

Abstract—As a newly developed 2-D extension of the wavelet transform using multiscale and directional filter banks, the contourlet transform can effectively capture the intrinsic geometric structures and smooth contours of a texture image that are the dominant features for texture classification. In this paper, we propose a novel Bayesian texture classifier based on the adaptive model-selection learning of Poisson mixtures on the contourlet features of texture images. The adaptive model-selection learning of Poisson mixtures is carried out by the recently established adaptive gradient Bayesian Ying-Yang harmony learning algorithm for Poisson mixtures. It is demonstrated by the experiments that our proposed Bayesian classifier significantly improves the texture classification accuracy in comparison with several current state-of-the-art texture classification approaches.

Index Terms—Bayesian Ying-Yang (BYY) harmony learning system, contourlet transform, model selection, Poisson mixtures, texture classification.

TEXTURE classification is a fundamental problem in computer vision and image processing. In recent years, there has been a significant amount of research on this topic, and many different approaches have been proposed. One of the most popular approaches is the use of wavelet transforms, which have been shown to be effective in capturing the local features of texture images. However, wavelet transforms are limited in their ability to capture the global structure of texture images, and this can lead to poor classification performance. In this paper, we propose a novel Bayesian texture classifier based on the contourlet transform, which is a 2-D extension of the wavelet transform that can capture both local and global features of texture images. The contourlet transform is based on the use of multiscale and directional filter banks, which allow it to capture the intrinsic geometric structures and smooth contours of a texture image. We propose a novel Bayesian classifier based on the adaptive model-selection learning of Poisson mixtures on the contourlet features of texture images. The adaptive model-selection learning of Poisson mixtures is carried out by the recently established adaptive gradient Bayesian Ying-Yang harmony learning algorithm for Poisson mixtures. It is demonstrated by the experiments that our proposed Bayesian classifier significantly improves the texture classification accuracy in comparison with several current state-of-the-art texture classification approaches.

The contourlet transform is a 2-D extension of the wavelet transform that can capture both local and global features of texture images. It is based on the use of multiscale and directional filter banks, which allow it to capture the intrinsic geometric structures and smooth contours of a texture image. The contourlet transform is defined as follows: Let $f(x, y)$ be a 2-D function. The contourlet transform of $f(x, y)$ is defined as

$$C(f)(x, y, \theta, s) = \int \int f(x', y') \psi(x - x', y - y', \theta, s) dx' dy'$$

where $\psi(x, y, \theta, s)$ is the contourlet wavelet, which is defined as

$$\psi(x, y, \theta, s) = \frac{1}{s^2} \psi\left(\frac{x}{s}, \frac{y}{s}, \theta\right)$$

where $\psi(x, y, \theta)$ is the contourlet wavelet, which is defined as

$$\psi(x, y, \theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x}{s} \cos \theta + \frac{y}{s} \sin \theta\right)^2\right) \exp\left(-\frac{1}{2} \left(\frac{-x}{s} \sin \theta + \frac{y}{s} \cos \theta\right)^2\right)$$

where s is the scale, θ is the orientation, and $\psi(x, y, \theta)$ is the contourlet wavelet. The contourlet transform is a linear transform, and it can be used to represent a texture image as a set of contourlet coefficients. The contourlet coefficients are defined as

$$C(f)(x, y, \theta, s) = \int \int f(x', y') \psi(x - x', y - y', \theta, s) dx' dy'$$

where $f(x, y)$ is the texture image, $\psi(x, y, \theta, s)$ is the contourlet wavelet, and $C(f)(x, y, \theta, s)$ is the contourlet coefficient. The contourlet coefficients are used to represent the texture image, and they are used in the Bayesian classifier to classify the texture image.

The contourlet transform is a 2-D extension of the wavelet transform that can capture both local and global features of texture images. It is based on the use of multiscale and directional filter banks, which allow it to capture the intrinsic geometric structures and smooth contours of a texture image. The contourlet transform is defined as follows: Let $f(x, y)$ be a 2-D function. The contourlet transform of $f(x, y)$ is defined as

$$C(f)(x, y, \theta, s) = \int \int f(x', y') \psi(x - x', y - y', \theta, s) dx' dy'$$

where $\psi(x, y, \theta, s)$ is the contourlet wavelet, which is defined as

$$\psi(x, y, \theta, s) = \frac{1}{s^2} \psi\left(\frac{x}{s}, \frac{y}{s}, \theta\right)$$

where $\psi(x, y, \theta)$ is the contourlet wavelet, which is defined as

$$\psi(x, y, \theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x}{s} \cos \theta + \frac{y}{s} \sin \theta\right)^2\right) \exp\left(-\frac{1}{2} \left(\frac{-x}{s} \sin \theta + \frac{y}{s} \cos \theta\right)^2\right)$$

where s is the scale, θ is the orientation, and $\psi(x, y, \theta)$ is the contourlet wavelet. The contourlet transform is a linear transform, and it can be used to represent a texture image as a set of contourlet coefficients. The contourlet coefficients are defined as

$$C(f)(x, y, \theta, s) = \int \int f(x', y') \psi(x - x', y - y', \theta, s) dx' dy'$$

where $f(x, y)$ is the texture image, $\psi(x, y, \theta, s)$ is the contourlet wavelet, and $C(f)(x, y, \theta, s)$ is the contourlet coefficient. The contourlet coefficients are used to represent the texture image, and they are used in the Bayesian classifier to classify the texture image.

b ... a ... h ...
 a ... n ... n ...
 b ... a ... n ...
 a ... h ...
 b ... a ... h ...
 a ... n ... a ...
 a ... n ... a ...
 h ... a ... a ...
 a ... Y ... Y ... (... Y ...)
 a ... Y ... a ... a ...
 h ... a ... a ...
 b ... a ... Y ...
 a ... (...) ...
 a ... a ... a ...
 a ... a ... a ...
 a ... a ... a ...
 a ... a ... a ...
 a ... a ... a ...

a ... b ...
 a ... Y ...
 a ... h ...
 a ... h ...
 a ... h ...
 h ... a ... a ...
 a ... Y ...
 a ... V ...
 a ... a ... a ...

hence $U_i(x_t) = \alpha_j q(x_t|\theta_j)$, $j = 1, 2, \dots, k$, and $q(x_t|\theta_j) = (\theta_j^{x_t}/x_t!)e^{-\theta_j}$.

Let $\alpha_j = e^{\beta_j}/\sum_{i=1}^k e^{\beta_i}$, $j = 1, 2, \dots, k$, hence $-\infty < \beta_1, \dots, \beta_k < +\infty$, $\alpha_j \geq 0$ and $\sum_{j=1}^k \alpha_j = 1$.

Let $J_t(\Theta_k)$ be the log-likelihood function of x_t with respect to β_j and θ_j .

$$\frac{\partial J_t(\Theta_k)}{\partial \beta_j} = \frac{1}{q(x_t|\Theta_k)} \sum_{i=1}^k A(x_t, i)(\delta_{i,j} - \alpha_j)U_i(x_t) \quad (1)$$

$$\frac{\partial J_t(\Theta_k)}{\partial \theta_j} = \frac{1}{q(x_t|\Theta_k)} A(x_t, j)\alpha_j \frac{\partial q(x_t|\theta_j)}{\partial \theta_j} \quad (2)$$

hence $A(x_t) = [1 - \sum_{l=1}^k (p(l|x_t) - \delta_{il}) \ln U_l(x_t)]$, $p(l|x_t) = (\alpha_l q(x_t|\theta_l)/q(x_t|\Theta_k))$, $\delta_{i,j} = 1$ if $i=j$ and 0 otherwise. $(\partial q(x_t|\theta_j)/\partial \theta_j) = (\theta_j^{x_t-1}/x_t!)e^{-\theta_j}(x_t - \theta_j)$.

Let β_j be the parameters of the distribution.

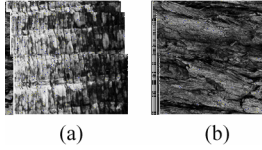


Figure 1: Two grayscale images of wood grain, labeled (a) and (b).

hence $E_{i,j}^1 = i \cdot f_{i,j}^1 = (1/M_{i,j}N_{i,j}) \sum_{x=1}^{M_{i,j}} \sum_{y=1}^{N_{i,j}} |w_{i,j}(x,y)|$ and $w_{i,j}(x,y) \in c_{i,j}[\mathbf{n}]$, hence $x = 1, 2, \dots, M_{i,j}$ and $y = 1, 2, \dots, N_{i,j}$.

Let $\mathbf{w}_I = [w_I(x,y)]_{x=1, y=1}^{M_I \times N_I}$ be the feature vector of the image \mathbf{w}_I and $\mathbf{a}_I = [a_I(x,y)]_{x=1, y=1}^{M_I \times N_I}$ be the feature vector of the image \mathbf{a}_I .

$$g_I^1 = \frac{1}{2^I \cdot M_I N_I} \sum_{x=1}^{M_I} \sum_{y=1}^{N_I} |w_I(x,y)| \quad (1)$$

$$g_I^2 = \frac{1}{2^I \cdot M_I N_I} \sum_{x=1}^{M_I} \sum_{y=1}^{N_I} ||w_I(x,y)| - E_I^1| \quad (2)$$

$$g_I^3 = \frac{1}{2^I} \left[\max_{x,y} w_I(x,y) - \min_{x,y} w_I(x,y) \right] \quad (3)$$

hence $E_I^1 = 2^I \cdot g_I^1 = (1/M_I N_I) \sum_{x=1}^{M_I} \sum_{y=1}^{N_I} |w_I(x,y)|$ and $w_I(x,y) \in a_I[\mathbf{n}]$, hence $x = 1, 2, \dots, M_I$ and $y = 1, 2, \dots, N_I$.

Let $\mathbf{f}_I = [f_I^1, f_I^2, f_I^3]$ be the feature vector of the image \mathbf{f}_I and $\mathbf{f}_I = [f_I^1, f_I^2, f_I^3]$ be the feature vector of the image \mathbf{f}_I .

$$\mathbf{F} = (g_I^1, g_I^2, g_I^3, \mathbf{f}_I, \dots, \mathbf{f}_I) \quad (4)$$

hence $\mathbf{f}_i = (f_{i,1}^1, \dots, f_{i,8}^1, f_{i,1}^2, \dots, f_{i,8}^2, f_{i,1}^3, \dots, f_{i,8}^3)$ and $\mathbf{f}_i = (f_{i,1}^1, \dots, f_{i,8}^1, f_{i,1}^2, \dots, f_{i,8}^2, f_{i,1}^3, \dots, f_{i,8}^3)$.

$$\mathbf{F} = (f(m))_{m=1}^M \quad (5)$$

hence $M = 24 * I + 3$. Let $\mathbf{f}(m)$ be the feature vector of the image $\mathbf{f}(m)$.

Let $\mathbf{f}(m) = [f(m,1), f(m,2), \dots, f(m,24 * I + 3)]$ be the feature vector of the image $\mathbf{f}(m)$ and $\mathbf{f}(m) = [f(m,1), f(m,2), \dots, f(m,24 * I + 3)]$ be the feature vector of the image $\mathbf{f}(m)$.

Let $\mathbf{f}(m) = [f(m,1), f(m,2), \dots, f(m,24 * I + 3)]$ be the feature vector of the image $\mathbf{f}(m)$ and $\mathbf{f}(m) = [f(m,1), f(m,2), \dots, f(m,24 * I + 3)]$ be the feature vector of the image $\mathbf{f}(m)$.

Let $\mathbf{f}(m) = [f(m,1), f(m,2), \dots, f(m,24 * I + 3)]$ be the feature vector of the image $\mathbf{f}(m)$ and $\mathbf{f}(m) = [f(m,1), f(m,2), \dots, f(m,24 * I + 3)]$ be the feature vector of the image $\mathbf{f}(m)$.

Let $\mathbf{f}(m) = [f(m,1), f(m,2), \dots, f(m,24 * I + 3)]$ be the feature vector of the image $\mathbf{f}(m)$ and $\mathbf{f}(m) = [f(m,1), f(m,2), \dots, f(m,24 * I + 3)]$ be the feature vector of the image $\mathbf{f}(m)$.

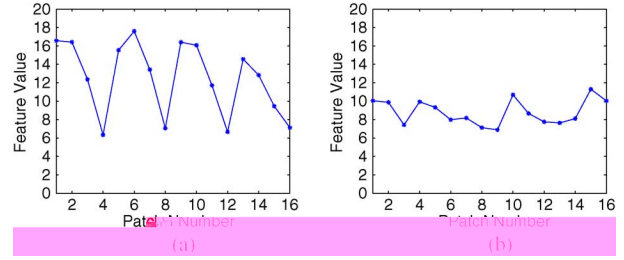


Figure 2: Two line plots showing Feature Value vs. Patch Number for two different images. Plot (a) shows a highly oscillatory signal, while plot (b) shows a smoother signal.

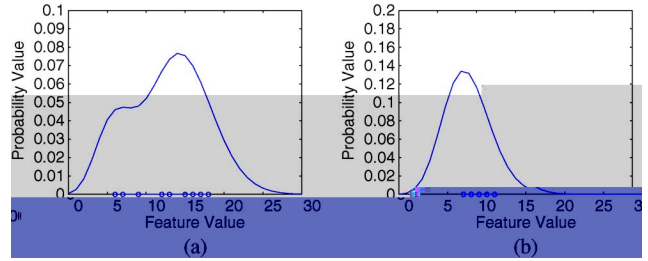


Figure 3: Two probability density function plots showing Probability Value vs. Feature Value for two different images. Plot (a) shows a broad distribution, while plot (b) shows a narrower distribution.

Let $\mathbf{f}_{3,1}^2$ be the feature vector of the image $\mathbf{f}_{3,1}^2$ and $\mathbf{f}_{3,1}^2 = [f_{3,1}^2(1), f_{3,1}^2(2), \dots, f_{3,1}^2(24 * I + 3)]$ be the feature vector of the image $\mathbf{f}_{3,1}^2$.

Let $\mathbf{f}_{3,1}^2$ be the feature vector of the image $\mathbf{f}_{3,1}^2$ and $\mathbf{f}_{3,1}^2 = [f_{3,1}^2(1), f_{3,1}^2(2), \dots, f_{3,1}^2(24 * I + 3)]$ be the feature vector of the image $\mathbf{f}_{3,1}^2$.

Let $\mathbf{f}_{3,1}^2$ be the feature vector of the image $\mathbf{f}_{3,1}^2$ and $\mathbf{f}_{3,1}^2 = [f_{3,1}^2(1), f_{3,1}^2(2), \dots, f_{3,1}^2(24 * I + 3)]$ be the feature vector of the image $\mathbf{f}_{3,1}^2$.

Let $\mathbf{f}_{3,1}^2$ be the feature vector of the image $\mathbf{f}_{3,1}^2$ and $\mathbf{f}_{3,1}^2 = [f_{3,1}^2(1), f_{3,1}^2(2), \dots, f_{3,1}^2(24 * I + 3)]$ be the feature vector of the image $\mathbf{f}_{3,1}^2$.

Let $\mathbf{f}_{3,1}^2$ be the feature vector of the image $\mathbf{f}_{3,1}^2$ and $\mathbf{f}_{3,1}^2 = [f_{3,1}^2(1), f_{3,1}^2(2), \dots, f_{3,1}^2(24 * I + 3)]$ be the feature vector of the image $\mathbf{f}_{3,1}^2$.

Let $\mathbf{f}_{3,1}^2$ be the feature vector of the image $\mathbf{f}_{3,1}^2$ and $\mathbf{f}_{3,1}^2 = [f_{3,1}^2(1), f_{3,1}^2(2), \dots, f_{3,1}^2(24 * I + 3)]$ be the feature vector of the image $\mathbf{f}_{3,1}^2$.

Let $\mathbf{f}_{3,1}^2$ be the feature vector of the image $\mathbf{f}_{3,1}^2$ and $\mathbf{f}_{3,1}^2 = [f_{3,1}^2(1), f_{3,1}^2(2), \dots, f_{3,1}^2(24 * I + 3)]$ be the feature vector of the image $\mathbf{f}_{3,1}^2$.

Let $\mathbf{f}_{3,1}^2$ be the feature vector of the image $\mathbf{f}_{3,1}^2$ and $\mathbf{f}_{3,1}^2 = [f_{3,1}^2(1), f_{3,1}^2(2), \dots, f_{3,1}^2(24 * I + 3)]$ be the feature vector of the image $\mathbf{f}_{3,1}^2$.

$$q(f(m)|\theta_{m,j}) = \frac{\alpha_{m,j}^{k_m} f(m)^{k_m-1}}{\Gamma(k_m)} e^{-\alpha_{m,j} f(m)},$$

where \mathbf{F} is the feature vector of the patch, $\alpha_{m,j}$ and $\theta_{m,j}$ are the parameters of the Poisson mixture, k_m is the order of the Poisson mixture, and $f(m)$ is the feature vector of the patch. The feature vector \mathbf{F} is defined as $\mathbf{F} = [f(1), f(2), \dots, f(M)]^T$, where $f(m)$ is the feature vector of the patch at scale m . The feature vector \mathbf{F} is used to calculate the likelihood of the patch given the texture class c and the Poisson mixture parameters $\Theta_{c,m}$.

B. Bayesian Texture Classifier Based on Poisson Mixtures

The Bayesian texture classifier is based on the Poisson mixture model. The likelihood of the patch given the texture class c is calculated as follows:

$$P(\mathbf{F}|T_c) = \frac{P(\mathbf{F}|T_c)P(T_c)}{\sum_{i=1}^C P(\mathbf{F}|T_i)P(T_i)} \quad (1)$$

where T_c is the texture class, $c = 1, 2, \dots, C$, $P(T_c)$ is the prior probability of the texture class, $P(\mathbf{F}|T_c)$ is the likelihood of the patch given the texture class, and \mathbf{F} is the feature vector of the patch. The likelihood of the patch given the texture class c is calculated as follows:

$$P(\mathbf{F}|T_c) = \prod_{m=1}^M P(f(m)|T_c) \quad (2)$$

where $P(f(m)|T_c) = \sum_{i=1}^{k_m} \alpha_{c,m,i} q(f(m)|\theta_{c,m,i})$, $\{\alpha_{c,m,i}, \theta_{c,m,i}, i = 1, 2, \dots, k_m\}$ are the parameters of the Poisson mixture, $f(m)$ is the feature vector of the patch at scale m , and k_m is the order of the Poisson mixture.

[Input:] Training texture patches selected from each texture image or class (the number of training patches for each texture image is equal).
[Output:] The parameter sets of all the Poisson mixtures for each texture class.

- (1) Decompose a patch of a given texture image or class c with contourlets into an output of L scales with 8 directional subbands at each scale.
- (2) Calculate the value of the feature vector (15) with Eqs (8)-(13) for the patch and obtain the value of the quantized feature vector \mathbf{F} by the deadzone quantization.
- (3) Repeat the above two steps for all the training patches of the c -th texture class and obtain the training samples of the feature vector \mathbf{F} for the c -th class.
- (4) Implement the adaptive gradient BYY algorithm on the m -th components of all the training samples of the feature vector \mathbf{F} for the c -th texture class, and obtain the parameters of the Poisson mixture distribution of the variable $f(m)$: $\Theta_{c,m} = \{\alpha_{c,m,i}, \theta_{c,m,i}, i = 1, 2, \dots, k_m\}$.
- (5) Repeat Step (4) for all the m components of the training samples of the feature vector \mathbf{F} for the c -th texture class, and obtain the c -th class parameters: $\Theta_c = \{\Theta_{c,m}, m = 1, 2, \dots, M\}$.
- (6) Repeat the above steps for all texture classes, i.e., $c = 1, 2, \dots, C$.

The likelihood of the patch given the texture class c is calculated as follows:

$$P(\mathbf{F}|T_c) = \prod_{m=1}^M P(f(m)|T_c) \quad (3)$$

where $P(f(m)|T_c) = \sum_{i=1}^{k_m} \alpha_{c,m,i} q(f(m)|\theta_{c,m,i})$, $\{\alpha_{c,m,i}, \theta_{c,m,i}, i = 1, 2, \dots, k_m\}$ are the parameters of the Poisson mixture, $f(m)$ is the feature vector of the patch at scale m , and k_m is the order of the Poisson mixture.

A. Performance Evaluation

The performance of the proposed method is evaluated using the Receiver Operating Characteristic (ROC) curve. The ROC curve is a plot of the True Positive Rate (TPR) versus the False Positive Rate (FPR). The ROC curve is used to evaluate the performance of the classifier. The ROC curve is calculated as follows:

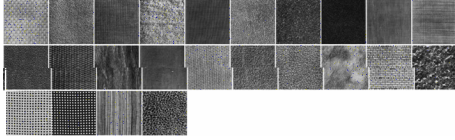


Figure 1: A grid of 15 grayscale images showing various textures and patterns, likely representing different data sets or model outputs.

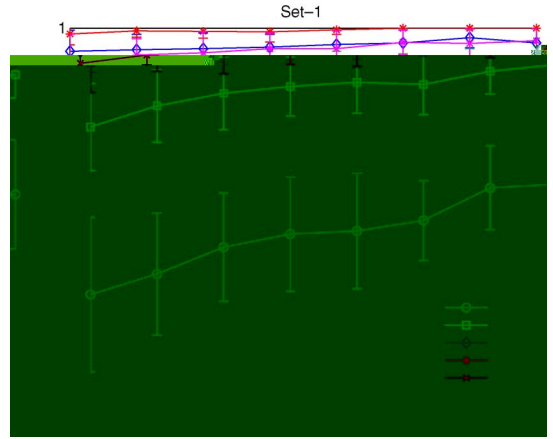


Figure 2: A plot titled 'Set-1' showing a green grid with red and purple lines and markers. The plot appears to be a visualization of a model's performance or a specific data set configuration.

Let Θ_k denote the parameter vector at iteration k . The optimization process is defined by the following steps:

$$f(m) \quad m = 1, 2, \dots, M.$$

For $k \geq 10$, the update rule is given by:

$$\Theta_k = \Theta_{k-1} + \alpha \nabla_{\Theta} J(\Theta_{k-1})$$

For $k \leq 3$, the update rule is given by:

$$\Theta_k = \Theta_{k-1} + \beta \nabla_{\Theta} J(\Theta_{k-1})$$

$$0, \quad \text{where } \alpha, \beta \text{ are learning rates.}$$

$$J(\Theta_k)$$

$$|J(\Theta_k^{new}) - J(\Theta_k^{old})| < 10^{-4}$$

$$10^3$$

$$10^3$$

$$|J(\Theta_k^{new}) - J(\Theta_k^{old})| < 10^{-4}$$

$$0$$

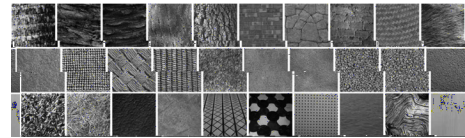
$$W \times N$$

$$00$$

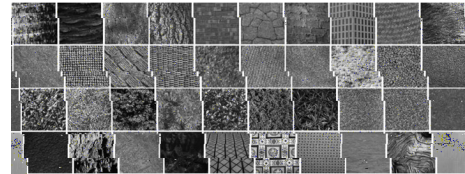
For $N = 6, 7, \dots, 13$, the results are summarized in the following table:

N	Performance Metric
6	0%
7	0%
8	0%
9	0%
10	0%
11	00%
12	00%
13	00%

b n n n n n h n n b b
 n n n n n n n n n n
 n n b n n n n n
 h n n b b b h n n n
 b n n n n n n n n n
 h n n N n n n n n n
 n n n n n n n n n n
 0. % 0 h n N n n n n b
 b n n n n n n n n n
 n n n n n n n n n n
 n n n n n n n n n n
 b n n n n n n n n n N
 n n n n n n n n n n
 n n n n n () n n n
 n n n n n n n n n n
 n n n n n n n n n n 00%.
 h n n n n n n n n n
 n n n n n n n n n n
 n n n n n n n n n n
 n n n n n n n n n n
 b n n n n n n n n n n
 n n n n n n n n n n
 n n n n n n n n n n
 n n n n n n n n n n
 n n n n n n n n n n
 n n n n n n n n n n



V



V



B. Comparison With the Other Existing Methods

1) Methods in the Comparison Study:

-) $V + \dots$
-) $+ \dots$
-) $K-n \dots$ $K = 1$
-) $+ \dots$

-) \dots
- 2) Data Sets: V
- 3) Results: $V + \dots$

	Set-2	Set-3	Set-4	Set-5
SVD + KLD	45.60 ± 1.86	34.64 ± 2.05	64.62 ± 1.65	32.66 ± 1.10
HMT + MD	58.92 ± 2.91	56.08 ± 3.37	78.14 ± 2.93	n.a.
HMT + KNN	77.21 ± 2.02	75.73 ± 2.05	91.05 ± 1.36	n.a.
BP + L1	85.48 ± 1.98	83.91 ± 1.74	94.56 ± 0.84	78.03 ± 1.48
BC-PMC	91.60 ± 1.42	91.78 ± 1.38	97.70 ± 0.99	86.53 ± 1.12

	1	2	3	4	5		1	2	3	4	5
D1	99.38	100	93.13	58.13	48.75	D49	100	100	100	98.75	100
D3	98.75	90.63	73.75	37.50	39.38	D52	95.00	41.25	96.25	56.88	69.38
D4	100	98.75	82.50	72.50	53.75	D56	100	100	100	100	83.75
D6	100	100	100	95.63	96.88	D57	100	100	90.00	76.25	76.25
D8	100	100	98.13	62.50	56.25	D64	100	100	98.13	98.75	65.63
D9	100	96.88	84.38	83.13	30.63	D65	100	100	98.75	95.63	86.88
D10	99.38	86.25	98.75	95.00	68.13	D66	100	100	100	100	91.88
D11	100	100	100	93.13	39.38	D68	100	100	100	99.38	58.75
D15	93.75	86.88	94.38	75.63	83.75	D74	99.38	98.75	99.38	100	58.13
D17	100	100	93.13	80.63	78.75	D75	97.50	100	83.13	75.00	96.88
D18	100	100	95.00	76.25	75.00	D79	100	100	96.25	93.75	61.25
D20	100	100	98.75	89.38	75.63	D82	100	100	99.38	63.13	84.38
D21	100	100	91.88	20.00	100	D87	100	100	97.50	71.25	42.50
D22	95.00	84.38	86.88	75.63	81.88	D93	100	90.00	86.88	21.25	15.00
D25	95.63	93.13	100	99.38	45.00	D94	94.38	73.75	84.38	63.75	45.63
D26	100	99.38	100	99.38	49.38	D95	100	100	98.13	99.38	85.63
D28	100	100	96.88	80.63	45.00	D101	100	88.13	43.13	40.63	55.00
D32	100	100	93.13	89.38	55.63	D102	100	88.13	50.00	58.75	63.75
D33	100	99.38	96.88	88.75	48.13	D103	69.38	86.25	70.63	68.13	52.50
D34	100	100	100	94.38	93.75	D104	60.00	81.88	56.88	46.25	61.25
D35	100	100	75.00	36.88	43.13	D106	100	100	99.38	99.38	90.00
D37	100	89.38	98.75	74.38	53.75	D109	93.75	75.63	95.63	91.88	44.38
D46	98.13	100	98.75	88.75	61.88	D110	100	90.00	84.38	93.13	56.88
D47	100	100	94.38	74.38	75.00	D111	100	100	98.13	98.13	57.50
Mean	97.70	94.56	91.05	78.14	64.62						

C. Computational Cost

... X ... YY ... Y ...

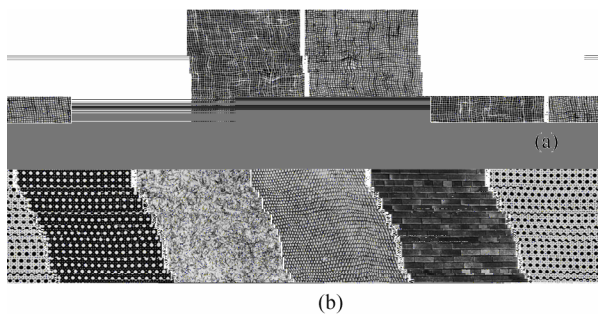


Fig. 0. (a) ... (b) ...

$$\left(\dots, \times 10^3 \right) \dots \mathbf{V}$$

IEEE Trans. Pattern Anal. Mach. Intell., 2000.

IEEE Trans. Image Process., 2000.

IEEE Trans. Image Process., 2000.

IEEE Trans. Image Process., 2000.

IEEE Trans. Image Process., 2000.

IEEE Trans. Commun., 2000.

IEEE Trans. Signal Process., 2000.

IEEE Trans. Image Process., 2000.

IEEE Trans. Image Process., 2000.

Int. Statist. Rev., 2000.

J. R. Statist. Soc. B, 2000.

J. Appl. Stat., 2000.

Ann. Math. Stat., 2000.

Pattern Recognit., 2000.

Pattern Recognit., 2000.

Pattern Recognit. Lett., 2000.

Int. J. Neural Syst., 2000.

Neural Netw., 2000.



Yongsheng Dong received the B.S. degree in 1998 and the M.S. degree in 2000 from the Department of Electronic Engineering, Tsinghua University, Beijing, China. He is currently a Ph.D. candidate in the Department of Electronic Engineering, Tsinghua University. His research interests include image processing and pattern recognition.



Jinwen Ma received the B.S. degree in 1998 and the M.S. degree in 2000 from the Department of Electronic Engineering, Tsinghua University, Beijing, China. He is currently a Ph.D. candidate in the Department of Electronic Engineering, Tsinghua University. His research interests include image processing and pattern recognition.