Expert Systems with Applications 38 (2011) 6184–6189

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Texture recognition by generalized probabilistic decision-based neural networks

Yeong-Yuh Xu [a,*], C.-L. Tseng [b], Hsin-Chia Fu [b]

[a] Department of Computer Science and Information Engineering, HungKuang University, Taichung, Taiwan
[b] Department of Computer Science, National Chiao-Tung University, Hsinchu, Taiwan

ARTICLE INFO

Keywords:
Bayesian decision-based neural networks
Generalized probabilistic decision-based neural networks
GPDNN
Texture recognition
Supervised learning

ABSTRACT

Texture recognition have received tremendous attentions in the past decades, due to its wide applications in computer vision and pattern recognition. For various applications, formulating texture features in distributional forms can sometimes provide meaningful representation than in numerical forms. In this paper, a generalized probabilistic decision-based neural network (GPDNN), based on a novel methodology for the measurement of the difference between two distributions, is proposed for texture recognition. Based on a two-layer pyramid-type network structure, the proposed GPDNN receives texture data via 2-D grid input nodes, and outputs the classification and/or retrieval results at the top layer node. Our prototype system demonstrates a successful utilization of GPDNN to the texture recognition on 40 texture images selected from the MIT Vision Texture (VisTex) database. Regarding the performance, experiment results show that (1) based on the proposed distribution difference measurement method, the texture retrieval accuracy is improved from 77% to 82% by comparing with some recently published leading methods, and (2) the proposed GPDNN has significant improvements in classification accuracy from 82.2% to 90.1% and retrieval accuracy from 79.9% to 88.6% by comparing with traditional approaches.

1. Introduction

Texture recognition has received tremendous attention in the past decades, for its wide applications in computer vision and pattern recognition (Tuceryan & Jain, 1993, chap. texture analysis). Numerous methods have been proposed for texture feature extraction and classification (DO & Vetterli, 2002; Ojala, Pietikainen, & Harwood, 1996; Reed & Buf, 1993). In the recent years, wavelet-like approaches (Avci, 2007; Laine & Fan, 1993; Manjunath & Ma, 1996; Randen & Husoy, 1999; Sengur, Turkoglu, & Ince, 2007; Sengur, 2008; Unser, 1995), such as wavelet packets and Gabor wavelet transforms, are popular in formulating texture features, since these methods are partly supported by physiological studies of the visual cortex (Daugman, 1980; Hubel & Wiesel, 1962). Based on the extracted feature set, finding a human-perception based similarity measurement between various texture patterns is a challenging task and still an ongoing research topic. These approach usually computes the norm-based distances (e.g., Euclidean distance) as the similarity measurement (Manjunath & Ma, 1996) between two feature sets. In general, the texture features can be formulated in either distributional or numerical forms. For some pattern recognition applications, formulating features in distributional forms can sometimes provide more meaningful representation than numerical forms can do (DO & Vetterli, 2002; Manjunath & Ma, 1996).

In the case that texture features are represented by unimodal Gaussian distributions, the weighted Euclidean distance (Manjunath & Ma, 1996) was used to measure the difference between two textures, where the weighting factors are the standard deviations of Gaussian distributions. However, if distributions of texture features need to be estimated by mixture models, such as Gaussian mixture models (GMMs), weighted Euclidean distance will not be suitable for similarity measurement.

In this paper, a generalized probabilistic decision-based neural network (GPDNN), based on a novel methodology for the measurement of the difference between two GMMs, is proposed for texture recognition. Forty texture images are selected from the MIT Vision Texture (VisTex) database (Picard et al., 1995) to demonstrate and evaluate the proposed GPDNN. Experimental results show that, by using the proposed difference measurement method, the texture retrieval rates can be improved from 77% to 82%, compared with some published leading methods, and the proposed GPDNN demonstrated high texture classification and retrieval accuracy, which is about 90.1% and 88.6%, respectively.

This paper is organized as follows. In the next section, the proposed GMM difference measurement is presented. Then, the mathematical background, the architecture, and the learning rules of the GPDNN are introduced in Section 3. Experimental results are presented and discussed in Section 4. Finally, Section 5 draws some concluding remarks.

* Corresponding author.
E-mail addresses: yyxu@sunrise.hk.edu.tw (Y.-Y. Xu), tsengcl@csie.nctu.edu.tw (C.-L. Tseng), hcfu@cs.nctu.edu.tw (H.-C. Fu).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.11.051

2. GMM difference measurement

In order to measure the difference between two GMMs, we proposed a novel methodology for the measurement of the difference between two complex distributions.

2.1. Definition

For two distributions $\mathcal{P}_a$ and $\mathcal{P}_b$ in a feature space R^D, the difference between $\mathcal{P}_a$ and $\mathcal{P}_b$ is defined and denoted as

$$\varphi(\mathcal{P}_a, \mathcal{P}_b) = \int_{R^D} (\mathcal{P}_a - \mathcal{P}_b)^2 d\mathbf{z}. \tag{1}$$

Suppose that $\mathcal{P}_a$ and $\mathcal{P}_b$ are two mixture Gaussian distributions:

$$\mathcal{P}_a = \sum_{r_a=1}^{R_a} P(\theta_{r_a}) p(\mathbf{z}|\theta_{r_a}), \tag{2}$$

$$\mathcal{P}_b = \sum_{r_b=1}^{R_b} P(\theta_{r_b}) p(\mathbf{z}|\theta_{r_b}), \tag{3}$$

where $p(\mathbf{z}|\theta_{r_a})$ and $p(\mathbf{z}|\theta_{r_b})$ are the Gaussian components in $\mathcal{P}_a$ and $\mathcal{P}_b$, respectively. In general, the integration in (1) is very complicated. Thus, we introduce the product moment $\mathcal{F}$ for distributions $\mathcal{P}_a$ and $\mathcal{P}_b$:

$$\mathcal{F}(\mathcal{P}_a, \mathcal{P}_b) = \int_{R^D} \mathcal{P}_a \mathcal{P}_b d\mathbf{z}. \tag{4}$$

Suppose that each Gaussian component in $\mathcal{P}_a$ and $\mathcal{P}_b$ is a D-dimensional Gaussian distribution with uncorrelated variables. Let $\theta_{r_a} = \{\mu_{r_a}, \Sigma_{r_a}\}$ and $\theta_{r_b} = \{\mu_{r_b}, \Sigma_{r_b}\}$ are the parameter sets of $p(\mathbf{z}|\theta_{r_a})$ and $p(\mathbf{z}|\theta_{r_b})$, respectively, where $\mu_{r_a} = \left[\mu_{r_a(1)} \cdots \mu_{r_a(D)}\right]^T$ and $\mu_{r_b} = \left[\mu_{r_b(1)} \cdots \mu_{r_b(D)}\right]^T$ are the mean vectors, and diagonal matrices $\Sigma_{r_a} = diag\left[\sigma^2_{r_a(1)} \cdots \sigma^2_{r_a(D)}\right]$ and $\Sigma_{r_b} = diag\left[\sigma^2_{r_b(1)} \cdots \sigma^2_{r_b(D)}\right]$ are the covariance matrices. The following lemma simplify the product moment $\mathcal{F}$ to a closed-form expression.

Lemma 2.1. *Suppose that $\mathcal{P}_a$ and $\mathcal{P}_b$ are two mixture Gaussian distributions defined in (2) and (3), respectively. The product moment $\mathcal{F}$ of $\mathcal{P}_a$ and $\mathcal{P}_b$ is expressed as:*

$$\mathcal{F}(\mathcal{P}_a, \mathcal{P}_b) = \sum_{r_a=1}^{R_a} \sum_{r_b=1}^{R_b} P(\theta_{r_a}) P(\theta_{r_b}) \mathcal{G}(\theta_{r_a}, \theta_{r_b}), \tag{5}$$

where

$$\mathcal{G}(\theta_{r_a}, \theta_{r_b}) = \frac{\exp\left\{-\frac{1}{2}\sum_{d=1}^{D} \frac{(\mu_{r_b(d)} - \mu_{r_a(d)})^2}{\sigma^2_{r_b(d)} + \sigma^2_{r_a(d)}}\right\}}{\sqrt{(2\pi)^D \prod_{d=1}^{D} \left(\sigma^2_{r_b(d)} + \sigma^2_{r_a(d)}\right)}}. \tag{6}$$

The proof of Lemma 2.1 is in Appendix A. According to Lemma 2.1, $\varphi(\mathcal{P}_a, \mathcal{P}_b)$ can be further expressed as follows.

Theorem 2.2. *Let $\mathcal{P}_a$ and $\mathcal{P}_b$ denote $p(\mathbf{z}|\theta_{r_a})$ and $p(\mathbf{z}|\theta_{r_b})$, respectively. The discriminate function $\varphi(\mathcal{P}_a, \mathcal{P}_b)$ defined in (1) is expressed as*

$$\varphi(\mathcal{P}_a, \mathcal{P}_b) = \mathcal{F}(\mathcal{P}_a, \mathcal{P}_a) - 2\mathcal{F}(\mathcal{P}_a, \mathcal{P}_b) + \mathcal{F}(\mathcal{P}_b, \mathcal{P}_b). \tag{7}$$

Proof. As shown in (1),

$$\begin{aligned}
\varphi(\mathcal{P}_a, \mathcal{P}_b) &= \int_{R^D} (\mathcal{P}_a - \mathcal{P}_b)^2 d\mathbf{z} \\
&= \int_{R^D} (\mathcal{P}_a^2 - 2\mathcal{P}_a\mathcal{P}_b + \mathcal{P}_b^2) d\mathbf{z} \\
&= \int_{R^D} \mathcal{P}_a^2 d\mathbf{z} - 2\int_{R^D} \mathcal{P}_a\mathcal{P}_b d\mathbf{z} + \int_{R^D} \mathcal{P}_b^2 d\mathbf{z} \\
&= \mathcal{F}(\mathcal{P}_a, \mathcal{P}_a) - 2\mathcal{F}(\mathcal{P}_a, \mathcal{P}_b) + \mathcal{F}(\mathcal{P}_b, \mathcal{P}_b). \qquad \square
\end{aligned}$$

3. Generalized probabilistic decision-based neural network

The generalized probabilistic decision-based neural network (GPDNN) is a generalized model from its predecessor, the self-growing probabilistic decision-based neural network (SPDNN) (Fu & Xu, 1998). Similar to its predecessor SPDNN, a GPDNN has a modular network structure. One subnet of a GPDNN is designed to represent one object class. The decision-based learning rules are adopted to train a GPDNN. Based on the teacher information that only tells the correctness of the classification for each training datum, a GPDNN performs a distributed and localized updating rule. The updating rule applies reinforced learning to a subnet corresponding to the correct class and antireinforced learning to the (unduly) winning subnets. The detailed description of the GPDNN model is given in the following sections.

3.1. Discriminant Functions of GPDNN

One of the major differences between SPDNN and GPDNN is that GPDNN receives data in distributional forms instead of numerical forms as network inputs. That is, the subnet discriminant functions of GPDNN are designed to measure the difference between the input distribution and the modelled or learned distributions. Thus, reinforced or antireinforced learning is applied to all the subnets of the global winner and the supposed (i.e., the correct) winner, with a weighting distribution proportional to the degree of possible involvement by each subnet.

In order to measure the difference between two distributions in each subnet, we use (7) as the discriminate function of the multi-class GPDNN. As shown in Fig. 1, a GPDNN contains k subnets that are used to represent a k-category classification problem. Assume that the modelled distribution for each class $\omega_i : i = 1, 2, \ldots, k$ is $\mathcal{P}_i$. For an input datum $\mathbf{x}(t)$ with a distribution $\mathcal{P}_t$, according to the expression in (7), the subnet discriminant function is implemented by a two-layer pyramid network as shown in Fig. 2. The bottom layer contains three structurally identical pyramid subnetworks, each of which computes the $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_i)$, $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_t)$, and $\mathcal{F}(\mathcal{P}_t, \mathcal{P}_t)$, respectively. By summing the results of these three components, the difference between the input datum $\mathbf{x}(t)$ and the modelled class ω_i is output at the top layer of the pyramid subnetwork.

Fig. 3 depicts the internal architecture of the pyramid subnetwork corresponding to $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_t)$. Suppose that the mixture Gaussian distributions $\mathcal{P}_i$ and $\mathcal{P}_t$ consist of R_i and R_t Gaussian clusters, respectively. Thus, the subnetwork for $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_t)$ contains a $R_i \times R_t$ input grids and R_i hidden nodes, where a node in input grids is marked as G_{r_i, r_t}.

While the subnetwork receives an input datum $\mathbf{x}(t)$ with a distribution $\mathcal{P}_t$, each G_{r_i, r_t} node performs the computation according to (6) to measure the difference between two Gaussian components r_i and r_t, which are in $\mathcal{P}_i$ and $\mathcal{P}_t$, respectively. Then, the outputs of all the G_{r_i, r_t} nodes are weighted by $P(\theta_{r_t})$ and summed to the hidden node h_{r_i}:

$$h_{r_i} = \sum_{r_t=1}^{R_t} P(\theta_{r_t}) \cdot G_{r_i, r_t}. \tag{8}$$

Finally, the output of each hidden node h_{r_i}, $r_i = 1, \ldots, R_i$, are weighted by $P(\theta_{r_i})$ and then summed to the output node of the pyramid subnetwork:

$$\mathcal{F}(\mathcal{P}_i|\mathcal{P}_t) = \sum_{r_i=1}^{R_i} P(\theta_{r_i}) \cdot h_{r_i}. \tag{9}$$

When the outputs of all subnets reach the top layer, the MINNET is activated to select the minimum of the values from the lower subnet and its corresponding subnet ID. It means that if the output

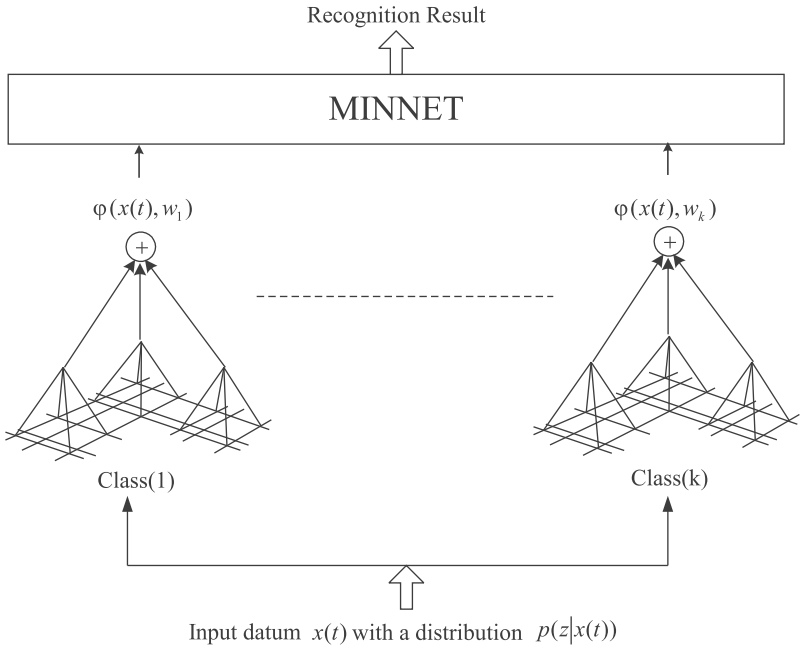

Fig. 1. The schematic diagram of a *k-class* GPDNN. The detail of a subnet is shown in Fig. 2. An input of a GPDNN is represented in distributional forms instead of numerical values.

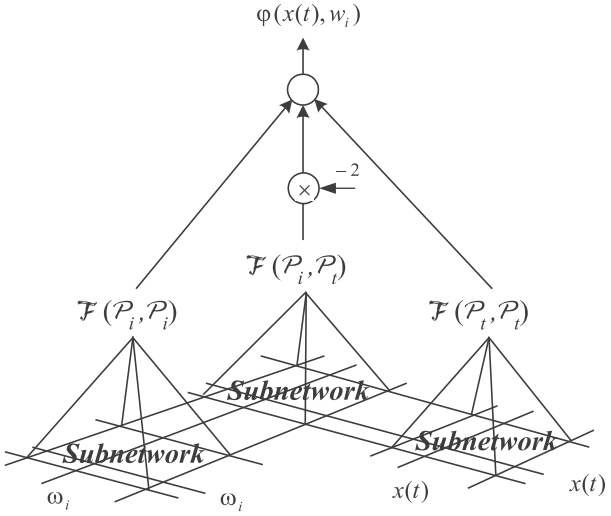

Fig. 2. The diagram of a subnet in *generalized probabilistic decision based neural network* (GPDNN).

value of subnet i is the minimum among the outputs of all subnets in a GPDNN, the input datum $\mathbf{x}(t)$ is classified as ω_i.

3.2. Learning Rules for GPDNN

Recall that the training scheme for a multiclass GPDNN follows the SPDNN learning rules. While the input datum $\mathbf{x}(t)$ belonging to the class ω_i is misclassified to the class ω_j, the following reinforced and antireinforced learning rules are applied to the subnets of ω_i and ω_j, respectively.

Reinforced Learning rule :

$$\mathbf{w}_i^{(m+1)} = \mathbf{w}_i^{(m)} + \eta \nabla \varphi(\mathbf{x}(t), \mathbf{w}_i), \tag{10}$$

Antireinforced Learning rule :

$$\mathbf{w}_j^{(m+1)} = \mathbf{w}_j^{(m)} - \eta \nabla \varphi(\mathbf{x}(t), \mathbf{w}_j). \tag{11}$$

The gradient vectors in (10) and (11) are computed as follows:

$$\frac{\partial \varphi(\mathbf{x}(t), \mathbf{w}_i)}{\partial \mu_{r_i(d)}} = 2P(\theta_{r_i}) \left[\sum_{r_n=1}^{R_i} \left(\frac{P(\theta_{r_n}) \mathcal{G}(\theta_{r_n}, \theta_{r_i})}{\sigma^2_{r_n(d)} + \sigma^2_{r_i(d)}} \right) \times \left(\mu_{r_n(d)} - \mu_{r_i(d)} \right) - \sum_{r_t=1}^{R_t} \left(\frac{P(\theta_{r_t}) \mathcal{G}(\theta_{r_t}, \theta_{r_i})}{\sigma^2_{r_t(d)} + \sigma^2_{r_i(d)}} \right) \times \left(\mu_{r_t(d)} - \mu_{r_i(d)} \right) \right], \tag{12}$$

$$\frac{\partial \varphi(\mathbf{x}(t), \mathbf{w}_i)}{\partial \sigma^2_{r_i(d)}} = P(\theta_{r_i}) \left[\sum_{r_n=1}^{R_i} \left(\frac{P(\theta_{r_n}) \mathcal{G}(\theta_{r_n}, \theta_{r_i})}{\sigma^2_{r_n(d)} + \sigma^2_{r_i(d)}} \right) \times \left(\frac{(\mu_{r_n(d)} - \mu_{r_i(d)})^2}{\sigma^2_{r_n(d)} + \sigma^2_{r_i(d)}} - 1 \right) - \sum_{r_t=1}^{R_t} \left(\frac{P(\theta_{r_t}) \mathcal{G}(\theta_{r_t}, \theta_{r_i})}{\sigma^2_{r_t(d)} + \sigma^2_{r_i(d)}} \right) \times \left(\frac{(\mu_{r_t(d)} - \mu_{r_i(d)})^2}{\sigma^2_{r_t(d)} + \sigma^2_{r_i(d)}} - 1 \right) \right], \tag{13}$$

where $r_i = 1,2,\ldots,R_i$ and $d = 1,2,\ldots,D$. D is the dimension of the feature space.

4. Experiments

A total of 40 different texture images are selected from the MIT Vision Texture (VisTex) database (Picard et al., 1995) to demonstrate and evaluate the proposed GPDNN for the texture recognition. The original size of the images is 512 × 512, and each image was divided into 16 disjoint subimages, i.e., 128 × 128 pixels. Hence, the classification problem involved a total of 640 subimages, 16 subimages in each of the 40 texture categories (DO & Vetterli, 2002), and only the gray-scale images were used in the experiments.

4.1. Texture feature extraction

The Gabor wavelet decomposition proposed by Manjunath and Ma (1996) is applied to extract the texture features from an image of multiple scales and orientations. Advantage of using Gabor wavelet decomposition is that the Gabor representation is optimal (Daugman, 1988) in the sense of minimizing the uncertainty in the spatial and frequency domain. The filter parameters we used are 4 scales and 6 orientations, which produced a bank of 24 filters

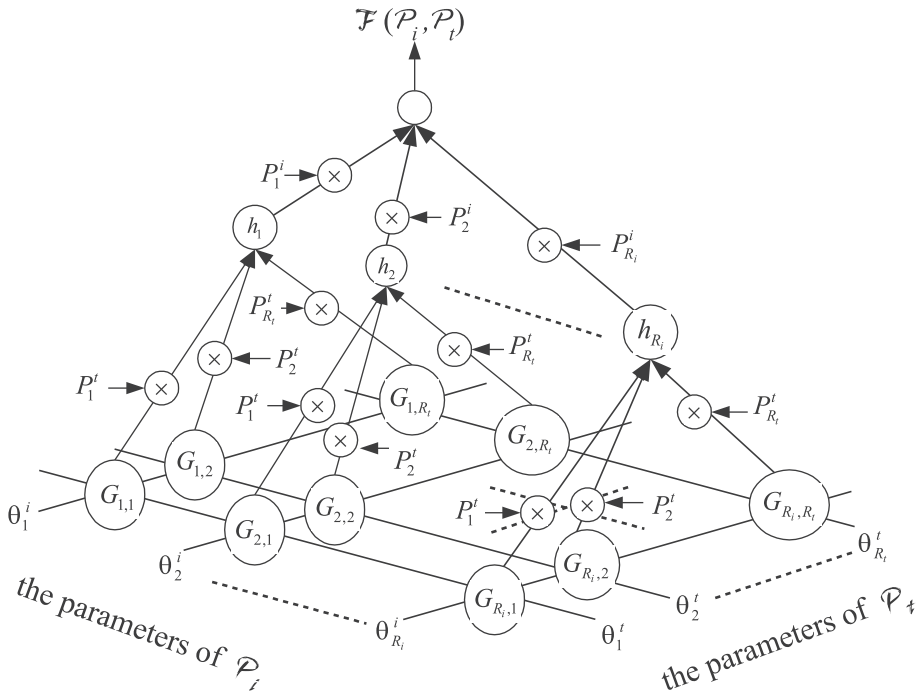

Fig. 3. The internal architecture of a model in Fig. 2 for computing of $\mathcal{F}(\mathcal{P}_i, \mathcal{P}_t)$.

$\{g_{t,k} : 1 \leqslant t \leqslant 4$, and $1 \leqslant k \leqslant 6\}$. The filter response of a subimage I is formulated as $I_{t,k} = I \overset{*}{} g_{t,k}$, for $1 \leqslant t \leqslant 4$, and $1 \leqslant k \leqslant 6$. Then, the texture feature around a pixel $\mathbf{x}$ is represented by a 24-dimension vector $[|I_{1,1}(\mathbf{x})|^2, \ldots, |I_{t,k}(\mathbf{x})|^2, \ldots, |I_{4,6}(\mathbf{x})|^2]^T$.

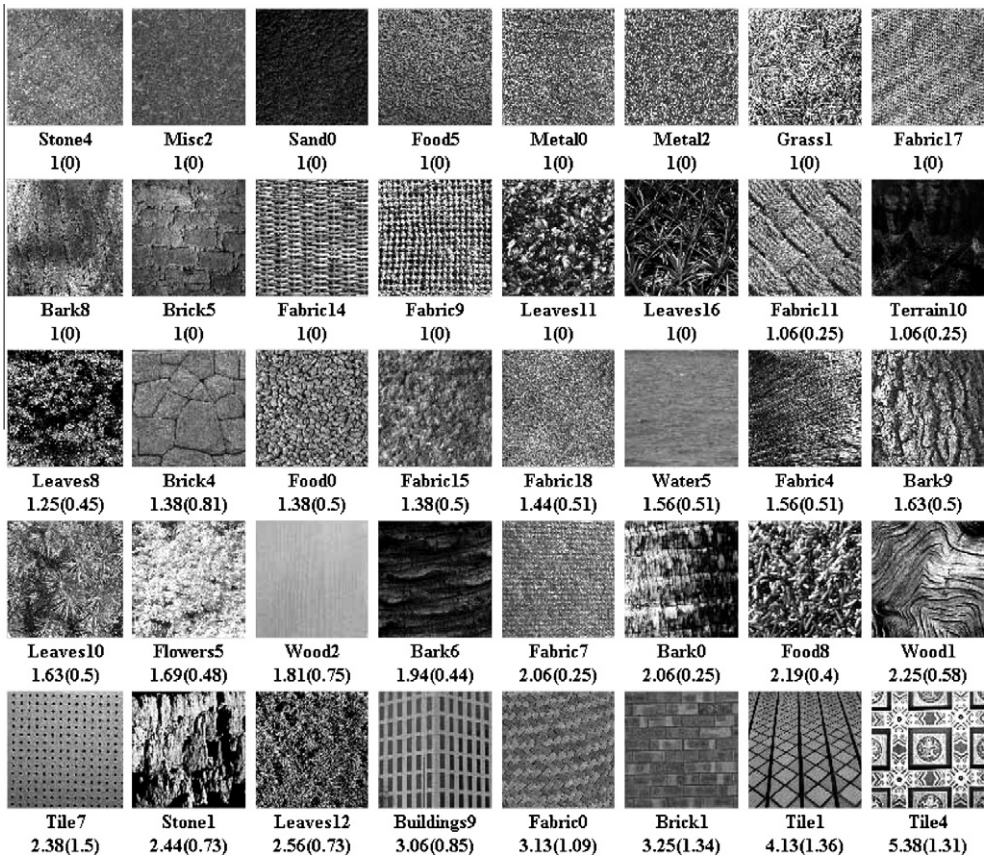

Fig. 4. The 40 texture categories with their names (shown in the first row), means and standard deviations (shown in the second row in the form of "mean (SD)") of the number of mixture components in GMMs. It is indicated that the more complicated the texture is, the more mixture components are needed to approximate the texture feature distribution.

Gaussian mixture models (GMMs) are used to approximate the texture feature distributions of the 640 subimages. For each subimage, an EM algorithm is employed for finding maximum likelihood estimates of parameters, and the Bayesian information criterion (BIC) (Schwarz, 1978) is used to determine the number of mixture components of the corresponding GMM. Fig. 4 shows the mean and standard deviation (SD) of the number of mixture components per texture category, where the first row is the texture image, the second row is the texture name, and the third row is the mean and SD in the form of "mean (SD)". As we can see, the more complicated the texture is, the more mixture components are needed to approximate the texture feature distribution.

4.2. Evaluation of the GMM difference measurement

The first experiment is conducted to examine the performance of the proposed GMM difference measurement. Similar to the experiments described in DO and Vetterli (2002), we used each of 640 images in the databases as a simulated query image, and defined the relevant images for each query as the other 15 subimages from the same original VisTex image. The retrieval rate is the average percentages of retrieving relevant images in the top 15 matches. As shown in Table 1, the proposed method improves the retrieval performance from 77% to 82%, which is 5% better than the recent published leading methods ([DO & Vetterli, 2002).

4.3. Texture recognition system

In this experiment, a GPDNN-based texture recognition system is developed. In order to evaluate the performance of the proposed GPDNN, the cross validation method (Devijver & Kittler, 1982) is used to randomly split the dataset into training and testing data. For each such split, the GPDNN is trained and tested using the training and testing data, respectively. The results are then averaged over the splits. For comparison purpose, two other classifiers, a minimum-distance classifier (MDC) and an SPDNN (Fu & Xu, 1998), were also built. Each classifier was trained by randomly chosen eight texture subimages, and tested by the rest of eight texture subimages in each texture category. The experimental results of both training and testing set are shown in Table 2. It is clear to see that the proposed GPDNN has significant improvements in classification accuracy from 82.2% to 90.1% and retrieval accuracy from 79.9% to 88.6%.

Table 1
Average retrieval rates for individual texture class using the proposed GMM difference measurement. Method I denotes the *generalized Gaussian density (GGD) and kullback–Leibler distance (KLD) method* proposed in DO and Vetterli (2002), and II denotes the proposed GMM difference measurement method.

Texture class	Methods		Texture class	Methods	
	I	II		II	
Bark0	53.12	68.75	Food8	97.66	100.00
Bark6	50.39	77.50	Grass1	69.14	66.67
Bark8	73.44	83.33	Leaves8	68.36	75.83
Bark9	61.33	74.17	Leaves10	34.38	51.67
Brick1	71.88	91.67	Leaves11	71.48	76.67
Brick4	66.41	81.67	Leaves12	74.61	83.33
Brick5	83.20	77.92	Leaves16	84.77	78.33
Buildings9	86.72	85.00	Metal0	73.05	83.75
Fabric0	87.50	92.50	Metal2	100.00	100.00
Fabric4	64.84	68.33	Misc2	78.12	86.67
Fabric7	100.00	100.00	Sand0	80.08	81.25
Fabric9	87.89	86.25	Stone1	53.52	56.25
Fabric11	81.25	81.25	Stone4	79.30	84.58
Fabric14	100.00	100.00	Terrain10	52.73	55.00
Fabric15	94.53	97.08	Tile1	53.12	60.00
Fabric17	90.23	100.00	Tile4	99.22	94.17
Fabric18	98.83	100.00	Tile7	100.00	99.58
Flowers5	58.20	74.58	Water5	96.48	100.00
Food0	83.59	92.92	Wood1	35.55	25.00
Food5	89.45	87.08	Wood2	78.52	87.08

Table 2
Means and standard deviations of classification and retrieval accuracies for GPDNN, SPDNN, and MDC, where the standard deviation is given in the parentheses.

	Classification accuracy		Retrieval accuracy	
	Training set	Testing set	Training set	Testing set
GPDNN	0.989(0.006)	0.901(0.021)	0.988(0.008)	0.886(0.026)
SPDNN	0.947(0.025)	0.851(0.016)	0.940(0.026)	0.826(0.020)
MDC	0.873(0.022)	0.822(0.016)	0.858(0.025)	0.799(0.023)

5. Concluding remarks

In this paper, a generalized probabilistic decision-based neural network (GPDNN) is proposed and implemented, based on a novel methodology for measuring the difference between two Gaussian mixture models (GMMs). 40 texture images are selected from the MIT Vision Texture (VisTex) database to demonstrate and to evaluate the proposed GPDNN for the texture recognition. Experimental results show that (1) the proposed GMM difference measurement improves the retrieval rates, e.g., from 77% to 82%, compared with some published leading methods, and (2) the proposed GPDNN shows significant texture classification and retrieval performance, which are about 90.1% and 88.6% of the accuracy, and much better than the traditional methods, i.e., 82.2% and 79.9%, respectively. Although this paper presents applications of the GPDNN for the texture classification and retrieval, more image/pattern recognition and classification problems and applications will be solved and/or applied by the proposed GPDNN.

Acknowledgement

The authors acknowledge Prof. S.Y. Kung and Prof. Y.H. Hu for their helpful suggestions regarding the probabilistic DBNN and statistical pattern recognition methods.

Appendix A

Suppose $\mathcal{P}_a$ and $\mathcal{P}_b$ are two mixture Gaussian distributions defined in (2) and (3), respectively. The product moment of $\mathcal{P}_a$ and $\mathcal{P}_b$,

$$\mathcal{F}(\mathcal{P}_a,\mathcal{P}_b)=\sum_{r_a=1}^{R_a}\sum_{r_b=1}^{R_b}P(\theta_{r_a})P(\theta_{r_b})\mathcal{G}(\theta_{r_a},\theta_{r_b}),$$

where

$$\mathcal{G}(\theta_{r_a},\theta_{r_b})=\frac{\exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\frac{(\mu_{r_b(d)}-\mu_{r_a(d)})^2}{\sigma^2_{r_b(d)}+\sigma^2_{r_a(d)}}\right\}}{\sqrt{(2\pi)^D\prod_{d=1}^{D}\left(\sigma^2_{r_b(d)}+\sigma^2_{r_a(d)}\right)}}.$$

Proof

$$\begin{aligned}\mathcal{F}(\mathcal{P}_a,\mathcal{P}_b)&=\int_{R^D}\mathcal{P}_a\mathcal{P}_b d\mathbf{z}\\&=\int_{R^D}\left(\sum_{r_a=1}^{R_a}P(\theta_{r_a})p(\mathbf{z}|\theta_{r_a})\times\sum_{r_b=1}^{R_b}P(\theta_{r_b})p(\mathbf{z}|\theta_{r_b})\right)d\mathbf{z}\\&=\sum_{r_a=1}^{R_a}\sum_{r_b=1}^{R_b}\left(P(\theta_{r_a})P(\theta_{r_b})\times\int_{R^D}p(\mathbf{z}|\theta_{r_b})p(\mathbf{z}|\theta_{r_a})d\mathbf{z}\right).\end{aligned}$$

Let $M(\theta_{r_b}|\theta_{r_a})$ be the expected value of the $p(\mathbf{z}|\theta_{r_b})$ under the Gaussian distribution $p(\mathbf{z}|\theta_{r_a})$,

$$M(\theta_{r_b}|\theta_{r_a}) = \int_{R^D} p(\mathbf{z}|\theta_{r_b})p(\mathbf{z}|\theta_{r_a})d\mathbf{z} = \underbrace{\int\cdots\int}_{D} \{p(z_1,\ldots,z_D|\theta_{r_b}) \times p(z_1,\ldots,z_D|\theta_{r_a})\}dz_1\cdots dz_D. \tag{14}$$

Since the random variables $z_1, z_2, \ldots, z_D$ are mutually independent, the following joint probability densities can be expressed in product forms:

$$p(z_1,\cdots,z_D|\theta_{r_b}) = \prod_{d=1}^{D} p(z_d|\theta_{r_b}), \text{and}$$
$$p(z_1,\cdots,z_D|\theta_{r_a}) = \prod_{d=1}^{D} p(z_d|\theta_{r_a}). \tag{15}$$

Substitute (15) into (14),

$$M(\theta_{r_b}|\theta_{r_a}) = \prod_{d=1}^{D} M_d(\theta_{r_b}|\theta_{r_a}). \tag{16}$$

Each of the product terms, $M_d(\theta_{r_b}|\theta_{r_a})$, can be derived as follows:

$$\begin{aligned} M_d(\theta_{r_b}|\theta_{r_a}) &= \int_{-\infty}^{\infty} p(z_d|\theta_{r_b})p(z_d|\theta_{r_a})dz_d = \frac{1}{2\pi\sigma_{r_a(d)}\sigma_{r_b(d)}} \\ &\quad\times \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2}\left(\frac{(z_d-\mu_{r_a(d)})^2}{\sigma^2_{r_a(d)}} + \frac{(z_d-\mu_{r_b(d)})^2}{\sigma^2_{r_b(d)}}\right)\right\}dz_d \\ &= \frac{1}{2\pi\sigma_{r_a(d)}\sigma_{r_b(d)}} \times \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2}\left[\left(\frac{1}{\sigma^2_{r_a(d)}} + \frac{1}{\sigma^2_{r_b(d)}}\right)z_d^2 \right.\right. \\ &\quad \left.\left. -2\left(\frac{\mu_{r_a(d)}}{\sigma^2_{r_a(d)}} + \frac{\mu_{r_b(d)}}{\sigma^2_{r_b(d)}}\right)z_d + \left(\frac{\mu^2_{r_a(d)}}{\sigma^2_{r_a(d)}} + \frac{\mu^2_{r_b(d)}}{\sigma^2_{r_b(d)}}\right)\right]\right\}dz_d. \end{aligned}$$

Let

$$\mathcal{A}_{(d)} = \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2_{r_a(d)}} + \frac{1}{\sigma^2_{r_b(d)}}\right) \times \left(z_d - \frac{\sigma^2_{r_b(d)}\mu_{r_a(d)} + \sigma^2_{r_a(d)}\mu_{r_b(d)}}{\sigma^2_{r_a(d)} + \sigma^2_{r_b(d)}}\right)^2\right\}dz_d.$$

Then

$$\begin{aligned} M_d(\theta_{r_b}|\theta_{r_a}) &= \frac{\mathcal{A}_{(d)}}{2\pi\sigma_{r_a(d)}\sigma_{r_b(d)}} \\ &\quad\times \exp\left[-\frac{1}{2}\left(\frac{\mu^2_{r_a(d)}}{\sigma^2_{r_a(d)}} + \frac{\mu^2_{r_b(d)}}{\sigma^2_{r_b(d)}}\right) + \frac{1}{2}\frac{\left(\frac{\mu_{r_a(d)}}{\sigma^2_{r_a(d)}} + \frac{\mu_{r_b(d)}}{\sigma^2_{r_b(d)}}\right)^2}{\left(\frac{1}{\sigma^2_{r_a(d)}} + \frac{1}{\sigma^2_{r_b(d)}}\right)}\right] \\ &= \frac{\mathcal{A}_{(d)}}{2\pi\sigma_{r_a(d)}\sigma_{r_b(d)}} \times \exp\left[-\frac{1}{2}\frac{(\mu_{r_a(d)}-\mu_{r_b(d)})^2}{\sigma^2_{r_a(d)} + \sigma^2_{r_b(d)}}\right]. \end{aligned} \tag{17}$$

Let $s_d = z_d - \frac{\sigma^2_{r_b(d)}\mu_{r_a(d)} + \sigma^2_{r_a(d)}\mu_{r_b(d)}}{\sigma^2_{r_a(d)} + \sigma^2_{r_b(d)}}$; then

$$\mathcal{A}_{(d)} \times \mathcal{A}_{(d)} = \mathcal{A}^2_{(d)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2_{r_a(d)}} + \frac{1}{\sigma^2_{r_b(d)}}\right) \times (s_d^2 + t_d^2)\right\}ds_d dt_d.$$

Change (s_d, t_d) coordinate system to (ρ_d, α_d) polar coordinate system; let $s_d = \rho_d\cos\alpha_d$ and $t_d = \rho_d\sin\alpha_d$, then

$$\begin{aligned} \mathcal{A}^2_{(d)} &= \int_0^{\infty} \rho_d \int_0^{2\pi} \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2_{r_a(d)}} + \frac{1}{\sigma^2_{r_b(d)}}\right) \times \rho_d^2\right\}d\rho_d d\alpha_d \\ &= 2\pi\left(\frac{\sigma^2_{r_a(d)}\sigma^2_{r_b(d)}}{\sigma^2_{r_a(d)} + \sigma^2_{r_b(d)}}\right). \end{aligned}$$

Hence,

$$\mathcal{A}_{(d)} = \sqrt{2\pi\left(\frac{\sigma^2_{r_a(d)}\sigma^2_{r_b(d)}}{\sigma^2_{r_a(d)} + \sigma^2_{r_b(d)}}\right)}. \tag{18}$$

Substitute (18) into (17);

$$M_d(\theta_{r_b}|\theta_{r_a}) = \frac{\exp\left(-\frac{1}{2}\frac{(\mu_{r_b(d)}-\mu_{r_a(d)})^2}{\sigma^2_{r_a(d)}+\sigma^2_{r_b(d)}}\right)}{\sqrt{2\pi}\sqrt{\sigma^2_{r_a(d)} + \sigma^2_{r_b(d)}}}. \tag{19}$$

Finally, substitute (19) into (16), and

$$M(\theta_{r_b}|\theta_{r_a}) = \frac{\exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{(\mu_{r_b(d)}-\mu_{r_a(d)})^2}{\sigma^2_{r_a(d)}+\sigma^2_{r_b(d)}}\right)}{\sqrt{(2\pi)^D\prod_{d=1}^{D}\left(\sigma^2_{r_a(d)} + \sigma^2_{r_b(d)}\right)}} = \mathcal{G}(\theta_{r_b}|\theta_{r_a}). \qquad \square$$

References

Avci, E. (2007). An expert system based on wavelet neural network-adaptive norm entropy for scale invariant texture classification. *Expert Systems with Applications, 32*(3), 919–926.

Daugman, J. (1980). Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research, 20*, 847–856.

Daugman, J. G. (1988). Complete discrete 2d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on ASSP, 36*, 1169–1179.

Devijver, P. A., & Kittler, J. (1982). *Pattern recognition: A statistical approach*. Prentice-Hall.

DO, M. N., & Vetterli, M. (2002). Wavelet-based texture retrieval using generalized Gaussian density and Kullback–Leibler distance. *IEEE Transactions on Image Processing, 11*(2), 146–158.

Fu, H. C., & Xu, Y. Y. (1998). Multilinguistic handwritten character recognition by bayesian decision-based neural networks. *IEEE Transactions on Signal Processing, 46*(10), 2781–2789.

Hubel, D., & Wiesel, T. (1962). Receptive field, binocular interaction, and functional architecture in the cats visual cortex. *Journal of Physiology, 160*, 106–154.

Laine, A., & Fan, J. (1993). Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Recognition and Machine Intelligence, 15*(11), 1186–1190.

Manjunath, B., & Ma, W. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Recognition and Machine Intelligence, 18*(8), 837–842.

Ojala, T., Pietikainen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition, 29*(1), 51–59.

Randen, T., & Husoy, J. H. (1999). Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Recognition and Machine Intelligence, 21*(4), 291–310.

Reed, T. R., & Buf, J. M. H. D. (1993). A review of recent texture segmentation and feature extraction techniques. *Computer Vision Graphics Image Process. Image Understanding, 57*, 359–372.

Schwarz, G. (1978). Estimation the dimension of a model. *Annals of Statistics, 6*, 461V464.

Sengur, A. (2008). Wavelet transform and adaptive neuro-fuzzy inference system for color texture classification. *Expert Systems with Applications, 34*(3), 2120–2128.

Sengur, A., Turkoglu, I., & Ince, M. C. (2007). Wavelet packet neural networks for texture classification. *Expert Systems with Applications, 32*(2), 527–533.

Tuceryan, M., & Jain, A. (1993). *Handbook pattern recognition and computer vision*. Singapore: World Scientific. pp. 235–276.

Unser, M. (1995). Texture classification and segmentation using wavelet frames. *IEEE Transactions on Image Processing, 4*(11), 1549–1560.

Picard, R., Graczyk, C., Mann, S., Wachman, J., Picard, L., & Campbell, L. (1995). *Vistex vision texture database*. <http://vismod.www.media.mit.edu>