Int. J. Mach. Learn. & Cyber. (2011) 2:261–271
DOI 10.1007/s13042-011-0024-1

ORIGINAL ARTICLE

# Positive and negative fuzzy rule system, extreme learning machine and image classification

**Wu Jun · Wang Shitong · Fu-lai Chung**

Received: 23 November 2010 / Accepted: 31 May 2011 / Published online: 21 June 2011


**Abstract** We often use the positive fuzzy rules only for image classification in traditional image classification systems, ignoring the useful negative classification information. Thanh Minh Nguyen and QMJonathan Wu introduced the negative fuzzy rules into the image classification, and proposed combination of positive and negative fuzzy rules to form the positive and negative fuzzy rule system, and then applied it to remote sensing image/natural image classification. Their experiments demonstrated that their proposed method has achieved promising results. However, since their method was realized using the feedforward neural network model which requires adjusting the weights in the gradient descent way, the training speed is very slow. Extreme learning machine (ELM) is a single hidden layer feedforward neural network (SLFNs) learning algorithm, which has distinctive advantages such as quick learning, good generalization performance. In this paper, the equivalence between ELM and the positive and negative fuzzy rule system is revealed, so ELM can be naturally used for training the positive and negative fuzzy rule system quickly for image classification. Our experimental results indicate this claim.

**Keywords** Image classification · Positive and negative fuzzy rules · Extreme learning machine · Fuzzy systems

W. Jun · W. Shitong (✉)
School of Digital Media, JiangNan University, WuXi, China
e-mail: wxwangst@yahoo.com.cn

W. Shitong · F. Chung
Department of Computing, Hong Kong Polytechnic University, Hong Kong, China

## 1 Introduction

With the fast development of the digital image processing technologies and a huge of demands for practical applications, image classification and recognition technologies have been developing rapidly recent years. Image classification technologies are specific applications of pattern recognition in image processing [3], and their purpose is to develop automatic image processing systems that can help us complete image classification and recognition tasks and provide us a lot of useful information mined from images for further experiments and researches. Among these technologies, due to its strong nonlinear approximation capability, feedforward neural networks have been attracting more and more attentions and obtaining various applications in image classification [4–6]. As we may know well, images can be easily corrupted by noise, so it is not a trivial task to classify noisy images using feedforward neural networks, which results in a growing research interests in the application of fuzzy rule systems and/or their fuzzy neural systems to image classification tasks recent years [7–9]. Most of these fuzzy systems use positive fuzzy rules about useful positive classification information to classify images, ignoring the valuable negative classification information. For example, a typical fuzzy classification rule $r$ [10–12]: IF $x_1$ is $A_{r1}$ and $x_2$ is $A_{r2}$… and $x_N$ is $A_{rN}$, Then $y_1$ is $C_1$ with $W_{r1}$ and $y_2$ is $C_2$ with $W_{r2}$… and $y_M$ is $C_M$ with $W_{rM}$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]$ is the $N$ dimensional input, $\boldsymbol{y} = [y_1, y_2, \ldots, y_M]$ is the corresponding output, $M$ classes are denoted by $C_1, C_2, \ldots C_M$; $A_{rn}$, $n = (1, 2, \ldots, N)$ is the fuzzy membership function; $W_{rm} \geq 0$, $m = (1, 2, \ldots, M)$ is the weight of each class. It can be seen that such a fuzzy classification rule only considers the positive information (i.e., the right value is positive), ignoring possibly useful

negative classification information which can be represented by negative weights.

In order to circumvent the shortcoming of discovering negative classification information, Thanh Minh Nguyen proposed the combination strategy of positive and negative fuzzy rules to effectively classify images [2]. Although their experimental results are promising, just like most of feedforward neural networks [13], since all the parameters within the network need to be adjusted in the gradient decent way, their method heavily suffers from the very slow learning speed for the training set with several hundreds and even thousands of samples constructed from the image and easily falling in local minima of the cost function of the network [14]. Extreme learning machine (ELM) [1], as the latest advance in training the single-hidden layer feedforward neural network, has the following distinctive advantages over all other learning algorithms of feedforward neural networks: it randomly chooses hidden nodes and analytically determines weights of the single-hidden layer feedforward neural network without falling in the so-called local minima and tends to provide good generalization performance at extremely fast learning speed. In this paper, we will reveal the equivalence between ELM and the positive and negative fuzzy rule system, and then utilize ELM for fast training this positive and negative fuzzy rule system for image classification. The contributions of this work here are twofold:

(1) Since the equivalence between ELM and the positive and negative fuzzy rule system is revealed, we can explain ELM as the positive and negative fuzzy rule system, or vice versa when it is applied to classification tasks. In other words, we can extract positive and negative rules from the single-hidden layer feedforward neural network using ELM learning while we can only extract positive rules from the network using BP or BP-like learning in previous researches [15].
(2) The work done by Thanh Minh Nguyen for image classification [2] is highlighted here by using ELM learning instead of BP learning for the positive and negative fuzzy rule system.

This paper is organized as follows. In Sect. 2 the positive and negative fuzzy rule system is briefly reviewed; In Sect. 3, Extreme Learning Machine (ELM) and its theoretical characteristics are briefly introduced. In Sect. 4, the equivalence between ELM and the positive and negative fuzzy rule system is revealed. In Sect. 5, the experimental results about image classification for remote sensing images and natural images using the proposed method in this paper, the fuzzy C means clustering algorithm and BP neural network are reported. Section 6 concludes the paper.

## 2 Positive and negative fuzzy rule system

In this section, let us briefly review the positive and negative fuzzy rule system. More details can be seen in [2]. In general, fuzzy rule systems can be categorized into two families. The first includes linguistic models based on collections of fuzzy rules, whose antecedents and consequents utilize fuzzy values. The famous Mamdani model falls into this group. The second family, based on Sugeno-type systems, uses a rule structure that has fuzzy antecedent and functional or singleton consequent parts. In a fuzzy rule system, if we assume A is the premise of a fuzzy rule and B is the consequent of the fuzzy rule, a typical fuzzy rule of the "IF–Then" type is "IF A then do B". This type of fuzzy rule is called positive rule (weight is positive) because the consequent prescribes something that should be done, or an action to be taken. However, if an action might lead to severe damage, then the action should be avoided. This kind of action is also possible to augment the rule-base with fuzzy rules in the form: "IF A, Then do not do B". This type of fuzzy rule is often called negative rules (weight is negative) because the consequent prescribes something that should be avoided instead of done. In most existing fuzzy-rule-system based image classification systems, only positive information (i.e., positive fuzzy rules) are considered with ignoring negative information. In fact, both positive and negative information may be very useful for image classification.

Now let us consider the following example about two fuzzy rules in [2, 16, 17]:

*Rule* 1: IF customer is a child
Then he buys Coke and he does not buy bottled water.
*Rule* 2: IF customer is an adult
Then he buys Coke and he buys bottled water.

In this example, the negative rule (Rule 1) guides the system away from scenarios to be avoided, and after avoiding these scenarios, the positive rules (Rule 2) once again take over and direct the process. Depending on the probability of such an association, marketing personnel can develop better planning of the shelf space in the store or can base their marketing strategies on such correlations found in the data.

Another limitation of the above fuzzy rules is that these two classes (Coke, bottled water) appearing in the consequence parts of the above fuzzy rules have the same degree of importance. Obviously, to help marketing personnel develop better planning of different products (Coke, bottled water) for different customers (child, adult), we should assign different weights to different classes appearing in the consequence part of the rule.

These discussions about the above example motivate us to propose the following positive and negative fuzzy rule

system for classification. In this system, the structure of a fuzzy rule takes the following form:

*Rule r* IF $x_{1k}$ is $A_{r1}$ and $x_{2k}$ is $A_{r2}$… and $x_{Nk}$ is $A_{rN}$, Then $y_{k1}$ is the class $C_1$ with $W_{r1}$ and $y_{k2}$ is the class $C_2$ with $W_{r2}$… and $y_{kM}$ is the class $C_M$ with $W_{rM}$

where $\boldsymbol{x_k} = [x_{1k}, x_{2k}, \ldots, x_{Nk}]$, $k = (1, 2, \ldots, K)$ is a $N$ dimensional input vector. $A_{rn}, n = (1, 2, \ldots, N)$ denotes a Gaussian fuzzy membership function [18] or other types of fuzzy membership functions, $W_{rm} \in R$, $r = (1, 2, \ldots, R)$, $m = (1, 2, \ldots, M)$, is the weight of the class $C_m$. When $W_{rm} > 0$, it denotes the weight of $x_k$ belonging to the class $C_m$. When $W_{rm} < 0$, it will narrow the choices for the class $C_m$. Clearly, $R$, $M$, $K$ and $N$ denote the number of fuzzy rules, the number of classes, the number of patterns and dimension of patterns, respectively. Without loss of generality, we adopt Gaussian fuzzy membership functions here.

Obviously, the distinctive advantage of this type of fuzzy rules exists in the fact that it can represent more than one class therein. Based on the above structure of a fuzzy rule, given the input $\boldsymbol{x_k}$, the multi-output of the corresponding positive and negative fuzzy rule system is of the following type:

$$\begin{aligned} y_{km} &= \sum_{r=1}^{R} \overline{\beta}_r(\mathbf{x_k}) W_{rm} \\ &= \frac{\sum_{r=1}^{R} W_{rm} \beta_r(\mathbf{x_k})}{\sum_{r=1}^{R} \beta_r(\mathbf{x_k})} \\ &= \frac{\sum_{r=1}^{R} W_{rm} \exp\left[-\sum_{n=1}^{N} \frac{(x_{nk}-\mu_{rn})^2}{\sigma_{rn}^2}\right]}{\sum_{r=1}^{R} \exp\left[-\sum_{n=1}^{N} \frac{(x_{nk}-\mu_{rn})^2}{\sigma_{rn}^2}\right]}, \\ & m = (1, 2, \ldots, M) \end{aligned} \tag{1}$$

where

$$\overline{\beta}_r(\mathbf{x_k}) = \frac{\beta_r(\mathbf{x_k})}{\sum_{r=1}^{R} \beta_r(\mathbf{x_k})},$$

$$\beta_r(\mathbf{x_k}) = \exp\left[-\sum_{n=1}^{N} \frac{(x_{nk}-\mu_{rn})^2}{\sigma_{rn}^2}\right] \tag{2}$$

in which $\overline{\beta}_r(\mathbf{x_k})$is often called as the fuzzy basis function, and $\mu_{rn}$, $\sigma_{rn}$, $r = (1, 2, \ldots, R)$, $n = (1, 2, \ldots, N)$ are the mean and variance of $\beta_r(\mathbf{x_k})$, respectively. The fuzzy rule weight $W_{rm}$ will be mentioned in detail in the next section. The output of the classifier based on this fuzzy rule system is determined by the winner-takes-all strategy. That is, $\mathbf{x_k}$ will belong to the class with the highest activation, i.e.,

$$y_k = C_{m'}; m' = \max_{1 \le m \le M}(y_{km}) \tag{3}$$

The above fuzzy rule system can be easily realized using the following fuzzy feedforward neural network, as shown in Fig. 1, which consists of two visible layers (input and output layer) and three hidden layers.

Layer 1 (input layer) Each node in this layer only transmits input $x_{nk}$, $n = (1, 2, \ldots, N)$, $k = (1, 2, \ldots, K)$ to the next layer directly, i.e., $O_{1n} = x_{nk}$ $n = (1, 2, \ldots, N)$, $k = (1, 2, \ldots, K)$

Layer 2 The number of nodes in this layer is equal to the number of fuzzy rules. Each node in this layer has $N$ inputs from $N$ nodes of the input layer, and feeds its output to the node of the layer 3. The output of each node in this layer is

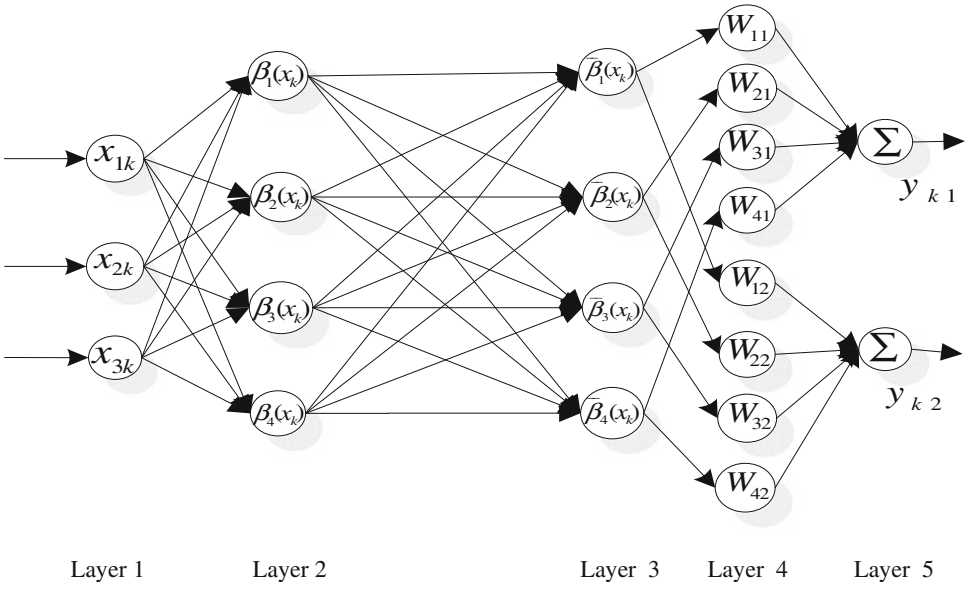



**Fig. 1** Fuzzy neural network with three inputs, two outputs and four fuzzy rules

$$O_{2rk} = \beta_r(\mathbf{x_k}) = \exp\left[-\sum_{n=1}^{N}\frac{(x_{nk}-\mu_{rn})^2}{\sigma_{rn}^2}\right] \quad (4)$$

in which $\mu_{rn}, \sigma_{rn}$, $r = (1, 2, \ldots, R)$, $n = (1, 2, \ldots, N)$ are the mean and variance of $\beta_r(\boldsymbol{x_k})$, respectively.

Layer 3 This layer performs the normalization operation. The output of each node in this layer is represented by

$$O_{3r} = \overline{\beta}_r(\mathbf{x_k}) = \frac{\beta_r(\mathbf{x_k})}{\sum_{r=1}^{R}\beta_r(\mathbf{x_k})} \quad (5)$$

Layer 4 Each node of this layer represents the rule weight. The output of each node in this layer is represented by

$$O_{4rm} = W_{rm}\frac{\beta_r(\mathbf{x_k})}{\sum_{r=1}^{R}\beta_r(\mathbf{x_k})} = W_{rm}\overline{\beta}_r(\mathbf{x_k}) \quad (6)$$

Layer 5 This layer contains $M$ nodes. Each node in the output layer determines the value of $y_{km}$

$$y_{km} = \sum_{r=1}^{R} O_{4rm} = \sum_{r=1}^{R}\left(W_{rm}\frac{\beta_r(\mathbf{x_k})}{\sum_{r=1}^{R}\beta_r(\mathbf{x_k})}\right) = \sum_{r=1}^{R} W_{rm}\,\overline{\beta}_r(\mathbf{x_k}) \quad (7)$$

When the above positive and negative fuzzy rule system is applied to image classification, we may take the following steps:

Step 1 Select patterns as the inputs of the proposed fuzzy rule system here from the images with considering the smoothing and textures of the images and normalize the selected patterns

Step 2 Initialize the mean $\mu$ and the variance $\sigma$ of every Gaussian fuzzy membership function in every fuzzy rule

Step 3 Compute $\bar{\beta}_r(x_k)$ using Eq. (5)

Step 4 Compute $W_{rm}$, $\mu$ and $\sigma$ in every Gaussian fuzzy membership function in every fuzzy rule, using the corresponding gradient descending update rules similar to BP algorithm in [6]

Step 5 Repeat Step 4 and Step 5 until the given termination criterion is satisfied. Thus, the fuzzy neural network is trained well

Step 6 Apply the above trained fuzzy neural network to the normalized input patterns to finish the classification task for this image after preprocessing the original image as the normalized input patterns

Because the proposed fuzzy rule system here considers both positive and negative classification information, the above method using step 1 to step 6 can exhibit better classification power than existing methods in image classification. However, when applied to image classification, because we must often choose thousands of patterns from the images, its training performance heavily suffers from the following shortcomings: (1) Because the parameters including $\mu$ and $\sigma$ in every Gaussian fuzzy membership function in every fuzzy rule must be simultaneously adjusted by using the gradient descending method and the parameters $W_{rm}$ must be adjusted by using the least-squares algorithm in each iteration, the corresponding training procedure is generally very slow. (2) The adopted gradient descending method can not assure that it surely converges to the global optimum. (3) Thousands of patterns often raise so-called overfiting issue in training, which results in its poor generalization capability. Our work below will indicate that the extreme learning machine (ELM) can help us circumvent these shortcomings.

## 3 ELM

### 3.1 Single-layer feedforward neural network

Given $N$ arbitrary distinct patterns $(\boldsymbol{x}_k, \boldsymbol{y}_k)$, where $\boldsymbol{x}_k = [x_{k1}, x_{k2}, \ldots, x_{kN}]^T$, $\boldsymbol{y}_k = [y_{k1}, y_{k2}, \ldots, y_{kM}]^T$, the standard single-layer feedforward neural network with $R$ hidden nodes and activation function $f(\boldsymbol{a_r x_k} + b_r)$ is mathematically modeled as

$$O_k = \sum_{r=1}^{R}\mathbf{w}_r f_r(\mathbf{x_k}) = \sum_{r=1}^{R}\mathbf{w}_r f(\mathbf{a}_r\cdot\mathbf{x}_k + b_r), \quad k = (1, 2, \ldots, K) \quad (8)$$

where $\boldsymbol{a_r} = [a_{r1}, a_{r2}, \ldots, a_{rN}]^T$ is the weight vector connecting the $i$th hidden node and the input nodes, $b_r$ is the threshold of the $i$th hidden node, $\boldsymbol{a_r}\cdot\boldsymbol{x_k}$ denotes the inner product of $\boldsymbol{a_r}$ with $\boldsymbol{x_k}$, $\boldsymbol{w_r} = [w_{r1}, w_{r2}, \ldots, w_{rM}]^T$ is the weight vector, connecting the $i$ the hidden node and the output nodes.

The standard single-layer feedforward neural network with $R$ hidden nodes and activation function $f(x)$ can approximate these patterns with zero error. That is to say, there exist $\boldsymbol{a_r}$, $b_r$, $\boldsymbol{w_r}$ such that

$$\mathbf{y}_k = \sum_{r=1}^{R}\mathbf{w}_r f(\mathbf{a}_r\cdot\mathbf{x}_k + b_r) \quad k = 1, 2, \ldots, K \quad (9)$$

The above $R$ equations can be compactly written as

$$\mathbf{HW} = \mathbf{Y} \quad (10)$$

where

$$\mathbf{H}(\mathbf{a}_1,\mathbf{a}_2,\ldots,\mathbf{a}_R,b_1,b_2,\ldots,b_R,\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_K) = \begin{bmatrix} f(\mathbf{a}_1\cdot\mathbf{x}_1+b_1) & f(\mathbf{a}_2\cdot\mathbf{x}_1+b_2) & \ldots & f(\mathbf{a}_R\cdot\mathbf{x}_1+b_R) \\ f(\mathbf{a}_1\cdot\mathbf{x}_2+b_1) & f(\mathbf{a}_2\cdot\mathbf{x}_2+b_2) & \ldots & f(\mathbf{a}_R\cdot\mathbf{x}_2+b_R) \\ \cdots & \cdots & \cdots & \cdots \\ f(\mathbf{a}_1\cdot\mathbf{x}_K+b_1) & f(\mathbf{a}_2\cdot\mathbf{x}_K+b_2) & \ldots & f(\mathbf{a}_R\cdot\mathbf{x}_K+b_R) \end{bmatrix}_{K\times R} \tag{11}$$

$$\mathbf{W} = \begin{bmatrix} \boldsymbol{w}_1^T \\ \boldsymbol{w}_2^T \\ \cdots \\ \boldsymbol{w}_R^T \end{bmatrix}_{R\times M}, \quad \mathbf{Y} = \begin{bmatrix} \boldsymbol{y}_1^T \\ \boldsymbol{y}_2^T \\ \cdots \\ \boldsymbol{y}_K^T \end{bmatrix}_{K\times M} \tag{12}$$

in which $\boldsymbol{H}$ is called the hidden layer output matrix of the neural network here.

According to theorem 2.1 and theorem 2.2 in [1], for any small positive value $\varepsilon$, and activation function $f$ which is infinitely differentiable in any real interval, and for any $\boldsymbol{a}_r$, $b_r$ randomly chosen from any real interval, we have: (1) with probability 1, the hidden layer output matrix $\mathbf{H}$ of the single-layer feedforward neural network with $N$ hidden nodes is invertible and $||\mathbf{HW}-\mathbf{Y}||=0$; (2) with probability 1, there exists the single-layer feedforward neural network with $K$ ($<N$) such that $\|\mathbf{H}_{K\times R}\mathbf{W}_{R\times M}-\mathbf{Y}_{K\times M}\|\le\varepsilon$.

Traditionally, in order to train a single-layer feedforward neural network, one may wish to find the specific $\widehat{\mathbf{a}}_\mathbf{r},\widehat{b}_r,\widehat{\mathbf{W}}$ (r = 1,2,… R) such that

$$||\mathbf{H}(\widehat{\mathbf{a}}_1,\ldots\widehat{\mathbf{a}}_R,\widehat{b}_1,\ldots\widehat{b}_R)\widehat{\mathbf{W}}-\mathbf{Y}|| = \min_{\widehat{\mathbf{a}}_\mathbf{r},\widehat{b}_r,\widehat{\mathbf{W}}} ||\mathbf{H}(\mathbf{a}_1,\ldots\mathbf{a}_R,b_1,\ldots b_R)\widehat{\mathbf{W}}-\mathbf{Y}|| \tag{13}$$

which is equivalent to minimizing the following cost function:

$$E(\Theta)=\sum_{k=1}^{K}\left(\sum_{r=1}^{R}\mathbf{w}_r f(\mathbf{a}_\mathbf{r}\cdot\mathbf{x}_k+b_r)-\mathbf{y}_k\right)^2 \tag{14}$$

where $\Theta=[\mathbf{a}_\mathbf{r},b_r,\mathbf{w}_r]$. When $\boldsymbol{H}$ is unknown, gradient decent based algorithms are often used to find $\Theta=[\mathbf{a}_\mathbf{r},b_r,\mathbf{w}_r]$ such that the minimum of $||\mathbf{HW}-\mathbf{Y}||$is reached. The update rule of $\Theta$ is

$$\Theta^{new}=\Theta^{old}-\eta\frac{\partial E(\Theta)}{\partial\Theta} \tag{15}$$

where $\eta$ is a learning rate.

The popular learning algorithm used in feedforward neural networks is the BP learning algorithm where gradients can be computed efficiently by propagation from the output to the input. There are several shortcomings on BP learning algorithms: (1) If the learning rate $\eta$ is too small, the learning algorithm converges very slowly. However, when $\eta$ is too large, the algorithm becomes unstable and diverges; (2) The algorithm converges very slowly, often at a local minima rather that the global minima, which results in its poor performance; (3) The neural network is often over-trained by the algorithm such that the worse generalization performance is obtained.

In order to circumvent the above shortcomings, Huang et al. proposed the so-called extreme learning machine ELM in [1]. According to the theory of ELM, input weights and hidden layer biases can be randomly assigned and the hidden layer output matrix $\boldsymbol{H}$ can remain unchanged if only the activation function is infinitely differentiable. This is quite different from the most common understanding that all the parameters in the single-layer feedforward neural network need to be adjusted. For the fixed input weights and the hidden layer biases, training the neural network is simply equivalent to finding a least squares solution $\boldsymbol{W}$ of the linear equation, $\boldsymbol{HW}=\boldsymbol{Y}$, i.e.

$$||\mathbf{H}(\widehat{\mathbf{a}}_1,\ldots\widehat{\mathbf{a}}_R,\widehat{b}_1,\ldots\widehat{b}_R)\widehat{\mathbf{W}}-\mathbf{Y}|| = \min_{\widehat{\mathbf{W}}}||\mathbf{H}(\mathbf{a}_1,\ldots\mathbf{a}_R,b_1,\ldots b_R)\widehat{\mathbf{W}}-\mathbf{Y}|| \tag{16}$$

the smallest norm least squares solution of the above linear system is

$$\mathbf{W}=\mathbf{H}^\dagger\mathbf{Y} \tag{17}$$

where $\mathbf{H}^\dagger$ is Moore–Penrose generalized inverse of matrix $\mathbf{H}$ [1]. This solution has the following characteristics:

(1) *Minimal training error*: The special solution in Eq. (17) is only one of the least squares solutions of a general linear system $\mathbf{HW}=\mathbf{Y}$. This special solution means that the minimal training error can be obtained by this special solution:

$$||\mathbf{H}\widehat{\mathbf{W}}-\mathbf{Y}|| = ||\mathbf{H}\mathbf{H}^\dagger\mathbf{Y}-\mathbf{Y}|| = \min_{\widehat{\mathbf{W}}}||\mathbf{H}\mathbf{W}-\mathbf{Y}|| \tag{18}$$

However, most of learning algorithms cannot reach it because of local minimum or infinite training iteration is usually not allowed in applications.

(2) *minimal norm of weights*: This special solution has the minimal norm of weights among all the least squares solutions of $\mathbf{HW}=\mathbf{Y}$:

$$||\widehat{\mathbf{W}}|| = ||\mathbf{H}^\dagger\mathbf{Y}||\le||\mathbf{W}||, \quad \forall\mathbf{W}\in\{\mathbf{W}:||\mathbf{H}\mathbf{W}-\mathbf{Y}||\le||\mathbf{H}z-\mathbf{Y}||,\forall z\in\mathbf{R}^{R\times M}\} \tag{19}$$

As pointed out in [19] by Barlett, for a feedforward neural network, with a small training error, the generalization capability of the neural network depends on all the weight values adopted in the neural network. The smaller all the weight values, the better the generalization capability of the neural network. Because ELM can ensure both the minimal training error and the minimal norm of

weights of the neural network, it can make the neural network have excellent generalization capability.

### 3.2 ELM

**ELM** can be summarized as follows.

Given a training set $\aleph = \{(\boldsymbol{x}_k, \boldsymbol{y}_k) | \boldsymbol{x}_k \in \mathbf{R}^N, \boldsymbol{y}_k \in \mathbf{R}^M, \quad k = 1, 2, \ldots, K\}$, activation function $f(x)$, hidden node number $R$:

Step 1 Randomly assign input weight $\boldsymbol{a}_r$ and bias $b_r$, $r = (1,2,\ldots, R)$

Step 2 Compute the hidden layer output matrix $\mathbf{H}$

Step 3 Compute the output weight $\mathbf{W}$

$$\mathbf{W} = \mathbf{H}^{\dagger}\mathbf{Y} \quad \text{where} \quad \mathbf{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K]^T \tag{20}$$

ELM is a very simple but very effective learning algorithm and has exhibited its thousands of times faster than traditional feedforward network learning algorithms like BP algorithm with better generalization performance. This is because ELM not only tends to reach the smallest training error but also the smallest norm of weights, which is in line with the assertion given by Bartlett that the smaller both the training error and the norm of weights in feedforward neural networks, the better the generalization performance of this neural network.

## 4 Positive and negative fuzzy rule system using ELM for image classification

In order to explain the equivalence between ELM and the positive and negative fuzzy rule system when applied to classification tasks, we first observe their types of outputs. Obviously, ELM has outputs $W_{rm}$, $r = (1,2,\ldots, R)$, $m = (1,2,\ldots, M)$ of real number type. Since the positive and negative fuzzy rule system has the typical form as follows:

*Rule r* IF $x_{1k}$ is $A_{r1}$ and $x_{2k}$ is $A_{r2}$…and $x_{Nk}$ is $A_{rN}$, Then $y_{k1}$ is $C_1$ with $W_{r1}$ and $y_{k2}$ is $C_2$ with $W_{r2}$…and $y_{kM}$ is $C_M$ with $W_{rM}$

where the weight $W_{rm}$ is of real number type. According to Eq. (1), the output of this system is obviously of real number type. Let us consider the typical form of fuzzy rules in the positive fuzzy rule system as follows:

*Rule r*[4–6] IF $\boldsymbol{x}_1$ is $A_{r1}$ and $\mathbf{x}_2$ is $A_{r2}$… and $\boldsymbol{x}_N$ is $A_{rN}$, then $y_1$ is $C_1$ with $W_{r1}$ and $y_2$ is $C_2$ with $W_{r2}$… and $y_M$ is $C_M$ with $W_{rM}$

where the weight $W_{rm}$ is positive. According to Eq.(1), the output of the positive fuzzy rule system is certainly of positive real number type. Similarly, the output of the negative fuzzy rule system is certainly of negative real number type. In other words, only the positive and negative fuzzy rule system rather than the positive or negative fuzzy rule system is of the same output type as ELM.

Next, let us observe the neural network structure of the positive and negative fuzzy rule system. According to Fig. 1, it has the network structure of 5 layers. However, according to Eq. (1), we can easily implement it using a single hidden layer neural network with the activation function in Eq. (5) in the hidden layer to achieve the same output. Obviously, such an activation function in Eq. (5) is infinitely differentiable. With the above analysis about both the output types and the neural network structures of ELM and the positive and negative fuzzy rule system, we can easily conclude that the positive and negative fuzzy rule system can be trained using ELM, and they are equivalent when this fuzzy rule system is applied to classification tasks.

Now, let us design such a positive and negative fuzzy rule system using ELM for image classification. First of all, we suggest that the structure of the positive and negative fuzzy rule system consists of two visible layers (input and output layer) and the only hidden layer as follows:

Layer 1 (input layer) Each node in this layer only transmits input $x_{nk}$, $n = (1,2,\ldots, N)$, $k = (1,2,\ldots, K)$ to the next layer directly. There are totally $N$ nodes in this layer. The output of each node is $O_{1n} = x_{nk}$,

Layer 2 The number of nodes in this layer is equal to the number of fuzzy rules. This layer performs the normalization operation. The output of each node in this layer is represented by:

$$O_{2rk} = \overline{\beta}_r(\boldsymbol{x}_k) = \frac{\beta_r(\boldsymbol{x}_k)}{\sum_{r=1}^{R} \beta_r(\boldsymbol{x}_k)} = \frac{\exp\left[-\sum_{n=1}^{N} \frac{(\boldsymbol{x}_{nk} - \mu_m)^2}{\sigma_m^2}\right]}{\sum_{r=1}^{R} \exp\left[-\sum_{n=1}^{N} \frac{(\boldsymbol{x}_{nk} - \mu_{rn})^2}{\sigma_{rn}^2}\right]} \tag{21}$$

in which $\mu_{rn}$, $\sigma_{rn}$, $r = (1,2,\ldots, R)$, $n = (1,2,\ldots, N)$ are the mean and variance of $\beta_r(\boldsymbol{x}_\mathbf{k})$, respectively.

Layer 3 There are $M$ nodes in this layer in total. In this paper, for pattern $\boldsymbol{x}_k$, the output of proposed fuzzy rule system is determined by winner-takes-all strategy. As a result, when the rule weight $W_{rm}$ has a negative value, it will narrow the choices for class $C_m$ (the more negative value of $W_{rm}$is the smaller value of $y_{km}$). That is

to say, this negative value prescribes actions to be avoided than performed. The value of $W_{rm}$ will be discussed below, i.e., see Eq. (24). Each node in the output layer determines the value of $y_{km}$.

From the output of feedforward neural networks defined in Eq. (1), we can see that the activation function of the proposed fuzzy rule system is:

$$f(\boldsymbol{x}_k) = \overline{\beta}_r(\boldsymbol{x}_k) \tag{22}$$

According to Eq. (10) we know the output matrix corresponding to the hidden layer in the proposed fuzzy rule system is:

$$\begin{aligned}&\mathbf{H}(u_1, u_2, \ldots, u_R, \sigma_1, \sigma_2, \ldots, \sigma_R)\\ &= \begin{bmatrix} \overline{\beta}_1(\boldsymbol{x}_1) & \overline{\beta}_2(\boldsymbol{x}_1) & \ldots & \overline{\beta}_R(\boldsymbol{x}_1) \\ \overline{\beta}_1(\boldsymbol{x}_2) & \overline{\beta}_2(\boldsymbol{x}_2) & \ldots & \overline{\beta}_R(\boldsymbol{x}_2) \\ \ldots & \ldots & \ldots & \ldots \\ \overline{\beta}_1(\boldsymbol{x}_K) & \overline{\beta}_2(\boldsymbol{x}_K) & \ldots & \overline{\beta}_R(\boldsymbol{x}_K) \end{bmatrix}_{K\times R}\end{aligned} \tag{23}$$

The output weight $\boldsymbol{W}$ of the proposed fuzzy rule system can be obtained by using Moore–Penrose generalized inverse in Eq. (17):

$$\mathbf{W} = \mathbf{H}^{\dagger}\mathbf{Y}_q \tag{24}$$

where $\mathbf{Y}_q = [\,\mathbf{y}_{q1} \quad \mathbf{y}_{q2} \quad \cdots \quad \mathbf{y}_{qK}\,]^T_{M\times K}$. The desired output $\boldsymbol{y}_{qk}$ is the form

$$\begin{aligned}\mathbf{y}_{qk} &= \left(y_{qk1}, y_{qk2}, \ldots, y_{qkM}\right)^T\\ &= \begin{cases} (1,0,\ldots 0)^T, & if \quad \mathbf{x}_k \in classC_1 \\ (0,1,\ldots 0)^T, & if \quad \mathbf{x}_k \in classC_2 \\ & \ldots \\ (0,0,\ldots 1)^T, & if \quad \mathbf{x}_k \in classC_{\mathrm{M}} \end{cases}\end{aligned} \tag{25}$$

Because the positive and negative fuzzy rule system we proposed here takes full account of the role of the negative rules in image classification, compared with the traditional fuzzy system based on only positive or negative fuzzy rules, it has obvious advantages. Except for this, we may connect the positive and negative fuzzy rule system with the ELM theory. So the proposed fuzzy rule system can be obviously with ELM features, that is, parameters in the hidden layer can be randomly assigned, fast learning and good generalization ability of the proposed fuzzy rule system can be achieved.

When the proposed fuzzy rule system using ELM is applied to image classification, we can take the following steps:

*Step* 1 Select the training patterns appropriately in the original image, and preprocess these training patterns as the normalized input patterns.

*Step* 2 Assign the mean $\mu$ and the variance $\sigma$ randomly of every Gaussian fuzzy membership function which connects input nodes to hidden nodes of the proposed fuzzy rule system.

*Step* 3 Calculate the output matrix $\mathbf{H}$ of the hidden layer in the proposed fuzzy rule system and the weight matrix $\mathbf{W}$ in form of $\mathbf{W} = \mathbf{H}^{\dagger}\mathbf{Y}_q$.

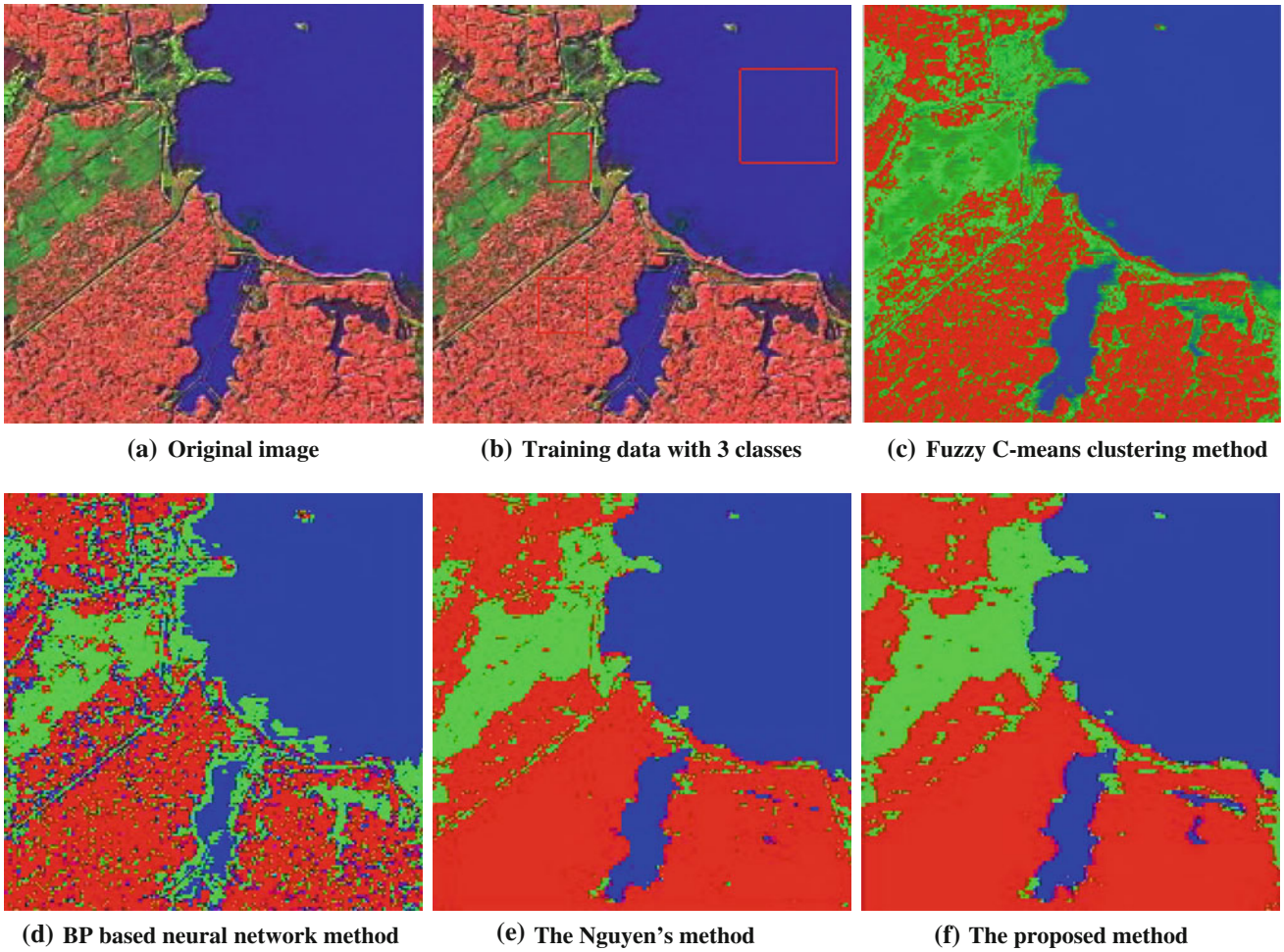
*Step* 4 Apply the above trained fuzzy rule system to the normalized input patterns to finish the classification task for this image, and then restore the pixels of the image by using the classification result for the training patterns of the image.

## 5 Experimental results and discussions

In this section, we will demonstrate the performance of the positive and negative fuzzy rule system using ELM learning for remote sensing and natural image classification. At the same time, we also compare the proposed method here with the fuzzy C means clustering method and the feedforward neural network using BP learning. Experimental platform is Matlab R2009a, AMD Athlon × 2 with 2.0 GHz CPU and 1 GB memory. Experimental results about remote sensing and natural images show that the proposed method is very fast on the training samples, and has obvious image classification superiority over the fuzzy C means clustering method and the BP based neural network method.

### 5.1 On remote sensing image

In this experiment, we applied the proposed method as above to a remote sensing image in [20]. Figure 2a is the original image, whose size is 200 × 200 pixels. The goal is to train the above fuzzy rule system to classify three different terrains in this image, namely *urban*, *vegetation* and *water* areas. The 3,800 training patterns are enclosed in red boxes, as shown in Fig. 2b, where every input pattern takes the form of its R, G, B gray values, its desired output is one of three distinctive classes, i.e., *urban* or *vegetation* or *water* which are represented $[\,1 \quad 0 \quad 0\,], [\,0 \quad 1 \quad 0\,], [\,0 \quad 0 \quad 1\,]$, respectively. All these patterns are used to train such that the squared sum of errors of the outputs of the above fuzzy rule system is minimized. In this experiment, we adopted ELM for the fuzzy rule system with 3 inputs ($N = 3$), 40 rules ($R = 40$) and 3 classes ($M = 3$), and the BP based neural network with the 3-40-3 structure, i.e., the input layer with 3 inputs, the hidden layer with 40 neurons taking the *tanh sigmoid* functions as their activation functions and the output layer with 3 neurons The Nguyen's fuzzy system (i.e., the Nguyen's method here for brevity) in [2] is the structure with 3 inputs, 3 outputs and 40 rules. After training, the trained fuzzy rule system and BP based neural network were then



(a) Original image (b) Training data with 3 classes (c) Fuzzy C-means clustering method

(d) BP based neural network method (e) The Nguyen's method (f) The proposed method

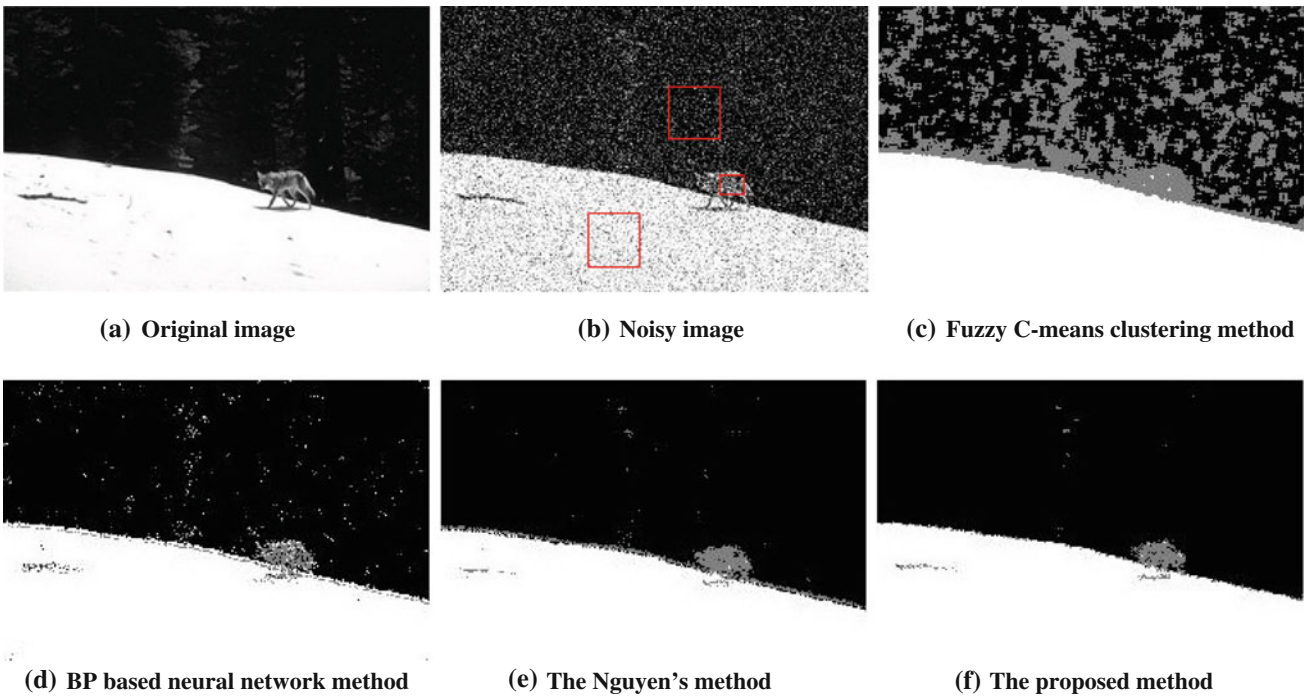
**Fig. 2** Classification results for remote sensing image

used to classify the entire original image. We also carried out the fuzzy C-means clustering method to classify this image. Both the BP based neural network and the fuzzy C-means method are executed on Matlab R2009a with default parameters.

In order to make our experiment fair, we run the corresponding proposed method ten times for the above fuzzy system. It spent the average 0.2250 s CPU time with MSE = 0.0026 to finish the training task for this image, however, The BP based neural network method spent 148.7203 s CPU time with MSE = 0.0473, the Nguyen's method [2] takes 198.0703 s CPU time with MSE = 0.0045. In other words, with the smaller MSE, the propose methods runs 660 times faster than the BP based neural network method. The testing time of the proposed method is 0.3047 s CPU time, while the testing time of BP based neural network is 0.4047 and the testing time of Nguyen's method is 4.1485 s CPU time. So the testing time of the proposed method is 75.29% of the BP based neural network's. The clustering time of the fuzzy C means clustering (FCM) method requires 1.9500 s CPU time which is much longer than the proposed method. Except for this, we can see from Fig. 2 that the classification accuracy of the proposed method is much better than that of the fuzzy C means clustering method. Therefore, compared with the BP based neural network method, the proposed method has better classification accuracy and robustness.

### 5.2 On natural images

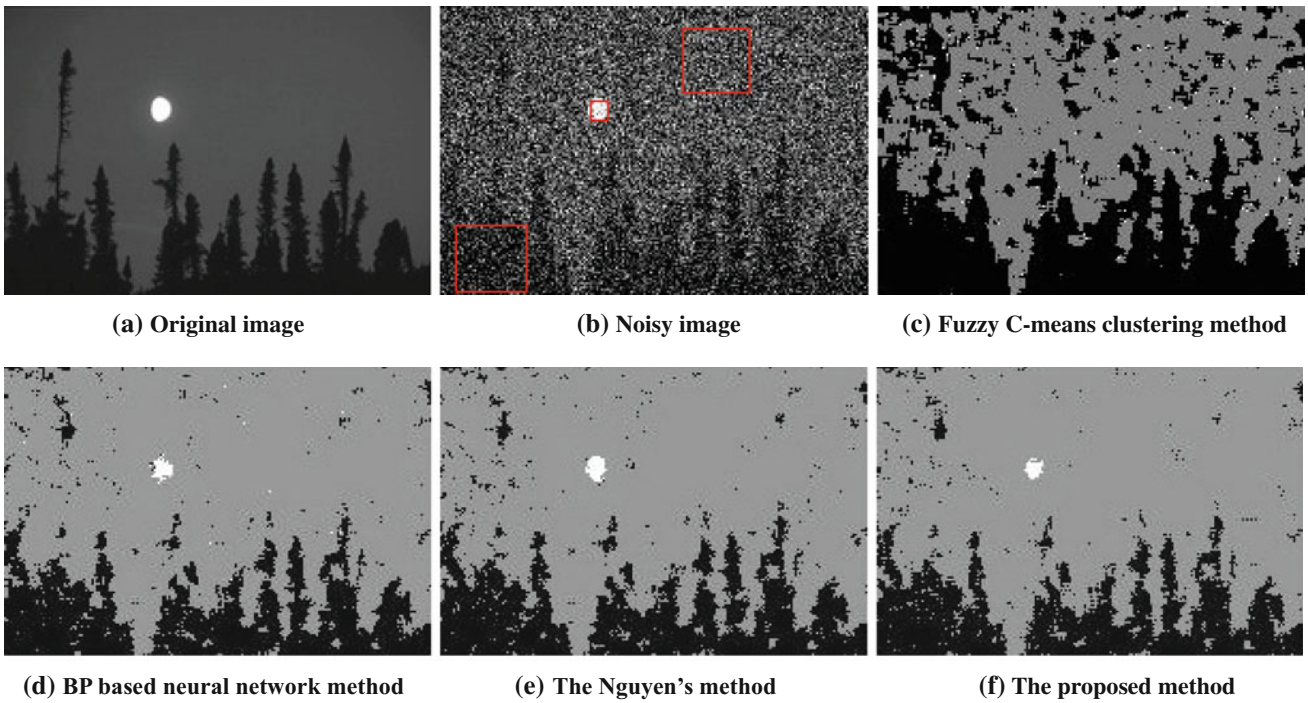
In this experiment, we applied the proposed method to natural images [21] with noise to examine its fast training capability and robustness. The original images in Figs. 3a and 4a are taken from the Berkeley database (http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench), and the original image in Fig. 5a is taken from the literature [22]. Their sizes are 200 × 165, 200 × 135, 150 × 150 pixels respectively. Their corresponding noisy images corrupted by Gaussian noise (0 mean, 0.05 variance) are shown in Figs. 3b, 4b and 5b. In order to suppress noise existing in these images, we used the 5 × 5 pixel window to generate the corresponding input patterns from these images.



(a) Original image (b) Noisy image (c) Fuzzy C-means clustering method

(d) BP based neural network method (e) The Nguyen's method (f) The proposed method

**Fig. 3** Classification results for the natural image

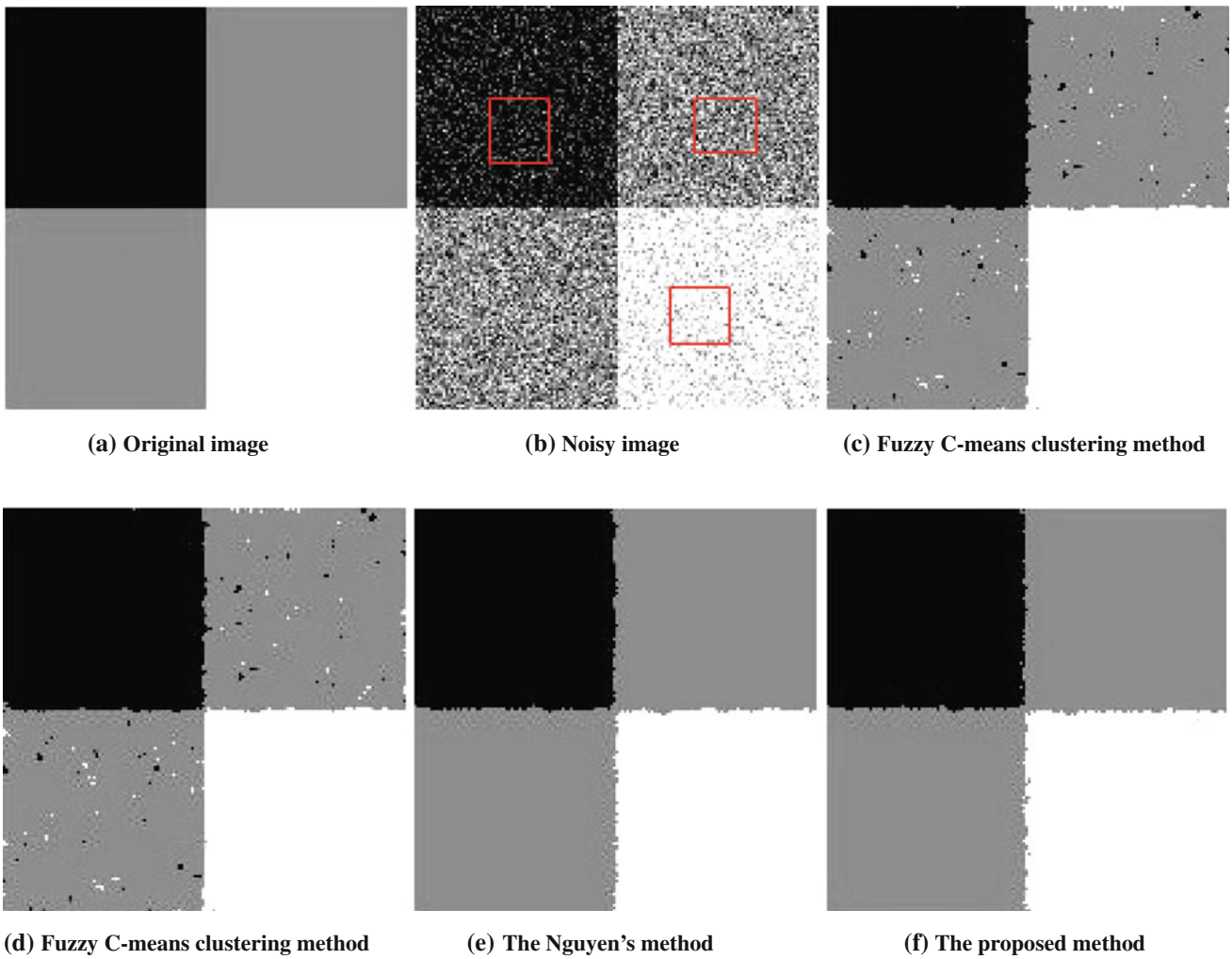


(a) Original image (b) Noisy image (c) Fuzzy C-means clustering method

(d) BP based neural network method (e) The Nguyen's method (f) The proposed method

**Fig. 4** Classification results for the natural image

The 3,300, 2,700 and 1,200 training patterns were taken from the red regions in Figs. 3b, 4b, and 5b, respectively. The desired output of every pattern is one of three classes, i.e., *snow, wolf, tree* for the image in Fig. 3b, *moon, sky, tree* for the image in Fig. 4b, and *white, grey, black* for the image Fig. 5b. For the images in Figs. 3 and 4, we designed the proposed fuzzy system with 25 inputs ($N = 25$), 100 fuzzy rules ($R = 100$) and 3 outputs [i.e., 3 classes ($M = 3$)] while for the image in Fig. 5, the proposed fuzzy system with 25 inputs ($N = 25$), 60 fuzzy rules ($R = 60$), and 3 outputs ($M = 3$). The corresponding classification results for these three images are illustrated in Figs. 2e, 3e and 4e. After



(a) Original image (b) Noisy image (c) Fuzzy C-means clustering method

(d) Fuzzy C-means clustering method (e) The Nguyen's method (f) The proposed method

**Fig. 5** Classification results for the natural image

training, for each window, we can feed it into the trained fuzzy rule system using the proposed method here to decide which class the center pixel of this $5 \times 5$ window should belong to.

We reported the classification results obtained by the fuzzy C-means clustering method and BP based neural network method, see Figs. 3c, 4c and 5c and Figs. 3d, 4d and 5d, respectively, where the fuzzy C-means clustering method is the same as in the last experiment and the BP based neural network method and The Nguyen's method takes the same activation function as in the last experiment with the 25-100-3 structure for Figs. 3b and 5b, and the 25-60-3 structure for Fig. 5b. Obviously, we can see from these noisy images that the proposed method obtained the best classification accuracy among all three methods, that is to say, the proposed method has strong classification capability and robustness.

Next, let us report the training time of the proposed method and the BP based neural network method and the fuzzy C-means clustering method. Ten trials have been conducted for all algorithms and their average performance are reported here. For the image Fig. 3b, the proposed method spent 0.6141 s CPU time with MSE = 0.0779, however, the BP based neural network method spent 218.3860 s CPU time with MSE = 0.1471, the Nguyen's method takes 485.7031 s CPU time with MSE = 0.0095. The testing time of ELM algorithm for the proposed fuzzy rule system is 0.9641 s CPU time, while the testing time of BP based neural network is 1.0813 s CPU time and the testing time of Nguyen's method is 6.4531 s CPU time. We also compare the performance of the FCM clustering method and the proposed method. The FCM clustering time for this image requires 6.2125 s CPU time which is much longer than the proposed method. With the smaller MSE, the proposed method runs 355 times faster than the BP based neural network method. For the image Fig. 4b, the proposed method spent 0.4563 s CPU time with MSE = 0.1777, however, the BP based neural network

method spent 185.9422 s CPU time with MSE = 0.2046 the Nguyen's method takes 398.3258 s CPU time with MSE = 0.0617. The testing time of the proposed method is 0.6234 s CPU time, while the testing time of BP based neural network is 0.7312 s CPU time and the testing time of Nguyen's method is 4.2344 s CPU time. The FCM clustering method requires 2.1641 s CPU time which is much longer than the proposed method. For the image Fig. 5b, the proposed method spent 0.2625 s CPU time with MSE = 0.0860, however, the BP based neural network method spent 96.9000 s CPU time with MSE = 0.1477 the Nguyen's method takes 278.9844 s CPU time with MSE = 0.0268. The testing time of the proposed method is 0.5516 s CPU time, while the testing time of BP based neural network is 0.6141 s CPU time and the testing time of Nguyen's method is 3.4844 s CPU time. The FCM clustering method for this image requires 3.8406 s CPU time which is much longer than the proposed method. Therefore, with the smaller MSE, the proposed method runs much faster than the BP based neural network method and FCM algorithm for these noisy images.

## 6 Conclusions

The positive and negative fuzzy rule system used in this paper could effectively use the negative information existing in image classification. We proved that such a fuzzy rule system can be equivalently implemented by using ELM. Accordingly, we proposed the image classification method based on the positive and negative fuzzy rule system using ELM. Experimental results showed that the proposed method can achieve better results in the learning speed and classification accuracy and robustness for image classification tasks, compared with the BP based neural network method and the FCM clustering method.

As we may know well, when applied to large datasets, ELM is still ineffective, due to the complicated calculation of the inverse of the output matrix. Therefore, further research includes how to extend the proposed method to large scale image classification tasks. This is an on-going work we are doing.

**Acknowledgments** This work was supported in part by the Hong Kong Polytechnic University under Grant 1-ZV5V, by the National Natural Science Foundation of China under Grants 60903100, 60975027 and 90820002, and by the Natural Science Foundation of Jiangsu province under Grant BK2009067.

## References

1. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3): 489–501
2. Nguyen TM, Wu J-QM (2008) A combination of positive and negative fuzzy rules for image classification problem. In: Proceedings of the 2008 seventh international conference on machine learning and applications, pp 741–746
3. Tang Y, Yan P, Yuan Y et al (2011) Single-image super-resolution via local learning. Int J Mach Learn Cybern 2(1):15–23
4. Kang S, Park S (2009) A fusion neural network classifier for image classification. Pattern Recogn Lett 30(9):789–793
5. Tzeng YC, Chen KS (1998) A fuzzy neural network to SAR image classification. IEEE Trans Geosci Remote Sens 36:301–307
6. Zhou W-Y (1999) Verification of the nonparametric characteristics of backpropagation neural networks for image classification. IEEE Trans Geosci Remote Sens 37(1):771–779
7. Nakashima T, Schaefer G, Yokota Y, Ishibuchi H (2007) A weighted fuzzy classifier and its application to image processing tasks. Fuzzy Sets Syst 158:284–294
8. de Moraes RM, Banon GJF, Sandri SA (2002) Fuzzy expert systems architecture for image classification using mathematical morphology operators. Inf Sci 142(1):7–21
9. Liu DM, Wang ZX (2008) A united classification system of X-ray image based on fuzzy rule and neural networks. In: 3rd international conference on intelligent system and knowledge engineering, ISKE 2008, pp 1717–1722
10. Mandai DP, Murthy CA, Pal SK (1992) Formulation of a multivalued recognition system. IEEE Trans Syst Man Cybern 22(4):607–620
11. Pal S, Mandai DP (1992) Linguistic recognition system based on approximate reasoning. Inf Sci 61:135–161
12. Ishibuchi H, Yamamoto T (2005) Rule weight specification in fuzzy rule-based classification systems. IEEE Trans Fuzzy Syst 13(4):428–435
13. Tong DL, Mintram R (2010) Genetic Algorithm-Neural Network (GANN): a study of neural network activation functions and depth of genetic algorithm search applied to feature selection. Int J Mach Learn Cybern 1(1–4):75–87
14. Yi W, Lu M, Liu Z (2011) Multi-valued attribute and multi-labeled data decision tree algorithm. Int J Mach Learn Cybern. doi:10.1007/s13042-011-0015-2
15. Yu SW, Zhu KJ, Diao FQ (2008) A dynamic all parameters adaptive BP neural networks model and its application on oil reservoir prediction. Appl Math Comput 195:66–75
16. Branson JS, Lilly JH (1999) Incorporation of negative rules into fuzzy inference systems. In: Proceedings of the 38th IEEE Conference on Decision and Control, vol 5, pp 5283–5288
17. Lilly JH (2007) Evolution of a negative-rule fuzzy obstacle avoidance controller for an autonomous vehicle. IEEE Trans Fuzzy Syst 15(4):718–728
18. Li Y, Deng J-M, Wei M-Y (2002) Meaning and precision of adaptive fuzzy systems with Gaussian-type membership functions. Fuzzy Sets Syst 127:85–97
19. Bartlett PL (1996) For valid generalization, the size of weights is more important than the size of networks. Adv Neural Inform Process Syst 9(1):134–140
20. Mitra P, Shankar BU, Pall SK (2004) Segmentation of multispectral remote sensing images using active support vec tor machines. Pattern Recogn Lett 25:1067–1074
21. Fu Y, Wang Y-W, Wang W-Q, Gao W (2003) Content-based natural image classification and retrieval using SVM. Chin J Comput 26(10):1261–1265
22. Guo Y-H, Cheng D (2009) New neutrosophic approach to image segmentation. Pattern Recogn 42(5):587–595