Multiple Object Extraction of Remote Sensing Images Based on Convolutional Neural Networks and Support Vector Machines

Tao-xing FENG, Wei-yu WANG, Xiao-meng QIAN, Zhan-hong ZHANG, Hui LIU, Ying XING and Bin YANG

Beijing University of Posts and Telecommunications, Beijing 100876, China

Keywords: Machine learning, Deep learning, Convolutional neural network (CNN), Support vector machine (SVM), Remote sensing images, Multiple object extraction.

Abstract. The data provided by remote sensing technology is characterized by wide coverage, high real-time performance and a supply of rich and objective information. Therefore, extracting geo-objects such as roads from remote sensing images plays important roles in many urban applications and used in a wide range of areas. However, in the past, such tasks generally performed manually, which is very costly and time-consuming. And many of existing automating attempts are designed for specific features and classifiers with limitation. This paper proposes a CNN-SVM-based road extraction system (a convolutional neural network with three support vector machines), which takes raw pixel values in aerial imagery as inputs and predicted three-channel label images (background-buildings-roads) as outputs. By training a single CNN efficiently, feature extractors and classifiers are automatically constructed and multiple kinds of objects are extracted simultaneously. Using SVM, the classification results can be optimized. At the same time, we improve our experimental performance by the Dropout method and predicting with spatial displacement. Finally, comparing our model with previous methods, we get the result that the precision of CNN-SVM increases 8.37%.

Introduction

Extraction of multiple urban objects like roads and buildings from aerial imagery has many applications in military and civilian fields, such as aerospace, military reconnaissance, disaster monitoring and forecasting, land planning and utilization, etc. However, due to the various shapes of those objects and some interference factors, accurate pixel labeling of the remote sensing images is still a task of high research value. Thus, effective extraction of buildings and roads is highly demanded, and there have been plenty of attempts utilizing different methods.

Objects extraction methods can be classified into two categories, which are semi-automatic and fully automatic [1]. The former requires human interaction while the latter does not [2]. Information extraction from satellite images done by semi-automatic methods is usually expensive and time-consuming. To overcome these limitations, automatic methods are explored and applied.

Based on various features and classifiers, numerous perspectives about automatic methods were proposed: In Singh, PP et al (2014), assisted by a weighted membership function (W-mf), the fuzzy-C-means (FCM) technique successfully enhanced the classification results and detected the objects [3]. Using two object-based filters and support vector machine (SVM), Zeling Miao et al (2015) computed object features to select road candidates and then extracted the road class [4]. In Wang JH et al (2016), they utilized a knowledge-based method to extract the spatial texture feature for image segmentation and worked out more proper features to construct the model [5]. In 2017, Kumar KM et al proposed a new approach using particle filter and novel hybrid multi-kernel partial least squares (PLS) [6].

Among those automatic methods, SVM is an effective classification method. It has the pretty good robustness to intensity changes when road materials are similar but different [4]. What’s more, in contrast with other competing methods, SVM has equivalent or superior performance in most cases [7].

Although these traditional automatic methods have almost achieved accurate extraction of terrestrial objects from remote sensing imagery, only one specific kind of object can be extracted [8].
In actual application, contrasted with multiple objects, a single object can provide less information
and the reference value. Therefore, we put the emphasis on the simultaneous extraction of multiple
objects. In recent years, approaches exploiting convolutional neural network (CNN) have been
proposed [9][10][11] and applied in solving multi-objective extraction problems with pretty good
effects. Such method can divide images into patches and get good feature extractors from raw pixel
values without complex pre-processing. In 2016, Shunta Saito et al generated three-channel maps
from raw remote sensing imagery input. They took advantage of detecting road and building pixels
from remote sensing imagery considering multiple labels synchronously and a new output function
called channel-wise inhibited Softmax to train the CNN [8]. On the basis of the previous work, Rasha
Alshehhi et al (2017) combined the convolutional neural network (CNN) features with several low-
level features of roads and buildings in the post-processing section so as to smooth the irregular and
disjoint regions [12].

In the thesis, we combine the high speed and multiple object extraction advantages of the
convolutional neural network with the great classification results of SVM to construct the model.
Following the previous work, we use support vector machine (SVM) to train the CNN features and
proposed the Dropout method and predicting with spatial displacement to improve performance. Our
experiments were conducted on Massachusetts road aerial imagery data sets and the results showed
that our proposed methods outperformed the previous achievements. Our methods avoided the
complicated feature extraction and multifarious design of classifiers which independently trains
objects to be extracted. And pixels in our chosen images were more accurately classified into
background, buildings, and roads. In other words, in our experiments, three-channel label images
could be more exactly constructed.

The rest of this article is organized as follows. The second and third section introduces the public
data set we utilized and the environment configuration including hardware part and software part.
The fourth section presents the basic knowledge of convolution neural networks and support vector
Machine. And then, we propose the methods used in this article in details in the fifth section. The
sixth section is a description of the experiment including training, prediction, and evaluation. The
seventh section is the discussion of the SVM classification strategy and the superiority of our model
compared with other three models. In the final section, we present a conclusion for our important
findings.

**Data Sets**

We obtained publicly available data proposed by Mnih on the website:
http://www.cs.toronto.edu/~vmnih/data/. Merging data from Massachusetts Buildings Dataset and
Massachusetts Roads Dataset, we created Massachusetts Buildings and Roads dataset. We took
building, road, and background as the three channels. Background labels were created by calculating
the XOR of building and road label images. The size of all images in this dataset are 1500×1500 in
size and the resolution is 1m²/pixel.

**Convolution Neural Network and Support Vector Machine**

Convolution neural network and support vector machine are most essential parts of our extractor.
Therefore, we give the details of these two parts and the basic architecture we used in this article as
follows.

**Basic Theory of CNN**

We present CNN from three basic parts: convolution layers, pooling layers, and fully connected
layers.

**Convolutional Layer.** A convolutional layer of CNN usually contains several feature maps. Each
feature map consists of some rectangularly arranged neurons and the neurons in the same feature map
share the weights. The shared weights here are convolution kernel, also called a filter. The
convolution operation uses a convolution kernel to convolve with the corresponding region of the image to obtain a value, and then continuously moves the convolution kernel to complete the convolution of the entire image. After convolution operation, each filter will get the image of corresponding extracted features. And then, we need to input these features into an activation function to get the final output. In this article, we use function ReLU, which has the expression: $f(x) = \max(0, x)$. The advantages of ReLU are fast convergence and simple gradient.

**Pooling.** Conventional CNN is a continuous convolution operation and pooling is an important step to reduce parameters to make feature extraction more concise. The most common methods of pooling are max-pooling and mean-pooling. We adopt the former one in this article. The extracted features are treated as a matrix and several non-overlapping regions are divided on this matrix. We calculate the maximum of the features in each region and those values are used to participate in the subsequent training. Taking the maximum value represents that we retain only the strongest of these features and leave other weak features of this type.

**Fully Connected Layer.** Fully connected layers connect their all nodes to the nodes in the previous layer and map the learned features to the sample markup space.

**Basic Theory of SVM**

The purpose of SVM is to find a hyperplane that divides the sample into two categories with the largest interval. We use $\omega$ to represent the coefficient of the hyperplane which we aim to find. It can be represented by Eq.1.

$$\max \frac{1}{||\omega||}, \text{s.t., } y_i(\omega^T x_i + b) \geq 1, i = 1, \ldots, n. \tag{1}$$

And the schematic diagram of SVM is presented as Figure 1.

![SVM Diagram](image)

**Figure 1. Schematic diagram of SVM.**

Since SVM itself is a binary classifier, we should classify one type of samples into the same class and the remaining ones into another class. Therefore, when there are $k$ types of samples, we need to construct $k$ SVMs.
Basic Architecture

Figure 2. Basic architecture of this article.

Figure 2 shows the base architecture we use in this article. Our architecture following the characteristic of CNN in which convolutional layers and spatial polling layers are stacked with fully connected layers followed. Then we input the features extracted by CNN into three SVM classifiers. There are five layers containing trainable parameters. We take a 64*64-sized three-channel RGB aerial imagery patch as the input and a 768-dimensional vector as the output. Then we reshape the output into a 16*16-sized three-channel patch made up of buildings-roads-background channels. We assume that C (a, b* b/c) is a convolutional layer with b*b-sized filters and the convolution stride c, P(a/b) is an a*a max pooling layer with stride b, and FC(a) is a fully connected layer with a units. In this way, the architecture can be interpreted as C (64, 16*16/4)-P (2/1)-C (112, 4*4/1)-C (80, 3*3/1)-FC (4096)-FC (768).

Methodology

By training our CNN, we can directly learn a mapping from raw pixels in an input aerial image S to a true label image \( \tilde{M} \). And we aim to predict a multi-channel label image \( \tilde{M} \) from S. In this article, a label image consists of three channels including buildings, roads, and background. And we map these three channels to RGB channels (R-roads, G-buildings, B-background) so that each pixel on the label image is a 3-dimensional vector. Since each pixel should always be either background, buildings or roads in a label image, the sum over all elements of a pixel vector should always be 1. Figure 3 is an example.

![input image](image1.png) ![true label image](image2.png)

Figure 3. Example of an input image and its true label image.

Patch-based Formulation

Our pixel labeling method is similar to the method have been proposed by Mnih et al. [1] We use a \( w_8 \) * \( w_8 \)-sized aerial imagery patch \( s \) to obtain a \( w_m \) * \( w_m \)-sized true label patch \( \tilde{m} \) by training the CNN and use \( \tilde{m} \) to denote the predicted patch. We describe pixel located at \( i \) in \( \tilde{m} \) as a 3-dimensional one-hot vector, \( \tilde{m}_i = [\tilde{m}_{i1}, \tilde{m}_{i2}, \tilde{m}_{i3}] \). In a predicted label patch \( \hat{m} \), each pixel at \( i \) is also a 3-dimensional vector \( [\hat{m}_{i1}, \hat{m}_{i2}, \hat{m}_{i3}] \). In this situation, when a corresponding aerial imagery patch is
given, all pixels in a true label patch $\tilde{m}_t(i=1, \ldots, w_m^2)$ are conditionally independent of each other. Therefore, the posterior of a true label patch can be expressed as Eq.2.

$$p(\tilde{m}|s) = \prod_{i=1}^{w_m^2} p(\tilde{m}_i|s).$$

(2)

The loss function can be described as Eq.3.

$$L = -\sum_{i=1}^{w_m^2} \ln p(\tilde{m}_i|s).$$

(3)

Figure 4 shows the input and output patches. Concentrating on a small-region patch, it is so abstract that we cannot recognize what it is. Therefore, we consider a wider region using context information to help us to predict labels. In this way, we can find that the patch shows a part of a building. Based on context consideration, the size of an input patch $w_s$ is set larger than the size of a predicted label patch $w_m$. This technique is also implemented to have better performance by Mnih et al. [14]

![Figure 4. Input and output patches.](image)

**Channel-wise Inhibited Softmax [8]**

In this article, $w_s = 64$, $w_m = 16$. We reshape a 768-dimensional vector to a $16 \times 16 \times 3$-sized image patch. We use $x_i = [x_{i1}, x_{i2}, x_{i3}]^T$ to denote the $i$th pixel of the output patch. The Softmax is defined as Eq.4.

$$\tilde{m}_{ik} = \frac{\exp(x_{ik})}{\sum_j \exp(x_{ij})}.$$  (4)

We convert $x_i$ to the label probability vector $\tilde{m}_i = [\tilde{m}_{i1}, \tilde{m}_{i2}, \tilde{m}_{i3}]^T$ as Eq.5.

$$\tilde{m}_{ik} = \frac{\exp(c_k x_{ik})}{\sum_j \exp(c_j x_{ij})}, \quad c_k = \begin{cases} 0, & \text{if } k = 1, \\ 1, & \text{otherwise.} \end{cases}$$  (5)

In the city, the number of background pixels is much smaller than the number of pixels of buildings and roads. Therefore, we set $k = 1$, $c_k = 0$ to eliminate the influence of the background.

**CNN-SVM**

Feature extraction plays an important role in classification. The features we need to extract should effectively distinguish between different classes. For SVM, using traditional methods to extract features is a difficult and time-consuming task. And the trained CNN network can simply and efficiently extract the advanced features that are not able to be described semantically. Therefore, we envisage improving the performance of the model by combining CNN and SVM together. First, we train a CNN model. Then, we replace the CNN's output layer with three SVMs with the CNN model remaining unchanged. After that, we use the features and labels extracted by CNN to fit these three SVMs. When we use CNN-SVM to predict remote sensing images, we can use appropriate strategies to adjust the precision and recall.

**Dropout**

We use Dropout to effectively prevent overfitting. In the process of network training in deep learning, the neural network units are discarded from the network temporarily in a certain probability. For stochastic gradient descent, each mini-batch is trained on a different network because of the random
dropping. In this way, the renewal of the weights no longer depends on the cooperation of implicit nodes with fixed relations, preventing the certain features from being effective only under other specific features.

**Predicting with Spatial Displacement**

Figure 5 shows the spatial displacement we use in predicting. We displace the original input aerial imagery patches for seven times with one pixel each time. Then, the predicted label patches of those versions will have the same displacement. We tile those predicted label patches together and divide all pixel values by eight to get an average.

![Figure 5. Spatial displacement in prediction.](image)

For the predicted label patches, this method can play an important role in smoothing the outputs over the boundaries and can significantly improve the performance. Figure 6 shows its effectiveness.

![Figure 6. Comparison chart of using spatial displacement.](image)

**Experiment**

**Environment Configuration**

**Hardware Configuration.** CPU: 2 Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10GHz, which has 12 cores.

GPU: NVIDIA Corporation GK106GL [Quadro K4000], which has 3017MiB memory.

Hard Disk: Crucial_CT512MX100SSD1, which has 512 GB capacity.

**Software Configuration.** Anaconda3(python3.6): Cython; Chainer 1.5.0.2; NumPy; Tqdm; Lmdb; OpenCV2; OpenCV 3.1.0; Boost 1.59.0; Boost.NumPy (26aa5b).

**Training**
We train Mnih-CNN, Mnih-CNN with CIS (Channel-wise Inhibited Softmax) [3][4], Mnih-CNN-SVM and Mnih-CNN-SVM with CIS four models to show the effectiveness of our CNN-SVM.

During training, we adopt mini-batch stochastic gradient descent method with momentum. In the learning stage, the hyper-parameters are the mini-batch size, the learning rate (LR) $\eta$, the LR reducing rate $\gamma$, the LR reducing frequency $\tau$, a weight of the momentum term $\alpha$, and a weight of the L2 weight decay $\beta$. The values we used for all experiments in this article are chosen based on Ref.4 as follows: $\eta = 0.0005$, $\tau = 104$, $\gamma = 0.1$, $\alpha = 0.9$, $\beta = 0.0005$, with the mini-batch size equals to 128.

We train SVMs with extracted features from the trained CNN. We design multi-classification SVM to classify background, buildings, and roads and adopt the one-versus-rest method. The training sets are drawn as:

1. Take the vector corresponding to the background is as the positive set, and the vector corresponding to buildings and the road as the negative set;
2. Take the vector corresponding to buildings as the positive set, and the vector corresponding to the background and roads as the negative set;
3. Take the vector corresponding to the road as the positive set, and the vector corresponding to the background and buildings as the negative set.

Using these three training sets, we get three SVMs. For prediction, we reshape the three prediction results to an n x 3 matrix. After that, we input mini-batch labels and features extracted from trained models Mnih-CNN and Mnih-CNN with CIS to fit three SVM separately.

**Prediction**

To demonstrate the effectiveness of our CNN-SVM, we predict 10 remote sensing images of the test set with Mnih-CNN, Mnih-CNN with CIS [8], Mnih-CNN-SVM and Mnih-CNN-SVM with CIS four model. Then, we make background, building and road, components of the prediction results correspond to B, G, and R, components of RGB, respectively, and saving them into one image, such as Figure 7.

![Figure 7. Example of prediction results.](image)

**Evaluation**

The We use precision and recall to evaluate the extraction results. Precision is the ratio of the number of buildings or roads pixels in true label images to the number of those in the predicted label images. Recall is the ratio of the predicted pixels to the true pixels. Here is a comparison of precision and recall of the four models.
Table 1. Precision and recall on the test dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Building Channel</th>
<th>Road Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Precision</td>
<td>Average Recall</td>
</tr>
<tr>
<td>Mnih-CNN</td>
<td>0.90004357</td>
<td>0.90021818</td>
</tr>
<tr>
<td>Mnih-CNN with CIS</td>
<td>0.89444072</td>
<td>0.89417839</td>
</tr>
<tr>
<td>Mnih-CNN-SVM</td>
<td>0.8987596</td>
<td>0.87341092</td>
</tr>
<tr>
<td>Mnih-CNN-SVM with CIS</td>
<td>0.89154397</td>
<td>0.89168285</td>
</tr>
</tbody>
</table>

Figure 8 and Figure 9 shows the precision-recall curve of Mnih-CNN and Mnih-CNN with CIS.

![Figure 8. Precision-recall curve of Mnih-CNN.](image1)

![Figure 9. Precision-recall curve of Mnih-CNN with CIS.](image2)
In Figure 10, Mnih-CNN and Mnih-CNN with CIS points represent the precision of the Mnih-CNN model and the Mnih-CNN with CIS model after 400 epochs. After 11 iterations of our Mnih-CNN-SVM model and Mnih-CNN-SVM with CIS model, we evaluated them with a test set and get two corresponding curves. Comparing the curves and points, it is evident that classifying features extracted by Mnih-CNN-SVM with CIS model can improve precision.

![Figure 10. Precision comparison of 4 models.](image)

**Discussion**

Table 1 and Figure 10 shows the prediction probabilities of the building channel and the road channel of the four models respectively. It can be seen that our model (CNN-SVM with CIS) has the best performance in prediction. We predicting with spatial displacement in order to smooth the outputs over the boundaries and significantly improve the performance of CNN. And the better performance of our SVM is related to our classification strategies, which is used to deal with controversial predicted pixels.

The controversial predicted pixel may be with the vector \( \hat{m}_{13}, \hat{m}_{12}, \hat{m}_{15} \) mapped to [1,1,1] because a pixel cannot belong to three classes (background, buildings, and roads) at the same time. Since only the background class may have the same features with buildings and roads, for the case of the prediction result [1, 1, 1], we assume that this pixel belongs to a road class, which means that the prediction result is changed to [0, 0, 1].

Pixels with vector \( \hat{m} \) equal to [0, 0, 0] are also controversial, because a pixel must belong to one of the three classes (background, buildings, and roads). Since the background has no significant features, we classify such pixels into the background class.

We also the prediction results [1, 1, 0], [1, 0, 1] and [0, 1, 1] into our consideration. Both of the first two results contain a background class. Owing to the interference of the background, we decide the prediction results to be [0, 1, 0] and [0, 0, 1]. Actually, if there is a background class in the context, the building class will also interfere with the detection of roads, so we take the third prediction result as [0, 0, 1].

With other prediction results unchanged, the prediction strategy is shown in Table 2.
Table 2. SVM Classification strategy.

<table>
<thead>
<tr>
<th>SVMs prediction results</th>
<th>Final prediction results</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,0,0],[0,0,0]</td>
<td>[1,0,0]</td>
<td>Background</td>
</tr>
<tr>
<td>[0,1,0],[1,1,0]</td>
<td>[0,1,0]</td>
<td>Buildings</td>
</tr>
<tr>
<td>[0,0,1],[1,1,1],[1,0,1],[0,1,1]</td>
<td>[0,0,1]</td>
<td>Roads</td>
</tr>
</tbody>
</table>

Conclusion and Future Work

In this article, we propose a CNN-SVM model for extracting the road from remote imagery. We study the difference between the prediction results and the truth labels to develop a proper conversion strategy, which can effectively solve the interference problems between background, buildings, and roads and significantly improve the performance of the model. For specific needs, the conversion strategy can help to get better precision. In the future, we will improve our CNN-SVM model to identify the context of the picture and change the conversion strategy according to the semantics, so that we will have a better prediction. Finally, we implemented our models with a new and flexible deep learning framework, Chainer. And we will show our datasets and the codes of our methods and experiments at https://github.com/nattrueSwitch/CNN-SVM.

Acknowledgement

This research was financially supported by Research Innovation Fund for College Students of Beijing University of Posts and Telecommunications, the National Natural Science Foundation of China (No. 61702044), and the Fundamental Research Funds for the Central Universities (No.2017RC27).

References


