Vehicle Detection in Satellite Images by Incorporating Objectness and Convolutional Neural Network

Shenquan Qu, Ying Wang, Gaofeng Meng, and Chunhong Pan
Institute of Automation, Chinese Academy of Sciences, Beijing, China
Email:{shenquan.qu, ywang, gfmeng, chpan}@ nlpr.ia.ac.cn

***Abstract*—Automatic vehicle detection from high-resolution remote sensing images plays a fundamental role in a wide range of applications. Various approaches have been proposed to address this issue in decades, however a fast and robust approach has not yet been found. It is still a very challenging task, due to the complex background, diverse colors and occlusions caused by buildings and trees. Traditional methods suffer from either high time complexity or low accuracy rate. To overcome the above shortcomings and consider the aforementioned difficulties, in this paper, we propose a simple and efficient approach to detect vehicles automatically. The proposed model lays emphasis on both speed and accuracy of vehicle detection, thus our proposed model consists of two stages: (1) To speed up localization, we apply the newly proposed Binary Normed Gradients (BING) to extract region proposals. (2) To enhance the robustness and improve the accuracy rate, we use Convolutional Neural Network (CNN), which combines feature extraction and classification. Therefore we modify the BING and design our own architecture of CNN to solve our problem. By comparing with start-of-the-art methods in extensive experiments, we demonstrate the effectiveness of the proposed approach in both speed and accuracy. Specifically, our method is more than 10 times faster than traditional methods, and our average accuracy is higher than state-of-the-art methods.**

***Index Terms*—vehicle detection, Binary Normed Gradients (BING), Convolutional Neural Network (CNN), objectness**

I. Introduction

Due to the recent advances in sensor technology, high-resolution images are accessible easily. The analysis of high-resolution images has become an important research area [1]-[3], with a wide variety of applications, such as image understanding, object detection and image classification. Detecting small objects, such as vehicles, aircrafts and ships etc., is a well-known challenging task in high-resolution images. Vehicle detection, as an active research area, has been widely used in military surveillance, intelligent traffic system, maritime search and rescue. Although various approaches [1]-[5] attempt to solve this problem, there is no widely recognized solution to the problem. The difficulties mainly lie in three aspects: the diversity of colors and shapes for different vehicles, complex background and occlusions caused by buildings and trees.

Generally speaking, the existing approaches mainly consist of three stages: object location, feature extraction and object classification. Various object location methods have been applied to vehicles detection. T. Zhao *et al.* [6] employed the prior that vehicles are on the road. Thus they used the Canny operator to detect straight lines of road firstly. Then they located the road of the image, finally they detected vehicles on the road. H. Zheng *et al.* [1] presented a threshold segmentation method which is based on morphological operations, namely grey-scale top-hat and bottom-hat transforms. This method provides a convenient way for detecting vehicles, while it fails when the vehicles are occluded by buildings and trees. X. Y. Chen *et al.* [2] located vehicles with sliding a window based method. This method greatly improves the accuracy of vehicle detection, while the sliding window based method is time-consuming and it is hard to generalize to the complex background.

Similarly, various feature extraction methods are used in vehicle detection, such as Haar-like wavelets, Scale Invariant Feature Transform (SIFT) [7], Histogram of Oriented Gradients (HOG) [8], Local Binary Pattern (LBP) [9] or their combinations. For instance, hierarchical 3D-model was used by Hinz [3] to describe the prominent geometric features of cars. P. Liang *et al.* [10] combined HOG and Haar descriptors in the Generalized Multiple Kernel Learning (GMKL) framework, in which trade-off between HOG and Haar descriptors were learned by constructing an optimal kernel with many basis kernels. Kembhavi *et al.* [11] proposed a model based on multi-scale HOG features. Thus it can effectively detect vehicles in different sizes and scales. All the above features have achieved great success in object detection of nature images. However aerial images have some differences with natural images. For instance, satellite images have lower resolution, lower color contrast and more noise. Meanwhile, those features are all handcrafted and not specifically for the problem of vehicle detection, thus they ignore the specific features of vehicles.

Traditional classification approaches include Support Vector Machine (SVM), boosting and CNN, which are

Manuscript received January 10, 2015; revised November 12, 2015.

doi: 10.18178/jiii.4.2.158-162

the state-of-the-art classifiers. They are widely used in vehicle detection. Chen *et al*. [4] employed SVM to detect vehicles on road. Grabner [5] proposed a robust boosting-based system for car detection from aerial images. Chen *et al*. [2] showed that a CNN can achieve a higher accuracy of vehicle detection dramatically. The performance of classifiers heavily depend on what features they use, no matter which classifier is chose.

Both speed and accuracy are very important for practical application like object detection in satellite images. As vehicle detection requires localizing objects within an image, a commonly used approach which has been used for several decades is the sliding-window based detector [2]. This method is not practical since it is time-consuming. As mentioned above, handcrafted features like SIFT, HOG, LBP can't reach an optimal balance between the discriminability and the robustness without considering the details of real data. To overcome the disadvantages of the classical methods, we propose a new framework for vehicle detection considering both speed and accuracy. In this way, it consists of two stages: the first is to localize region proposals, the second is vehicle classification. Many recent researchers provide methods for generating category-independent region proposals. Some examples like: BING [12], selective search [13], category-independent object proposals [14]. Considering both speed and accuracy, we use BING to speed up the stage of localization. The main motivation of BING is that generic object with well-defined closed boundaries share surprisingly strong correlation when looking at the norm of the gradient, after resizing their corresponding image windows to small fixed size (e.g. 8 × 8). Our experimental results show the advantages of BING. It not only speeds up the stage of extracting region proposals, but also provides benefits to CNN training by offering many hard samples, which can enhance the performance of CNN. Furthermore, CNN has been used to learn rich features from satellite images automatically, which has yielded superior performance in many object recognition tasks. As a feature learning architecture, CNN combines extracted features and classification. For handwritten zip code recognition, LeCun *et al.* [15] showed that stochastic gradient descent via back propagation was effective for training CNN. Then in document recognition, LeCun *et al.* [16] gave the classical architecture of CNN called LetNet. CNN is specifically designed to deal with the variability of two dimensional (2-D) shapes by extracting local features that only depend on small sub-regions of the image. As satellite images are 2-D shapes, so CNN is chosen to deal with them. Recently, there are many frameworks for deep learning. Caffe [17] is a deep learning framework developed with cleanliness, readability, and speed in mind. Experiments show that our method dramatically reduces the time required and increases the accuracy by using BING and Caffe in CUDA mode.

The remainder of this paper is organized as follows. In Section 2, we provide more detailed descriptions on our approach. Section 3 shows experimental evaluation and analysis of our method on our own dataset, which demonstrates that our method outperforms the state-of-the-art approaches in vehicle detection. Finally, we conclude in Section 4.

II. Module Design

In this section, we introduce the details of the two stages in our framework mentioned above. The first stage generates category-independent region proposals. These proposals are the input data for the next stage. Then the second stage uses CNN to decide which proposals are vehicles. Fig. 1 presents an overview of our method.

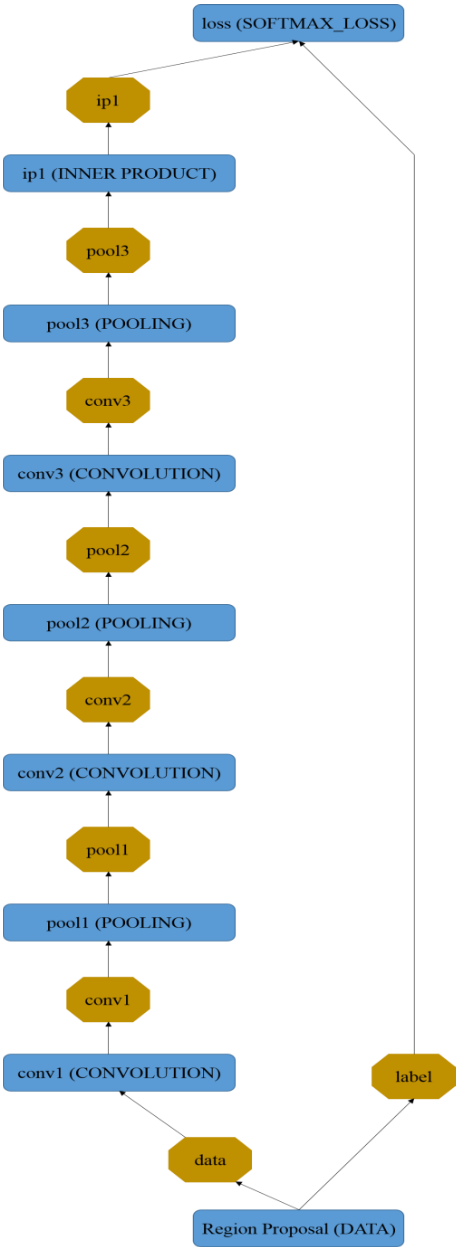

Figure 1. Our CNN structure

A. Region Proposals

Since sliding-window based detector is time-consuming, we use BING for efficiently capturing the objectness of an image window. Objectness is usually represented as a value which reflects how likely an image window covers an object of any category [18]. Vehicles are just the objects we want to find in satellite images. As vehicles in satellite images have similar sizes, we don't have to consider multi-scale problem. Otherwise, there will be many false positive proposals which have bigger sizes than vehicles. The acceleration benefitted from BING mainly owing to the use of binary approximation. In the following part, we explain the details of our method and how it works in the framework.

Since corresponding 48×48 image windows are resized to 8×8 size, we can get a score with a linear model $w \in R^{64}$ for each window. The linear model w will be learned automatically. With the use of binary approximation [19], our learned linear model can be approximated with a set of basis vectors $w \approx \sum_{i=1}^{N_b} \alpha_i b_i$ using Alg. 1, where N_b denotes the number of basis vectors, $b_i \in \{-1, 1\}^{64}$ denotes a basis vector, and $\alpha_i \in \mathrm{R}$ denotes the corresponding coefficient. We use a binary vector and its complement $b_i = b_i^+ - \overline{b_i^+}$, where $b_i^+ \in \{0, 1\}^{64}$, to repesent each b_i, thus the score of a binarized feature x could be calculated efficiently as (see [20]):

$$\langle w, x \rangle \approx \sum_{i=1}^{N_b} \alpha_i (2\langle b_i^+, x \rangle - |x|) \qquad (1)$$

If we approximate the Normed Gradient (NG) values (each saved as BYTE value) of the corresponding image windows using the top N_t binary bits of the BYTE value, then a 64D NG feature f_l can be approximated by N_t Binarized Normed Gradients (BING) features as

$$f_l = \sum_{j=1}^{N_t} 2^{8-j} x_{j,l} \qquad (2)$$

where $l = (m, n)$ denotes the location of a window, $x_{j,l}$ denotes the corresponding BING feature. Then the filter score of an image window corresponding to BING features $x_{j,l}$ can be efficiently tested as

$$\begin{aligned} s_l = \langle w, f_l \rangle &\approx \sum_{i=1}^{N_b} \alpha_i \sum_{j=1}^{N_t} 2^{8-j} \left(2\langle b_i^+, x_{j,l} \rangle - |x_{j,l}|\right) \\ &= \sum_{i=1}^{N_b} \alpha_i \sum_{j=1}^{N_t} C_{i,j} \end{aligned} \qquad (3)$$

where $C_{i,j}$ can be cacluated using fast BITWISE and POPCNT SSE operators.

Algorithm 1 Binary approximate model w [20]

```
**Input:** $w, N_b$
**Output:** $\{\alpha_i\}_{i=1}^{N_t}$, $\{b_i\}_{i=1}^{N_t}$
**Initialize residual:** $\varepsilon = w$
**for** i=1 to $N_t$ **d006F**
    $b_i = \mathrm{sign}(\varepsilon)$
    $\alpha_i = \langle b_i, \varepsilon \rangle / ||b_i||^2$
    $\varepsilon \leftarrow \varepsilon - \alpha_i b_i$
**end for**
```

We use linear SVM [12] to learn our linear model w. NG features of the ground truth object windows and random sampled background windows are used as positive and negative training samples respectively. After performing the Non-Max Suppression (NMS), we select a set of region proposals. These region proposals are the input data of CNN. In the following part, we explain the details of our CNN structure and how it works in the framework.

B. Feature Extraction

To learn robust features from real data automatically instead of handcrafted features, we choose CNN to extract smart features. Also, there are already various deep learning frameworks, such as, Caffe [17], Cuda-Convnet2 [21], Theano/Pylearn2 [22] and so on. Caffe is an excellent toolbox with cleanliness, readability, and speed. What's more, Caffe net can have any arbitrary Directed Acyclic Graph (DAG) structure. Thus we use Caffe to implement our CNN structure. There are three convolution layers, three pooling layers and one inner product layer as well as a softmax loss layer at the end. Fig. 1 shows our CNN structure.

As mentioned above, our region proposals havea size of 48×48 with three channels. We use a batch size of 64, and scale the incoming pixels so that they are in the range [0, 1). The following *conv1* layer produces outputs of 20 channels, with the convolutional kernel size 7 and carries out with stride 1. The filters allow us to randomly initialize the values of the weights and bias. The convolutional layers have three mechanisms: (i) local receptive fields, (ii) weight sharing, and (iii) subsampling. These mechanisms guarantee local correlations namely pixels that are spatially nearby are highly correlated incorporated into the automatical features. The outputs of the convolutional units form the inputs to the pooling layer of the network. Next, we perform max pooling namely *pool1* with a pool kernel size 2 and a stride of 2. It means continuous and no overlapping between neighboring pooling regions. The pooling layers help us reduce the dimension of the feature and can also improve the results (less over-fitting). Similarly, the following *conv2* layer has outputs of 8 channels, with the convolutional kernel size 4. The *pool2* layer is the same to the *pool1* layer. Then we define *conv3* layer which is the same to the *conv2* layer. After conv3 layer, we also define *pool3* layer with a pool kernel size 2 and a stride of 2. Therewith, a fully connected layer named *ip1* layer is defined and corresponds to a traditional Multilayer Perception (MLP). It has 2 outputs. The input of the *ip1* layer is the set of all features maps at the *pool3* layer. Finally, we organize a softmax loss layer at the end of the network to classify the features of the region proposals.

III. Experiment

Our dataset includes 63 satellite images from google earth of San Francisco city which contains 6,887 vehicles and 224,366 window samples. 31 images including 3,874 vehicles and 134,430 window samples are used as training set, the remaining 32 images are used as test set.

A. Region Proposals via BING

We first introduce the experiments on region proposals. NG features of the ground truth object windows and random sampled background windows are used as positive and negative training samples respectively. Therefore there are 3,847 positive training samples. To keep the balance between the number of positive and negative training samples, we produce 5,202 negative samples. Experiments show our method generates a small set of high quality object windows, yielding 96.9% object Detection Rate (DR) with 1,000 proposals. Increasing the number of proposals for computing BING features, our performance can be further improved to 99.4% with 2,000 proposals and 99.7% with 3,000 proposals. Table I reports the performance of our method. As can be seen from Table I, the performance would be improved when there are more than 3,000 proposals, thus we choose 3,000 proposals for every satellite images. To achieve the same locating precision, the normal sliding window based methods need 10,400 sliding windows per image, thus our method is more efficient in searching. Furthermore, the learning stage of our model takes 4.547 seconds and all the procedures are automatically learned and completed without fine tuning parameters manually. Fig. 2 shows our results of extracting region proposals. The red boxes are correctly detected and the green boxes are not. Fig. 2 shows that BING reduces search space in smooth area meanwhile captures the vehicles we want to locate.

TABLE I. DETECTION RATE VIA BING METHOD

Proposals	100	1,000	2,000	3,000	4,000	5,000
DR	0.41	0.969	0.994	0.997	0.997	0.997

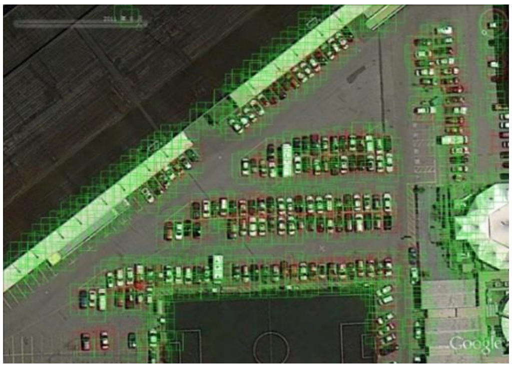

Figure 2. Region proposals via BING

Generally speaking, the ground truth object windows and random sampled background windows are used as positive and negative CNN training samples respectively. Through producing abundant region proposals via our trained BING model, we get more meaningful labelled training samples. If the max union area ratio exceed 0.6, we will label it as positive sample. If the max union area ratio is less than 0.4, we will label it as negative sample. These hard samples can make the classifier more accurate than random sampled background windows.

B. Experiments on CNN

Here we define three quantization indexes False Alarm Rate (FAR), Precision Rate (PR) and Recall Rate (RR) as follow:

$$\begin{cases} FAR = \dfrac{number\ of\ false\ alarms}{number\ of\ vehicles} \times 100\% \\ PR = \dfrac{number\ of\ detected\ vehicles}{number\ of\ detected\ objects} \times 100\% \\ RR = \dfrac{number\ of\ detected\ vehicles}{number\ of\ vehicles} \times 100\% \end{cases}$$

The training data of CNN comes from 3 parts, the ground truth, random sampled data and the region proposals of BING. In order to get rotation-invariant neural network, we rotate every ground truth location window 10 times by: $9^{\circ}, 18^{\circ}, 27^{\circ} \cdots 90^{\circ}$. Then we get 44,000 ground truth training samples, and 5,202 random negative training samples. BING also produce 3225 positive and 82003 negative training samples. The architecture of our CNN is described in Sec. 2.2. After training our CNN with the data above, we use BING to produce region proposals on our 32 test images. Then our trained CNN will test on the region proposals. The results are shown in Table II. Our method is much faster than other methods meanwhile has comparability accuracy detection rate. Fig. 3 shows the detecting results of our method. The red boxes are correctly detected and the blue ones are not.

TABLE II. FAR AND TRAINING TIME OF OUR METHOD

Method	Train (hour)	Give recall rate					
		95%	90%	85%	80%	75%	70%
Our	**0.5**	22.9	**12.10**	7.66	**5.01**	**3.42**	**2.50**
DNN[3]	5.81	**23.5**	12.2	7.5	5.07	3.45	2.61
HOG+SVM[5]	3.25	67.5	43.4	29.3	20.2	14.3	10.3
LBP+SVM[6]	7.84	87.6	59.2	43.0	32.8	24.5	19.4
Adaboost[22]	2.31	91.6	65.3	49.1	40.1	31.6	25.8

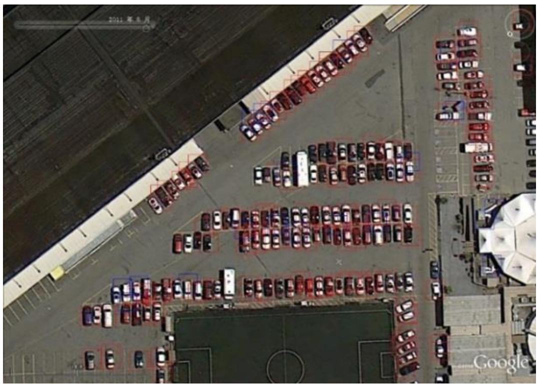

Figure 3. The detection results of our method. The red boxes are right results and the blue ones are false alarms

IV. CONCLUSION

In this work, we introduce a new automatic vehicle detection approach, which is based on Binary Normed Gradients (BING) and Convolutional Neural Network (CNN). Experiments on the satellite images validated that our proposed approach achieve better performance both in speed and accuracy. Specifically, our method speeds up more than 10 times comparing with traditional approaches, and our accuracy outperforms the state-of-the-art methods.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant nos. 61370039, 61305049, 61375024 and 91438105).

References

[1] H. Zheng, L. Pan, and L. Li, "A morphological neural network approach for vehicle detection from high resolution satellite imagery," in *Proc. 13th International Conference on Neural Information Processing*, 2006.

[2] X. Y. Chen, S. M. Xiang, C. L. Liu, and C. H. Pan, "Vehicle detection in satellite images by parallel deep convolutional neural networks," in *Proc. 2nd IAPR Asian Conference on Pattern Recognition*, Nov. 2013.

[3] S. Hinz, "Detection and counting of cars in aerial images," in *Proc. ICIP*, 2003.

[4] L. Chen, Z. Jiang, J. Yang, and Y. Ma, "A coarse-to-fine approach for vehicles detection from aerial images," in *Proc. International Conference on Computer Vision in Remote Sensing*, 2012, pp. 221-225.

[5] H. Grabner, T. Nguyen, B. Gruber, and H. Bischof, "On-line boosting-based car detection from aerial images," *J. Photogrammetry and Remote Sensing*, vol. 63, no. 3, pp. 382-396, 2008.

[6] T. Zhao and R. Nevatia, "Car detection in low resolution aerial images," in *Proc. International Conference on Computer Vision*, 2001, vol. 1, pp. 710-717.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 886-893.

[9] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, July 2002.

[10] P. Liang, G. Teodoro, H. Ling, E. Blasch, G. Chen, and L. Bai, "Multiple kernel learning for vehicle detection in wide area motion imagery," in *Proc. 15th International Conference on Information Fusion*, 2012, pp. 1629-1636.

[11] A. Kembhavi, D. Harwood, and L. S. Davis, "Vehicle detection using partial least squares," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1250–1265, Jun. 2011.

[12] M. M. Cheng, Z. M. Zhang, W. Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE CVPR*, 2014.

[13] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *IJCV*, 2013.

[14] I. Endres and D. Hoiem, "Category independent object proposals," in *Proc. ECCV*, 2010.

[15] Y. L. Cun, B. Boser, J. Denker, D. Henderson, *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comp.*, 1989.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradientbased learning applied to document recognition," *Proc. of the IEEE*, 1998.

[17] Y. Jia. (2013). Caffe: An open source convolutional architecture for fast feature embedding. [Online]. Available: http://caffe.berkeleyvision.org/

[18] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE TPAMI*, vol. 34, no. 11, 2012.

[19] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. "Fast, accurate detection of 100,000 object classes on a single machine," in *Proc. CVPR*, 2013.

[20] S. Hare, A. Saffari, and P. H. Torr, "Efficient online structured output learning for keypoint-based object tracking," in *Proc. CVPR*, 2012, pp. 1894–1901.

[21] [Online]. Available: https://code.google.com/p/cuda-convnet2/

[22] Theano. [Online]. Available: http://deeplearning.net/software/theano/

Shenquan Qu received the B.S. degree in computer science and technology from University of Science and Technology of China, Hefei, China, in 2012. He is working toward the M.S. degree in the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His interests include machine learning and big data.

Ying Wang received his B.S. degree from the Department of Information and Communications Technologies, Nanjing University of Information Science and Technology, China in2005, and M.S. degree from the Department of Automation Engineering, Nanjing University of Aeronautics and Astronautics, China, in 2008, and his Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, China, in 2012. He is currently an assistant professor at the National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences. His research interests include computer vision and pattern recognition.

Gaofeng Meng received the B.S. degree in applied mathematics from Northwestern Polytechnical University, Xian, China, in 2002, and the M.S. degree in applied mathematics from Tianjin University, Tianjin, China, in2005, and the Ph.D. degree in control science and engineering from Xian Jiaotong University, Xian, Shaanxi, China, in 2009. In 2009, he joined the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, as an assistant professor. His research interests include image processing, computer vision, and pattern recognition.

Chunhong Pan received his B.S. degree in automatic control from Tsinghua University, Beijing, China, in1987, his M.S. degree from Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, China, in 1990, and his Ph.D. degree in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences, Beijing, in 2000. He is currently a professor at National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, and remote sensing.