Nonnegative Multiresolution Representation Based Texture Image Classification

YONGSHENG DONG, Chinese Academy of Sciences
DACHENG TAO, Chinese Academy of Sciences
XUELONG LI, Chinese Academy of Sciences

Effective representation of image texture is important for an image classification task. Statistical modelling in wavelet domains has been widely used to image texture representation. However, due to the intra-class complexity and inter-class diversity of textures, it is hard to use a predefined probability distribution function to fit adaptively all wavelet subband coefficients of different textures. In this paper, we propose a novel modeling approach, namely, Heterogeneous and Incrementally Generated Histogram (HIGH), to indirectly model the wavelet coefficients by use of four local features in wavelet subbands. By concatenating all the HIGHs in all wavelet subbands of a texture, we can construct a nonnegative multiresolution vector (NMV) to represent a texture image. Considering the NMV's high dimensionality and nonnegativity, we further propose a Hessian regularized discriminative nonnegative matrix factorization to compute a low-dimensional basis of the linear subspace of NMVs. Finally, we present a texture classification approach by projecting NMVs on the low-dimensional basis. Experimental results show that our proposed texture classification method outperforms seven representative approaches.

Categories and Subject Descriptors: I.5.m [Pattern Recognition]: Miscellaneous; I.4.10 [Image Processing and Computer Vision]: Image Representation

General Terms: Theory, Experimentation

Additional Key Words and Phrases: Nonnegative matrix factorization, Texture classification, Histogram, Manifold regularization, Hessian regularization,
popularity, since the multiresolution and orientated representations of the wavelet transforms are consistent with the human perception of images [Do and Vetterli 2002; Li et al. 2010; Dong and Ma 2012].

In wavelet-based texture classification methods, a predefined probability distribution function is usually employed to model the wavelet subband coefficients. However, due to the diversity of image textures, as well as the complicated distribution of wavelet coefficients, the predefined probability distribution function cannot adaptively fit all the wavelet subband coefficients from different textures. It obviously affects the generalization of this modelling method. To alleviate this problem, in this paper, we propose a specific histogram called Heterogeneous and Incrementally Generated Histogram (HIGH) to model the local features of any wavelet subband instead of the subband coefficients, and construct a nonnegative multiresolution vector (NMV) to represent a texture image by concatenating the HIGHs of all the subbands.

As we all know, a good data representation can effectively reveal the latent structure of a dataset, and thus, be used to improve the performance and reduce the subsequent computational cost. They include low-rank representation methods [Shi et al. 2013a; Tang et al. 2011; Shi et al. 2013b], dictionary learning methods [Ding et al. 2015; Zhu and Shao 2014; Shao et al. 2014d], feature learning methods [Shao et al. 2014b; Shao et al. 2014c], and so on. Among those representations, nonnegative matrix factorization (NMF) is a simple and effective one, which decomposes the nonnegative data into two low-rank matrices under the nonnegativity constraints. Recently, NMF achieves more attention and is used for solve a variety of pattern recognition and computer vision problems, including consensus clustering [Li et al. 2007], blind spectral unmixing [Yang et al. 2011] and visual tracking [Wu et al. 2014]. Furthermore, manifold regularization is an effective trick for pattern analysis and representation [Shao et al. 2011; Gong et al. 2014], and is used for improving the performance of NMF [Guan et al. 2011]. However, the Laplacian matrix in graph regularization does not exploit the intrinsic local geometry of the data manifold.

For the purpose of exploit the intrinsic structure of NMVs, we in this paper propose a Hessian regularized discriminative nonnegative matrix factorization (HNMF) to process the NMVs due to their high dimensionality and nonnegativity. Finally, we present a texture classification approach by projecting NMVs on the low-dimensional basis obtained by HNMF. In particular, we utilize four local features to represent the information in the local neighborhood of each wavelet subband coefficient. By concatenating the HIGHs of all the subbands, we can further construct a NMV to represent a texture image. Then the HNMF is used to compute the low-dimensional basis of the linear subspace of NMVs. Finally, we can compute the low-dimensional representations of all the training samples and testing samples, and furthermore, use a minimum distance classifier to perform texture classification.

Particularly, four contributions are performed in this paper, which are given as follows. First, we propose a HIGH to effectively model local features of wavelet subbands. The HIGH is discriminative in distinguishing different textures. Moreover, it can effectively model the sparse data by using less bins than the histograms with a fixed small bin-width. Second, we present a novel HNMF by incorporating Hessian Regularization and discriminative information into NMF. In replace of computing the Laplacian matrix, we construct a block-diagram matrix to represent the between-class discriminative information and introduce it to NMF. Third, we can obtain a low-dimensional basis of linear subspace of the NMVs by using HNMF to construct the low-dimensional representation of training and testing textures. This can alleviate the wavelet subband imbalance problem. Finally, we present a texture classification method based on NMVs and HNMF. Our classification method has a better generalization than the other.
model-based classification methods due to the superior effectiveness of HIGH modeling. This is verified by our experimental results on several large texture datasets.

The remainder of the paper is organized as follows. Section 2 introduces the related work. In Section 3, we present nonnegative multiresolution representation of texture. Section 4 describes our proposed Hessian Regularized Discriminative Nonnegative Matrix Factorization. Section 5 presents the NMV based texture classification method with the HNMF. The experimental results presented in Section 6 demonstrate the effectiveness of our proposed image classification methods. Finally, we briefly conclude in Section 7.

2. RELATED WORK

2.1. Texture classification

The current existing texture classification methods can be broadly divided into two categories, namely, spatial-domain methods and transformed-domain methods, which are also referred to as filter-based methods or signal-processing methods. Spatial-domain methods can be further divided into three subcategories, namely, structural methods [Garcia-Sevilla and Petrou 1999], statistical methods [Haralick et al. 1973], and model-based methods [Speis and Healey 1996; Kayabol and Zerubia 2013]. In fact, spatial structural methods are based on regular or semiregular placements of textural primitives [Garcia-Sevilla and Petrou 1999; Henning et al. 2010; Ojala et al. 2002; Guo et al. 2010]. In the case of observable or visual textures, it is usually rather difficult to extract the primitives and their placements. Thus, these approaches are apt only for highly regular deterministic textures. Most of spatial statistical and model-based approaches for texture classification, such as gray-level co-occurrence matrices [Haralick et al. 1973] and Markov random field models [Speis and Healey 1996], are restricted to the analysis of spatial interactions over relatively small neighborhoods. Therefore, their performances are only good for the class of so-called microtextures.

To alleviate these essential problems of spatial-domain methods, transformed-domain methods have been proposed from a different point of view. Among them, texture classification methods based on the wavelet transform are popular in recent years [Choy and Tong 2007; Liu and Wang 2003]. The main reason is that the wavelet transform enables the decomposition of the image into different frequency subbands, being similar in the way the human visual system operates.

Wavelet transform-based methods can be divided into two subcategories, i.e., feature-based and model-based. The features used in wavelet domains include energy features [Manjunath and Ma 1996; Srinivasan and Srinivasan 2007; Laine and Fan 1993], statistical features [Van de Wouwer et al. 1999], random features [Liu and Fieguth 2012], and so on. The models used in wavelet domains include the generalized Gaussian density model [Do and Vetterli 2002], characteristic generalized Gaussian density [Choy and Tong 2007], the bit-plane probability model, the generalized Gamma density model [Choy and Tong 2010] and Gaussian mixture model [Kim and Kang 2007]. All of these methods need previously assume a probability distribution function to model the subband coefficients. It will cause the low generalization. In contrast, refined histogram (RH)-based method [Li et al. 2010] and local energy histogram (LEH)-based method [Dong and Ma 2011] alleviate this issue. But RH only is used to model the coefficients and thus neglect the information in the local neighborhood of each coefficient. Although LEH-based method alleviates this problem by modeling the first-order features in every local neighborhood, it neglects the second-order features (e.g., standard deviation) that represents the dispersion of wavelet coefficient values and are also important for texture representation. Meanwhile, the number of bins in the LEH is too big, which increases the subsequent computational cost.
2.2. Nonnegative matrix factorization
The NMF can decompose a matrix into two low-rank matrices. Intuitively, the NMF represent images having different distributions into parts. Recently, several NMF variants have been developed by introducing additional constraints to the original NMF. By incorporating the localization constraint, Li et al. [2001] presented the local NMF used to learn spatially localized, parts-based subspace representation for visual patterns. Hoyer [2002] proposed the nonnegative sparse coding by adding the s-sparseness constraint to compute sparse encoding vectors. In order to utilize the data geometric structure information, Cai et al. [2008] proposed a graph-regularized NMF (GNMF). In this approach, a nearest-neighbor (NN) graph is used to encode the data geometric structure. GNMF was specifically designed for clustering tasks, so it cannot perform well for classification problems. By incorporating the Fishers discriminative information, Zafeiriou et al. [2006] presented the discriminant NMF for frontal face verification. Yang et al. [2008] proposed the nonnegative graph embedding (NGE) that simultaneously learns two subspaces to incorporate the marginal Fisher discriminative information with NMF. However, NGE requires the matrix inverse operator in each iteration round. To improve the efficiency, Wang et al. [2009] presented a multiplicative algorithm for optimizing NGE. Liu et al. [2010] further developed PNGE that learns nonnegative projection matrix for dimension reduction. By introducing Laplacian regularization and margin maximization, Guan et al. [2011] recently proposed a Laplacian regularized discriminative nonnegative matrix factorization (LNMF). But the Laplacian matrix does not exploit the intrinsic local geometry of the data manifold. Moreover, in this approach, except for the Laplacian regularization, the discriminative mechanism is also implemented by computing a Laplacian matrix.

Fig. 1. The two texture classes (a) Fabric.0008, (b) Fabric.0009.

3. NONNEGATIVE MULTIRESOLUTION REPRESENTATION OF TEXTURE
Effective texture representation can improve the classification performance. In this section, we will present a nonnegative multiresolution representation representation method by modeling local features of wavelet subbands.

3.1. Heterogeneous and Incrementally Generated Histogram (HIGH) of Local Features in Wavelet Subbands
3.1.1. Local Feature Extraction. Modeling wavelet subband coefficients by a predefined probability distribution is usually used for texture classification. However, the classification performance resulting from these modeling methods is usually not satisfactory. The reasons are two-folds. First, the predefined probability distribution function may
Fig. 2. The HIGHs (B=9) corresponding to the four local features of the two textures: Fabric.0008 and Fabric.0009. In the left column, the four figures from top to bottom respectively represent the HIGHs of local mean, local median, local sample standard deviation, and local norm-2 energy of Fabric.0008, and in the right column, the four figures from top to bottom respectively represent the HIGHs of local mean, local median, local sample standard deviation, and local norm-2 energy of Fabric.0009.

not be effectively fitted to the wavelet subband coefficients. Second, the local infor-
Information of each subband coefficient does not used for classification. To alleviate these problems, we propose a novel histogram modeling method to model local features of wavelet subbands in this subsection.

We first use four statistics to describe the statistical characterization of the coefficients in the local $S \times S$ neighborhood $N_S$ of each coefficient. Typically, in the $i$-th wavelet subband of size $\Omega_i \times \Omega_i$, we extract the following four local features on $S \times S$ neighborhoods.

- Sample mean: $e_1^i = \frac{1}{N_i} \sum_{\tau=1}^{N_i} \omega_\tau$.
- Median: $e_2^i = \frac{1}{2} \left( \tilde{\omega}_{n/2} + \tilde{\omega}_{n/2+1} \right)$ if $N_i$ is even, otherwise, $e_2^i = \tilde{\omega}_{n/2}$ where $\tilde{\omega}_n$ is the $n$-th order statistic from the sample $V_i$.
- Sample standard deviation: $e_3^i = \left( \frac{1}{N_i} \sum_{\tau=1}^{N_i} (\omega_\tau - \tilde{\omega}_n)^2 \right)^{1/2}$.
- Norm-2 energy: $e_4^i = \left( \frac{1}{N_i} \sum_{\tau=1}^{N_i} \omega_\tau^2 \right)^{1/2}$.

where $\omega_\tau \in V_i$, $V_i$ is the set consisting of the absolute coefficients in the neighborhood $N_S$ of the $i$-th subband, and $N_i$ is the number of coefficients in $V_i$. Note that all the above local features are non-negative. Note that both $e_1^i$ and $e_2^i$ measure the one-order information of coefficients in $N_S$. Due to that $e_3^i$ is the square root of sample variance (two-order central moment) and $e_4^i$ is the square root of two-order origin moment, both $e_3^i$ and $e_4^i$ measure the degree of dispersion of coefficients in $N_S$.

### 3.1.2. Heterogeneous and Incrementally Generated Histogram (HIGH).

Generally speaking, the histogram with a fixed bin width represents the distribution of a given dataset [Dong and Ma 2011; Shao et al. 2014a]. However, due to that the four local features are sparse, resulting from the sparsity of wavelet coefficients, we need a histogram with more bins to model them in order to capture the distribution rule of them, but the subsequent computational cost will be more. To alleviate this problem, we here present a novel histogram, which is defined as follows.

In a particular wavelet subband, for a local feature variable with $M$ observations $E = (e_1, e_2, \cdots, e_M)$, there must exist an positive integer $N_0$ such that the maximum in $E$ is less than or equal to $2^{N_0}$. So we construct $\Delta_k$, $k = 1, 2, \cdots, B$ such that $\Delta_1 \leq \Delta_2 \leq \cdots \leq \Delta_B$ and $[0, 2^{N_0}] = \bigcup_{k=1}^{B} \Delta_k$. It follows that, given any $e_1$, there must exist a positive integer $n_0$ ($1 \leq n_0 \leq B$) such that $e_1 \in \Delta_{n_0}$. Then we define a discrete function as

$$p(e) = \eta_k, e \in \Delta_k$$

where $\eta_k$ is the number of the local features appearing in $\Delta_k$.

In fact, the function defined by (1) is a frequency histogram. Moreover, its bin-width is heterogeneous and incrementally generated, so we call it the Heterogeneous and Incrementally Generated Histogram (HIGH). Note that the HIGH can effectively model the sparse data by using less bins than the histograms with a fixed small bin-width. In this paper, we specify the HIGH and its bins as follows.

$$p(e) = \begin{cases} \eta_1, & 0 \leq e \leq 2 \\ \eta_k, & 2^{k-1} < e \leq 2^k, k = 2, \cdots, B \end{cases}$$

where $B$ is the number of bins of the HIGH.

Figure 1 shows two VisTex texture images Fabric.0008 and Fabric.0009, obtained from the VisTex database and used to represent two texture classes. While a 3-level wavelet transform is implemented on each of them, we can compute the HIGHs in each wavelet subband. Figure 2 shows their HIGHs of the horizontal subband at 2-scale-scale. For clarity, the number of bins is selected as 9, that is, $B=9$. It can be
observed that the four HIGHs of Fabric.0008 are different from those of Fabric.0009. In other words, each of the four HIGHs has the discriminative ability to distinguish the two texture classes.

3.2. Nonnegative Multiresolution Representation

Given a texture image, we perform an L-level wavelet decomposition to it, and obtain $3L + 1$ wavelet subbands. In $i$-th wavelet subband, we compute the four local features, and then compute their HIGH feature vectors, denoted by $h_1^i = (\eta_{1,1}^i, \eta_{1,2}^i, \cdots, \eta_{1,B}^i)$ and $h_2^i = (\eta_{2,1}^i, \eta_{2,2}^i, \cdots, \eta_{2,B}^i)$, $h_3^i = (\eta_{3,1}^i, \eta_{3,2}^i, \cdots, \eta_{3,B}^i)$ and $h_4^i = (\eta_{4,1}^i, \eta_{4,2}^i, \cdots, \eta_{4,B}^i)$, respectively. So we can use $v_i = (h_1^i, h_2^i, h_3^i, h_4^i)$ to represent the $i$-th subband. Furthermore, we construct the following vector

$$v = (v_1, v_2, \cdots, v_{3L+1})^T$$

(3)

to represent the texture. Note that $v$ is a column vector, we use this expression only for simplicity. So, we characterize the texture by $v$, which is referred as a nonnegative multiresolution vector (NMV).

In the following section, we will consider the NMVs in a manifold and compute the low-dimensional basis of the linear subspace of NMVs of textures in a given dataset.

4. HESSIAN REGULARIZED DISCRIMINATIVE NONNEGATIVE MATRIX FACTORIZATION

Given $n$ sample points $\{v_1, v_2, \cdots, v_n\}$ in $R^n$, whose elements are all nonnegative, we can obtain a matrix $V = (v_1, v_2, \cdots, v_n) \in R_{+}^{m \times n}$. Nonnegative matrix factorization (NMF) aims to find two nonnegative matrices $W \in R_{+}^{m \times r}$ and $F \in R_{+}^{r \times n}$, with $r < \min\{m, n\}$, to approximate the original matrix, that is, $V \approx WF$. To this end, NMF algorithm can be formulated as

$$\min_{W \geq 0, F \geq 0} KLD(V, WF).$$

(4)

Recently, a variety of NMF's variants have been developed by modifying the measurement between $V$ and $WF$ or further introducing additional constraints to the resulting minimization problems. For the pattern classification task, it is worth to mention that Guan et al. [2011] proposed a Laplacian regularized discriminative nonnegative matrix factorization. In this approach, both the Laplacian regularization (LR) and the discriminative mechanism are implemented by computing a Laplacian matrix. However, the Laplacian matrix does not effectively exploit the intrinsic local geometry of the data manifold [Kim et al. 2009]. Moreover, in this approach, except for the Laplacian regularization, the discriminative mechanism is also implemented by computing a Laplacian matrix. As a novel regularization method, Hessian Regularization (HR) can not only properly fit the data within the domain defined by training samples, but it can also nicely predict the data points beyond the boundary of the domain. Moreover, Hessian has a richer nullspace and drives the learned function which varies linearly along the underlying manifold [Donoho and Grimes 2003; Florian and Matthias 2009]. Compared with LR, HR has been demonstrated to be more effective for kernel regression [Kim et al. 2009].

Motivated by the advantages of HR, we propose a Hessian regularized discriminative nonnegative matrix factorization, which encodes the data geometric structure and the discriminative information of different classes in the parts-based representation. The difference from the LNMF is two folds. First, we incorporate the geometrical structure into NMF by using the Hessian manifold. Second, in replace of Laplacian matrix, we construct a block-diagram matrix to represent the between-class discriminative information and introduce it to NMF. In the following subsections, we will present the objective function and its optimization algorithms.
4.1. Objective Function

From the perspective of dimension reduction, NMF learns to represent original samples \( V = \{v_1, v_2, \cdots, v_n\} \) denoted by \( m \)-dimensional nonnegative vectors \( v_i \in \mathbb{R}_+^m \) as a linear combination of low-dimensional basis denoted by \( W \in \mathbb{R}_{+}^{m \times r} \), i.e., \( V = WF \), wherein \( F \in \mathbb{R}_{+}^{r \times n} \) is the nonnegative coefficient matrix. Thus, it can be deemed as a function \( g(v) = f \), subject to \( v = Wf \), to preserve the local geometry of the distribution of samples \( V \). Suppose \( V \) is sampled from a probability distribution on a manifold \( M \) embedded in a high-dimensional ambient space, and thus we can construct the Hessian energy \( S_{\text{Hessian}}(g) \) [Kim et al. 2009] as a regularization functional on the manifold \( M \). However, the manifold is unknown in practice, so we approximate \( S_{\text{Hessian}}(g) \) as

\[
S_{\text{Hessian}}(g) \approx \text{tr}(HFHT),
\]

where \( H \) is the accumulated Hessian matrix. Refer to [Kim et al. 2009] for the details.

We next construct our objective function by three steps. First, we use Frobenius norm to measure the difference between \( V \) and \( WF \) instead of the traditional Kullback-Leibler divergence, and employ the estimator of \( S_{\text{Hessian}}(g) \) as the regularizer. Then we can obtain

\[
\min_{W \geq 0, F \geq 0} \|V - WF\|^2 + \frac{\gamma}{2} \text{tr}(HFHT). \tag{6}
\]

Second, to obtain a lower dimensional basis, that is, suppress the over decomposition of the basis matrix \( W \), we penalize the objective function in the above equation by

\[
\text{tr}(W^TW). \tag{7}
\]

Finally, in order to encode the discriminative information, we minimize the within-class dispersion degree, which can be measured with

\[
\text{tr}(FCF^T) \tag{8}
\]

where

\[
C = \begin{pmatrix}
C_1 \\
C_2 \\
\vdots \\
C_n
\end{pmatrix}
\]

where \( C_i \) is the sample covariance of the \( i \)-th class.

By combining (6)-(8), we arrive at the objective function of the proposed HNMF

\[
 f(W, F) = \|V - WF\|^2 + \frac{\alpha}{2} \text{tr}(W^TW) + \frac{\beta}{2} \text{tr}(FCF^T) + \frac{\gamma}{2} \text{tr}(HFHT). \tag{10}
\]

where \( \alpha \) and \( \beta \) are the trade-off parameters of the low-dimensionality and the discriminative regularization. We use 1/2 over \( \alpha, \beta \) and \( \gamma \) only for simplifying the following deductions.

4.2. The Update Rule

For the objective function defined in (10), we can update \( W \) and \( F \) by solving the following minimization problem

\[
\min_{W, F \geq 0} f(W, F). \tag{11}
\]

In fact, the above optimization problem can be solved by alternatively solving the following two subproblems until convergence:
we only need to prove 

\[
\begin{aligned}
F_k + 1 &= \arg \min_{F \geq 0} f(W_k, F) \\
W_k + 1 &= \arg \min_{W \geq 0} f(W, F_{k+1}).
\end{aligned}
\]  

For each of the above minimization problems, we can compute the gradients of 
\(f(W, F)\) with respect to \(F\) and \(W\) as

\[
\begin{aligned}
\nabla_F f(W, F) &= W^T W F - (W^T V) + \beta FC + \gamma FH \\
\nabla_W f(W, F) &= W F F^T - (V F^T) + \alpha W.
\end{aligned}
\]  

We then obtain Multiplicative Update Rule (MUR) for HNMF as

\[
\begin{aligned}
F &= F \odot \frac{(W^T V)}{W^T W F + \beta FC + \gamma FH} \\
W &= W \odot \frac{(V F^T)}{W F F^T + \alpha W}.
\end{aligned}
\]

In the following subsection, we will prove the objective function \(f(W, F)\) is nonincreasing under both update (15) and (16). So we can obtain a local solution of (12) by alternatively using (15) with \(W\) fixed and using (16) with \(F\) fixed.

4.3. Convergence Analysis

To prove the convergence of the above algorithm, we first introduce an auxiliary function, which is defined as follows.

**Definition 4.1.** The function \(G(x, x')\) is an auxiliary function for \(F(x)\), if \(F(x) \leq G(x, x')\) and \(F(x) = G(x, x)\).

Based on the definition of auxiliary function, we can easily obtain the following lemma.

**Lemma 4.2.** IF \(G(x, x')\) is an auxiliary function of \(F(x)\), then \(F(x)\) is nonincreasing under the update \(x_{t+1} = \arg \min_x G(x, x')\).

According to the above lemma, we have Theorem 4.3 that the objective function \(f(W, F)\) is nonincreasing under the alternative update rules.

**Theorem 4.3.** The objective function \(f(W, F)\) is nonincreasing under update (15) with \(W\) fixed and under (16) with \(F\) fixed.

**Proof.** In the iteration round \(k\), given \(W_k\), we construct an auxiliary function for \(f(W_k, F)\) as

\[
G(F, F_k) = f(W_k, F_k) + \langle \nabla F f(W_k, F_k), F - F_k \rangle + \left\{ \frac{W_k^T W_k F_k + \beta F_k C + \gamma F_k H}{2 F_k} (F - F_k)^2 \right\}
\]

where \((Y)^2\) is the elementwise square of the matrix \(Y\). Since \(G(F_k, F_k) = f(W_k, F_k)\), we only need to prove \(f(W_k, F) \leq G(F, F_k)\). To this end, we compute the Taylor series expansion of \(f(W_k, F)\) at \(F_k\), and then the \((i, j)\)th element of \(F\) is

\[
f(W_k, F_{ij}) = f(W_k, F_{kij}) + \langle \nabla F f(W_k, F_k), F_{ij} - F_{kij} \rangle + \frac{1}{2} (W_k^T W_k + \beta C + \gamma H)_{ij} (F_{ij} - F_{kij})^2.
\]

Due to that \(W_k \geq 0\) and \(H_k \geq 0\), we have

\[
(W_k^T W_k)_{ij} \leq \frac{\sum (W_k^T W_k)_{ij} F_{kij}}{F_{kij}} = \frac{W_k^T W_k F_k}_{ij}
\]
Meanwhile, we have

\[
\beta C + \gamma H \leq \frac{\sum_i F_{ki}(\beta C + \gamma H)_{ij}}{F_{ki}}.
\]

(19)

So we prove that \(f(W_k, F_k) \leq G(F_j, F_k)\) and thus \(G(F, F_k)\) is an auxiliary function of \(f(W, F)\). According to the Lemma 1 [Guan et al. 2011], \(f(W_k, F)\) is non-increasing at the minimum of \(G(F, F_k)\). Let \(\partial_F G(F, F_k) = 0\) and we can obtain \(F_{k+1} = F_k \odot \frac{W_k^T W_k F_k + \beta F_k C + \gamma F_k H}{W_k^T W_k F_k + \beta F_k C + \gamma F_k H} - F_k\) and thus (15) does not increase the objective function \(f(W, F)\). Based on the same method, we can prove that (16) does not increase the objective function \(f(W, F)\) with \(F\) fixed. □

4.4. Fast Gradient Descent

Due to that MUR converges slowly, we construct the following fast gradient descent update rules of \(F\) and \(W\) at the iteration round \(k\) for HNMF,

\[
F_{k+1} = F_k + \theta_F^{k+1} \nabla_{F_k} f(W_k, F_k) \quad \text{(20)}
\]

\[
W_{k+1} = W_k + \theta_W^{k+1} \nabla_{W_k} f(W_k, F_{k+1}) \quad \text{(21)}
\]

where

\[
\nabla_{F_k} f(W_k, F_k) = F_k \odot \frac{(W_k^T V)}{W_k^T W_k F_k + \beta F_k C + \gamma F_k H} - F_k \quad \text{(22)}
\]

\[
\nabla_{W_k} f(W_k, F_{k+1}) = W_k \odot \frac{(V F_{k+1}^{T})}{W_k F_{k+1} F_{k+1}^{T} + \alpha W_k} - W_k \quad \text{(23)}
\]

and the step lengths \(\theta_F^{k+1}\) and \(\theta_W^{k+1}\) are selected by the same way as in [Guan et al. 2011]. Note that MUR in (15) and (16) can be respectively looked on as a special case of (20) and (21) by setting \(\theta_F^{k+1} = 1\) and \(\theta_W^{k+1} = 1\).

5. NMV-BASED TEXTURE CLASSIFICATION VIA HNMF

In this section, we will present a NMV-based texture classification method. In particular, given a set of training samples, we first perform a L-scale wavelet decomposition on each training texture. By computing the HIGHs of each wavelet subband, we then obtain the NMV representation of each texture. Considering the high dimension and nonnegativity of the NMVs, we use the HNMF to compute the low-dimensional basis of linear subspace of NMVs of training textures. Figure 3 shows the flow chart of computing the low-dimensional basis \(W\) of the linear subspace of NMVs \(V = \{v_1, v_2, \cdots, v_n\}\) of a training texture dataset. After that, the NMV \(v_i\) of each training texture image is projected into the linear space, and a feature vector \(\hat{f}_i = W^{-1} v_i\) is obtained as a compact representation of the training texture. Similarly, a test texture image \(q\) to be classified is compactly represented by the projection of its NMV \(v^q\) into the space as \(\hat{f} = W^{-1} v^q\). The Euclidean distance between the test and each training sample, \(d(\hat{f}, \hat{f}_i)\), is calculated. The test texture is classified to the class to which the closest training texture belongs.

As we all know, different wavelet subbands have different contributions for texture classification, which can be named as the wavelet subband imbalance problem. For the purpose of convenience, many early texture classification methods treat all the subbands equally by directly summing the distances between the corresponding subbands of two textures. This is an important reason that they do not have satisfactory classification performances. However, in our proposed texture classification method, we avoid...
Fig. 3. The flow chart of computing the low-dimensional basis $W$ of the linear subspace of NMVs $V = \{v_1, v_2, \ldots, v_n\}$ of a training texture dataset, where HNMF denotes our proposed Hessian regularized discriminative nonnegative matrix factorization.

to directly process the NMVs obtained from wavelet subbands by computing a low-dimensional basis of the linear subspace of NMVs. Based on the low-dimension basis, we can obtain the low-dimension representations of training and testing textures for classification. So our proposed method can alleviate the subband imbalance problem. In addition, if we look on wavelet subbands as subspaces in wavelet domains, then sub-space selection methods [Tao et al. 2006; Tao et al. 2007; Tao et al. 2009] can be used for settling the subband imbalance problem. On the other hand, the distance from texture to class is important for classification. As a learning scheme to the distance, the metric learning method has been used for pattern recognition [Zhai et al. 2012; Zhang and Yeung 2012; Wang et al. 2013]. We will discuss the metric learning based texture classification in future work.

6. EXPERIMENTAL RESULTS

Here we perform various experiments to demonstrate the efficiency of our proposed texture classification methods. For these experiments, we select Daubechies 1 (db1)\(^1\) as the wavelet function. The number $B$ of bins in HIGHs is selected as $B = 10$ in our experiments. In fact, for texture images, the wavelet subband coefficients are generally smaller than $2^{10}$. Certainly, we can also use a bigger value for $B$, but it will lead to more computational cost. In the following subsection, we will evaluate our proposed texture classification method based on HIGH and HNMF. For clarity, we refer to our proposed texture classification method based on HIGH and HNMF as H-HNMF.

\(^1\)In addition to db1, we also considered db2, db6, and db110 for our experiments, and the results are similar.
The ACRs on Set-1 of the two methods H-LNMF and H-HNMF with different subspace dimensions when the wavelet decomposition scale varies from 1 to 4. (a) H-LNMF, (a') H-HNMF.

6.1. Performance Evaluation
In this subsection, we evaluate our texture classification method on the typical set of 30 512 × 512 VisTex texture images (denoted by Set-1, and shown in Figure 4). The dataset was used in [Arivazhagan et al. 2006] and [Dong and Ma 2013]. In this experiment, each image is divided into 64 × 64 nonoverlapping patches, and thus there are 1290 samples available in total. We select 32 training samples from each of 30 classes and leave the remaining samples for testing. To measure the classification accuracy, we define an average classification accuracy rate of the $c$-th class as

$$ACAR^c = \frac{1}{\Gamma} \sum_{\tau=1}^{\Gamma} \frac{A^\tau,c}{B^\tau,c}$$

where $\Gamma=10$ is the number of random splits of the training and test sets, $A^\tau,c$ is the number of the correctly classified $c$-th test samples in the $\tau$-th split, and $B^\tau,c$ is the total number of the $c$-th test samples in the $\tau$-th split. The average classification accuracy rate for a dataset is the average value of the $ACAR^c$s, and denoted by ACAR for simplicity.
In addition, in order to verify the effectiveness of our proposed the Hessian Regularized discriminative Nonnegative Matrix Factorization (HNMF), we compare HNMF with the Laplacian regularized discriminative Nonnegative Matrix Factorization (LNMF) [Guan et al. 2011] on our HIGHs features. That is, we replace the HNMF in our classification method by the LNMF to obtain a new texture classification method, denoted by H-LNMF.

We first investigate the sensitivity of decomposition scale \( L \) to classification performance. Figure 5 shows the ACARs of H-LNMF and H-HNMF with respect to the dimension of low-dimensional subspace. The error bars are also plotted in Figure 5, where each error bar represents the standard deviation of the ACAR. It can be seen from Figure 5 (a) that the ACAR of H-LNMF with \( L = 1 \) decreases rapidly with the increasing of the dimension of subspace from 20 to 50. Meanwhile, the ACARs of H-LNMF with \( L = 2 \) outperforms those with \( L = 1, 3 \) and 4. The average of error bars of H-HNMF with \( L = 2 \) are notably smaller than those with \( L = 1, 3 \) and 4. This implies that the reasonable number \( L \) of decomposition scales for H-LNMF should be 2. Furthermore, it can be seen from Figure 5(b) that the ACARs of H-HNMF with \( L = 1 \) and \( L = 2 \) outperforms those with \( L = 3 \) and \( L = 4 \). The average of error bars of H-HNMF with \( L = 1 \) and \( L = 2 \) are notably smaller than those with \( L = 3 \) and 4. This implies that the reasonable number \( L \) of decomposition scales for H-HNMF should be 1 or 2. So, considering the two methods H-LNMF and H-HNMF, we select the number \( L = 2 \) in our comparison with other methods. The optimality of \( L \) will also be verified in the following experiments on other datasets.

We further evaluate our proposed H-HNMF by comparing it with H-LNMF, as well as the four recently developed methods described as follows. The first method is based on the bit-plane probability model and minimum distance classifier (BP-MD), proposed in [Choy and Tong 2008]. The second method is based on the local energy histograms and one-nearest-neighbor classifier (LEH-NN) [Dong and Ma 2011]. The third is the method based on the Poisson mixtures in the contourlet domains and the Bayesian classifier (PMC-BC), proposed in [Dong and Ma 2012]. The fourth is the method based on contourlet subband clustering and the k-nearest neighbor classifier (CSC-NN), proposed in [Dong and Ma 2013].

The second column of Table I reports the ACARs of the six texture classification methods performed on Set-1 in the case of 32 training samples. As seen in the second column, our proposed H-HNMF outperforms the other five methods, and its standard deviation is visibly less than the other five methods.

### Table I. The average classification accuracy rates of the five methods (BP-MD, LEH-NN, PMC-BC, CSC-NN, H-LNMF(\(r=30\)) and H-HNMF(\(r=30\))) on the five texture datasets.

<table>
<thead>
<tr>
<th></th>
<th>Set-1</th>
<th>Set-2</th>
<th>Set-3</th>
<th>Set-4</th>
<th>Set-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP-MD</td>
<td>93.31 ± 0.62</td>
<td>76.77 ± 0.82</td>
<td>69.02 ± 0.82</td>
<td>90.57 ± 0.60</td>
<td>69.78 ± 0.88</td>
</tr>
<tr>
<td>LEH-NN</td>
<td>95.22 ± 0.59</td>
<td>82.77 ± 0.71</td>
<td>73.06 ± 0.52</td>
<td>94.89 ± 0.32</td>
<td>77.32 ± 0.37</td>
</tr>
<tr>
<td>PMC-BC</td>
<td>95.49 ± 0.70</td>
<td>83.50 ± 0.78</td>
<td>74.25 ± 0.53</td>
<td>95.60 ± 0.73</td>
<td>77.96 ± 0.77</td>
</tr>
<tr>
<td>CSC-NN</td>
<td>97.01 ± 0.38</td>
<td>86.23 ± 0.86</td>
<td>78.76 ± 1.01</td>
<td>97.22 ± 0.22</td>
<td>80.07 ± 0.35</td>
</tr>
<tr>
<td>H-LNMF</td>
<td>98.77 ± 0.25</td>
<td>90.35 ± 0.51</td>
<td>82.65 ± 0.83</td>
<td>98.37 ± 0.26</td>
<td>83.35 ± 0.55</td>
</tr>
<tr>
<td>H-HNMF</td>
<td><strong>99.30 ± 0.22</strong></td>
<td><strong>93.50 ± 0.55</strong></td>
<td><strong>86.19 ± 0.72</strong></td>
<td><strong>98.77 ± 0.30</strong></td>
<td><strong>87.65 ± 0.61</strong></td>
</tr>
</tbody>
</table>

### 6.2. Further Comparison with Other Methods

#### 6.2.1. Comparison Using Larger VisTex Datasets

In this subsection, we compare our method with BP-MD, LEH-NN, PMC-BC, CSC-NN and H-LNMF using two larger VisTex datasets. The first is the dataset consisting of 40 \(512 \times 512\) VisTex texture images...
Fig. 6. The ACRs of H-LNMF and H-HNMF with different subspace dimensions when the wavelet decomposition scale varies from 1 to 4. (a) H-LNMF on Set-2. (a’) H-HNMF on Set-2. (b) H-LNMF on Set-3. (b’) H-HNMF on Set-3.

(denoted by Set-2), which was used in [Do and Vetterli 2002]. The second consists of 60 VisTex texture images (denoted by Set-3). Each image is divided into 64 × 64 non-overlapping patches, and thus there are 2560 and 3840 samples available in total, respectively. 32 texture patches in each class are selected for training, and the other patches in each class are used for testing. The ACRs of these methods are listed in Table I. It can be seen that H-HNMF outperforms BP-MD, LEH-NN, PMC-BC, CSC-NN and H-LNMF by over 3.15% on these two VisTex datasets. We also see that our H-HNMF has a better generalization than the other model-based classification methods (BP-MD, LEH-NN and PMC-BC). This is due to the superior effectiveness of HIGH modeling. Figure 6 shows the ACRs of H-HNMF and H-LNMF with respect to the dimension of low-dimensional subspace. It can be seen from Figure 6 that the ACRs of H-HNMF and H-LNMF have the similar rules with those observed in Figure 5.

For a more intensive comparison with BP-MD, LEH-NN, PMC-BC, CSC-NN and H-LNMF, the classification accuracy rates of all 40 texture classes in Set-2 are shown in Table II. It can be seen that our method does not perform worse than the

four methods (BP-MD, LEH-NN, PMC-BC and CSC-NN) for 27 texture classes. It is worth mentioning that our method outperforms the four methods on Bark.0000, Bark.0008, Bark.0009, Brick.0005, Leaves.0011, and Leaves.0016 by over 10%. Figure 7 show these six textures. It is worth to note that H-HNMF outperforms H-LNMF on 33 texture classes except for the five texture classes on which they have the same ACARs. The worst classification accuracy rates of the six methods, BP-MD, LEH-NN, PMC-BC, CSC-NN, H-LNMF, and H-HNMF, are reached on Bark.0009 (20.94%), Leaves.0010 (36.88%), Bark.0009 (14.69%), Bark.0009 (53.44%), Leaves.0010 (61.25%), and Leaves.0010 (70.00%), respectively. As far as the ACAR for the whole data set is concerned, our proposed H-HNMF performs better than the five methods by 3.15% – 16.73%.

6.2.2. Comparison Using Brodatz Datasets. In this subsection, we compare H-HNMF with the five methods presented above on two larger texture datasets obtained from the Brodatz database. The first consists of 60 Brodatz texture images (denoted by Set-4, shown in Figure 8). The second large data set is the whole Brodatz texture dataset, consisting of 111 texture images (denoted by Set-5). All the texture images in these two datasets are of size 640 × 640, and each is divided into 64 80 × 80 non-overlapping patches. Thus there are 3840 and 7104 samples available in total, respectively. 32 texture patches in each class are selected for training, and the other patches in each class are used for testing. The ACARs of these methods are listed in Table I. It can be seen that HIGH-NMF slightly outperforms BP-MD, LEH-NN, PMC-BC, CSC-NN and H-LNMF on the Set-4, but by over 4.30% outperforms the five methods on the Set-5. These also verify that our H-HNMF has a better generalization than the other model-based classification methods (BP-MD, LEH-NN and PMC-BC). Figure 9 shows the ACARs of H-HNMF and H-LNMF with respect to the dimension of low-dimensional subspace. It can be seen from Figure 9 that the ACARs of H-HNMF and H-LNMF have the similar rules with those in Figure 5 and Figure 6.

6.2.3. Comparison on the CUREt Dataset. To further demonstrate the efficiency of H-HNMF, we provide an addition comparison with the recent random projection method (RP) [Liu and Fieguth 2012] and the dominant local binary pattern method (DLBP) [Liao et al. 2009]. The dataset used in this comparison is the CUREt dataset, which
Table II. The average classification accuracy rate (%) for each of 40 texture classes in Set-2 with the five methods: Column 1: BP-MD, Column 2: LEH-NN, Column 3: PMC-BC, Column 4: CSC-NN, Column 5: H-LNMF(r=30), Column 6: Our proposed H-HNMF (r=30)

<table>
<thead>
<tr>
<th>Texture classes</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bark.0000</td>
<td>50.31</td>
<td>71.56</td>
<td>56.56</td>
<td>74.69</td>
<td>94.06</td>
<td>97.81</td>
</tr>
<tr>
<td>Bark.0006</td>
<td>66.56</td>
<td>66.87</td>
<td>77.50</td>
<td>72.19</td>
<td>78.13</td>
<td>86.25</td>
</tr>
<tr>
<td>Bark.0008</td>
<td>42.19</td>
<td>65.00</td>
<td>62.81</td>
<td>53.75</td>
<td>71.56</td>
<td>74.06</td>
</tr>
<tr>
<td>Bark.0009</td>
<td>20.94</td>
<td>45.62</td>
<td>14.69</td>
<td>54.34</td>
<td>62.50</td>
<td>76.25</td>
</tr>
<tr>
<td>Brick.0001</td>
<td>99.69</td>
<td>90.63</td>
<td>95.31</td>
<td>93.13</td>
<td>97.19</td>
<td>97.50</td>
</tr>
<tr>
<td>Brick.0004</td>
<td>94.06</td>
<td>90.31</td>
<td>82.50</td>
<td>96.56</td>
<td>97.50</td>
<td>99.06</td>
</tr>
<tr>
<td>Brick.0005</td>
<td>81.56</td>
<td>65.00</td>
<td>69.06</td>
<td>77.19</td>
<td>85.00</td>
<td>93.13</td>
</tr>
<tr>
<td>Buildings.0009</td>
<td>98.75</td>
<td>95.00</td>
<td>96.88</td>
<td>96.25</td>
<td>97.19</td>
<td>98.12</td>
</tr>
<tr>
<td>Brick.0000</td>
<td>90.63</td>
<td>99.69</td>
<td>98.75</td>
<td>100</td>
<td>99.38</td>
<td>100</td>
</tr>
<tr>
<td>Brick.0004</td>
<td>94.06</td>
<td>90.31</td>
<td>82.50</td>
<td>96.56</td>
<td>97.50</td>
<td>99.06</td>
</tr>
<tr>
<td>Brick.0005</td>
<td>81.56</td>
<td>65.00</td>
<td>69.06</td>
<td>77.19</td>
<td>85.00</td>
<td>93.13</td>
</tr>
<tr>
<td>Brick.0011</td>
<td>65.63</td>
<td>75.00</td>
<td>90.00</td>
<td>85.00</td>
<td>89.38</td>
<td>95.31</td>
</tr>
<tr>
<td>Brick.0014</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Brick.0015</td>
<td>92.19</td>
<td>95.00</td>
<td>94.69</td>
<td>97.19</td>
<td>94.69</td>
<td>95.31</td>
</tr>
<tr>
<td>Fabric.0000</td>
<td>90.63</td>
<td>99.69</td>
<td>98.75</td>
<td>100</td>
<td>99.38</td>
<td>100</td>
</tr>
<tr>
<td>Fabric.0004</td>
<td>94.06</td>
<td>90.31</td>
<td>82.50</td>
<td>96.56</td>
<td>97.50</td>
<td>99.06</td>
</tr>
<tr>
<td>Fabric.0007</td>
<td>85.94</td>
<td>98.12</td>
<td>98.75</td>
<td>100</td>
<td>98.68</td>
<td>99.06</td>
</tr>
<tr>
<td>Fabric.0009</td>
<td>96.88</td>
<td>93.44</td>
<td>100</td>
<td>96.88</td>
<td>96.88</td>
<td>98.12</td>
</tr>
<tr>
<td>Fabric.0011</td>
<td>65.63</td>
<td>75.00</td>
<td>90.00</td>
<td>85.00</td>
<td>89.38</td>
<td>95.31</td>
</tr>
<tr>
<td>Fabric.0014</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Fabric.0015</td>
<td>92.19</td>
<td>95.00</td>
<td>94.69</td>
<td>97.19</td>
<td>94.69</td>
<td>95.31</td>
</tr>
<tr>
<td>Fabric.0017</td>
<td>100</td>
<td>90.63</td>
<td>100</td>
<td>100</td>
<td>99.06</td>
<td>99.69</td>
</tr>
<tr>
<td>Fabric.0018</td>
<td>100</td>
<td>99.69</td>
<td>100</td>
<td>100</td>
<td>94.69</td>
<td>99.38</td>
</tr>
<tr>
<td>Flowers.0005</td>
<td>74.69</td>
<td>92.81</td>
<td>74.69</td>
<td>83.44</td>
<td>95.63</td>
<td>97.50</td>
</tr>
<tr>
<td>Food.0000</td>
<td>90.94</td>
<td>86.56</td>
<td>87.50</td>
<td>85.62</td>
<td>90.31</td>
<td>93.44</td>
</tr>
<tr>
<td>Food.0004</td>
<td>44.69</td>
<td>92.50</td>
<td>75.94</td>
<td>95.94</td>
<td>96.25</td>
<td>97.19</td>
</tr>
<tr>
<td>Food.0008</td>
<td>80.31</td>
<td>64.38</td>
<td>91.87</td>
<td>86.88</td>
<td>90.94</td>
<td>98.12</td>
</tr>
<tr>
<td>Grass.0001</td>
<td>83.75</td>
<td>85.00</td>
<td>94.69</td>
<td>92.50</td>
<td>89.38</td>
<td>87.81</td>
</tr>
<tr>
<td>Leaves.0008</td>
<td>46.25</td>
<td>74.06</td>
<td>83.44</td>
<td>69.37</td>
<td>79.69</td>
<td>89.06</td>
</tr>
<tr>
<td>Leaves.0010</td>
<td>45.00</td>
<td>36.88</td>
<td>63.12</td>
<td>61.88</td>
<td>61.25</td>
<td>70.00</td>
</tr>
<tr>
<td>Leaves.0011</td>
<td>45.31</td>
<td>64.69</td>
<td>76.88</td>
<td>64.06</td>
<td>91.56</td>
<td>95.94</td>
</tr>
<tr>
<td>Leaves.0012</td>
<td>94.06</td>
<td>80.94</td>
<td>75.62</td>
<td>88.44</td>
<td>86.88</td>
<td>91.87</td>
</tr>
<tr>
<td>Leaves.0016</td>
<td>50.31</td>
<td>60.31</td>
<td>69.69</td>
<td>57.50</td>
<td>73.75</td>
<td>82.50</td>
</tr>
<tr>
<td>Metal.0000</td>
<td>97.19</td>
<td>96.25</td>
<td>100</td>
<td>99.38</td>
<td>96.25</td>
<td>98.44</td>
</tr>
<tr>
<td>Metal.0002</td>
<td>100</td>
<td>97.50</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Misc.0002</td>
<td>96.56</td>
<td>91.56</td>
<td>93.75</td>
<td>99.38</td>
<td>98.44</td>
<td>100</td>
</tr>
<tr>
<td>Sand.0000</td>
<td>91.87</td>
<td>100</td>
<td>99.38</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Stone.0001</td>
<td>70.94</td>
<td>68.44</td>
<td>84.38</td>
<td>75.94</td>
<td>90.63</td>
<td>94.06</td>
</tr>
<tr>
<td>Stone.0004</td>
<td>96.25</td>
<td>97.81</td>
<td>98.75</td>
<td>98.12</td>
<td>95.94</td>
<td>96.56</td>
</tr>
<tr>
<td>Terrain.0010</td>
<td>33.12</td>
<td>60.94</td>
<td>45.31</td>
<td>75.94</td>
<td>67.19</td>
<td>73.12</td>
</tr>
<tr>
<td>Tile.0001</td>
<td>54.06</td>
<td>92.50</td>
<td>73.12</td>
<td>87.19</td>
<td>92.50</td>
<td>93.75</td>
</tr>
<tr>
<td>Tile.0004</td>
<td>27.19</td>
<td>75.00</td>
<td>85.00</td>
<td>70.94</td>
<td>90.94</td>
<td>93.44</td>
</tr>
<tr>
<td>Tile.0007</td>
<td>95.94</td>
<td>96.25</td>
<td>85.94</td>
<td>94.06</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Water.0005</td>
<td>100</td>
<td>98.12</td>
<td>95.63</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Wood.0001</td>
<td>75.62</td>
<td>59.69</td>
<td>58.13</td>
<td>69.69</td>
<td>75.62</td>
<td>80.63</td>
</tr>
<tr>
<td>Wood.0002</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>99.38</td>
</tr>
<tr>
<td>Mean</td>
<td>76.77</td>
<td>82.77</td>
<td>83.50</td>
<td>86.23</td>
<td>90.35</td>
<td>93.50</td>
</tr>
</tbody>
</table>

was used in [Liu and Fieguth 2012]. This dataset consists of 61 texture classes, where each texture class is represented by only a single image. Each image is partitioned into nine $106 \times 106$ nonoverlapping subimages, with five samples for training and the
other four for testing. As reported in [Liu and Fieguth 2012], the best classification performances of RP and DLBP are 95.85% and 92.77% respectively. In contrast, the best classification accuracy of our H-HNMF for this data set is 97.54%.

In summary, our proposed H-HNMF is efficient for performing texture classification and outperforms seven representative texture classification methods. Furthermore, our proposed HNMF outperforms the LNMF in computing the low-dimensional basis of NMVs for texture classification.

6.3. Computational Cost

Table III reports the time for texture classification (TTC) running each of the BP-MD, LEH-NN, PMC-BC, CSC-NN, H-LNMF and H-HNMF approaches on the datasets Set-2. From Table III, it is observed that BP-MD is the most efficient, the reason is that the dimension of BP features is low and there is no learning process in estimating the parameters. In contrast, LEH-NN is the most time-consuming method among them. Our H-HNMF is over three times faster than LEH-NN on Set-2, which implies that, as a different modeling method in the wavelet domain, our proposed HIGH and subsequent HNMF learning are significantly more efficient than the LEH model without learning. In addition, our H-HNMF is also faster than the current PMC-BC, CSC-NN and H-LNMF. In PMC-BC approach, the most costly part is the parameter estimation of the Poisson mixtures by the BYY harmony learning. In the CSC-NN approaches, the dimension of feature used for classification is high. Due to that two large Laplacian matrix need be computed in LR-NMF, the computational cost of H-LNMF is slightly more than H-HNMF. Furthermore, if we compare them from the two perspectives of TCC and ACAR, our proposed is clearly superior to other five methods.
Fig. 9. The ACRs of H-LNMF and H-HNMF with different subspace dimensions when the wavelet decomposition scale varies from 1 to 4. (a) H-LNMF on Set-4. (a') H-HNMF on Set-4. (b) H-LNMF on Set-5. (b') H-HNMF on Set-5.

Table III. The TTC ($s \times 10^2$) and ACAR of the six methods on Set-2.

<table>
<thead>
<tr>
<th></th>
<th>BP-MD</th>
<th>LEH-NN</th>
<th>PMC-BC</th>
<th>CSC-NN</th>
<th>H-LNMF</th>
<th>H-HNMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTC</td>
<td>1.30</td>
<td>26.48</td>
<td>24.79</td>
<td>8.38</td>
<td>8.11</td>
<td>7.23</td>
</tr>
<tr>
<td>ACAR</td>
<td>76.77</td>
<td>82.77</td>
<td>83.50</td>
<td>86.23</td>
<td>90.35</td>
<td>93.50</td>
</tr>
</tbody>
</table>

7. CONCLUSIONS

In this work, we propose an efficient and effective texture classification approach. To this end, we propose a novel Heterogeneous and Incrementally Generated Histogram (HIGH) to model local features in each wavelet subband. By concatenating all the HIGHs in all wavelet subbands, a NMV can be constructed to represent a texture image. Due to NMVs high dimensionality as well as its nonnegativity, we propose a novel Hessian regularized discriminative nonnegative matrix factorization (HNMF) to compute the low-dimensional basis of linear subspace of NMVs of training textures, by which the low-dimensional and more discriminative representations of training and testing textures are obtained and used for texture classification. Furthermore, our proposed HNMF outperform the LNMF implemented by Laplacian regularization. Experimental results on several large texture image datasets show that our proposed classification method outperforms seven representative methods.
REFERENCES


