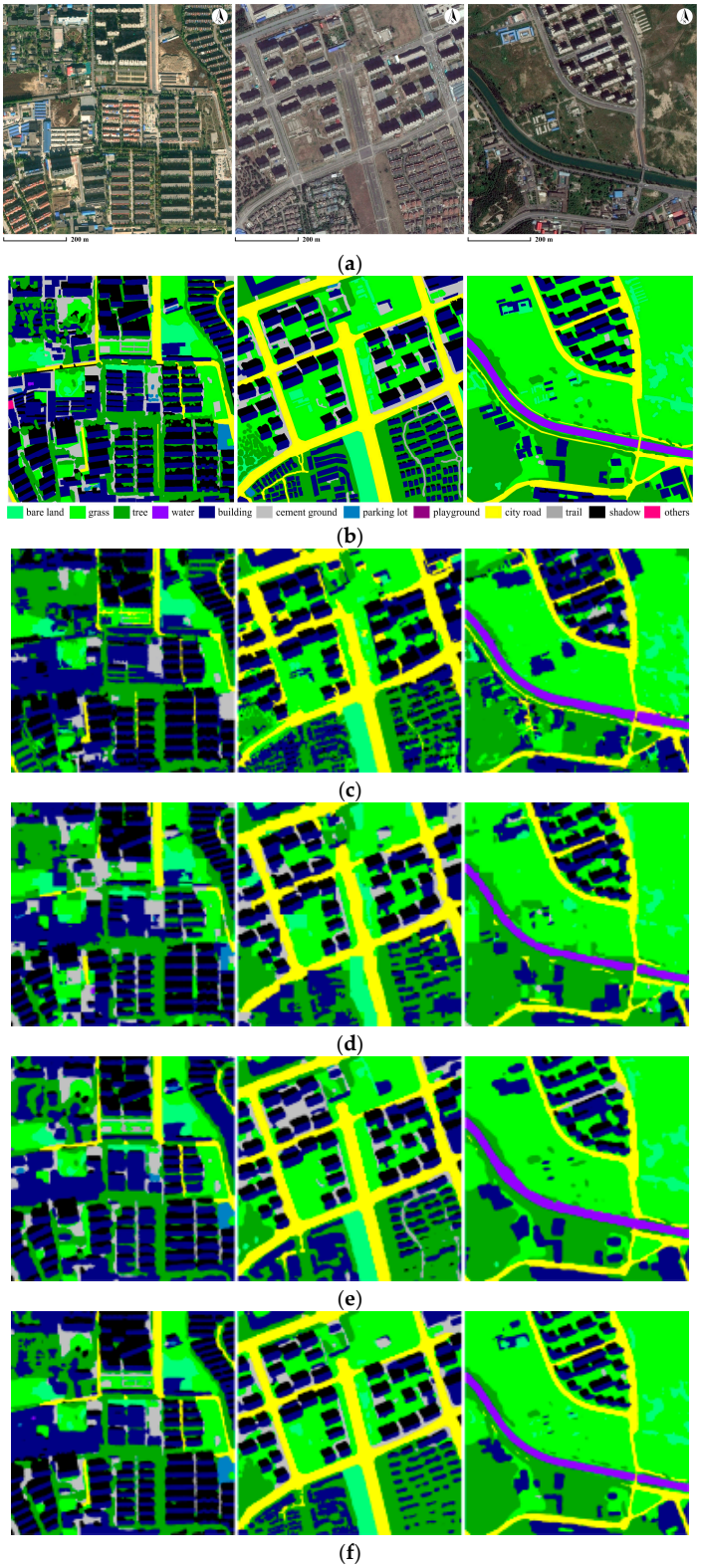


Figure 12. Classification result on IKONOS images (Experiment B). (**a**) Original images; (**b**) GT labels corresponding to the images in (**a**); (**c**–**e**) Results of the MR-SVM object-oriented classification, patch-based CNN classification, and FCN-8s classification corresponding to the images in (**a**), respectively; (**f**) Our classification results corresponding to the images in (**a**).

Table 3 lists the partial confusion matrix (only involves the above three classes) of our classification results. From the table, we can see that our approach achieves higher classification performance. In the above example, our recall for "cement ground" is 0.79. The proportions that are wrongly classified as "building" and "city road" are 0.05 and 0.06, respectively.

Table 3. Partial confusion matrix of our approach for "building", "cement ground", and "city road".

Experiment	GT/Predicted Class	Building	Cement Ground	City Road
Exp.A-(1)	Building	**0.91**	0.05	0.02
	Cement ground	0.13	**0.76**	0.02
	City road	0.02	0.01	**0.95**
Exp.A-(2)	Building	**0.92**	0.03	0.03
	Cement ground	0.05	**0.79**	0.06
	City Road	0.01	0.04	**0.89**
Exp.A-(3)	Building	**0.91**	0.02	0.05
	Cement ground	0.10	**0.82**	0.03
	City road	0.05	0.04	**0.82**
Exp.A-(4)	Building	**0.95**	0.03	0.00
	Cement ground	0.07	**0.81**	0.05
	City road	0.01	0.01	**0.93**
Exp.B-(1)	Building	**0.90**	0.02	0.01
	Cement ground	0.26	**0.65**	0.01
	City road	0.11	0.03	**0.84**
Exp.B-(2)	Building	**0.83**	0.01	0.00
	Cement ground	0.08	**0.75**	0.15
	City road	0.01	0.01	**0.96**
Exp.B-(3)	Building	**0.87**	0.06	0.01
	Cement ground	0.03	**0.70**	0.04
	City road	0.10	0.01	**0.87**

4. Discussion

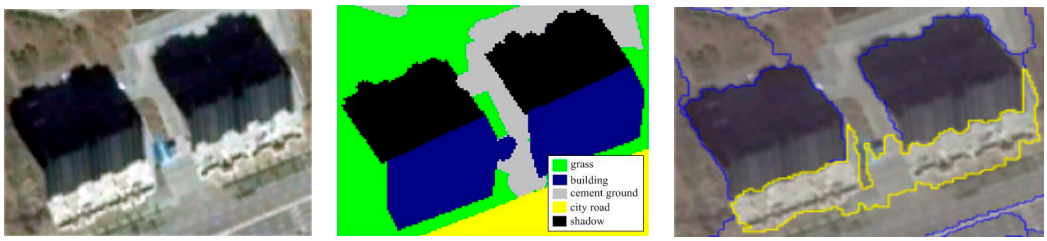
This paper presents a classification approach for high resolution images using the improved FCN model. Compared with the object-oriented method and two typical deep learning-based approaches, the classification accuracy is obviously improved. In the following sections, we will discuss the reasons.

4.1. MR-SVM vs. Our Approach

Most of the traditional object-oriented classification approaches employ their classification in a "segmentation-classification" manner. In an ideal segmentation, each segment represents a single ground object. In other words, an ideal image object is neither over-covered nor over-segmented. However, most of the segmentation was conducted in an unsupervised way, which relies only on image information, but no prior class information. When the spectral and geometric features are similar, it is difficult to obtain high-quality image objects. Once the image objects are incorrect, subsequent object-oriented classification cannot lead to an accurate result. For an image, it is difficult to find universal segmentation parameters so that all image objects can be correctly generated. Figure 13 shows one image object (with a yellow boundary) generated by MR segmentation that incorrectly covers both building and cement ground.

In the classification stage, it is very difficult to choose expressive features for an image object as the input of the classifier. The feature selection usually needs many attempts and largely depends on experience. Therefore, the uncertainty introduced by the two stages, together affects the final classification accuracy.

In our FCN-based approach, the class information, which is the ultimate objective for classification, is taken as the supervisory signal that controls the whole process including both feature extraction and classification. Our approach combines the segmentation and classification stages, and achieves high quality classification in an end-to-end way. This is also the most obvious advantage of the deep learning theory.

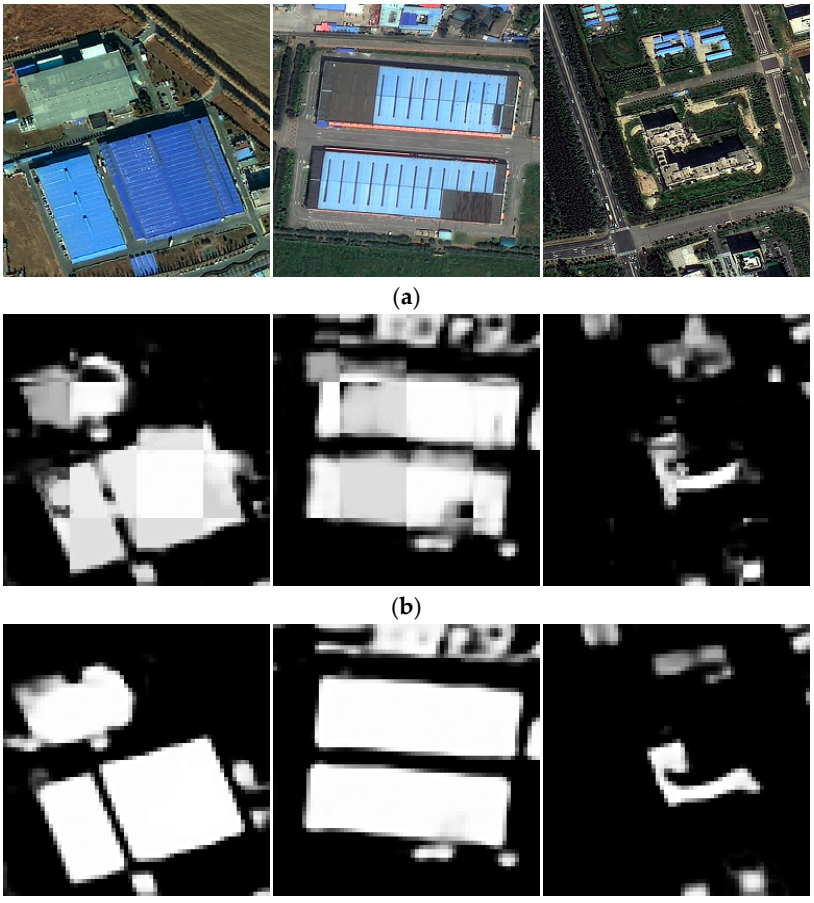


(**a**) (**b**) (**c**)

Figure 13. Incorrect image object generated by MR segmentation. (**a**) Original images; (**b**) GT labels corresponding to the images in (**a**); (**c**) Incorrect image object covers both the building and cement ground (with yellow boundary).

4.2. Patch-Based CNN vs. Our Approach

In the patch-based CNN approach, each image patch is input to the model independently, which means that only the "intra-patch" context information is considered. However, correlations between patches are not taken into account, which might lead to obvious gaps between patches. Especially for objects with strong continuity, such as road and building edges, the problem is more serious. Figure 14 shows the differences between patch-based CNN and our approach for building heat map generation.



(**a**)

(**b**)

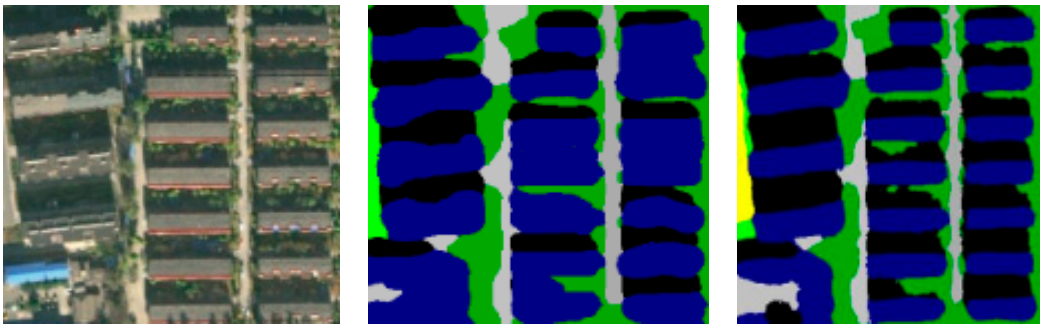
(**c**)

Figure 14. Heat map for the building generated by patch-based CNN and our approach. (**a**) Original images; (**b**) Heat map generated by patch-based CNN classification using 128 × 128 patches; (**c**) Heat map generated by the FCN model.

Compared with the patch-based approaches, our model takes the whole image as the input, and performs the classification in a single-loop manner, which considers the context information overall and seamlessly. Our model eliminates the discontinuities at the patch boundaries. This is also the most remarkable advantage of FCN.

4.3. FCN-8s vs. Our Approach

FCN model is a *convolutionalized* version of standard CNN through a simple modification. The most significant feature of the FCN model is: on the one hand, FCN inherits the high accuracy feature for image-label classification from standard CNN. On the other hand, it maintains the 2-D spatial information of the input image, thus achieving dense class prediction. However, pooling operations cause serious reduction of the resolution. The output is not fine enough, which will result in the loss of valuable detail information. As can be seen from Figure 15, our approach outperforms FCN-8s in terms of detail preserving. Therefore, the classification accuracy is greatly improved.



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

Figure 15. Detail comparison between FCN-8s and our approach. (**a**) Original images; (**b**) Classification result from FCN-8s; (**c**) Classification result from our approach.

As the most accurate model in the FCN family, FCN-8s combines the feature maps with different resolutions from different pooling stages, to obtain a more intensive class prediction. In FCN models, the lost resolution is compensated by the deconvolution operation. However, deconvolution is difficult for efficiently restoring the resolution by way of learning. Benefiting from the "atrous" convolution, the resolution of the feature map is maintained naturally in our approach. Besides, FCN models do not consider the relationship between pixels, ignoring the spatial regularization that is commonly employed in remote sensing image analysis. In our approach, the relationship between pixels is taken into account by CRF-based post-processing. The class map predicted by FCN is further refined, and the accuracy is therefore improved.

5. Conclusions

This paper presents a classification approach for high resolution images using an improved FCN model. Compared with the object-oriented method and two typical deep learning-based approaches, the classification accuracy is obviously improved.

Our FCN-based classification combines the segmentation and classification stages, taking the class accuracy as the only constraint, and achieves high quality classification in an end-to-end way. The GT classes of ground objects are taken as the supervised information that guides both the feature extraction and the region generation. The classification results of using "atrous" convolution and CRF-based post-processing allows us to obtain a high resolution class prediction. In addition, due to the use of a multi-scale model, the model trained from the GF-2 images also has high classification

accuracy on the IKONOS images. It is proven that our approach has a strong applicability for images with different resolutions.

The main limitation of our approach is that it needs a large number of high quality GT-labels for the model training, which relies on professional interpretation experiences and lots of manual work. Therefore, the main aspect of our future work is training the model in a weak supervision way, to further enhance its applicability.

Acknowledgments: This work was jointly supported by the National Natural Science Foundation of China (Grant No. 41571414. Title: Eco-environment assessment of Yanhe watershed based on temporal-spatial entropy), Beijing Municipal Science and Technology Project (Grant No. Z161100001116102. Title: Research on Remote Sensing Technology in Soil and Water Conservation and Demonstrative Application in Beijing), and the Fundamental Research Project in China Institute of Water Resources and Hydropower Research (Grant No. JZ0145B2017. Title: Research on Spatial-Temporal Variable Source Runoff Model and Mechanism).

Author Contributions: Gang Fu and Changjun Liu proposed and designed the technique roadmap, and performed the programming works; Gang Fu and Tao Sun designed and performed the experiments; Rong Zhou, Tao Sun, and Qijian Zhang collected the training and testing image data, and analyzed the experimental results.

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

References

1. MacQueen, J.B. Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability; University of California Press: Berkeley, CA, USA, 1967.; pp. 281–297.
2. Miller, D.M.; Kaminsky, E.J.; Rana, S. Neural network classification of remote-sensing data. *Comput. Geosci.* **1995**, *21*, 377–386.
3. Mas, J.; Flores, J. The application of artificial neural networks to the analysis of remotely sensed data. *Int. J. Remote Sens.* **2008**, *29*, 617–663.
4. Camps-Valls, G.; Bruzzone, L. Kernel-based methods for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 1351–1362.
5. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 247–259.
6. Yuan, Y.; Lin, J.; Wang, Q. Hyperspectral Image Classification via Multitask Joint Sparse Representation and Stepwise MRF Optimization. *IEEE Trans. Cybern.* **2016**, *46*, 2966–2977.
7. Wang, Q.; Lin, J.; Yuan, Y. Salient Band Selection for Hyperspectral Image Classification via Manifold Ranking. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1279–1289.
8. Walter, V. Object-based classification of remote sensing data for change detection. *ISPRS J. Photogramm. Remote Sens.* **2004**, *58*, 225–238.
9. Definients Image. *eCognition User's Guide 4*; Definients Image: Bernhard, Germany, 2004.
10. Feature Extraction Module Version 4.6. In *ENVI Feature Extraction Module User's Guide*; ITT Corporation: Boulder, CO, USA, 2008.
11. Baatz, M.; Schäpe, A. Multiresolution Segmentation: An Optimization Approach for High Quality Multi-scale Image Segmentation. In *Angewandte Geographische Information Sverarbeitung XII*; Herbert Wichmann Verlag: Heidelberg, Germany, 2000; pp. 12–23.
12. Robinson, D.J.; Redding, N.J.; Crisp, D.J. *Implementation of a Fast Algorithm for Segmenting SAR Imagery*; DSTO Electronics and Surveillance Research Laboratory: Edinburgh, Australia, 2002.
13. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pat.* **1995**, *17*, 790–799.
14. Fu, G.; Zhao, H.; Li, C.; Shi, L. Segmentation for High-Resolution Optical Remote Sensing Imagery Using Improved Quadtree and Region Adjacency Graph Technique. *Remote Sens.* **2013**, *5*, 3259–3279.
15. Hinton, G.; Osindero, S.; Welling, M.; Teh, Y.-W. Unsupervised Discovery of Nonlinear Structure Using Contrastive Backpropagation. *Science* **2006**, *30*, 725–732.
16. Deep Learning. Available online: https://en.wikipedia.org/wiki/Deep_learning (accessed on 3 May 2017).
17. Hinton, G.E.; Osindero, S.; Teh, Y.W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554.

18. Convolutional Neural Networks (LeNet)—DeepLearning 0.1 Documentation. DeepLearning 0.1. LISA Lab. Retrieved 31 August 2013. Available online: http://deeplearning.net/tutorial/lenet.html (accessed on 5 May 2017).
19. Graves, A.; Liwicki, M.; Fernandez, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 855–868.
20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Neural Information Processing Systems (NIPS) Conference, La Jolla, CA, USA, 3–8 December 2012.
21. Ciresan, D.; Meier, U.; Schmidhuber, J. Multi-column deep neural networks for image classification. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3642–3649.
22. Nguyen, T.; Han, J.; Park, D.C. Satellite image classification using convolutional learning. In Proceedings of the AIP Conference, Albuquerque, NM, USA, 7–10 October 2013; pp. 2237–2240.
23. Wang, J.; Song, J.; Chen, M.; Yang, Z. Road network extraction: A neural-dynamic framework based on deep learning and a finite state machine. *Int. J. Remote Sens.* **2015**, *36*, 3144–3169.
24. Castelluccio, M.; Poggi, G.; Sansone, C.; Verdoliva, L. Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. *arXiv*, **2015**, arXiv:1508.00092.
25. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sens.* **2015**, *7*, 14680–14707.
26. Zhou, W.; Newsam, S.; Li, C.; Shao, Z. Learning Low Dimensional Convolutional Neural Networks for High-Resolution Remote Sensing Image Retrieval. *arXiv*, **2016**, arXiv:1610.03023.
27. Mnih, V. Machine Learning for Aerial Image Labeling. Ph.D. Thesis, University of Toronto, Toronto, ON, Canada, 2013.
28. Lagkvist, M.; Kiselev, A.; Alirezaie, M.; Loutfi, A. Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks. *Remote Sens.* **2016**, *8*, 329.
29. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
30. Chen, L.C.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
31. Paisitkriangkrai, S.; Sherrah, J.; Janney, P.; Van den Hengel, A. Effective Semantic Pixel labelling with Convolutional Networks and Conditional Random Fields. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
32. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineety, V.; Su, Z. Conditional Random Fields as Recurrent Neural Networks. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
33. Sherrah, J. Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery. *arXiv*, **2016**, arXiv:1606.02585.
34. Marmanis, D.; Wegner, J.D.; Galliani, S.; Schindler, K.; Datcu, M.; Stilla, U. Semantic segmentation of aerial images with an ensemble of CNSS. In Proceedings of the ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic, 12–19 July 2016.
35. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Fully Convolutional Neural Networks for Remote Sensing Image Classification. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 5071–5074.
36. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional Neural Networks for Large-Scale Remote Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *2*, 645–657.
37. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. High-Resolution Semantic Labeling with Convolutional Neural Networks. *arXiv*, **2016**, arXiv:1611.01962.
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv*, **2014**, arXiv:1409.1556.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv*, **2015**, arXiv:1512.03385.

40. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv*, **2016**, arXiv:1602.07261.
41. Bertasius, G.; Shi, J.; Torresani, L. Deepedge: A multiscale bifurcated deep network for top-down contour detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
42. Xie, S.; Tu, Z. Holistically-Nested Edge Detection. *arXiv*, **2015**, arXiv:1504.06375.
43. Krahenbuhl, P.; Koltun, V. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS), Granada, Spain, 12–15 December 2011.
44. Nussbaum, S.; Niemeyer, I.; Canty, M.J. SEATH—A new tool for automated feature extraction in the context of object-based image analysis. In Proceedings of the 1st International Conference on Object-Based Image Analysis, Salzburg, Austria, 4–5 July 2006. XXXVI-4/C42.
45. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*.