Available online at www.sciencedirect.com

ScienceDirect

Image and Vision Computing 26 (2008) 647–656

www.elsevier.com/locate/imavis

Unsupervised texture classification: Automatically discover and classify texture patterns ☆

Lei Qin [a,b,*], Qingfang Zheng [a], Shuqiang Jiang [a], Qingming Huang [a,b], Wen Gao [a,c]

[a] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*
[b] *Graduate School of the Chinese Academy of Sciences, Beijing, China*
[c] *School of Electronics Engineering and Computer Science, Peking University, Beijing, China*

Received 29 September 2005; received in revised form 1 July 2007; accepted 2 August 2007

Abstract

In this paper, we present a novel approach to classify texture collections. This approach does not require experts to provide annotated training set. Given the image collection, we extract a set of invariant descriptors from each image. The descriptors of all images are vector-quantized to form 'keypoints'. Then we represent the texture images by 'bag-of-keypoints' vectors. By analogy text classification, we use Probabilistic Latent Semantic Indexing (PLSI) and Non-negative Matrix Factorization (NMF) to perform unsupervised classification. The proposed approach is evaluated using the UIUC database which contains significant viewpoint and scale changes. We also report the results for simultaneously classifying 111 texture categories using the Brodatz database. The performances of classifying new images using the parameters learnt from the unannotated image collection are also presented. The experiment results clearly demonstrate that the approach is robust to scale and viewpoint changes, and achieves good classification accuracy even without annotated training set.

Keywords: Unsupervised texture classification; NMF; PLSI; Invariant descriptor

1. Introduction

Texture analysis is an essential problem in computer vision domain, and is extensively studied in the past two decades. Many texture classification methods [1–6] have been reported in literatures. Unfortunately, these methods require some form of supervision. This may range from using a registered stack of texture images [1], to provide labels ascertaining textures' classes in learning process [3,4]. However, for classifying a large image collection, any auxiliary processing is costly and labor-intensive. Thus this raises the question: Can we automatically discover the patterns in a large texture image collection and classify the images into groups without any supervision?

A number of recent studies have provided the possibility to unsupervised classification. First, by analogy with text classification, 'bag-of-keypoints' approach is proposed in [12]. It is an extension of 'bag-of-words' approach. The 'bag-of-keypoints' approach quantizes the descriptors of local invariant regions to form 'keypoints', and uses a histogram of the number of occurrences of 'keypoints' to represent an image. It is a oversimple approach because it discards all spatial relationships between features. However, it achieves remarkable success in visual categorization [10,11] and video retrieval [9]. This is partly because the local invariant regions are not only powerful to effectively to describe image contents [17,26], but also invariant under

☆ This work is supported by the National Hi-Tech Development Program of China under Grant No. 2006AA01Z117 and the Knowledge Innovation Program of the Chinese Academy of Sciences under Grant No. 20076032.

* Corresponding author. Address: Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

E-mail addresses: lqin@jdl.ac.cn (L. Qin), qfzheng@jdl.ac.cn (Q. Zheng), sqjiang@jdl.ac.cn (S. Jiang), qmhuang@jdl.ac.cn (Q. Huang), wgao@jdl.ac.cn (W. Gao).

0262-8856/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2007.08.003

viewpoints and illumination changes. The discrimination of local regions makes them play a similar role like ‘key words’ in text. The dependence on features co-occurrences makes ‘bag-of-keypoints’ approach well appropriate to texture analysis. A kind of texture is composed of some primitive patterns (keypoints), which are repeated throughout the texture. Different kinds of textures have different keypoints. This will make the keypoint distributions of different textures discriminable. Thus texture images can be classified based on their ‘bag-of-keypoints’ vectors. See Fig. 1 for example. Each row represents one category. In right panel, we show the average ‘bag-of-keypoints’ vectors. Clearly, the ‘bag-of-keypoints’ vectors of T15 and T20 are quite different.

Second, motivated by the success of unsupervised text classification approaches, we exploit two models in unsupervised text classification domain to perform unsupervised texture image classification in this paper. The two models are: the Probabilistic Latent Semantic Indexing (PLSI) [13,14], and Non-negative Matrix Factorization (NMF) [15,16]. Such models use latent space representation for unsupervised classification. The latent space representation can extract interpretable concepts within the co-occurrences matrix. The idea of applying text analysis methods to vision categorization is not novel (*e.g.,* [10]). Our contribution lies in rigorously demonstrating that probabilistic latent space model is well-suited to texture classification, and this unsupervised model achieves comparable classification performance with ‘state-of-art’ supervised approach.

Fig. 2 is a sketch of our unsupervised texture classification approach. In Section 2, we review the PLSI and NMF models. In Section 3, we describe the feature detection methods and ‘bag-of-keypoints’ model. In Section 4, we give the unsupervised classification results on UIUC Dataset and Brodatz Dataset. We also show the results of inferring unseen images. In Section 5, we summarize the paper and draw some conclusions.

1.1. Related texture classification works

Leung and Malik [1] are among the pioneers to recognize textures subjected to viewpoint and lighting changes. Their solution to the 3D structure of the surface is using a registered stack of images. Filter responses over the registered stack of images are clustered. The cluster centers represent the 3D textons. The histograms of textons are used to classification. However their approach requires specially registered texture image sets. Cula et al. [2] and Varma et al. [3] implement the 2D texton representation. The 2D textons can be generated from unregistered images instead of registered stack of images. Although their approach can recognize affine transformed textures, the representation of their texture is not invariant under affine changes. So Lazebnik et al. [4] propose to use local regions to represent the content of images. Each region is represented by two descriptors. Descriptors of each texture images are clustered to form its signature. They use EMD to measure the similarity of two signatures. This approach obtains high accuracy on the UIUC and Brodatz database. However their method also needs auxiliary data to specify the categories of images in the training set.

Table 1 gives the technique summary of our approach and the other 2D texton-based approaches.

2. The unsupervised classification models

We begin with some notations and definitions for the models in this section. Suppose we have n documents $D = \{d_1, \ldots, d_n\}$ comprising words from a vocabulary $W = \{w_1, \ldots, w_m\}$. By counting words occurred in docu-

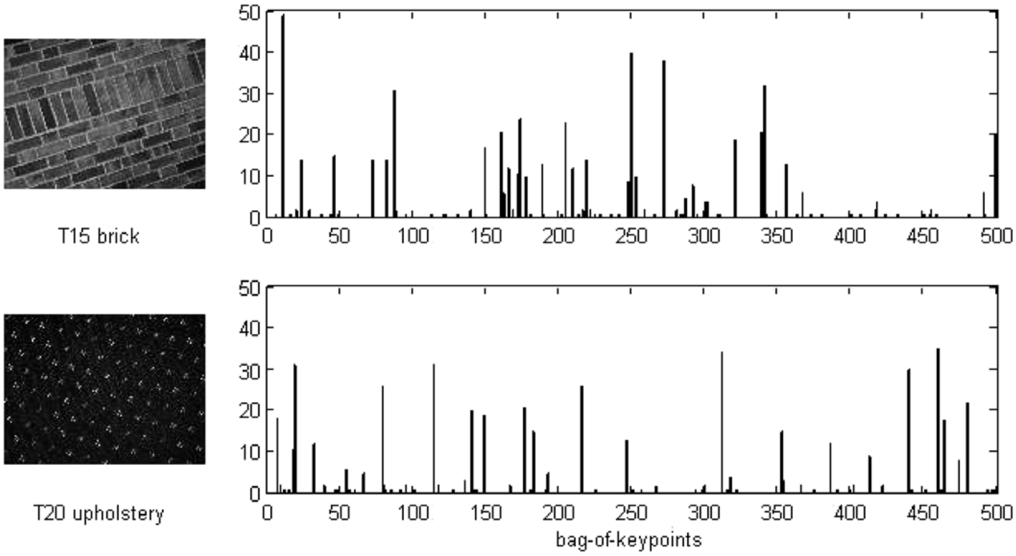


Fig. 1. Keypoints distribution of two categories. The left panel shows the categories. The right panel shows the distributions of keypoints for the categories.

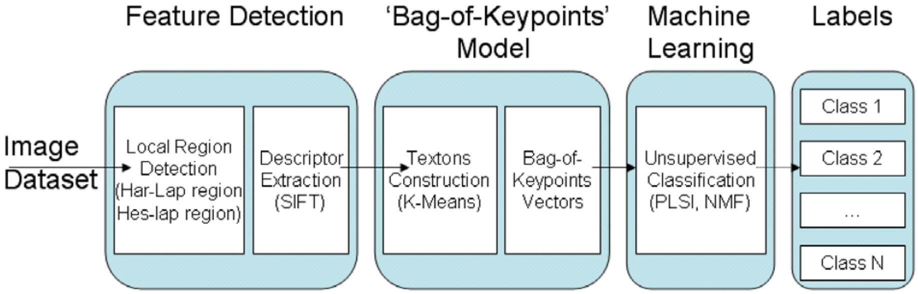

Feature Detection
'Bag-of-Keypoints' Model
Machine Learning
Labels
Image Dataset
Local Region Detection (Har-Lap region Hes-lap region)
Descriptor Extraction (SIFT)
Textons Construction (K-Means)
Bag-of-Keypoints Vectors
Unsupervised Classification (PLSI, NMF)
Class 1
Class 2
...
Class N

Fig. 2. The framework of our approach for unsupervised texture recognition.

Table 1
The main components of our approach and the other 2D texton-based approaches

Components	Leung et al. [1], Cula et al. [2], Varma et al. [3]	Lazebnik et al. [4]	Our approach
Local region	None: all pixels are used	Harris-affine regions & Laplace-affine regions	Harris-affine regions & Hessian-affine regions
Descriptor	Filter banks	Spin image, RIFT	SIFT
Textons construction	Universal clustering for all classes	Separate clustering for each images	Universal clustering for all classes
Representing	Histograms	Signatures	'bag-of-keypoints' vectors
Classification method	Nearest neighbor classification: supervised, two-class classification	Nearest neighbor classification: supervised, two-class classification	PLSI, NMF: unsupervised, multi-class classification

ments and discarding the sequential information of words, the collection of documents is capsulized in a $m \times n$ co-occurrence matrix N, where $N(w,d)$ represents the number of occurrences of a word w in document d. The simplified representation of documents is called 'bag-of-words' model.

2.1. The PLSI model

This model assumes a latent (hidden) class variable z_k associated with the occurrences of a word w_i in a particular document d_j. The graphical model of PLSI is shown in Fig. 3(a). The joint probability $P(w_i,d_j)$ is defined by the following: $P(w_i,d_j) = P(d_j)P(w_i|d_j)$. The conditional probability of the observed variables $P(w_i|d_j)$ is obtained by marginalization over the latent variable z_k.

$$P(w_i|d_j) = \sum_k P(w_i|z_k)P(z_k|d_j) \tag{1}$$

where $P(z_k|d_j)$ is the probability of latent variable z_k occurring in document d_j, where $\sum_k P(z_k|d_j) = 1$; and $P(w_i|z_k)$ is the probability of word w_i occurring in a particular latent variable z_k, where $\sum_i P(w_i|z_k) = 1$. Eq. (1) equals to a matrix decomposition as shown in Fig. 3(b).

The Expectation Maximization approach is used to fit the model by maximizing the data loglikelihood.

$$\begin{aligned} J_{\mathrm{PLSI}} &= \sum_{ij}(N(w_i,d_j)\log P(w_i,d_j)) \\ &= \sum_j N(d_j)[\log P(d_j) \\ &\quad + \sum_i \frac{N(w_i,d_j)}{N(d_j)} \log \sum_k P(w_i|z_k)P(z_k|d_i)] \end{aligned} \tag{2}$$

where $N(w_i,d_j)$ is the number of word w_i in document d_j, and $N(d_j) = \sum_i N(w_i,d_j)$ is the document length.

Expectation-step: Posterior probabilities of the latent variables $P(z_k|w_i,d_j)$ are computed.

$$P(z_k|w_i,d_j) = \frac{P(w_i|z_k)P(z_k|d_j)}{\sum_{k'} P(w_i|z'_k)P(z'_k|d_j)} \tag{3}$$

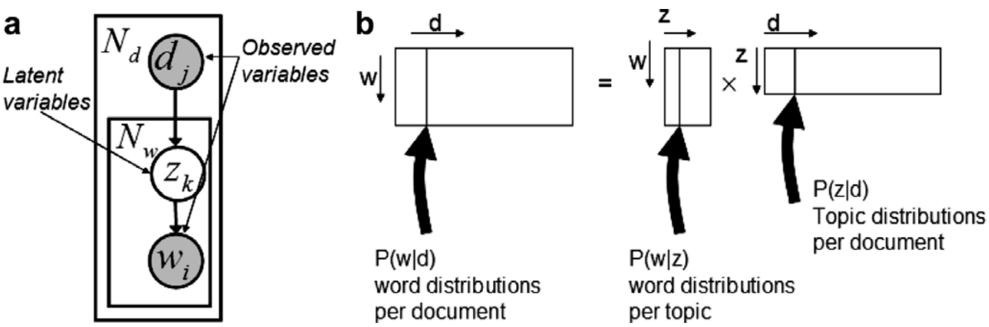

a
b
Latent variables
Observed variables
N_d
d_j
N_w
z_k
w_i
d
w
=
z
w
×
z
d
P(w|d) word distributions per document
P(w|z) word distributions per topic
P(z|d) Topic distributions per document

Fig. 3. (a) The graphical illustration of PLSI model. (b) The factorization of $P(w|d)$.

Maximization-step: The parameters $P(d_j|z_k)$ and $P(w_i|z_k)$ are updated.

$$P(w_i|z_k)=\frac{\sum_j N(w_i,d_j)P(z_k|w_i,d_j)}{\sum_{i'j}N(w_{i'},d_j)P(z_k|w_{i'},d_j)} \tag{4}$$

$$P(z_k|d_j)=\frac{\sum_i N(w_i,d_j)P(z_k|w_i,d_j)}{N(d_j)} \tag{5}$$

Alternating the E and M step approaches a local maximum of the loglikelihood [14].

2.2. The NMF model

Given a positive $m \times n$ matrix N, NMF model finds a low rank approximation of N by factoring N into a non-negative $m \times k$ matrix U and a non-negative $k \times n$ matrix V^T, such that $N \approx UV^T$. k is the reduced rank. Fig. 4 is the graphical illustration of this factorization. Usually the choice of k is application and data dependent, and much smaller than n. Each element u_{ij} of matrix U is the probability of word w_i associated with a particular latent variable z_j and each element v_{ij} of matrix V indicates the probability of document d_i associated with latent variable z_j.

The optimal choice of matrices U and V are those non-negative matrices that minimize the following objective function:

$$J_{\mathrm{NMF}}=\sum_{ij}N_{ij}\log\frac{N_{ij}}{(UV^T)_{ij}}-N_{ij}+(UV^T)_{ij} \tag{6}$$

This is a typical constraint optimization problem. This optimization problem is solved using the multiplicative update rules:

$$\begin{aligned} V_{jk} &\leftarrow \frac{V_{jk}}{\sum_i U_{ik}}\sum\nolimits_i \frac{N_{ij}}{(UV^T)_{ij}}U_{ik}, \\ U_{ik} &\leftarrow \frac{U_{ik}}{\sum_j V_{jk}}\sum_j \frac{N_{ij}}{(UV^T)_{ij}}V_{jk} \end{aligned} \tag{7}$$

Under the above updating rules, U and V remain nonnegative and the object function J_{NMF} is non-increasing [16].

2.3. The baseline model

We implement a clustering algorithm as our baseline method. The algorithm uses the k-means to cluster 'bag-of-keypoint' vectors. The images whose 'bag-of-keypoint' vectors have the same cluster label are classified into a category.

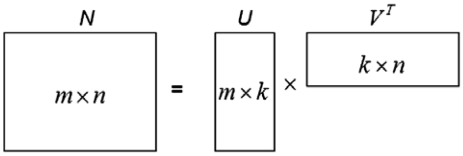


Fig. 4. The graphical illustration of NMF model.

3. Feature detection and textons construction

Recent researches show that local invariant region detectors get notable success in many computer vision domains, such as image matching [18,21,27], image retrieval [19,26], object recognition [17], and video retrieval [22]. Compared with global features, local features are robust to clutter, occlusions and spatial variations. We use two types of local regions in our implementation. The first is based on the Harris-Affine detector that is described in [19]. The second is constructed using the Hessian-Affine detector [20]. The regions detected by Hessian-Affine detector are similar to those detected by a Laplacian-Affine detector of Lindeberg and Garding [23]. Both detectors have the following three steps:

(1) Spatial coordinate localization: [19] relies on a multi-scale Harris corner detector to localize position of local regions in spatial, while [20] relies on multi-scale Hessian blob detector.
(2) Automatic scale selection: both detectors select the characteristic scale at which a normalized Laplacian function attains a local maximum.
(3) Affine adaption process: This process is based on isotropy of the second moment matrix of the local region. This step makes the regions invariant under the affine transformations. More details about the affine adaption can be found in [23].

We use the binaries available at http://www.robots.ox.ac.uk/~vgg/research/affine/. Both types of regions are represented by ellipses. The regions extracted by the two detectors are totally different. The Harris detector extracts regions which have significant intensity changes, while the Hessian detector finds blobs of uniform intensity. So they are complementary detectors. This phenomenon is also reported in [4].

Each region is represented by a 128-dimension SIFT descriptor [17]. The SIFT descriptor is scale and affine invariant by incorporating the previous estimated scale and affine parameters of the local region, and is rotation invariant by computing relative to the dominant orientation of the region.

Given the collection of SIFT descriptors from each image of all categories, we quantize them into 'keypoints' by K-means clustering algorithm. We provide two examples of 'keypoints' in Fig. 5. Both 'keypoints' are extracted from UIUC database. In each case, we show 9 instances of this 'keypoint'. The instances are the elliptical regions in the center of each image. Note in each case the contents of elliptical regions are very similar, while the shapes of elliptical regions exhibit rotation and scale variation.

The number of cluster K is an important parameter. We test the performance of classifying UIUC database with different number of K. The results are shown in Fig. 6. The mean classification rate at $K = 1000$ only increases slightly against the mean classification rate at $K = 500$.

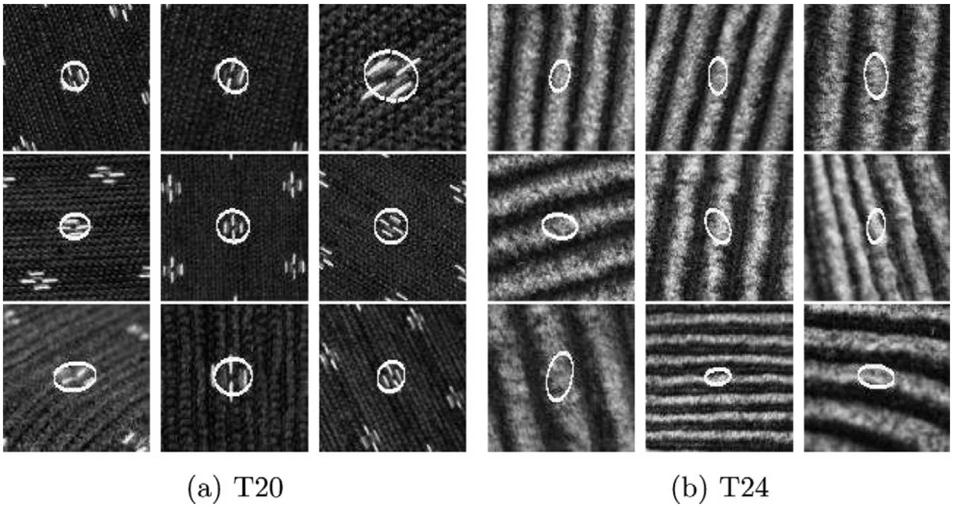


Fig. 5. Examples of two 'keypoints' extracted from UIUC database. (a) A 'keypoint' of fabric (T20). (b) A 'keypoint' of corduroy (T24).

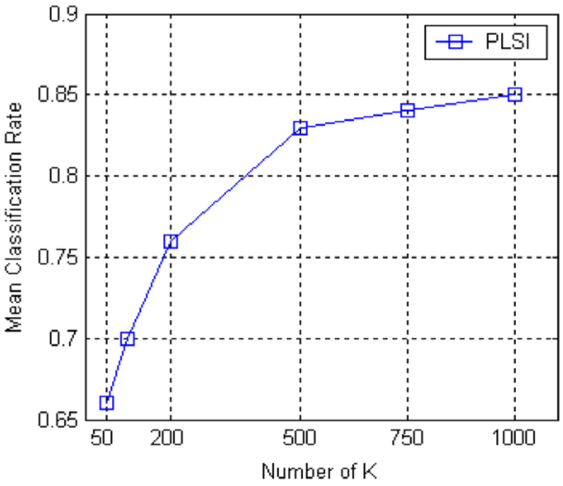


Fig. 6. Number of textons vs. performance.

Thus, we choose $K = 500$ in the following K-means algorithm.

4. Experiment results

We present results from two experiments. In the first, we explore the ability of our approach to automatically discover visual categories presented in a collection of totally unlabeled images. We carry out this experiment on UIUC database (25 classes, 40 images per class) and Brodatz database (111 classes, 9 images per class). Then we explore the performance of classifying unseen images with learned visual categories. Three performance measures are used to evaluate our approach.

The confusion matrix:

$$C_{ij} = \frac{|\{I_k \in L_j : f(I_k) = i\}|}{|L_j|} \tag{8}$$

L_j is the set of images which belong to category j, and $f(I_k)$ is the class label which obtains the highest classifier score by the multi-class classifier for image I_k.

The mean classification rate:

$$O = \frac{\sum_{j=1}^{N}|L_j|C_{jj}}{\sum_{j=1}^{N}|L_j|} \tag{9}$$

The rank statistics:

$$R_i = \frac{\sum_{j=1}^{N}\frac{|I_k \in L_j : f(I_k) \in S_{ij}|}{|L_j|}}{N} \tag{10}$$

These are the percentages that the correct label is in the first i class labels. N is the number of categories, S_{ij} is the set of labels which get i maximum classifier scores for category j.

4.1. Visual category discovery

In the following, we carry out experiments on two databases. In each case, the images of K categories are mixed together forming a test image set. The methods described in Section 2 are fitted to the test image set for the number of visual patterns, K. We can infer the number of visual patterns using the nonparametric Bayesian method [24]. In the case of PLSI, model computes the probability coefficients $P(z_k|d_j)$ for each image d_j. The decision of an image d_j is made to the category label k that obtains the maximum $P(z_k|d_j)$. While, in the case of NMF, matrix V is used to determine the label of each image. For an image d_j, assign it to category k, if $k = \arg\max_i V_{ji}$.

4.1.1. Dataset 1: UIUC database

The UIUC database is a texture database containing 1000 images in 25 classes. Each class has 40 images. The whole database is publicly available at http://www-cvr.ai.uiuc.edu/ponce_grp/. It is a challenging database, not only because of significant viewpoint changes and scale variations in each class, but also because it contains images with nonplanar surface, significant nonrigid deformations, and inhomogeneous texture patterns. Fig. 7 shows examples of four categories T01, T10, T20 and T23 from the UIUC database.

We extract about 1.2 M regions from the UIUC database. The median number of regions extracted per image is 1205 (545 for Harris Affine region, 660 for Hessian Affine region). We carry out a set of experiments with increasing number of classes. The numbers of classes used in these experiments are 3, 8 and 25. We summarize the results in Table 2.

(1) Three texture classes (T23–T25). Although the three classes all are fabric textures, this is a relatively easy experiment with only 3 classes. Both PLSI and NMF models perform perfectly well with 100% correct recognition rate. The baseline model only obtains 54.1% correct recognition rate. This low rate reveals that Euclidean distance may not be a good similarity measure in the ‘bag-of-keypoints’ model.

(2) Eight texture classes (T18–T25). Here we add five classes (fabric, wall paper, fur and two carpets). As the number of classes becomes bigger, the classification experiment becomes more challenging. However, both PLSI and NMF model perform very well with 5 misclassified images and 2 misclassified images, respectively. Table 3 shows the performance of NMF in confusion matrix formation.

(3) Twenty five texture classes (T1–T25). In this experiment, we use all the 25 classes in the UIUC database. The whole UIUC database are combined together. It is a really challenging experiment. Again PLSI and

Table 2
The overall classification results of UIUC database

Categories	Lazebnik [4]	PLSI	NMF	Baseline method
T23–T25	0.9589	1.00	1.00	0.541
T18–T25	0.9370	0.984	0.994	0.618
T1–T25	0.9261	0.830	0.804	0.531

Table 3
Confusion matrix of NMF model for 8 classes, $K = 500$

True classes →	T18	T19	T20	T21	T22	T23	T24	T25
T18	40	0	0	0	0	0	0	0
T19	0	40	0	0	0	0	0	0
T20	0	0	40	0	0	0	0	0
T21	0	0	0	39	0	0	0	0
T22	0	0	0	0	40	1	0	0
T23	0	0	0	0	0	39	0	0
T24	0	0	0	1	0	0	40	0
T25	0	0	0	0	0	0	0	40

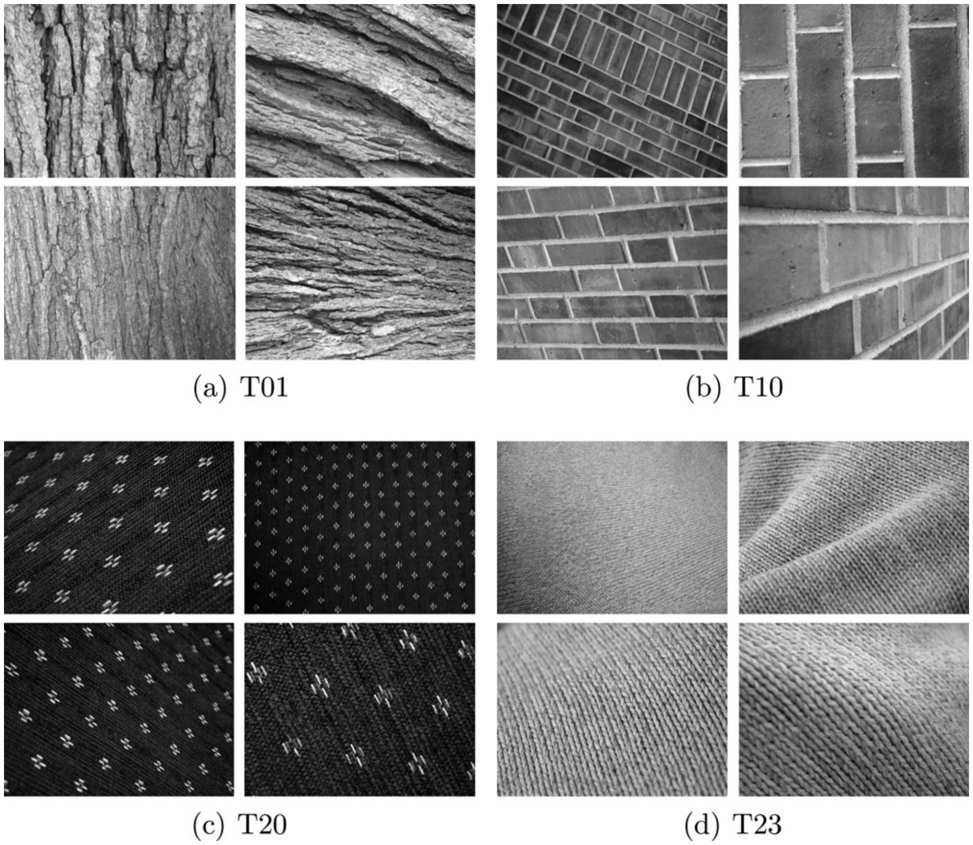
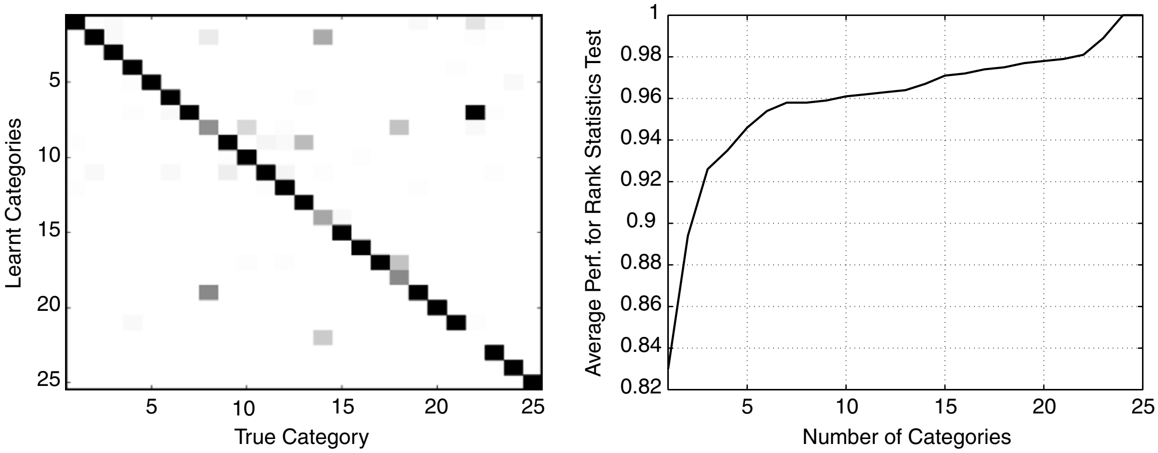

(a) T01
(b) T10
(c) T20
(d) T23

Fig. 7. Textures from the UIUC database.

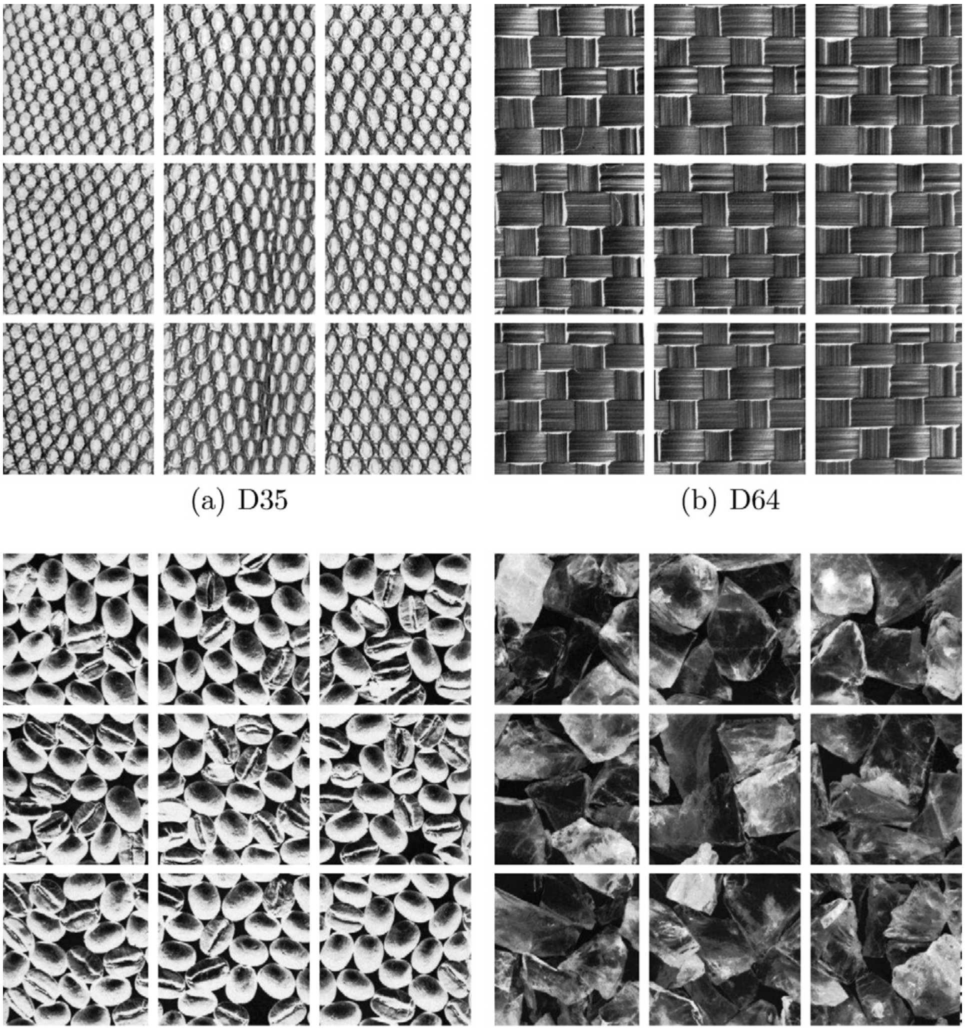
NMF model exhibit a very similar performance. Fig. 8(a) is an overview of the performance of the PLSI model. The mean classification rate is 83.0%. Fig. 8(b) is the rank statistics of the classification



(a) Confusion matrix

(b) Rank statistics

Fig. 8. (a) Confusion matrix of PLSI model on UIUC database, black = 1 and white = 0. (b) Rank statistics of the confusion matrix.



(a) D35

(b) D64

(c) D74

(d) D99

Fig. 9. Textures from the Brodatz database.

results. Using the first three best choices, the mean classification result increases to 92.6%.

Observing Fig. 8(a) carefully, we find that most of classes are classified with high accuracy. However the class T22 is totally misclassified. Most of the T22 are classified as T07. This is partly because the extracted features of T07 and T22 are quite similar.

4.1.2. Dataset 2: Brodatz database

The Brodatz database is a well known texture database. It is derived from the Brodatz Album [7] which contains 111 images. It is formed by dividing each image of Brodatz Album into nine nonoverlapping 215 × 215 images [4,8,25]. Thus the Brodatz database consists of 999 images. The number of categories in the Brodatz database is quite larger than that in the previous section, while each category has a relatively small number of instances. This makes it more challenging. Fig. 9 shows examples of four categories D35, D64, D74 and D99 from the Brodatz database.

We extract about 770 K regions from the Brodatz database. The median number of regions extracted per image is 771 (345 for Harris Affine region, 426 for Hessian Affine region). In this experiment, we use all the 111 classes in the Brodatz database. The results are summarized in Table 4. As the previous section, the PLSI and NMF model achieves similar classification rate.

Fig. 10(a) is an overview of the performance of the PLSI model. The mean classification rate is 64.46%. Fig. 10 (b) is the rank statistics of the classification results. Using the first three best choices, the mean classification result increases to 72.97%.

4.1.3. Experiment results analysis

PLSI and NMF seem to be quite different clustering techniques. NMF decomposes a matrix N into a product of non-negative matrix UV, and uses a multiplicative update rule to minimize the KL divergence $J_{\mathrm{NMF}} = KL(N\|UV)$. PLSI is a model base clustering technique. PLSI models the joint probability matrix as arising from a mixture model with K latent classes, and uses EM algorithm to maximize loglikelihood J_{PLSI}. However, there are some fundamental relationship between NMF and PLSI. Firstly, compare Fig. 3 (b) with Fig. 4, we can see that they are quite similar. This similarity might not seem surprising. PLSI factorizes the joint probability matrix $[P(w_i,d_j)]_{I\times J} = [P(w_i|z_k)]_{I\times K}[P(z_k|d_j)P(d_j)]_{K\times J} = U'_{I\times K}V'_{K\times J}$. Probability matrices U' and V' are obviously non-negative. So PLSI corresponds to a nonnegative matrix factorization. Secondly, Ding et al. in [28] propose that the objective function of PLSI is identical to the objective function of NMF, $\max J_{\mathrm{PLSI}} \Longleftrightarrow \min J_{\mathrm{NMF}}$. And they conclude that NMF and PLSI are equivalent in this sense.

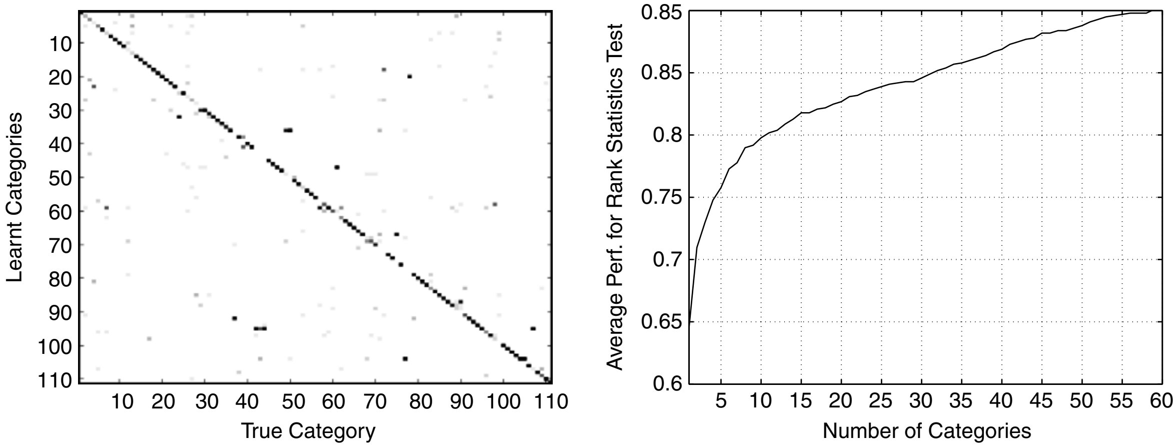
Our experiment results validate the conclusion of [28] that NMF and PLSI are equivalent in theory. From those results in the Tables 2 and 4, we can see that across all the datasets we used, the classification performance of PLSI and NMF are consistently indistinguishable in the case of multi-classes unsupervised texture classification. As there is no benefit in choosing NMF model over PLSI model, we now select PLSI model for following experiments.

4.2. Inference of new images

For classifying an unseen image d_{test}, the conditional distribution over learned topics has to be computed. In the case of PLSI, the 'folding-in' query method proposed

Table 4
The overall classification results of Brodatz database

Categories	PLSI	NMF	Baseline method
D1–D111	0.6446	0.6137	0.4965



(a) Confusion matrix (b) Rank statistics

Fig. 10. (a) Confusion matrix of PLSI model on Brodatz database, black = 1 and white = 0. (b) Rank statistics of the confusion matrix.

Table 5
Detail classification results of UIUC database

Class	PLSI-1	PLSI-3
T01	0.850	0.950
T02	0.450	1.000
T03	0.750	1.000
T04	0.850	1.000
T05	1.000	1.000
T06	0.950	1.000
T07	1.000	1.000
T08	0.850	1.000
T09	0.650	1.000
T10	0.900	1.000
T11	0.900	0.950
T12	0.950	1.000
T13	0.700	0.850
T14	0.450	0.800
T15	0.950	1.000
T16	1.000	1.000
T17	0.600	0.950
T18	0.100	0.850
T19	0.600	1.000
T20	0.950	1.000
T21	1.000	1.000
T22	0.000	0.000
T23	0.950	1.000
T24	0.900	1.000
T25	1.000	1.000
Mean	0.772	0.934

Column 1: class labels of UIUC database. Column 2: classification accuracy of PLSI-1, which using the label which obtains highest $P(z_k|d_{\text{test}})$. Column 3: classification accuracy of PLSI-3, which using class labels which obtain top three $P(z_k|d_{\text{test}})$.

in [13] is used to compute the topic mixing coefficients $P(z|d_{\text{test}})$. The method maximizes the likelihood of image d_{test} with respect to learned $P(w|z)$. This is obtained by a similar version of the EM algorithm used in learning, where only $P(z_k|d_{\text{test}})$ are adapted in each M-step. The factors $P(w|z)$ are kept fixed.

We use the UIUC database to perform this test. Each category of UIUC database is randomly split into two separate sets of images, one for learning and the other for testing. Each set has 20 images. The learning sets of all categories are mixed together. The PLSI model fits the mixed learning sets with 25 categories. When asked to classify one test image d_{test}, the method described above is used to 'folding-in'. We use two models to measure the classification performance. The first is PLSI-1 model, which returns the class with highest classification score $P(z_k|d_{\text{test}})$. The second is PLSI-3 model, which returns three classes which get top three $P(z_k|d_{\text{test}})$. The details of classification performance is presented in Table 5. The classification rate of PLSI-1 model is 77.2%. The classification rate of PLSI-3 increases to 93.4%. The most significant changes are the class T18(from 0.1 to 0.85), the class T02(from 0.45 to 1.0) and class T09(from 0.65 to 1.0). This reveals that most of the time, the correct class label is the first class label or in the top three possible class labels that PLSI model returns. The results show our method can successfully infer unseen images using learned topics.

Fig. 11 shows the histograms of classification rates for all 25 classes. Fig. 11(a) is the results using PLSI-1 model, and Fig. 11(b) is the results using PLSI-3 model. The histograms reveal most of textures are classified very well. In PLSI-1 model, 5 textures have 100% classification rate, 13 textures have classification rate at least 90%. While for PLSI-3 model, the results are improved significantly. 18 textures have 100% classification rate, 21 textures (more than eighty percent of all textures) have classification rate at least 95%.

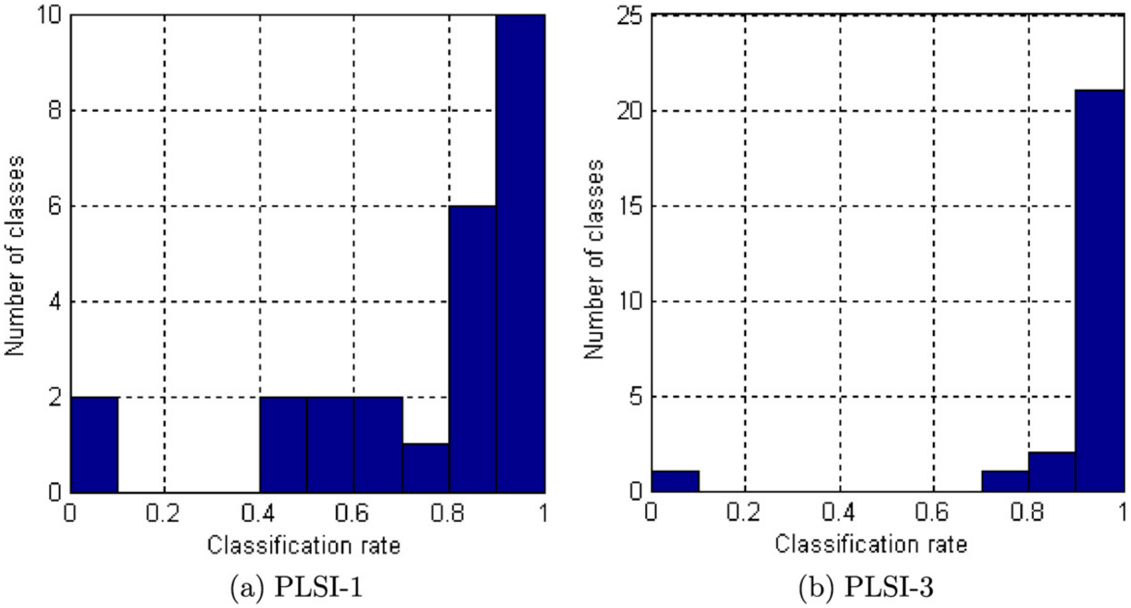


Fig. 11. Histogram of classification rates for PLSI-1 model and PLSI-3 model.

5. Conclusion

In this paper, we have demonstrated that it is possible to discover texture categories from a set of unlabeled images in an unsupervised manner. Furthermore, we successfully infer unseen images using discovered categories. Our approach has been evaluated on a 25 categories database. It is well demonstrated our approach is robust to significant scale and viewpoint changes, and it achieves good classification accuracy in the same time. We also evaluate our approach on the Brodatz database, which has 111 texture classes. To our knowledge this is the largest number of texture categories that have ever been subjected to unsupervised experiments. Our approach obtains accepted classification results.

References

[1] T. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, IJCV 43 (1) (2001) 29–44.

[2] O.G. Cula, K.J. Dana, Compact representation of bidirectional texture functions, Proc. Comput. Vis. Pattern Recognit. 1 (2001) 1041–1047.

[3] M. Varma, A. Zisserman, Classifying images of materials: achieving viewpoint and illumination independence, Proc. Eur. Conf. Comput. Vis. 3 (2002) 255–271.

[4] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005).

[5] S. Lazebnik, C. Schmid, J. Ponce, A maximum entropy framework for part-based texture and object recognition, in: ICCV 2005.

[6] B. Georgescu, I. Shimshoni, P. Meer, Mean shift based clustering in high dimensions: a texture classification example, Proc. Int. Conf. Comput. Vis. (2003) 456–463.

[7] P. Brodatz, Textures: A Photographic Album for Artists and Designers, Dover, New York, 1966.

[8] F. Liu, R.W. Picard, Periodicity, directionality, and randomness: word features for image modeling and retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 18 (7) (1996) 722–733.

[9] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: Proc. ICCV, 2003.

[10] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, W.T. Freeman, Discovering objects and their localization in images, in: Proc. ICCV 2005, October 2005.

[11] A. Opelt, A. Fussenegger, P. Auer, Weak hypotheses and boosting for generic object detection and recognition, in: Proc. ECCV, 2004.

[12] G. Csurka, C. Bray, C. Dance, L. Fan, Visual categorization with bags of keypoints, in: Workshop Stat. Learn. Comput. Vis., ECCV, pp. 1–22, 2004.

[13] T. Hofmann, Probabilistic latent semantic indexing, in: SIGIR, 1999.

[14] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Mach. Learn. 43 (2001) 177–196.

[15] D. Lee, H. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.

[16] D. Lee, H. Seung, Algorithms for non-negative matrix factorization, in: Advvances Neural Information Processing Systems, vol. 13, pp. 556–562, 2001.

[17] D. Lowe, Object recognition from local scale-invariant features, in: Proc. Int. Conf. Comput. Vis., pp. 1150–1157, 1999.

[18] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: Proc. BMVC., pp. 384–393, 2002.

[19] K. Mikolajczyk, C. Schmid, An affine invariant interest point detector, in: Proc. ECCV, Springer-Verlag, 2002.

[20] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, A Comparison of Affine Region Detectors, Int. J. Comput. Vis. (2005).

[21] F. Schaffalitzky, A. Zisserman, Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?"Proc. ECCV, vol. 1, Springer-Verlag, 2002, pp. 414–431.

[22] J. Sivic, F. Schaffalitzky, A. Zisserman, Object level grouping for video shots, in: Proc. ECCV, 2004.

[23] T. Lindeberg, Feature detection with automatic scale selection, IJCV 30 (2) (1998) 77–116.

[24] Y.W. Teh, M.I. Jordan, M.J. Beal, D.M. Blei, Hierarchical dirichlet processes, in: Proc. NIPS, 2004.

[25] K. Xu, B. Georgescu, D. Comaniciu, P. Meer, Performance analysis in content-based retrieval with textures, in: Proceedings of the International Conference Pattern Recognition, vol. 4, pp. 275–278, 2000.

[26] C. Schmid, R. Mohr, Local Gray-Value Invariants for Image Retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 19 (5) (1997) 530–535.

[27] T. Tuytelaars, L. Van Gool, Matching Widely Separated Views Based on Affinely Invariant Neighbourhoods, Int. J. Comput. Vis. 59 (1) (2004) 61–85.

[28] Chris Ding, Tao Li, Wei Peng: Nonnegative matrix factorization and probabilistic latent semantic indexing: equivalence, chi-square statistic, and a hybrid method, in: Proc. AAAI Natl. Conf. Artif. Intell. (AAAI-06), July 2006.