Soft Computing (2019) 23:4969–4985
https://doi.org/10.1007/s00500-018-3161-6

METHODOLOGIES AND APPLICATION

Design of face recognition system based on fuzzy transform and radial basis function neural networks

Seok-Beom Roh[1] · Sung-Kwun Oh[1] · Jin-Hee Yoon[2] · Kisung Seo[3]

Published online: 9 April 2018

Abstract
In this study, a face recognition method based on fuzzy transform and radial basis function neural networks is proposed. In order to reduce the dimensionality and extract the important features of face images, fuzzy transform with fuzzy partition techniques is used. Fuzzy radial basis function neural networks (FRBFNNs) are used as a classifier to identify face images into several categories. Radial basis functions are defined by fuzzy C-means clustering method which can analyze the distribution of data points over the input spaces. In order to validate the proposed face recognition system, experimental comparative studies are conducted on the benchmark face datasets such as YALE, ORL, and ABERDEEN databases. A comparative analysis demonstrates that the proposed face recognition system is superior to the conventional face recognition techniques.

Keywords Fuzzy C-means clustering (FCM clustering) · Fuzzy transform (F-transform) · Fuzzy radial basis function neural networks (FRBFNNs) · Preprocessing technique

1 Introduction

Face recognition has been a key issue in the pattern recognition and computer vision research fields (Liu et al. 2016). The various methods to recognize faces such as eigenface (Turk and Pentland 1991) and fisherface (Belhumeur et al. 1997) have been introduced in the past decade and applied to various application fields such as the person identification for the mobile payment and video surveillance. It is well known that the face data are usually described over high-dimensional space although the available training samples are limited (Pang et al. 2015). The high dimensionality of the given data may result in the poor classification performance (Fukunnaga 1990). In order to deal with this difficulty, a lot of approaches have been introduced. These approaches can be categorized into two categories such as "holistic approaches and local feature/component-based approaches" (Bansal et al. 2012).

In general, the representative approaches of the holistic features include principal component analysis (PCA) (Turk and Pentland 1991), linear discriminant analysis (LDA) (Belhumeur et al. 1997), and independent component analysis (ICA) (Bartlett et al. 2002).

While PCA is based on the unsupervised learning scheme, LDA is a kind of supervised learning method. In other words, PCA projects data samples onto specific linear directions with maximal variance. PCA using the unsupervised learning scheme may be unsuitable for the classification problem. Unlike PCA, LDA minimizes the predefined objective function based on class-specific information such as within-class scatter matrix and between-class scatter matrix. Although LDA is a kind of the supervised learning method, which can consider the class label of each datum, there is a limitation that the number of the extracted features by using LDA is lower than the number of classes (Huang et al. 2015).

Unlike the holistic feature extraction approaches, local feature representation methods include Gabor (Liu and Wechsler 2002; Cament et al. 2015), wavelets (Farokhi et al. 2014), and local binary pattern (LBP) (Ahonen et al. 2006; Liu et al. 2016). Although LBP approach has the potential classification ability, the robustness and illumination prob-

Communicated by V. Loia.

✉ Sung-Kwun Oh
ohsk@suwon.ac.kr

[1] Department of Electrical Engineering, The University of Suwon, 17 Wauan-gil, Bongdam-eup, Hwaseong-si, Gyeonggi-do 18323, South Korea

[2] School of Mathematics and Statistics, Sejong University, Neungdongro 209, Gwangjin-gu, Seoul 05006, South Korea

[3] Department of Electronic Engineering, Seokyeong University, Jungneung-Dong 16-1, Sungbuk-gu, Seoul 02713, South Korea

lems of LBP approach limit the application fields. In other words, the codes decided by LBP algorithm may be affected by some noise easily (Chai et al. 2014).

In the case of Gabor feature-based approaches, they have achieved high performance. However, Gabor-based filters have suffered from the curse of dimension due to the convolution operation (Binsaadoon and El-Alfy 2015).

Especially when the face recognition approaches are considered, the dimension of the input space is very high and the face image is easily contaminated by the variation of the illumination.

Therefore, the feature extraction approaches, which are robust to the illumination noise and can deal with the high dimension of the input space, should be necessary.

Several filters based on fuzzy techniques such as the fuzzy inference ruled by else-action (FIRE) filter, weighted fuzzy mean filter, and the iterative fuzzy control-based filter have been developed (De Ville et al. 2003).

In order to improve the recognition performance, we focus on the feature extraction technique which can reduce the impulse noise and deal with the high dimension of the input space.

In this study, we introduce a new feature extraction method by using fuzzy transform (F-transform) which is very useful in many applications such as image processing, image compression, and time series prediction (Perfilieva 2010; Martino et al. 2014; Perfilieva et al. 2012; Novak et al. 2014). F-transform, which was proposed by Perfilieva (2006), is a technique that performs a weighted orthogonal projection of a function into a finite number of real numbers.

In this study, F-transform is used to transform an image into a predetermined number of dimensional spaces. In other words, the dimension of an image can be reduced by using F-transform.

F-transform-based feature extraction technique can be considered in two aspects such as the noise reduction approach and the dimension reduction technique.

When the F-transform and the inverse F-transform are conducted iteratively, the impulse noise involved in images can be reduced.

Meanwhile, F-transform can extract the good features from the high-dimensional input data by using the local analysis.

When it comes to classifiers to improve classification rate, fuzzy radial basis function neural networks (FRBFNNs) are used as a classifier whose inputs are the extracted feature from an image by using F-transform. FRBFNN is a sort of neuro-fuzzy techniques, which have been applied to the various fields (Petkovic et al. 2016, 2017; Nikolic et al. 2017). Since FRBFNNs accommodate fuzzy rules and fuzzy inference algorithm in the architecture of neural networks (Huang and Oh 2017), the FRBFNNs can show their advantages such as structural simplicity, local approximation capability, and especially function equivalence with a simple fuzzy inference system (Li and Hori 2006). It is well known that FRBFNNs exhibit the advantages of the technologies of fuzzy sets and neural networks.

This study is organized as follows. In Sect. 2, we review fuzzy transform and the inverse fuzzy transform in one and two variables. Next, in Sect. 3, the feature extraction technique based on fuzzy transform is explained. In Sect. 4, a concept of granulation of information is revisited and the architecture of the general RBFNNs and fuzzy RBFNNs is discussed. Extensive experimental studies to validate the proposed face recognition technique are covered in Sect. 5, while Sect. 6 offers some concluding comments.

2 Fuzzy transform of functions in one variable and two variables

In this study, we use fuzzy transform (F-transform) to reduce and extract features from an image. We recall the basic definition and properties of the F-transform (see Perfilieva 2006)

2.1 Fuzzy transform in one variable

Let $x_1 < x_2 < \cdots < X_n$ be fixed nodes within $[a, b]$, such that $x_1 = a$, $x_n = b$, and $n \geq 2$, we say that fuzzy sets $\boldsymbol{A} = \{A_1, A_2, \ldots, A_n\}$, identified by their membership functions $A_1(x), A_2(x), \ldots, A_n(x)$ defined on $[a, b]$, form a fuzzy partition of $[a, b]$ if they fulfill the following conditions for $k = 1, \ldots, n$

(1) $A_k : [a, b] \to [0, 1],\ A_k(x_k) = 1$
(2) $A_k(x) = 0,\ if\ x \in (x_{k-1}, x_{k+1})$
(3) $A_k(x)$ is continuous
(4) $A_k(x),\ k = 2, \ldots, n$, strictly increases on $\left[x_{k-1}\, x_k\right]$, and $A_k,\ k = 1, \ldots, n-1$, strictly decreases on $\left[x_k, x_{k+1}\right]$
(5) For all $x \in [a, b],\ \sum_{k=1}^{n} A_k(x) = 1$

These fuzzy sets $\{A_1(x), A_2(x), \ldots, A_n(x)\}$ are called basis function. Moreover, we say that they form a uniform fuzzy partition if the following conditions are satisfied

(6) $n \geq 3$ and $x_i = a + h \cdot (i-1)$, where $h = (b-a)/(n-1)$ and $i = 1, 2, \ldots, n$ (that is the nodes are equidistant)
(7) $A_i(x_i - x) = A_i(x_i + x)$ for every $x \in [0, h]$ and $i = 2, \ldots, n-1$
(8) $A_{i+1}(x) = A_i(x-h)$ for every $x \in \left[x_i, x_{i+1}\right]$ and $i = 1, 2, \ldots, n-1$

Let $A_1, A_2, \ldots, A_n$ be basic functions which form a fuzzy partition of $[a, b]$ and f be any function from $C([a, b])$.

We say that the n-tuple of real number $[F_1, F_2, \ldots, F_n]$ is given by

$$F_k = \frac{\int_a^b f(x) A_k(x)\,dx}{\int_a^b A_k(x)\,dx}, \quad k = 1, \ldots, n. \tag{1}$$

That is the (integral) F-transform of f with respect to $A_1, A_2, \ldots, A_n$.

The discrete version of F-transform is defined as follows:

Let a function f be given at nodes $x_1, x_2, \ldots, x_l \in [a, b]$ and $A_1, A_2, \ldots, A_n$, $n < l$ be basic functions which form a fuzzy partition of $[a, b]$. We say that the n-tuple of real numbers $[F_1, F_2, \ldots, F_n]$ is the discrete F-transform of f with respect to $A_1, A_2, \ldots, A_n$ if

$$F_k = \frac{\sum_{j=1}^{l} f(x_j) A_k(x_j)}{\sum_{j=1}^{l} A_k(x_j)}, \quad k = 1, \ldots, n. \tag{2}$$

On the basis of the F-transform values, we can define the inverse F-transform of f with respect to $\{A_1, A_2, \ldots, A_n\}$ as follows:

$$f_{F,n}(x) = \sum_{i=1}^{n} F_i A_i(x). \tag{3}$$

It approximates a given continuous function f on $[a, b]$ with arbitrary precision (Perfilieva 2006).

2.2 Fuzzy transform in two variables

In order to deal with the image datasets, which are two-dimensional data, we should extend the F-transform of one variable to two variables. The universe of discourse is assumed to be the rectangle $[a, b] \times [c, d]$ and let $n, m \geq 2$, $x_1, x_2, \ldots, x_n \in [a, b]$ and $y_1, y_2, \ldots, y_n \in [c, d]$ be $n + m$ assigned points, called nodes, such that $x_1 = a < x_2 < \cdots < x_n = b$ and $y_1 = c < y_2 < \cdots < y_m = d$. In addition, let $A_1, \ldots, A_n : [a, b] \to [0, 1]$ be a fuzzy partition of $[a, b]$, $B_1, B_2, \ldots, B_n : [c, d] \to [0, 1]$ be a fuzzy partition of $[c, d]$ and $f(x, y)$ be a continuous function on $[a, b] \times [c, d]$. Then we can define the $n \times m$ matrix $[F_{kl}]$ as the F-transform of f with respect to $\{A_1, A_2, \ldots, A_n\}$ and $\{B_1, B_2, \ldots, B_m\}$ if we have for each $k = 1, \ldots, n$ and $l = 1, \ldots, m$, $x \in [a, b]$ and $y \in [c, d]$:

$$F_{kl} = \frac{\int_c^d \int_a^b f(x, y) A_k(x) B_l(y)\,dx\,dy}{\int_c^d \int_a^b A_k(x) B_l(y)\,dx\,dy}. \tag{4}$$

In the discrete version of F-transform, we assume that the function f would be defined at some points $(p_j, q_j) \in [a, b] \times [c, d]$, where $i = 1, \ldots, N$ and $j = 1, \ldots, M$. In addition, the sets $P = \{p_1, p_2, \ldots, p_N\}$ and $Q = \{q_1, q_2, \ldots, q_M\}$ of these nodes are sufficiently dense with respect to the chosen partitions, i.e., for each $i = 1, 2, \ldots, N$ there exists an index $k \in \{1, 2, \ldots, n\}$ such that $A_k(p_i) > 0$ and for each $j = 1, 2, \ldots, M$ there exists an index $l \in \{1, 2, \ldots, m\}$ such that $B_l(q_j) > 0$. In this case, we define the discrete version of F-transform of two variables as follows.

$$F_{kl} = \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} f(p_i, q_j) A_k(p_i) B_l(q_j)}{\sum_{j=1}^{M} \sum_{i=1}^{N} A_k(p_i) B_l(q_j)}. \tag{5}$$

Similarly, as the inverse F-transform of one variable, we can define the discrete inverse F-transform in two variables of f with respect to $\{A_1, A_2, \ldots, A_n\}$ and $\{B_1, B_2, \ldots, B_m\}$ to be the following function defined in the same points $(p_i, q_j) \in [a, b] \times [c, d]$ with $i \in \{1, 2, \ldots, N\}$ and $j \in \{1, 2, \ldots, M\}$ as follows.

$$f_{nm}^{F}(p_i, q_j) = \sum_{k=1}^{n} \sum_{l=1}^{m} F_{kl} A_k(p_i) B_l(q_j). \tag{6}$$

3 Feature extraction by using fuzzy transform

It is well known that the learning techniques used by the classifier were limited to low-dimensional spaces with easily separable classes (Lecun et al. 1998). Many researchers have been studying how to reduce the dimensionality of the original dataset and obtain good features from those. Since F-transform has been proposed by Perfilieva in 2006, the F-transform has been applied to many applications such as image and signal processing, image compression, and time series prediction (Holcapek and Tichy 2011; Novak et al. 2010; Perfilieva et al. 2016). In this study, fuzzy transform (F-transform) with a fuzzy partition is used to reduce the dimensionality of an image and extract features from the original image.

We can encode or compress an image by using F-transform and decode or reconstruct the encoded (or compressed) image with the original image size. By using the encoding property of F-transform, we can extract the features from the original image.

Let assume that an image I of size $N \times M$ pixels is represented by a function of two variables $f_I = \mathbb{N} \times \mathbb{N} \to [0, 1]$ defined at every point $(i, j) \in [1, N] \times [1, M]$ of lattice. The value $f_I(i, j)$ represents the image intensity of each pixel.

The matrix $\mathbf{F}_{nm}\left[f_I\right]$ of the discrete F-transform of f_I is defined as follows.

$$\mathbf{F}_{nm}\left[f_I\right] = \begin{bmatrix} F_{11} & \cdots & F_{1m} \\ \vdots & \vdots & \vdots \\ F_{n1} & \cdots & F_{nm} \end{bmatrix}. \tag{7}$$

where

$$F_{kl} = \frac{\sum_{j=1}^{M}\sum_{i=1}^{N} f\left(i,\ j\right) A_k\left(i\right) B_l\left(j\right)}{\sum_{j=1}^{M}\sum_{i=1}^{N} A_k\left(i\right) B_l\left(j\right)},$$
$$k = 1,\ \ldots,\ n;\, l = 1,\ \ldots,\ m.$$

The basic functions $A_1,\ \ldots,\ A_n$ and $B_1,\ \ldots,\ B_m$ are defined as fuzzy sets which constitute fuzzy partitions of $[1,\ N]$ and $[1,\ M]$, respectively.

The inverse F-transform is used to reconstruct (or decode) the images compressed (or encoded) by using the discrete F-transform. The reconstructed image can be described as (8).

$$f_{nm}^{F}\left(i,\ j\right) = \sum_{k=1}^{n}\sum_{l=1}^{m} F_{kl} A_k\left(i\right) B_l\left(j\right). \tag{8}$$

We focus on the face recognition to identify "who the person is" from the image. Let us consider the image of size 112×92 with intensity of pixels between 0 and 255 described in Fig. 1.

The compressed images of the original images are obtained by using F-transform as shown in Fig. 2. The size of the matrix $\mathbf{F}_{nm}$, which describes the compressed images, is 21×15.

The reconstructed images can be formed by calculating the inverse F-transform based on F-transform of the original images. The reconstructed images are shown in Fig. 3.

Fig. 1 Original images

Fig. 2 Compressed images

Fig. 3 Reconstructed images

Fig. 4 Original image

4 Noise cancelation of corrupted image by F-transform and inverse F-transform

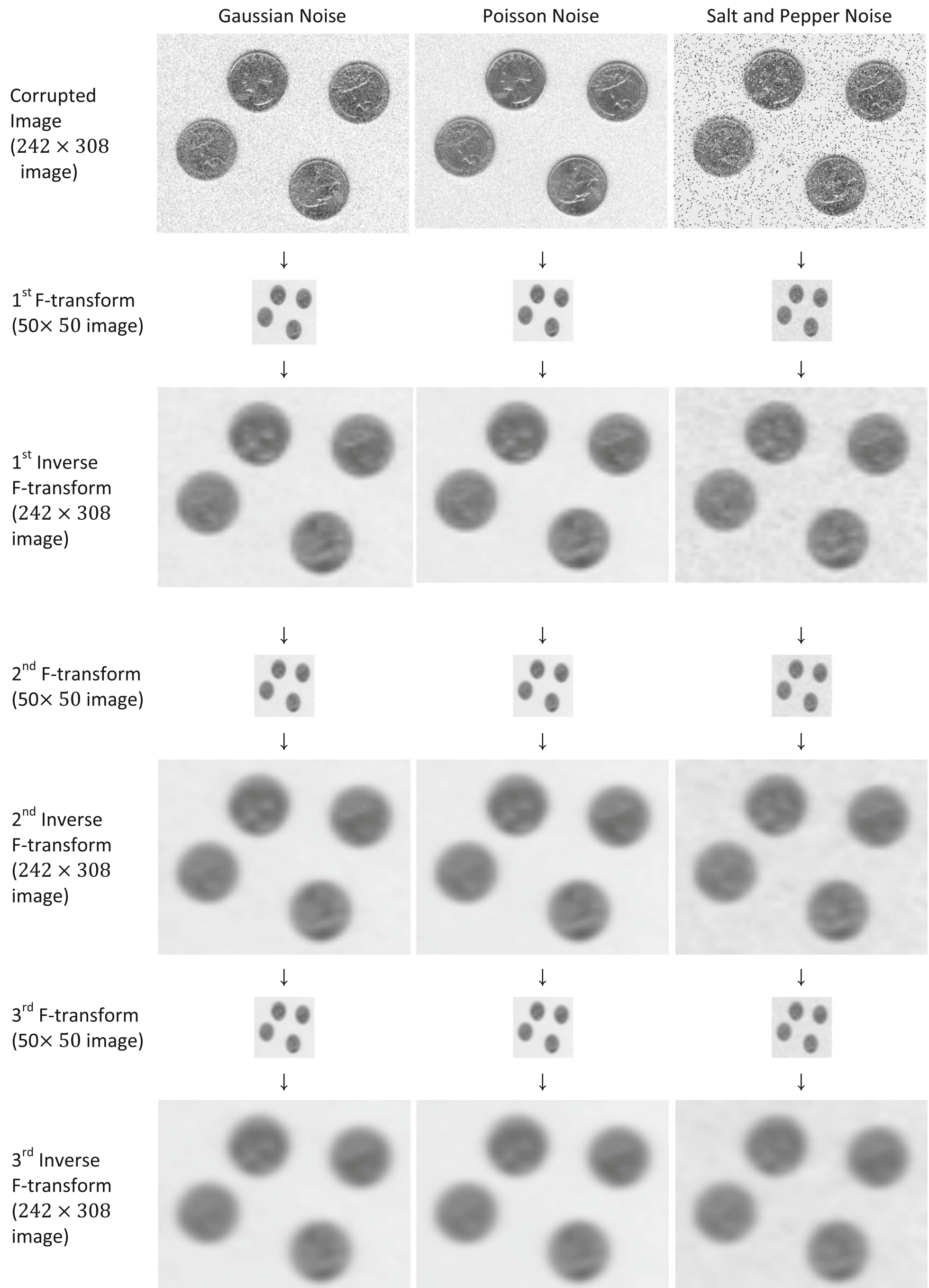
It is known that the inverse F-transform approximates the original function with an arbitrary precision (Perfilieva 2006). From this characteristic of F-transform and the inverse F-transform, one can remove noise from the corrupted image. We try to reconstruct the image corrupted by noise using the pair operations such as F-transform and the inverse F-transform.

The considered original image of size 242×308 is shown in Fig. 4.

We corrupt the original image shown in Fig. 4 by using the built-in function of MATLAB. The three kinds of noise such as Gaussian noise, Poisson noise, and salt and pepper noise are used to corrupt the image. To remove the noise from the corrupted image, F-transform and the inverse F-transform are used iteratively. The fuzzy partition for F-transform is 50×50 triangular membership functions over image space. In the case of Gaussian noise corrupted image, the noise cancelation procedure is shown in Fig. 5.

The reconstructed images by using F-transform and the inverse F-transform iteratively are applied to the feature extraction stage. F-transform is used in the stage of the feature extraction.

Fig. 5 Noise cancelation procedure of corrupted image



In Fig. 6, the feature extraction procedure is described which is based on F-transform and the inverse F-transform iteratively.

Some features extracted through the above-explained iterative F-transform and the inverse F-transform are applied to a classifier. In this study, radial basis function neural networks are used as a classifier to identify the given image.

5 Radial basis function neural networks classifier

It is known from several studies (Belhumeur et al. 1997; Pang et al. 2015; Novak et al. 2014; Samaria and Harter 1994; Bartlett et al. 2002) that the generic RBFNNs exhibit some advantages including global optimal approximation and classification capabilities as well as rapid convergence

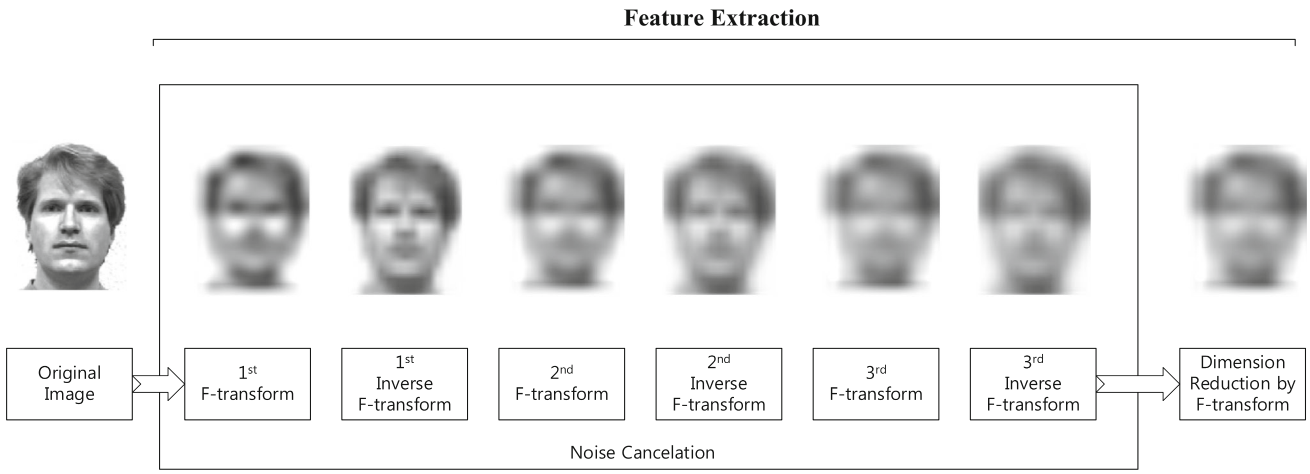


Fig. 6 Feature extraction based on iterative F-transform and the inverse F-transform

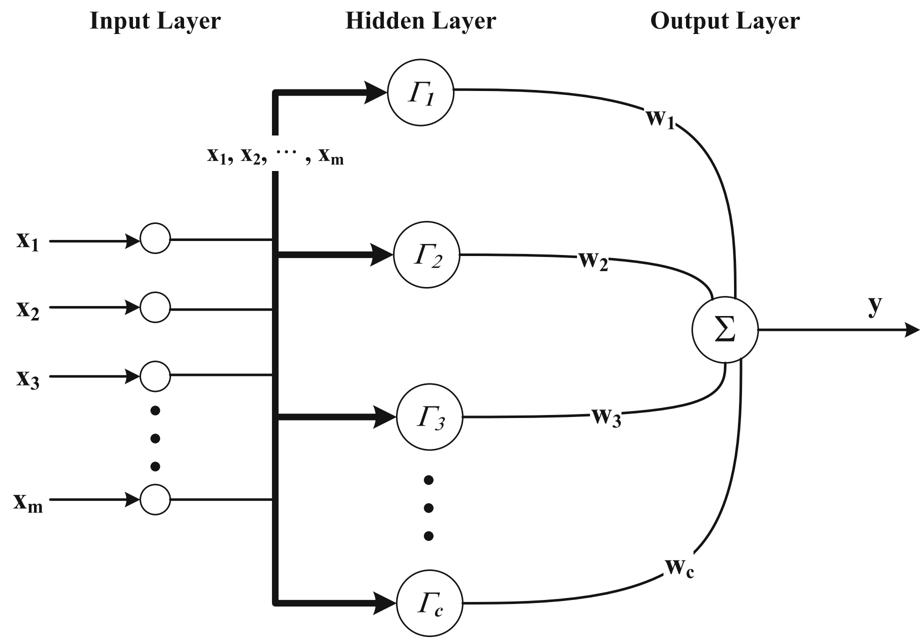


Fig. 7 General architecture of the generic RBF neural networks

of the underlying learning procedures. The generic topology of RBFNNs is depicted in Fig. 7.

In Fig. 7, Γ_i, $i = 1, 2, \ldots, c$, denotes receptive fields (radial basis functions), while "m" stands for the number of the input variables. The output of the generic RBFNN comes as a linear combination of the outputs ($\Gamma(\mathbf{x})$) of the hidden layer with the connection weights $w_1, w_2, \ldots, w_c$ as shown below

$$\hat{y}(\boldsymbol{x}) = \sum_{i=1}^{c} w_i \cdot \Gamma_i(\boldsymbol{x}), \tag{9}$$

where $\boldsymbol{x} = \left[x_1 \cdots x_m\right] \in \Re^m$ and $\Gamma_i(\boldsymbol{x})$ is the activation level of the ith node present at the hidden layer.

Generally, the Gaussian-type RBFs are used as receptive fields

$$\Gamma_i(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{v}_i\|^2}{2\sigma_i^2}\right) \tag{10}$$

where $\boldsymbol{v}_i$ and $\boldsymbol{\sigma}_i$ are the apex (center) and the spread of the ith receptive field, respectively.

There are two major differences between the FRBFNNs and the generic version of RBFNNs.

In FRBFNNs, the prototypes of the receptive fields (i.e., the nodes of the hidden layer) are determined by running fuzzy clustering while general RBFNNs use Gaussian-type RBFs as the receptive fields. In general, to define fuzzy sets (i.e., information granules), we analyze data distribution over the input space by engaging some clustering techniques (Bel-

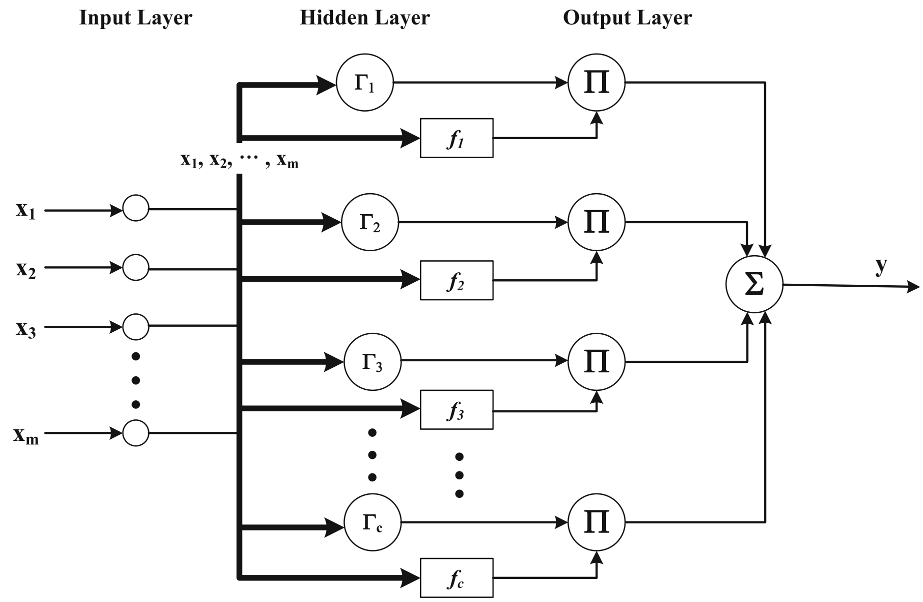


Fig. 8 Architecture of fuzzy RBF neural networks

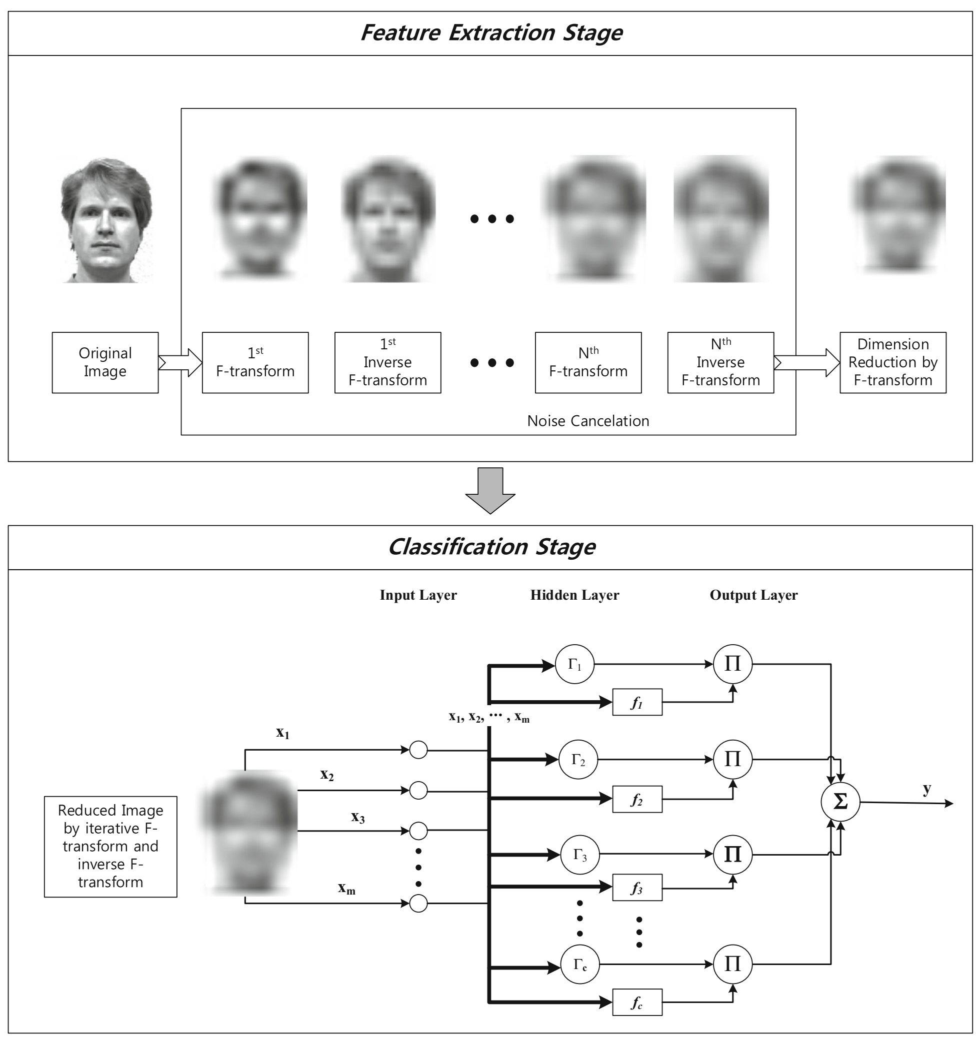


Fig. 9 Whole procedure of face identification system

Table 1 Selected numeric values of the parameters of the proposed model

Algorithm	Parameters		Values
F-transform	Fuzzy partitions	Row	5, 7, 9, 11, 13, 15, 17, 21
		Col	3, 5, 7, 9, 11, 13, 15
	Layers of F-transform and the inverse F-transform (L)		2, 3, 5 layers
	Type of membership function		Uniform triangular membership function
Principal component analysis	The dimension of extracted features (D)		10, 15, 20, 25, 30, 35, 40
Fuzzy RBFNNs	Polynomial order (O)		1 (linear)
	Number of RBFs (c)		2, 3, 4, 5, 7, 9
	Fuzzification coefficient (p)		1.2, 1.5, 2.0, 2.5, 3.0

Fig. 10 Sample images of two persons from the ORL database

humeur et al. 1997). Granulation of information is supported by a suite of procedures that are used to extract meaningful concepts from numeric data or other sources of experimental evidence. In this process of granulation, we actively involve human expertise and judgment of experts to describe the phenomena to be modeled. When dealing with information granulation using numeric data, clustering and fuzzy clustering are commonly encountered (Turk and Pentland 1991; Martino et al. 2014). In particular, we can consider here the

Table 2 Detail information of images from ORL database

Image size	92 × 112
Number of persons	40
Number of images per person	10
Image conditions	All frontal view, slight tilt of the head, glass wearing or not

fuzzy C-means (FCM) clustering algorithm. A brief summary of the method is outlined below.

Let us consider a finite set of data $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{x}_k \in \Re^m$, $1 \leq k \leq N$ The FCM clustering aimed at data granulation optimizes (minimizes) the following objective function

$$J_p = \sum_{i=1}^{c} \sum_{k=1}^{N} w_{ik}^p \left\| \boldsymbol{x}_k - \boldsymbol{v}_i \right\|^2, \; 1 < p < \infty, \tag{11}$$

where the optimization is carried out subject to the following intuitively appealing constraints (i) $\sum_{i=1}^{c} w_{ik} = 1$ and (ii) $0 < \sum_{k=1}^{N} w_{ik} < N$, where $\|\boldsymbol{x}_k - \boldsymbol{v}_i\|$ is any distance between the data $\boldsymbol{x}_k$ and the prototype $\boldsymbol{v}_i$. The parameter "p," $p > 1$ used in the minimized objective function, is referred to as the fuzzification coefficient. "N" and "c" denote the number of data and the number of clusters, respectively. $\boldsymbol{W}$ stands for the partition matrix of activation levels (membership degrees).

Fig. 11 Result images reduced by using the iterative F-transform and the inverse F-transform

Table 3 Comparison of face classification performance about ORL database

Preprocessing algorithm	No. of preprocessing layers (L)	Dimensionality of the extracted features		Fuzzification coefficient	No. of clusters	Classification rate (%)	
		Row	Col			Training dataset	Test dataset
Iterative F-transform and inverse F-transform	2	9	7	1.5	2	100 ± 0	98.50 ± 1.75
	3	9	9	2.5	2	100 ± 0	98.25 ± 2.06
	5	13	5	2.0	3	100 ± 0	***99.0* ± *1.29***
PCA	N/A	15		1.5	7	99.92 ± 0.13	96.25 ± 3.58
	N/A	25		1.5	4	99.83 ± 0.14	97.50 ± 2.36
	N/A	35		1.5	4	100 ± 0	97.75 ± 2.49
	N/A	45		1.5	3	100 ± 0	98.0 ± 1.97

N/A means not available

Bold-italic indicates the superior classification performance

Usually, the value of the fuzzification coefficient is equal to 2. In particular, the distance $\|\cdot\|$ is specified as the Euclidean one coming in the form $\|\boldsymbol{a}-\boldsymbol{b}\|^2 = \sum_{l=1}^{m} (a_l - b_l)^2$. One may consider its generalized weighted version of this distance such as $\|\boldsymbol{a}-\boldsymbol{b}\|^2 = \sum_{l=1}^{m} \frac{(a_l-b_l)^2}{\sigma_l^2}$ where σ_l^2 is the variance of the l-th variable, $l = 1, 2, \ldots, n$.

In the FCM, when optimizing the objective function J_p, we are concerned with a sequence of iterations involving successive calculations of the partition matrix and the related prototypes. In other words, the essence of the method concerns an iterative scheme, which involves the following expressions (here we confine ourselves to the Euclidean distance or any of its weighted generalizations): partition matrix $\boldsymbol{W}$

$$w_{ik} = \frac{1}{\sum_{l=1}^{c} \left(\frac{\|\boldsymbol{x}_k - \boldsymbol{v}_i\|}{\|\boldsymbol{x}_k - \boldsymbol{v}_l\|} \right)^{2/(p-1)}} \tag{12}$$

and prototypes $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c$ are

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{N} w_{ik}^{p} \boldsymbol{x}_k}{\sum_{k=1}^{N} w_{ik}^{p}}, \quad \text{where } i = 1, \ldots, c. \tag{13}$$

The partition matrix and the prototypes are updated in an iterative fashion, and process proceeds until a certain termination criterion is satisfied. The output of each node in the hidden layer is an activation level of the corresponding linguistic term (fuzzy set)

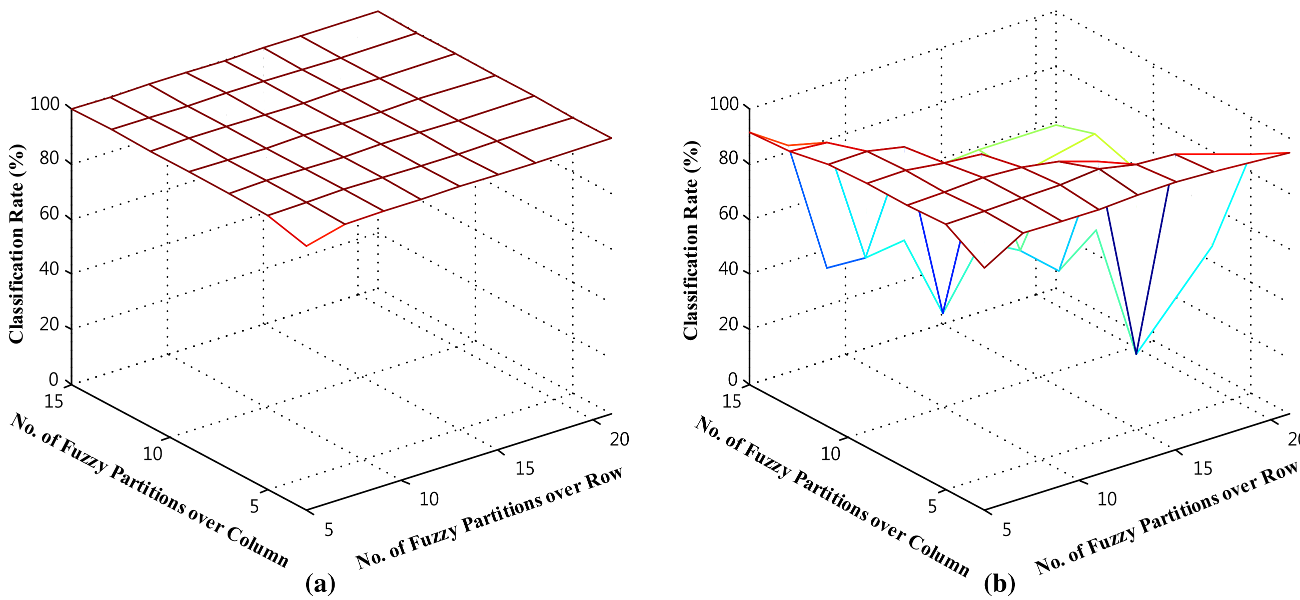


Fig. 12 Classification performance according to the number of fuzzy partitions. **a** Training data, **b** test data

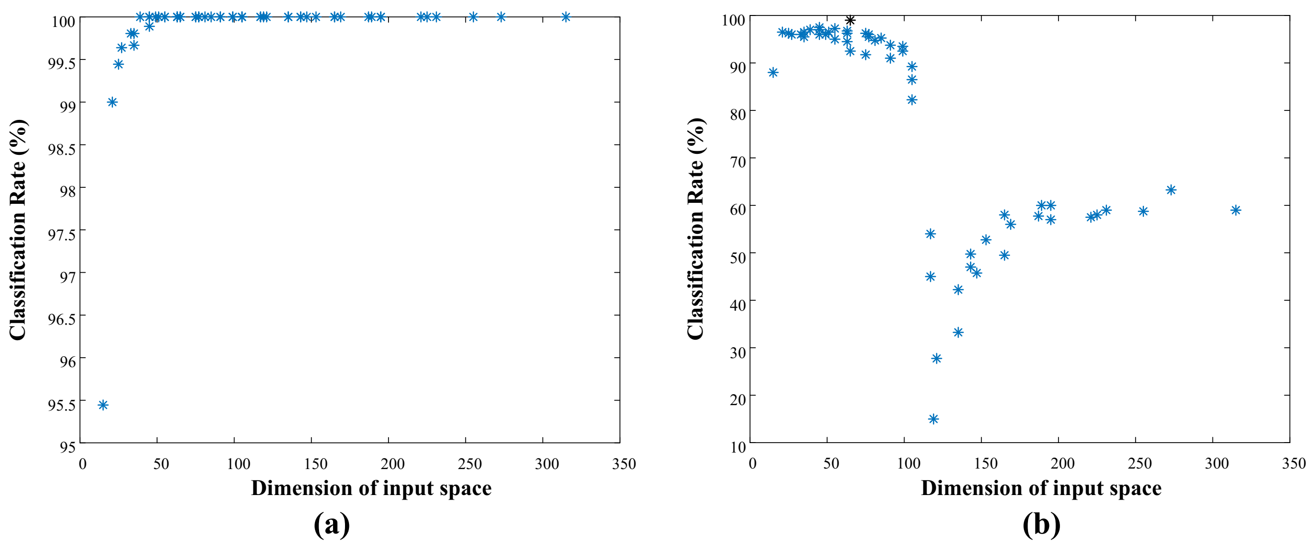


Fig. 13 Classification performance according to the number of input variables. **a** Training data, **b** test data

$$RF_i = \Gamma_i(\boldsymbol{x}) = \frac{1}{\sum_{l=1}^{c}\left(\frac{\|\boldsymbol{x}_k - \boldsymbol{v}_i\|}{\|\boldsymbol{x}_k - \boldsymbol{v}_l\|}\right)^{2/(p-1)}}. \tag{14}$$

The second difference arises in terms of the type of the connection (weights) between the hidden layer and output layer. In FRBFNNs, we use linear functions or the second-order polynomials rather than confining ourselves to some fixed numeric values.

The architecture of FRBFNNs and the type of connection weights considered above are shown in Fig. 8.

In Fig. 8, f_i denotes the connection (weight) between the ith node of hidden layer and the node in the output layer. The connection f_i is expressed as a linear function or the second-order polynomial. More specifically, we have

$$f_i(\boldsymbol{x}_k) = a_{i0} + \sum_{j=1}^{m} a_{ij}x_{kj} = \left[\,1\ \boldsymbol{x}_k\,\right]\mathbf{a}_i. \tag{15}$$

Here, $\mathbf{a}_i = [a_{i0} \quad a_{i1} \quad \cdots \quad a_{im}]^T \in \Re^{(m+1)}$

$$\begin{aligned} f_i(\boldsymbol{x}_k) &= a_{i0} \\ &\quad + \sum_{j=1}^{m} a_{ij}x_{kj} + \sum_{j=1}^{m}\sum_{l=1}^{m} a_{ijl}x_{kj}x_{kl} \\ &= [1 \quad \boldsymbol{x}_k \quad x_{k1}x_{k1} \quad x_{k1}x_{k2} \quad \cdots \quad x_{k1}x_{km} \\ &\quad x_{k2}x_{k2} \quad \cdots \quad x_{km}x_{km}]\,\mathbf{a}_i. \end{aligned} \tag{16}$$

Fig. 14 Sample images of two persons from the YALE database

Table 4 Detail information of images from YALE database

Image size	178 × 236
Number of persons	15
Number of images per person	11
Image conditions	All frontal view, various lightening conditions, glass wearing or not

Here, $\mathbf{a}_i = \left[a_{i0} \quad a_{i1} \quad \cdots \quad a_{im} \quad a_{i11} \quad a_{i12} \quad \cdots \quad a_{i1m} \quad a_{i22} \quad \cdots \quad a_{imm}\right]^T \in \Re^{\frac{(m+1)(m+2)}{2}}$.

The activation level of each node in the hidden layer is determined using (14). The normalized activation level u_{ik} follows the expression

$$u_{ik} = u_i\left(\boldsymbol{x}_k\right) = \frac{\Gamma_i\left(\boldsymbol{x}_k\right)}{\sum_{j=1}^{c} \Gamma_j\left(\boldsymbol{x}_k\right)}. \tag{17}$$

In virtue of the use of the FCM method, the following relationship holds

$$\sum_{i=1}^{c} u_i\left(\boldsymbol{x}_k\right) = \sum_{i=1}^{c} \frac{\Gamma_i\left(\boldsymbol{x}_k\right)}{\sum_{j=1}^{c} \Gamma_j\left(\boldsymbol{x}_k\right)} = 1. \tag{18}$$

For the output node we obtain

$$\hat{y}_k = \sum_{i=1}^{c} u_{ik} f_i\left(\boldsymbol{x}_k\right) = \frac{\sum_{i=1}^{c} \Gamma_i\left(\boldsymbol{x}_k\right) \cdot f_i\left(\boldsymbol{x}_k\right)}{\sum_{i=1}^{c} \Gamma_i\left(\boldsymbol{x}_k\right)}. \tag{19}$$

The whole procedure of the proposed face identification based on F-transform, inverse F-transform, and FRBFNNs is shown in Fig. 9.

6 Experimental results

In order to evaluate the effectiveness of the proposed face identification technique based on iterative F-transform and the inverse F-transform, three kinds of face databases, namely the ORL, YALE, and ABERDEEN databases, are considered. Each experiment is carried out in the tenfold cross-validation mode to validate the proposed face identification technique. In the tenfold cross-validation, the entire dataset is partitioned into ten folds and each fold is left out of the learning process to be used as a testing set. The remaining nine folds are used as a training set to optimize the parameters of the classifiers. The parameter values that yield the best testing error results on all folds will be chosen for evaluating the proposed method. The choice of these particular numeric values has been motivated by the need to come up with a possibility to investigate the performance of the model in a fairly comprehensive range of scenarios.

We investigate and report the results of each experiment in terms of the mean and the standard deviation of the performance index. The classification rate of the face identification technique is used as the performance index. The classification rate is defined as (20).

$$\text{Classification Rate} = \left(1 - \frac{N_e}{N}\right) \times 100\ (\%). \tag{20}$$

Here, N is the number of data patterns, and N_e denotes the number of the misclassification data patterns.

We report some predefined values of the parameters whose values are summarized in Table 1.

Fig. 15 Result images reduced by using the iterative F-transform and the inverse F-transform

Table 5 Comparison of face classification performance about YALE database

Preprocessing algorithm	No. of preprocessing layers (L)	Dimensionality of the extracted features		Fuzzification coefficient	No. of clusters	Classification rate (%)	
		Row	Col			Training dataset	Test dataset
Iterative F-transform and inverse F-transform	2	9	5	2.5	2	100 ± 0	92.06 ± 6.61
	3	15	3	1.5	2	100 ± 0	89.82 ± 6.18
	5	5	7	2.5	3	100 ± 0	91.47 ± 6.66
PCA	N/A	15		2.5	2	99.66 ± 0.36	93.31 ± 8.23
	N/A	25		1.5	2	100 ± 0	97.02 ± 4.19
	N/A	30		2	2	100 ± 0	***97.61 ± 3.09***
	N/A	35		2.0	2	100 ± 0	97.57 ± 3.14

N/A means not available
Bold-italic indicates the superior classification performance

6.1 ORL database

The ORL face database (Samaria and Harter 1994) is a representative database which is usually used in the face recognition applications. This database is composed of 400 different images from 40 individuals. Each individual has ten different images. Some sample images of two persons from the ORL face database are shown in Fig. 10. The detail information of images from the ORL database is described in Table 2.

The result images obtained from the preprocessing techniques based on the iterative F-transform and the inverse F-transform in terms of the number of layers are shown in Fig. 11. The dimensionality of the result images shown in

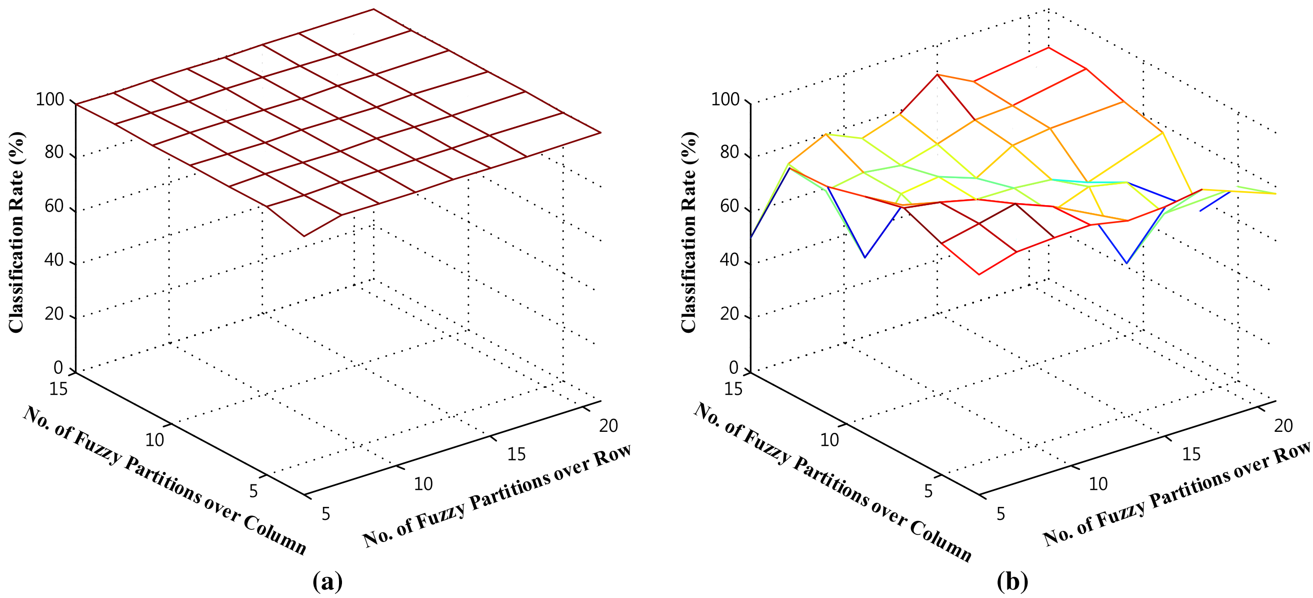


Fig. 16 Classification performance according to the number of fuzzy partitions. **a** Training data, **b** test data

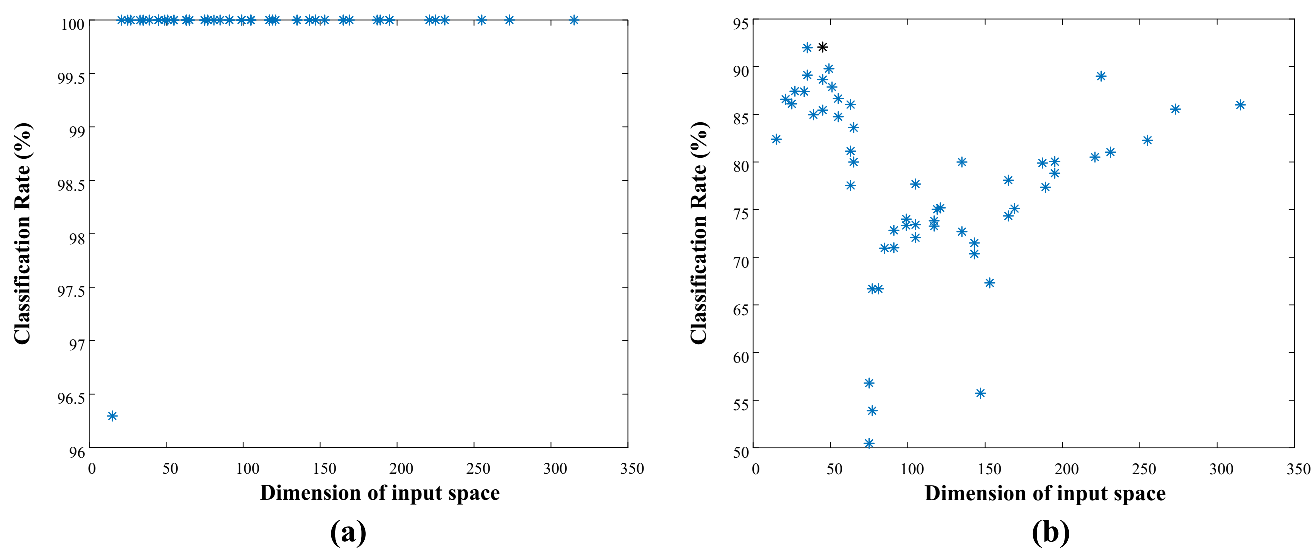


Fig. 17 Classification performance according to the number of input variables. **a** Training data, **b** test data

Fig. 11 is 15 × 21 which is reduced by using the iterative F-transform and the inverse F-transform. The reduced image is used as the input of the neuro-fuzzy classifier; FRBFNNs are used in the study.

The face classification ability of the proposed face identification technique is described in Table 3.

The classification performances of the proposed face identification technique in terms of the number of fuzzy partitions over row and column of an image are shown in Fig. 12. The parameters of the classifier shown in Fig. 12 such as the number of layers (L), the fuzzification coefficient (p), and the number of clusters (c) are $L = 5$, $p = 2.0$ and $c = 3$, respectively.

In Fig. 13, the classification rate is shown in terms of the increase in the number of fuzzy partitions.

One can note that the classification performance on the test data becomes better according to the increase in the number of the fuzzy partitions.

6.2 YALE database

The YALE [17] database is composed of 165 frontal-view face images of 15 subjects. Each subject has 11 different images with various facial expression and lighting conditions. Some sample images of two persons from the YALE

Fig. 18 Sample images of two persons from the ABERDEEN database

Table 6 Detail information of images from ABERDEEN database

Image size	528 × 432
Number of persons	30
Number of images per person	10
Image conditions	All frontal view

face database are shown in Fig. 14. The detail information of images from the YALE database is described in Table 4.

The result images obtained from the preprocessing techniques based on the iterative F-transform and the inverse F-transform in terms of the number of layers are shown in Fig. 15. The dimensionality of the result images shown in Fig. 15 is 15 × 21 which is reduced by using the iterative F-transform and the inverse F-transform.

The face classification ability of the proposed face identification technique applied to YALE database is described in Table 5.

The classification performances of the proposed face identification technique in terms of the number of fuzzy partitions over row and column of an image are shown in Fig. 16. The parameters of the classifier shown in Fig. 16 such as the number of layers (L), the fuzzification coefficient (p), and the number of clusters (c) are L = 2, p = 2.5 and c = 2, respectively.

In Fig. 17, the classification rate is shown in terms of the increase of the number of fuzzy partitions.

From Figs. 16 and 17, one can see that the generalization ability of the proposed face recognition algorithm becomes better according to the increase in the input variables.

6.3 ABERDEEN database

ABERDEEN database has 687 color face images of 90 persons. The number of images of each person is between 1 and 18. We select 30 persons from 90 persons who has more than 10 images, and among the images of the selected person, we choose 10 images randomly. Finally, we select 300 front-view images of 30 persons. Each person has 10 images. Some sample images of two persons from the ABERDEEN face database are shown in Fig. 18. The detail information of images from the ABERDEEN database is described in Table 6.

The face classification ability of the proposed face identification technique applied to ABERDEEN database is described in Table 7.

The classification performances of the proposed face identification technique are shown in terms of the number of fuzzy partitions over row and column of an image in Fig. 19. The parameters of the classifier shown in Fig. 19 such as the number of layers (L), the fuzzification coefficient (p), and the number of clusters (c) are L = 3, p = 1.2 and c = 2, respectively.

In Fig. 20, the classification rate is shown in terms of the increase in the number of fuzzy partitions.

In Table 8, the comparison of the proposed face identification technique with the classifier using conventional principal component analysis to reduce dimensionality is described in terms of the classification rate and the computational cost.

In case that a face image of YALE dataset is overlapped and contaminated by a wide shaded area under the diverse conditions of biased illumination brightness, it is possible for fuzzy transform to partially lose the face data information involved in the shadow part over a large area around the

Table 7 Comparison of face classification performance about ABERDEEN database

Preprocessing algorithm	No. of preprocessing layers (L)	Dimensionality of the extracted features		Fuzzification coefficient	No. of clusters	Classification rate (%)	
		Row	Col			Training dataset	Test dataset
Iterative F-transform and inverse F-transform	2	15	5	2.5	2	100 ± 0	99.0 ± 1.61
	3	15	5	1.2	2	100 ± 0	***99.33 ± 1.41***
	5	11	5	1.2	2	100 ± 0	98.67 ± 2.33
PCA	N/A	15		1.5	7	100 ± 0	95.0 ± 5.50
	N/A	25		1.5	4	100 ± 0	97.02 ± 4.19
	N/A	30		2	2	100 ± 0	98.0 ± 3.22
	N/A	35		1.5	3	100 ± 0	98.33 ± 2.36

N/A means not available
Bold-italic indicates the superior classification performance

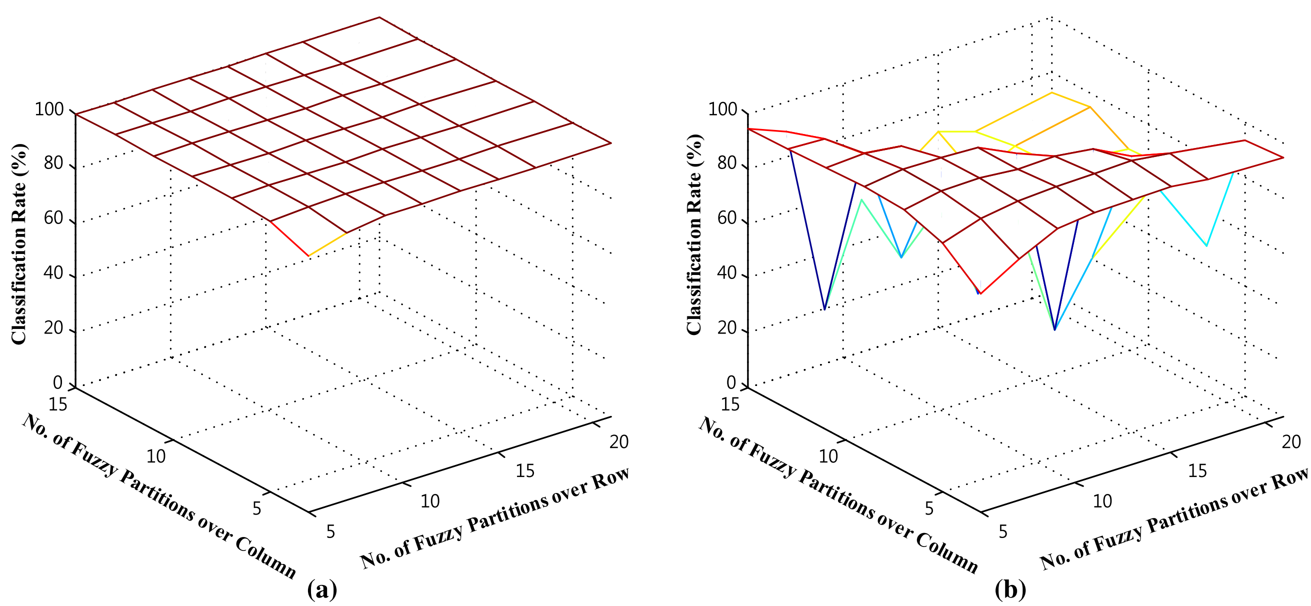


Fig. 19 Classification performance according to the number of fuzzy partitions. **a** Training data, **b** test data

facial image. Due to such effect, the preprocessing of FT-base used in case of YALE dataset could lead to the performance deterioration of FRBFNNs classifier when compared to the PCA-base as shown in Table 8.

The CPU time to extract the new features from original images (in this case, 380 images with 92×112 pixels are used) according to the increase in the dimension of the extracted features is shown in Fig. 21.

While the computational time of PCA is not varied according to the increase of the dimension of features, the computational time for F-transform increases as the dimension increase. However, when the dimension is lower than 80, it takes less time for F-transform to extract the new features than PCA.

7 Conclusion remarks

In this study, we proposed the new preprocessing method to remove the noise from images and reduce the dimension of the input space. Face images are usually very high-dimensional and highly noisy data. In order to reduce the dimension of and to remove noise from face images, the appropriate preprocessing method based on F-transform and inverse F-transform which can be considered as the noise reduction technique as well as the dimension reduction technique. F-transform can be referred as a weighted average mapping method to extract the representative feature over the related subspace defined by a fuzzy set. In other words,

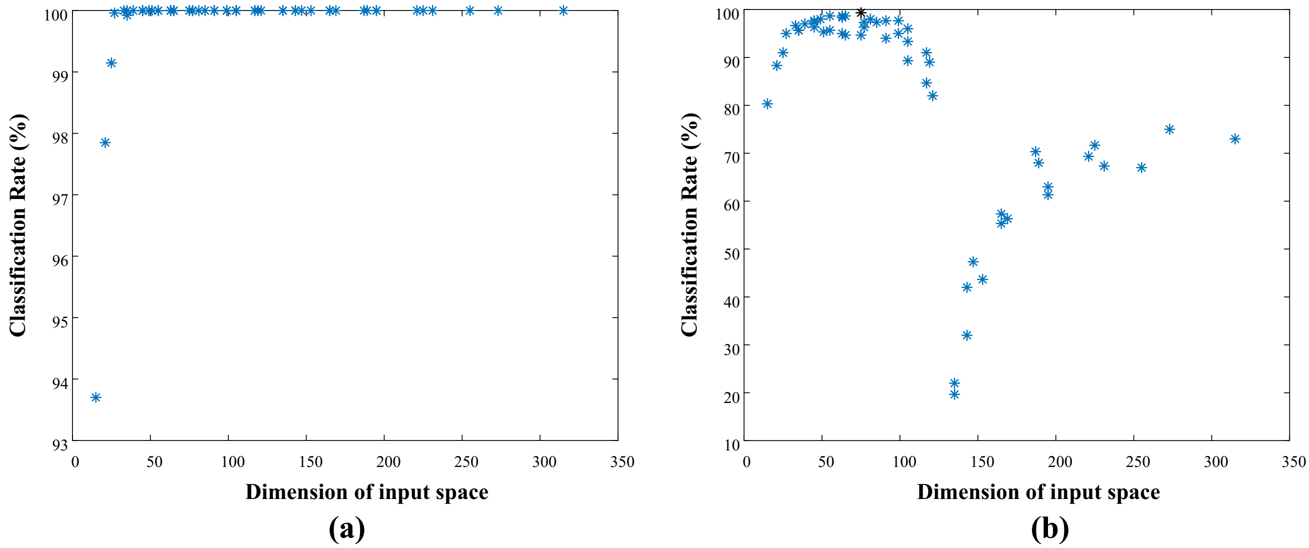


Fig. 20 Classification performance according to the number of input variables. **a** Training data, **b** test data

Table 8 Comparison results of the proposed identification technique with PCA-based identification technique in terms of the classification rate (%) and computational cost (s)

Database			ORL	YALE	ABERDEEN
PCA-based FRBFNNs	Dimension of extracted features (D)		45	30	35
	Classification rate (%)	Training data	100 ± 0	100 ± 0	100 ± 0
		Test data	98.0 ± 1.97	$\boldsymbol{97.61 \pm 3.09}$	98.33 ± 2.36
	Training time for PCA [dimension]		9.4303 [45]	7.7065 [30]	131.2171 [35]
F-transform-based FRBFNNs	Layers of F-transform		5	2	3
	Fuzzy partitions	Row	13	9	15
		Col	5	5	5
	Classification rate (%)	Training data	100 ± 0	100 ± 0	100 ± 0
		Test data	$\boldsymbol{99.0 \pm 1.29}$	92.06 ± 6.61	$\boldsymbol{99.33 \pm 1.41}$
	Training time for F-transform per second [dimension]		7.1736 [45]	5.1952 [45]	178.3197 [75]
				3.4673 [30]	82.6502 [35]

Bold-italic indicates the superior classification performance

the feature extraction or dimension reduction is conducted by using F-transform.

Inverse F-Transform can reconstruct and approximate the original function based on the features extracted by using F-transform. The approximation ability of the Inverse F-transform means the noise filtering properties which can be used to reduce the impulse noise.

Through the iterative F-transform and the inverse F-transform, one may expect that the noise involved in an image can be reduced. The features extracted by using F-transform and inverse F-transform constitute the representative and noise-canceled input data which can result in the good classification performance.

From several experimental results, one can see that the classifier (FRBFNNs) with the proposed preprocessing technique is preferred in the face identification application with respect to the recognition ability. When it comes to the computational time, the proposed feature extraction technique is faster than PCA-based feature extraction method.

In the future research, we will optimize the locations of fuzzy partitions by using several optimization algorithms such as particle swarm optimization and differential evolutionary algorithm. Moreover, the combined preprocessing of both PCA-base and FT-base could be anticipated to effectively enhance the performance of the overall classifier when compared to each use of preprocessing technique with PCA-base or FT-base.

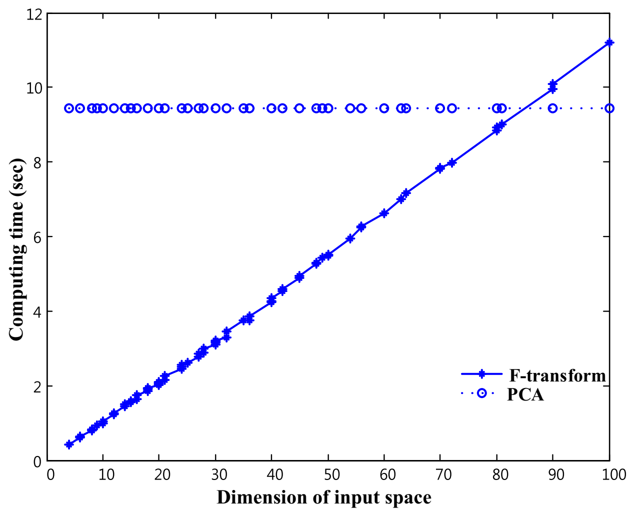


Fig. 21 Computational time to extract features according to the dimension of features

Acknowledgements This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1 B03032333).

Compliance with ethical standards

Conflict of interest Seok-Beom Roh, Sung-Kwun Oh, Jin-Hee Yoon, Kisung Seo declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent None.

References

Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041

Bansal A, Mehta K, Arora S (2012) Face recognition using PCA & LDA algorithms. In: International Conference on Advanced Computing & Communication Technologies (ACCT), 2012 Second, India, pp 251–254

Bartlett M, Movellan JR, Sejnowski T (2002) Face recognition by independent component analysis. IEEE Trans Neural Netw 13(6):1450–1464

Belhumeur PN, Hespanha JP, Kriegman D (1997) Eigenface vs. fisherface: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720

Binsaadoon AG, El-Alfy EM (2015) Statistical Gabor-based gait recognition using region-level analysis. IEEE Eur Model Symp 2015:137–141

Cament LA, Galdames FJ, Bowyer KW, Perez CA (2015) Face recognition under pose variation with local Gabor features enhanced by active shape and statistical models. Pattern Recognit 48(11):3371–3384

Chai Z, Sun Z, Méndez-Vázquez H, He R, Tan T (2014) Gabor ordinal measures for face recognition. IEEE Trans Inf Forensics Secur 9(1):14–26

De Ville DV, Nachtegael M, der Wekenm DV, Kerre EE, Philips W, Lemahieu I (2003) Noise reduction by fuzzy image filtering. IEEE Trans Fuzzy Syst 11(4):429–436

Farokhi S, Mariyam Shamsuddin S, Sheikh UU, Flusser J, Khansari M, Jafari-Khouzani K (2014) Near infrared face recognition by combining Zernike moments and undecimated discrete wavelet transform. Digit Signal Proc 31:13–27

Fukunnaga K (1990) Introduction to statistical pattern recognition, 2nd edn. Academic Press, New York

Holcapek M, Tichy T (2011) A smoothing filter based on fuzzy transform. Fuzzy Sets Syst 180:69–97

Huang W, Oh S-K (2017) Optimized polynomial neural network classifier designed with the aid of space search simultaneous tuning strategy and data preprocessing techniques. J Electr Eng Technol 12(2):911–917

Huang P, Yang Z, Chen C (2015) Fuzzy local discriminant embedding for image feature extraction. Comput Electr Eng 46:231–240

Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324

Li W, Hori Y (2006) An algorithm for extracting fuzzy rules based on RBF neural network. IEEE Trans Ind Electron 53(4):1269–1276

Liu C, Wechsler H (2002) Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. IEEE Trans Image Process 11(4):467–476

Liu L, Fieguth P, Zhao G, Pietikainen M, Hu D (2016) Extended local binary pattern for face recognition. Inf Sci 358–359:56–72

Liu T, Mi J-X, Liu Y, Li C (2016) Robust face recognition via sparse boosting representation. Neurocomputing 214:994–957

Martino FD, Martino P, Perfilieva I, Sessa S (2014) A color image reduction based on fuzzy transform. Inf Sci 266:101–111

Nikolic V, Mitic VV, Kocic L, Petkovic D (2017) Wind speed parameters sensitivity analysis based on fractals and neuro-fuzzy selection technique. Knowl Inf Syst 52(1):255–265

Novak V, Stepnicka M, Dvorak A, Perfilieva I, Pavliska V, Vavrikova L (2010) Analysis of seasonal time series using fuzzy approach. Int J Gen Syst 39(3):305–328

Novak V, Perfilieva I, Holcapek M, Kreinovich V (2014) Filtering out high frequencies in time series using F-transform. Inf Sci 274:192–209

Pang YH, Teoh ABJ, Hiew FS (2015) Locality regularization embedding for face verification. Pattern Recognit 48:86–102

Perfilieva I (2006) Fizzuy transform: theory and applications. Fuzzy Sets Syst 157:993–1023

Perfilieva I (2010) Fuzzy transforms of monotone functions with application to image compression. Inf Sci 180:3304–3315

Perfilieva I, Dubois D, Prade H, Esteva F, Godo L, Hodakova P (2012) Interpolation of fuzzy data: analytical approach and overview. Fuzzy Sets Syst 192:134–158

Perfilieva I, Holcapek M, Kreinovich V (2016) A nre reconstruction from the F-transform comonents. Fuzzy Sets Syst 288:3–25

Petkovic D, Gocic M, Shamshirban S (2016) Adaptive neuro-fuzzy computing technique for precipitation estimation. Facta Univ 14(2):209–218

Petkovic D, Gocic M, Trajkpvic S, Milovancevic M, Sevic D (2017) Precipitation concentration index management by adaptive neuro-fuzzy methodology. Clim Change 141:655–669

Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings 2nd IEEE international workshop on applications of computer vision, Sarasota, FL, pp 138–142

Turk M, Pentland A (1991) Eigenfaces for recognition, Winter. J Cognit Neurosci 3(1):71–86