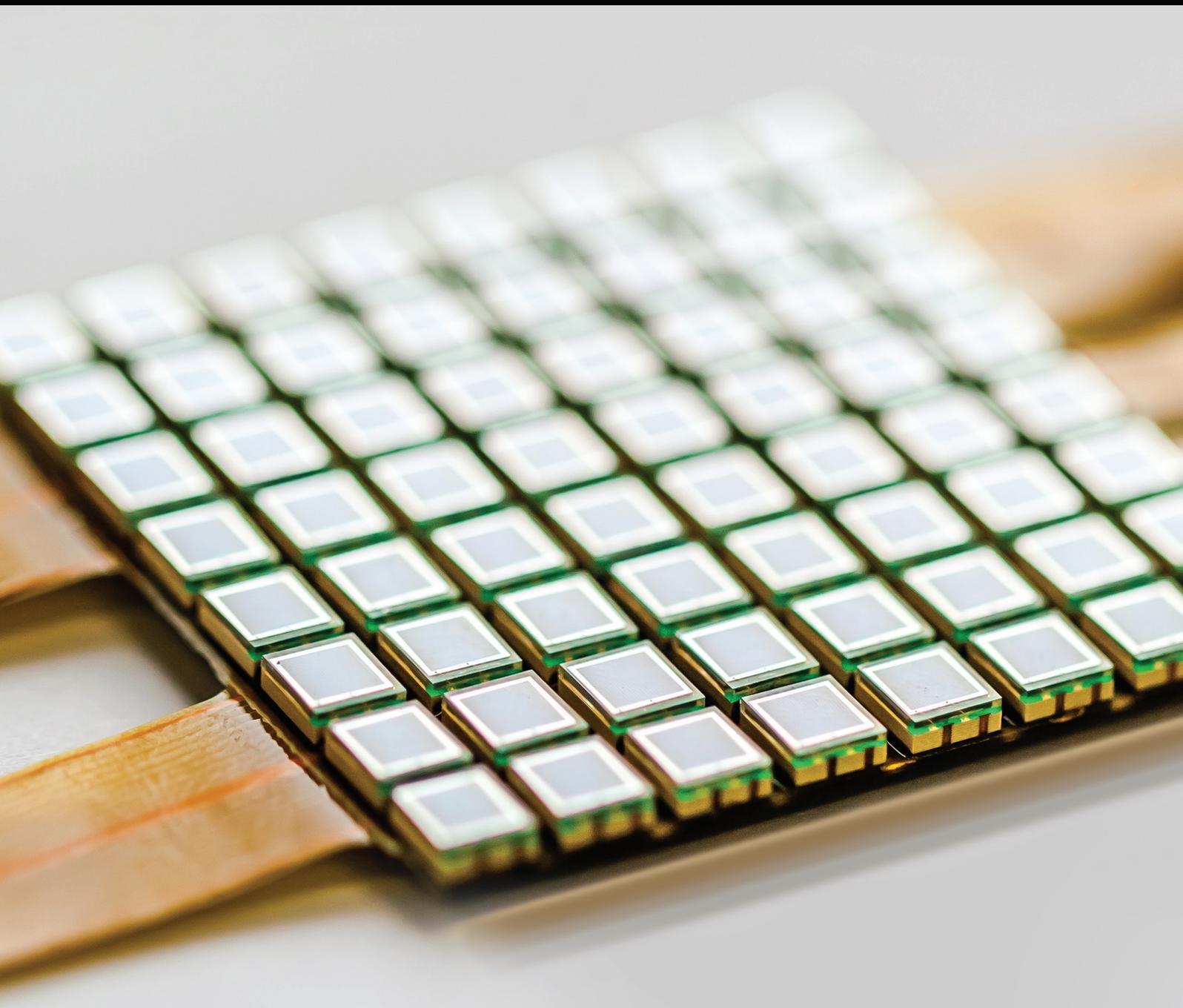


Deep Learning for Remote Sensing Image Understanding

Guest Editors: Liangpei Zhang, Gui-Song Xia, Tianfu Wu, Liang Lin,
and Xue Cheng Tai





Deep Learning for Remote Sensing Image Understanding

Deep Learning for Remote Sensing Image Understanding

Guest Editors: Liangpei Zhang, Gui-Song Xia, Tianfu Wu, Liang Lin, and Xue Cheng Tai



Copyright © 2016 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in "Journal of Sensors." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Harith Ahmad, Malaysia
Sheikh Akbar, USA
Francisco J. Arregui, Spain
Francesco Baldini, Italy
Fernando Benito-Lopez, Ireland
Romeo Bernini, Italy
Shekhar Bhansali, USA
Wojtek J. Bock, Canada
Hubert Brändle, Switzerland
Davide Brunelli, Italy
Belén Calvo, Spain
Stefania Campopiano, Italy
Jian-Nong Cao, Hong Kong
Sara Casciati, Italy
Chi Chiu Chan, Singapore
Nick Chaniotakis, Greece
Nicola Cioffi, Italy
Elisabetta Comini, Italy
Marco Consales, Italy
Jesus Corres, Spain
Andrea Cusano, Italy
Dzung V. Dao, Japan
Catherine Dehollain, Switzerland
Manel del Valle, Spain
Ignacio Del Villar, Spain
Utkan Demirci, USA
Junhang Dong, USA
Abdelhamid Errachid, France
Stephane Evoy, Canada
Vittorio Ferrari, Italy
Luca Francioso, Italy
Laurent Francis, Belgium

Mohammad Reza Ganjali, Iran
Wei Gao, Japan
Michele Giordano, Italy
Banshi D. Gupta, India
María del Carmen Horrillo, Spain
Wieslaw Jakubik, Poland
Hai-Feng Ji, USA
Kouros Kalantar-Zadeh, Australia
Sher Bahadar Khan, KSA
Sang Sub Kim, Republic of Korea
Challa Kumar, USA
Hiroki Kuwano, Japan
Laura M. Lechuga, Spain
Chengkuo Lee, Singapore
Chenzhong Li, USA
Eduard Llobet, Spain
Yu-Lung Lo, Taiwan
Oleg Lupan, Moldova
Eugenio Martinelli, Italy
Jose R. Martinez-de Dios, Spain
Yasuko Y. Maruo, Japan
Ignacio R. Matias, Spain
Mike McShane, USA
Igor L. Medintz, USA
Fanli Meng, China
Aldo Minardo, Italy
Joan Ramon Morante, Spain
Lucia Mosiello, Italy
Masayuki Nakamura, Japan
Vincenzo Paciello, Italy
Marimuthu Palaniswami, Australia
Alberto J. Palma, Spain

Gyuhae Park, Republic of Korea
Alain Pauly, France
Giorgio Pennazza, Italy
Michele Penza, Italy
Andrea Ponzoni, Italy
Biswajeet Pradhan, Malaysia
Ioannis Raptis, Greece
Christos Riziotis, Greece
Maria Luz Rodríguez-Méndez, Spain
Albert Romano-Rodriguez, Spain
Josep Samitier, Spain
Giorgio Sberveglieri, Italy
Luca Schenato, Italy
Michael J. Schöning, Germany
Andreas Schütze, Germany
Woosuck Shin, Japan
Pietro Siciliano, Italy
Vincenzo Spagnolo, Italy
Weilian Su, USA
Tong Sun, UK
Hidekuni Takao, Japan
Isao Takayanagi, Japan
Pierre Temple-Boyer, France
Guiyun Tian, UK
Suna Timur, Turkey
Hana Vaisocherova, Czech Republic
Qihao Weng, USA
Matthew J. Whelan, USA
Stanley E. Woodard, USA
Hai Xiao, USA
Hyeonseok Yoon, Republic of Korea

Contents

Deep Learning for Remote Sensing Image Understanding

Liangpei Zhang, Gui-Song Xia, Tianfu Wu, Liang Lin, and Xue Cheng Tai
Volume 2016, Article ID 7954154, 2 pages

Improved Quantum Particle Swarm Optimization for Mangroves Classification

Zhehuang Huang
Volume 2016, Article ID 9264690, 8 pages

Deep Convolutional Neural Networks for Hyperspectral Image Classification

Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li
Volume 2015, Article ID 258619, 12 pages

Urban Land Use and Land Cover Classification Using Remotely Sensed SAR Data through Deep Belief Networks

Qi Lv, Yong Dou, Xin Niu, Jiaqing Xu, Jinbo Xu, and Fei Xia
Volume 2015, Article ID 538063, 10 pages

Bayesian Information Criterion Based Feature Filtering for the Fusion of Multiple Features in High-Spatial-Resolution Satellite Scene Classification

Da Lin, Xin Xu, and Fangling Pu
Volume 2015, Article ID 142612, 10 pages

Automatic Change Detection Method of Multitemporal Remote Sensing Images Based on 2D-Otsu Algorithm Improved by Firefly Algorithm

Liang Huang, Yuanmin Fang, Xiaoqing Zuo, and Xueqin Yu
Volume 2015, Article ID 327123, 8 pages

Automatic Fusion of Hyperspectral Images and Laser Scans Using Feature Points

Xiao Zhang, Aiwu Zhang, and Xiangang Meng
Volume 2015, Article ID 415361, 9 pages

Editorial

Deep Learning for Remote Sensing Image Understanding

Liangpei Zhang,¹ Gui-Song Xia,¹ Tianfu Wu,² Liang Lin,³ and Xue Cheng Tai⁴

¹State Key Lab. LIESMARS, Wuhan University, Wuhan 430079, China

²Department of Statistics, University of California, Los Angeles (UCLA), Los Angeles, CA 90095, USA

³School of Advanced Computing, Sun Yat-Sen University, Guangzhou 510275, China

⁴Department of Mathematics, University of Bergen, 5020 Bergen, Norway

Correspondence should be addressed to Liangpei Zhang; zlp62@whu.edu.cn

Received 28 June 2015; Accepted 28 June 2015

Copyright © 2016 Liangpei Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep learning has become one of the most important breakthroughs in artificial intelligence over the past decade. Deep learning contains a variety of methods, including neural networks, hierarchical probabilistic models, and many specific unsupervised and supervised feature-learning algorithms. The biggest difference between deep learning and the classic visual recognition methods is that deep learning methods automatically learn features from a huge amount of data, rather than requiring engineering features by hand. Therefore, we can conveniently learn good features for new specific tasks without much expertise and effort of designing features. In addition, deep learning approaches make better use of big data and provide an end-to-end learning framework in which jointly learning feature transformations and classifiers via the back propagation technique makes their integration optimal.

Deep learning is about learning hierarchical feature representations. Deep architectures with multiple levels attempt to learn hierarchical structures and seem promising in learning simple concepts first and then successfully building up more complex concepts by composing the simpler ones together. It accords with human's visual cognition of learning abstract concepts on top of less abstract ones. These high-level feature representations are more powerful and robust in typical visual tasks.

In the intelligent interpretation of remote sensing images, the automatic target detection (or recognition) and high-resolution remotely sensed image classification are two hot topics, and both of these two tasks are carried out by first computing the low-level features in the raw images. For different kinds of remote sensing images (e.g., SAR images

and hyperspectral images), the corresponding specific feature representations are available. Through applying deep learning methods, we are free of these handcrafted low-level features and can automatically learn mid-level and higher-level ones from a large amount of unlabeled raw samples beyond the types and domains of remote sensing images. Deep learning methods can undoubtedly offer better feature representations for the related remote sensing task, and there is a bright prospect of seeing more and more researchers dedicated to learning better features for the target detection and scene classification tasks by utilizing deep learning methods appropriately.

This special issue concentrates on the research in new methods, algorithms, and architectures of deep learning to handle the practical challenges in remote sensing image processing. The papers in this issue can be roughly organized into three main categories: (a) remote sensing imagery classification, (b) change detection of multitemporal remote sensing images, and (c) fusion of diverse types of images.

Papers in category (a) deal with the classic classification problem for distinct types of remote sensing images. The paper by Z. Huang explores the synergetic neural networks optimized by an improved quantum particle swarm algorithm for mangroves classification. The paper by Q. Lv et al. introduces deep belief networks to extract effective contextual mapping features for the task of PolSAR image classification. The paper by W. Hu et al. specially designs a five-layer convolutional neural network to classify hyperspectral images directly in spectral domain. The paper by D. Lin et al. proposes a novel multifeature fusion strategy via the sparse

canonical correlation analysis and Bayesian information criterion for high-resolution satellite image classification.

Paper by L. Huang et al. introduces the 2D-Otsu threshold segmentation methods to analyze the changing area between two temporal remote sensing images, which belongs to category (b). Category (c) also contains one paper by X. Zhang et al., which presents novel methods for automatic fusion of 2D hyperspectral images and 3D laser scans by exploring collinearity equation and direct linear transformation to create the initial corresponding relationship of the two raw datasets.

Note that this special issue only provides a small number of papers and relatively narrow insights on remote sensing image understanding, and it is impossible to cover complete aspects of remote sensing applications with deep learning methods due to the page limitations.

Acknowledgment

Many individuals made contributions to the success of this special issue. We appreciate all the authors for their submissions, as well as all the reviewers for their careful and professional review.

*Liangpei Zhang
Gui-Song Xia
Tianfu Wu
Liang Lin
Xue Cheng Tai*

Research Article

Improved Quantum Particle Swarm Optimization for Mangroves Classification

Zhehuang Huang^{1,2}

¹*Cognitive Science Department, Xiamen University, Xiamen 361005, China*

²*School of Mathematics Sciences, Huaqiao University, Quanzhou 362021, China*

Correspondence should be addressed to Zhehuang Huang; tingqian3@163.com

Received 20 October 2014; Revised 12 February 2015; Accepted 12 February 2015

Academic Editor: Xue Cheng Tai

Copyright © 2016 Zhehuang Huang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Quantum particle swarm optimization (QPSO) is a population based optimization algorithm inspired by social behavior of bird flocking which combines the ideas of quantum computing. For many optimization problems, traditional QPSO algorithm can produce high-quality solution within a reasonable computation time and relatively stable convergence characteristics. But QPSO algorithm also showed some unsatisfactory issues in practical applications, such as premature convergence and poor ability in global optimization. To solve these problems, an improved quantum particle swarm optimization algorithm is proposed and implemented in this paper. There are three main works in this paper. Firstly, an improved QPSO algorithm is introduced which can enhance decision making ability of the model. Secondly, we introduce synergetic neural network model to mangroves classification for the first time which can better handle fuzzy matching of remote sensing image. Finally, the improved QPSO algorithm is used to realize the optimization of network parameter. The experiments on mangroves classification showed that the improved algorithm has more powerful global exploration ability and faster convergence speed.

1. Introduction

Quantum particle swarm optimization algorithm (QPSO) is a new evolutionary algorithm proposed in 2005 [1, 2]. Researchers found that the human learning process has great uncertainty which is very similar to the quantum behaviour of particle, so each individual can be described as a particle in quantum space. In recent years, a series of papers have focused on the application of QPSO, such as financial forecasting [3], sensor array [4], clinical disease diagnoses [5], classification and clustering [6], fuel management optimization [7], feature selection [8], and other areas [9, 10]. But traditional QPSO is easy to fall into local optimum value and the convergence rate is slow. How to avoid prematurity and maintain fast convergence rate at the same time is a major problem. In order to improve the performance of QPSO, researchers have made some attempts. Sun et al. [11] reported a global search strategy of quantum behaved particle swarm optimization. In [12], chaotic mutation operator is introduced to quantum particle swarm optimization, instead of random sequences in QPSO; chaotic mutation operator

is a powerful strategy to diversify the QPSO population and can improve the performance in preventing premature convergence to local minima. In [13], an improved quantum particle swarm optimization algorithm based on real coding method is presented which can improve the performance of QPSO.

Mangrove ecosystems play irreplaceable and important roles for the stabilization and equilibrium of coastal estuary as a unique ecosystem of land and sea [14–17]. Grasping the status of mangrove communities timely and accurately can provide important information for the protection and restoration of mangrove ecosystems. But the spatial resolution of TM images is not high and the spectral similarity of mangrove communities is strong [18–20]. Due to these problems, it is necessary to adopt a more intelligent approach to improve the accuracy of mangrove classification.

We can use synergetic neural network (SNN) proposed by Haken [21] to realize intelligent mangroves classification. The basic principle of synergetic neural network is that the pattern recognition procedure can be viewed as the competition progress of many order parameters. One advantage

of synergetic neural network method is robustness against noise and occlusion; using this method we will be able to better handle fuzzy matching of mangroves classification whose contextual information is not complete. Synergetic pattern recognition method has been successfully used in face recognition [22], automatic control field [23, 24], and exon recognition [25]. Mangroves classification can also be considered as a problem of pattern recognition, and it is also entirely possible to use this method to solve mangroves classification. At present, the mainstream studies of SNN focus on the selection of prototype pattern vector [26], setting of attention parameter [27–29], reconstruction algorithm of order parameters [30], and so on. The network parameters directly influence the synergetic recognition performance. The adjustment of network parameters is a global behavior and has no general research theory to control the parameters in the recognition process at present. In this paper, we proposed an improved algorithm which can effectively choose network parameters of SNN.

This paper is organized as follows. An improved QPSO algorithm is presented in Section 2. In Section 3, mangroves classification model based on synergetic neural network is introduced. An improved SNN model based on improved QPSO is presented in Section 4. In Section 5, some experimental results and conclusions are given.

2. Improved QPSO Based on Adaptive Behavior Selection (AQPSO)

In this section, we present an improved QPSO. First, diversity function is employed as feature functions. Second, adaptive movement behavior is introduced into quantum particle swarm optimization for the first time.

2.1. Diversity Function. Diversity function is used to describe the dispersion degree of particles. Diversity function $g_1(t)$ describes adaptive diversity, where t refers to the current iteration number. Consider

$$D(t) = \frac{\min(f_{gb}, f_{avg}^t)}{\max(f_{gb}, f_{avg}^t)}, \quad g_1(t) \in (0, 1], \quad i = 1, 2, \dots, n, \quad (1)$$

where f_{avg}^t means the average fitness value of the i th iterative and f_{gb} is current global optimal value. $g_1(t) = 1$ means the diversity is poor. $g_1(t) \ll 1$ means the diversity is good.

2.2. Adaptive Movement Behavior. From the QPSO algorithm, we can know local attractor q is determined by local optimum $pbest$ and global optimum $gbest$:

$$q = \varphi \cdot pbest + (1 - \varphi) \cdot gbest, \quad (2)$$

where $\varphi = c_1 r_1 / (c_1 r_1 + c_2 r_2)$, $r_1 \sim U(0, 1)$, $r_2 \sim U(0, 1)$.

In standard QPSO algorithm, c_1 and c_2 are employed as two accelerating factors which not only will affect convergence speed, but also may lead to the occurrence of premature phenomenon. In the early iterations, the current position and the optimal location of particles are far away. At later stage,

the current position and the optimal location are closer. To effectively avoid the occurrence of premature phenomenon, $D(t)$ can be used to describe the closer degree between particles and the global optimal position, so as to select the corresponding acceleration factor:

$$c_1 = \begin{cases} (Mc - mc) \times \tan\left(\frac{MAXITER - t}{MAXITER}\right) + mc, & D(t) > R_1, \\ (Mc - mc) \times \frac{MAXITER - t}{MAXITER} + mc, & D(t) \leq R_1, \end{cases} \quad (3)$$

$$c_2 = \begin{cases} (Mc - mc) \times \frac{MAXITER - t}{MAXITER} + mc, & D(t) > R_1, \\ (Mc - mc) \times \tan\left(\frac{MAXITER - t}{MAXITER}\right) + mc, & D(t) \leq R_1. \end{cases}$$

2.3. The Improved Algorithm. The improved QPSO algorithm is shown as follows.

- (1) Initialize variables.
- (2) Generate initial artificial particle swarm.
- (3) Update the local optimum position P_{ij} and the global best position P_{gi} .
- (4) Each particle updates its location through different behavior.
- (5) Perform adaptive movement behavior by (3).
- (6) The threshold gradually reduced which can lead to the dispersion decreasing of particles.
- (7) Record the optimal value.
- (8) If the iteration is terminated, output the optimal value; otherwise return to Step (3).

3. An Improved SNN Model Based on AQPSO

3.1. Mangroves Classification Based on SNN Model. SNN model is a top-down network constructed by synergetics different from traditional neural network, and it does not produce the pseudostate. An unrecognized pattern, q , is constructed by a dynamic process which translates q into one of the prototype pattern vectors v_k through status $q(t)$; namely, this prototype pattern is closest to $q(0)$. The process is described as in the following equation: $q \rightarrow q(t) \rightarrow v_k$.

A dynamic equation can be given for unrecognized pattern:

$$\dot{q} = \sum_{k=1}^M \lambda_k v_k (v_k^+ q) - B \sum_{k' \neq k} (v_{k'}^+ q)^2 (v_k^+ q) v_k - C (q^+ q) q + F(t), \quad (4)$$

where q is the status vector of input pattern with initial value q_0 , λ_k is attention parameter, v_k is prototype pattern vector, and v_k^+ is the adjoint vector of v_k .

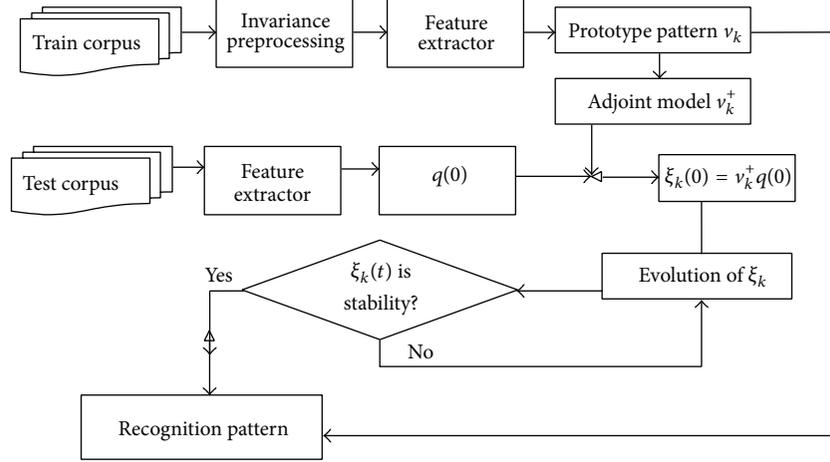


FIGURE 1: The flow chart of recognition process.

Corresponding dynamic equation of order parameters is

$$\dot{\xi}_k = \lambda_k \xi_k - B \sum_{k' \neq k} \xi_{k'}^2 \xi_k - C \left| \sum_{k'=1}^M \xi_{k'}^2 \right| \xi_k, \quad (5)$$

where ξ_k satisfies initial condition

$$\xi_k = v_k^+ q(0). \quad (6)$$

A method for mangroves classification using synergetic neural network technique is presented, as shown in Figure 1.

The mangrove classification model based on SNN mainly includes the following steps.

Step 1. Extract feature from training corpus and calculate prototype pattern vectors which satisfied the condition of normalization and zero-mean.

Step 2. Obtain the adjoint pattern of the prototype patterns.

Step 3. Extract feature from test corpus and calculate test pattern vectors $q_k(0)$ ($k = 1, 2, \dots, N$) which satisfied the condition of normalization and zero-mean.

Step 4. Calculate the initial values of order parameters according to (6).

Step 5. Evolution equation is performed for the competition among order parameters according to (5).

Step 6. If the values of ξ_k in the evolution can reach stable condition, output the final results; otherwise continue performing evolution equation in Step 5.

3.2. Parameters Optimization of SNN Based on AQPSO. The network parameters of synergetic neural network directly influence the synergetic recognition performance. There is no general research theory to control the parameters in the recognition process at present. How to construct network parameter and choose a more efficient optimization method

is an important task. To solve these problems, AQPSO is used in this section to effectively select network parameters.

A parameters optimization of synergetic neural network based on AQPSO is shown as Figure 2. Firstly, we can reconstruct the attention parameter λ_k based on the measuring similarity between the prototype pattern v_k and the test pattern q_l :

$$\lambda_k = \alpha \cos(v_k, q_l), \quad (7)$$

$$\cos(v_k, q_l) = \frac{v_k \cdot q_l}{|v_k| |q_l|}.$$

Secondly, AQPSO algorithm is used to search the global optimum parameters of network (α, B, C) in the corresponding parameter space.

The reconstruction and optimization of network parameters based on AQPSO can be described as follows.

- (1) Obtain feature vectors from train corpus and test corpus and construct prototype pattern v_k ($k = 1, 2, \dots$) and test pattern q_l ($l = 1, 2, \dots$).
- (2) Calculate initial order parameter ξ_{lk} according to (6).
- (3) Set λ_k , ($k = 1, 2, \dots, m$) according to (7).
- (4) Optimization algorithms are used to search the global optimum parameters (α, B, C) of SNN in the corresponding parameter space.
- (5) Get best mangrove categories through the evaluation of order parameter equation (5).

4. Experiment

In the experiment, we take 10 mangrove images as training samples. The size of images is 320×320 . The categories of the mangrove images are shown as Table 1.

The prototype pattern of training samples is shown in Figure 3.

The corresponding adjoint mode v_k^+ is shown in Figure 4.

In the next section, we use different mangrove images to test the performance of our proposed model.

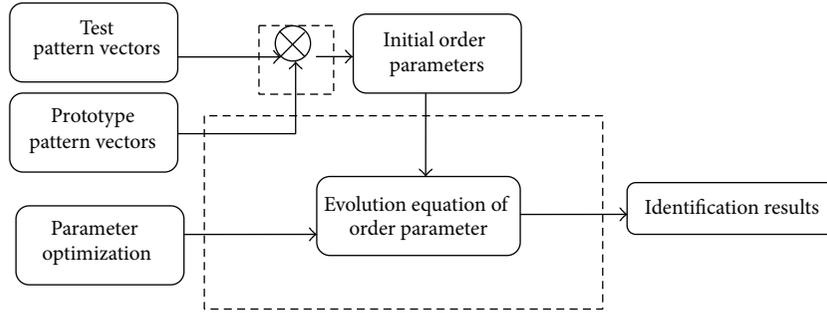


FIGURE 2: Parameters optimization of SNN based on AQPSO algorithm.

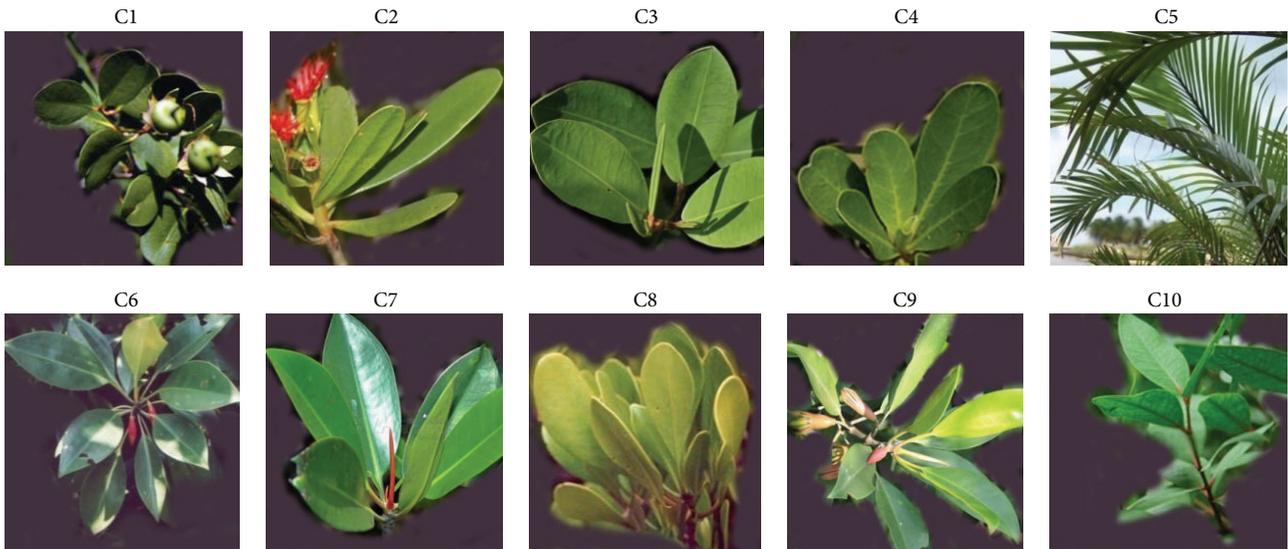


FIGURE 3: The corresponding prototype pattern of training samples.

TABLE 1: Mangrove categories.

Class name	Mangrove
C1	<i>Avicennia</i>
C2	<i>Lumnitzera</i>
C3	<i>Pelliciera</i>
C4	<i>Conocarpus</i>
C5	<i>Nypa</i>
C6	<i>Rhizophora</i>
C7	<i>mucronata</i>
C8	<i>Ceriops</i>
C9	<i>Bruguiera</i>
C10	<i>Sonneratia</i>

4.1. Recognition of Single Mangrove Image

4.1.1. Recognition of Noisy Images. The samples to be identified are formed by adding noise to the original image, without translation, rotation, or scaling, as shown in Figure 5.

According to the stability analysis method, it can get fast and stable convergent curves when setting network

parameters $B = C = 1$. The recognition process is shown in Figure 6.

From Figure 7, we can find category C1 eventually won through the competition. The evolution curve of competitive process could converge quickly and became stable after the 95th iterations.

4.1.2. Recognition of Incomplete Image. The incomplete image to be identified is shown in Figure 8.

The evolution curve is shown in Figure 9.

From Figure 9, we can find the initial order parameter of category C3 is not the biggest at the beginning (the biggest is C7); however, it eventually won through the competition and desired pattern is recognized. In this process, the category of ambiguous mangrove image will be determined. Meanwhile, the convergence rate of the competitive process could converge quickly and became stable after the 85th iterations.

4.2. Recognition of Large-Scale Images. We take 10 mangrove images as training samples. The size of image is 320×320 . We get 120 test images by various processing methods, such as adding noise, rotation, scaling, cropping, and fuzziness on the training samples.

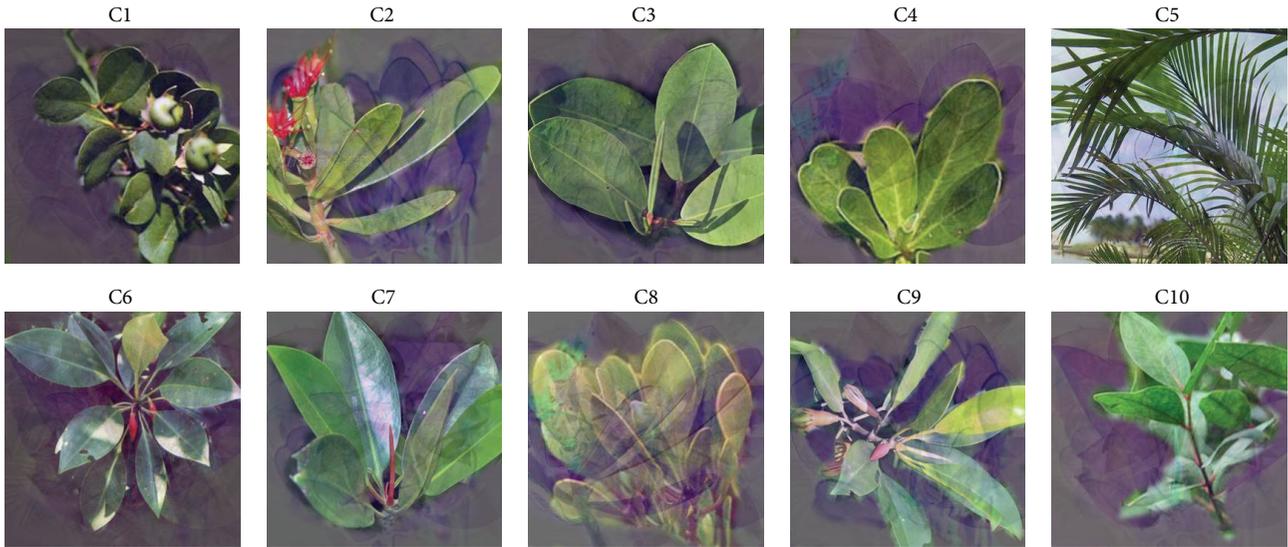


FIGURE 4: The corresponding adjoint pattern.



FIGURE 5: Noisy sample to be identified.

For comparison, we use four strategies:

SNN: mangroves classification based on SNN,

SNN + PSO: mangroves classification based on SNN and PSO,

SNN + QPSO: mangroves classification based on SNN and QPSO,

SNN + AQPSO: mangroves classification based on SNN and AQPSO.

The experiments are implemented in a notebook PC (Lenovo ThinkPad T430) which includes a 2.5 GHz CPU with 4 GB RAM. The operating system is Windows 7.

The parameter setting of PSO is shown as Table 2.

The parameter setting of QPSO and AQPSO is shown as Table 3.

The results are shown in Table 4. Each point is made from average values of over 30 repetitions.

TABLE 2: The parameter setting of PSO.

Algorithm	Population size	C1	C2	Maximum number of iterations
PSO	30	2	2	20

TABLE 3: The parameter setting of QPSO and AQPSO.

Algorithm	Population size	M_c	m_c	Maximum number of iterations
QPSO	30	3	0.005	20
AQPSO	30	3	0.005	20

TABLE 4: The testing performance comparison among different optimization methods.

Algorithm	SNN	SNN + PSO	SNN + QPSO	SNN + AQPSO
Accurate rate (%)	70.8	82.5	83.3	85.8
Time (s)	79	552	528	516

TABLE 5: The performances of SNN + QPSO and SNN + AQPSO.

Algorithm	Mean (%)	SD	Time (s)	Iterations
SNN + QPSO	83.3	0.072	528	12
SNN + AQPSO	85.8	0.052	516	8

From Tables 4 and 5, we can see no model performs better than others for all evaluation indicators, but the accurate rates of all the three parameter optimization models (SNN + PSO, SNN + QPSO, and SNN + AQPSO) are comparable to SNN model. The reason is that the attention parameters are very important for SNN and optimization algorithm is essential for better performance.

SNN + AQPSO can effectively improve accurate rate compared with the standard PSO algorithm and QPSO algorithm.



FIGURE 6: The recognition process.

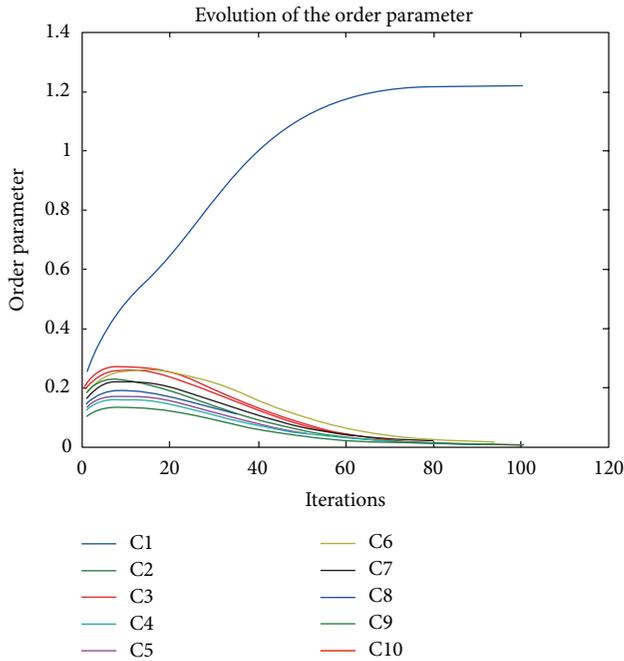


FIGURE 7: The evolution curve of order parameter.

The new behaviors of particle swarm presented in this paper are essential for better performance. At the same time, the behavior selection can enhance decision making ability of behavior selection. On the convergence time, there is a significant improvement as expected. In conclusion, the improved algorithm has better global search ability and fast convergence speed.

The convergence curve is shown in Figure 10. We can see there is a significant improvement as expected on the convergence accuracy. The experiment results show the improved QPSO has better global and local parameter searching abilities.

5. Conclusions

In the paper, we construct an improved quantum particle swarm optimization. Experiments on mangroves classification show the improved algorithm has more powerful global exploration ability with faster convergence speed.

We got the following conclusions.



FIGURE 8: Incomplete sample.

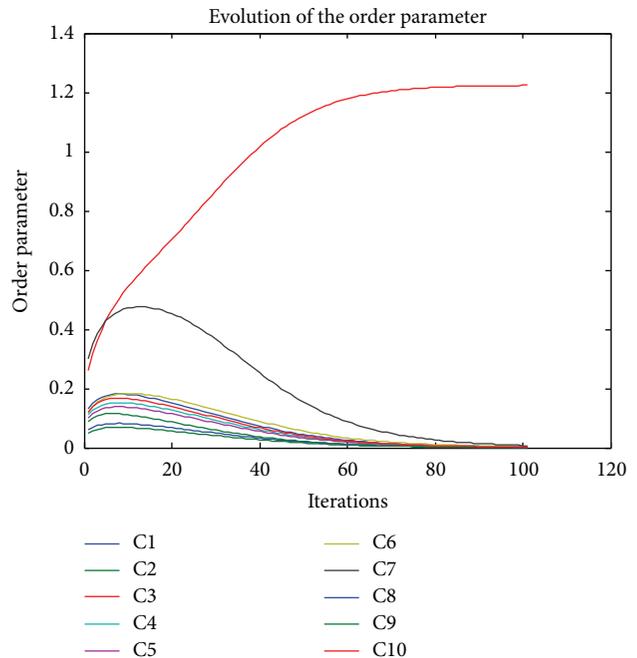


FIGURE 9: The evolution curve of order parameter.

- (1) The mangroves classification procedure can be viewed as the competition progress of many order parameters. The order parameter can reflect the similarity between the prototype and test pattern better.

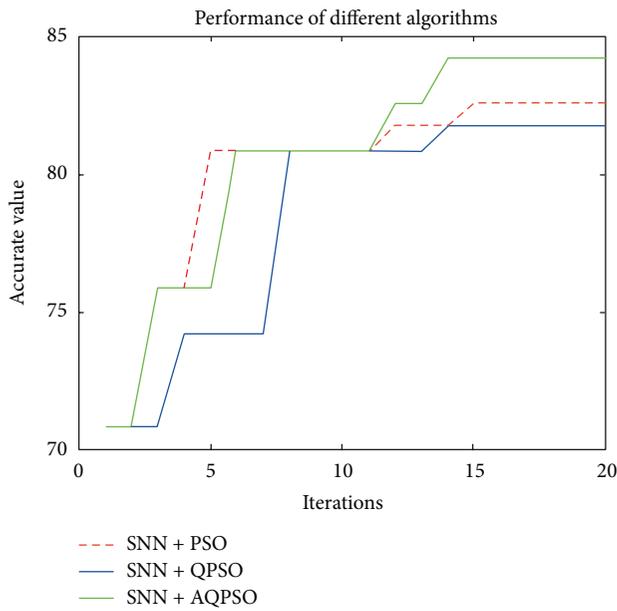


FIGURE 10: Convergence curve.

- (2) The AQPSO has both global and local search ability and can effectively choose network parameters of SNN.

In the future, we will focus on the following two aspects.

- (1) The network parameters of SNN are very important for better recognition performance. The change of attention parameters will lead to complete different recognition results. We will use other optimization algorithms to search the global optimum parameters of SNN in the corresponding parameter space.
- (2) The behavior of particle swarm has a critical influence for better performance. In the future, we will introduce some new behaviors and apply this idea to other optimization tasks.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

- [1] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, vol. 1, pp. 325–331, June 2004.
- [2] P. Moore and G. K. Venayagamoorthy, "Evolving combinational logic circuits using a hybrid quantum evolution and particle swarm inspired algorithm," in *Proceedings of the NASA/DoD Conference on Evolvable Hardware (EH '05)*, pp. 97–102, July 2005.
- [3] A. Bagheri, H. Mohammadi Peyhani, and M. Akbari, "Financial forecasting using ANFIS networks with Quantum-behaved Particle Swarm Optimization," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6235–6250, 2014.
- [4] P. Jia, F. Tian, S. Fan, Q. He, J. Feng, and S. X. Yang, "A novel sensor array and classifier optimization method of electronic nose based on enhanced quantum-behaved particle swarm optimization," *Sensor Review*, vol. 34, no. 3, pp. 304–311, 2014.
- [5] H. Ye, W. Luo, and Z. Li, "Convergence analysis of particle swarm optimizer and its improved algorithm based on velocity differential evolution," *Computational Intelligence and Neuroscience*, vol. 2013, Article ID 384125, 7 pages, 2013.
- [6] S. Ch, N. Anand, B. K. Panigrahi, and S. Mathur, "Streamflow forecasting by SVM with quantum behaved particle swarm optimization," *Neurocomputing*, vol. 101, pp. 18–23, 2013.
- [7] M. Jamalipour, R. Sayareh, M. Gharib, F. Khoshahval, and M. R. Karimi, "Quantum behaved Particle Swarm Optimization with Differential Mutation operator applied to WWER-1000 in-core fuel management optimization," *Annals of Nuclear Energy*, vol. 54, pp. 134–140, 2013.
- [8] A. R. Behjat, A. Mustapha, H. Nezamabadi-Pour, M. N. Sulaiman, and N. Mustapha, "Feature subset selection using binary quantum particle swarm optimization for spam detection system," *Advanced Science Letters*, vol. 20, no. 1, pp. 188–192, 2014.
- [9] J. Zhang and L. Shen, "An improved fuzzy c-means clustering algorithm based on shadowed sets and PSO," *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 368628, 10 pages, 2014.
- [10] N. G. Hedeshi and M. S. Abadeh, "Coronary artery disease detection using a fuzzy-boosting PSO approach," *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 783734, 12 pages, 2014.
- [11] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 111–116, December 2004.
- [12] L. D. S. Coelho, "A quantum particle swarm optimizer with chaotic mutation operator," *Chaos, Solitons & Fractals*, vol. 37, no. 5, pp. 1409–1418, 2008.
- [13] Y. Guofu, "An improved quantum particle swarm optimization algorithm based on real coding method," *International Journal of Advancements in Computing Technology*, vol. 4, no. 3, pp. 181–188, 2012.
- [14] D. M. Alongi, "Present state and future of the world's mangrove forests," *Environmental Conservation*, vol. 29, no. 3, pp. 331–349, 2002.
- [15] L. Wei, T. Balz, M. Liao, and L. Zhang, "TerraSAR-X StripMap data interpretation of complex urban scenarios with 3D SAR tomography," *Journal of Sensors*, vol. 2014, Article ID 386753, 7 pages, 2014.
- [16] B. W. Heumann, "Satellite remote sensing of mangrove forests: recent advances and future opportunities," *Progress in Physical Geography*, vol. 35, no. 1, pp. 87–108, 2011.
- [17] C. Giri, E. Ochieng, L. L. Tieszen et al., "Status and distribution of mangrove forests of the world using earth observation satellite data," *Global Ecology and Biogeography*, vol. 20, no. 1, pp. 154–159, 2011.
- [18] R. Jing and L. Kong, "Boundary detection method for large-scale coverage holes in wireless sensor network based on minimum critical threshold constraint," *Journal of Sensors*, vol. 2014, Article ID 985854, 13 pages, 2014.

- [19] Y. Luo, M. Liao, J. Yan, and C. Zhang, "A multi-features fusion support vector machine method (MF-SVM) for classification of mangrove remote sensing image," *Journal of Computational Information Systems*, vol. 8, no. 1, pp. 323–334, 2012.
- [20] V. F. Rodríguez-Galiano, F. Abarca-Hernández, B. Ghimire, M. Chica-Olmo, P. M. Atkinson, and C. Jeganathan, "Incorporating spatial variability measures in land-cover classification using random forest," *Procedia Environmental Sciences*, vol. 3, pp. 44–49, 2011.
- [21] H. Haken, *Synergetic Computers and Cognition: A Top-Down Approach to Neural Nets*, Springer, Berlin, Germany, 1991.
- [22] J. Shao, J. Gao, and X. Yang, "Synergetic face recognition algorithm based on ICA," in *Proceedings of the International Conference on Neural Networks and Brain Proceedings (ICNNB '05)*, pp. 249–253, Beijing, China, October 2005.
- [23] Z. H. Jiang and R. A. Dougal, "Synergetic control of power converters for pulse current charging of advanced batteries from a fuel cell power source," *IEEE Transactions on Power Electronics*, vol. 19, no. 4, pp. 1140–1150, 2004.
- [24] W. Z. Wang, L. Pan, and B. H. Liu, "Synergetic method of traffic state recognition based on manifold learning," in *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL '09)*, pp. 587–591, Shenyang, China, August 2009.
- [25] Z. H. Huang and Y. D. Chen, "A two-stage exon recognition model based on synergetic neural network," *Computational and Mathematical Methods in Medicine*, vol. 2014, Article ID 503132, 7 pages, 2014.
- [26] X. L. Ma and L. C. Jiao, "Reconstruction of order parameters based on immunity clonal strategy for image classification," in *Image Analysis and Recognition*, A. Campilho and M. Kamel, Eds., vol. 3211 of *Lecture Notes in Computer Science*, pp. 455–462, Springer, Berlin, Germany, 2004.
- [27] X. L. Ma, S. Wang, and L. C. Jiao, "Robust classification of immunity clonal synergetic network inspired by fuzzy integral," in *Advances in Neural Networks—ISNN 2005*, J. Wang, X. Liao, and Y. Zhang, Eds., vol. 3497 of *Lecture Notes in Computer Science*, pp. 26–31, Springer, Berlin, Germany, 2005.
- [28] Z. H. Huang and Y. D. Chen, "Log-linear model based behavior selection method for artificial fish swarm algorithm," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 685404, 10 pages, 2015.
- [29] J. Gao, H. M. Dong, and S. Jing, "Parameters optimization of synergetic recognition approach," *Chinese Journal of Electronics*, vol. 14, no. 2, pp. 192–197, 2005.
- [30] H. L. Wang, *The research of application of image recognition using synergetic neural network [Ph.D. thesis]*, Shanghai Jiao Tong University, Shanghai, China, 2000.

Research Article

Deep Convolutional Neural Networks for Hyperspectral Image Classification

Wei Hu,¹ Yangyu Huang,¹ Li Wei,¹ Fan Zhang,¹ and Hengchao Li^{2,3}

¹College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 10029, China

²Sichuan Provincial Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, Chengdu 610031, China

³Department of Aerospace Engineering Sciences, University of Colorado, Boulder, CO 80309, USA

Correspondence should be addressed to Wei Hu; thu.wei.hu@qq.com

Received 23 November 2014; Accepted 22 January 2015

Academic Editor: Tianfu Wu

Copyright © 2015 Wei Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, convolutional neural networks have demonstrated excellent performance on various visual tasks, including the classification of common two-dimensional images. In this paper, deep convolutional neural networks are employed to classify hyperspectral images directly in spectral domain. More specifically, the architecture of the proposed classifier contains five layers with weights which are the input layer, the convolutional layer, the max pooling layer, the full connection layer, and the output layer. These five layers are implemented on each spectral signature to discriminate against others. Experimental results based on several hyperspectral image data sets demonstrate that the proposed method can achieve better classification performance than some traditional methods, such as support vector machines and the conventional deep learning-based methods.

1. Introduction

The hyperspectral imagery (HSI) [1] is acquired by remote sensors, which are characterized in hundreds of observation channels with high spectral resolution. Taking advantages of the rich spectral information, numerous traditional classification methods, such as k -nearest-neighbors (k -nn), minimum distance, and logistic regression [2], have been developed. Recently, some more effective feature extraction methods as well as advanced classifiers were proposed, such as spectral-spatial classification [3] and local Fisher discriminant analysis [4]. In the current literatures, support vector machine (SVM) [5, 6] has been viewed as an efficient and stable method for hyperspectral classification tasks, especially for the small training sample sizes. SVM seeks to separate two-class data by learning an optimal decision hyperplane which best separates the training samples in a kernel-included high-dimensional feature space. Some extensions of SVM in hyperspectral image classification were presented to improve the classification performance [3, 7, 8].

Neural networks (NN), such as multilayer perceptron (MLP) [9] and radial basis function (RBF) [10] neural networks, have already been investigated for classification of remote sensing data. In [11], the authors proposed a semi-supervised neural network framework for large-scale HSI classification. Actually, in remote sensing classification tasks, SVM is superior to the traditional NN in terms of classification accuracy as well as computational cost. In [12], a deeper architecture of NN has been considered a powerful model for classification, whose classification performance is competitive to SVM.

Deep learning-based methods achieve promising performance in many fields. In deep learning, the convolutional neural networks (CNNs) [12] play a dominant role for processing visual-related problems. CNNs are biologically-inspired and multilayer classes of deep learning models that use a single neural network trained end to end from raw image pixel values to classifier outputs. The idea of CNNs was firstly introduced in [13], improved in [14], and refined and simplified in [15, 16]. With the large-scale sources

of training data and efficient implementation on GPUs, CNNs have recently outperformed some other conventional methods, even human performance [17], on many vision-related tasks, including image classification [18, 19], object detection [20], scene labeling [21], house number digit classification [22], and face recognition [23]. Besides vision tasks, CNNs have been also applied to other areas, such as speech recognition [24, 25]. The technique has been verified as an effective class of models for understanding visual image content, giving some state-of-the-art results on visual image classification and other visual-related problems. In [26], the authors presented DNN for HSI classification, in which stacked autoencoders (SAEs) were employed to extract discriminative features.

CNNs have been demonstrated to provide even better classification performance than the traditional SVM classifiers [27] and the conventional deep neural networks (DNNs) [18] in visual-related area. However, since CNNs have been only considered on visual-related problems, there are rare literatures on the technique with multiple layers for HSI classification. In this paper, we have found that CNNs can be effectively employed to classify hyperspectral data after building appropriate layer architecture. According to our experiments, we observe that the typical CNNs, such as LeNet-5 [14] with two convolutional layers, are actually not applicable for hyperspectral data. Alternatively, we present a simple but effective CNN architecture containing five layers with weights for supervised HSI classification. Several experiments demonstrate excellent performance of our proposed method compared to the classic SVM and the conventional deep learning architecture. As far as we know, it is the first time to employ the CNN with multiple layers for HSI classification.

The paper is organized as follows. In Section 2, we give a brief introduction to CNNs. In Section 3, the typical CNN architecture and the corresponding training process are presented. In Section 4, we experimentally compare the performance of our method with SVM and some neural networks with different architectures. Finally, we conclude by summarizing our results in Section 5.

2. CNNs

CNNs represent feed-forward neural networks which consist of various combinations of the convolutional layers, max pooling layers, and fully connected layers and exploit spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers. Convolutional layers alternate with max pooling layers mimicking the nature of complex and simple cells in mammalian visual cortex [28]. A CNN consists of one or more pairs of convolution and max pooling layers and finally ends with a fully connected neural networks. A typical convolutional network architecture is shown in Figure 1 [24].

In ordinary deep neural networks, a neuron is connected to all neurons in the next layer. CNNs are different from ordinary NN in that neurons in convolutional layer are only sparsely connected to the neurons in the next layer, based on their relative location. That is to say, in a fully connected

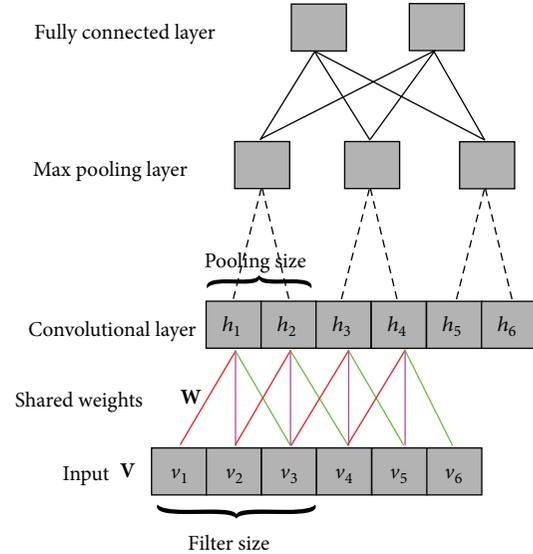


FIGURE 1: A typical CNN architecture consisting of a convolutional layer, a max pooling layer, and a fully connected layer.

DNNs, each hidden activation h_i is computed by multiplying the entire input V by weights W in that layer. However, in CNNs, each hidden activation is computed by multiplying a small local input against the weights W . The weights W are then shared across the entire input space, as shown in Figure 1. Neurons that belong to the same layer share the same weights. Weight sharing is a critical principle in CNNs since it helps reduce the total number of trainable parameters and leads to more efficient training and more effective model. A convolutional layer is usually followed by a max pooling layer.

Due to the replication of weights in a CNN, a feature may be detected across the input data. If an input image is shifted, the neuron detecting the feature is shifted as much. Pooling is used to make the features invariant from the location, and it summarizes the output of multiple neurons in convolutional layers through a pooling function. Typical pooling function is maximum. A max pooling function basically returns the maximum value from the input. Max pooling partitions the input data into a set of nonoverlapping windows and outputs the maximum value for each subregion and reduces the computational complexity for upper layers and provides a form of translation invariance. To be used for classification, the computation chain of a CNN ends in a fully connected network that integrates information across all locations in all the feature maps of the layer below.

Most of CNNs working in image recognition have the lower layers composed to alternate convolutional and max pooling layers, while the upper layers are fully connected traditional MLP NNs. For example, LeNet-5 is such a CNN architecture presented for handwritten digit recognition [14] firstly and then it is successfully used for solving other visual-related problems. However, LeNet-5 might not be directly employed for HSI classification, especially for small-size data sets, according to our experiments in Section 4. In this paper, we will explore what is the suitable architecture and strategy for CNN-based HSI classification.

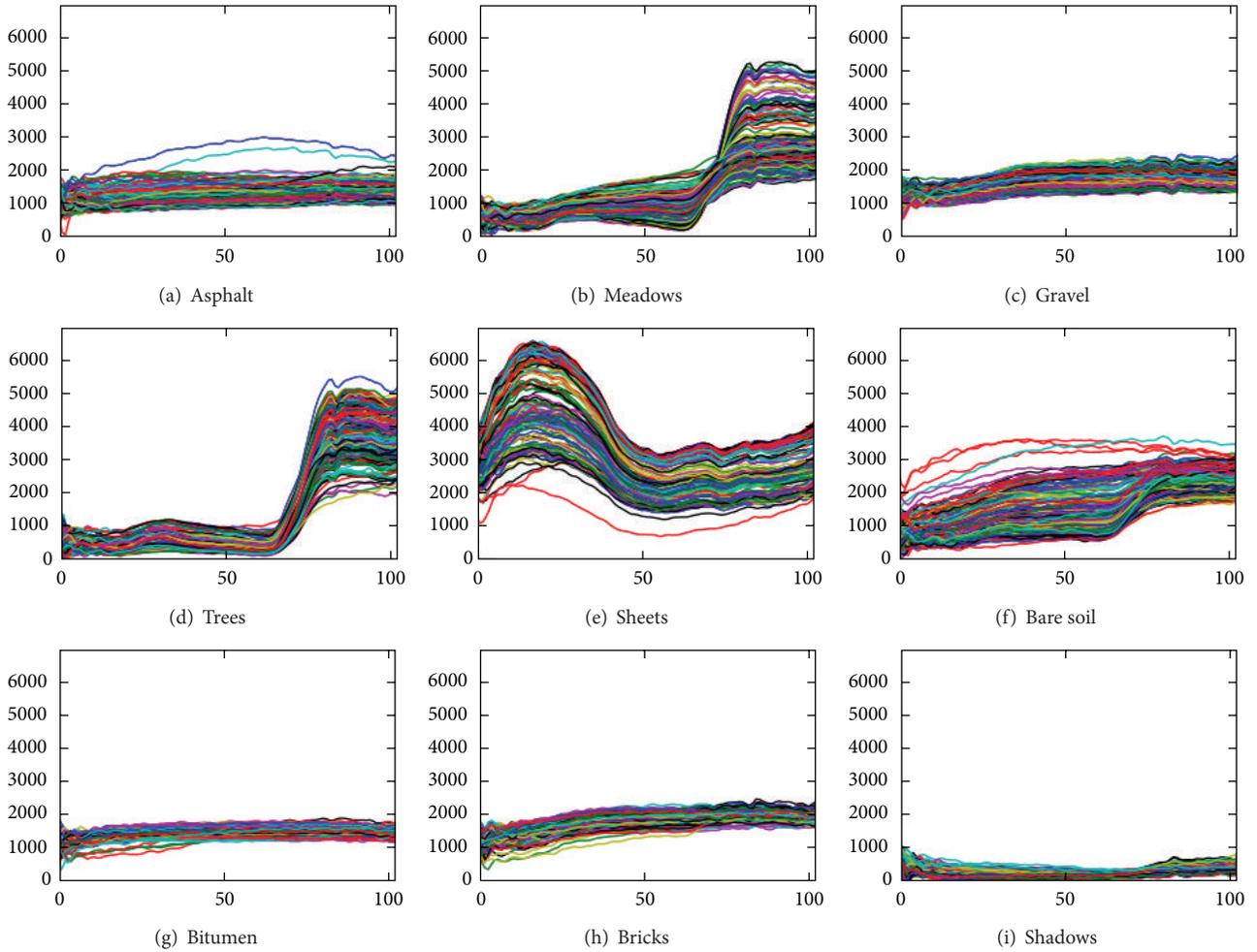


FIGURE 2: Spectral signatures of the 9 classes selected from University of Pavia data set with 103 channels/spectral bands.

3. CNN-Based HSI Classification

3.1. Applying CNNs to HSI Classification. The hierarchical architecture of CNNs is gradually proved to be the most efficient and successful way to learn visual representations. The fundamental challenge in such visual tasks is to model the intraclass appearance and shape variation of objects. The hyperspectral data with hundreds of spectral channels can be illustrated as 2D curves (1D array) as shown in Figure 2 (9 classes are selected from the University of Pavia data set). We can see that the curve of each class has its own visual shape which is different from other classes, although it is relatively difficult to distinguish some classes with human eye (e.g., gravel and self-blocking bricks). We know that CNNs can achieve competitive and even better performance than human being in some visual problems, and its capability inspires us to study the possibility of applying CNNs for HSI classification using the spectral signatures.

3.2. Architecture of the Proposed CNN Classifier. The CNN varies in how the convolutional and max pooling layers are realized and how the nets are trained. As illustrated in

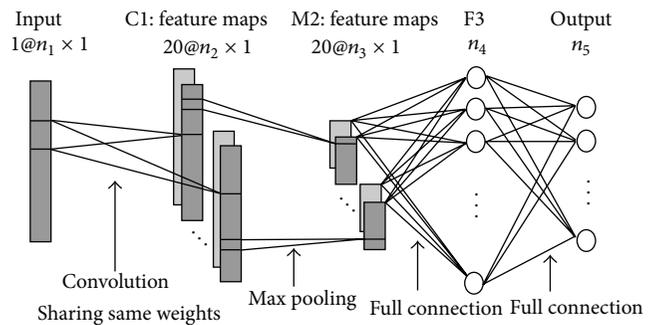


FIGURE 3: The architecture of the proposed CNN classifier. The input represents a pixel spectral vector, followed by a convolution layer and a max pooling layer in turns to compute a set of 20 feature maps classified with a fully connected network.

Figure 3, the net contains five layers with weights, including the input layer, the convolutional layer C1, the max pooling layer M2, the full connection layer F3, and the output layer. Assuming θ represents all the trainable parameters (weight

values), $\theta = \{\theta_i\}$ and $i = 1, 2, 3, 4$, where θ_i is the parameter set between the $(i - 1)$ th and the i th layer.

In HSI, each HSI pixel sample can be regarded as a 2D image whose height is equal to 1 (as 1D audio inputs in speech recognition). Therefore, the size of the input layer is just $(n_1, 1)$, and n_1 is the number of bands. The first hidden convolutional layer C1 filters the $n_1 \times 1$ input data with 20 kernels of size $k_1 \times 1$. Layer C1 contains $20 \times n_2 \times 1$ nodes, and $n_2 = n_1 - k_1 + 1$. There are $20 \times (k_1 + 1)$ trainable parameters between layer C1 and the input layer. The max pooling layer M2 is the second hidden layer, and the kernel size is $(k_2, 1)$. Layer M2 contains $20 \times n_3 \times 1$ nodes, and $n_3 = n_2/k_2$. There is no parameter in this layer. The fully connected layer F3 has n_4 nodes and there are $(20 \times n_3 + 1) \times n_4$ trainable parameters between this layer and layer M2. The output layer has n_5 nodes, and there are $(n_4 + 1) \times n_5$ trainable parameters between this layer and layer F3. Therefore, the architecture of our proposed CNN classifier totally has $20 \times (k_1 + 1) + (20 \times n_3 + 1) \times n_4 + (n_4 + 1) \times n_5$ trainable parameters.

Classifying a specified HSI pixel requires the corresponding CNN with the aforementioned parameters, where n_1 and n_5 are the spectral channel size and the number of output classes of the data set, respectively. In our experiments, k_1 is better to be $\lceil n_1/9 \rceil$, and $n_2 = n_1 - k_1 + 1$. n_3 can be any number between 30 and 40, and $k_2 = \lceil n_2/n_3 \rceil$. n_4 is set to be 100. These choices might not be the best but are effective for general HSI data.

In our architecture, layer C1 and M2 can be viewed as a trainable feature extractor to the input HSI data, and layer F3 is a trainable classifier to the feature extractor. The output of subsampling is the actual feature of the original data. In our proposed CNN structure, 20 features can be extracted from each original hyperspectral, and each feature has n_3 dimensions.

Our architecture has some similarities to architectures that applied CNN for frequency domain signal in speech recognition [24, 25]. We think it is caused by the similarity between 1D input of speech spectrum and hyperspectral data. Different from [24, 25], our network varies according to the spectral channel size and the number of output classes of input HSI data.

3.3. Training Strategies. Here, we introduce how to learn the parameter space of the proposed CNN classifier. All the trainable parameters in our CNN should be initialized to be a random value between -0.05 and 0.05 . The training process contains two steps: forward propagation and back propagation. The forward propagation aims to compute the actual classification result of the input data with current parameters. The back propagation is employed to update the trainable parameters in order to make the discrepancy between the actual classification output and the desired classification output as small as possible.

3.3.1. Forward Propagation. Our $(L + 1)$ -layer CNN network ($L = 4$ in this work) consists of n_1 input units in layer INPUT, n_5 output units in layer OUTPUT, and several so-called hidden units in layers C2, M3, and F4. Assuming \mathbf{x}_i is

the input of the i th layer and the output of the $(l - 1)$ th layer, then we can compute \mathbf{x}_{i+1} as

$$\mathbf{x}_{i+1} = f_i(\mathbf{u}_i), \quad (1)$$

where

$$\mathbf{u}_i = \mathbf{W}_i^T \mathbf{x}_i + \mathbf{b}_i, \quad (2)$$

and \mathbf{W}_i^T is a weight matrix of the i th layer acting on the input data, and \mathbf{b}_i is an additive bias vector for the i th layer. $f_i(\cdot)$ is the activation function of the i th layer. In our designed architecture, we choose the hyperbolic tangent function $\tanh(\mathbf{u})$ as the activation function in layer C1 and layer F3. The maximum function $\max(\mathbf{u})$ is used in layer M2. Since the proposed CNN classifier is a multiclass classifier, the output of layer F3 is fed to n_5 way softmax function which produces a distribution over the n_5 class labels, and the softmax regression model is defined as

$$\mathbf{y} = \frac{1}{\sum_{k=1}^{n_5} e^{\mathbf{W}_{L,k}^T \mathbf{x}_L + \mathbf{b}_{L,k}}} \begin{bmatrix} e^{\mathbf{W}_{L,1}^T \mathbf{x}_L + \mathbf{b}_{L,1}} \\ e^{\mathbf{W}_{L,2}^T \mathbf{x}_L + \mathbf{b}_{L,2}} \\ \vdots \\ e^{\mathbf{W}_{L,n_5}^T \mathbf{x}_L + \mathbf{b}_{L,n_5}} \end{bmatrix}. \quad (3)$$

The output vector $\mathbf{y} = \mathbf{x}_{L+1}$ of the layer OUTPUT denotes the final probability of all the classes in the current iteration.

3.3.2. Back Propagation. In the back propagation stage, the trainable parameters are updated by using the gradient descent method. It is realized by minimizing a cost function and computing the partial derivative of the cost function with respect to each trainable parameter [29]. The loss function used in this work is defined as

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{j=1}^{n_5} 1\{j = \mathbf{Y}^{(i)}\} \log(\mathbf{y}_j^{(i)}), \quad (4)$$

where m is the number of training samples. \mathbf{Y} is the desired output. $\mathbf{y}_j^{(i)}$ is the j th value of the actual output $\mathbf{y}^{(i)}$ (see (3)) of the i th training sample and is a vector whose size is n_5 . In the desired output $\mathbf{Y}^{(i)}$ of the i th sample, the probability value of the labeled class is 1, and the probability values of other classes are 0. $1\{j = \mathbf{Y}^{(i)}\}$ means, if j is equal to the desired label of the i th training sample, its value is 1; otherwise, its value is 0. We add a minus sign to the front of $J(\theta)$ in order to make the computation more convenient.

The derivative of the loss function with respect to \mathbf{u}_i is

$$\delta_i = \frac{\partial J}{\partial \mathbf{u}_i} = \begin{cases} -(\mathbf{Y} - \mathbf{y}) \circ f'(\mathbf{u}_i), & i = L \\ (\mathbf{W}_i^T \delta_{i+1}) \circ f'(\mathbf{u}_i), & i < L, \end{cases} \quad (5)$$

where \circ denotes element-wise multiplication. $f'(\mathbf{u}_i)$ can be easily represented as

$$f'(\mathbf{u}_i) = \begin{cases} (1 - f(\mathbf{u}_i)) \circ (1 + f(\mathbf{u}_i)), & i = 1, 3 \\ \text{null}, & i = 2 \\ f(\mathbf{u}_i) \circ (1 - f(\mathbf{u}_i)), & i = 4. \end{cases} \quad (6)$$

```

▷ Constructing the CNN Model
function INITCNNMODEL ( $\theta$ , [ $n_{1-5}$ ])
  layerType = [convolution, max-pooling, fully-connected, fully-connected];
  layerActivation = [tanh(), max(), tanh(), softmax()]
  model = new Model();
  for  $i = 1$  to 4 do
    layer = new Layer();
    layer.type = layerType[ $i$ ];
    layer.inputSize =  $n_i$ 
    layer.neurons = new Neuron [ $n_{i+1}$ ];
    layer.params =  $\theta_i$ ;
    model.addLayer(layer);
  end for
  return model;
end function
▷ Training the CNN Model
Initialize learning rate  $\alpha$ , number of max iteration  $ITER_{max}$ , min error  $ERR_{min}$ , training
batches  $BATCHES_{training}$ , batch size  $SIZE_{batch}$ , and so on;
Compute  $n_2, n_3, n_4, k_1, k_2$ , according to  $n_1$  and  $n_5$ ;
Generate random weights  $\theta$  of the CNN;
cnnModel = InitCNNModel( $\theta$ , [ $n_{1-5}$ ]);
iter = 0; err = +inf;
while  $err > ERR_{min}$  and  $iter < ITER_{max}$  do
  err = 0;
  for  $batch = 1$  to  $BATCHES_{training}$  do
    [ $\nabla_{\theta} J(\theta), J(\theta)$ ] = cnnModel.train (TrainingDatas, TrainingLabels), as (4) and (8);
    Update  $\theta$  using (7);
    err = err + mean( $J(\theta)$ );
  end for
  err = err/ $BATCHES_{training}$ ;
  iter++;
end while
Save parameters  $\theta$  of the CNN;

```

ALGORITHM 1: Our CNN-based method.

Therefore, on each iteration, we would perform the update

$$\theta = \theta - \alpha \cdot \nabla_{\theta} J(\theta) \quad (7)$$

for adjusting the trainable parameters, where α is the learning factor ($\alpha = 0.01$ in our implementation), and

$$\nabla_{\theta} J(\theta) = \left\{ \frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \dots, \frac{\partial J}{\partial \theta_L} \right\}. \quad (8)$$

We know that θ_i contains \mathbf{W}_i and \mathbf{b}_i , and

$$\frac{\partial J}{\partial \theta_i} = \left\{ \frac{\partial J}{\partial \mathbf{W}_i}, \frac{\partial J}{\partial \mathbf{b}_i} \right\}, \quad (9)$$

where

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{W}_i} &= \frac{\partial J}{\partial \mathbf{u}_i} \circ \frac{\partial \mathbf{u}_i}{\partial \mathbf{W}_i} = \frac{\partial J}{\partial \mathbf{u}_i} \circ \mathbf{x}_i = \delta_i \circ \mathbf{x}_i, \\ \frac{\partial J}{\partial \mathbf{b}_i} &= \frac{\partial J}{\partial \mathbf{u}_i} \circ \frac{\partial \mathbf{u}_i}{\partial \mathbf{b}_i} = \frac{\partial J}{\partial \mathbf{u}_i} = \delta_i. \end{aligned} \quad (10)$$

With an increasing number of training iteration, the return of the cost function is smaller, which indicates that the actual output is closer to the desired output. The iteration stops when the discrepancy between them is small enough. We use average sum of squares to represent the discrepancy. Finally, the trained CNN is ready for HSI classification. The summary of the proposed algorithm is shown in Algorithm 1.

3.4. Classification. Since the architecture and all corresponding trainable parameters are specified, we can build the CNN classifier and reload saved parameters for classifying HSI data. The classification process is just like the forward propagation step, in which we can compute the classification result as (3).

4. Experiments

All the programs are implemented using Python language and Theano [30] library. Theano is a Python library that makes us easily define, optimize, and evaluate mathematical expressions involving multidimensional arrays efficiently and conveniently on GPUs. The results are generated on a PC

TABLE 1: Number of training and test samples used in the Indian Pines data set.

Number	Class	Training	Test
1	Corn-notill	200	1228
2	Corn-mintill	200	630
3	Grass-pasture	200	283
4	Hay-windrowed	200	278
5	Soybean-notill	200	772
6	Soybean-mintill	200	2255
7	Soybean-clean	200	393
8	Woods	200	1065
Total		1600	6904

equipped with an Intel Core i7 with 2.8 GHz and Nvidia GeForce GTX 465 graphics card.

4.1. The Data Sets. Three hyperspectral data, including Indian Pines, Salinas, and University of Pavia scenes, are employed to evaluate the effectiveness of the proposed method. For all the data, we randomly select 200 labeled pixels per class for training and all other pixels in the ground truth map for test. Development data are derived from the available training data by further dividing them into training and testing samples for tuning the parameters of the proposed CNN classifier. Furthermore, each pixel is scaled to $[-1.0, +1.0]$ uniformly.

The Indian Pines data set was gathered by Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in northwestern Indiana. There are 220 spectral channels in the 0.4 to 2.45 μm region of the visible and infrared spectrum with a spatial resolution of 20 m. From the statistical viewpoint, we discard some classes which only have few labeled samples and select 8 classes for which the numbers of training and testing samples are listed in Table 1. The layer parameters of this data set in the proposed CNN classifier are set as follows: $n_1 = 220$, $k_1 = 24$, $n_2 = 197$, $k_2 = 5$, $n_3 = 40$, $n_4 = 100$, and $n_5 = 8$, and the number of total trainable parameters in the data set is 81408.

The second data employed was also collected by the AVIRIS sensor, capturing an area over Salinas Valley, California, with a spatial resolution of 3.7 m. The image comprises 512×217 pixels with 220 bands. It mainly contains vegetables, bare soils, and vineyard fields (http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes). There are also 16 different classes, and the numbers of training and testing samples are listed in Table 2. The layer parameters of this data set in our CNN are set to be $n_1 = 224$, $k_1 = 24$, $n_2 = 201$, $k_2 = 5$, $n_3 = 40$, $n_4 = 100$, and $n_5 = 16$, and the number of total trainable parameters in the data set is 82216.

The University of Pavia data set was collected by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor. The image scene, with a spatial coverage of 610×340 pixels covering the city of Pavia, Italy, was collected under the HySens project managed by DLR (the German Aerospace Agency). The data set has 103 spectral bands prior to water

TABLE 2: Number of training and test samples used in the Salinas data set.

Number	Class	Training	Test
1	Broccoli green weeds 1	200	1809
2	Broccoli green weeds 2	200	3526
3	Fallow	200	1776
4	Fallow rough plow	200	1194
5	Fallow smooth	200	2478
6	Stubble	200	3759
7	Celery	200	3379
8	Grapes untrained	200	11071
9	Soil vineyard develop	200	6003
10	Corn senesced green weeds	200	3078
11	Lettuce romaine, 4 wk	200	868
12	Lettuce romaine, 5 wk	200	1727
13	Lettuce romaine, 6 wk	200	716
14	Lettuce romaine, 7 wk	200	870
15	Vineyard untrained	200	7068
16	Vineyard vertical trellis	200	1607
Total		3200	50929

TABLE 3: Number of training and test samples used in University of Pavia data set.

Number	Class	Training	Test
1	Asphalt	200	6431
2	Meadows	200	18449
3	Gravel	200	1899
4	Trees	200	2864
5	Sheets	200	1145
6	Bare soil	200	4829
7	Bitumen	200	1130
8	Bricks	200	3482
9	Shadows	200	747
Total		1800	40976

band removal. It has a spectral coverage from 0.43 to 0.86 μm and a spatial resolution of 1.3 m. Approximately 42776 labeled pixels with 9 classes are from the ground truth map, and the numbers of training and testing samples are shown in Table 3. The layer parameters of this data set in our CNN are set to be $n_1 = 103$, $k_1 = 11$, $n_2 = 93$, $k_2 = 3$, $n_3 = 30$, $n_4 = 100$, and $n_5 = 9$, and the number of total trainable parameters in the data set is 61249.

4.2. Results and Comparisons. Table 4 provides the comparison of classification performance between the proposed method and the traditional SVM classifier. SVM with RBF kernel is implemented using the `libsvm` package (<http://www.csie.ntu.edu.tw/~cjlin/libsvm>); cross validation is also employed to determine the related parameters, and all optimal ones are used in following experiments. It is obvious that our proposed method has better performance

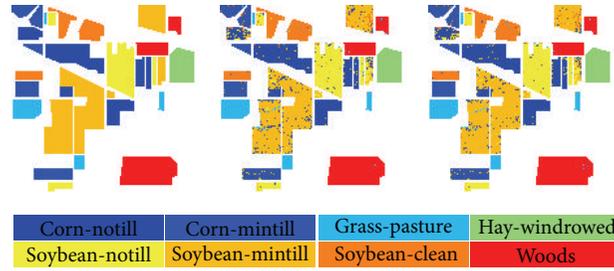


FIGURE 4: RGB composition maps resulting from classification for the Indian Pines data set. From left to right: ground truth, RBF-SVM, and the proposed method.

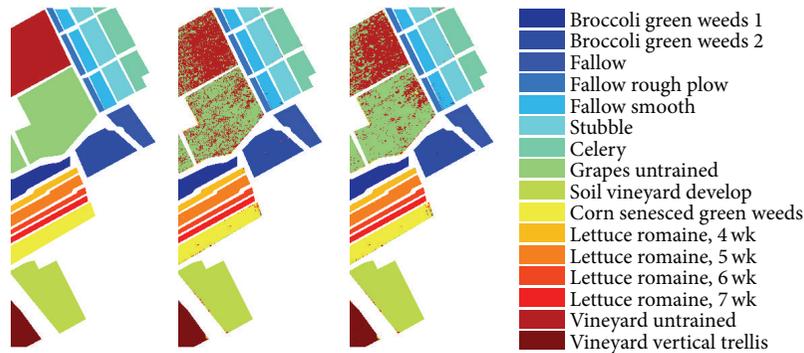


FIGURE 5: RGB composition maps resulting from classification for the Salinas data set. From left to right: ground truth, RBF-SVM, and the proposed method.

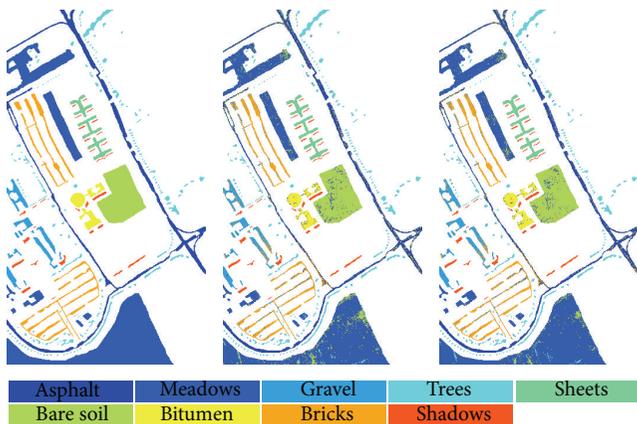


FIGURE 6: Thematic maps resulting from classification for University of Pavia data set. From left to right: ground truth, RBF-SVM, and the proposed method.

(approximate 2% gain) than SVM classifier using all the three data sets. Figures 4, 5, and 6 illustrate the corresponding classification maps obtained with our proposed method and RBF-SVM classifier. Furthermore, compared with RBF-SVM, the proposed CNN classifier has higher classification accuracy not only for the overall data set but also for almost all the specific classes as shown in Figure 7.

Figure 8 further illustrates the relationship between classification accuracies and the training time (the test time is

TABLE 4: Comparison of results between the proposed CNN and RBF-SVM using three data sets.

Data set	The proposed CNN	RBF-SVM
Indian Pines	90.16%	87.60%
Salinas	92.60%	91.66%
University of Pavia	92.56%	90.52%

TABLE 5: Results of comparison with different neural networks on the Indian Pines data set.

Method	Training time	Testing time	Accuracy
Two-layer NN	2800 s	1.65 s	86.49%
DNN	6500 s	3.21 s	87.93%
LeNet-5	5100 s	2.34 s	88.27%
Our CNN	4300 s	1.98 s	90.16%

also included) for three experimental data sets. With the increasing of training time, the classification accuracy of each data can reach over 90%. We must admit that the training process is relatively time-consuming to achieve good performance; however, the proposed CNN classifier shares the same advantages (e.g., fast on testing) of deep learning algorithms (see Table 5). Moreover, our implementation of CNN could be improved greatly on efficiency, or we can use other CNN frameworks, such as Caffe [31], to reduce training

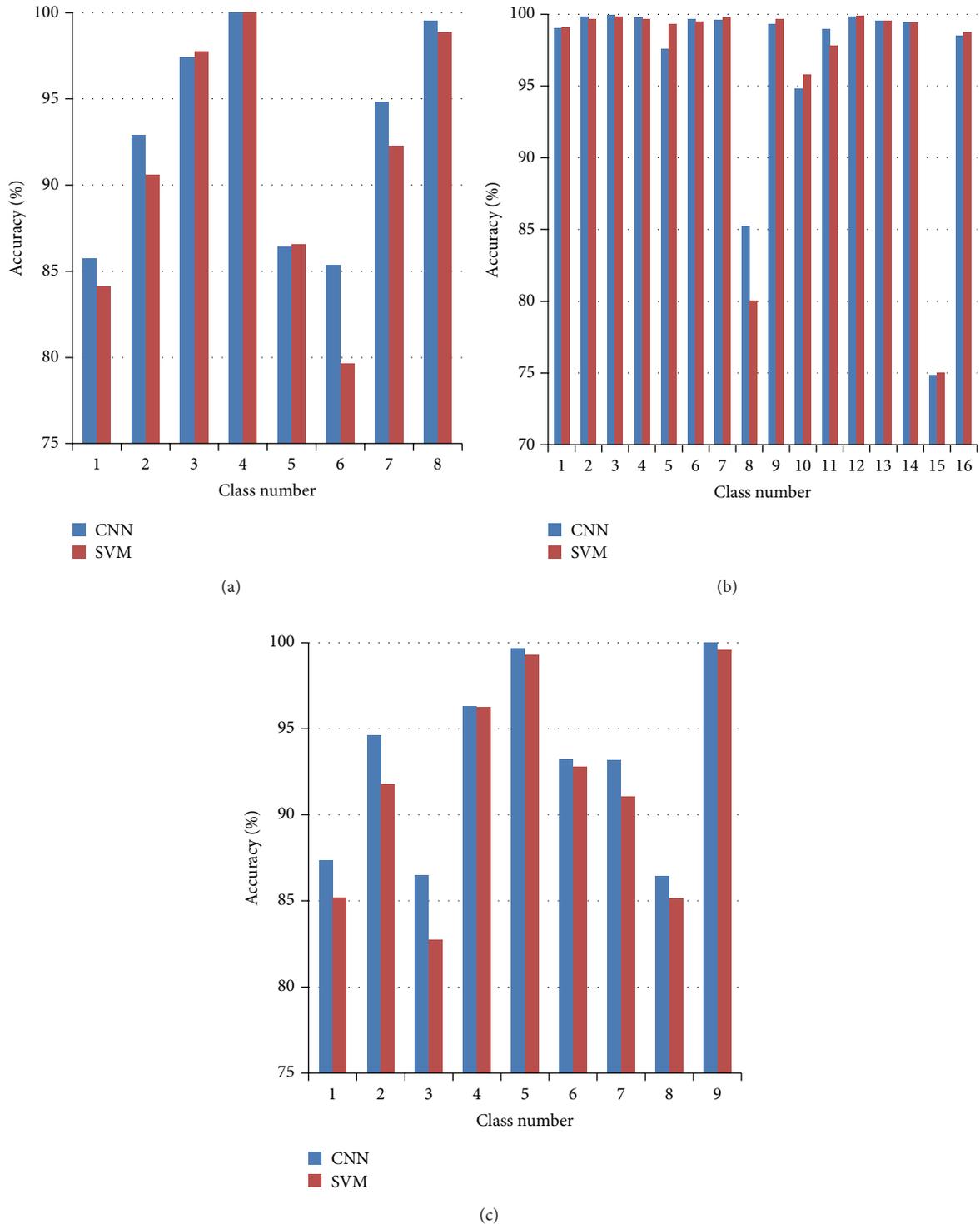


FIGURE 7: Classification accuracies of all the classes for experimental data sets. From (a) to (c): Indian Pines, Salinas, and University of Pavia. The class number is corresponding to the first column in Tables 1, 2, and 3.

and test time. According to our experiments, it takes only 5 minutes to achieve 90% accuracy on MNIST dataset [32] by using Caffe compared to more than 120 minutes by using our implemented framework.

Figure 9 illustrates the relationship between cost value (see (4)) and the training time for the University of Pavia data set. The value of the loss function is reduced with an increasing number of training iteration, which demonstrates

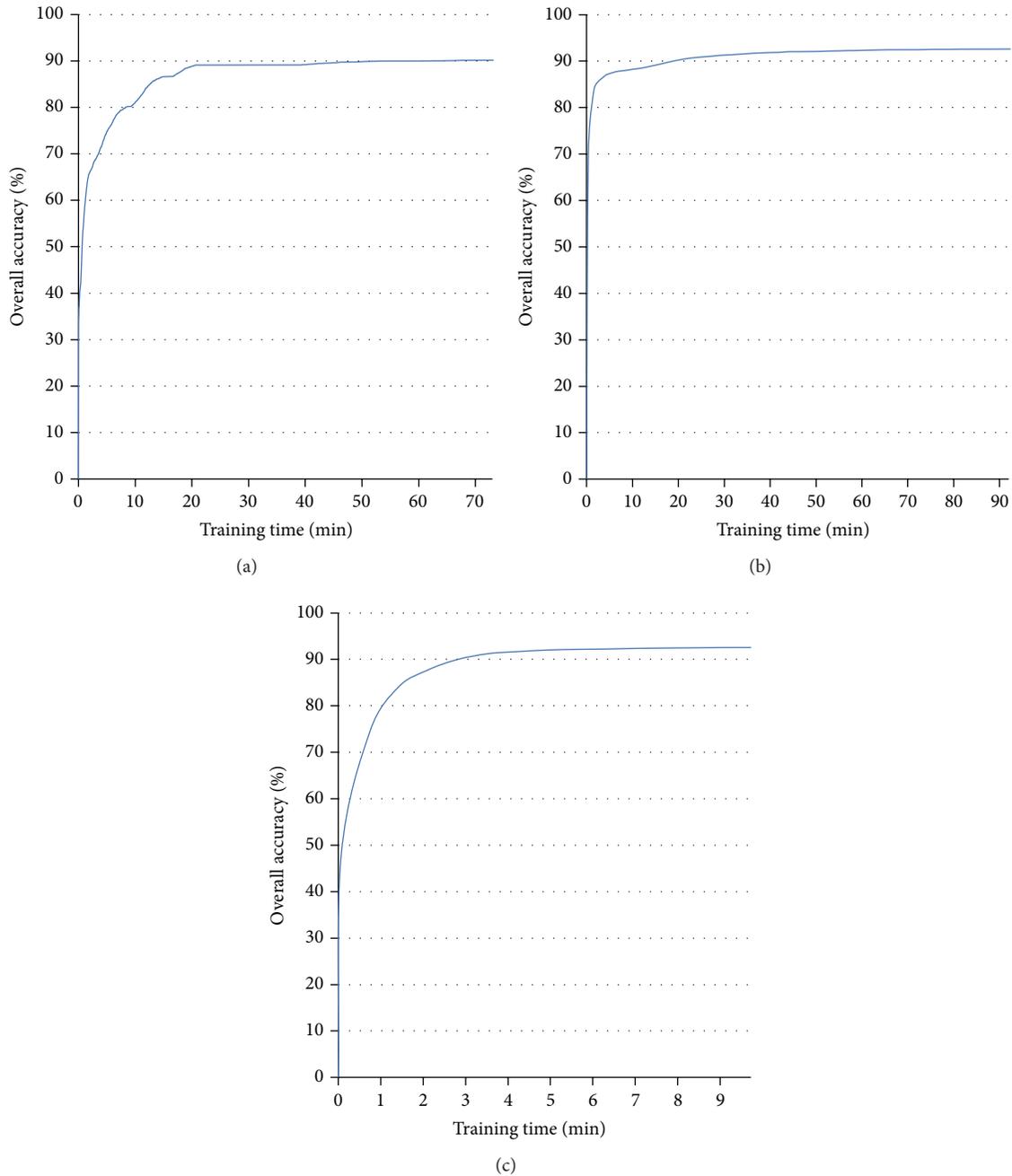


FIGURE 8: Classification accuracies versus the training time for experimental data sets. From (a) to (c): Indian Pines, Salinas, and University of Pavia. Note that the test time is also included in the training time.

the convergence of our network with only 200 training samples for each class. Moreover, the cost value is still reduced after 5-minute training, but the corresponding test accuracy is relatively stable (see Figure 8(a)), which indicates the overfitting problem in this network.

To further verify that the proposed classifier is suitable for classifying data sets with limited training samples, we also compare our CNN with RBF-SVM under different training sizes on the University of Pavia data set as shown in Figure 10. It is obvious that our proposed CNN consistently provides

higher accuracy than SVM. However, although the conventional deep learning-based method [26] can outperform the SVM classifier, it requires plenty of training samples for constructing autoencoders.

To demonstrate the relationship between classification accuracies and the visual differences of curve shapes (see Figure 2), we present the detailed accuracies of our proposed CNN classifiers for the University of Pavia data set in Table 6. In the table, the cell in the i th row, j th column means the percentage of the i th class samples (according to ground

TABLE 6: The detailed classification accuracies of all the classes for University of Pavia data set.

	Asphalt	Meadows	Gravel	Trees	Sheets	Bare soil	Bitumen	Bricks	Shadows
Asphalt	87.34%	0.26%	2.32%	0.00%	0.19%	0.37%	6.25%	3.25%	0.02%
Meadows	0.00%	94.63%	0.02%	1.26%	0.00%	4.03%	0.00%	0.06%	0.00%
Gravel	0.53%	0.47%	86.47%	0.00%	0.00%	0.00%	0.05%	12.43%	0.05%
Trees	0.00%	2.67%	0.00%	96.29%	0.03%	1.01%	0.00%	0.00%	0.00%
Sheets	0.00%	0.09%	0.00%	0.00%	99.65%	0.26%	0.00%	0.00%	0.00%
Bare soil	0.12%	6.15%	0.00%	0.10%	0.08%	93.23%	0.00%	0.31%	0.00%
Bitumen	6.37%	0.00%	0.35%	0.00%	0.09%	0.00%	93.19%	0.00%	0.00%
Bricks	1.90%	0.20%	10.48%	0.00%	0.06%	0.60%	0.34%	86.42%	0.00%
Shadows	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%

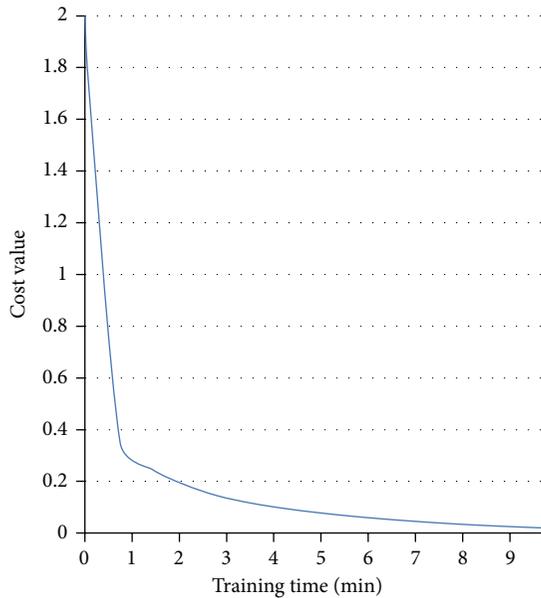


FIGURE 9: Cost value versus the training time for University of Pavia data sets.

truth) which is classified to the j th class. For example, 87.34% of class Asphalt samples are classified correctly, but 6.25% of class Asphalt samples are wrongly classified to class Bitumen. The percentages on diagonal line are just the classification accuracies of corresponding classes. As for one class, the more unique the corresponding curve shape is, the higher accuracy the proposed CNN classifier can achieve (check the class Shadow and class Sheets in Figure 2 and Table 6). The more similar two curves are, the higher opportunity they are wrongly classified to each other (check the class Gravel and class Bricks in Figure 2 and Table 6). Furthermore, the excellent performance verifies that the proposed CNN classifier has discriminative capability to extract subtle visual features, which is even superior to human vision for classifying complex curve shapes.

Finally, we also implement three other types of neural network architectures for the Indian Pines data set using the same training and test samples. The first one is a simple architecture with only two fully connected layers beside the input layer. The second one is LeNet-5 which is a classic CNN architecture with two convolutional layers. The third one is

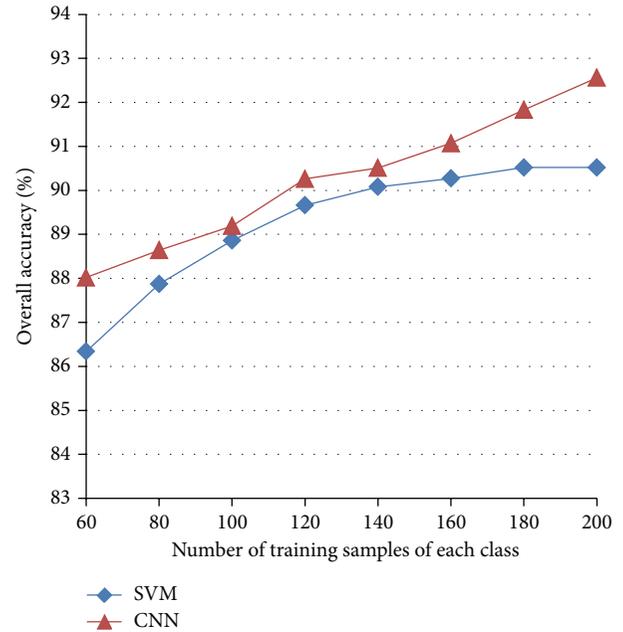


FIGURE 10: Classification accuracies versus numbers of training samples (each class) for the University of Pavia data sets.

a conventional deep neural networks (DNNs) with 3 hidden fully connected layers (a 220-60-40-20-8 architecture as suggested in [26]). The classification performance is summarized in Table 5. From Table 5, we can see that our CNN classifier achieves the highest accuracy with competitive training and testing computational cost. LeNet-5 and DNNs cost more time to train models due to their complex architecture, but limited training samples restrict their capabilities of classification (only 20% samples selected for testing in [26] compared with 95% in our experiment). Another reason for the difficulty that deeper CNNs and DNNs face to achieve higher accuracies could be that the HSI lacks the type of high frequency signal commonly seen in the computer vision domain (see Figure 2).

5. Conclusion and Future Work

In this paper, we proposed a novel CNN-based method for HSI classification, inspired by our observation that HSI classification can be implemented via human vision.

Compared with SVM-based classifier and conventional DNN-based classifier, the proposed method could achieve higher accuracy using all the experimental data sets, even with a small number of training samples.

Our work is an exploration of using CNNs for HSI classification and has excellent performance. The architecture of our proposed CNN classifier only contains one convolutional layer and one fully connected layer, due to the small number of training samples. In the future, a network architecture called a Siamese Network [33] might be used, which has been proved to be robust in the situation where the number of training samples per category is small. Some techniques, such as Dropout [34], can also be used to alleviate the overfitting problem caused by limited training samples. Furthermore, recent researches in deep learning have indicated that unsupervised learning can be employed to train CNNs, reducing the requirement of labeled samples significantly. Deep learning, especially deep CNNs, should have great potentiality for HSI classification in the future. Moreover, in the current work, we do not consider the spatial correlation and only concentrate on the spectral signatures. We believe that some spatial-spectral techniques also can be applied to further improve the CNN-based classification. At last, we plan to employ efficient deep CNN frameworks, such as Caffe, to improve our computing performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported jointly by the National Natural Science Foundation of China (nos. 61371165 and 61302164), the 973 Program of China (no. 2011CB706900), the Program for New Century Excellent Talents in University under Grant no. NCET-11-0711, and the Interdisciplinary Research Project in Beijing University of Chemical Technology. Wei Hu and Fan Zhang are also supported by the Beijing Higher Education Young Elite Teacher Project under Grant nos. YETP0501 and YETP0500, respectively.

References

- [1] D. Landgrebe, "Hyperspectral image data analysis," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 17–28, 2002.
- [2] G. M. Foody and A. Mathur, "A relative evaluation of multiclass image classification by support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 6, pp. 1335–1343, 2004.
- [3] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectral-spatial classification of hyperspectral imagery based on partitioned clustering techniques," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 8, pp. 2973–2987, 2009.
- [4] W. Li, S. Prasad, F. James, and B. Lour, "Locality-preserving dimensionality reduction and classification for hyperspectral image analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1185–1198, 2012.
- [5] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.
- [6] J. A. Gualtieri and S. Chettri, "Support vector machines for classification of hyperspectral data," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '00)*, vol. 2, pp. 813–815, IEEE, July 2000.
- [7] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: a review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 3, pp. 247–259, 2011.
- [8] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 844–856, 2013.
- [9] P. M. Atkinson and A. R. L. Tatnall, "Introduction neural networks in remote sensing," *International Journal of Remote Sensing*, vol. 18, no. 4, pp. 699–709, 1997.
- [10] L. Bruzzone and D. F. Prieto, "A technique for the selection of kernel-function parameters in RBF neural networks for classification of remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 1179–1184, 1999.
- [11] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2271–2282, 2010.
- [12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [13] K. Fukushima, "Neocognitron: a hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, no. 2, pp. 119–130, 1988.
- [14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [15] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11)*, vol. 22, pp. 1237–1242, July 2011.
- [16] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proceedings of the 7th International Conference on Document Analysis and Recognition*, vol. 2, pp. 958–963, IEEE Computer Society, Edinburgh, UK, August 2003.
- [17] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proceedings of the International Joint Conference on Neural Network (IJCNN '11)*, pp. 2809–2813, IEEE, August 2011.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS '12)*, pp. 1097–1105, 2012.
- [19] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 3642–3649, IEEE, June 2012.
- [20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, IEEE, June 2014.

- [21] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.
- [22] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, pp. 3288–3291, IEEE, November 2012.
- [23] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 1701–1708, Columbus, Ohio, USA, June 2014.
- [24] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 8614–8618, IEEE, Vancouver, Canada, May 2013.
- [25] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 4277–4280, IEEE, March 2012.
- [26] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.
- [27] I. Sutskever and G. E. Hinton, "Deep, narrow sigmoid belief networks are universal approximators," *Neural Computation*, vol. 20, no. 11, pp. 2629–2636, 2008.
- [28] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of Physiology*, vol. 195, no. 1, pp. 215–243, 1968.
- [29] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, pp. 9–48, Springer, Berlin, Germany, 2012.
- [30] J. Bergstra, F. Bastien, O. Breuleux et al., "Theano: deep learning on GPUs with python," in *Proceedings of the NIPS 2011, Big Learning Workshop*, pp. 712–721, Granada, Spain, December 2011.
- [31] Y. Jia, E. Shelhamer, J. Donahue et al., "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, pp. 675–678, ACM, Orlando, Fla, USA, November 2014.
- [32] Y. LeCun, C. Cortes, and C. J. Burges, "The MNIST database of handwritten digits," 1998, <http://yann.lecun.com/exdb/mnist/>.
- [33] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 539–546, IEEE, June 2005.
- [34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

Research Article

Urban Land Use and Land Cover Classification Using Remotely Sensed SAR Data through Deep Belief Networks

Qi Lv,^{1,2} Yong Dou,^{1,2} Xin Niu,^{1,2} Jiaqing Xu,² Jinbo Xu,² and Fei Xia³

¹Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha 410073, China

²School of Computer, National University of Defense Technology, Changsha 410073, China

³Electronic Engineering College, Naval University of Engineering, Wuhan 430033, China

Correspondence should be addressed to Qi Lv; lvqi@nudt.edu.cn

Received 13 November 2014; Accepted 29 January 2015

Academic Editor: Tianfu Wu

Copyright © 2015 Qi Lv et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Land use and land cover (LULC) mapping in urban areas is one of the core applications in remote sensing, and it plays an important role in modern urban planning and management. Deep learning is springing up in the field of machine learning recently. By mimicking the hierarchical structure of the human brain, deep learning can gradually extract features from lower level to higher level. The Deep Belief Networks (DBN) model is a widely investigated and deployed deep learning architecture. It combines the advantages of unsupervised and supervised learning and can archive good classification performance. This study proposes a classification approach based on the DBN model for detailed urban mapping using polarimetric synthetic aperture radar (PolSAR) data. Through the DBN model, effective contextual mapping features can be automatically extracted from the PolSAR data to improve the classification performance. Two-date high-resolution RADARSAT-2 PolSAR data over the Great Toronto Area were used for evaluation. Comparisons with the support vector machine (SVM), conventional neural networks (NN), and stochastic Expectation-Maximization (SEM) were conducted to assess the potential of the DBN-based classification approach. Experimental results show that the DBN-based method outperforms three other approaches and produces homogenous mapping results with preserved shape details.

1. Introduction

Urban land use and land cover (LULC) mapping is one of the core applications in remote sensing. Up-to-date LULC maps obtained by classifying remotely sensed data are essential to modern urban planning and management. In many remote sensing systems, the synthetic aperture radar (SAR) has long been recognized as an effective tool for urban analysis, as it is less influenced by solar illumination or weather conditions in contrast to optical or infrared sensors [1]. Since more scattering information can be collected in multipolarizations, polarimetric SAR (PolSAR) data have been increasingly used for urban LULC classification [2–4].

Nevertheless, most studies about urban mapping using SAR or PolSAR data are limited in identifying the urban extent or mapping very few urban classes. Few studies have focused on detailed urban mapping using SAR data. The

difficulty in detailed urban mapping using SAR data is mainly due to the complexity of the urban environment. The urban environment is comprised of various natural and man-made objects with several kinds of materials, different orientations, various shapes and sizes, and so forth, which complicates the interpretation of SAR images. Problems can also originate from the nature of polarimetric SAR imaging such as inherent SAR speckle or geometry distortions such as shadow and layover [1, 2]. As a consequence, detailed urban mapping using high resolution SAR data is still a challenging task.

Regarding the method of urban land cover mapping, approaches can be generally divided into pixel-based or object-based classification. Object-based methods, which directly explore the contextual information to improve the mapping accuracy, have been increasingly employed recently [5]. By using object-based approaches, shape characteristics and inner statistics of segmented objects can be used as

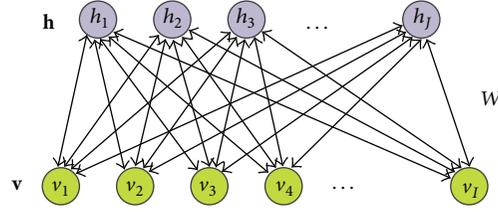


FIGURE 1: Schematic of an RBM with I visible units and J hidden units, where W is the weight matrix.

classification features [6–8]. However, the ideal segmentation on urban areas using SAR data is often difficult to achieve. Pixel-based approaches have been traditionally used for coarse-resolution SAR data with reasonable results. However, when dealing with high-resolution SAR data, the pixel-by-pixel approach is usually limited because of the speckles and increased interclass variance [9]. To cope with the problem of pixel-based approaches, some contextual analyses, such as Markov random field (MRF), have been employed [10–12]. Although contextual approaches [10–18] can learn the statistics within the local neighborhood, their capability to represent spatial patterns is limited. Moreover, although some texture indices can be used to describe certain spatial patterns, most of them are still limited in their relatively simple representation capabilities [19, 20].

From the perspective of data modeling, LULC classification methods can be grouped into parametric and nonparametric approaches. Parametric approaches, such as the minimum distance classifier, maximum likelihood classifier, and the expectation-maximization (EM) algorithm, often require proper assumptions of data distribution [21]. However, for multitemporal or multisource data, the class distributions are hard to model. On the other hand, nonparametric approaches, such as artificial neural networks, decision tree, and support vector machine (SVM), are widely used in land cover classification [22]. Nevertheless, the performance of nonparametric approaches strongly depends on the selected classification features.

As an advanced machine learning approach, deep learning has been successfully applied in the field of image recognition and classification in recent years [23–27]. By mimicking the hierarchical structure of the human brain, deep learning approaches, such as Deep Belief Networks (DBN), can exploit complex spatiotemporal statistical patterns implied in the studied data [28, 29]. For remotely sensed data, deep learning approaches can automatically extract more abstract, invariant features, thereby facilitating land cover mapping. However, to the best of our knowledge, no research has been reported using deep learning for detailed urban LULC mapping on SAR data.

The present study proposes a detailed urban LULC mapping approach based on the popular deep learning architecture DBN. This study is one of the first attempts to apply the deep learning approach to detailed urban classification. Two-date high-resolution RADARSAT-2 PolSAR data over the Great Toronto Area (GTA) have been used for evaluation.

The rest of this paper is organized as follows. Section 2 describes the proposed land cover classification approach

based on the DBN model. Section 3 introduces the data and the process of the experiment. Section 4 presents and discusses the experimental results. Finally, we conclude this paper in Section 5.

2. Methodology

The proposed approach is based on the DBN model. This section briefly reviews the principle of the DBN model and describes the proposed method for land cover classification.

2.1. Deep Belief Networks. The DBN model was introduced by Hinton et al. in 2006 [28] for learning complex data patterns. It has become one of the extensively investigated and deployed deep learning architectures [24, 25]. The DBN is a probabilistic multilayer neural network composed of several stacked Restricted Boltzmann Machines (RBMs) [28, 30]. In a DBN, every two sequential hidden neural layers form an RBM. The input of the current RBM is actually the output features of a previous one. A DBN is therefore expected to hierarchically explore the pattern features in several abstract levels, given that the features obtained by a higher-level RBM are more representative than those obtained by lower ones. The training of DBN can be divided into two steps: pretraining and fine-tuning. This training process is further discussed below.

2.1.1. Restricted Boltzmann Machines. As the basic component of a DBN, Restricted Boltzmann Machine (RBM) can be treated as an unsupervised energy-based generative model. An RBM consists of a layer of visible units \mathbf{v} and a layer of hidden units \mathbf{h} , connected by symmetrically weighted connections, as shown in Figure 1.

Assuming binary-valued units, the RBM defines the energy of the joint configuration of visible and hidden units (\mathbf{v}, \mathbf{h}) as

$$E(v, h) = -\sum_{i=1}^I a_i v_i - \sum_{j=1}^J b_j h_j - \sum_{i=1}^I \sum_{j=1}^J w_{ij} v_i h_j, \quad (1)$$

where w_{ij} represents the weight associated with the connection between the visible unit v_i and the hidden unit h_j , a_i and b_j are the bias terms, and I and J are the numbers of visible and hidden units, respectively. The RBM assigns a probability to each configuration (\mathbf{v}, \mathbf{h}) using the energy function given by

$$p(v, h) = \frac{e^{-E(v, h)}}{Z}, \quad (2)$$

where Z is a normalization factor obtained by summing up the energies of all the possible (\mathbf{v}, \mathbf{h}) configurations:

$$Z = \sum_{\mathbf{v}} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}. \quad (3)$$

The conditional probabilities can be analytically computed as

$$p(h_j = 1 | \mathbf{v}) = \sigma \left(b_j + \sum_{i=1}^I v_i w_{ji} \right), \quad (4)$$

$$p(v_i = 1 | \mathbf{h}) = \sigma \left(a_i + \sum_{j=1}^J h_j w_{ji} \right), \quad (5)$$

where $\sigma(x)$ is the sigmoid function; that is, $\sigma(x) = 1/(1+e^{-x})$.

The training process of the RBM can be described as follows. After the random initialization of the weights and biases, iterative training of the RBM on the training data is performed. Given the training data on the visible units $\{v_i\}$, the states of hidden units $\{h_j\}$ are sampled according to (4). This step is called the positive phase of the RBM training. In the negative phase, the ‘‘reconstruction’’ of the visible units $\{v'_i\}$ is obtained according to (5). The positive phase is once more conducted to generate $\{h'_j\}$. Afterwards, the RBM weights and biases can be updated by the contrastive-divergence (CD) algorithm [31] through gradient ascent, which can be formulated as

$$\begin{aligned} \Delta w_{ij} &= \varepsilon \left(\langle v_i h_j \rangle - \langle v'_i h'_j \rangle \right), \\ \Delta a_i &= \varepsilon \left(\langle v_i \rangle - \langle v'_i \rangle \right), \\ \Delta b_j &= \varepsilon \left(\langle h_j \rangle - \langle h'_j \rangle \right), \end{aligned} \quad (6)$$

where ε denotes the learning rate and $\langle \cdot \rangle$ represents the mathematical expectation under the corresponding data distribution.

2.1.2. Pretraining. The DBN takes a layer-wise greedy learning strategy, in which RBMs are individually trained one after another and then stacked on the top of each other. When the first RBM has been trained, its parameters are fixed, and the hidden unit values are used as the visible unit values for the second RBM. The DBN repeats this process until the last RBM. Since pretraining is unsupervised, no label is needed. Unsupervised learning is believed to capture the crucial distribution of the data and can therefore help supervise learning when labels are provided. A batch-learning method is usually applied to accelerate the pretraining process; that is, the weights of the RBMs are updated every minibatch [32, 33].

2.1.3. Fine-Tuning. After the pretraining phase, the fine-tuning procedure is performed. A softmax output layer can be placed on top of the last RBM as a multiclass classifier, and the output-layer size is set to the same value as the total number of classes. To accomplish classification by utilizing the learned feature, we use the ordinary back-propagation technique through the whole pretrained network to fine-tune the weights for enhanced discriminative ability. Given

that the fine-tuning procedure is supervised learning, the corresponding labels for the training data are needed. After training, the predicted class label of a test sample can be obtained by forward propagation, in which the test data pass from the lowest-level visible layer through multi-RBM layers to the softmax output layer.

2.2. LULC Classification Based on DBN. To better understand the structure of the DBN-based LULC classification, a flowchart is given in Figure 2. To delineate the high variance and speckles of the PolSAR image, a neighbor window is used for local analysis, with the to-be-classified pixel placed at the center. Such neighbor window with size of $winsize * winsize$ can be represented by a vector formed by the pixel values from the window. The original input feature for the DBN consists of the processed Pauli parameters, which are the diagonal elements $(0.5|HH + VV|^2, 0.5|HH - VV|^2, \text{ and } 2|HV|^2$ under the reciprocal assumption) of the coherency matrix with their logarithm form stretched by linear scaling [2]. One kind of Pauli feature in a window is reshaped in a vector by sequentially connecting each feature line. A Pauli vector of a day can then be formed by connecting the three Pauli feature vectors. For multitemporal analysis, the input to DBN can be formed by connecting the m dates’ Pauli vectors, with the dimension of $winsize * winsize * 3 * m$.

For the training of DBN, Pauli vectors of the training samples are assigned to the visible layer of the first RBM as input training features. With a layer-by-layer pretraining strategy, the spatiotemporal dependencies are successively encoded in the hidden layers $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots, \mathbf{h}^{(n-1)}$, and $\mathbf{h}^{(n)}$. In the output layer, the labels of the training samples are provided, and the weights of the DBN are fine-tuned in a supervised manner.

For the prediction, the input features of the test samples are prepared in the same way as that of the training samples. The classification labels for the test samples can be obtained from the forward propagation of the test features through the trained network.

3. Data and Experiment

The study area is located in northern Greater Toronto Area (GTA), Ontario, Canada. The ten major LULC classes in the study area are as follows: high-density residential areas (HD), low-density residential areas (LD), industrial and commercial areas (Ind.), construction sites (Cons.), Water, Forest, Pasture, golf courses (Golf), and two types of crops (Crop1 and Crop2).

Two fine-beam full polarimetric SAR images were acquired by the RADARSAT-2 SAR sensor on June 19, 2008, and July 5, 2008. The center frequency is 5.4 GHz, that is, C-band. The June 19 data were obtained from the descending orbit, whereas the July 5 data were obtained from the ascending orbit, as shown in Figures 3(a) and 3(b). The data from the ascending and descending orbits were expected to complement each other from two different look directions. A total of 4952065 pixels of the overlap between the two images were classified.

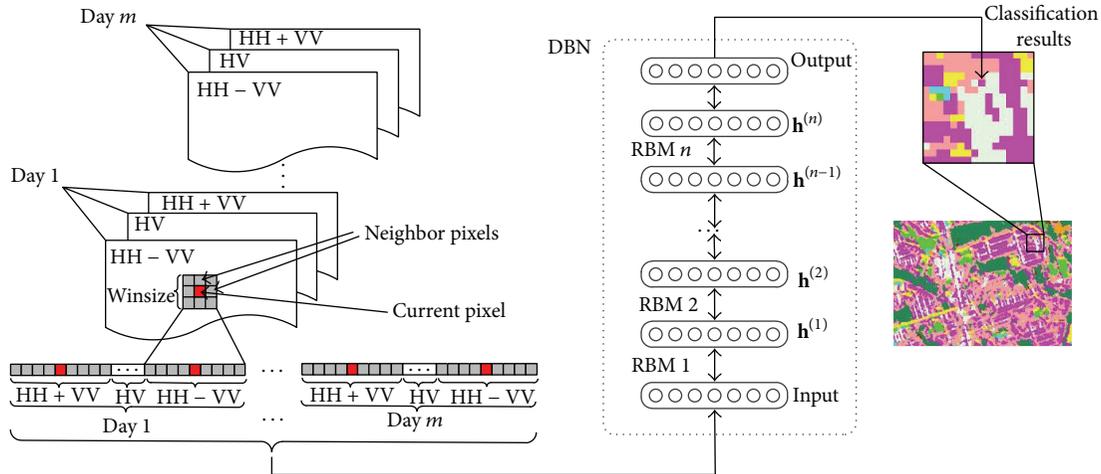


FIGURE 2: Flowchart of the proposed DBN-based classification approach.

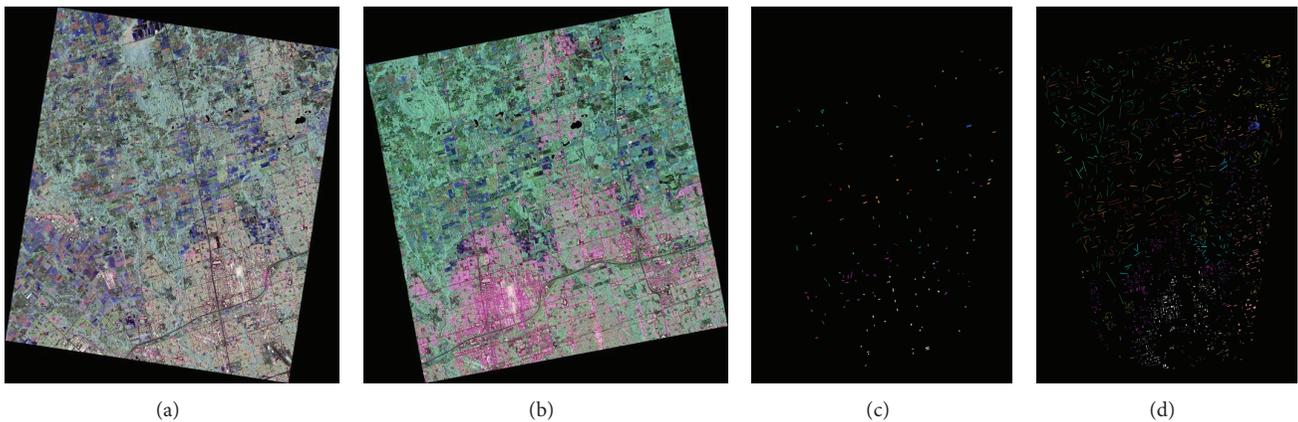


FIGURE 3: PolSAR images of northern Greater Toronto Area. (a) Pauli RGB image of RADARSAT-2 data on June 19, 2008. (b) Pauli RGB image of RADARSAT-2 data on July 5, 2008. (c) Training set. (d) Test set.

During the preprocessing, the multitemporal raw data were first orthorectified using the satellite orbital parameters and a 30 m resolution DEM. Then, they were registered to a vector file National Topographic Database (NTDB). A multilook process was further applied to generate the PolSAR features with the final spatial resolution of about 10 meters.

In the classification scheme, 19 subclasses were defined for the abovementioned 10 major land cover classes according to different scattering characteristics (e.g., the man-made structures have varying scattering appearance due to their distinctive shapes and directions). Approximately 1000 training pixels were assigned to each subclass. 120617 pixels evenly distributed over the classification area were randomly selected as the test samples. The training and test samples are visually shown in Figures 3(c) and 3(d), respectively.

The effective configurations of the DBN for detailed urban mapping were investigated. Comparisons with SVM, conventional neural networks (NN), and stochastic Expectation-Maximization (SEM) were conducted to assess the potential of our approach.

4. Results and Discussions

In this study, several experiments were conducted to validate the impact of different DBN configurations, including different network depths and hidden layer node numbers. To evaluate its classification efficiency, the DBN-based approach was compared with three other land cover methods: SVM, traditional neural networks (NN), and stochastic Expectation-Maximization (SEM). To quantitatively compare and estimate the capabilities of the proposed method, the overall accuracy (OA) and Kappa coefficient [34] were used as performance measurements.

The performance of the DBN-based classification method is sensitive to the neighbor window size. As the window size increases, more spatial dependencies could be captured by the DBN; thus, it is expected that better classification accuracy could be obtained with larger neighbor window size. Nevertheless, larger neighbor window size does not ensure better classification performance. Overly large window sizes could decrease the classification performance because bound areas tend to be confounded under an overlarge window. In

TABLE 1: DBN parameters setting.

Pretraining stage	
Learning rate	0.01
Number of epochs	50
Size of minibatch	100
Momentum	0.5 for the first 5 epochs, 0.9 thereafter
Weight decay rate	0.0002
Fine-tuning stage	
Learning rate	0.1
Number of epochs	20

the following experiments, the neighbor window size is set to $11 * 11$; thus, the dimension of the input data would be $11 * 11 * 3 * 2 = 726$.

Several parameters of the DBN are listed in Table 1; some of these parameters are based on experimentation, while the others are based on the recommendation of Hinton [33]. All the hidden layers in the DBN have the same number of hidden units. For all the DBN depths mentioned below, only the hidden layers were counted.

4.1. Effect of Network Depth. We first examine how the DBN depth influences the classification performance. The number of hidden layers is one of the key factors to the deep learning strategy. On one hand, it is proved that additional RBM layer can yield improved modeling power [35]. A higher level of representation leads to potentially more abstract features [27]. On the other hand, Larochelle et al. [36] argue that unnecessary RBM layers may degenerate the generalization capability of the DBN because more layers engender a more complex network model with more parameters to fit. With relatively less training samples, complex models often cause the overfitting problem [35]. The best depth of the DBN is usually related to a specific application and dataset.

To find a proper network depth, DBN models with increased number of RBM layers (i.e., from one to four layers) were compared. Each DBN model had the same constant structure; that is, all the RBM layers had the same number of hidden neurons. Comparisons were also conducted by varying the number of hidden neurons from 100 to 600 per layer. The results in Figure 4 show that, regardless of the number of neurons, improved overall accuracies were all obtained by the two-layer DBN model. Although the comparisons were made only up to 4 layers, it is expected that, with more layers, the overfitting problem will become more serious, which will lead to worse results. As such, the depth of DBN was set to two layers in the following experiment.

4.2. Comparison with Other Classification Methods. To demonstrate the effectiveness of the proposed LULC classification method, a comparison was conducted with three other land cover classification approaches (i.e., SVM, conventional NN, and SEM). The same Pauli features as the DBN-based method were used in SVM and traditional NN. The SEM method [9] applied an adaptive Markov Random Field (MRF) to explore contextual information, and we used the same

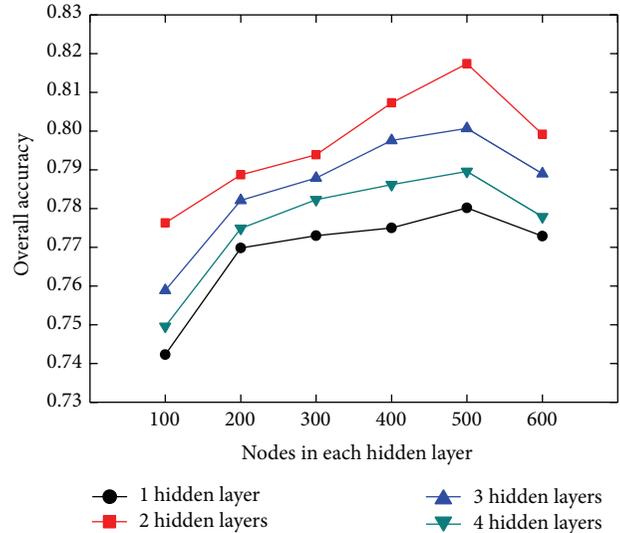


FIGURE 4: Impact of network depth.

settings reported there. The DBN contained two RBM layers, and each hidden layer had 500 units. Conventional NN had the same parameters as those of DBN; their only difference was that the weights of NN were not pretrained with unsupervised learning. The LIBSVM [37] toolkit was used as the implementation of SVM. SVM is a binary classifier, and the one-against-one strategy was used to convert the multiclass categorization problem to the binary classification problem. Experiments were performed using a radial basis function (RBF) kernel. The penalty term C and the RBF kernel width σ were selected using grid search within a given set $\{2^{-10}, \dots, 2^{10}\}$. The fivefold cross validation method indicated that the best validation rate was achieved when $C = 32$ and $\sigma = 2^{-7}$. These parameters were then used to train the SVM model. The classification accuracies using different classification approaches are presented in Table 2, where P and U stand for the producer's accuracy and user's accuracy, respectively.

Table 2 shows that, among the four classification methods, the DBN method results in the best performance, with an overall accuracy (OA) of 81.74%. Tables 3, 4, 5, and 6 list the confusion matrices of the four classification methods in percent.

Obviously, SEM obtained the highest accuracies in most natural classes (Water, Golf, Pasture, Crop1, and Forest). However, it performed extremely badly in several man-made classes (LD, HD, and Ind.). Generally, SEM provided the lowest overall classification accuracy of 72.43%.

Although SVM attained higher producer's accuracies in Cons., LD, and Crop1, its overall accuracy was still below DBN by 5%. The improved classification accuracy by the DBN method mainly originated from the significant increase of Pasture and Crop2. Tables 3 and 6 show that the accuracy of Pasture was greatly improved owing to the decrease of the confusion with the Golf class. The improvement of the accuracy of Crop2 was mainly due to the decrease of the

TABLE 2: Comparison of different classification methods.

	SVM		NN		SEM		DBN	
	<i>P</i>	<i>U</i>	<i>P</i>	<i>U</i>	<i>P</i>	<i>U</i>	<i>P</i>	<i>U</i>
Water	0.8521	0.9169	0.7847	0.9560	0.9668	0.9733	0.8697	0.9052
Golf	0.8588	0.5364	0.8922	0.6048	0.9245	0.8346	0.8118	0.7727
Pasture	0.5776	0.8949	0.6095	0.9198	0.8502	0.8499	0.8139	0.8987
Cons.	0.7639	0.6879	0.6383	0.6657	0.7239	0.7750	0.7265	0.7899
LD	0.6847	0.8509	0.5771	0.8175	0.3160	0.7697	0.6703	0.8884
Crop1	0.9020	0.7548	0.7991	0.8971	0.9617	0.6497	0.8800	0.8804
Crop2	0.7965	0.8882	0.8615	0.7671	0.8306	0.8649	0.8986	0.8469
Forest	0.8703	0.9098	0.8908	0.9408	0.9542	0.7076	0.9095	0.9489
HD	0.7203	0.5830	0.7195	0.4570	0.6264	0.4898	0.7824	0.5867
Ind.	0.7593	0.7556	0.6817	0.7394	0.4135	0.5811	0.7936	0.7632
OA	0.7679		0.7437		0.7243		0.8174	
Kappa	0.7398		0.7119		0.6906		0.7945	

TABLE 3: Confusion matrix (in percent) of the SVM method.

	Water	Golf	Pasture	Cons.	LD	Crop1	Crop2	Forest	HD	Ind.
Water	85.21	3.70	0.07	0.96	0.00	0.02	0.00	0.00	0.01	0.18
Golf	11.85	85.88	26.41	5.24	1.36	3.27	2.51	2.46	0.05	0.48
Pasture	0.02	6.55	57.76	0.00	0.09	0.75	0.72	1.07	0.02	0.00
Cons.	2.92	2.00	0.34	76.39	0.00	0.03	10.87	0.00	0.47	0.20
LD	0.00	0.17	0.31	0.03	68.47	1.65	0.24	3.03	4.57	3.04
Crop1	0.00	0.55	6.80	0.83	5.34	90.20	1.53	0.20	4.31	1.12
Crop2	0.00	0.58	3.67	15.52	0.11	0.16	79.65	0.68	0.06	0.02
Forest	0.00	0.11	2.25	0.07	1.57	0.65	4.20	87.03	0.13	0.04
HD	0.00	0.07	2.27	0.09	15.65	3.12	0.21	3.71	72.03	18.98
Ind.	0.00	0.37	0.12	0.88	7.40	0.15	0.07	1.83	18.34	75.93

TABLE 4: Confusion matrix (in percent) of the NN method.

	Water	Golf	Pasture	Cons.	LD	Crop1	Crop2	Forest	HD	Ind.
Water	78.47	2.05	0.00	0.35	0.00	0.00	0.00	0.00	0.00	0.00
Golf	19.56	89.22	19.21	6.71	0.24	3.05	1.12	0.40	0.02	0.17
Pasture	0.00	1.93	60.95	0.77	0.13	2.68	1.18	0.13	0.01	0.01
Cons.	1.64	3.28	0.68	63.83	0.00	0.04	9.88	0.00	0.04	0.01
LD	0.00	0.14	0.01	0.00	57.71	5.25	0.02	2.34	5.61	2.07
Crop1	0.00	0.30	2.37	0.38	1.74	79.91	0.64	0.14	0.33	0.53
Crop2	0.15	2.45	14.48	26.70	0.71	3.13	86.15	0.94	0.19	0.02
Forest	0.00	0.47	1.96	0.18	2.06	0.64	0.98	89.08	0.16	0.09
HD	0.00	0.06	0.33	0.37	31.99	5.13	0.03	6.21	71.95	28.95
Ind.	0.17	0.10	0.03	0.71	5.41	0.18	0.00	0.77	21.69	68.17

TABLE 5: Confusion matrix (in percent) of the SEM method.

	Water	Golf	Pasture	Cons.	LD	Crop1	Crop2	Forest	HD	Ind.
Water	96.68	0.53	0.08	0.13	0.08	0.12	0.09	0.09	0.04	0.11
Golf	3.10	92.45	4.36	4.41	0.54	0.03	0.38	0.28	0.53	1.01
Pasture	0.08	5.40	85.02	1.15	2.11	1.32	4.82	0.75	0.69	0.07
Cons.	0.06	1.01	0.60	72.39	0.21	0.00	6.12	0.03	0.78	0.35
LD	0.00	0.00	0.01	0.03	31.60	0.16	0.03	0.36	6.54	3.18
Crop1	0.06	0.07	6.80	1.21	6.28	96.17	3.58	0.85	4.67	10.90
Crop2	0.04	0.35	2.55	20.16	0.94	0.22	83.06	1.21	0.73	0.43
Forest	0.00	0.15	0.53	0.32	34.84	1.69	1.89	95.42	2.43	0.53
HD	0.00	0.02	0.04	0.13	10.20	0.29	0.01	0.76	62.64	42.07
Ind.	0.00	0.01	0.00	0.06	13.20	0.00	0.01	0.24	20.95	41.35

TABLE 6: Confusion matrix (in percent) of the DBN method.

	Water	Golf	Pasture	Cons.	LD	Crop1	Crop2	Forest	HD	Ind.
Water	86.97	5.72	0.02	0.21	0.00	0.00	0.00	0.00	0.00	0.00
Golf	11.92	81.18	5.93	3.42	0.16	0.66	0.29	0.40	0.01	0.38
Pasture	0.11	10.55	81.39	0.31	0.03	2.28	0.55	0.17	0.09	0.00
Cons.	0.99	1.26	0.26	72.65	0.00	0.01	6.16	0.00	0.31	0.05
LD	0.00	0.09	0.27	0.03	67.03	2.22	0.23	1.95	3.35	1.46
Crop1	0.00	0.14	2.49	0.93	2.47	88.00	1.22	0.23	0.71	0.26
Crop2	0.00	0.66	7.20	21.05	0.23	0.63	89.86	1.70	0.12	0.00
Forest	0.00	0.22	1.11	0.19	1.70	0.69	1.39	90.95	0.15	0.08
HD	0.00	0.15	1.31	0.31	19.67	5.40	0.24	2.61	78.24	18.41
Ind.	0.00	0.04	0.02	0.90	8.70	0.11	0.05	2.00	17.04	79.36

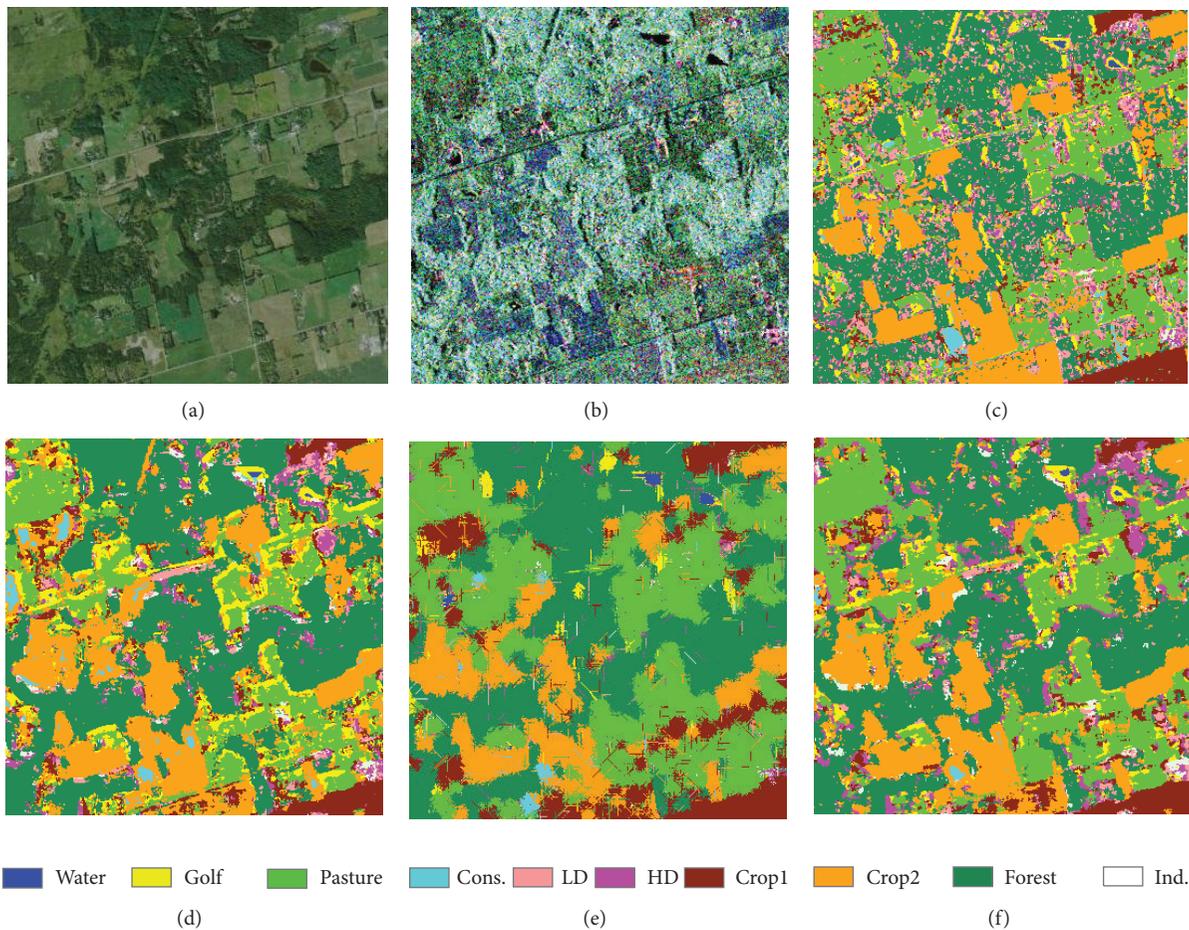


FIGURE 5: Zooming comparison of (a) Google Earth image and (b) PolSAR Pauli image and the classification results using (c) SVM, (d) NN, (e) SEM, and (f) DBN in a selected area.

commission to Cons. One plausible explanation for this improvement is that, with the effective features represented by the hidden layers, DBN could extract additional underlying dependencies and structures for the SAR data.

Compared with conventional NN, DBN obtained higher classification accuracies for almost all land cover types,

resulting in a notable increase in OA of 7%. The reason behind the superiority of DBN over NN is that, with an unsupervised pretraining process, more appropriate initial weights are assigned to the network, while the traditional neural network just sets random values for initial weights. The DBN-based method combines the advantages of both



FIGURE 6: Zooming comparison of (a) Google Earth image and (b) PolSAR Pauli image and the classification results using (c) SVM, (d) NN, (e) SEM, and (f) DBN in an Ind. area.

unsupervised and supervised learning; thus it can better distill spatiotemporal regularities from SAR data and improve classification performance.

The effects of different land cover classification methods are further illustrated in Figure 5. As can be observed in Figure 5, compared with SVM, the DBN method significantly reduces the misclassification of Forest. Compared with NN, DBN greatly decreases the misclassification of Pasture from Golf. Compared with SEM, DBN preserves the detail of residential areas. Figure 6 shows another example from an Ind. area. The figure shows that the DBN-based method provides classification map with more homogenous regions of the Ind. land cover type, which is more in line with reality.

5. Conclusion

A detailed urban LULC classification method based on the DBN model for PolSAR data is proposed. The effects of different network configurations are discussed. It is found that DBN with two hidden layers were appropriate for such detailed LULC mapping application. The experimental results demonstrate that the proposed method provides homogeneous mapping results with preserved shape details and that it

outperforms other land cover classification approaches (i.e., SVM, NN, and SEM) in a complex urban environment. Our future work will focus on more deep learning models for SAR data to further improve the classification results.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is mainly supported by the National Natural Science Foundation of China under Grants U1435219, 61125201, 61202126, 61202127, and 61402507.

References

- [1] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 1, pp. 6–43, 2013.

- [2] X. Niu and Y. Ban, "Multi-temporal RADARSAT-2 polarimetric SAR data for urban land-cover classification using an object-based support vector machine and a rule-based approach," *International Journal of Remote Sensing*, vol. 34, no. 1, pp. 1–26, 2013.
- [3] P. Lombardo, M. Sciotti, T. M. Pellizzeri, and M. Meloni, "Optimum model-based segmentation techniques for multi-frequency polarimetric SAR images of urban areas," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1959–1975, 2003.
- [4] T. Macri Pellizzeri, "Classification of polarimetric SAR images of suburban areas using joint annealed segmentation and "H/A/ α " polarimetric decomposition," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 58, no. 1-2, pp. 55–70, 2003.
- [5] Y. Ban and A. Jacob, "Object-based fusion of multitemporal multiangle ENVISAT ASAR and HJ-1B multispectral data for urban land-cover mapping," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 4, pp. 1998–2006, 2013.
- [6] S. W. Myint, P. Gober, A. Brazel, S. Grossman-Clarke, and Q. Weng, "Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery," *Remote Sensing of Environment*, vol. 115, no. 5, pp. 1145–1161, 2011.
- [7] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 1, pp. 2–16, 2010.
- [8] U. C. Benz, P. Hofmann, G. Willhauck, I. Lingenfelder, and M. Heynen, "Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 58, no. 3-4, pp. 239–258, 2004.
- [9] X. Niu and Y. Ban, "An adaptive contextual SEM algorithm for urban land cover mapping using multitemporal high-resolution polarimetric SAR data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 4, pp. 1129–1139, 2012.
- [10] Y. Wu, K. Ji, W. Yu, and Y. Su, "Region-based classification of polarimetric SAR images using wishart MRF," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 4, pp. 668–672, 2008.
- [11] D. Gleich, "Markov random field models for non-quadratic regularization of complex SAR images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 3, pp. 952–961, 2012.
- [12] A. Voisin, V. A. Krylov, G. Moser, S. B. Serpico, and J. Zerubia, "Classification of very high resolution SAR images of urban areas using copulas and texture in a hierarchical Markov random field model," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 1, pp. 96–100, 2013.
- [13] P. Yu, A. K. Qin, and D. A. Clausi, "Unsupervised polarimetric SAR image segmentation and classification using region growing with edge penalty," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1302–1317, 2012.
- [14] D. H. Hoekman, M. A. M. Vissers, and T. N. Tran, "Unsupervised full-polarimetric SAR data segmentation as a tool for classification of agricultural areas," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, no. 2, pp. 402–411, 2011.
- [15] K. Ersahin, I. G. Cumming, and R. K. Ward, "Segmentation and classification of polarimetric SAR data using spectral graph partitioning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 1, pp. 164–174, 2010.
- [16] B. Liu, H. Hu, H. Wang, K. Wang, X. Liu, and W. Yu, "Superpixel-based classification with an adaptive number of classes for polarimetric SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 907–924, 2013.
- [17] L. Bombrun, G. Vasile, M. Gay, and F. Totir, "Hierarchical segmentation of polarimetric SAR images using heterogeneous clutter models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 2, pp. 726–737, 2011.
- [18] V. Akbari, A. P. Doulgeris, G. Moser, T. Eltoft, S. N. Anfinsen, and S. B. Serpico, "A textural-contextual model for unsupervised segmentation of multipolarization synthetic aperture radar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 4, pp. 2442–2453, 2013.
- [19] R. J. Dekker, "Texture analysis and classification of ERS SAR images for map updating of urban areas in the Netherlands," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1950–1958, 2003.
- [20] V. V. Chamundeeswari, D. Singh, and K. Singh, "An analysis of texture measures in PCA-based unsupervised classification of SAR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 2, pp. 214–218, 2009.
- [21] X. Niu and Y. Ban, "Multitemporal polarimetric RADARSAT-2 SAR data for urban land cover mapping through a dictionary-based and a rule-based model selection in a contextual SEM algorithm," *Canadian Journal of Remote Sensing*, vol. 39, no. 2, pp. 138–151, 2013.
- [22] B. Waske and J. A. Benediktsson, "Fusion of support vector machines for classification of multisensor data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 3858–3866, 2007.
- [23] N. Jones, "Computer science: the learning machines," *Nature*, vol. 505, no. 7482, pp. 146–148, 2014.
- [24] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning—a new frontier in artificial intelligence research [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13–18, 2010.
- [25] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.
- [26] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.
- [27] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [28] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [29] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [30] N. Lopes and B. Ribeiro, "Towards adaptive learning with improved convergence of deep belief networks on graphics processing units," *Pattern Recognition*, vol. 47, no. 1, pp. 114–127, 2014.
- [31] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [32] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, vol. 7700 of *Lecture Notes in Computer Science*, pp. 437–478, Springer, Berlin, Germany, 2nd edition, 2012.

- [33] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," Tech. Rep., Department of Computer Science, University of Toronto, 2010.
- [34] W. D. Thompson and S. D. Walter, "A reappraisal of the kappa coefficient," *Journal of Clinical Epidemiology*, vol. 41, no. 10, pp. 949–958, 1988.
- [35] N. le Roux and Y. Bengio, "Representational power of restricted Boltzmann machines and deep belief networks," *Neural Computation*, vol. 20, no. 6, pp. 1631–1649, 2008.
- [36] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *Journal of Machine Learning Research*, vol. 10, pp. 1–40, 2009.
- [37] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

Research Article

Bayesian Information Criterion Based Feature Filtering for the Fusion of Multiple Features in High-Spatial-Resolution Satellite Scene Classification

Da Lin,¹ Xin Xu,¹ and Fangling Pu²

¹Signal Processing Laboratory, School of Electronic Information, Wuhan University, Wuhan 430072, China

²Wireless Communication and Sensor Network Laboratory, School of Electronic Information, Wuhan University, Wuhan 430072, China

Correspondence should be addressed to Xin Xu; xinxu@whu.edu.cn

Received 12 November 2014; Accepted 18 February 2015

Academic Editor: Tianfu Wu

Copyright © 2015 Da Lin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a novel classification method for high-spatial-resolution satellite scene classification introducing Bayesian information criterion (BIC)-based feature filtering process to further eliminate opaque and redundant information between multiple features. Firstly, two diverse and complementary feature descriptors are extracted to characterize the satellite scene. Then, sparse canonical correlation analysis (SCCA) with penalty function is employed to fuse the extracted feature descriptors and remove the ambiguities and redundancies between them simultaneously. After that, a two-phase Bayesian information criterion (BIC)-based feature filtering process is designed to further filter out redundant information. In the first phase, we gradually impose a constraint via an iterative process to set a constraint on the loadings for averting sparse correlation descending below to a lower confidence limit of the approximated canonical correlation. In the second phase, Bayesian information criterion (BIC) is utilized to conduct the feature filtering which sets the smallest loading in absolute value to zero in each iteration for all features. Lastly, a support vector machine with pyramid match kernel is applied to obtain the final result. Experimental results on high-spatial-resolution satellite scenes demonstrate that the suggested approach achieves satisfactory performance in classification accuracy.

1. Introduction

Scene classification has aroused more and more attention in remote sensing domain. For satellite imagery of high-spatial-resolution, it evokes a great deal of challenging problems in scene classification due to high intraclass variability, low interclass disparity, and other external factors such as changes of viewpoint, illuminations and shadows, background clutter, partial occlusions, and multiple instances. In addition, with considerable increase of the spatial resolution of images, details of the targets become clearer, and a host of cues also become more distinctive, such as structure and colour. As a consequence, it is of great importance to appropriately combine and fuse them in various respects. In the past decade or so, many researchers and practitioners have made great efforts in exploiting different sources of information in

the high-spatial-resolution satellite imagery to enhance classification performance [1–5].

Unlike the case of low-spatial-resolution satellite images, where solitary type of feature descriptor has been proved to be effective and efficient for classification [6, 7], it is universally recognized that instead of adopting a solitary type of feature, it is more favorable to fuse and combine a set of diverse and complementary features such as features based on structure and colour information [8, 9]. Hence, how to fuse these pieces of diverse and complementary information and eliminate the ambiguities and redundancies between them becomes a critical problem. One widely praised approach is feature-level fusion, in which features from different channels are fused, developing a new pattern for scene classification. Many approaches have been reported in the literature [10–12].

The canonical correlation analysis (CCA) [13] method has been spotlighted, especially in feature fusion realm [14, 15], due to its capability in expressing inherent correlation between two sets of features. In order to extract canonical correlation features from two groups of features, the CCA method firstly constructs a correlation criterion function by extracting two diverse features from the identical samples. Then the CCA method creates efficient and effective discriminant features for classification.

However, when the dimensions of features are too large, such as in case of high-spatial-resolution satellite scene classification researches, traditional CCA methods are no longer proper. Besides, when the features are extracted from the identical image, sample covariance matrices turn into being undefined or unstable, inducing extra difficulty in parameter estimation. As a consequence, a dimension reduction approach is imperative for CCA to tackle this problem. In the past decade, abundant approaches have been proposed for feature shrinkage and selection, containing the non-negative garrote by Breiman [16], least absolute shrinkage and selection operator (Lasso) by Tibshirani [17], smoothly clipped absolute deviation (SCAD) by Fan and Li [18], and Elastic-net by Zou and Hastie [19]. Recently, these approaches have been employed to CCA for relationship assessment between two sets of high-dimensional remote sensing data. The feature selection strategies are utilized to the canonical feature loadings which set some of the coefficients to exact zeros for selecting the remaining features. The crucial features selected are then called the sparse set of features, and the canonical correlation analysis exploiting these features is often known as sparse canonical correlation analysis (SCCA). Initially, Waaijenborg et al. [20] suggested a penalized form of CCA adopting an iterative regression process with the Univariate Soft Threshold (UST) form of the Elastic-net penalty. Subsequently, Parkhomenko et al. [21] suggested the SCCA approach utilizing a form of regularization resembling UST Elastic-net [19]. Witten and Tibshirani [22] incorporated the Lasso penalty in their SCCA. These approaches are interesting. Yet, they do not control the sparsity in a direct way. At the same time, it has been revealed that sundry penalized likelihood approaches have the oracle property under certain conditions [18, 23–26]. However, in practical application, without an appropriate model of feature selection, the oracle property cannot be realized. Accordingly, the approaches may not necessarily generate sparse set of features. To address the issue, this paper suggests a two-phase process in implementing SCCA in which L1 penalty is utilized on the feature loadings during the first phase, and then a Bayesian Information Criterion (BIC) based feature filtering algorithm is conducted to further remove redundant and noisy information. To be specific, in the first phase, we gradually impose a constraint via an iterative process to set a constraint on the loadings for averting sparse correlation descending below to a lower confidence limit of the approximated canonical correlation. In the second phase, Bayesian information criterion (BIC) is utilized to conduct the feature filtering which sets the smallest loading in absolute value to zero in each iteration for all features.

The rest of this paper is organized as follows. The feature extraction process is presented in Section 2. Section 3 offers a concrete and detailed depiction of the methodology exploited in this paper. Section 4 displays experimental results and gives a performance assessment. Finally, Section 5 summarizes the work and points the directions for the future work.

2. Multiset Feature Extraction

2.1. Scale-Invariant Feature Transform (SIFT) Descriptor. In the detected region, SIFT descriptor extracts a gradient orientation histogram [27]. The gradient image is sampled over a 4×4 grid in each of eight orientation planes; thus resulting descriptor is of dimension 128. A weight to the magnitude of each sample point is gained by implementing a Gaussian window function. This puts more highlights on the gradients that are near the center of the region and renders the descriptor less sensitive to the small changes in the position of the detected region. The gradient magnitude is used to weigh the contribution to the orientation and location bins. The descriptor is immune to small errors in the region detection and small geometric distortions, largely due to the quantization of orientations and gradient locations. The square root of the sum of squared components is calculated to normalize the descriptor for acquiring illumination invariance.

2.2. Colour Histogram Descriptor. Colour histogram descriptor is three separated histograms for the R, G, and B channels [28]. A colour histogram represents the approximate distribution of the colours in an image, due to the fact that each histogram bin stands for a local colour range in the given colour space. Colour histograms are invariant to the translation and rotation of the image content; meanwhile they are unsophisticated to compute. In our experiments, the number of bins is 40, and the resulting descriptor is of dimension 120 through concatenating the three independent histograms.

3. Methodology

3.1. Sparse Canonical Correlation Analysis (SCCA). Canonical correlation analysis (CCA) is a multivariate statistical approach proposed to grope for the correlation between two sets of features [13]. Suppose that two feature sets $X = X_1, X_2, \dots, X_p$ and $Y = Y_1, Y_2, \dots, Y_q$ are of dimensions $n \times p$ and $n \times q$ ($p \leq n, q \leq n$), which extracted from the same image. Let the columns of X and Y be standardized to have standard deviation 1 and mean 0, let u and v be $p \times 1$ and $q \times 1$ vector of weights, and let $\xi = Xu$ and $\eta = Yv$ be linear combinations of the features of data sets X and Y , respectively. Note that ξ and η are $n \times 1$ vectors. Subsequently, (1) will be maximized to estimate coefficient vectors u and v :

$$\begin{aligned} \rho &= \frac{\text{Cov}(\xi, \eta)}{[\text{var}(\xi) \text{var}(\eta)]^{1/2}} \\ &= \frac{u^T X^T Y v}{[(u^T X^T X u)(v^T Y^T Y v)]^{1/2}}, \end{aligned} \quad (1)$$

where $X^T X$ and $Y^T Y$ are within data covariance matrices and $X^T Y$ is the between data covariance matrix. Equation (1) can be reformulated as follows for scaling u and v has very insignificant influence on the correlation coefficient:

$$\begin{aligned} \rho &= \text{corr}(\xi, \eta) = \max u^T X^T Y v \\ \text{subject to } & u^T X^T X u = v^T Y^T Y v = 1. \end{aligned} \quad (2)$$

The aforementioned CCA approach is not applicable when the quantity of features is excessive. Latent multicollinearity between predictor features further complicates the computation for the covariance matrices can turn into undefined or unstable. As a consequence, some critical features should be selected by standard model selection criteria. In the subsequent process, the selected set of features is utilized to compute the canonical correlation for making the results understandable, which is called sparse canonical correlation analysis (SCCA). Theoretically, SCCA is implemented by maximizing the penalized objective function below:

$$\begin{aligned} \rho &= \text{corr}(\xi, \eta) = \max_{u,v} u^T X^T Y v \\ \text{subject to } & u^T X^T X u = v^T Y^T Y v = 1. \end{aligned} \quad (3)$$

In order to tackle the multicollinearity problem, a host of approaches have been introduced. Vinod suggested incorporating penalty terms to the diagonal elements of the covariance matrix, which appears to resemble the ridge regression thought in regression analysis [29]. This needs to estimate additional ridge parameters. Other regularization forms have been proposed where the variance matrices are substituted with their corresponding identity matrices [22] or diagonal matrices [21]. In our work, the matrices $X^T X$ and $Y^T Y$ are substituted with their corresponding diagonal matrices.

3.2. Shrinkage Methods. Penalized linear regression mechanisms have been widely applied to analyze high-dimensional data, and it has incorporated feature selection and shrinkage techniques. Assume that y is an $n \times 1$ vector and x is an $n \times p$ matrix. Then the estimation of penalized regression coefficient β can be yielded by using following penalized regression model:

$$\hat{\beta}^{\text{penalized}} = \arg \min |y - x\beta|^2 + p_\lambda(\beta), \quad (4)$$

where $p_\lambda(\beta)$ is the penalty term and λ is a tuning parameter which is estimated utilizing permutation approaches or cross validation (CV). As the first penalized regression approach, ridge regression was proposed to temper the multicollinearity among the predictors in which a quadratic penalty term is embedded to the regular least square estimating equations [30]. Ridge regression implements a penalty on the coefficients to shrink them towards zero. Yet the shrunken coefficients are never equal to zero. As a consequence, ridge regression fails to conduct feature selection. The least absolute shrinkage and selection operator (Lasso), Elastic-net, and

smoothly clipped absolute deviation (SCAD) are different from ridge regression which solve the multicollinearity problem (i.e., shrinkage) and set some of the coefficient to exact zero, creating sparse set of features (i.e., feature selection). In this paper, we apply different penalty functions to SCCA utilizing the algorithm elaborated by Parkhomenko et al. [21]. The tuning parameters for all penalty functions are estimated through cross validation (CV).

3.2.1. Least Absolute Shrinkage and Selection Operator (Lasso) Penalty. The least absolute shrinkage and selection operator (Lasso) penalty is a shrinkage approach and has the competence of selecting discriminant features by shrinking some coefficients and setting others to zero [31]. The penalty term of Lasso is defined as follows:

$$p_\lambda^{\text{lasso}}(\beta) = \lambda \sum_{j=1}^p |\beta_j|, \quad (5)$$

where λ is a tuning parameter. The solution of Lasso is given as

$$\hat{\beta}_j^{\text{lasso}} = (|\hat{\beta}_j| - \lambda)_+ \text{sign}(\hat{\beta}_j). \quad (6)$$

This is similar to the soft thresholding rule introduced by Donoho et al. [32] and Donoho and Johnstone [33], which was utilized to estimate wavelet coefficients.

3.2.2. Elastic-Net Penalty. Elastic-net is a regularization mechanism that carries out continuous shrinkage and feature selection simultaneously [34]. To be specific, this approach utilizes both L_1 penalty of Lasso and L_2 quadratic penalty of ridge regression to create a convex combination. Consequently, this approach preserves the abilities of feature selection and coefficients shrinkage. The definition of the Elastic-net penalty can be formulated as follows:

$$p_\lambda^{\text{elastic-net}}(\beta) = \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2. \quad (7)$$

Nevertheless, thanks to two tuning parameters needed to be estimated, the computational cost of Elastic-net is somewhat higher.

As a substitute to Elastic-net, Zou and Hastie proposed a predigested version of the Elastic-net called univariate soft thresholding (UST) [34], the solution of which is shown as

$$\hat{\beta}_j^{\text{elastic-net}} = (|\hat{\beta}_j| - 0.5\lambda)_+ \text{sign}(\hat{\beta}_j). \quad (8)$$

In this paper, the Elastic-net based on the univariate soft threshold is adopted to implement feature loadings u and v as follows:

$$\begin{aligned} \hat{u}_j &= (|\hat{u}_j| - 0.5\lambda_1)_+ \text{sign}(\hat{u}_j), \\ \hat{v}_j &= (|\hat{v}_j| - 0.5\lambda_2)_+ \text{sign}(\hat{v}_j). \end{aligned} \quad (9)$$

3.2.3. Smoothly Clipped Absolute Deviation (SCAD) Penalty. Fan and Li proposed a nonconvex penalty function named smoothly clipped absolute deviation (SCAD) [18]. They suggested three criteria for determining an excellent penalty function, namely, (i) sparsity, (ii) continuity, and (iii) unbiasedness. They made further efforts to claim that the SCAD penalty possesses these properties. The SCAD penalty is shown as follows:

$$p_{\lambda}^{\text{SCAD}}(\beta) = \begin{cases} \lambda |\beta_j|, & \text{if } |\beta_j| \leq \lambda, \\ -\left(\frac{|\beta_j|^2 - 2a\lambda |\beta_j| + \lambda^2}{2(a-1)}\right), & \text{if } \lambda < |\beta_j| \leq a\lambda, \\ \frac{(a+1)\lambda^2}{2}, & \text{if } |\beta_j| > a\lambda. \end{cases} \quad (10)$$

When value within the range of λ and $a\lambda$, SCAD penalty function coincides with a quadratic spline function. The function is continuous, and when $a > 2$ and $\beta > 0$ the first derivative can be formulated as follows:

$$p'_{\lambda}(\beta) = \lambda \left\{ I(\beta \leq \lambda) + \frac{(a\lambda - \beta)_+}{(a-1)\lambda} I(\beta > \lambda) \right\}. \quad (11)$$

The SCAD penalty is continuously differentiable on $(-\infty, 0) \cup (0, \infty)$, but singular at 0, with its derivatives zero outside the range $[-a\lambda, a\lambda]$. This penalty function sets small coefficients to zero, shrinks mid-size coefficients towards zero, and keeps large coefficients untouched. Consequently, the SCAD penalty generates almost unbiased coefficients and a sparse solution for large coefficients. The solution of SCAD penalty is shown as follows:

$$\widehat{\beta}_j^{\text{SCAD}} = \begin{cases} (|\widehat{\beta}_j| - \lambda)_+ \text{sign}(\widehat{\beta}_j), & \text{if } |\widehat{\beta}_j| \leq 2, \\ \frac{(a-1)\widehat{\beta}_j - \text{sign}(\widehat{\beta}_j)a\lambda}{a-2}, & \text{if } 2\lambda < |\widehat{\beta}_j| \leq a\lambda, \\ \widehat{\beta}_j, & \text{if } |\widehat{\beta}_j| > a\lambda. \end{cases} \quad (12)$$

This thresholding rule has two unknown parameters: λ and a . In an ideal situation, the optimal results (λ, a) can be acquired utilizing a scheme involving a two-dimensional grid-search with criteria resembling cross validation approaches. However, such an execution is computationally intensive. In the Bayesian perspective, Fan and Li advised $a = 3.7$ is a wise option for many issues [18]. They have further highlighted that the performance of feature selection issues does not boost tremendously when data-driven approaches are adopted. In this paper, we set a to 3.7 and λ was selected by cross validation. Meanwhile, the thresholding rule (12) was adopted to load vectors u and v .

3.2.4. Hard-Threshold Penalty. Hard-thresholding directly sets several coefficients to zero [35, 36]. However, this penalty function does not tackle the issue of multicollinearity among the predictors, for it does not shrink any coefficients toward zero. Nevertheless, the results obtained by this penalty are unbiased estimators with large effects. The solution of the hard-thresholding rule is revealed as follows:

$$\widehat{\beta}_j^{\text{hard}} = \beta_j I(|\beta_j| > \lambda). \quad (13)$$

3.3. The Suggested BIC Based Feature Filtering Algorithm. The major drawback of the current SCCA approaches is that they do not control sparsity directly. Hence, it is hard to complete efficient and effective elimination of noisy and redundant information. There is a trade-off between the sparsity of the features and the maximum correlation. In this paper, we suggest a two-phase process to establish an equilibrium between the sparsity of the features and the maximum correlation. In the first phase, we gradually impose a constraint via an iterative process to set a constraint on the loadings for averting sparse correlation descending below to a lower confidence limit of the approximated canonical correlation. In the second phase, Bayesian information criterion (BIC) is utilized to conduct the feature filtering which sets the smallest loading in absolute value to zero in each iteration for all features.

The proposed feature filtering process is iterative and simple. One more coefficient of $\widehat{\beta}_j$, at each iteration, is set to be 0 according to the magnitudes of the coefficients in absolute value. Let $\widehat{\beta}_i$ and $\widehat{\beta}_j$ be the constrained effective dimension reduction direction; the proposed feature filtering process is shown as follows.

- (i) Let $d = p + q$.
- (ii) Define a new direction $\widehat{\beta}'_j(d)$ by maintaining the largest d coefficients of $\widehat{\beta}_j$ in absolute value and assigning the other $p+q-d$ coefficients to 0. Searching $\widehat{\beta}^p_j(d)$ as projection of $\widehat{\beta}'_j(d)$ into the space B_i , the set of all $\widehat{\beta}^p_j(d)$ should satisfy the following.
 - (1) The set of zero coefficients in $\widehat{\beta}^p_j(d)$ is the same as that in $\widehat{\beta}'_j(d)$.
 - (2) Consider $\widehat{\beta}^p_j(d)^T X^T X \widehat{\beta}_i = 0, i \leq j - 1$.
 - (3) Consider $\widehat{\beta}^p_j(d)^T X^T X \widehat{\beta}^p_j(d) = 1$.

- (iii) Compute the correlation

$$r_d = \text{Corr}(\widehat{\beta}'_j(d), \widehat{\beta}^p_j(d)) = \frac{\text{Cov}(\widehat{\beta}'_j(d), \widehat{\beta}^p_j(d))}{[\text{var}(\widehat{\beta}'_j(d)) \text{var}(\widehat{\beta}^p_j(d))]^{1/2}}, \quad (14)$$

and the BIC-type criterion $\text{BIC}(d) = n \log(1 - r_d^2) + d \log(n)$, where n is the sample size.

- (iv) Let $d = d - 1$. Repeat Steps (ii)–(iv) until $d = 0$.

After the above feature filtering process is implemented, we obtain a sequence of $BIC(d)$ as d descends from $p + q$ to 0. Let d_0 be the integer at which $BIC(d)$ is minimized. Then, the $p + q - d_0$ smallest coefficients of $\hat{\beta}_j$ in absolute value are assigned to 0. This proposed feature filtering process is a streamlined feature selection process. In feature filtering, at most $p + q$ possibilities are taken into account, which make it viable to conduct even when $p + q$ are large. Lastly, the features corresponding to the minimum BIC value are the final selected features.

3.4. Support Vector Machine (SVM) with Pyramid Match Kernel (PMK-SVM) Classifier. Establishment of kernel-based learning algorithms is based on the notion of mapping data into a Euclidean space and then discovering linear relations within the mapped data. Taking a typical issue as an example, the SVM unearths the optimal separating hyperplane between two classes in a feature space. The assistance provided by a kernel function is to map pairs of data points in an input space to their inner product in the feature space, thereby estimating the similarities between all points and deciding their relative positions. Linear relations are discovered in the feature space, although a decision boundary may still be nonlinear in the input space, depending on the method of a feature mapping function. The support vector machine (SVM) with pyramid match kernel (PMK-SVM) [37] furnishes an accurate and time-saving solution for classification and the pyramid match kernel function is formulated as follows:

$$K(\Psi(y), \Psi(z)) = \sum_{i=0}^L \frac{1}{2^i} (\Gamma(H_i(y), H_i(z)) - \Gamma(H_{i-1}(y), H_{i-1}(z))), \quad (15)$$

where y, z are the input sets, $L = \lceil \log_2 D \rceil$ in which D is a sphere of diameter, Ψ is the feature extraction function, $H_i(x)$ is the i th histogram in $\Psi(x)$, and Γ is a histogram intersection function which measures the overlap between two histograms' bins:

$$\Gamma(A, B) = \sum_{j=1}^r \min(A^{(j)}, B^{(j)}), \quad (16)$$

where A and B are histograms with r bins, and $A^{(j)}$ denotes the count of the j th bin of A . Since in the construction of the pyramid phase $\Gamma(H_L(y), H_L(z)) = \min(|y|, |z|)$, and $\Gamma(H_{-1}(y), H_{-1}(z)) = 0$, (15) is equivalent to

$$K(\Psi(y), \Psi(z)) = \frac{\min(|y|, |z|)}{2^L} + \sum_{i=0}^{L-1} \frac{1}{2^{i+1}} \Gamma(H_i(y), H_i(z)). \quad (17)$$

In order to preserve generality and obtain promising and satisfactory classification results. Here, we employ support vector machine (SVM) with pyramid match kernel (PMK-SVM) as classifier. Under the multiclass circumstances, a set

of binary classifiers and majority vote technique are utilized to perform multiclass categorization. Figure 1 illustrates our classification scheme based on the suggested two-phase BIC filtering process.

4. Experiments and Results

Experiments were performed on a high-spatial-resolution satellite image with a size of 4000 pixels \times 4000 pixels, as presented in Figure 2. The view was captured by the GeoEye-1 satellite on 21 November 2009 at Majuqiao town, which locates in the southwest of Tongzhou, southeast of Beijing, where the latitudes and longitudes at the lower right and upper left corners are $39^\circ 43' N, 116^\circ 32' E$ and $39^\circ 44' N, 116^\circ 30' E$ respectively. The ground sampling distance is approximately 0.5 m and the band assignment is red for band 3, green for band 2, and blue for band 1. The image mainly contains eight-class satellite scenes: factories, roads, water, farm land, high buildings, low buildings, bare land, and green land. With associated geographic information, the reference data presented in Figure 6(a) was manually labeled. Moreover, some classes have not been taken into account, which can be neglected, such as clearance between the green land and its neighbours. Figure 3 exhibits one example of each class from the eight-class satellite scene.

In this paper, the k -fold cross validation approach is used to optimize the penalty parameters for each canonical feature pair, where k in k -fold cross validation is set to 10 by empirical. Specifically, the obtained feature data are split into two parts: $1/k$ proportion of feature data for validation (testing) and the remaining $k - 1/k$ proportion of feature data for training. The loading vectors are yielded in the training procedure and used in the testing procedure. We maximize the correlation of testing feature data to select the sparseness parameters λ_u and λ_v using cross validation approach. The weighted vectors u and v are obtained by the preset values of λ_u and λ_v . Subsequently, the correlation is calculated by

$$\Delta_{\text{cor}} = \frac{1}{k} \sum_{j=1}^k |\text{cor}(X_j u^{-j}, Y_j v^{-j})|, \quad (18)$$

where u^{-j} and v^{-j} are the weight vectors in the training sets X_j and Y_j , respectively, and k is the implementing times of cross validation. Here, X_j and Y_j are the test sets. Lastly, λ_u and λ_v values are determined as optimum sparseness parameters according to the maximum value of Δ_{cor} .

To achieve reliable results and see the convergence tendency of the classification accuracy, we conducted the entire experiments with different proportions of training-testing samples by using linear increasing number of training samples and learnt that classification accuracy is gradually boosted with an increasing number of training samples. To avert training and testing samples are too spatially correlated, we trained on 5, 10, 15, 20, 25 and 30 images in each class and tested on the remainder. At the same time, we chose training data and testing data from the different high-spatial-resolution satellite scene for independent data set and reconducted the classification experiments with these

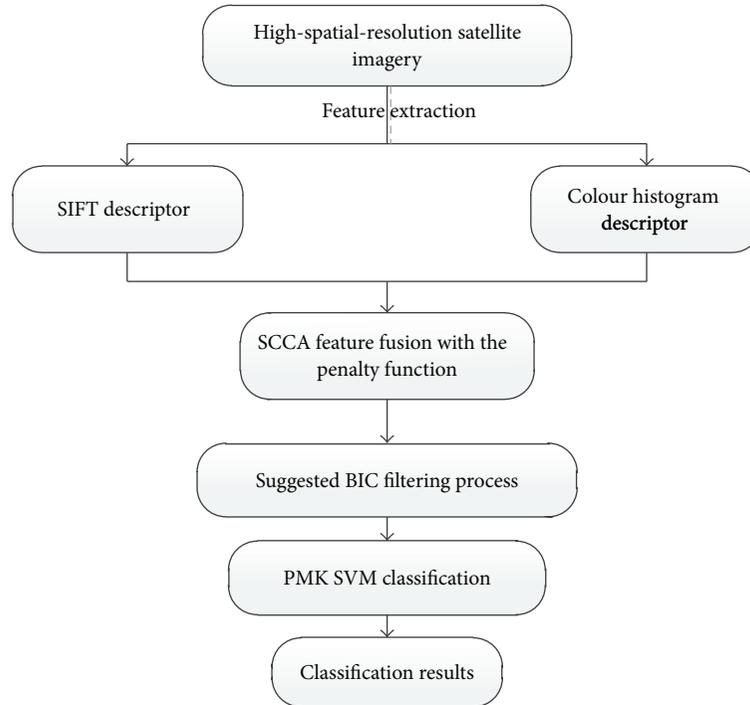


FIGURE 1: Flow chart of the suggested classification method.

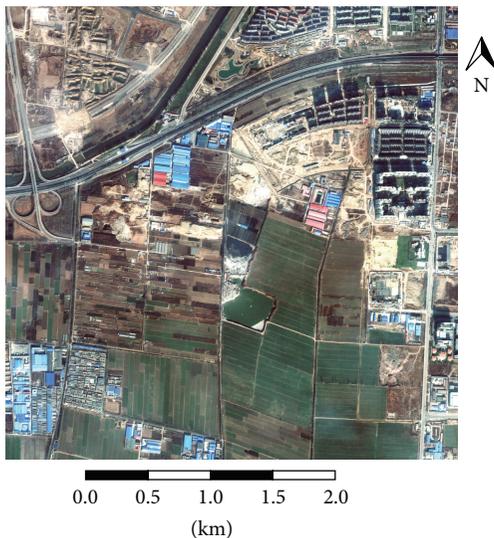


FIGURE 2: High-spatial-resolution satellite scene.

selected data; the acquired results were coincident as well. The classifiers were not conspicuously biased within this particular image for the moderate selected proportion of training samples and hence classification results would not be extremely exaggerated. We divided training\testing data at random and reiterated the experiments ten times. Meanwhile, we kept an account of the average of each-class classification accuracies for every run. The mean and standard deviation of the results from each individual experiment were used to quantify the final result.

For the sake of boost in classification performance to some extent, these two features should be fused in an apt tactic. For comparison purposes, SCCA with different penalty functions (Lasso, Elastic-net, SCAD, and Hard-threshold) was introduced to fuse the features yielded by feature extraction methods adopted in this paper. It is all obvious from Figure 4 and easy to observe that SCCA with SCAD penalty function acquires a relatively outstanding result over the other penalty functions. The reasons are sketched as follows. On one side, in order to grip the complete source of image information, we utilized two typical and representative features to depict structure and colour properties of the image, and these features contain the essential and intrinsic information of the image. On the other side, SCAD penalty function possesses the competence to manipulate multicollinearity between features as compare to the other penalty functions. However, these methods do not have immediate control over the sparsity. Consequently, an extra two-phase feature filtering process was indispensable to conduct for further removing redundant and opaque information between and within features. Figure 5 compares the classification performance of four different penalty functions with the suggested two-phase BIC based filtering process. It is evident that classification accuracy enhances after utilizing the suggested BIC filter, which can be explained by the possibility that without the suggested BIC filter the SCCA may not necessarily engender sparse set of features. In other words, after using the suggested BIC filter the noisy and redundant information among features have been further eliminated.

Note that the SCAD penalty function with the suggested BIC filter yielded the best classification accuracy. Figures

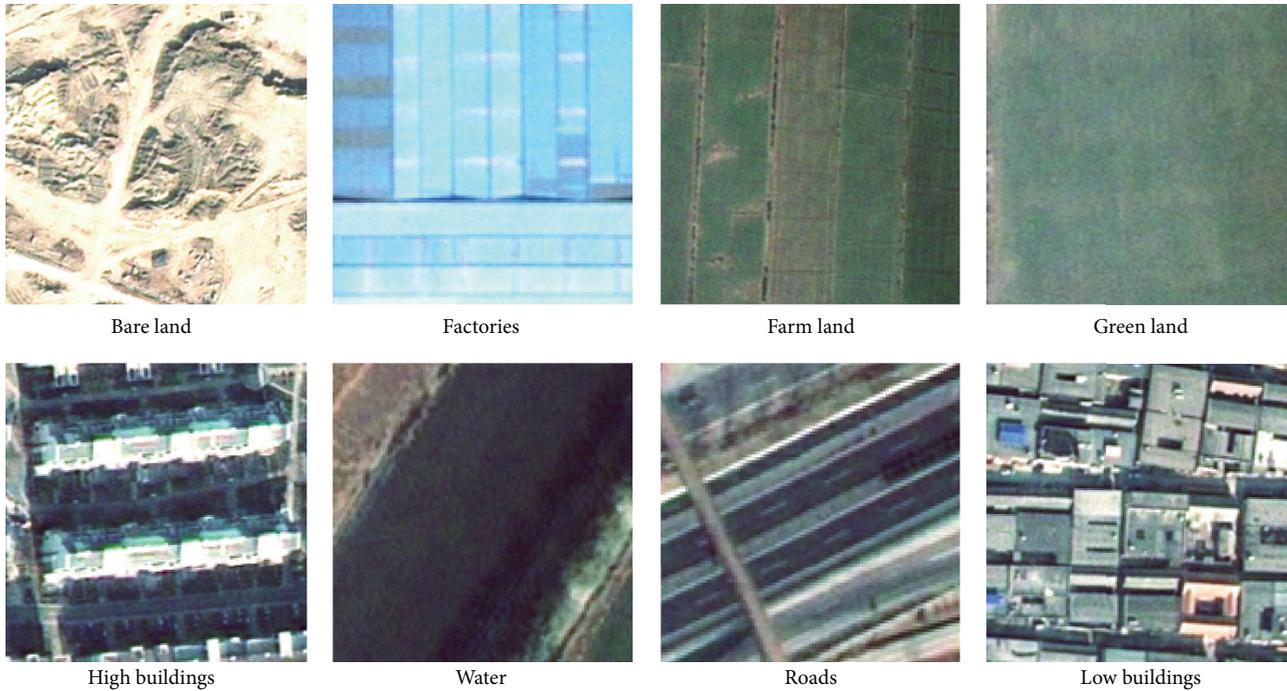


FIGURE 3: Examples of each class in the eight-class satellite scene.

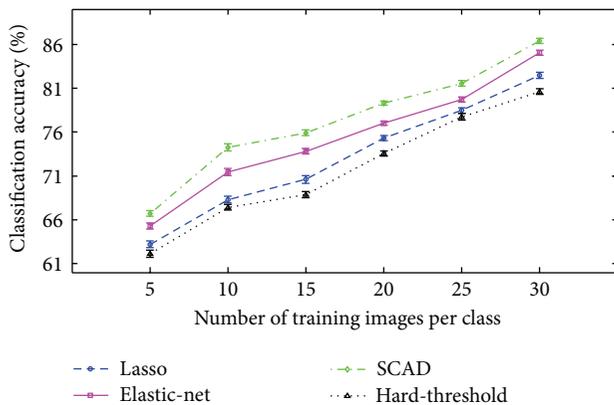


FIGURE 4: Performance comparison using different penalty functions on SCCA without the suggested BIC filter.

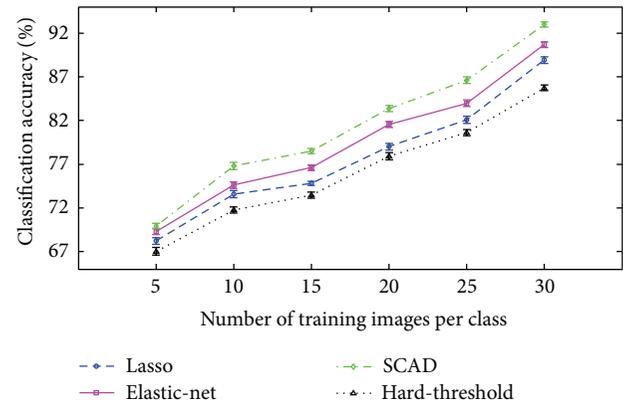


FIGURE 5: Performance comparison using different penalty functions on SCCA with the suggested BIC filter.

6(a) and 6(b) show the comparison between the reference image in the same satellite scene and the best classified image acquired by our method (SCAD penalty function SCCA with the suggested BIC filter), respectively, and the confusion matrices obtained via SCAD penalty function SCCA without and with the suggested BIC filter are presented in Figures 7 and 8, respectively.

It is as expected that misclassification was more inclinable to emerge between short buildings and factories. This is due to the fact that factories often include dense houses and level and perpendicular lines, which resemble short buildings. Meantime green land was misclassified as farm land or bare land; the most contributing factor to this

misclassification is that these classes possess similar elements and patterns. However, confusions between some classes were tough to understand. Namely, some short buildings, roads, and factories were classified as bare land.

5. Conclusions

In this paper, a BIC based feature filtering approach was presented for high-spatial-resolution satellite scene classification. The SCCA with a two-phase BIC feature filtering process acts as a crucial component in satellite scene classification and can strikingly meliorate the classification accuracy by

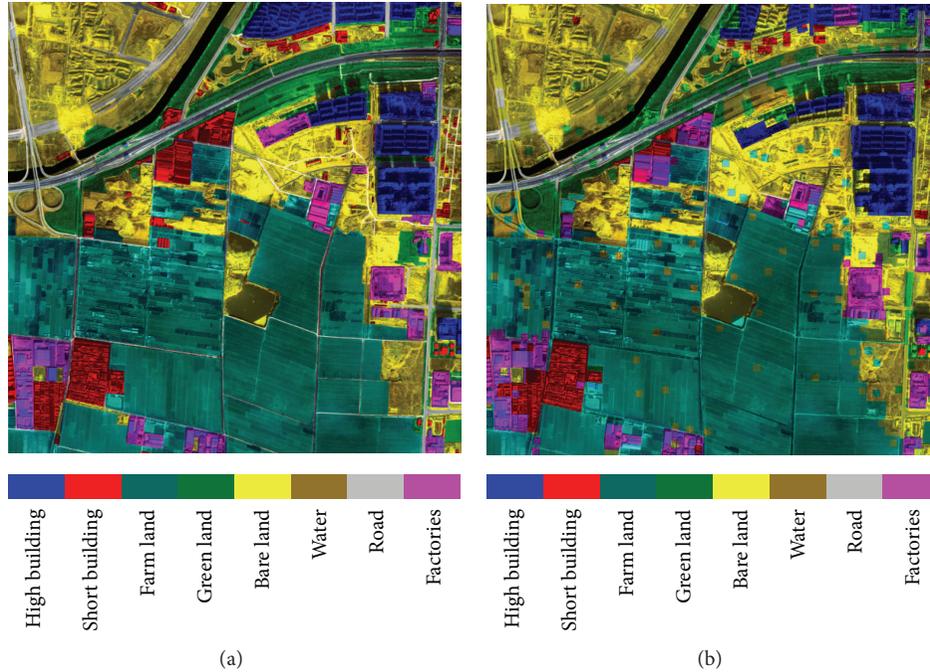


FIGURE 6: Classification results for the eight-class satellite image. (a) The hand-labelled reference data. (b) The best classified image using our method (SCAD penalty function SCCA with the suggested BIC filter).

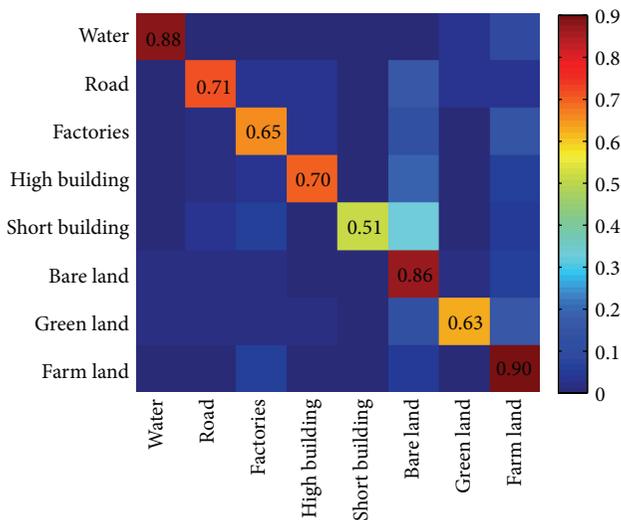


FIGURE 7: Confusion matrix for eight-class high-spatial-resolution satellite scene data using SCAD penalty function SCCA without the suggested BIC filter. The entry in the i th row and j th column is the percentage of images from class i that were misclassified as class j .

efficiently filtering and selecting two distinct and complementary features, SIFT and colour histogram. However, in the absence of a prior original features processing step before using the suggested SCCA feature fusion method, current method is greatly restricted in the experiments of Majuqiao town satellite scene classification. To solve the aforementioned issue, we intend to implement a sparse coding step to the original features before utilizing the feature fusion

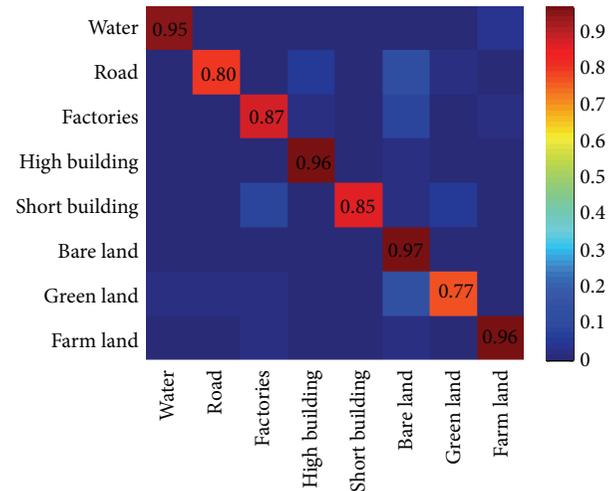


FIGURE 8: Confusion matrix for eight-class high-spatial-resolution satellite scene data using SCAD penalty function SCCA with the suggested BIC filter. The entry in the i th row and j th column is the percentage of images from class i that were misclassified as class j .

method for reducing the computational cost and enhancing classification accuracy.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Basic Research Program of China (973 Program) under Grant no. 2011CB707102 and the National High Technology Research and Development Program of China (863 Program) under Grant no. 2013AA122301.

References

- [1] A. Bekkari, S. Idbraim, D. Mammass, M. El Yassa, and D. Ducrot, "SVM classification of high resolution urban satellites images using composite kernels and haralick features," *Journal of Emerging Technologies in Web Intelligence*, vol. 6, no. 1, pp. 69–74, 2014.
- [2] G.-S. Xia, W. Yang, J. Delon et al., "Structural high-resolution satellite image indexing," in *Proceedings of the ISPRS TC VII Symposium—100 Years ISPRS*, pp. 298–303, Vienna, Austria, July 2010.
- [3] D. Dai and W. Yang, "Satellite image classification via two-layer sparse coding with biased image representation," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 173–176, 2011.
- [4] B. A. Johnson, "High-resolution urban land-cover classification using a competitive multi-scale object-based approach," *Remote Sensing Letters*, vol. 4, no. 2, pp. 131–140, 2013.
- [5] J. J. Mitchell, R. Shrestha, C. A. Moore-Ellison, and N. F. Glenn, "Single and multi-date landsat classifications of basalt to support soil survey efforts," *Remote Sensing*, vol. 5, no. 10, pp. 4857–4876, 2013.
- [6] L. Ruiz, A. Fdez-Sarría, and J. Recio, "Texture feature extraction for classification of remote sensing data using wavelet decomposition: a comparative study," in *Proceedings of the 20th ISPRS Congress*, 2004.
- [7] C.-S. Li and V. Castelli, "Deriving texture feature set for content-based retrieval of satellite image database," in *Proceedings of the International Conference on Image Processing*, pp. 576–579, IEEE, October 1997.
- [8] M. Shimoni, D. Borghys, R. Heremans, C. Perneel, and M. Acheroy, "Fusion of PolSAR and PolInSAR data for land cover classification," *International Journal of Applied Earth Observation and Geoinformation*, vol. 11, no. 3, pp. 169–180, 2009.
- [9] J. Muñoz-Marí, G. Camps-Valls, L. Gómez-Chova, and J. Calpe-Maravilla, "Combination of one-class remote sensing image classifiers," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS '07)*, pp. 1509–1512, IEEE, Barcelona, Spain, June 2007.
- [10] B. Bigdeli, F. Samadzadegan, and P. Reinartz, "A multiple svm system for classification of hyperspectral remote sensing data," *Journal of the Indian Society of Remote Sensing*, vol. 41, no. 4, pp. 763–776, 2013.
- [11] P. Du, Y. Chen, J. Xia, and K. Tan, "A novel remote sensing image classification scheme based on data fusion, multiple features and ensemble learning," *Journal of the Indian Society of Remote Sensing*, vol. 41, no. 2, pp. 213–222, 2013.
- [12] E. Pasolli, F. Melgani, D. Tuia, F. Pacifici, and W. J. Emery, "SVM active learning approach for image classification using spatial information," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 2217–2223, 2014.
- [13] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3-4, pp. 321–377, 1936.
- [14] Q.-S. Sun, Z.-D. Liu, P.-A. Heng, and D.-S. Xia, "A theorem on the generalized canonical projective vectors," *Pattern Recognition*, vol. 38, no. 3, pp. 449–452, 2005.
- [15] Q.-S. Sun, S.-G. Zeng, Y. Liu, P.-A. Heng, and D.-S. Xia, "A new method of feature fusion and its application in image recognition," *Pattern Recognition*, vol. 38, no. 12, pp. 2437–2448, 2005.
- [16] L. Breiman, "Better subset regression using the nonnegative garrote," *Technometrics*, vol. 37, no. 4, pp. 373–384, 1995.
- [17] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B: Methodological*, vol. 58, no. 1, pp. 267–288, 1996.
- [18] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.
- [19] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.
- [20] S. Waaijenborg, P. C. Verselewe De Witt Hamer, and A. H. Zwinderman, "Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis," *Statistical Applications in Genetics and Molecular Biology*, vol. 7, no. 1, article 3, 2008.
- [21] E. Parkhomenko, D. Tritchler, and J. Beyene, "Sparse canonical correlation analysis with application to genomic data integration," *Statistical Applications in Genetics and Molecular Biology*, vol. 8, no. 1, article 1, 2009.
- [22] D. M. Witten and R. J. Tibshirani, "Extensions of sparse canonical correlation analysis with applications to genomic data," *Statistical Applications in Genetics and Molecular Biology*, vol. 8, no. 1, article 28, 2009.
- [23] J. Fan and H. Peng, "Nonconcave penalized likelihood with a diverging number of parameters," *The Annals of Statistics*, vol. 32, no. 3, pp. 928–961, 2004.
- [24] Y. Kim, H. Choi, and H.-S. Oh, "Smoothly clipped absolute deviation on high dimensions," *Journal of the American Statistical Association*, vol. 103, no. 484, pp. 1665–1673, 2008.
- [25] N. Meinshausen and B. Yu, "Lasso-type recovery of sparse representations for high-dimensional data," *The Annals of Statistics*, vol. 37, no. 1, pp. 246–270, 2009.
- [26] P. Zhao and B. Yu, "On model selection consistency of Lasso," *The Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.
- [27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [28] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 703–715, 2001.
- [29] H. D. Vinod, "Canonical ridge and econometrics of joint production," *Journal of Econometrics*, vol. 4, no. 2, pp. 147–166, 1976.
- [30] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, 2000.
- [31] R. Tibshirani, "Regression shrinkage and selection via the lasso: a retrospective," *Journal of the Royal Statistical Society, Series B. Statistical Methodology*, vol. 73, no. 3, pp. 273–282, 2011.
- [32] D. L. Donoho, I. M. Johnstone, G. Kerkycharian, and D. Picard, "Wavelet shrinkage: Asymptopia," *Journal of the Royal Statistical Society: Series B*, vol. 57, no. 2, pp. 301–369, 1995.

- [33] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.
- [34] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.
- [35] A. Antoniadis, "Wavelets in statistics: a review," *Journal of the Italian Statistical Society*, vol. 6, no. 2, pp. 97–130, 1997.
- [36] J. Fan, "Comments on 'wavelets in statistics: a review' by A. Antoniadis," *Journal of the Italian Statistical Society*, vol. 6, no. 2, pp. 131–138, 1997.
- [37] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1458–1465, October 2005.

Research Article

Automatic Change Detection Method of Multitemporal Remote Sensing Images Based on 2D-Otsu Algorithm Improved by Firefly Algorithm

Liang Huang,¹ Yuanmin Fang,¹ Xiaoqing Zuo,¹ and Xueqin Yu²

¹Faculty of Land Resource Engineering, Kunming University of Science and Technology, Kunming 650093, China

²Kunming Surveying and Mapping Institute, Kunming 650051, China

Correspondence should be addressed to Liang Huang; kmhuangliang@163.com

Received 30 October 2014; Accepted 3 February 2015

Academic Editor: Xue Cheng Tai

Copyright © 2015 Liang Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a new automatic change detection method of multitemporal remote sensing images based on 2D-Otsu algorithm improved by Firefly algorithm. The proposed method is designed to automatically extract the changing area between two temporal remote sensing images. First, two different temporal remote sensing images were acquired through difference value method of remote sensing images; then, the 2D-Otsu threshold segmentation principles are analyzed and the optimal threshold of 2D-Otsu threshold segmentation method is figured out by using the Firefly algorithm, where the difference images are conducted with binary classification to obtain the changing category and the nonchanging category; finally, the proposed method is used to carry out change detection experiments on the two selected areas, where a variety of methods are compared. Experimental results show that the proposed method can effectively and quickly extract the changing area between the two temporal remote sensing images; thus, it is an effective method of change detection for remote sensing images.

1. Introduction

With the development of society and technology, human activities are changing the landscape of ground surface and its utilization mode every day. The rapid population growth and frequent natural disasters have also accelerated the speed of such changes. Therefore, quickly and effectively monitoring the information of such changes and analyzing the characteristics, causes, impacts and results of the changes have great significance to the global sustainable development [1]. Emergence and development of remote sensing technology have provided technical support for quickly and effectively monitoring the information of these changes. Detection of remote sensing image changes aims to acquire the changing information of desired surface features by analyzing and processing two or multiple remote sensing images of the same area at different times [2, 3]. Currently, the change detection technology of remote sensing image has been widely used in many fields, such as disaster assessment

[4], land use/coverage monitoring [5], environmental change monitoring [6], agricultural survey [7], and urban planning [8].

Currently, a number of methods have been proposed for the change detection of remote sensing image. At present, change detection mainly includes two strategies: direct comparison and postclassification comparison, of which the postclassification comparison strategy is an approach to classifying (supervised or unsupervised method) different temporal images of the same area to compare and analyze the classified results in order to obtain the location and type of changing information. The advantage of this strategy is to minimize the impact of nonfeature change factors, but the change detection accuracy through this strategy is substantially equal to the product of the two classification accuracy values, and the classification error of each temporal classification result will be amplified in the process of comparison, thus inevitably exaggerating the extent of changes. Meanwhile, it is often more complex and difficult to get high-accuracy

classification results, resulting in low-accuracy and uncertain change detection results [9]. Therefore, the strategy of direct comparison has always been highlighted in researches on change detection. Direct comparison is a strategy to obtain difference images by directly operating and alternating the pixel values of different temporal remote sensing images of the same area that have been calibrated so as to figure out the changing area by analyzing the difference images. In analyzing these difference images, it would be the most simple thing to directly use one-dimensional threshold processing methods, but the processing accuracy of threshold is not high [10, 11]. In order to improve the processing accuracy of threshold, many scholars have proposed to extend the one-dimensional threshold selection method to two dimensions (2D). Currently, a number of two-dimensional threshold selection methods have been raised, such as 2D entropy (2D maximum entropy, 2D minimum cross entropy) [12], 2D-Otsu [13], and 2D maximum fuzzy entropy [14], but it is still difficult to select the threshold values of the two-dimensional histogram, for which unremitting exploration efforts have been made by many scholars. To solve this problem, many scholars attempt to achieve fast optimization for the 2D threshold by combining 2D threshold segmentation method and optimization algorithm. Shen et al. [15], Zheng et al. [16], and Alim et al. [17] sought to figure out the optimal threshold of 2D maximum entropy, respectively, through genetic algorithm, ant colony algorithm, PSO (Particle Swarm Optimization) algorithm, and ABCO (Artificial Bee Colony Optimization). Qian used PSO algorithm to find the optimal threshold of 2D-Otsu [18]. Tian and Zeng used QPSO (Quantum-behaved Particle Swarm Optimization) algorithm to carry out image threshold segmentation in combination with 2D maximum fuzzy entropy [19]. The applications of PSO algorithm, QPSO algorithm, bee colony algorithm, ant colony algorithm, and genetic algorithm have improved the speed of figuring out thresholds, but the optimization results may be inaccurate as these algorithms are prone to falling into local extremum.

The Firefly algorithm [20] is a global optimization algorithm proposed by Yang, which can overcome the problem of easily falling into local optimum. Chen et al. [21] and Alomoush et al. [22], respectively, applied the Firefly algorithm to find optimal threshold and then used for image segmentation and achieved good segmentation results. Given the fast global search capability of the Firefly algorithm and the good segmentation result of 2D-Otsu threshold segmentation method, this paper segments the thresholds of difference images by combing the Firefly algorithm and 2D-Otsu threshold method to obtain the binary change detection map.

2. Architecture of the Proposed Method and Problem Formulation

Assume $T_1 = \{T_1(i, j), 1 \leq i \leq M, 1 \leq j \leq N\}$ and $T_2 = \{T_2(i, j), 1 \leq i \leq M, 1 \leq j \leq N\}$ are the remote sensing images of the same area at different times t_1 and t_2 that underwent standard product preprocessing and

coregistration; the remote sensing image sizes are $M \times N$. The ultimate aim of the proposed method is to generate the binary change map. The proposed method consists of three steps: (1) construction of difference image; (2) threshold optimization based on Firefly algorithm; and (3) generation of final binary change map.

First, it is construction of difference images. Currently, there are two methods, namely, image algebraic operation and image transformation, to construct difference images. The method based on image algebraic operation includes difference method, ratio method, and the combination of difference method and ratio method; the method based on image transformation includes principal component analysis, change vector analysis and correlation analysis. Since the image algebraic operation method requires simple algorithm but can obtain high accuracy, this paper applied the image difference method of the image algebraic operation, which can be expressed below:

$$DI(x, y) = T_1(i, j) - T_2(i, j), \quad (1)$$

where $DI(x, y)$ is the difference image constructed.

Second, it is optimization of Firefly threshold. After acquiring the difference images, the difference images need to be analyzed so as to obtain the binary change map. Binary classification through threshold segmentation is one of the most frequently used methods. Selecting thresholds via one-dimensional histogram is the easiest method, of which the main method is OTSU algorithm. However, as OTSU algorithm does not use local spatial data of the image, the segmentation will be ineffective when the image is subject to noise disturbance or gray scale distribution intersection. Therefore, many scholars have expanded it to two-dimensional approach, and better segmentation results have been achieved [23]. To this end, this paper used the 2D-Otsu algorithm to analyze difference images acquired. However, compared with the thresholds selected in one-dimensional histogram, 2D-Otsu algorithm requires a large amount and a long time of computation. To solve the above problem, the Firefly algorithm is introduced to optimize the threshold of 2D-Otsu algorithm.

Third, it is generation of the final binary change map. After figuring out the optimal threshold of 2D-Otsu algorithm, threshold segmentation is given to difference images to obtain the binary change detection map of the two temporal remote sensing images.

3. Change Detection Based on 2D-Otsu Algorithm Improved by Firefly Algorithm

3.1. 2D-Otsu Threshold Segmentation Method. The size of the difference image $DI(x, y)$ is $M \times N$, where $1 \leq x \leq M$, $1 \leq y \leq N$, the gray scale of the image is L , and the average gray-scale of pixel neighborhood is also divided into L level. A binary group is obtained by calculating the average gray scale of its neighborhood at each pixel point, that is, the gray-scale value of the pixel point and the average gray-scale value of its neighborhood. Assume the probability of the binary group

(i, j) is f_{ij} ; the corresponding joint probability density p_{ij} can be defined as [13, 24]

$$p_{ij} = \frac{f_{ij}}{(M \times N)}, \quad (2)$$

where $M \times N$ is the number of pixels of the difference image, $i, j = 1, 2, \dots, L$, and

$$\sum_{i=1}^L \sum_{j=1}^L p_{ij} = 1. \quad (3)$$

Assume that there are two categories, namely, C_0 (changing category) and C_1 (nonchanging category) in the 2D histogram, as well as two different probability density distribution functions. Set the threshold value as (s, t) ; then, the probabilities of C_0 and C_1 are shown below:

$$\begin{aligned} \omega_0 &= \sum_{i=1}^s \sum_{j=1}^t p_{ij}, \\ \omega_1 &= \sum_{i=s+1}^L \sum_{j=t+1}^L p_{ij}. \end{aligned} \quad (4)$$

The corresponding mean vectors of C_0 and C_1 are

$$\bar{\mu}_0 = (\mu_{0i}, \mu_{0j})^T = \left(\sum_{i=1}^s \sum_{j=1}^t \frac{i p_{ij}}{\omega_0}, \sum_{i=1}^s \sum_{j=1}^t \frac{j p_{ij}}{\omega_0} \right)^T, \quad (5)$$

$$\begin{aligned} \bar{\mu}_1 &= (\mu_{1i}, \mu_{1j})^T \\ &= \left(\sum_{i=s+1}^L \sum_{j=t+1}^L \frac{i p_{ij}}{\omega_1}, \sum_{i=s+1}^L \sum_{j=t+1}^L \frac{j p_{ij}}{\omega_1} \right)^T. \end{aligned} \quad (6)$$

The total mean vector of 2D histogram is

$$\bar{\mu}_T = (\mu_{Ti}, \mu_{Tj})^T = \left(\sum_{i=1}^L \sum_{j=1}^L i p_{ij}, \sum_{i=1}^L \sum_{j=1}^L j p_{ij} \right)^T. \quad (7)$$

Usually, the probability away from the histogram diagonal can be negligible; then, it can be assumed in the two areas: $i = s + 1, \dots, L$; $j = 1, \dots, t$ and $i = 1, \dots, s$; $j = t + 1, \dots, L$ have $p_{ij} \approx 0$, where

$$\omega_0 + \omega_1 \approx 1, \quad \bar{\mu}_T = \omega_0 \bar{\mu}_0 + \omega_1 \bar{\mu}_1. \quad (8)$$

Define a between-class dispersion matrix:

$$\begin{aligned} S_B &= \omega_0 (\mu_{0i} - \mu_{Ti}) (\mu_{0j} - \mu_{Tj}) \\ &+ \omega_1 (\mu_{1i} - \mu_{Ti}) (\mu_{1j} - \mu_{Tj}). \end{aligned} \quad (9)$$

Use the trace of S_B as a measure of the between-class dispersion matrix; then, the 2D-Otsu function of the threshold value (s, t) corresponding to the difference image can be defined as

$$\begin{aligned} t_r \sigma_B(s, t) &= \omega_0 \left[(\mu_{0i} - \mu_{Ti})^2 + (\mu_{0j} - \mu_{Tj})^2 \right] \\ &+ \omega_1 \left[(\mu_{1i} - \mu_{Ti})^2 + (\mu_{1j} - \mu_{Tj})^2 \right]. \end{aligned} \quad (10)$$

By using Formula (8), it can be simplified into

$$t_r \sigma_B(s, t) = \frac{[\omega_0 \mu_{Ti} - \bar{\mu}_i]^2 + [\omega_1 \mu_{Tj} - \bar{\mu}_j]^2}{\omega_0 [1 - \omega_0]}, \quad (11)$$

where $\bar{\mu}_i = \sum_{i=1}^s \sum_{j=1}^t i p_{ij}$, $\bar{\mu}_j = \sum_{i=1}^s \sum_{j=1}^t j p_{ij}$.

The segmentation criteria corresponding to the 2D-Otsu threshold segmentation method has to maximize the 2D-Otsu function $t_r \sigma_B(s, t)$ for the purpose of obtaining the optimal threshold (s^*, t^*) .

3.2. 2D-Otsu Threshold Segmentation Method Based on Firefly Algorithm. In order to solve the problem that 2D-Otsu threshold segmentation method requires a large amount and long time of computation, this paper applied the Firefly algorithm to the threshold optimization for 2D-Otsu threshold segmentation method and proposed the 2D-Otsu threshold segmentation method based on the firefly algorithm, so as to convert the threshold value selection of 2D-Otsu threshold segmentation method into the optimization of 2D-Otsu function $t_r \sigma_B(s, t)$ based on the Firefly algorithm.

The Firefly algorithm [25], proposed by Yang in the Cambridge University, is a bionic swarm intelligent optimization algorithm by simulating the natural behaviors of fireflies. The algorithm is characterized by simple parameter setting, high-accuracy optimization, and powerful global optimization [20]. The principle of Firefly algorithm is to simulate firefly individuals in nature by searching for space dots. In the optimization process, by using the phototaxis characteristics of firefly, the searching and optimization process is simulated into the attraction and movement process of firefly individuals, where the target function of the problem is measured by the merits of the position of firefly individuals. In other words, the target function of the problem is converted into the firefly seeking maximum brightness. The process of survival of the fittest among firefly individuals is also the iterative process of feasible solutions in the course of target function optimization. Therefore, the Firefly algorithm can rapidly conduct global optimization [21].

Firefly algorithm consists of two main elements: brightness and attraction, where brightness reflects the merits of the position of fireflies and determines the movement direction of firefly individuals, while attraction determines the movement distance of firefly individuals. Through the continuous updating and iteration of brightness and attraction, the optimal solution to the target function will be achieved [21, 26, 27]. In the Firefly algorithm, the relative fluorescence brightness of firefly individuals is defined as

$$I = I_0 \times e^{-\gamma r_{ij}^2}, \quad (12)$$

where I_0 is the maximum fluorescence brightness of the firefly, γ is the attraction coefficient of light intensity, which is usually set to a constant; r_{ij} is the spatial distance between Fireflies i and j , that is, $r_{ij} = \|x_i - x_j\|$, where x_i and x_j are, respectively, the spatial position of Fireflies i and j .

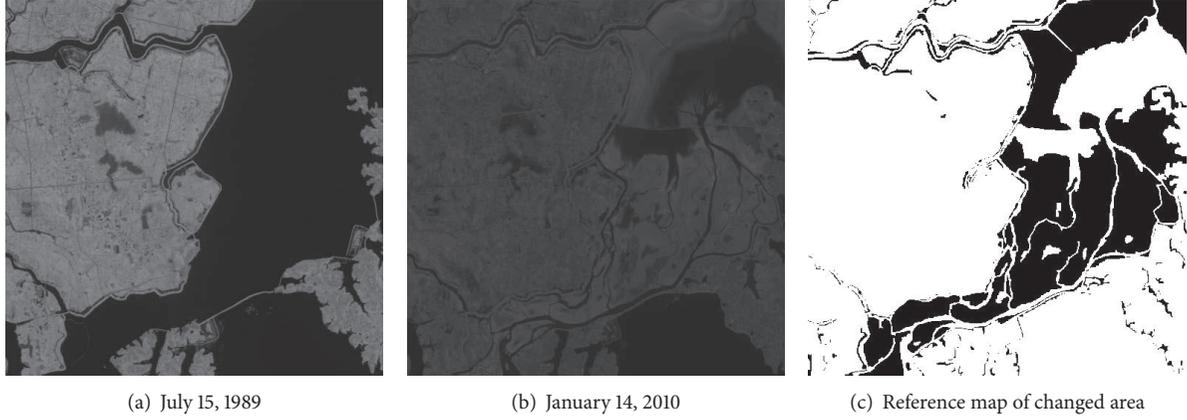


FIGURE 1: Multitemporal remote sensing images of Poyang Lake in Jiangxi Province.

The attraction degree of firefly is defined as

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2}, \quad (13)$$

where β_0 is the biggest attraction of the firefly.

In the case that Firefly i is attracted by Firefly j and moves towards the position of Firefly j , it can be defined in the formula below:

$$x_i = x_i + \beta \times (x_j - x_i) + \alpha \times \left(\text{rand} - \frac{1}{2} \right), \quad (14)$$

where α is the step size factor which is a constant between $[0, 1]$; rand is the uniformly distributed random factor between $[0, 1]$; $\alpha \times (\text{rand} - 1/2)$ is the random disturbance to avoid untimely falling into local optimum.

By setting the 2D-Otsu function $t_r \sigma_B(s, t)$ as the target function of the Firefly algorithm, the optimization result of the Firefly algorithm is the position $t_r \sigma_B(s^*, t^*)$ of the firefly with maximum brightness and (s^*, t^*) is the desired threshold. The implementation process is as follows:

- (1) initialize the basic parameters of the Firefly algorithm. Set the number of fireflies as n , of which the maximum attraction is β_0 , the light intensity attraction coefficient is γ , the step factor is α , and the maximum number of iterations is T ;
- (2) by randomly initializing the position of fireflies, calculate the 2D-Otsu function value $t_r \sigma_B(s_i, t_i)$ corresponding to each firefly and take the target function as their brightness to get the position of the firefly with maximum brightness;
- (3) calculate the relative brightness I and attraction degree β of fireflies based on (12) and (13), and determine their movement direction according to the relative brightness;
- (4) update the spatial position of fireflies based on Formula (14), and carry out random disturbance on fireflies in the best position;
- (5) recalculate the brightness of fireflies based on the updated position of fireflies;

- (6) when the maximum number of searches is reached, go to (6); otherwise, the number of searches should plus 1; then, go to (3) for the next search;
- (7) output the position and brightness of the firefly with maximum brightness, and segment the difference images by taking (s^*, t^*) as a threshold.

4. Experiments and Results

4.1. Description of Experimental Data. In order to verify the validity and reliability of the proposed method as well as the advantages of the proposed method on the operating speed, the paper selected two groups of experimental data set, both of which were Landsat5 TM remote sensing images, with a spatial resolution of 30 m. The data set is provided by International Scientific & Technical Data Mirror Site, Computer Network Information Center, and Chinese Academy of Sciences (<http://www.gscloud.cn/>).

The first group of data set consists of two Landsat5 TM remote sensing images of the local area of Poyang Lake in Jiangxi Province, as shown in Figures 1(a) and 1(b). Figures 1(a) and 1(b) were, respectively, acquired on July 15, 1989, and on January 14, 2010, and the size of two remote sensing images was 510×510 pixels, with a gray-scale of 256; their reference changing map is shown in Figure 1(c), where the black area is the changing area, and 69,413 pixels were changed and 190,687 pixels were unchanged.

The second group of data set consists of two Landsat5 TM remote sensing images of the local area of Dongting Lake in Hunan Province, as shown in Figures 4(a) and 4(b). Figures 4(a) and 4(b) were, respectively, acquired on July 19, 1991, and on October 24, 2009, and the size of two remote sensing images was 610×610 pixels, with a gray-scale of 256; their reference changing map is shown in Figure 4(c), where the black area is the changing area, and 102,096 pixels were changed and 270,004 pixels were unchanged.

4.2. Change Detection Results and Analysis. The proposed method was verified by two experimental schemes.

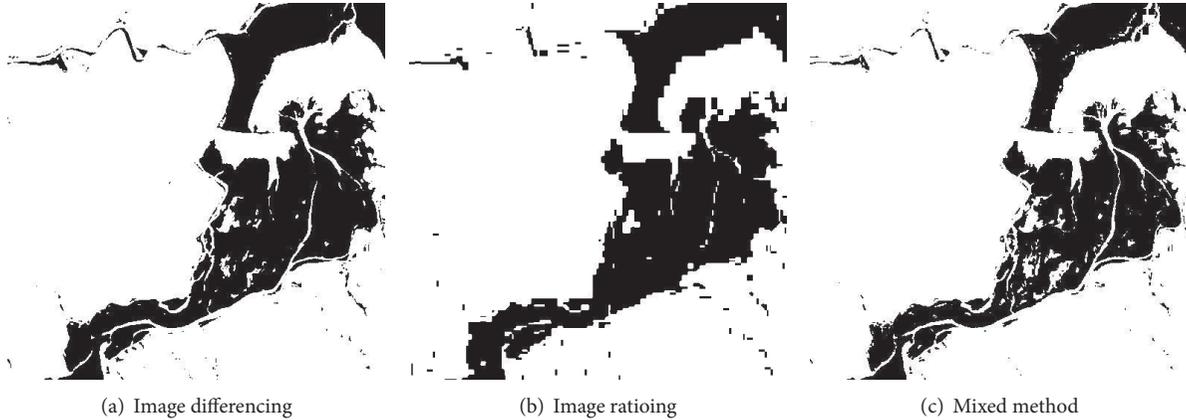


FIGURE 2: The first set of experiments of Poyang Lake in Jiangxi Province.

TABLE 1: False alarms, missed alarms, and total errors for three difference methods in Poyang Lake.

Difference image acquisition methods	False alarms		Missed alarms		Total errors	
	Pixels	%	Pixels	%	Pixels	%
Difference operator	4945	2.59%	8636	12.44%	13581	5.22%
Ratio operator	10142	5.32%	7670	11.05%	17812	6.85%
IMTF operator	3204	1.68%	12499	18.01%	15703	6.04%

The first experimental scheme, respectively, applied difference method, ratio method, and the combination of difference method and ratio method to construct difference images and used the improved 2D-Otsu algorithm based on Firefly algorithm for segmentation to get the binary change map, where, after several experiments, the parameters of the Firefly algorithm were set below: the number of fireflies $n = 50$, the initial attraction degree $\beta_0 = 0.2$, light intensity coefficient $\gamma = 1$, step factor $\alpha = 0.25$, and the maximum number of iterations $T = 100$. The second experimental scheme, respectively, applied the proposed 2D-Otsu improved by Firefly algorithm, two-dimensional maximum entropy and two-dimensional maximum fuzzy entropy to process the difference images obtained from the image difference method to get the binary change map. In order to quantitatively evaluate the accuracy of the proposed method, the false detection rate, missed detection rate, and overall error rate were taken as evaluation factors [2].

In the first experimental scheme, the 2D-Otsu algorithm improved by the proposed Firefly algorithm based on the two groups of data set, respectively, processed the difference images constructed through difference method, ratio method, and the combination of difference method and ratio method to get the binary change map, of which the detection results are shown in Figures 2 and 5, and the change detection accuracy is shown in Tables 1 and 3. As can be seen from Tables 1 and 3, the difference method led to the minimum number of pixel errors, followed by the result produced by the combination method, and the ratio method led to the maximum number of pixel errors.

According to the three evaluation factors, namely, overall false detection rate, missed detection rate, and overall error rate, the difference image resulting from the image difference method can get better change detection results. In addition, as can be seen from Figure 5, the detection results from the ratio method produced a number of missed and false changed pixel elements; although the detection resulting from the combination method reduced the number of false detections, it led to a huge number of missed detections; the detection results from difference method did not produce a number of false or missed detections; thus, it was much closer to the reference map.

In the second experimental scheme, the 2D-Otsu algorithm, two-dimensional maximum entropy and two-dimensional maximum fuzzy entropy improved by the Firefly algorithm based on the two groups of data set processed the difference images obtained from the image difference method to get the binary change map, of which the detection results are shown in Figures 3 and 6, and the comparison results among the change detection accuracy based on the three methods are shown in Tables 2 and 4. As can be seen from Tables 2 and 4, although the detection results based on the two-dimensional maximum entropy and the two-dimensional maximum fuzzy entropy had a very low false detection rate, a number of changed pixel elements were missed. The missed detection rate based on the two-dimensional maximum entropy method was above 80%, while the missed detection rate based on the two-dimensional maximum fuzzy rate was between 55% and 65%. The false detection rate and the missed detection rate based on the proposed method were lower, of which the false detection rate was maintained below 3%, and the missed detection rate was maintained below 13%. As can also be seen from Figures 3 and 6, compared with the reference change map, the binary change detection results based on the two-dimensional maximum entropy and the two-dimensional fuzzy maximum entropy showed a number of changed pixel elements.

As can be seen from the two experimental schemes, the proposed method is an effective and reliable change detection method for multitemporal remote sensing images.

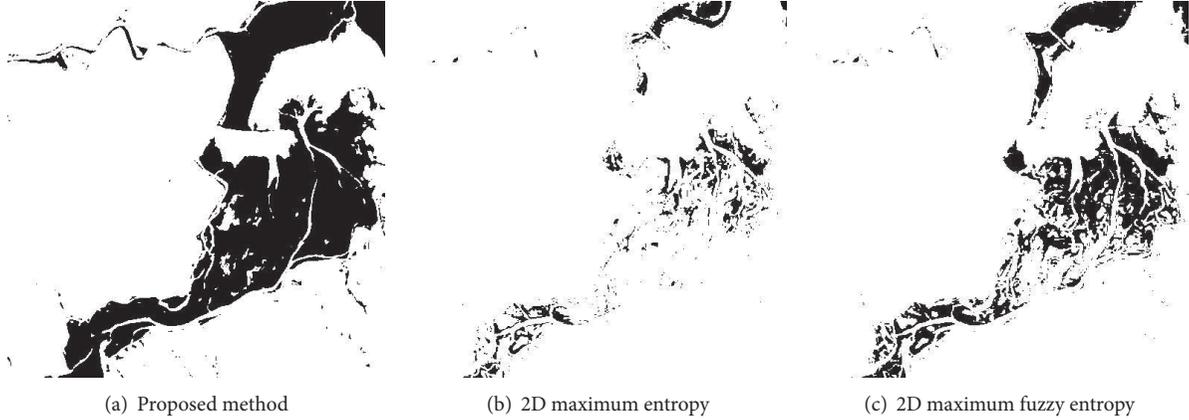


FIGURE 3: The second set of experiments of Poyang Lake in Jiangxi Province.

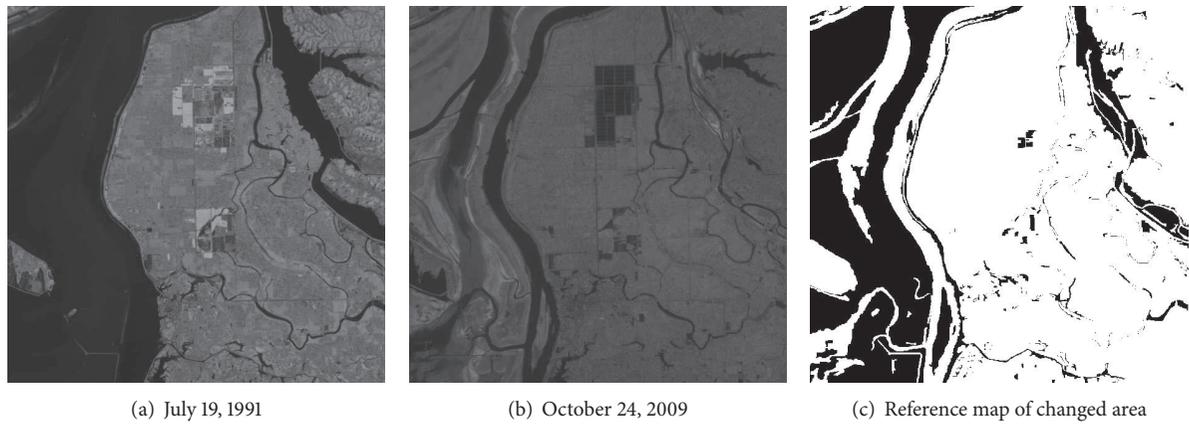


FIGURE 4: Multitemporal remote sensing images of Dongting Lake in Hunan Province.

TABLE 2: False alarms, missed alarms, and total errors for three change detection methods in Poyang Lake.

Change detection methods	False alarms		Missed alarms		Total errors	
	Pixels	%	Pixels	%	Pixels	%
Proposed method	4945	2.59%	8636	12.44%	13581	5.22%
2D maximum entropy	28	0.01%	58870	84.81%	58898	22.64%
2D maximum fuzzy entropy	164	0.09%	39906	57.49%	40070	15.41%

5. Conclusion

Thresholding processing is an effective method to detect changes in multitemporal remote sensing images. One-dimensional thresholding can be simply implemented, but the thresholding accuracy is not high. In order to improve the accuracy of detection change, the 2D-Otsu algorithm is used for the change detection of multitemporal remote sensing images. However, the 2D-Otsu algorithm has difficult threshold selection and requires a large amount of

TABLE 3: False alarms, missed alarms, and total errors for three difference methods in Dongting Lake.

Difference image acquisition methods	False alarms		Missed alarms		Total errors	
	Pixels	%	Pixels	%	Pixels	%
Difference operator	4548	1.68%	11268	11.04%	15816	4.25%
Ratio operator	85119	31.53%	29885	29.27%	115004	30.91%
IMTF operator	1295	0.47%	36496	35.75%	37791	10.16%

computation; thus, the Firefly algorithm is introduced to conduct threshold optimization for 2D-Otsu algorithm, from which the automatic change detection method of multitemporal remote sensing images based on 2D-Otsu algorithm improved by Firefly algorithm has been proposed. First, this paper first applied the image difference method to construct the difference images, and conducted threshold optimization for 2D-Otsu algorithm by using Firefly algorithm; then, the 2D-Otsu algorithm was used to segment the threshold values on the difference images to get the binary change map, and finally the false detection rate, missed detection rate,

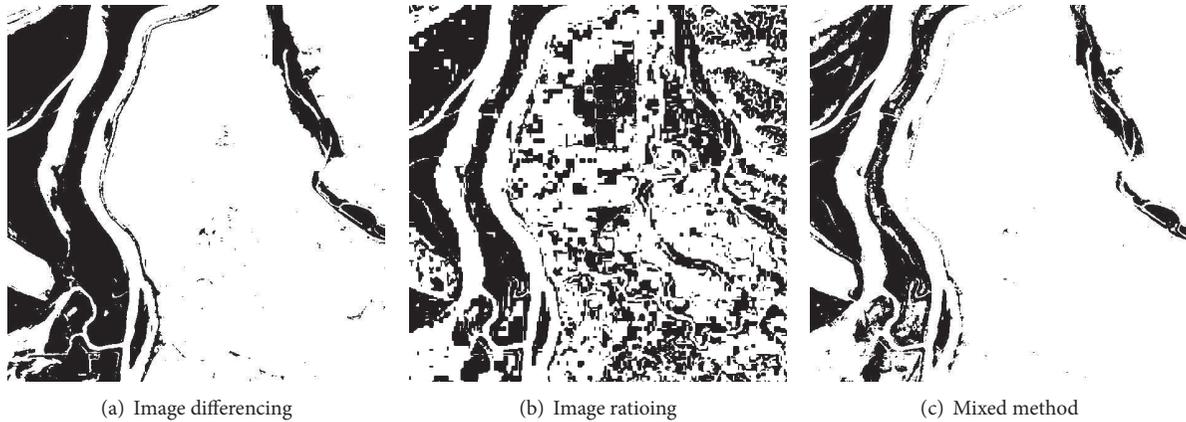


FIGURE 5: The first set of experiments of Dongting Lake in Hunan Province.

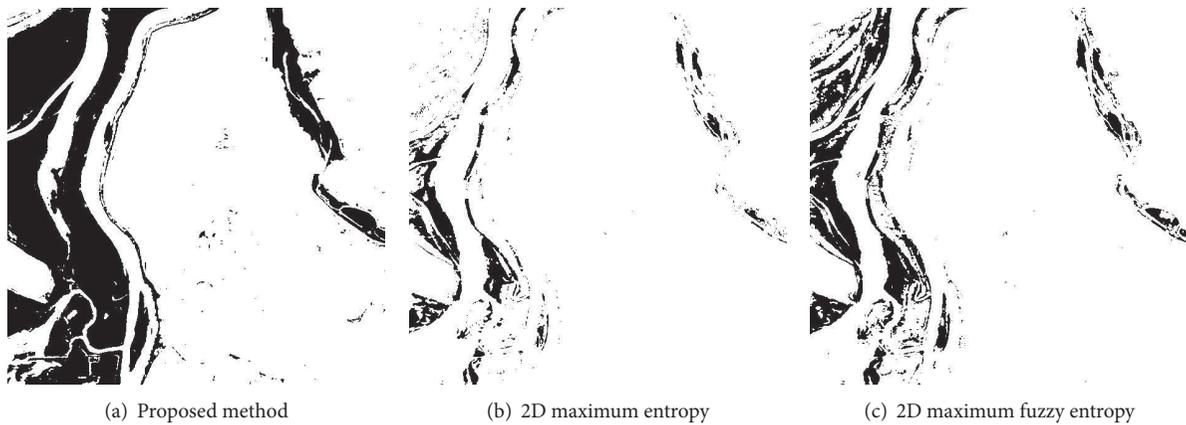


FIGURE 6: The second set of experiments of Dongting Lake in Hunan Province.

TABLE 4: False alarms, missed alarms, and total errors for three change detection methods in Dongting Lake.

Change detection methods	False alarms		Missed alarms		Total errors	
	Pixels	%	Pixels	%	Pixels	%
Proposed method	4548	1.68%	11268	11.04%	15816	4.25%
2D maximum entropy	27	0.01%	83949	82.23%	83976	22.57%
2D maximum fuzzy entropy	183	0.07%	63810	62.50%	63993	17.20%

and overall error rate were separately used to evaluate the accuracy of detection results.

In order to verify the effectiveness, reliability, and operating speed advantages of the proposed method, several change detection experiments were made to the two groups of data set, and the results were compared with the change detection methods based on the two-dimensional maximum entropy and the two-dimensional maximum fuzzy entropy. Experimental results showed that the change detection accuracy (overall error rate) of the proposed method averaged

4.74%, better than the detection accuracies based on the two-dimensional maximum entropy method (22.61%) and the two-dimensional maximum fuzzy entropy method (16.31%). Therefore, the proposed method can efficiently and accurately identify the changing area of multitemporal remote sensing images. Of course, further researches are still required in terms of how to set the initial parameters of fireflies so as to reduce the impacts of nonoptimal initial parameters on threshold optimization.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research work was supported by the National Natural Science Foundation of China (no. 41161071), the Department of Education Research Fund of Yunnan Province (no. 2013J062), and Scholarship Award for Excellent Doctoral Student granted by Yunnan Province. The authors would like to express their thanks to the anonymous reviewer for the constructive comments.

References

- [1] J. Q. Zhong, *Change Detection Based on Multitemporal Remote Sensing Image*, Graduate School of National University of Defense Technology, Hunan, China, 2005.
- [2] F. Bovolo and L. Bruzzone, "A detail-preserving scale-driven approach to change detection in multitemporal SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 12, pp. 2963–2972, 2005.
- [3] N. S. Mishra, S. Ghosh, and A. Ghosh, "Fuzzy clustering algorithms incorporating local information for change detection in remotely sensed images," *Applied Soft Computing Journal*, vol. 12, no. 8, pp. 2683–2692, 2012.
- [4] X. Tong, Z. Hong, S. Liu et al., "Building-damage detection using pre- and post-seismic high-resolution satellite stereo imagery: a case study of the May 2008 Wenchuan earthquake," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 68, no. 3, pp. 13–27, 2012.
- [5] R. S. Lunetta, J. F. Knight, J. Ediriwickrema, J. G. Lyon, and L. D. Worthy, "Land-cover change detection using multi-temporal MODIS NDVI data," *Remote Sensing of Environment*, vol. 105, no. 2, pp. 142–154, 2006.
- [6] J. O. Odindi and P. Mhangara, "Green spaces trends in the city of Port Elizabeth from 1990 to 2000 using remote sensing," *International Journal of Environmental Research*, vol. 6, no. 3, pp. 653–662, 2012.
- [7] H. J. Tang, W. B. Wu, P. Yang, Q. B. Zhou, and Z. X. Chen, "Recent progresses in monitoring crop spatial patterns by using remote sensing," *Scientia Agricultura Sinica*, vol. 43, no. 14, pp. 2879–2888, 2010.
- [8] Y. Q. Tang, X. Huang, and L. Zhang, "Fault-tolerant building change detection from urban high-resolution remote sensing imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 10, no. 5, pp. 1060–1064, 2013.
- [9] H. S. Zhang, *The Research of Object-Based Remote Sensing Change Detection for Coastal Surface*, Zhejiang University, Hangzhou, China, 2010.
- [10] W. B. Sun, H. X. Chen, H. Y. Tang, and G. Yu, "Unsupervised image change detection based on 2D fuzzy entropy," *Journal of Jilin University: Engineering and Technology*, vol. 41, no. 5, pp. 1461–1467, 2011.
- [11] P. A. Rogerson, "Change detection thresholds for remotely sensed images," *Journal of Geographical Systems*, vol. 4, no. 1, pp. 85–97, 2002.
- [12] A. D. Brink, "Thresholding of digital images using two-dimensional entropies," *Pattern Recognition*, vol. 25, no. 8, pp. 803–808, 1992.
- [13] J. Liu and W. Li, "Automatic thresholding of gray-level pictures via two-dimensional OTSU method," *Acta Automatica Sinica*, vol. 19, no. 1, pp. 101–105, 1993.
- [14] M. L. F. Velloso, T. A. Carneiro, and F. J. de Souza, "Unsupervised change detection using fuzzy entropy principle," in *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing*, vol. 4, pp. 2550–2553, September 2004.
- [15] X. Shen, Y. zhag, and F. Li, "An improved two-dimensional entropic thresholding method based on ant colony genetic algorithm," in *Proceedings of the WRI Global Congress on Intelligent Systems (GCIS '09)*, pp. 163–167, May 2009.
- [16] L. Zheng, G. Li, and Y. Bao, "Improvement of grayscale image 2D maximum entropy threshold segmentation method," in *Proceedings of the International Conference on Logistics Systems and Intelligent Management (ICLSIM '10)*, vol. 1, pp. 324–328, IEEE, Harbin, China, January 2010.
- [17] S. Alim, P. J. Du, and S. C. Liu, "Maximum 2D entropy image segmentation based on artificial bee colony optimization," *Computer Engineering*, vol. 38, no. 9, pp. 223–225, 2012.
- [18] X. J. Qian, "Application of PSO algorithm in 2D Otsu image segmentation," *Computer Simulation*, vol. 27, no. 12, pp. 279–281, 2010.
- [19] J. Tian and J. C. Zeng, "2D fuzzy maximum entropy image threshold segmentation method based on QPSO," *Computer Engineering*, vol. 35, no. 3, pp. 230–232, 2009.
- [20] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Computer Science*, pp. 169–178, Springer, Berlin, Germany, 2009.
- [21] K. Chen, F. Chen, M. Dai, Z.-S. Zhang, and J.-F. Shi, "Fast image segmentation with multilevel threshold of two-dimensional entropy based on firefly algorithm," *Optics and Precision Engineering*, vol. 22, no. 2, pp. 517–523, 2014.
- [22] W. K. Alomoush, S. N. H. S. Abdullah, S. Sahran, and R. I. Hussain, "Segmentation of MRI brain images using FCM improved by firefly algorithms," *Journal of Applied Sciences*, vol. 14, no. 1, pp. 66–71, 2014.
- [23] X.-M. Zhang, Y.-J. Sun, and Y.-B. Zheng, "Precise two-dimensional Otsu's image segmentation and its fast recursive realization," *Acta Electronica Sinica*, vol. 39, no. 8, pp. 1778–1784, 2011.
- [24] Q. Chen, L. Zhao, J. Lu, G. Kuang, N. Wang, and Y. Jiang, "Modified two-dimensional Otsu image segmentation algorithm and fast realisation," *IET Image Processing*, vol. 6, no. 4, pp. 426–433, 2012.
- [25] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Beckington, UK, 2008.
- [26] X. G. Du, J. W. Dang, Y. P. Wang, X. G. Liu, and S. Li, "Mutual information medical image registration based on firefly algorithm," *Computer Science*, vol. 40, no. 7, pp. 273–276, 2013.
- [27] C. P. Liu and C. M. Ye, "Novel bioinspired swarm intelligence optimization algorithm: firefly algorithm," *Application Research of Computers*, vol. 28, no. 9, pp. 3295–3297, 2011.

Research Article

Automatic Fusion of Hyperspectral Images and Laser Scans Using Feature Points

Xiao Zhang,^{1,2} Aiwu Zhang,² and Xiangang Meng²

¹Research Center of Artistic Heritage, Taiyuan University of Technology, Taiyuan 030012, China

²Key Laboratory of 3D Information Acquisition and Application, Capital Normal University, Beijing 100048, China

Correspondence should be addressed to Aiwu Zhang; zhangaw86@163.com

Received 14 November 2014; Accepted 7 February 2015

Academic Editor: Xue Cheng Tai

Copyright © 2015 Xiao Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Automatic fusion of different kinds of image datasets is so intractable with diverse imaging principle. This paper presents a novel method for automatic fusion of two different images: 2D hyperspectral images acquired with a hyperspectral camera and 3D laser scans obtained with a laser scanner, without any other sensor. Only a few corresponding feature points are used, which are automatically extracted from a scene viewed by the two sensors. Extraction method of feature points relies on SURF algorithm and camera model, which can convert a 3D laser scan into a 2D laser image with the intensity of the pixels defined by the attributes in the laser scan. Moreover, Collinearity Equation and Direct Linear Transformation are used to create the initial corresponding relationship of the two images. Adjustment is also used to create corrected values to eliminate errors. The experimental result shows that this method is successfully validated with images collected by a hyperspectral camera and a laser scanner.

1. Introduction

Hyperspectral imaging technology can quickly detect hundreds or even thousands of different light frequencies and relative intensities of surface features, which is unlike regular cameras that are typically sensitive to only three different frequencies (red, green, and blue). Laser scanning technology can quickly obtain the accurate geometry information of surface features in despite of the adverse external circumstances. If the hyperspectral data and the laser data can be fused, the spectral information and the spatial information of the same location can be obtained at the same time, which can effectively make up for the deficiency of single data source.

Currently, the registration and fusion of hyperspectral images and laser scans have been the research hotspots. However, because of the multiple different sensors and the different imaging modalities the fusion is very complicated. The most common approach is to perform registration using manual methods. However, this approach is very low precision in practice. Only several methods exist for aligning these two datasets of the same location. Nieto et al. [1] installed

a digital camera on top of the laser to acquire the color point clouds and translated it into the 2D color image; then the registration is completed by the piecewise linear transform. Kurz et al. [2] used two sensors to detect the position of the target and then corrected them to complete the registration. Zachary and Juan [3] obtained initial position between two sensors through GPS and then used the mutual information to achieve the registration. In addition, some methods of aligning regular digital image with laser scan can be used for reference. For example, Tsai camera calibration method [4, 5] was used to obtain the 2D-3D homologous points and the unknown parameters to implement the registration. However, the precise corresponding points were difficultly found because of the difference of color structure. The stereo matching [6] was used to convert multi-images to 3D point clouds to realize the 2D-3D registration. However, this method did not realize the registration of single image and point cloud. Moreover, the mutual information [7-9] was also used to complete the 2D-3D registration. The collinear equation was used to construct 2D-3D correspondence [10, 11] to implement the registration of aerial images and laser scans.

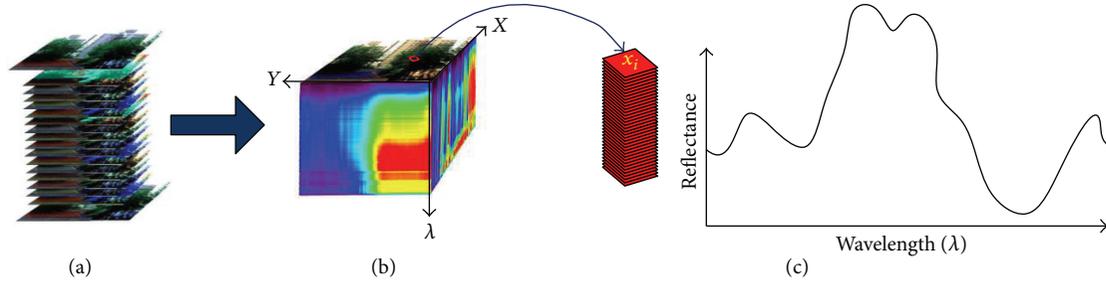


FIGURE 1: Hyperspectral Image cube model.

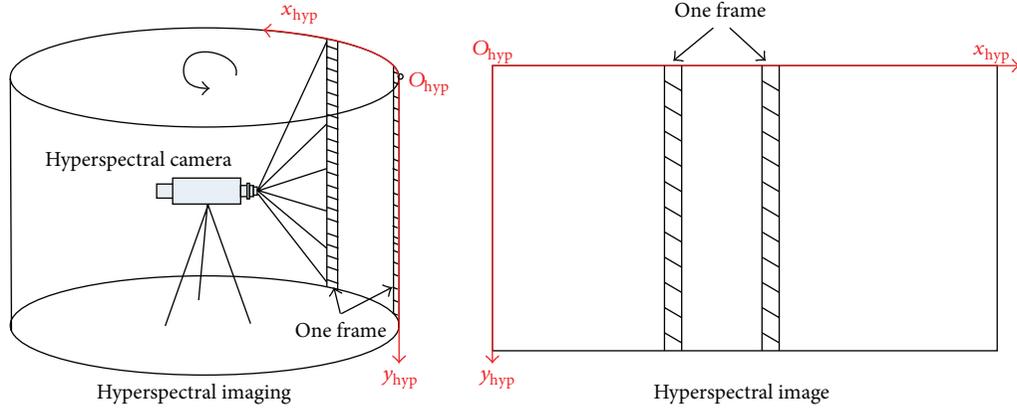


FIGURE 2: Coordinate system of Hyperspectral Image.

In summary, the attempt in this paper is to create a method that can correctly register and fuse the hyperspectral data and the laser data without the additional sensors. The remainder of this paper is organized as follows. Section 2 provides the mathematical model of this algorithm. Section 3 in great detail describes the algorithm of automatic fusion methodology. Finally Section 5 makes the concluding remarks and maps out the directions for future work.

2. Derivation of Mathematical Model

2.1. Definition of Coordinate System

2.1.1. Coordinate System of Hyperspectral Image. The data model of hyperspectral image, different from the model of remote sensing image and digital image, is the feature vector representation model and can be expressed by the data cube model, as shown in Figure 1. x -axis and y -axis denote the space dimensions and λ -axis denotes the spectral band. XOY -plane is the image information of a band or multiband of hyperspectral image; $XO\lambda$ -plane and $YO\lambda$ -plane are the spectrum information of a hyperspectral image line, as shown in Figures 1(a) and 1(b). The cube model and the spectrum oscillogram of a hyperspectral pixel are shown in Figure 1(c).

Rotary broom hyperspectral camera scans imaging with the line array scanning mode and the hyperspectral image is the 2D image. Imaging geometric model is to create the mapping relationship between object space and image space, as shown in Figure 2, where O_{hyp} is the coordinate origin;

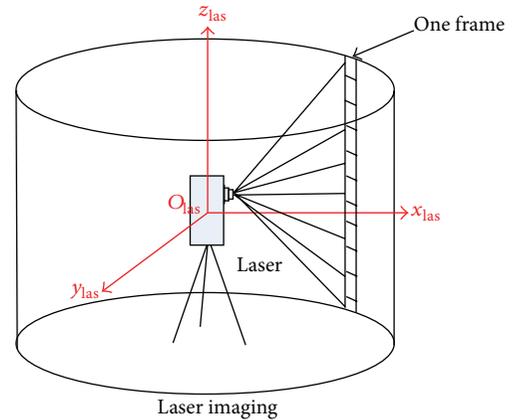


FIGURE 3: Coordinate system of Laser Scan.

x_{hyp} is the abscissa, which is along the rotation direction of hyperspectral camera from the rotation starting location; y_{hyp} is the ordinate, which is along the rotation vertical direction. The horizontal resolution is determined with the rotation speed of turntable and the vertical resolution is determined by the scanning speed of hyperspectral camera.

2.1.2. Coordinate System of Laser Scans. Terrestrial 3D laser scans imaging with the line array scanning mode and the laser image is 3D point clouds, whose coordinate system is confirmed by the self-laser, as shown in Figure 3, where O_{las} is the coordinate origin, which is the scanner electrooptical

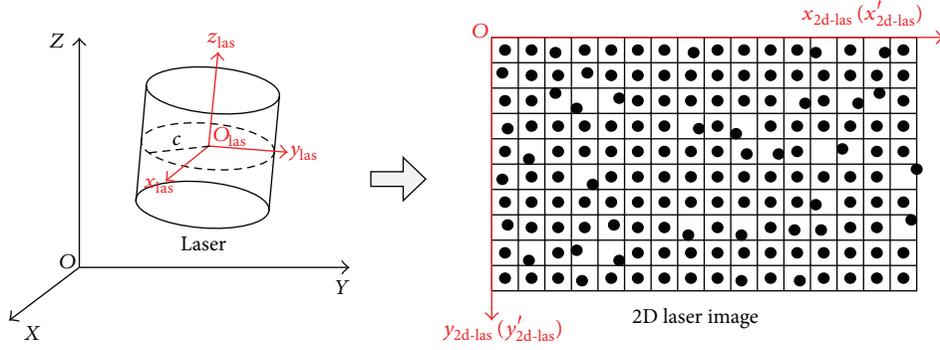


FIGURE 4: Camera model.

center; z_{las} is the scanner vertical rotation axis; x_{las} is along the optical axis scanner arbitrary horizontal angle, such as the first horizontal angle or the north direction of the built-in magnetic compass; y_{las} is orthogonal to x_{las} and z_{las} , which is formed by the right-hand system.

2.2. Definition of Camera Model. Rotary broom hyperspectral camera and Terrestrial 3D laser are both dependent on the cylindrical coordinate system, but their images are, respectively, 2D and 3D, so the camera model is formed to transform the 3D laser scan into the 2D laser image.

This camera model is inferred by panoramic camera model [12, 13], as shown in Figure 4. The formula is shown in

$$\begin{aligned} x_{2d-las} &= x_0 - c \arctan\left(-\frac{y_{las}}{x_{las}}\right) + \Delta x, \\ y_{2d-las} &= y_0 - \frac{cz_{las}}{\sqrt{x_{las}^2 + y_{las}^2}} + \Delta y, \end{aligned} \quad (1)$$

where $(x_{las}, y_{las}, z_{las})$ is the point of laser scan; (x_{2d-las}, y_{2d-las}) is the point of 2D laser image; c is the principle distance of the camera model; (x_0, y_0) is the principle point of 2D laser image; $(\Delta x, \Delta y)$ is the correction parameter.

Through (1), the point of 2D laser image (x_{2d-las}, y_{2d-las}) is calculated, but the type of their values is double and the type of 2D image pixel is integer, so (x_{2d-las}, y_{2d-las}) must be changed into the integer value, as shown in

$$\begin{aligned} x'_{2d-las} &= \text{int}\left(\frac{x_{2d-las} - \min(x_{2d-las})}{\max(x_{2d-las}) - \min(x_{2d-las})}\right) \\ &\quad \times \text{int}\left(\frac{\max(x_{2d-las}) - \min(x_{2d-las})}{A_m}\right), \\ y'_{2d-las} &= \text{int}\left(\frac{y_{2d-las} - \min(y_{2d-las})}{\max(y_{2d-las}) - \min(y_{2d-las})}\right) \\ &\quad \times \text{int}\left(\frac{\max(y_{2d-las}) - \min(y_{2d-las})}{A_n}\right), \end{aligned} \quad (2)$$

where $(x'_{2d-las}, y'_{2d-las})$ is the point position of the 2D image; A_m is the horizontal resolution; A_n is the vertical resolution.

The pixels value of 2D laser image is defined by the attributes of the laser scan, which may be the information such as color, curvature, and normal, and so forth. The formula is as follows:

$$\text{color} = \frac{\text{col} - \min y}{\max y - \min y} \times 255, \quad (3)$$

where color is the value of $(x'_{2d-las}, y'_{2d-las})$, ranged in [0–255]; col is the attribute value of $(x_{las}, y_{las}, z_{las})$; max y and min y are, respectively, the maximum and the minimum of this attribute.

3. Algorithm of Automatic Fusion

In this automatic fusion algorithm, first of all, the hyperspectral gray image (image_{hyp}) is extracted from the hyperspectral image and the 2D laser image (image_{las}) from the laser scan is created by the camera model. And then, the feature points of (image_{hyp}) and (image_{las}) are produced with SURF and SC-RANSAC, and the feature points of hyperspectral image and laser scan are generated with the inverse operation of camera model. The initial registration is achieved by Collinearity Equation and Direct Linear Transformation; the precision registration is completed through creating corrected values to eliminate errors with Adjustment. At length, the automatic fusion is accomplished by the registration result. This algorithm flowing chart is shown in Figure 5.

3.1. Initial Registration

3.1.1. Extraction of Feature Points. Feature Points between (image_{hyp}) and (image_{las}) are extracted by SURF algorithm [14], which is superior to SIFT algorithm in every aspect [15–17]. In order to eliminate the error feature points, SC-RANSAC [18] is used, which is the currently fastest RANSAC extension coalesced with RANSAC and spatial consistency check from the literature [19]. Moreover, the searching method, an important role to improve the speed, is the stochastic KD tree algorithm [20], which searches by multiple

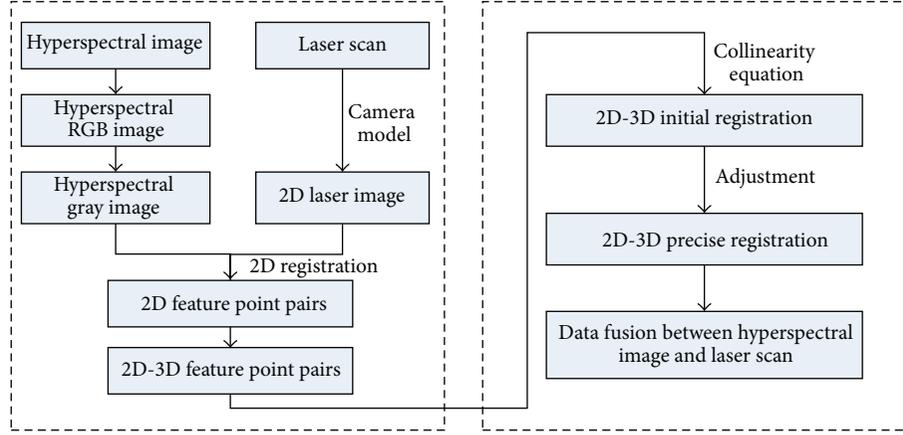


FIGURE 5: Algorithm flowing chart.

stochastic KD tree to advance searching nodes and whose accuracy and matching speed are better for high dimensional data search [21, 22].

After completing the extraction of feature points, according to the inversion operation of camera model, the corresponding 3D feature points of $(image_{las})$ is found. If P_i ($i = 1, \dots, n$) are the feature points of hyperspectral image and Q_i ($i = 1, \dots, n$) are the corresponding feature points of laser scan, the corresponding feature points between them are found, too.

3.1.2. Collinearity Equation. The Collinearity Equation, the basic equation of photogrammetry, is used to set up the mapping relationship between the 2D coordinate and the 3D coordinate. If $P(x_{hyp}, y_{hyp})$ in the hyperspectral coordinate system corresponds to $Q(x_{las}, y_{las}, z_{las})$ in the laser coordinate system, the corresponding relation between them can be expressed as

$$\begin{aligned} x_{hyp} - x_0 + \delta x &= -f \frac{a_1 x_{las} + b_1 y_{las} + c_1 z_{las}}{a_3 x_{las} + b_3 y_{las} + c_3 z_{las}} \\ y_{hyp} - y_0 + \delta y &= -f \frac{a_2 y_{las} + b_2 y_{las} + c_2 z_{las}}{a_3 y_{las} + b_3 y_{las} + c_3 z_{las}}, \end{aligned} \quad (4)$$

where (a_i, b_i, c_i) $i = 1, 2, 3$ is the direction cosine of rotation matrix; $(\delta x, \delta y)$ is the system errors correction; f is the principal distance.

Direct Linear Transformation (DLT) is the solution of direct linear relationship between the photo point coordinate and the corresponding object point coordinate, which is essentially a kind of space resection and space intersection method. This algorithm is applicable to a variety of no metric cameras without the known internal orientation elements and is also suitable for the close range photogrammetry of the large angle without the initial external orientation elements. According to DLT algorithm, (4) is translated into the following formula:

$$\begin{bmatrix} x_{las1} & y_{las1} & z_{las1} & 1 & 0 & 0 & 0 & 0 & x_{hyp1}x_{las1} & x_{hyp1}y_{las1} & x_{hyp1}z_{las1} \\ 0 & 0 & 0 & 0 & x_{las1} & y_{las1} & z_{las1} & 1 & y_{hyp1}x_{las1} & y_{hyp1}y_{las1} & y_{hyp1}z_{las1} \\ x_{las2} & y_{las2} & z_{las2} & 1 & 0 & 0 & 0 & 0 & x_{hyp2}x_{las2} & x_{hyp2}y_{las2} & x_{hyp2}z_{las2} \\ 0 & 0 & 0 & 0 & x_{las2} & y_{las1} & z_{las2} & 1 & y_{hyp2}x_{las2} & y_{hyp2}y_{las2} & y_{hyp2}z_{las2} \\ & & \vdots & & & & \vdots & & & & \vdots \\ x_{lasn} & y_{lasn} & z_{lasn} & 1 & 0 & 0 & 0 & 0 & x_{hypn}x_{lasn} & x_{hypn}y_{lasn} & x_{hypn}z_{lasn} \\ 0 & 0 & 0 & 0 & x_{lasn} & y_{las1n} & z_{lasn} & 1 & y_{hypn}x_{lasn} & y_{hypn}y_{lasn} & y_{hypn}z_{lasn} \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \\ l_7 \\ l_8 \\ l_9 \\ l_{10} \\ l_{11} \end{bmatrix} = \begin{bmatrix} -x_{hyp1} \\ -y_{hyp1} \\ -x_{hyp2} \\ -y_{hyp2} \\ \vdots \\ -x_{hypn} \\ -y_{hypn} \end{bmatrix}, \quad (5)$$

where l_i ($i = 1, \dots, 11$) are the coefficients. According to these coefficients, the interior orientation elements and the exterior

orientation elements between hyperspectral coordinate and laser coordinate are calculated. The corresponding relation

between $P(x_{\text{hyp}}, y_{\text{hyp}})$ and $Q(x_{\text{las}}, y_{\text{las}}, z_{\text{las}})$ is also calculated. However, because of the great errors of DTL, l_i ($i = 1, \dots, 11$) can only be considered the approximate results, and the corresponding relation results are the initial registration results.

3.2. Precise Registration. In order to accurately determine the corresponding relation between hyperspectral coordinate and laser coordinate, Adjustment is executed using the redundant observation values of the hyperspectral pixels. If the correction of the hyperspectral pixel observed value is (v_x, v_y) , the corresponding relation between $P(x_{\text{hyp}}, y_{\text{hyp}})$ and $Q(x_{\text{las}}, y_{\text{las}}, z_{\text{las}})$ is shown in

$$\begin{aligned} x_{\text{hyp}} - x_0 + v_x + \delta x &= -f \frac{a_1 x_{\text{las}} + b_1 y_{\text{las}} + c_1 z_{\text{las}}}{a_3 x_{\text{las}} + b_3 y_{\text{las}} + c_3 z_{\text{las}}}, \\ y_{\text{hyp}} - y_0 + v_y + \delta y &= -f \frac{a_2 x_{\text{las}} + b_2 y_{\text{las}} + c_2 z_{\text{las}}}{a_3 x_{\text{las}} + b_3 y_{\text{las}} + c_3 z_{\text{las}}}. \end{aligned} \quad (6)$$

If $A = l_9 x_{\text{las}} + l_{10} y_{\text{las}} + l_{11} z_{\text{las}} + 1$, the error equation of (v_x, v_y) is shown in

$$\begin{aligned} v_x &= -\frac{1}{A} (x_{\text{las}} l_1 + y_{\text{las}} l_2 + z_{\text{las}} l_3 + l_4 \\ &\quad + x_{\text{hyp}} x_{\text{las}} l_9 + x_{\text{hyp}} y_{\text{las}} l_{10} \\ &\quad + x_{\text{hyp}} z_{\text{las}} l_{11} + A (x_{\text{hyp}} - x_0) r^2 k_1 + x_{\text{hyp}}) \\ v_y &= -\frac{1}{A} (x_{\text{las}} l_5 + y_{\text{las}} l_6 + z_{\text{las}} l_7 + l_8 \\ &\quad + y_{\text{hyp}} x_{\text{las}} l_9 + y_{\text{hyp}} y_{\text{las}} l_{10} \\ &\quad + y_{\text{hyp}} z_{\text{las}} l_{11} + A (y_{\text{hyp}} - y_0) r^2 k_1 + y_{\text{hyp}}), \end{aligned} \quad (7)$$

where k_1 is the symmetric radial distortion coefficient; $r = \sqrt{(x_{\text{hyp}} - x_0)^2 + (y_{\text{hyp}} - y_0)^2}$ is the radius vector of hyperspectral pixel.

The operation is executed by least squares method, whose iterative condition is that the interpolation of adjacent f_x in $f_x = \sqrt{A} \cos d\beta$ is less than 0.01 mm. The calculating process of A value is also the iterative process, and each iterative A value is calculated by control point. Thus, the precise values of the coefficients $(l_1, l_2, \dots, l_{11}, k_1)$ are calculated.

3.3. Data Fusion. If Hype (x, y) is the arbitrary point of hyperspectral image and Point (X, Y, Z) in laser scan is the corresponding point of Hype (x, y) , according to l_i ($i = 1, \dots, 11$), the corresponding relation between Hype (x, y) and Point (X, Y, Z) is calculated by

$$\begin{aligned} x &= -\frac{l_1 X + l_2 Y + l_3 Z + l_4}{l_9 X + l_{10} Y + l_{11} Z + 1}, \\ y &= -\frac{l_5 X + l_6 Y + l_7 Z + l_8}{l_9 X + l_{10} Y + l_{11} Z + 1}. \end{aligned} \quad (8)$$

Therefore, based on (8), each point of laser data corresponds to each point of hyperspectral data.



FIGURE 6: Data acquisition map.

TABLE 1: Hyperspectral camera parameters.

Name	Parameter
Spectrum range	400–1000 nm
Spectral resolution	2.8 nm
FOV	27.2°
Spatial resolution	1600 pixel
Spectrometer entrance slit width	30 μm
Spectrometer entrance slit length	11.84 mm
Camera lens focal length	23 mm
Spectral band numbers	840

In case Hype (p_1, \dots, p_n) is the spectral information of Hype (x, y) in hyperspectral image, and Point $(X, Y, Z, q_1, \dots, q_m)$ is the property of Point (X, Y, Z) in laser scan, in which q_1, \dots, q_m is the feature except the spatial coordinates (X, Y, Z) , such as intensity, amplitude, and so forth, the hyperspectral data and the laser data are fused, and the property of Point (X, Y, Z) is expressed by Point $(X, Y, Z, q_1, \dots, q_m, p_1, \dots, p_n)$ which includes the spatial information of laser data and the spectral information of hyperspectral data, based on the corresponding relationship between Point (X, Y, Z) and Hype (x, y) .

4. Experiment and Analysis

In order to verify the effectiveness of the algorithm, the experiments are conducted. The dataset of the hyperspectral data and the laser data were obtained from the electronic display board of the playground of Capital Normal University by our own laboratory. The 3D laser used was a Riegl LSM-420i and the hyperspectral imager was integrated by our laboratory, which main parameters are shown in Table 1. The setup of data acquisition is shown in Figure 6. The algorithm code was written in Matlab with mex files written in Visual C++. The code was run on a dell computer with Inter i5CPU and 4 G RAM.

The initial RGB image data of hyperspectral data is shown in Figure 7(a). The initial laser scan is shown in Figure 7(b), whose horizontal and vertical angular resolution are 1° and whose horizontal and vertical spacing are about 7 cm, and the different colors show the intensity of laser scan.

4.1. Experiment. In this paper, because the geometry of electronic display board is very flat and the normal vectors are

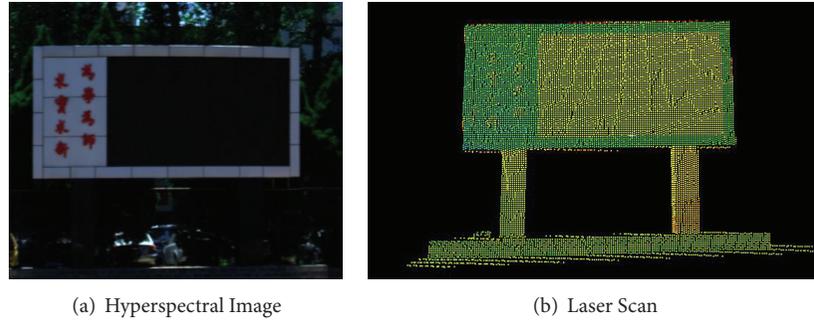


FIGURE 7: Initial data.

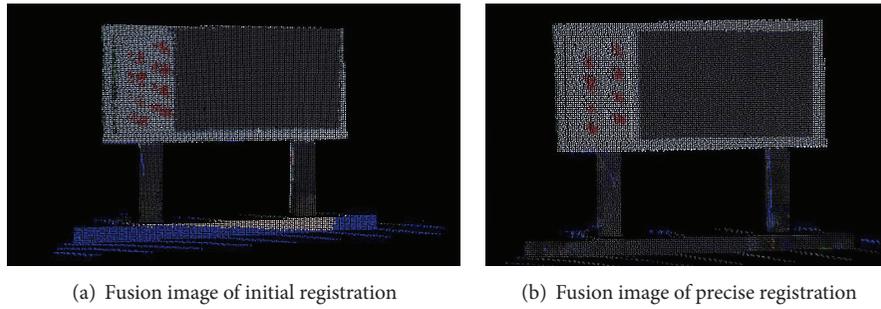


FIGURE 8: Fusion images.

TABLE 2: Feature points of Hyperspectral Image and Laser Scan.

Hyperspectral Image	Laser Scan
(78, 110)	(36.524, -2.535, 5.610)
(64, 748)	(35.819, -10.884, 5.804)
(369, 750)	(35.805, -11.016, 1.834)
(356, 292)	(36.378, -4.905, 2.058)
(600, 261)	(36.318, -4.564, -1.019)
(582, 667)	(35.895, -9.884, -0.898)

similar, so the point-cloud image is generated by the intensity as a pixel value and the image resolution is determined by the point-cloud distance. Then the points of hyperspectral image and laser scan are found by 2D-3D registration algorithm, and six pairs of feature points are chosen, as shown in Table 2.

According to Collinearity Equation and DLT in initial registration, the approximate coefficients $L_1 (l_1, l_2, \dots, l_{11})$ are obtained, as shown in Table 3.

According to Adjustment in precise registration, the precise coefficients $L_2 (l_1, l_2, \dots, l_{11})$ are obtained, as shown in Table 4.

The fusions of hyperspectral data and laser data are, respectively, executed by $L_1 (l_1, l_2, \dots, l_{11})$ and $L_2 (l_1, l_2, \dots, l_{11})$, as shown in Figure 8. The initial fusion image has basically been fused; nevertheless there are more great errors, as shown in Figure 8(a). For instance, the texts of electronic board have obvious deviation. The blue points represent no corresponding points, so the base of electronic board does not have the corresponding points. The precise fusion image has

greatly been fused, as shown in Figure 8(b). The texts and the base of electronic board have been the corresponding fusion.

4.2. Evaluation of Precision. To further verify the effectiveness of this algorithm, other feature points are selected as the check points to verify its accuracy. Firstly, the internal orientation elements and the external orientation elements are calculated by the approximate solution algorithm, and the corresponding hyperspectral pixels of laser scan are calculated. Then, the corresponding hyperspectral pixels of laser scan are calculated by the precise solution algorithm. The comparison is shown in Table 5. “Hyperspectral Image” is the hyperspectral coordinate, and “Laser Scan” is the laser coordinate. “Hyperspectral Image (L_1)” is the hyperspectral coordinate calculated by the approximate solution algorithm, and “Hyperspectral Image (L_2)” is the hyperspectral coordinate calculated by the precise solution algorithm.

To verify the errors of the check points, the residual errors of the horizontal and vertical direction are, respectively, calculated based on the distance between “Hyperspectral Image” and “Hyperspectral Image (L_1)” and “Hyperspectral Image (L_2)”, as shown in Table 6. “ $dx1$ ” and “ $dy1$ ” are, respectively, the residual errors of the horizontal and vertical direction between “Hyperspectral Image” and “Hyperspectral Image (L_1)”. “ $dx2$ ” and “ $dy2$ ” are, respectively, the residual errors of the horizontal and vertical direction between “Hyperspectral Image” and “Hyperspectral Image (L_2)”. From Table 6, the horizontal residual mean errors are decreased from -20.8094 to 4.6046 and the vertical residual mean errors are decreased from -27.8079 to 0.1148 . Precision is greatly improved.

TABLE 3: Approximate coefficients.

Approximate coefficients $L_1 (l_1, l_2, \dots, l_{11})$	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}
	11.7084	-0.9069	2.1177	-444.3699	7.4423	1.6245	-0.0975	-270.4678	-0.0265	0.0019	0.0002

TABLE 4: Precise coefficients.

Precise coefficients $L_2 (l_1, l_2, \dots, l_{11})$	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}
	11.5679	-0.9045	1.6447	-436.0369	7.4790	1.1372	-0.0844	-272.3880	-0.0267	0.0020	0.0002

TABLE 5: The corresponding coordinates of check points.

	Hyperspectral Image	Laser Scan	Hyperspectral Image (L_1)	Hyperspectral Image (L_2)
1	(105, 175)	(36.524, -3.241, 5.219)	(101.510, 163.671)	(104.060, 172.998)
2	(96, 752)	(35.819, -10.884, 5.804)	(90.316, 707.821)	(99.710, 754.157)
3	(369, 772)	(35.805, -11.016, 1.834)	(368.462, 721.168)	(396.928, 73.661)
4	(425, 130)	(36.482, -2.656, 1.663)	(396.083, 120.457)	(423.808, 28.865)
5	(365, 317)	(36.325, -4.961, 2.052)	(360.131, 295.554)	(386.726, 316.734)
6	(120, 313)	(36.325, -4.900, 5.351)	(112.758, 295.324)	(121.541, 314.931)
7	(430, 73)	(36.698, -2.672, 1.609)	(389.679, 81.230)	(426.318, 72.521)
8	(385, 316)	(36.378, -4.905, 2.058)	(355.868, 291.805)	(383.162, 314.289)
9	(670, 755)	(36.133, -9.203, -0.839)	(593.862, 661.657)	(666.248, 752.267)
10	(609, 126)	(36.158, -1.873, -1.577)	(597.238, 112.234)	(611.544, 129.725)

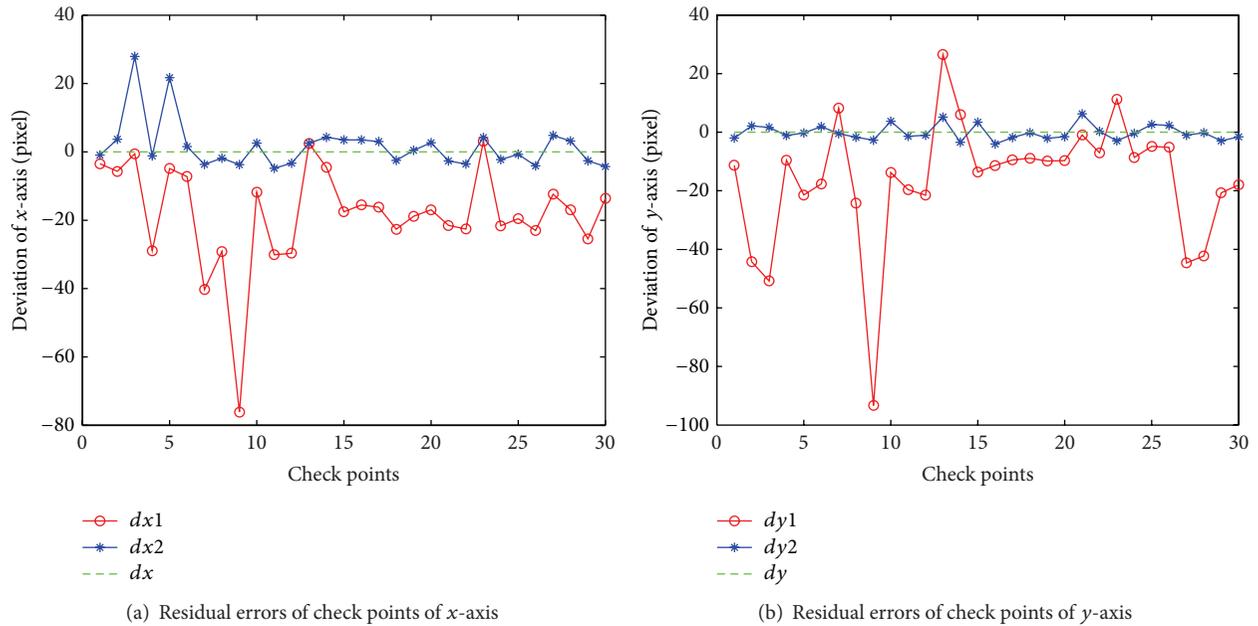


FIGURE 9: Analysis of residual errors of check points.

Moreover, the twenty check points are selected to verify this algorithm and the residual errors of the thirty check points are shown in Figure 9. In agreement with the above analysis, the residual errors of precise registration shapely reduce, and the horizontal residual mean errors are decreased from -18.3751 to 1.5820 and the vertical residual mean errors are decreased from -16.3553 to -0.11167 . To sum up, this method has reached a more satisfactory accuracy.

For the dataset used with natural images, very few methods are appropriate to attempt comparison with. We implemented the other approaches. However no images were successfully aligned by these methods. This failure was expected. The following are the reasons. Firstly, the methods are very few and the initial conditions are so many. For example, the method in Nieto et al. [1] needs that color point clouds are created by the calibration of laser scanner and

TABLE 6: Residual errors and residual mean errors of check points (unit: pixel).

	dx_1	dx_2	dy_1	dy_2
1	-3.4900	-0.9400	-11.3293	-2.0016
2	-5.6836	3.7100	-44.1793	2.1570
3	-0.5383	27.9281	-50.8319	1.6609
4	-28.9171	-1.1921	-9.5426	-1.1347
5	-4.8696	21.7266	-21.4457	-0.2665
6	-7.2416	1.5411	-17.6757	1.9312
7	-40.3209	-3.6824	8.2298	-0.4789
8	-29.1323	-1.8383	-24.1954	-1.7114
9	-76.1384	-3.7516	-93.3432	-2.7327
10	-11.7618	2.5444	-13.7657	3.7246
Residual mean errors	-20.8094	4.6046	-27.8079	0.1148

digital camera. The method in Zachary and Juan [3] needs that the initial site of the two sensors is obtained by GPS. However, the method in this paper does not need the initial site of the two sensors and the additional device such as digital camera and GPS. Moreover, there are some comparisons of the different registration methods of images and laser scans. For example, the method in Zhang et al. [10] applies the inspection line and collinear equation, but this algorithm is more suitable for the aerial image and airborne data. The method in Liu [23] needs to select manually 2D-3D the same points, which causes the great error of human factor.

5. Conclusion

A method for fusing the hyperspectral data with the laser data suitable for surface features is presented. This method operates by creating a 2D laser image using a camera model and extracting feature points of hyperspectral image and laser scan. The collinearity equation is used to create the correspondence to find correct alignment of hyperspectral image with laser scan. The adjustment is used to improve the registration accuracy of hyperspectral image with laser scan. The method was demonstrated to successfully work to fuse a hyperspectral image and a laser scan. In future work a dataset with nature environments will be obtained and the features will be more complex; therefore the feature extraction and the accuracy advance need further strengthening.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Science and Technology Support Program of China (2012BAH31B01), Key Program of Natural Science Foundation of Beijing, China (B) (KZ201310028035), and Youth Foundation of Taiyuan University of Technology, China (2013W014).

References

- [1] J. I. Nieto, S. T. Monteiro, and D. Viejo, "3D geological modelling using laser and hyperspectral data," in *Proceedings of the 30th IEEE International Geoscience and Remote Sensing Symposium (IGARSS '10)*, pp. 4568–4571, IEEE, Honolulu, Hawaii, USA, July 2010.
- [2] T. H. Kurz, S. J. Buckley, J. A. Howell, and D. Schneider, "Integration of panoramic hyperspectral imaging with terrestrial lidar data," *The Photogrammetric Record*, vol. 26, no. 134, pp. 212–228, 2011.
- [3] T. Zachary and N. Juan, "A mutual information approach to automatic calibration of camera and lidar in natural environments," in *Proceedings of the Australasian Conference on Robotics and Automation*, pp. 3–5, 2012.
- [4] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
- [5] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno, "Acquiring, stitching and blending diffuse appearance attributes on 3D models," *Visual Computer*, vol. 18, no. 3, pp. 186–204, 2002.
- [6] J. Shao, A. Zhang, S. Wang, X. Meng, L. Yang, and Z. Wang, "Research on fusion of 3D laser point clouds and CCD image," *Chinese Journal of Lasers*, vol. 40, no. 5, Article ID 0514001, 2013.
- [7] G. Palma, M. Corsini, M. Dellepiane, and R. Scopigno, "Improving 2D-3D registration by mutual information using gradient maps," in *Proceedings of the 8th Eurographics Italian Chapter Conference*, pp. 89–94, The Eurographics Association, November 2010.
- [8] M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno, "Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1755–1764, 2009.
- [9] A. Mastin, J. Kepner, and J. Fisher III, "Automatic registration of LIDAR and optical images of urban scenes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR '09)*, pp. 2639–2646, IEEE, June 2009.
- [10] L. Zhang, H. Ma, G. Gao, and Z. Chen, "Automatic registration of urban aerial images with airborne LiDAR points based on line-point similarity invariants," *Acta Geodaetica et Cartographica Sinica*, vol. 43, no. 4, pp. 372–379, 2014.
- [11] H. Ma, C. Yao, and J. Wu, "Registration of LiDAR point clouds and high resolution images based on linear features," *Geomatics and Information Science of Wuhan University*, vol. 37, no. 2, pp. 136–159, 2012.
- [12] D. Schneider and H.-G. Maas, "A geometric model for linear-array-based terrestrial panoramic cameras," *The Photogrammetric Record*, vol. 21, no. 115, pp. 198–210, 2006.
- [13] D. Schneider and G. H. Maas, "Geometric modelling and calibration of a high resolution panoramic camera," in *Optical 3-D Measurement Techniques VI*, vol. 2, pp. 122–129, 2003.
- [14] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [15] X. Pan and S. Lyu, "Detecting image region duplication using SIFT features," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 1706–1709, March 2010.

- [16] L. Juan and O. Gwun, "SURF applied in panorama image stitching," in *Proceedings of the 2nd International Conference on Image Processing Theory, Tools and Applications (IPTA '10)*, pp. 495–499, Paris, France, July 2010.
- [17] A. C. Murillo, J. J. Guerrero, and C. Sagüés, "SURF features for efficient robot localization with omnidirectional images," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, pp. 3901–3907, April 2007.
- [18] T. Sattler, B. Lelbe, and L. Kobbelt, "Improving RANSAC's efficiency with a spatial consistency filter," in *Proceedings of the International Conference on Computer Vision*, pp. 2090–2097, 2009.
- [19] C. Papazov and D. Burschka, "An efficient RANSAC for 3D object recognition in noisy and occluded scenes," in *Computer Vision—ACCV 2010*, vol. 6492 of *Lecture Notes in Computer Science*, pp. 135–148, Springer, Berlin, Germany, 2011.
- [20] C. Silpa-Anan and R. Hartley, "Optimised KD-trees for fast image descriptor matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, Anchorage, Alaska, USA, June 2008.
- [21] N.-N. Ding, Y.-Y. Liu, Y. Zhang, C.-N. Chen, and B.-G. He, "Fast image registration based on SURF-DAISY algorithm and randomized kd trees," *Journal of Optoelectronics Laser*, vol. 23, no. 7, pp. 1395–1402, 2012.
- [22] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proceedings of the 4th International Conference on Computer Vision Theory and Applications (VISAPP '09)*, pp. 331–340, February 2009.
- [23] S. M. Liu, *The 3D Laser Point Cloud Shading and Texture Mapping of Surface Model*, Nanjing Normal University, 2009.