# Superresolution Mapping of Remotely Sensed Image Based on Hopfield Neural Network With Anisotropic Spatial Dependence Model

Xiaodong Li, Yun Du, Feng Ling, Qi Feng, and Bitao Fu

***Abstract*—Superresolution mapping (SRM) based on the Hopfield neural network (HNN) is a technique that produces land cover maps with a finer spatial resolution than the input land cover fraction images. In HNN-based SRM, it is assumed that the spatial dependence of land cover classes is homogeneous. HNN-based SRM uses an isotropic spatial dependence model and gives equal weights to neighboring subpixels in the neighborhood system. However, the spatial dependence directions of different land cover classes are discarded. In this letter, a revised HNN-based SRM with anisotropic spatial dependence model (HNNA) is proposed. The Sobel operator is applied to detect the gradient magnitude and direction of each fraction image at each coarse-resolution pixel. The gradient direction is used to determine the direction of subpixel spatial dependence. The gradient magnitude is used to determine the weights of neighboring subpixels in the neighborhood system. The HNNA was examined on synthetic images with artificial shapes, a synthetic IKONOS image, and a real Landsat multispectral image. Results showed that the HNNA can generate more accurate superresolution maps than a traditional HNN model.**

***Index Terms*—Anisotropic spatial dependence model, Hopfield neural network (HNN), sobel operator, superresolution mapping (SRM).**

## I. Introduction

SUPERRESOLUTION land cover mapping (SRM) (or subpixel land cover mapping) is a process used to predict the spatial distribution of land cover classes within mixed pixels at a finer spatial resolution than the input data [1]. SRM use fraction images, which are the result of soft classification of multi/hyperspectral images, as input, and forms a postprocessing step applied to the fraction images. The SRM models include a pixel-swapping algorithm [2]–[4], a Hopfield neural network (HNN) [5]–[9], linear optimization [10], a genetic algorithm [11], SRM with geostatistical solutions [12], particle-swarm-optimization-based SRM [13], and SRM with a directly mapping model [14].

Manuscript received March 20, 2013; revised July 20, 2013 and September 25, 2013; accepted November 15, 2013. Date of publication December 16, 2013; date of current version March 11, 2014. This work was supported in part by the Natural Science Foundation of China under Grant 41301398 and Grant 41001236 and in part by the Chinese Academy of Sciences through the Strategic Priority Research Program under Grant XDA05050107.

X. Li, Y. Du, F. Ling, and Q. Feng are with the Key laboratory of Monitoring and Estimate for Environment and Disaster of Hubei Province, Institute of Geodesy and Geophysics, Chinese Academy of Sciences, Wuhan 430077, China (e-mail: lingf@whigg.ac.cn).

B. Fu is with the School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China.

Digital Object Identifier 10.1109/LGRS.2013.2291778

The HNN has been widely used, and it demonstrates considerable potential in SRM. It seeks to maximize the spatial dependence of land cover patches and maintain the land cover class proportional information derived for remotely sensed pixels from soft classification in the resulting superresolution map. All existing HNN-based SRM methods use an isotropic spatial dependence model that assumes the spatial dependence of land cover classes is homogeneous and oversimplified if spatial dependence changes with direction. A $3 \times 3$-sized neighborhood system is used in the HNN, giving equal weighs to neighboring subpixels for a target subpixel. The directions of subpixel spatial dependence for different land cover classes are isotropic. Although the HNN has been demonstrated to provide accurate superresolution land cover representation, it has several limitations in predicting the interclass boundaries.

Various methods have been proposed to improve the HNN in reconstructing class boundaries. In addition to the information from land cover fraction images produced by soft classification, Nguyen *et al.* used ancillary data, including fused images [15] and lidar data [16], to provide supplementary information at the subpixel level in the HNN to improve the boundaries representation. Ling *et al.* [5] used multishifted remotely sensed images in the HNN to reduce the uncertainty of labeling the class boundaries. Su *et al.* [7] combined contouring methods and the HNN to enhance the SRM accuracy at class boundaries. However, the spatial dependence model in the HNN was not revised.

The main drawback of all these existing HNN methods is that only the isotropic spatial dependence model is used. An important way to improve the SRM accuracy at land cover boundaries is using anisotropic spatial dependence instead of the isotropic model. Thornton *et al.* [17] proposed a linear pixel-swapping algorithm with an anisotropic spatial dependence model to describe linear features in the SRM. Ling *et al.* [18] proposed an SRM model with anisotropic spatial dependence model to map urban buildings. Although the SRM accuracy levels were improved by using anisotropic spatial dependence model, the aforementioned methods can be only applied to specific land cover classes.

The HNN with isotropic spatial dependence model disregards the direction of spatial dependence of different land cover classes. In particular, if there is an edge in a coarse-resolution pixel, the direction of subpixel spatial dependence in this coarse-resolution pixel should be anisotropic and identical with the edge direction to preserve the shape of class boundaries. Aiming to incorporate the spatial dependence direction information of different land cover classes in SRM to preserve the shape of class boundaries, a revised HNN-based SRM with an

anisotropic spatial dependence model (HNNA) is constructed in this letter. The HNNA can be applied to land cover fraction images without using any ancillary data and is not limited to land cover classes with specific shapes but suitable for various objects with different shapes. The HNNA was examined on synthetic images with artificial shapes, a synthetic IKONOS image, and a real Landsat multispectral image.

## II. Methodology

The HNNA model presented in this letter is founded based on the structure of the HNN [8]. The HNN predicts the location of land cover classes according to the output of the soft classification. The $C$ class fraction images ($C$ is the number of land cover classes) are represented by $C$ interconnected layers, and the neurons within these layers are referred to by coordinate notation at the subpixel scale. For example, neuron $(h, i, j)$ refers to a neuron in row $i$ and column $j$ of the subpixel grid in the land cover class $h$ and has input signal $u_{hij}$ and output signal $v_{hij}$. The zoom factor $z$ determines the increase in spatial resolution from the original satellite image to the high-resolution (subpixel) land cover map. The relationship between input and output is defined as

$$v_{hij} = \frac{1}{2}\left[1 + \tanh(\lambda u_{hij})\right] \tag{1}$$

where $\lambda$ is a parameter that determines the steepness of the function. The HNN is a fully recurrent neural network and runs until it converges to a stable state. At the stable state, the value of the energy function of HNN is at a minimum. The output values of the neurons are binary values at the stable state. If the output value of the neuron is 1, the subpixel is assigned to that land cover class. Otherwise, if the output value is 0, the subpixel does not belong to that class [8].

The energy function of the HNN is expressed as

$$E = -\sum_{h}\sum_{i}\sum_{j}(K_1 G1_{hij} + K_2 G2_{hij} + K_3 P_{hij} + K_4 M_{hij}) \tag{2}$$

where $K_1$, $K_2$, $K_3$, and $K_4$ are weighting constants that define the effects of the corresponding goal functions $G1$ and $G2$, proportion constraint $P$, and multiclass constraint $M$ to the energy function, respectively.

The energy function of the HNNA is composed of the same functions and constraints as the HNN in (2). The difference between the HNNA and the HNN lays in the calculation of spatial dependence values in the goal functions. The goal functions of the HNN and HNNA consider the spatial dependence between observations. For each neuron $(h, i, j)$, $G1_{hij}$ and $G2_{hij}$ are the goal function values, which are determined as

$$\frac{dG1_{hij}}{dv_{hij}} = \frac{1}{2}\left(1 + \tanh(S_{hij} - 0.5)\lambda\right)(v_{hij} - 1) \tag{3}$$

$$\frac{dG2_{hij}}{dv_{hij}} = \frac{1}{2}\left(1 + (-\tanh(S_{hij} - 0.5)\lambda)\right)v_{hij} \tag{4}$$

where $S_{hij}$ is the spatial dependence value of neuron $(h, i, j)$ and is computed according to the neighborhood system.

In the HNN with an isotropic spatial dependence model, a $3 \times 3$-sized neighborhood systems where the eight neighboring neurons for the target neuron are given equal weights is used. $S_{hij}$ in the HNN is computed as

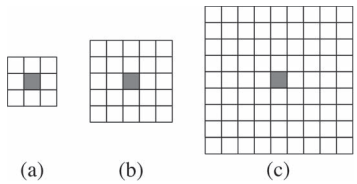


Fig. 1. (a) HNN neighborhood system. (b) and (c) HNNA neighborhood systems with $W = 5$ and $W = 9$. Gray indicates the target neuron, and white indicates neighboring neurons according to the target neuron.

$$S_{hij} = \frac{1}{8}\sum_{\substack{b=i-1\\ b\neq i}}^{i+1}\sum_{\substack{c=j-1\\ c\neq j}}^{j+1} v_{hbc} \tag{5}$$

where $v_{hbc}$ is the output of neighboring neuron $(h, b, c)$ in the neighborhood system.

In the HNNA, the neighboring neurons are usually given different weights. $S_{hij}$ in the HNNA is computed as

$$S_{hij} = \frac{1}{\Omega}\sum_{\substack{b=i-(W-1)/2\\ b\neq i}}^{i+(W-1)/2}\sum_{\substack{c=j-(W-1)/2\\ c\neq j}}^{j+(W-1)/2} v_{hbc}\cdot w_{hbc|hij} \tag{6}$$

where $W$ is the neighborhood-system window size, which is the length of the square side of the neighborhood (see Fig. 1). $w_{hbc|hij}$ is the weight of $v_{hbc}$ in the calculation of $S_{hij}$. $\Omega$ is the normalization constant chosen in order that $\sum_{b=i-(W-1)/2}^{i+(W-1)/2}\sum_{c=j-(W-1)/2}^{j+(W-1)/2} w_{hbc|hij} = 1$.

In the HNNA, the setting of $w_{hbc|hij}$ is based on the gradient magnitude and direction at pixel $(m, n)$ in $\mathbf{y}_h$, where $\mathbf{y}_h$ is the coarse-resolution fraction image of class $h$, and $(m, n)$ is the coarse-resolution pixel in $\mathbf{y}_h$, where $(h, i, j)$ is located in. In the HNNA, the direction of subpixel spatial dependence in the neighborhood system centered by $(h, i, j)$ is assumed to be orthogonal to the gradient direction, and the weights of $w_{hbc|hij}$ varies with gradient magnitude. The setting of $w_{hbc|hij}$ in the HNNA neighborhood system includes four steps.

*Step 1.* Construct a $W \times W$-sized neighborhood system, with $(h, i, j)$ as its central neuron.

*Step 2.* Calculate the gradient magnitude and direction of pixel $(m, n)$ in $\mathbf{y}_h$ using the Sobel operator. The Sobel operator calculates the gradient of the image intensity of $\mathbf{y}_h$ at each coarse-resolution pixel and the rate of change in that direction, by using two $3 \times 3$ kernels that are convolved with the original image to calculate the approximations of the derivatives for horizontal and vertical changes. The Sobel operator gives $G_{xhmn}$ and $G_{yhmn}$ values in the following to show the increase at pixel $(m, n)$ in intensity in $\mathbf{y}_h$ across the horizontal and vertical directions, respectively:

$$G_{xhmn} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * y_{hmn}$$

$$G_{yhmn} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * y_{hmn} \tag{7}$$

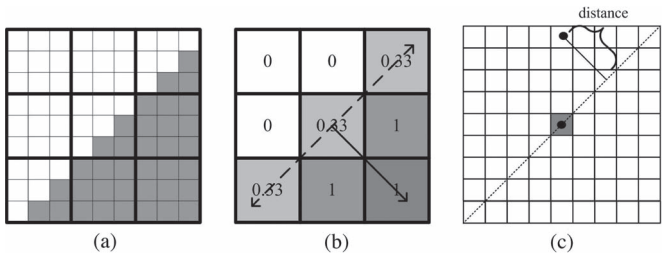


Fig. 2. (a) Subpixel locations in a 3 × 3 coarse-resolution pixel sized area ($z = 3$). Subpixels marked in gray indicate class $h$ and subpixels marked in white indicate other classes, except for class $h$. (b) 3 × 3 coarse-resolution pixel sized fraction image of $\mathbf{y}_h$. $(m, n)$ is the central coarse-resolution pixel in (b), and $(h, i, j)$ is one of the neurons in $(m, n)$. The solid line with an arrow indicates the gradient direction of pixel $(m, n)$ in $\mathbf{y}_h$, and the dashed line with arrows indicate the direction of edge in pixel $(m, n)$. (c) Construction of the neighborhood system centered by $(h, i, j)$ in the HNNA ($W = 9$). Neuron marked in gray indicates the neuron $(h, i, j)$, and neurons marked in white indicate the neighboring neurons. The dotted line indicates the axis of spatial dependence direction in the neighborhood system.

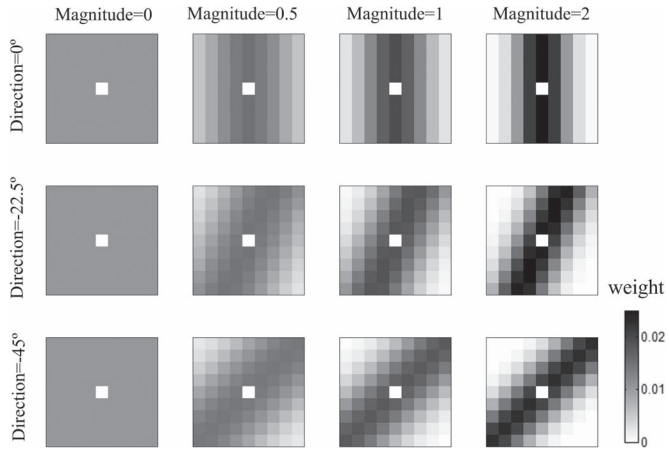


Fig. 3. Normalized weights of $w_{hbc|hij}$ in the neighborhood system of the HNNA with the variation of gradient direction and magnitude of pixel $(m, n)$ in $\mathbf{y}_h$. The window size $W = 9$ and the nonlinear parameter $\sigma = 2$.

where $y_{hmn}$ is the 3 × 3 pixel-sized subblock in $\mathbf{y}_h$ with pixel $(m, n)$ as its central pixel. The asterisk $(*)$ denotes the 2-D convolution operation. The gradient magnitude $G_{hmn}$ and gradient direction $\Theta_{hmn}$ at pixel $(m, n)$ in $\mathbf{y}_h$ are calculated according to the following:

$$G_{hmn} = \sqrt{G_{xhmn}^2 + G_{yhmn}^2} \tag{8}$$

$$\Theta_{hmn} = \text{atan}\left(\frac{G_{yhmn}}{G_{xhmn}}\right). \tag{9}$$

*Step 3.* Determine the direction of subpixel spatial dependence. The direction of subpixel spatial dependence in the HNNA neighborhood system centered by $(h, i, j)$ is assumed to be identical with the edge direction that passes through the pixel $(m, n)$, which is orthogonal to the gradient direction indicating the direction of the largest possible increase from low fraction to high fraction in $\mathbf{y}_h$ [see Fig. 2(a) and (b)]. The axis of spatial dependence direction in the neighborhood system centered by $(h, i, j)$, which passes across the center of the target neuron $(h, i, j)$, is then defined [see Fig. 2(c)].

*Step 4.* Set $w_{hbc|hij}$ in the HNNA neighborhood system according to an exponent decay function. The values of $w_{hbc|hij}$ are decreased with the distance between the center of the neighboring neuron and the axis line as

$$w_{hbc|hij} = e^{\left(-0.5 \times G_{hmn} \times d_{hbc}^2/\sigma^2\right)} \tag{10}$$

where $d_{hbc}$ is the Euclidian distance between neuron $(h, b, c)$ and the main axis line in the HNNA neighborhood system [see Fig. 2(c)]. $\sigma$ is a nonlinear parameter. In (10), the gradient magnitude $G_{hmn}$ is multiplied in the exponential part. Theoretically, it is more probable that there is an edge passing through pixel $(m, n)$ in $\mathbf{y}_h$ when $G_{hmn}$ is high; in this case, the weights of neighboring neurons vary sharply with $d_{hbc}$, and the class boundaries are expected to be constructed well due to the anisotropic spatial dependence. In contrast, there is probably a uniform intensity region in $\mathbf{y}_h$ when $G_{hmn} = 0$; in this case, an isotropic spatial dependence is assigned for class $h$. The figures of weights in the neighborhood with variations of different gradient orientations and magnitude levels are shown in Fig. 3.

The proportion constraint $P_{hij}$ of the HNNA is the same as the HNN and aims to retain the pixel class proportion output from soft classification. This was achieved by adding in the constraint that the total output from the set of neurons representing each coarse-resolution image pixel should be equal to the predicted class proportion for that pixel, i.e.,

$$\frac{dP_{hij}}{dv_{hij}} = \frac{1}{2z^2} \sum_{d=mz}^{mz+z-1} \sum_{e=nz}^{nz+z-1} (1 + \tanh(v_{hde} - 0.5)\lambda) - a_{hmn} \tag{11}$$

where $a_{hmn}$ is the proportion of class $h$ at pixel $(m, n)$.

The multiclass constraint value $M_{hij}$ of the HNNA is the same as HNN, aiming to make the sum of classes at the position $(i, j)$ to be equal to 1, i.e.,

$$\frac{dM_{hij}}{dv_{hij}} = \left(\sum_{k=0}^{C} v_{kij}\right) - 1. \tag{12}$$

## III. Experiments and Result Analysis

*1) Experiment 1:* Synthetic categorical images derived from three artificial shapes were applied as the reference map in this experiment. Three simulated artificial images, including shapes of "x", "annulus," and "triangle" with two classes representing the shape and the background were considered. Each image contains 120 × 120 pixels. The zoom factors $z$ was set as 6, 10, and 15, respectively. The reference map was degraded according to zoom factors $z$ to simulate the coarse-resolution fraction images, in order to avoid error introduced by soft classification. The corresponding fraction images were considered as the results generated by soft classification and were used as the inputs of the HNN and the HNNA. The performance of the HNNA was compared with the HNN and hard classification (HC), which assigns the coarse-resolution pixel to the class that occupies the majority of the area in that pixel. The weighting constants $K_1$, $K_2$, $K_3$, and $K_4$ in the HNN and HNNA were all set to 1. The steepness parameter $\lambda$ in the HNN and HNNA was set as 100. These parameters were set empirically because there is no single rule to determine the optimal set of parameters [5]–[9], [15], [16]. The neighborhood window size $W$ in the HNNA was set as 3, 5, 7, and 9, respectively, to show the influence of $W$ on the classification map. The nonlinear parameter $\sigma$ in the HNNA was set as 2 through many trials ($\sigma$ ranging from 1 to 3 can generate similar results). The accuracy values

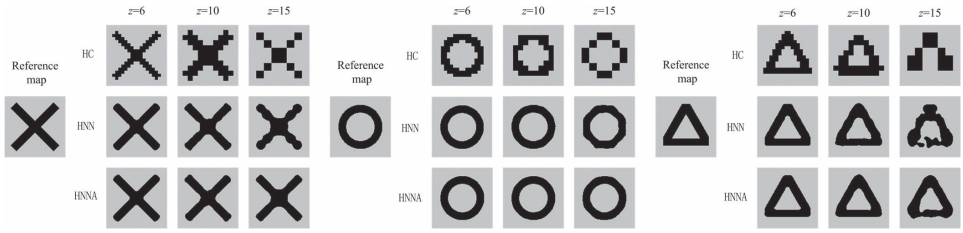


Fig. 4. Classification maps produced by the HC, HNN, and HNNA with the highest kappa value at different zoom factors for synthetic images.

TABLE I
ACCURACY OF KAPPA $K$ AND F-MEASURE $F$ FOR HC, HNN, AND HNNA FOR SYNTHETIC IMAGES

| | $z$ | | HC | HNN | HNNA | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | $W$=3 | $W$=5 | $W$=7 | $W$=9 |
| x | 6 | $K$ | 0.8361 | 0.9843 | 0.9850 | 0.9845 | 0.9863 | 0.9852 |
| | | $F$ | 0.8736 | 0.9883 | 0.9888 | 0.9884 | 0.9898 | 0.9890 |
| | 10 | $K$ | 0.7830 | 0.9626 | 0.9673 | 0.9682 | 0.9647 | 0.9659 |
| | | $F$ | 0.8452 | 0.9721 | 0.9755 | 0.9762 | 0.9736 | 0.9745 |
| | 15 | $K$ | 0.7317 | 0.9108 | 0.9183 | 0.9512 | 0.9453 | 0.9444 |
| | | $F$ | 0.7894 | 0.9333 | 0.9389 | 0.9635 | 0.9590 | 0.9583 |
| annulus | 6 | $K$ | 0.8787 | 0.9934 | 0.9955 | 0.9944 | 0.9905 | 0.9867 |
| | | $F$ | 0.9076 | 0.9910 | 0.9966 | 0.9957 | 0.9928 | 0.9908 |
| | 10 | $K$ | 0.7706 | 0.9851 | 0.9840 | 0.9880 | 0.9859 | 0.9820 |
| | | $F$ | 0.8248 | 0.9877 | 0.9878 | 0.9909 | 0.9893 | 0.9883 |
| | 15 | $K$ | 0.6519 | 0.9415 | 0.9285 | 0.9590 | 0.9678 | 0.9696 |
| | | $F$ | 0.7247 | 0.9554 | 0.9377 | 0.9687 | 0.9755 | 0.9768 |
| triangle | 6 | $K$ | 0.8797 | 0.9904 | 0.9932 | 0.9904 | 0.9895 | 0.9900 |
| | | $F$ | 0.9098 | 0.9919 | 0.9949 | 0.9929 | 0.9922 | 0.9926 |
| | 10 | $K$ | 0.7738 | 0.9592 | 0.9642 | 0.9713 | 0.9734 | 0.9712 |
| | | $F$ | 0.8330 | 0.9696 | 0.9733 | 0.9786 | 0.9802 | 0.9785 |
| | 15 | $K$ | 0.6384 | 0.8304 | 0.8152 | 0.9049 | 0.9301 | 0.9367 |
| | | $F$ | 0.7233 | 0.8734 | 0.8619 | 0.9290 | 0.9478 | 0.9527 |

were assessed using a kappa coefficient, which is used to assess the map-level accuracy and $F$-measure that is used to assess categorical-level accuracy [19]. $F$-measure was used to assess the accuracy of the three objects (which are "x", "annulus," and "triangle" that are marked in black in Fig. 4). Both kappa and $F$-measure values were measured at the spatial resolution that was the same as the fine-resolution pixel (subpixel). All models were tested on an Intel Core 2 Processor, with 2.66-GHz Duo CPU and 1.98-GB RAM using Matlab version 7.3. The running time was 14 min for the HNN and 21 min for the HNNA (shape of "x", $z = 6$; $W = 7$ for the HNNA).

Table I gives the accuracy values of HC, HNN, and HNNA. First, the performance of the HNNA was affected by the neighborhood window size $W$. In most cases, the kappa and $F$-measure values for the HNNA were high when $W$ ranged from 5 to 9. When $W = 3$, the neighborhood system was too small, and the HNNA generated similar classification maps as those produced by the HNN. Second, the zoom factor played an important role in the accuracy values. The kappa and $F$-measure values for different methods decreased when the zoom factor increased. This is because, when the zoom factor increases, the number of unknown variables in the SRM increases, and the SRM problem is strongly underdetermined [20]. Finally, the kappa and $F$-measure values for HC were lower than those for the HNN and HNNA at different zoom factors. This is because HC generated classification maps at the pixel scale, whereas the HNN and HNNA generated classification maps at the subpixel scale. In addition, the highest kappa and $F$-measure values for the HNNA were not lower than those for the HNN, particularly when $z = 15$. Therefore, the improvement of accuracy values for the HNNA was more obvious when the spatial resolution of remotely sensed image pixels was coarsen.

Fig. 4 shows the classification maps produced by HC, HNN, and HNNA with the highest kappa value at different zoom factors. First, HC generated classification maps with jagged and rough boundaries at different zoom factors because each mixed pixel was assigned to one land cover class. By contrast, the classification maps generated by the HNN and the HNNA matched better with the reference map. Second, the difference between classification maps produced by the HNN and the HNNA varied with the zoom factor. When $z = 6$, both HNN and HNNA generated classification maps that were similar with the reference map. The classification map became inaccurate when $z$ increased. The differences between classification maps produces by the HNN and HNNA were more obvious when $z = 15$. In classification maps produced by the HNN, the linear boundaries were jagged and not straight in image "x," the boundaries were not smoothed in image "annulus," and the linear patch in the bottom part of "triangle" was separated and unsmoothed. By contrast, in classification maps produced by the HNNA, the linear boundaries were straight in image "x," and the boundaries were smoothed in image "annulus," and the linear patch in the bottom part of "triangle" was connected.

*2) Experiment 2:* A subset of the IKONOS image taken over at Dujiangyan, Sichuan Province, China, was adopted to test the proposed model on multiclass scenarios. A part of the panchromatic image with 240 × 240 pixels was selected and manually digitized as the reference map with five land cover classes of different farmlands in it. The reference map was degraded according to zoom factor $(z = 10)$ to simulate the coarse-resolution land cover fraction images. In the HNN and HNNA, the weighting constants $K_1$, $K_2$, $K_3$, and $K_4$ were all set to 1, and the steepness parameter $\lambda$ was set as 100. The parameters in the HNNA were as follows: $W = 7$ and $\sigma = 2$. The running time was 158 min for the HNN and 252 min for the HNNA.

Fig. 5 shows the classification maps generated by different methods using the IKONOS image. The class boundaries were serrated in the classification map produced by HC and not smoothed in the classification map produced by the HNN. By contrast, the classification map produced by the HNNA matched better with the reference map, and class boundaries were straighter and smoother. Quantitative analysis shows that the kappa value for HC was 0.7552. The kappa value increased from 0.9350 for the HNN to 0.9451 for the HNNA.

*3) Experiment 3:* A subset of Landsat ETM+ image acquired on July 18, 2005 taken over at the Brazilian Amazon Basin was adopted to test the proposed model on real multispectral remotely sensed image. The subset of Landsat ETM+ image

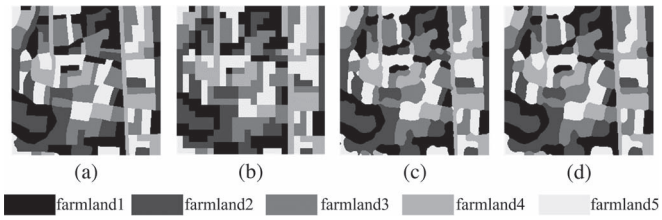


Fig. 5. (a) IKONOS reference map of different farmlands and the classification maps produced by (b) HC, (c) HNN, and (d) HNNA.

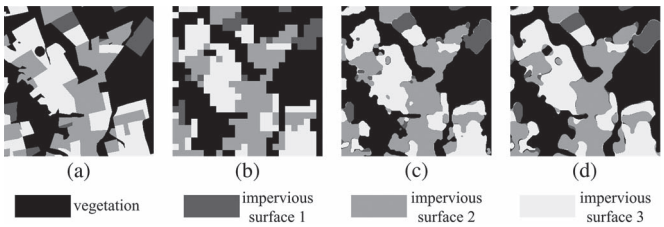


Fig. 6. (a) Landsat reference map of vegetation and different impervious surfaces and the classification maps produced by (b) HC, (c) HNN, and (d) HNNA.

used for analysis includes the bands 1, 2, 3, 4, 5, and 7 and $360 \times 360$ pixels. The coarse-resolution multispectral image was produced by averaging the digital number values of the original Landsat ETM+ image according to the zoom factor $(z = 15)$. The original Landsat ETM+ image was manually digitized as the reference map with four land cover classes of vegetation and different impervious surfaces in it [see Fig. 6(a)]. A maximum-likelihood classifier was used directly on the coarse-resolution multispectral image to generate the HC map. The land cover fraction images as the HNN and HNNA input were produced by spectral unmixing using the multiple endmember spectral mixture analysis model. The average root-mean-square error of the generated fraction images [21] was 0.1976. The model parameters for the HNN and the HNNA were set the same as in experiment 2. The running time was 312 min for the HNN and 525 min for the HNNA.

Fig. 6 shows the classification maps generated by different methods using the Landsat image. Similar to the results of the IKONOS image, the classification map generated by HC had jagged boundaries, whereas the classification maps generated by the HNN and HNNA had smoother boundaries. There were less granularity effects and less serrated boundaries in the HNNA result than in the HNN result. The kappa value for the HNNA was 0.8466, higher than that of 0.7722 for HC and 0.8362 for the HNN.

## IV. Conclusion

In this letter, a HNN model with anisotropic spatial dependence model has been proposed to preserve class boundaries. Experiments showed that the classification maps produced by the HNNA matched better with the reference map compared with those produced by the HNN. The advantage of the HNNA was more obvious when the zoom factor was large. The kappa value was improved by the HNNA. The HNNA is sensitive to the neighborhood window size. In particular, for the images analyzed, the window size should be set larger than 3 because the HNNA produces classification maps that are similar with those produced by the HNN when the window size is small. Automatic estimation of the optimal values of the model parameters in the HNNA should be studied in the future.

## References

[1] P. M. Atkinson, "Issues of uncertainty in super-resolution mapping and their implications for the design of an inter-comparison study," *Int. J. Remote Sens.*, vol. 30, no. 20, pp. 5293–5308, Sep. 2009.

[2] P. M. Atkinson, "Sub-pixel target mapping from soft-classified, remotely sensed imagery," *Photogramm. Eng. Remote Sens.*, vol. 71, no. 7, pp. 839–846, Jul. 2005.

[3] X. Li, Y. Du, F. Ling, S. Wu , and Q. Feng, "Using a sub-pixel mapping model to improve the accuracy of landscape pattern indices," *Ecolog. Indicators*, vol. 11, no. 5, pp. 1160–1170, Sep. 2011.

[4] F. Ling, W. B. Li, Y. Du, and X. Li, "Land cover change mapping at the subpixel scale with different spatial-resolution remotely sensed imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 1, pp. 182–186, Jan. 2011.

[5] F. Ling, Y. Du, F. Xiao, H. Xue, and S. Wu, "Super-resolution land-cover mapping using multiple sub-pixel shifted remotely sensed images," *Int. J. Remote Sens.*, vol. 31, no. 19, pp. 5023–5040, Jun. 2010.

[6] A. M. Muad and G. M. Foody, "Impact of land cover patch size on the accuracy of patch area representation in HNN-based super resolution mapping," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 5, pp. 1418–1427, Oct. 2012.

[7] Y. F. Su, G. M. Foody, A. M. Muad, and K.-S. Cheng, "Combining Hopfield neural network and contouring methods to enhance super-resolution mapping," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 5, pp. 1403–1417, Oct. 2012.

[8] A. J. Tatem, H. G. Lewis, P. M. Atkinson, and M. S. Nixon, "Multiple-class land-cover mapping at the sub-pixel scale using a Hopfield neural network," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 3, no. 2, pp. 184–190, 2001.

[9] A. J. Tatem, H. G. Lewis, P. M. Atkinson, and M. S. Nixon, "Super-resolution target identification from remotely sensed images using a Hopfield neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 4, pp. 781–796, Apr. 2001.

[10] J. Verhoeye and R. De Wulf, "Land cover mapping at sub-pixel scales using linear optimization techniques," *Remote Sens. Environ.*, vol. 79, no. 1, pp. 96–104, Jan. 2002.

[11] K. Mertens, L. Verbeke, E. Ducheyne, and R. R. De Wulf, "Using genetic algorithms in sub-pixel mapping," *Int. J. Remote Sens.*, vol. 24, no. 21, pp. 4241–4247, Jan. 2003.

[12] A. Boucher and P. C. Kyriakidis, "Super-resolution land cover mapping with indicator geostatistics," *Remote Sens. Environ.*, vol. 104, no. 3, pp. 264–282, Oct. 2006.

[13] Q. M. Wang, L. G. Wang, and D. F. Liu, "Particle swarm optimization-based sub-pixel mapping for remote-sensing imagery," *Int. J. Remote Sens.*, vol. 33, no. 20, pp. 6480–6496, Oct. 2012.

[14] Y. Ge, S. Li, and V. C. Lakhan, "Development and testing of a subpixel mapping algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 2155–2164, Jul. 2009.

[15] M. Q. Nguyen, P. M. Atkinson, and H. G. Lewis, "Superresolution mapping using a hopfield neural network with fused images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 3, pp. 736–749, Mar. 2006.

[16] M. Q. Nguyen, P. M. Atkinson, and H. G. Lewis, "Superresolution mapping using a Hopfield neural network with LIDAR data," *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 3, pp. 366–370, Jul. 2005.

[17] M. W. Thornton, P. M. Atkinson, and D. A. Holland, "A linearised pixel-swapping method for mapping rural linear land cover features from fine spatial resolution remotely sensed imagery," *Comput. Geosci.*, vol. 33, no. 10, pp. 1261–1272, Oct. 2007.

[18] F. Ling, X. Li, F. Xiao, S. Fang, and Y. Du, "Object-based sub-pixel mapping of buildings incorporating the prior shape information from remotely sensed imagery," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 18, pp. 283–292, Aug. 2012.

[19] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York, NY, USA: ACM Press, 1999.

[20] V. A. Tolpekin and A. Stein, "Quantification of the effects of land-cover-class spectral separability on the accuracy of Markov-random-field-based superresolution Mapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 9, pp. 3283–3297, Sep. 2009.

[21] J. Jin, B. Wang, and L. Zhang, "A novel approach based on fisher discriminant null space for decomposition of mixed pixels in hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 699–703, Oct. 2010.