

Article

Remote-Sensing Image Classification Based on an Improved Probabilistic Neural Network

Yudong Zhang ^{1,*}, Lenan Wu ¹, Nabil Neggaz ², Shuihua Wang ¹ and Geng Wei ¹

¹ School of Information Science and Engineering, Southeast University, Nanjing 210009, China; E-Mails: shuihuaw2007@gmail.com (S.W.); wei_geng@163.com (G.W.)

² Signal-Image-Parole Laboratory, Department of Computer Science, University of Science and Technology – Oran, Oran, Algeria; E-Mail: neggaz_nabil@yahoo.fr

* Author to whom correspondence should be addressed; E-Mail: zhangyudongnuaa@gmail.com.

Received: 12 June 2009; in revised form: 2 September 2009 / Accepted: 16 September 2009 /

Published: 23 September 2009

Abstract: This paper proposes a hybrid classifier for polarimetric SAR images. The feature sets consist of span image, the H/A/ α decomposition, and the GLCM-based texture features. Then, a probabilistic neural network (PNN) was adopted for classification, and a novel algorithm proposed to enhance its performance. Principle component analysis (PCA) was chosen to reduce feature dimensions, random division to reduce the number of neurons, and Brent's search (BS) to find the optimal bias values. The results on San Francisco and Flevoland sites are compared to that using a 3-layer BPNN to demonstrate the validity of our algorithm in terms of confusion matrix and overall accuracy. In addition, the importance of each improvement of the algorithm was proven.

Keywords: polarimetric SAR; Probabilistic neural network; gray-level co-occurrence matrix; principle component analysis; Brent's Search

1. Introduction

The classification of different objects, as well as different terrain characteristics, with single channel monopolarisation SAR images can carry a significant amount of error, even when operating after multilooking [1]. One of the most challenging applications of polarimetry in remote sensing is landcover classification using fully polarimetric SAR (PolSAR) images.

The Wishart maximum likelihood (WML) method has often been used for PolSAR classification [2]. This method uses the amplitudes of the elements in the covariance or coherency matrices. However, it does not explicitly take into consideration the phase information within polarimetric data, which plays a direct role in the characterization of a broad range of scattering processes. Furthermore, the covariance or coherency matrices are determined after spatial averaging and therefore can describe only stochastic scattering processes, while certain objects, such as man-made objects, are better characterized at a pixel-level [3].

To overcome above shortcomings, polarimetric decompositions were introduced with an aim to establish a correspondence between the physical characteristics of the considered areas and the observed scattering mechanisms. There are seven famous decomposition methods: Pauli [4], Krogager [5], Freeman [6], Huynen [7], Barnes [8], Cloude [9] and Holm [8]. The most effective method among these is the Cloude decomposition, also known as the H/A/ α method.

Recently, texture information has been extracted and used as a parameter to enhance the classification results. The texture parameters can be defined as many types, such as entropy [10], fractal dimension [11], lacunarity [12], wavelet energy [13], semivariograms [14], and gray-level co-occurrence matrix [15]. Particularly, the gray-level co-occurrence matrices (GLCM) were already successfully applied to classification problems.

Thus, we chose the combination of H/A/ α and GLCM as the parameter set of our method. The next problem is how to choose the best classifier. In the past, standard multi-layered feed-forward NNs with a back propagation (BP) algorithm have been applied for SAR image classification [16]. BPs are effective methods since they do not involve complex models and equations as compared to traditional regression analysis. In addition, they can easily adapt to new data through a re-training process.

However, BP needs much effort to determine the architecture of networks and more computations for training. Moreover, BP yields deterministic but not probabilistic results. This makes it technically impractical in classifications. Probabilistic neural networks (PNNs), therefore, are effective alternatives that are faster in determining the network architecture and in training. Moreover, PNNs provide probabilistic viewpoints and deterministic classification results [17].

The input weights and layer weights of PNN can be set directly from the available data, while the bias traditionally is difficult to determine, so it is usually obtained manually either by iterative experiments or by an exhaustive algorithm [18]. In this paper we propose a novel weights/biases setting method. Available input/target pairs are divided into training and validation subsets to reduce the number of neurons, and Brent's method [19] is adopted to find the optimal biases values since the problem can be regarded as a 1-D interval location problem. In addition, Principal Component Analysis (PCA) is employed [20] in order to reduce the feature dimensions and computation time.

The structure of this paper is as follows: In the next section, we introduce the concept of Pauli decomposition. Section 3 presents the feature set, namely, the span image, the H/A/ α decomposition, and the feature derived from GLCM. In section 4, the mechanism, structure and shortcomings of PNNs are introduced. Section 5 proposes our method and expatiates on the three important improvements: PCA, random division and optimization by Brent's Search. Section 6 applied our method to terrain classification on San Francisco site, and find that our method performs better than 3-layer BPNN

method. Section 7 applied our method to crop classification on Flevoland site. Section 8 discusses the significances of combined feature sets, random division, and PCA. Finally, Section 9 concludes this paper.

2. Pauli Decomposition

2.1. Basic Introduction

The features are derived from the multilook coherence matrix of the polarimetric SAR data. Suppose S stands for the measured scattering matrix:

$$S = \begin{bmatrix} S_{hh} & S_{hv} \\ S_{vh} & S_{vv} \end{bmatrix} = \begin{bmatrix} S_{hh} & S_{hv} \\ S_{hv} & S_{vv} \end{bmatrix} \quad (1)$$

where S_{qp} represents the scattering coefficients of the targets, p the polarization of the incident field, q the polarization of the scattered field. S_{hv} equals to S_{vh} since reciprocity applies in a monostatic system configuration.

The Pauli decomposition expresses the scattering matrix S in the so-called Pauli basis, which is given by the following three 2×2 matrices:

$$S_a = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, S_b = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, S_c = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (2)$$

Thus, S can be expressed as:

$$S = aS_a + bS_b + cS_c \quad (3)$$

where:

$$a = \frac{S_{hh} + S_{vv}}{\sqrt{2}}, b = \frac{S_{hh} - S_{vv}}{\sqrt{2}}, c = \sqrt{2}S_{hv} \quad (4)$$

An RGB image could be formed with the intensities $|a|^2$, $|b|^2$, $|c|^2$. The meanings of S_a , S_b , and S_c are listed in Table 1.

Table 1. Pauli bases and their corresponding meanings.

Pauli Bases	Meaning
S_a	Single- or odd-bounce scattering
S_b	Double- or even-bounce scattering
S_c	Those scatterers which are able to return the orthogonal polarization to the one of the incident wave (forest canopy)

2.2. Coherence Matrix

The coherence matrix is obtained as:

$$T = [a, b, c][a, b, c]^T = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{12}^* & T_{22} & T_{23} \\ T_{13}^* & T_{23}^* & T_{33} \end{bmatrix} \quad (5)$$

The average of multiple single-look coherence matrices is the multi-look coherence matrix. (T_{11}, T_{22}, T_{33}) usually are regarded the channels of the polarimetric SAR images.

3. Feature Extraction

The proposed features can be divided into three types, which are explained below.

3.1. Span

The span or total scattered power indicates the received power by a fully polarimetric system and is given by:

$$M = |S_{hh}|^2 + |S_{vv}|^2 + 2|S_{hv}|^2 \quad (6)$$

3.2. H/A/Alpha Decomposition

Cloude and Potter [9] proposed an algorithm to identify in an unsupervised way polarimetric scattering mechanisms in the H - α plane. The method extends the two assumptions of traditional ways: 1) azimuthally symmetric targets; 2) equal minor eigenvalues λ_2 and λ_3 .

T can be rewritten as:

$$T = U_3 \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} U_3^H \quad (7)$$

$$U_3 = \begin{bmatrix} \cos \alpha_1 & \cos \alpha_2 & \cos \alpha_3 \\ \sin \alpha_1 \cos \beta_1 \exp(i\delta_1) & \sin \alpha_2 \cos \beta_2 \exp(i\delta_2) & \sin \alpha_3 \cos \beta_3 \exp(i\delta_3) \\ \sin \alpha_1 \sin \beta_1 \exp(i\gamma_1) & \sin \alpha_2 \sin \beta_2 \exp(i\gamma_2) & \sin \alpha_3 \sin \beta_3 \exp(i\gamma_3) \end{bmatrix} \quad (8)$$

Then, the pseudo-probabilities of the T matrix expansion elements are defined as:

$$P_i = \frac{\lambda_j}{\sum_{j=1}^3 \lambda_j} \quad (9)$$

The entropy indicates the degree of statistical disorder of the scattering phenomenon. It can be defined as:

$$H = \sum_{i=1}^3 -P_i \log_3 P_i \quad 0 \leq H \leq 1 \quad (10)$$

For high entropy values, a complementary parameter (anisotropy) is necessary to fully characterize the set of probabilities. The anisotropy is defined as the relative importance of the second scattering mechanisms [21]:

$$A = \frac{P_2 - P_3}{P_2 + P_3} \quad 0 \leq A \leq 1 \quad (11)$$

The four estimates of the angles are easily evaluated as:

$$[\bar{\alpha}, \bar{\beta}, \bar{\delta}, \bar{\gamma}] = \sum_{i=1}^3 P_i [\alpha, \beta, \delta, \gamma] \quad (12)$$

Thus, vectors from coherence matrix can be represented as $(H, A, \bar{\alpha}, \bar{\beta}, \bar{\delta}, \bar{\gamma})$.

3.3. Texture Features

The Gray level co-occurrence matrix (GLCM) is a text descriptor which takes into account the specific position of a pixel relative to another. The GLCM is a matrix whose elements correspond to the relative frequency of occurrence of pairs of gray level values of pixels separated by a certain distance in a given direction [22]. Formally, the elements of a GLCM $G(i,j)$ for a displacement vector (a,b) is defined as

$$G(i, j) = |\{(x, y), (t, v) : I(x, y) = i \ \& \ I(t, v) = j\}| \quad (13)$$

Where $(t,v) = (x+a, y+b)$, and $|\cdot|$ is the cardinality of a set. The displacement vector (a,b) can be rewritten as (d, θ) in polar coordinates.

GLCMs are suggested to calculate from four displacement vectors with $d = 1$ and $\theta = 0^\circ, 45^\circ, 90^\circ$, and 135° respectively. In this study, the (a,b) are chosen as $(0,1)$, $(-1,1)$, $(-1,0)$, and $(-1,-1)$ respectively, and the corresponding GLCMs are averaged.

The four features are extracted from normalized GLCMs, the sum of which is equal to 1. Suppose the normalized GLCM value at (i,j) is $p(i,j)$, and their detailed definition are listed in Table 2.

Table 2. Properties of GLCM.

Property	Description	Formula
Contrast	Intensity contrast between a pixel and its neighbor	$\sum_{i,j} i - j ^2 p(i, j)$
Correlation	Correlation between a pixel and its neighbor (μ denotes the expected value, and σ the standard variance)	$\sum_{i,j} \frac{(i - \mu_i)(j - \mu_j) p(i, j)}{\sigma_i \sigma_j}$
Energy	Energy of the whole image	$\sum_{i,j} p(i, j)^2$
Homogeneity	Closeness of the distribution of GLCM to the diagonal	$\sum_{i,j} \frac{p(i, j)}{1 + i - j }$

3.4. Total Features

The texture features consist of 4 GLCM-based features, which should be multiplied by 3 since there exist three channels (T_{11}, T_{22}, T_{33}). In addition, there are one span feature, and six H/α parameters. In all, the total features are $1 + 6 + 4 \times 3 = 19$.

4. Probabilistic NN

4.1. Mechanism of PNN

Neural networks are widely used in pattern classification since they do not need any information about the probability distribution and the *a priori* probabilities of different classes. PNNs are basically pattern classifiers. They combine the well known Bayes decision strategy with the Parzen non-parametric estimator of the probability density functions (PDF) of different classes. PNNs have been of interest because they yield a probabilistic output and are easy to implement.

Taking a two categories situation as an example, we should decide the known state of nature θ to be either θ_A or θ_B . Suppose a set of measurements is obtained as p -dimensional vector $x = [x_1, \dots, x_p]$, the Bayes decision rule becomes:

$$d(x) = \begin{cases} \theta_A & \text{if } h_A l_A f_A(x) > h_B l_B f_B(x) \\ \theta_B & \text{if } h_A l_A f_A(x) < h_B l_B f_B(x) \end{cases} \quad (14)$$

Here, $f_A(x)$ and $f_B(x)$ are the PDF for categories A and B, respectively. l_A is the loss function associated with the wrong decision $d(x) = \theta_B$ when $\theta = \theta_A$, l_B is the loss function associated with the wrong decision $d(x) = \theta_A$ when $\theta = \theta_B$, and the losses associated with correct decisions are taken to be zero. h_A and h_B are the *a priori* probability of occurrence of patterns from category A and B, respectively.

In a simple case that assumes the loss function and *a priori* probability are equal, the Bayes rule classifies an input pattern to the class with higher PDF. Therefore, the accuracy of the decision boundaries depends on what the underlying PDFs are estimated. Parzen's results can be extended to estimate in the special case where the multivariate kernel is a product of univariate kernels. In the particular case of the Gaussian kernel, the multivariate estimates can be expressed as:

$$f_A(x) = \frac{1}{(2\pi)^{p/2} \sigma^p} \frac{1}{m} \sum_{i=1}^m \exp \left[-\frac{(x - x_{Ai})^T (x - x_{Ai})}{2\sigma^2} \right] \quad (15)$$

Here, m is the number of training vectors in category A, p is the dimensionality of the training vectors, x_{Ai} is the i th training vector for category A, and σ is the smoothing parameter. It should be noted that $f_A(x)$ is the sum of small multivariate Gaussian distributions centered at each training sample, but the sum is not limited to being Gaussian.

4.2. PNN Structure

Figure 1 shows the outline of PNN. When an input is presented, the first layer computes distances from the input vector to the input weights (IW), and produces a vector whose elements indicate how close the input is to the IW. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities. Finally, a *compet* transfer function on the output of the second layer picks the maximum of these probabilities, and produces a 1 for that class and a 0 for other classes.

The mathematical expression of PNN can be expressed as:

$$a = \text{radbas}(\|IW - x\| \cdot b) \quad (16)$$

$$y = \text{compet}(LW \cdot a) \quad (17)$$

In this paper, the *radbas* is selected as:

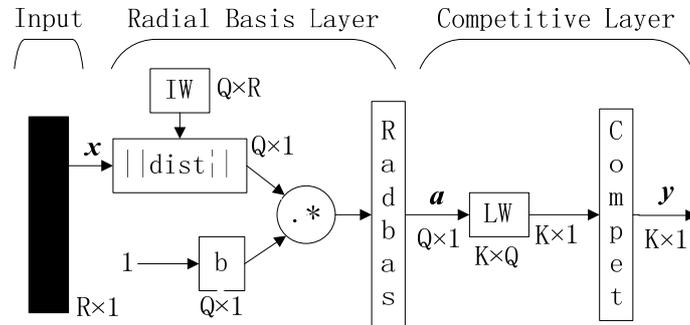
$$\text{radbas}(n) = \exp(-n^2) \quad (18)$$

The *compet* function is defined as:

$$\text{compet}(n) = e_i = [0 \ 0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0], \quad n(i) = \max(n) \quad (19)$$

This type of setting can produce a network with zero errors on training vectors, and obviously it does not need any training.

Figure 1. Outline of PNN (R, Q, and K represent number of elements in input vector, input/target pairs, and classes of input data, respectively. IW and LW represent input weight and layer weight, respectively).



4.3. Shortcomings of Traditional PNN

Suppose P and T denote the set of training vector x and corresponding target vector y , namely, $P = [x_1, x_2, \dots, x_Q]$, and $T = [y_1, y_2, y_Q]$. IW and LW are set traditionally as follows:

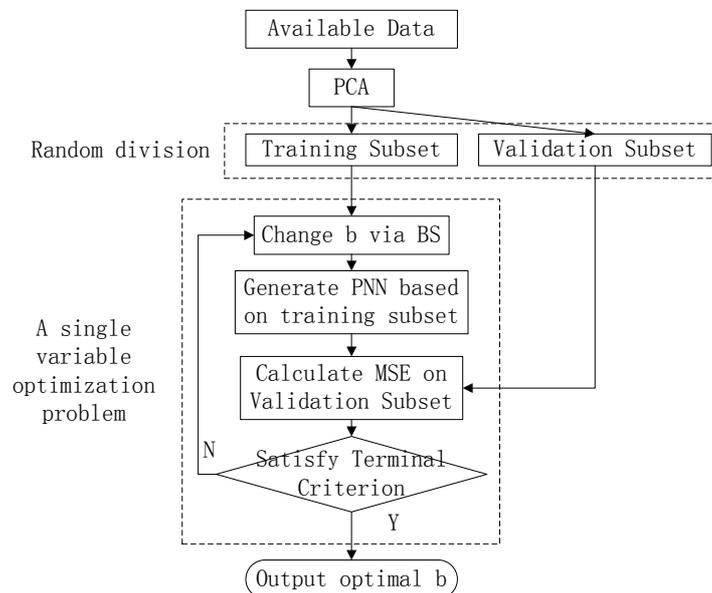
$$IW = P \tag{20}$$

$$LW = T \tag{21}$$

However, it is obvious that Q is usually very large, and then the net will be too big and consume too much computation time. On the other hand, to simplify the setting of bias b , all of its components are considered as equal [23]. Even so, the setting of b is still a challenge. Although errors on training vectors are always zero, the errors on test vectors are greatly dependent with the value of b .

If it is too small, the spread of each radial basis layer function becomes too large, and the network will take too many nearby design vectors into account, moreover, the radial basis neurons will output large values (near 1) for all the inputs used to design the network. If it is too larger, the spread becomes near zero, and the network will degrade as a nearest neighbor classifier.

Figure 2. The outline of our method.



5. A Novel Method of Weights/Biases Setting

Here we propose a novel method to solve the above two problems. The main idea is shown in Figure 2. Our improvement lies in the PCA, the random division, and the single variable optimization.

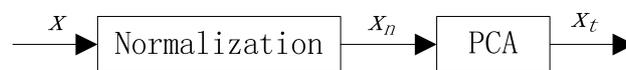
5.1. Feature Reduction

Excessive features increase computation times and storage memory. Furthermore, they sometimes make classification more complicated, which is called the curse of dimensionality. It is necessary to reduce the number of features.

Principal component analysis (PCA) is an efficient tool to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining most of the variations. It is achieved by transforming the data set to a new set of ordered variables. This technique has three effects: it orthogonalizes the components of the input vectors so that uncorrelated with each other, it orders the resulting orthogonal components so that those with the largest variation come first, and eliminates those components contributing the least to the variation in the data set.

It should be noted that the input vectors should be normalized to have zero mean and unity variance before performing PCA, which is shown in Figure 3.

Figure 3. Using normalization before PCA.



The normalization is a standard procedure. Details about PCA can be found in Ref. [24].

5.2. Random Division

Realistic sample numbers Q are generally very large, which leads to a quite large PNN. Thus, we divide the available data into two subsets: training subset and validation subset. The ratio of each is called *trainRatio* and *validRatio* respectively. In order to save the storage room of the net and to fasten the computation, the *trainRatio* is set as small as possible, and meanwhile it should not affect the accuracy of the NN.

5.3. Optimization by Brent's Search

The goal of finding optimal b can be obtained by solving this problem: find the minimum MSE on validation subset of the corresponding b . This can be depicted as a single-variable optimization problem in the dash-line rectangle in Figure 2. Brent's Search (BS) method is adopted to solve this optimization problem.

BS is a linear search, a hybrid of the golden section search and a quadratic interpolation. Golden section search has a first-order rate of convergence, while polynomial interpolations have an asymptotic rate faster than super-linear. On the other hand, the rate of convergence for the golden section search starts when the algorithm is initialized, whereas the asymptotic behavior for the

polynomial interpolations can take many iterations to become apparent. BS attempts to combine the best features of both approaches. BS has the advantage that it does not require computation of the derivative, which greatly fits the optimization problem.

6. Terrain Classification

The NASA/JPL AirSAR L-band data for the San Francisco (California, USA) area was used for the experiments. Its size is $1,024 \times 900$. In order to reduce the computations, the sub-area with size 600×600 was extracted from the left-upper point of original image. The ground truth of the test site can be found at Ref. [2].

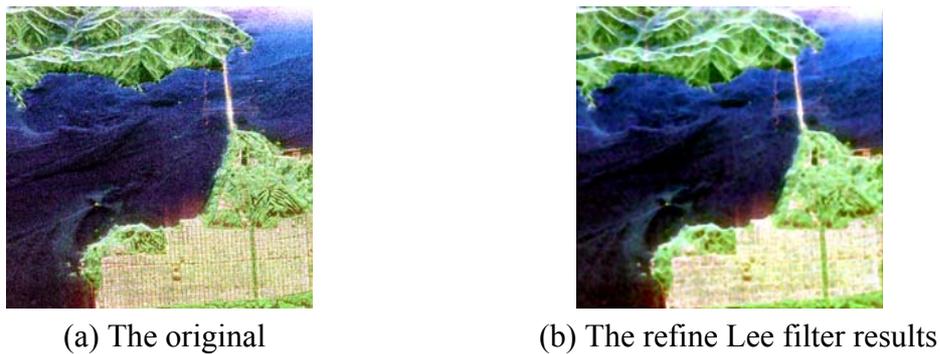
Quantitative information about the experiment is described as follows, where ‘●’ denotes parameters known before simulation and ‘◆’ denotes the parameters obtained at the initial stage of the experiment.

- Number of features: 19
- ◆ Number of reduced features by PCA: 11 (obtained by performing PCA on total available pairs)
- Location of Sub San Francisco Area:
 - X-range: 1–600
 - Y-range: 1–600
- Location of Training/Test Rectangular Area (the first and second pixels denote the coordinate of the left-upper point of the rectangle, the third and fourth pixels denote the width and length of the rectangle)
 - Sea:
 - Training Area1 [100 500 60 60]
 - Training Area2 [300 200 60 60]
 - Test Area [500 50 60 60]
 - Urban:
 - Training Area1 [450 400 60 60]
 - Training Area2 [500 250 60 60]
 - Test Area [500 530 60 60]
 - Vegetated
 - Training Area1 [50 50 60 60]
 - Training Area2 [50 250 60 60]
 - Test Area [320 450 60 60]
- Parameters of GLCM
 - local area: 5×5 (pixels)
- Number of gray levels: 8
- Offset: [0 1]
- Properties of available training/target pairs
 - Pairs = 21,600
 - R = 11
 - K = 3
 - P (size $11 \times 21,600$)
 - T (size $3 \times 21,600$)
 - ◆ Training Ratio: 0.01 (obtained by simple iterative tests)
 - Validation Ratio: 0.99
- Properties of NN optimized by our approach
 - Q = Pairs \times trainRatio = 216
 - ◆ b = 4.73 (obtained by BS method)
 - IW = P (size: 216×11)
 - LW = T (size: 3×216)
- Properties of BS Method
 - Tolerance X Value: $1e-3$
 - Tolerance Function Value: $1e-5$
 - Maximum Iterative Steps: 30
- Hardware: Pentium 4 CPU 1.66 GHz, 512 MB of RAM
- Software: PolSARpro v4.0, Neural Network Toolbox of Matlab 7.8(R2009)

6.1. Denoising by Lee Filter

The sub-area (600×600) is shown in Figure 4(a). The Refined Lee filter (Window size = 7) is used to reduce the speckle noise and the results are shown in Figure 4(b). The Lee filter adapts the amount of filtering to the local statistics. Homogeneous areas are filtered with the maximum strength where point scatterers are let unfiltered. The refined filter could use directional windows to preserve edges and heterogeneous features [25].

Figure 4. Pauli image of sub-area of San Francisco.



6.2. Full Features Set

Then, the basic span image and three channels (T_{11}, T_{22}, T_{33}) are easily obtained and shown in Figure 5. The parameters of H/A/Alpha decomposition are shown in Figure 6. The GLCM-based parameters of T_{11}, T_{22}, T_{33} are shown in Figures 7–9.

Figure 5. Basic span image and three channels image.

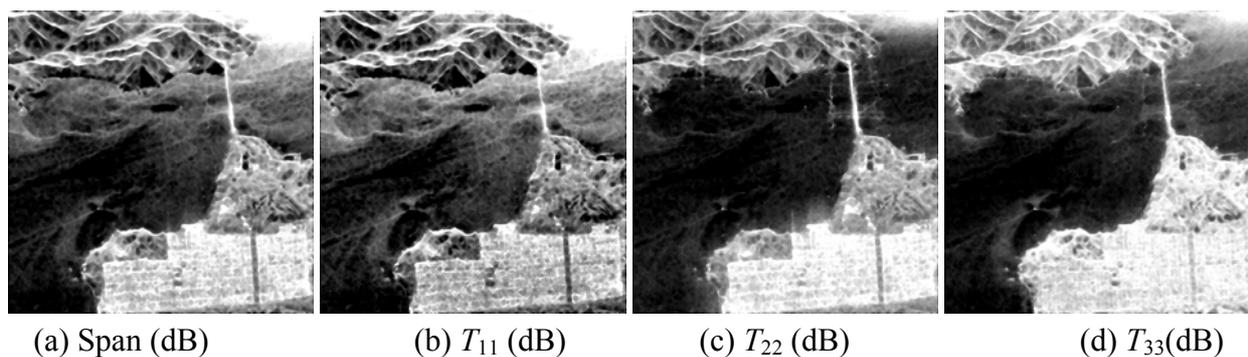


Figure 6. Parameters of H/A/Alpha decomposition.

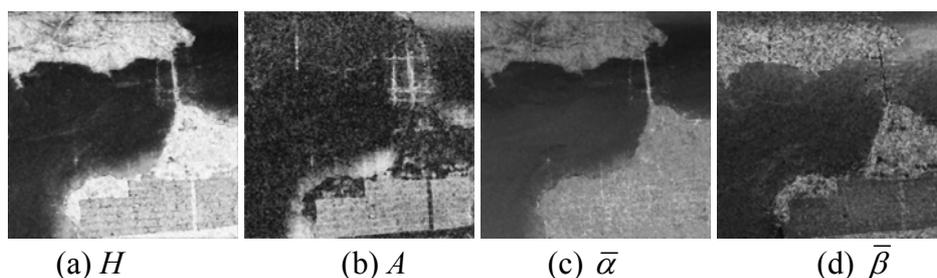
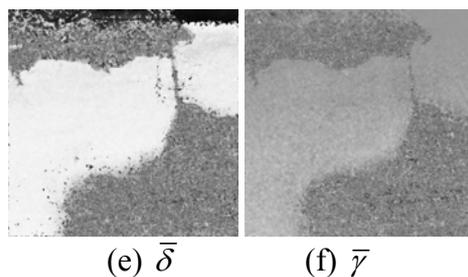
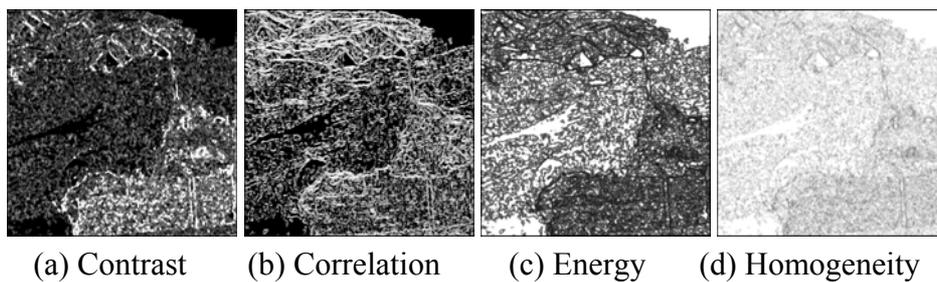
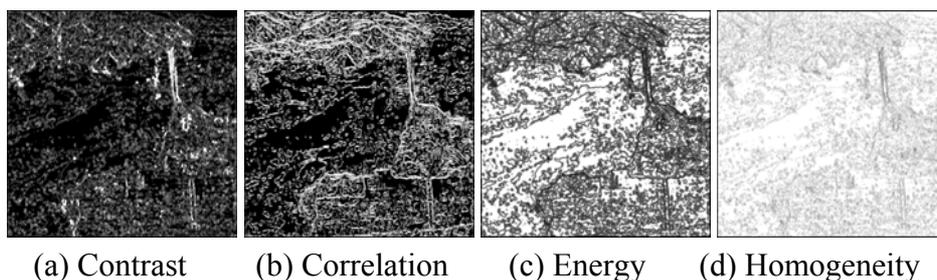
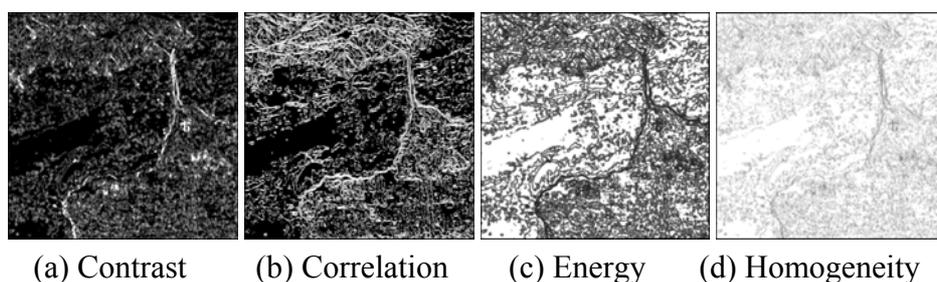
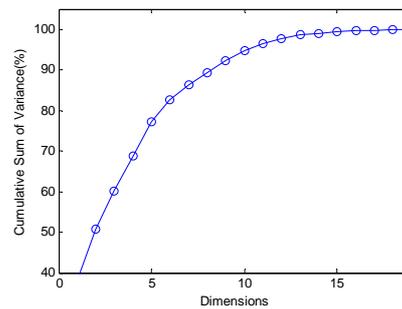


Figure 6. *Cont.***Figure 7.** GLCM-based features of T_{11} .**Figure 8.** GLCM-based features of T_{22} .**Figure 9.** GLCM-based features of T_{33} .

6.3. Feature Reduction by PCA

The curve of cumulative sum of variance with dimensions of reduced vectors via PCA is shown in Figure 10. The detailed data are listed in Table 3. It shows that only 11 features, half the original features only, could preserve 96.36% of variance.

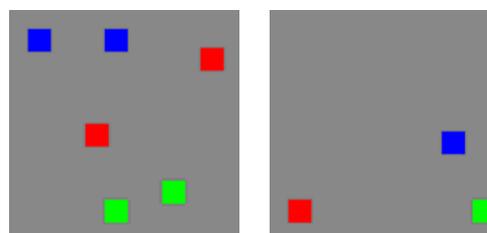
Figure 10. The curve of cumulative sum of variance with dimensions.**Table 3.** Detailed data of PCA on 19 features.

Dimensions	1	2	3	4	5	6	7	8	9
Variance (%)	37.97	50.81	60.21	68.78	77.28	82.75	86.27	89.30	92.27
Dimensions	10	11	12	13	14	15	16	17	18
Variance (%)	94.63	96.36	97.81	98.60	99.02	99.37	99.62	99.80	99.92

Thus, 11 new features obtained via PCA are input to the NN for classification training.

6.4. Training Preparation

The classification is run over three classes, the sea, the urban areas and the vegetated zones. The training and testing areas are selected manually shown in Figures 11(a)-(b), respectively. Each square has a size of 60×60 . In total, there are 21,600 pixels for training, and 10,800 pixels for testing. In this experiment, *trainRatio* is adjusted finally as 0.01, namely, the *validRatio* equals 0.99. In this way, the network only has 1% neurons of that constructed by traditional approach. The training subset and validation subset of the training area are divided randomly.

Figure 11. Sample data of San Francisco (Red denotes sea, green urban areas, blue vegetated zones).

(a) Training area (b) Testing area

6.5. Weights/Biases Setting

The IW and LW are easily set according to our novel approach, and the number of neurons decreases from 21,600 to only 216. The *b* is estimated by BS method. Its initial range is set as [0.01, 20], which is large enough to contain the optimal point. The curve of classification error versus the steps is shown in Figure 12. It is evident that the classification error converges at only three

steps shown in the red dot. However, BS will continue to search the best b value since the tolerance of b is set as small as $1e-3$. The whole process of the change of b is shown in Figure 13.

Figure 12. The curve of error versus step.

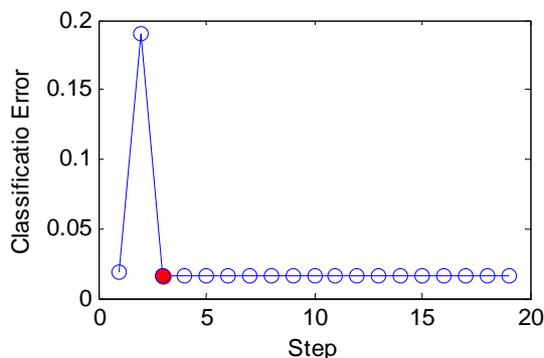
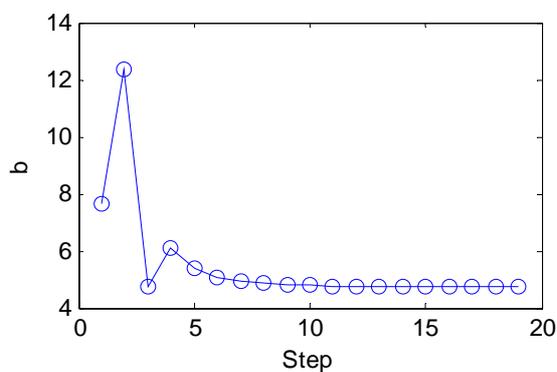


Figure 13. The curve of b versus step.

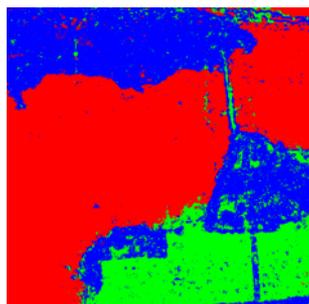


The optimal b is found as 4.73, with the smallest error 1.557%, namely, the highest classification accuracy 98.44%.

6.6. Application to the Whole Image

We use the trained PNN to classify the whole image, and the results are shown in Figure 14. The brims of length 3 are not calculated considering the local area of GLCM, so the size here is only 594×594 .

Figure 14. Classification results of the whole image.



From Figure 14 it makes clear that the sea can be classified perfectly, while the vegetated and urban areas are easily inter-confused. The next section will calculate the confusion matrix which reflects the degree of confusion between the three classes.

6.7. Comparison with Other Approaches

Finally, our method is compared to the 3-layer BPNN [16]. The confusion matrices (CM) by each methods on training area and testing area are listed in Table 4. The element of i th row and j th column in the 3×3 matrix represents the amount of pixels belonging to class j as user defined are assigned to class i after the supervised classification.

Table 4. Comparison of confusion matrix.(O denotes the output class, T denotes the target class).

		Training Area			Testing Area		
		Sea(T)	Urb(T)	Veg(T)	Sea(T)	Urb(T)	Veg(T)
3-layer BPNN	Sea(O)	7158 33.1%	4 0.0%	60 0.3%	3600 33.3%	42 0.4%	5 0.0%
	Urb(O)	0 0%	6882 31.9%	136 0.6%	0 0.0%	3429 31.7%	355 3.3%
	Veg(O)	42 0.2%	314 1.4%	7004 32.4%	0 0.0%	129 1.2%	3240 30.0%
Our Method	Sea(O)	7150 33.1%	0 0.0%	76 0.4%	3597 33.3%	33 0.3%	0 0.0%
	Urb(O)	2 0%	7074 32.8%	74 0.3%	0 0.0%	3445 31.9%	354 3.3%
	Veg(O)	48 0.2%	126 0.6%	7050 32.6%	3 0.0%	122 1.1%	3246 30.1%

It is obvious that the classification accuracies of our proposed method in training area are all higher than 32.5% (33.3% denotes the perfect classification). For the testing area, classification accuracies are all higher than 30.1%. The main drawback is around 3.3% of vegetated zones are misclassified as urban area.

Table 5. Overall accuracies (values are given in percent).

	Training Area	Testing Area
3-layer BPNN	97.4%	95.1%
Our Method	98.5%	95.3%

The overall accuracies are calculated as $CM_{11} + CM_{22} + CM_{33}$ and listed in Table 5, that demonstrates our method has a higher overall accuracy in both training area and testing area than those of 3-layer BPNN. The reason our method outperforms the 3-layer BPNN lies in not only the fact that PNN is adept at predicting the probabilistic results, but also the selected features sets are more discernable.

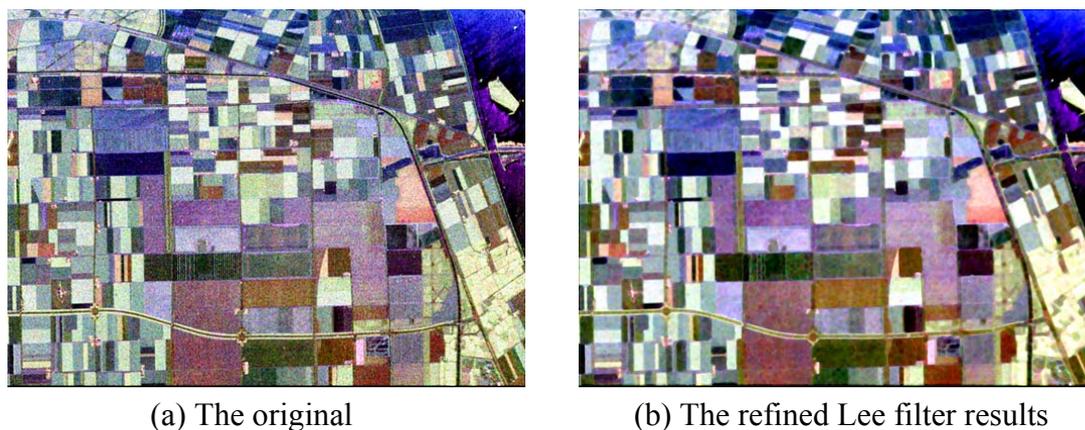
7. Crop Classification

Flevoland, an agricultural area in The Netherlands, was chosen as another example. The site is composed of strips of rectangular agricultural fields. The scene is designated as a supersite for the earth observing system (EOS) program, and is continuously surveyed by the authorities. The ground truth of the test site can be seen in Ref [26].

- Number of features: 19
- ◆ Number of reduced features by PCA: 13 (obtained by performing PCA on total available pairs)
- Location of Train/Test Rectangular Area
 - Bare Soil 1:
 - Train Area [240 300 20 20]
 - Test Area [770 490 20 20]
 - Bare Soil 2
 - Train Area [335 440 20 20]
 - Test Area [420 425 20 20]
 - Barley
 - Train Area [285 500 20 20]
 - Test Area [765 425 20 20]
 - Forest
 - Train Area [959 155 20 20]
 - Test Area [900 490 20 20]
 - Grass
 - Train Area [535 240 20 20]
 - Test Area [500 303 20 20]
 - Lucerne
 - Train Area [550 495 20 20]
 - Test Area [505 550 20 20]
 - Peas
 - Train Area [523 330 20 20]
 - Test Area [436 200 20 20]
 - Potatoes
 - Train Area [32 40 20 20]
 - Test Area [655 307 20 20]
 - Rapeseed
 - Train Area [188 200 20 20]
 - Test Area [280 250 20 20]
 - Stem Beans
 - Train Area [800 350 20 20]
 - Test Area [777 384 20 20]
 - Sugar beet
 - Train Area [877 444 20 20]
 - Test Area [650 225 20 20]
- Water
 - Train Area [965 50 20 20]
 - Test Area [961 201 20 20]
- Wheat
 - Train Area [780 710 20 20]
 - Test Area [700 520 20 20]
- Parameters of GLCM
 - local area: 5×5 (pixels)
 - Number of gray levels: 8
 - Offset: [0 1]
- Properties of available training/target pairs
 - Pairs = 5200
 - R = 13
 - K = 13
 - P (size 13 × 5200)
 - T (size 13 × 5200)
 - ◆ Training Ratio: 0.2 (obtained by simple iterative tests)
 - Validation Ratio: 0.8
- Properties of NN optimized by our approach
 - Q = Pairs × trainRatio = 1040
 - ◆ b = 1.0827 (obtained by BS method)
 - IW = P (size: 13 × 1040)
 - LW = T (size: 13 × 1040)
- Properties of BS Method
 - Tolerance X Value: 1e-3
 - Tolerance Function Value: 1e-5
 - Maximum Iterative Steps: 30
- Hardware: Pentium 4 CPU 1.66 GHz, 512 MB of RAM
- Software: PolSARpro v4.0, Neural Network Toolbox of Matlab 7.8(R2009)

7.1. Refine Lee Filter

The Pauli image of Flevoland is shown in Figure 15(a), and the refined Lee filtered image (Window Size = 7) is shown in Figure 15(b).

Figure 15. Pauli Image of Flevoland (1024×750).

7.2. Full Features

The basic span image and three channels (T_{11}, T_{22}, T_{33}) are easily obtained and shown in Figure 16. The parameters of H/A/Alpha decomposition are shown in Figure 17. The GLCM-based parameters of T_{11}, T_{22}, T_{33} are shown in Figures 18–20.

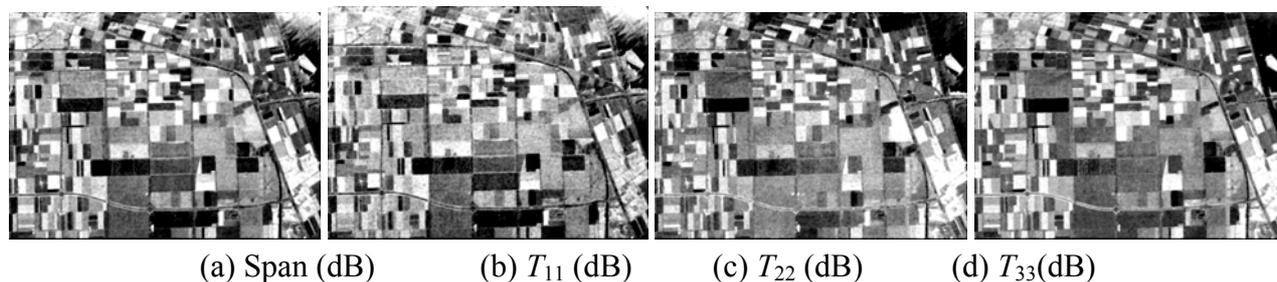
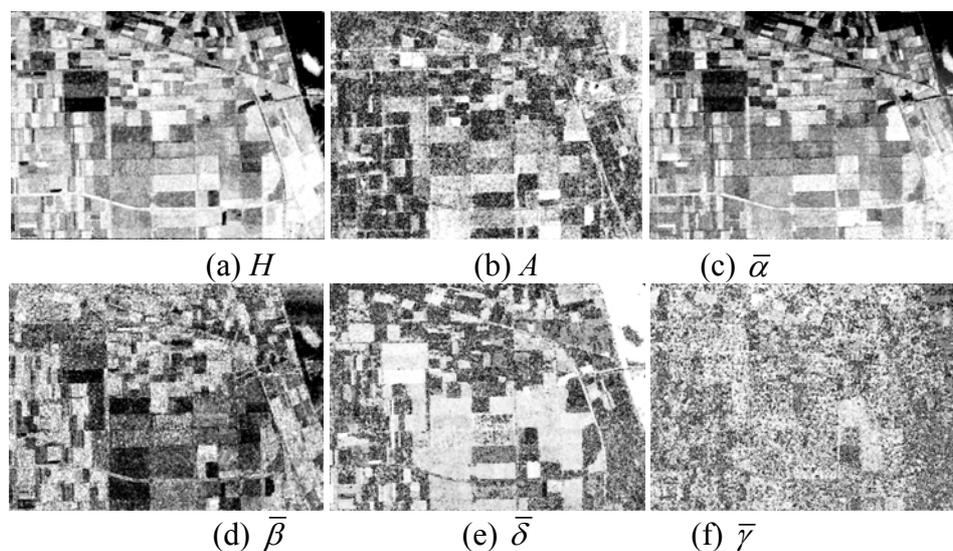
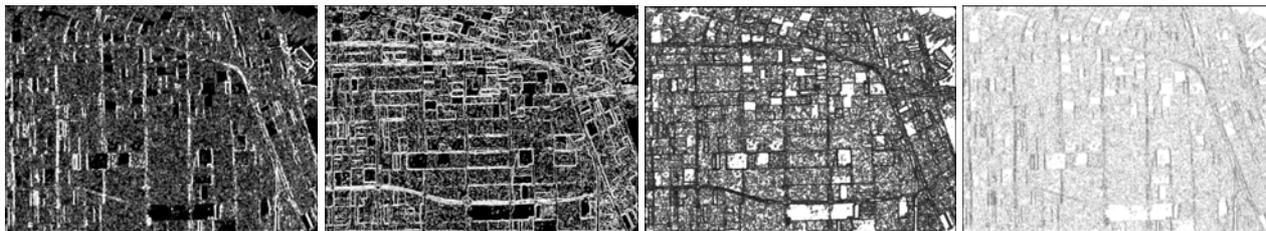
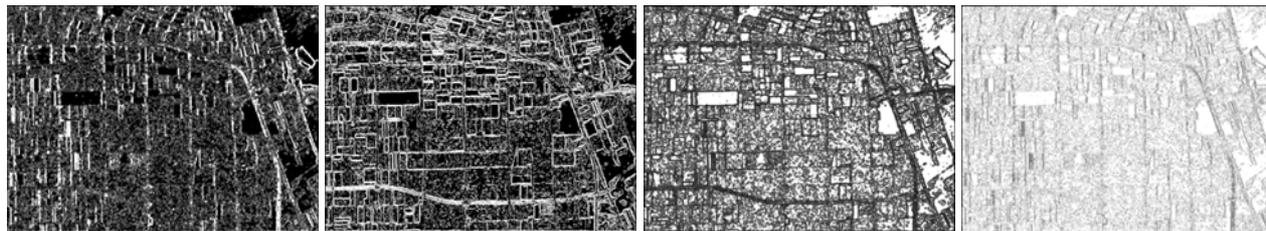
Figure 16. Basic span image and three channels image.**Figure 17.** Parameters of H/A/Alpha decomposition.

Figure 18. GLCM-based features of T_{11} .



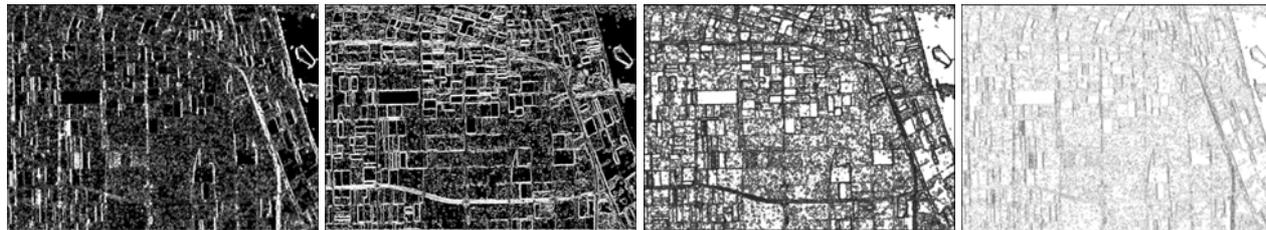
(a) Contrast (b) Correlation (c) Energy (d) Homogeneity

Figure 19. GLCM-based features of T_{22} .



(a) Contrast (b) Correlation (c) Energy (d) Homogeneity

Figure 20. GLCM-based features of T_{33} .



(a) Contrast (b) Correlation (c) Energy (d) Homogeneity

7.3. Feature Reduction

The curve of cumulative sum of variance with dimensions of reduced vectors via PCA is shown in Figure 21. The detailed data are listed in Table 6. It shows that only 13 features, which are only half the original features, could preserve 98.06% of variance.

Figure 21. The curve of cumulative sum of variance with dimensions.

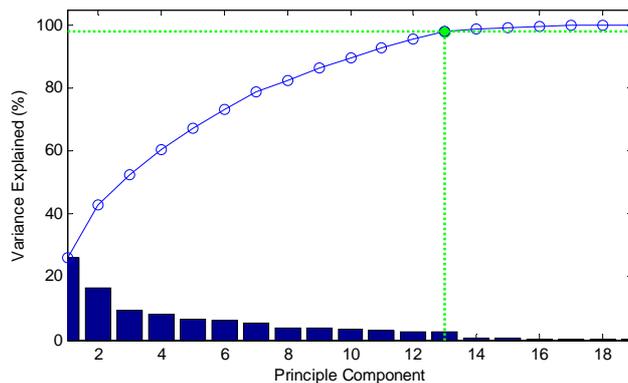


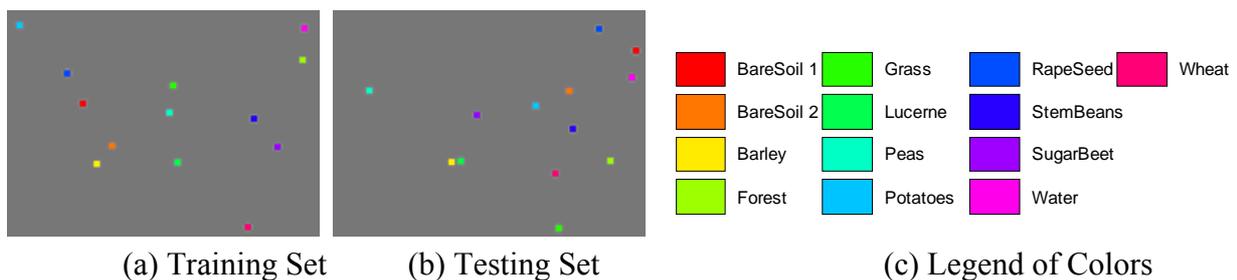
Table 6. Detailed data of PCA on 19 features.

Dimensions	1	2	3	4	5	6	7	8	9
Variance (%)	26.31	42.98	52.38	60.50	67.28	73.27	78.74	82.61	86.25
Dimensions	10	11	12	13	14	15	16	17	18
Variance (%)	89.52	92.72	95.50	98.06	98.79	99.24	99.63	99.94	99.97

7.4. Training Preparation

The classification is run over 11 classes, bare soil 1, bare soil 2, barley, forest, grass, Lucerne, peas, potatoes, rapeseed, stem beans, and sugar beet. They are selected manually according to the ground truth [26]. The training set and testing set are shown in Figure 22. Each square has a size of 20×20 . In total, there are 5,200 pixels for training, and 5,200 pixels for testing.

Since the types of classes increase (3 to 13) while the available data decrease (21,800 to 5,200), thus, we should dispose more data to the training subset. Finally, *trainRatio* is adjusted as 0.2, while *validRatio* is set as 0.8. In this way, the network only has 1/5 neurons of that constructed by traditional approach. The training subset and validation subset of the training area are divided randomly.

Figure 22. Sample data areas of Flevoland.

7.5. Weights/Biases Setting

The IW and LW are easily set according to our novel approach, and the number of neurons decreases from 5200 to only 1040. The *b* is estimated by BS method. Its initial range is set as before. The curve of classification error versus the steps is shown in Figure 23. It is evident that the classification error reaches the minimum at 17th steps shown in the red dot. The whole process of the change of *b* is shown in Figure 24.

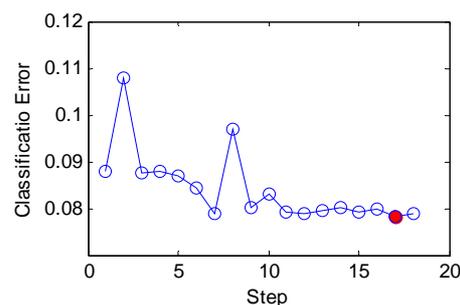
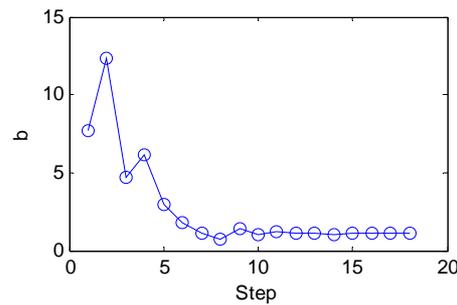
Figure 23. The curve of error versus step.

Figure 24. The curve of b versus step.



The optimal b is found as 1.0827, with the smallest error 7.8%, namely, the highest classification accuracy on the validation subset of training area is 98.44%.

7.6. Classification Results

The confusion matrices on training area and testing area are calculated and listed in Figure 25 and Figure 26. The overall accuracy of our method on training area and test area are 93.71% and 86.2% respectively.

We apply our method on the whole image. The results are shown in Figure 27. From Figure 27 it is clear that our method can classify most of areas correctly.

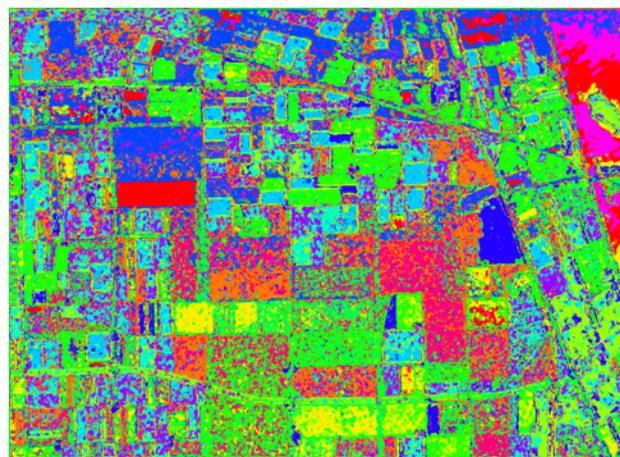
Figure 25. Confusion matrix comparison on train area (values are given in percent)
The overall accuracy is 93.71%.

1	399 7.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
2	0 0.0%	391 7.5%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 0.1%	0 0.0%	5 0.1%	97.3%
3	0 0.0%	0 0.0%	389 7.5%	0 0.0%	4 0.1%	27 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.6%
4	0 0.0%	0 0.0%	0 0.0%	335 6.4%	0 0.0%	0 0.0%	35 0.7%	18 0.3%	0 0.0%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	85.9%
5	0 0.0%	0 0.0%	8 0.2%	0 0.0%	383 7.4%	25 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	91.8%
6	0 0.0%	0 0.0%	3 0.1%	0 0.0%	13 0.3%	336 6.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 0.2%	92.6%
7	0 0.0%	0 0.0%	0 0.0%	31 0.6%	0 0.0%	0 0.0%	340 6.5%	10 0.2%	0 0.0%	0 0.0%	18 0.3%	0 0.0%	0 0.0%	85.2%
8	0 0.0%	0 0.0%	0 0.0%	27 0.5%	0 0.0%	0 0.0%	17 0.3%	363 7.0%	0 0.0%	1 0.0%	11 0.2%	0 0.0%	0 0.0%	86.6%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	399 7.7%	0 0.0%	0 0.0%	3 0.1%	0 0.0%	99.3%
10	0 0.0%	0 0.0%	0 0.0%	4 0.1%	0 0.0%	0 0.0%	3 0.1%	7 0.1%	0 0.0%	398 7.7%	0 0.0%	0 0.0%	0 0.0%	96.6%
11	0 0.0%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	3 0.1%	4 0.1%	2 0.0%	0 0.0%	0 0.0%	359 6.9%	0 0.0%	0 0.0%	97.0%
12	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	397 7.6%	0 0.0%	99.5%
13	0 0.0%	8 0.2%	0 0.0%	0 0.0%	0 0.0%	9 0.2%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	6 0.1%	0 0.0%	384 7.4%	94.1%
	99.8% 0.2%	97.8% 2.2%	97.3% 2.7%	83.8% 16.2%	95.8% 4.2%	84.0% 16.0%	85.0% 15.0%	90.8% 9.3%	99.8% 0.2%	99.5% 0.5%	89.8% 10.3%	99.3% 0.7%	96.0% 4.0%	93.7% 6.3%
	1	2	3	4	5	6	7	8	9	10	11	12	13	

Figure 26. Confusion matrix comparison on test area (values are given in percent)
The overall vccuracy is 86.2%.

Output Class	1	400 7.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	99.8% 0.2%	
	2	0 0.0%	373 7.2%	0 0.0%	1 0.0%	0 0.0%	7 1.3%	22 0.4%	3 0.1%	10 0.2%	0 0.0%	2 0.0%	1 0.0%	28 0.5%	83.4% 16.6%
	3	0 0.0%	0 0.0%	363 7.0%	0 0.0%	80 1.5%	66 1.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	71.3% 28.7%
	4	0 0.0%	0 0.0%	0 0.0%	321 6.2%	0 0.0%	1 0.0%	5 0.1%	7 0.1%	0 0.0%	0 0.0%	62 1.2%	0 0.0%	0 0.0%	81.1% 18.9%
	5	0 0.0%	0 0.0%	2 0.0%	0 0.0%	316 6.1%	45 0.9%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	86.6% 13.4%
	6	0 0.0%	0 0.0%	33 0.6%	0 0.0%	4 0.1%	277 5.3%	16 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 0.4%	79.1% 20.9%
	7	0 0.0%	1 0.0%	0 0.0%	45 0.9%	0 0.0%	0 0.0%	269 5.2%	4 0.1%	0 0.0%	0 0.0%	45 0.9%	0 0.0%	0 0.0%	73.9% 26.1%
	8	0 0.0%	0 0.0%	0 0.0%	24 0.5%	0 0.0%	0 0.0%	30 0.6%	381 7.3%	0 0.0%	0 0.0%	15 0.3%	0 0.0%	0 0.0%	84.7% 15.3%
	9	0 0.0%	14 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	371 7.1%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	95.9% 4.1%
	10	0 0.0%	0 0.0%	0 0.0%	8 0.2%	0 0.0%	0 0.0%	1 0.1%	5 0.1%	0 0.0%	400 7.7%	11 0.2%	0 0.0%	0 0.0%	94.1% 5.9%
	11	0 0.0%	12 0.2%	0 0.0%	1 0.0%	0 0.0%	4 0.1%	49 0.9%	0 0.0%	5 0.1%	0 0.0%	265 5.1%	0 0.0%	0 0.0%	78.9% 21.1%
	12	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 0.3%	0 0.0%	0 0.0%	395 7.6%	0 0.0%	96.6% 3.4%
	13	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	7 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	351 6.8%	97.2% 2.8%
		100% 0.0%	93.3% 6.8%	90.8% 9.3%	80.3% 19.8%	79.0% 21.0%	69.3% 30.8%	67.3% 32.8%	95.3% 4.7%	92.8% 7.3%	100% 0.0%	66.3% 33.8%	98.8% 1.2%	87.8% 12.3%	86.2% 13.8%
		1	2	3	4	5	6	7	8	9	10	11	12	13	
		Target Class													

Figure 27. Classification Map of our method.



8. Discussion

The BS has an important effect on our algorithm as shown in Figure 12 and Figure 23. It can guide users to find the optimal *b* value in quite short steps. Otherwise, the users will take a long time with the help of exhaustive search algorithm. The function of the combined feature, random division and PCA will be discussed in detail at following paragraphs.

8.1. Single Type of Feature Set versus Combined Feature Sets

The feature sets can be divided into two types. One is the polarimetric feature set, which contains the span, the six H/A/ α parameters; the other is the texture feature set, which contains the properties extracted from the GLCM.

Table 7 lists the classification accuracy of classifiers using polarimetric feature set, texture feature set, and combined feature set. It indicates that the polarimetric features contribute most to the classification while the texture feature contribute less. Then, we can find that the combined feature set performs better than each single. Thus, our classifier using combined feature set can be regarded as an feature fusion method.

Table 7. Comparison of PNNs using polarimetric feature set, texture feature set, and combined feature set (TR denotes Classification Accuracy of Total Random).

Site		Polarimetric feature set	Texture feature set	Combined feature set
<i>San Francisco</i> (TR=33.3%)	<i>Training Area</i>	97.1%	59.9%	98.5%
	<i>Test Area</i>	87.4%	45.9%	95.3%
<i>Flevoland</i> (TR=7.69%)	<i>Training Area</i>	92.2%	48.0%	93.7%
	<i>Test Area</i>	72.2%	24.1%	86.2%

8.2. With and without Random Division

If we do not use the random division, the structure of PNN will increase $1/trainRatio$ times. Consequently, the computation will become a burden with very little improve on classification overall accuracy. Taking the San Francisco area as the example, four square areas of different size are picked out randomly from the image, and are classified by PNNs with and without random division. The computation time and overall accuracy of each are listed in Table 8.

Table 8. Comparison of PNN with and without our weights/biases setting (RD denotes Random Division).

Area Size	Computation Time			Overall Accuracy	
	Without RD	With RD	Ratio	Without RD	With RD
10 × 10	1.0818	0.0231	46.8	94.8%	94.9%
20 × 20	4.0803	0.0386	105.7	95.5%	95.5%
30 × 30	22.4270	0.0751	298.6	96.3%	96.2%
40 × 40	58.1409	0.1125	516.8	95.9%	95.4%

Table 8 indicates that the computation time of traditional method is only 46 times of that of our method for 10×10 area, however, the ratio rockets to 516 for 40×40 area. Moreover, for a larger size area, such as 50×50 , it cannot work because of the lack of memory.

From another point of view, the overall accuracy of traditional method was expected to be much higher than that of our method since it uses a great many neurons, whereas, in fact, they are nearly the same. The reason may consist of the optimization of b in our method. Accordingly, our method of weights/biases setting is valid and effective, and it is superior to traditional method in terms of computation time and storage room while it can maintain a high overall accuracy.

8.3. With and without PCA

PNNs with and without PCA are investigated in the same manner as in Section 7.1. Their computation times are depicted in Figure 28, which indicates that PNN with PCA enjoys a less computation time than that of PNN without PCA. Their time differences are gradually becoming large as the width of the randomly selected area is increasing.

In addition, the overall accuracies of these two PNNs are observed. It should be noted that input data of the PNN without PCA still should be normalized although the PCA is omitted, otherwise the performance of PNN will decrease rapidly.

The overall accuracies obtained by the two PNNs are pictured in Figure 29. It demonstrates that the PNN with PCA outperforms PNN without PCA on the small test area (width < 40). As the area becomes large ($40 < \text{width} < 47$), the PNN without PCA is better. Finally, as the area becomes large enough (width > 47), these performances of the two PNNs are nearly equivalent. Therefore, our method embedding PCA can perform faster, and has no loss of overall accuracy.

Figure 28. Computation time with square width.

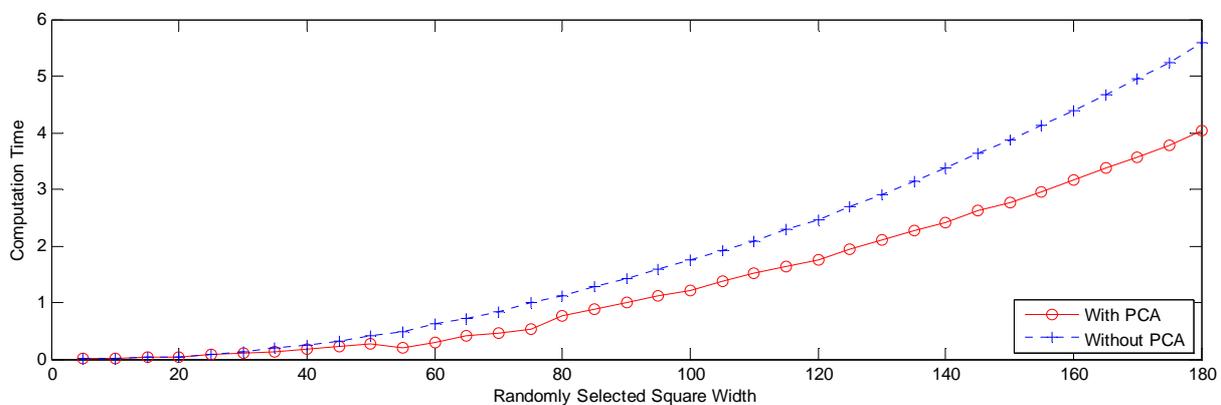
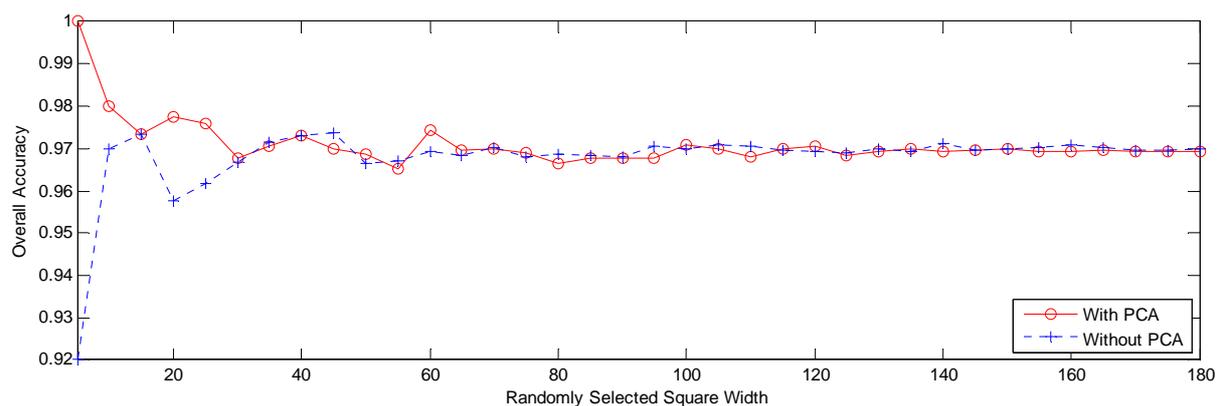


Figure 29. The overall accuracy versus square width.



9. Conclusions

In this paper, a hybrid feature set has been introduced which is made up of the span image, the H/A/ α decomposition, and the GLCM-based texture features. Then, a probabilistic neural network has been established. We proposed a novel weights/biases setting method based on Brent's method and PCA. The method can decrease the feature set, reduce the number of neurons, and find optimal bias values.

Experiments of terrain classification on a San Francisco site and a crop classification on Flevoland show that our method can obtain good results which are more accurate than those of 3-layer BPNN. Afterwards, combined feature set, random division and PCA are assumed to be omitted in turn, and the results prove the indispensability of each improvement.

References and Notes

1. Pellizzeri, T.M. Classification of polarimetric SAR images of suburban areas using joint annealed segmentation and "H/A/ α " polarimetric decomposition. *ISPRS J. Photogramm. Remote Sens.* **2003**, *58*, 55–70.
2. Lee, J.S.; Grunes, M.R.; Kwok, R. Classification of multi-look polarimetric SAR imagery based on complex Wishart distribution. *Int. J. Remote Sens.* **1994**, *15*, 2299–2311.
3. Shimoni, M.; Borghys, D.; Heremans, R.; Perneel, C.; Acheroy, M. Fusion of PolSAR and PolInSAR data for land cover classification. *Int. J. Appl. Earth Observ. Geoinf.* **2009**, *11*, 169–180.
4. Cloude, S.R.; Pottier, E. A review of target decomposition theorems in radar polarimetry. *IEEE Trans. Geosci. Remote Sens.* **1996**, *34*, 498–518.
5. Krogager, E. New decomposition of the radar target scattering matrix. *Electron. Lett.* **1990**, *26*, 1525–1527.
6. Freeman, A.; Durden, S.L. A three-component scattering model for polarimetric SAR data. *IEEE Trans. Geosci. Rem. Sens.* **1998**, *36*, 963–973.
7. Huynen, J.R. Phenomenological Theory of Radar Targets. PhD dissertation. University of Technology: Delft, The Netherlands, 1970.
8. Holm, W.A.; Barnes, R.M. On radar polarization mixed target state decomposition techniques. *Proc. of IEEE Radar Conference*, Ann Arbor, MI, USA, 1988; pp. 249–254.
9. Cloude, S.R.; Pottier, E. An entropy based classification scheme for land applications of polarimetric SAR. *IEEE Trans. Geosci. Remote Sens.* **1997**, *35*, 549–557.
10. Lin, C.J.; Chung, I.F.; Chen, C.H. An entropy-based quantum neuro-fuzzy inference system for classification applications. *Neurocomputing* **2007**, *70*, 2502–2516.
11. Kido, S.; Tamura, S. Computerized classification of interstitial lung abnormalities on chest radiographs with normalized radiographic index and normalized fractal dimension. *Eur. J. Radiol.* **2001**, *37*, 184–189.
12. Ranson, K.J.; Sun, G.Q. An evaluation of AIRSAR and SIR-C/X-SAR images for mapping northern forest attributes in Maine, USA. *Remote Sens. Environ.* **1997**, *59*, 203–222.
13. Avci, E.; Turkoglu, I.; Poyraz, M. Intelligent target recognition based on wavelet packet neural network. *Expert Syst. Appl.* **2005**, *29*, 175–182.

14. Acqua, F.D.; Gamba, P.; Trianni, G. Semi-automatic choice of scale-dependent features for satellite SAR image classification. *Patt. Recogn. Lett.* **2006**, *27*, 244–251.
15. Cooper, G.R.J.; Cowan, D.R. The use of textural analysis to locate features in geophysical data. *Comput. Geosci.* **2005**, *31*, 882–890.
16. Khan, K.U.; Yang, J. Polarimetric synthetic aperture radar image classification by a hybrid method. *Tsinghua Sci. Technol.* **2007**, *12*, 97–104.
17. Lee, J.J.; Kim, D.; Chang, S.K.; Nocete, C.F.M. An improved application technique of the adaptive probabilistic neural network for predicting concrete strength. *Comput. Mater. Sci.* **2009**, *44*, 988–998.
18. Mostafa, M.M. Modeling the competitive market efficiency of Egyptian companies: A probabilistic neural network analysis. *Expert Syst. Appl.* **2009**, *36*, 8839–8848.
19. Armand, P.; Benoist, J.; Bousquet, E.; Delage, L.; Olivier, S.; Reynaud, F. Optimization of a one dimensional hypertelescope for a direct imaging in astronomy. *Eur. J. Oper. Res.* **2009**, *195*, 519–527.
20. Luukka, P. Classification based on fuzzy robust PCA algorithms and similarity classifier. *Expert Syst. Appl.* **2009**, *36*, 7463–7468.
21. Pottier, E.; Cloude, S.R. Application of the H/A/ α polarimetric decomposition theorems for land classification. *Proc. SPIE Conference on Wideband Interferometric Sensing and Imaging Polarimetry*, San Diego, CA, USA, 1997; pp. 132–143.
22. Tien, C.L.; Lyu, Y.R.; Jyu, S.S. Surface flatness of optical thin films evaluated by gray level co-occurrence matrix and entropy. *Appl. Surf. Sci.* **2008**, *254*, 4762–4767.
23. Chen, C.H.; Chu, C.T. High performance iris recognition based on 1-D circular feature extraction and PSO–PNN classifier. *Expert Syst. Appl.* **2009**, *36*, 10351–10356.
24. Luukka, P. Classification based on fuzzy robust PCA algorithms and similarity classifier. *Expert Syst. Appl.* **2009**, *36*, 7463–7468.
25. Gupta, K.K.; Gupta, R. Despeckle and geographical feature extraction in SAR images by wavelet transform. *ISPRS J. Photogramm. Remote Sens.* **2007**, *62*, 473–484.
26. Chen, K.S.; Huang, W.P.; Tsay, D.H.; Amar, F. Classification of multifrequency polarimetric SAR imagery using a dynamic learning neural network. *IEEE Trans. Geosci. Remote Sens.* **1996**, *34*, 814–820.

© 2009 by the authors; licensee Molecular Diversity Preservation International, Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).