Fusing Subcategory Probabilities for Texture Classification

Yang Song[1], Weidong Cai[1], Qing Li[1], Fan Zhang[1], David Dagan Feng[1], Heng Huang[2]
[1]BMIT Research Group, School of IT, University of Sydney, Australia
[2]Department of Computer Science and Engineering, University of Texas, Arlington, USA

Abstract

Texture, as a fundamental characteristic of objects, has attracted much attention in computer vision research. Performance of texture classification is however still lacking for some challenging cases, largely due to the high intra-class variation and low inter-class distinction. To tackle these issues, in this paper, we propose a sub-categorization model for texture classification. By clustering each class into subcategories, classification probabilities at the subcategory-level are computed based on between-subcategory distinctiveness and within-subcategory representativeness. These subcategory probabilities are then fused based on their contribution levels and cluster qualities. This fused probability is added to the multiclass classification probability to obtain the final class label. Our method was applied to texture classification on three challenging datasets – KTH-TIPS2, FMD and DTD, and has shown excellent performance in comparison with the state-of-the-art approaches.

1. Introduction

Texture provides important information for many computer vision applications, such as material classification and scene and object recognition. Accurate classification of texture images is however quite challenging. Some of the main challenges include the wide variety of natural texture patterns and large intra-class variation caused by illumination and geometric changes, and relatively low inter-class distinction [26, 41].

To tackle these challenges, extensive research has been conducted to design highly discriminative and descriptive texture features. Local texture descriptors have been the predominant approaches with different ways of keypoint selection, local descriptor design and histogram encoding [31, 25, 45, 29, 40, 37, 8]. Another research focus is to incorporate feature invariance to key transformations to accommodate the intra-class variations and enhance the discriminative capability of texture descriptors [27, 41, 44, 9, 38, 35, 21, 34]. Classification at the finer subcategory-level has also been explored [3, 8] by discovering subtypes in a texture category based on the concept of visual attributes [16, 2, 4, 24, 32].

On the other hand, the classification models used in texture classification are usually quite standard. The most often used classifiers include the nearest neighbor (NN) approach [41, 44, 9, 40, 38], and support vector machine (SVM) with various kernel types [45, 29, 37, 35, 34, 8]. SVM is widely recognized as highly effective and usually boosts the discriminative power of feature descriptors compared to NN. However, it is a monolithic model and its performance can be affected with large intra-class variations especially when there is considerable inter-class overlap in the feature space.

The sub-categorization method has recently been proposed to alleviate the problem of intra-class variations and inter-class ambiguity. It works by identifying the visual subcategory structures of individual classes and modeling the different subcategories independently. This could facilitate better separation between different classes compared to generating a monolithic model for the entire feature space. The sub-categorization method generally contains three main components: generation of subcategories, classifier learning for individual subcategories, and fusion of subcategory results. Unsupervised clustering is usually performed for subcategory generation [46, 15, 1, 47]. Samples from different classes are incorporated as additional constraints to improve the clustering performance [12]. Fusion of subcategory results is typically performed by max or mean pooling [46, 15, 1, 47], and second layer of classifier learning [19, 12]. The clustering step could also be discriminative and integrated with the classification objective [20]. Such methods have been shown to improve the classification performance in various applications, including face recognition [46], traffic sign categorization [15], object detection and recognition [19, 1, 47, 12], and discovering head orientations [20].

Different from the current studies in texture classification, which mostly focus on designing new texture feature descriptors, our aim is to improve the classification accuracy with a new classification model using existing features.

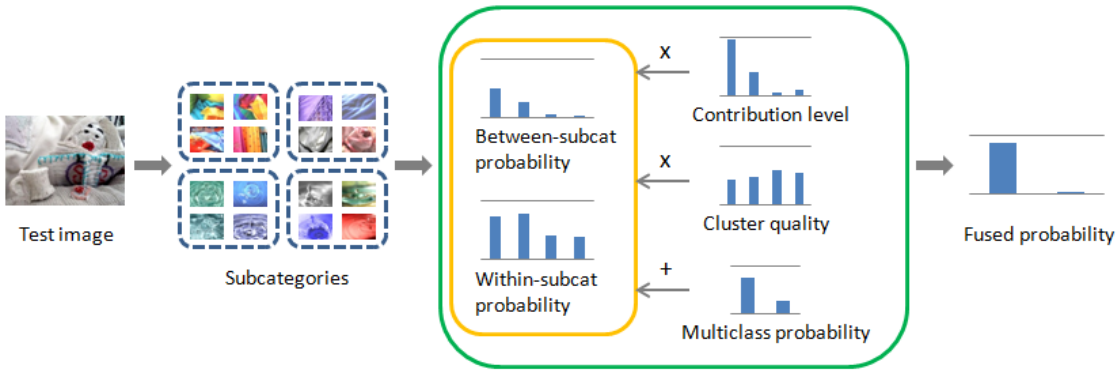

Figure 1. Overview of our proposed method. During testing, the between- and within-subcategory probabilities are computed, then fused based on the contribution levels, cluster qualities and multiclass probability to classify the test image. During training, the training images of each class are sub-categorized, subcategory-level models exploring between-subcategory distinctiveness and within-subcategory representativeness are built, the cluster qualities are computed, and multiclass SVM is trained.

In this paper, we propose a sub-categorization model for texture classification. We first design a locality-constrained subspace clustering method to efficiently generate subcategories of individual classes. At the subcategory-level, two probability measures are computed based on between-subcategory distinctiveness and within-subcategory representativeness, to quantify the probability of a test data belonging to each subcategory. The subcategory probabilities are then fused weighted by contribution level and cluster quality together with class-level probabilities to classify the test data. An overview of our method flow is shown in Figure 1. For texture descriptors, we use the improved feature vector (IFV) [33] and convolutional architecture for fast feature embedding (Caffe) [22]. Experiments are conducted on three challenging datasets: the KTH-TIPS2 database [5], Flickr Material Database (FMD) [36], and the Describable Textures Dataset (DTD) [8].

Our technical contributions can be summarized as follows: (i) We propose a sub-categorization model for texture classification. (ii) We designed a locality-constrained subspace clustering method for subcategory generation, and between- and within-subcategory probability computation and weighted fusion for classification. (iii) We obtained better classification performance than the state-of-the-art on three challenging datasets.

2. Subcategory Generation

2.1. Sparse Subspace Clustering

Suppose a dataset $X = \{x_i : i = 1, ..., N\} \in \mathbb{R}^{H \times N}$ comprises N vectors of H dimensions. The sparse subspace clustering (SSC) [14] algorithm segments the dataset into multiple clusters based on the underlying feature subspaces. A sparse representation coefficient matrix $Z \in \mathbb{R}^{N \times N}$ is obtained to represent the similarities between data samples by solving the following function:

$$\min_{Z} \|Z\|_1 \;\; s.t. \; X = XZ, diag(Z) = 0 \tag{1}$$

Each data sample x_i is thus expressed as a linear combination of the other data $x_i = \sum_{j \neq i} z_{ij} x_j$, and each entry $z_{ij} \in Z$ represents the similarity between data samples x_i and x_j. An affinity matrix $A = (|Z| + |Z|^T)/2$ is finally computed and used to segment the dataset with spectral clustering.

SSC has been applied successfully for motion segmentation. The related low rank representation (LRR) [28] algorithm enforces the affinity matrix to be low rank. The least square regression (LSR) [30] technique explores the grouping effect for improving the segmentation performance. Block-diagonal priors [43, 18] have also been proposed to encourage clean separation in the affinity matrix. In our method, the subspace clustering is performed to generate subcategories within each class and the outputs do not correspond to the final classification results. Therefore, we do not expect complete separation between the subcategories and our design emphasis is not to further improve the clustering accuracy. SSC provides satisfactory performance for our purpose but is generally slow due to the sparse approximation process for deriving the coefficient matrix. We thus modify the SSC algorithm with locality constraints to enhance the efficiency.

2.2. Locality Constrained Subspace Clustering

We formulate the following objective function to obtain the sparse representation coefficient matrix Z:

$$\begin{aligned} &\min_{\{z_i\}} \sum_{i=1}^{N} \|x_i - X z_i\|^2 + \lambda \|d_i \odot z_i\|^2 \\ &s.t. \;\; \mathbf{1}^T z_i = 1, \;\; \|z_i\|_0 \leq P, \;\; z_{ii} = 0, \;\; \forall i \end{aligned} \tag{2}$$

where $z_i \in \mathbb{R}^N$ is the ith column in Z indicating the similarity between x_i and each sample in the dataset X. z_i is P-sparse and Xz_i is supposed to well approximate x_i. The second term, which is adopted from the locality-constrained linear coding (LLC) [42], encourages smaller coefficients to be assigned to samples that are more different from x_i with $d_i \in \mathbb{R}^N$ containing the pairwise Euclidean distances. The constant λ controls the balance between the two terms.

The coefficient vector z_i can be efficiently obtained by first constructing a local codebook $\tilde{X}_i \in \mathbb{R}^{H \times P}$ and distance vector $\tilde{d}_i \in \mathbb{R}^P$ from P data samples in X that are the most similar to x_i (excluding x_i). Then a P-dimensional coefficient vector $\tilde{z}_i$ is computed analytically via:

$$\begin{aligned} \tilde{z}_i^* &= (V_i + \lambda diag(\tilde{d}_i)) \setminus \mathbf{1} \\ \tilde{z}_i &= \tilde{z}_i^* / (\mathbf{1}^T \tilde{z}_i^*) \end{aligned} \tag{3}$$

where $V_i = (\tilde{X}_i - x_i \mathbf{1}^T)^T (\tilde{X}_i - x_i \mathbf{1}^T)$. The coefficient vector z_i is thus derived by mapping $\tilde{z}_i$ back to the N-dimensional space. Subsequently, similar to SSC, an affinity matrix is computed as $A = (|Z| + |Z|^T)/2$ and spectral clustering is performed to generate the clusters.

In our formulation, the dataset X contains the training data of one class. The clustering outputs of X then correspond to the subcategories of that class. Each subcategory would exhibit lower intra-class feature variation compared to the entire class. Formally, we denote the dataset of one class c as X_c. Assume that K_c subcategories are generated for X_c, which are denoted as $\{S_{ck} : k = 1, ..., K_c\}$. These subcategories from all classes are then the bases for our probability estimation in the subsequent steps.

3. Subcategory Probabilities

We design two types of probability estimates at the subcategory-level: the between-subcategory distinctiveness and within-subcategory representativeness. Both metrics measure the probabilities of a test data belonging to a certain subcategory. The difference is that the between-subcategory metric focuses on identifying the distinction between subcategories of different classes while the within-subcategory metric captures the representativeness of the test data by the particular subcategory.

3.1. Between-Subcategory Distinctiveness

The between-subcategory distinctiveness is obtained based on binary classification between a subcategory S_{ck} of class c and all subcategories $\{S_{c'k'}\}$ of the other classes $\forall c' \neq c$ and $k' = 1, ..., K_{c'}$. One binary classifier is trained for each subcategory using linear-kernel SVM. For a test data x, a set of probability estimates $\{P_b(x, S_{ck}) : \forall c, k\}$ is thus derived to describe the probabilities of x belonging to each subcategory using these trained binary classifiers [6].

3.2. Within-Subcategory Representativeness

The within-subcategory probability is derived based on the representativeness of the test data by a subcategory. Different from the between-subcategory metric, the within-subcategory metric utilizes the training data of a certain subcategory S_{ck} only and describes how well this subcategory represents the test data x. A better representation corresponds to a higher probability of x belonging to S_{ck}.

Specifically, given a test data x and a subcategory S_{ck}, we first obtain an approximated x'_{ck} by averaging the M-nearest neighbors of x from S_{ck}. In this way, x is adapted to the feature space of S_{ck}. We choose the simple nearest-neighbor computation rather than the other encoding techniques such as the LLC, since we do not want x to be overly well approximated by S_{ck} and lose the subcategory-specific characteristics of the feature space. Next, assume the center of S_{ck} is $f_{ck} \in \mathbb{R}^H$. The Euclidean distance between x'_{ck} and f_{ck} then describes the representativeness of x by the subcategory S_{ck}. The probability of x belonging to this subcategory is subsequently computed as:

$$P_w(x, S_{ck}) = \exp(-\|x'_{ck} - f_{ck}\|) \tag{4}$$

For the test data x, such within-subcategory probabilities are computed for all subcategories, and we obtain a set of probability estimates $\{P_w(x, S_{ck}) : \forall c, k\}$ at this step.

To derive the subcategory center f_{ck}, we adopt the support vector data description (SVDD) [39] method with the following objective:

$$\begin{aligned} &\min_{R_{ck}, f_{ck}} \; R_{ck}^2 + C \sum_{i=1}^{N_{ck}} \xi_i \\ &s.t. \;\; \|x'_{ick} - f_{ck}\|^2 \leq R_{ck}^2 + \xi_i, \;\; \xi_i \geq 0, \;\; \forall i \end{aligned} \tag{5}$$

where R_{ck} is a radius originating from the center f_{ck}, N_{ck} indicates the number of training data in S_{ck}, x'_{ick} denotes the approximated $x_i \in S_{ck}$ from the M-nearest neighbors in S_{ck} (excluding x_i), ξ_i is the slack variable and C is the trade-off parameter. With this construct, R_{ck} is expected to be the smallest radius enclosing the data in S_{ck} with slack variables ξ_i to accommodate the outliers. The objective function can be solved by introducing Lagrange multipliers α_i as detailed in [39]. The data samples corresponding to nonzero α_i form the support vectors, which are then linearly combined as the center f_{ck}:

$$f_{ck} = \sum_i \alpha_i x'_{ick} \tag{6}$$

4. Subcategory Fusion

We finally fuse the subcategory-level probabilities together with class-level classification probability to classify

the test data. The probability of test data x belonging to class c is defined as:

$$P(x,c) = P_m(x,c) + \sum_{k=1}^{K_c} w_{ck} q_{ck} \{\beta P_b(x, S_{ck}) + (1-\beta) P_w(x, S_{ck})\} \quad (7)$$

The first term $P_m(x,c)$ is the probability of x belonging to class c obtained using multiclass linear-kernel SVM, which is trained without considering the subcategory structures. The second term is the class-level probability fused from the subcategory-level probabilities. The constant β controls the ratio between the two subcategory-level probabilities. The two weighting factors, w_{ck} and q_{ck}, represent the contribution level and cluster quality of subcategory S_{ck}.

We suppose that higher contributions should come from subcategories that are more similar to x in the feature space. This level of similarity can be estimated by finding a sparse representation of x from the various subcategories. Specifically, we formulate the following function to obtain the weight vector w in a LLC construct:

$$\min_w \|x - Uw\|^2 + \lambda \|d \odot w\|^2 \quad s.t.\ \ \mathbf{1}^T w = 1, \ \ \|w\|_0 \leq P, \quad (8)$$

The weight vector w of dimension $J = \sum_c K_c$ is a concatenation of factors w_{ck} from all subcategories. The matrix $U \in \mathbb{R}^{H \times J}$ contains the approximated x'_{ck} from each subcategory S_{ck} (Section 3.2), and Uw is expected to represent x closely. The pairwise Euclidean distances between x and the approximated vectors are stored in d to enforce the locality constraints. The derived w is P-sparse and then used as the weight factor in fusing the subcategory probabilities.

Our second design consideration is that subcategories representing more compact and isolated clusters should carry higher weights. The weight factor q_{ck} is computed to quantify such cluster quality based on the Dunn index (DI) [13]. DI measures the ratio between the minimal inter-cluster distance to maximal intra-cluster distance. In our problem, each subcategory is a cluster and one index q_{ck} is computed per subcategory by the following:

$$q_{ck} = \frac{\min_{i,j} \|x_i - x_j\|}{\max_{i,i'} \|x_i - x_{i'}\|}, \quad \forall x_i \in S_{ck}, x_{i'} \in S_{ck}, x_j \in \{S_{c'k'}\} \quad (9)$$

with $c' \neq c$ and $k' = 1, ..., K_{c'}$. The vector containing all weight factors is finally normalized to mean value of 1.

5. Experiments

5.1. Datasets and Implementation

Three challenging texture datasets are used for our evaluation, including the KTH-TIPS2 [5], FMD [36] and DTD [8]. We choose these datasets since the state-of-the-art results on them are all below 80% [8], while the performance on the other popular datasets (e.g. UMD [44]) has become rather saturated with over 99% accuracy. Table 1 summarizes the characteristics of the three datasets.

Table 1. Summary of the datasets used.

Dataset	Num. of images	Num. of classes
KTH-TIPS2	4752	11
FMD	1000	10
DTD	5640	47

Two types of texture descriptors are applied to represent the images: the IFV [33] and Caffe [22] descriptors. The combination of IFV and deep convolutional network activation features (DeCAF) [11] has shown excellent texture classification accuracy in the state-of-the-art [8], achieving about 9% improvement over the previous best result. In addition, Caffe, being a more efficient and modularized framework, has recently been released to replace DeCAF. Since our focus is on the classification model rather than feature design, we choose to follow [8] and adopt IFV and Caffe as our feature descriptors.

IFV works by extracting local SIFT descriptors densely at multiple scales, reducing the local feature dimension using principal component analysis (PCA), and encoding them using a Gaussian mixture model (GMM) with multiple modes. Signed square-rooting and L2 normalization are also incorporated. The descriptor dimension is determined by the reduced local feature length and the number of modes with GMM. Similar to [8], SIFT descriptors are computed with a spatial extent of 6×6 pixels at scales $2^{i/3}, i = 0, ..., 4$ and sampled every two pixels. The 128-dimensional SIFT descriptor is reduced to 64 dimensions using PCA. For KTH-TIPS2 and FMD datasets, 64 Gaussian modes are used; while for DTD, 256 modes are used. We try to use a small number of modes for KTH-TIPS2 and FMD in order to reduce the feature dimension. More modes are assigned for DTD compared to the other two datasets since DTD contains a larger number of classes. The feature encoding toolbox [7] is adopted in our implementation.

Caffe features are computed using deep convolutional neural network [23], which is trained from the ImageNet challenge. The network involves several convolution, rectification, max pooling, and fully-connected layers. Similar to [8], we remove the softmax and last fully-connected layer of the network to obtain a 4096-dimensional descriptor, which is then L2 normalized. The same approach is applied to all three datasets.

The parameter settings in our sub-categorization classification model are summarized as follows. For both subcategory generation and fusion, the sparsity constant P is

Table 2. The classification accuracies (%) compared to the state-of-the-art.

Dataset	IFV		Caffe		IFV+Caffe		State-of-the-art	
	SVM	Ours	SVM	Ours	SVM	Ours	1st	2nd
KTH-TIPS2	65.4±2.8	71.5±2.9	73.0±2.4	75.1±2.5	75.4±3.0	**79.3±2.7**	76.0±2.9 [8]	66.3 [40]
FMD	56.5±1.5	61.2±1.3	65.3±1.8	66.0±1.4	65.2±1.2	**68.4±1.5**	65.6±1.4 [8]	57.1 [35]
DTD	58.4±1.8	62.3±1.9	53.2±1.4	60.4±1.3	65.1±1.4	**67.8±1.6**	64.7±1.7 [8]	–

Results of [8] are taken from http://www.robots.ox.ac.uk/~vgg/data/dtd/eval.html.

set to 15 and the balance parameter λ is 0.01. The number of subcategories is set to $N_c/20$ with N_c denoting the number of images of class c. For subcategory probabilities, the number of nearest neighbors M is set to 6. The ratio constant β in subcategory fusion is set to 0.2. The trade-off parameters C in the linear-kernel SVMs and Eq. (5) are all 1.6. We try to minimize the number of parameters by using the same setting for parameters of the same meaning (e.g. λ in subcategory generation and fusion) and across all three datasets. These parameter settings are empirically chosen based on a small subset of the datasets.

For training and testing, we follow the typical setup in the existing studies [40, 35, 8]. In KTH-TIPS2, each class has four samples and each sample has 108 images. One sample from each class is used for training and the other three samples are used for testing. In FMD, each class contains 100 images, from which half are randomly selected for training and the other half for testing. In DTD, each class contains 120 images, which are randomly divided into three parts for training, validation and testing. For each dataset, four splits are performed to obtain the overall results.

5.2. Results

5.2.1 Overall Performance

Table 2 lists the classification accuracies of our method and the compared approaches. The state-of-the-art [8] is based on IFV+DeCAF with linear-kernel SVM. With our configurations of IFV and Caffe descriptors, we obtained similar results to [8] when using SVM as the classifier. Note that we found the linear-kernel SVM was more effective than using polynomial kernels with about 10% difference in classification accuracy. With the sub-categorization model, we achieved about 3% improvement over [8]. The improvement over the second best previous results [40, 35] is about 13%, which is attributed to both the IFV+Caffe feature descriptor and the sub-categorization classification model.

Another finding is that our sub-categorization model provides different degrees of benefits when coupled with different texture descriptors for different datasets. Specifically, if the Caffe descriptor is used alone, the sub-categorization model achieves about 7% improvement for the DTD dataset, but only 2% and 0.7% improvement for KTH-TIPS2 and FMD. At the same time, Caffe with SVM provides much better accuracies than IFV with SVM for KTH-TIPS2 and FMD, but lower accuracies for DTD. These observations suggest that when the Caffe descriptor is more discriminative for a certain dataset, there is less scope to explore intra-class variation and inter-class ambiguity with subcategories. For KTH-TIPS2 and FMD, the benefit of modeling the subcategories is mainly from the IFV feature space. We thus also performed another set of experiments by generating subcategories and computing subcategory probabilities and fusion weights based on IFV only, while Caffe is only incorporated when training the multi-class SVM. The results obtained are similar to our final results listed in Table 2 with $< 0.5\%$ difference in accuracy.

It is also worth to note that for KTH-TIPS2 and FMD, we actually obtained higher accuracies using Caffe with SVM compared to DeCAF with SVM [8], and lower accuracies using IFV with SVM compared to that in [8] (exact numbers are referred to [8]). We suggest that this could be due to the improved framework of Caffe and the smaller numbers of Gaussian modes we used to reduce the feature dimension. The combined effect of IFV and Caffe with SVM is nevertheless similar to the results of [8]. For DTD, we used almost identical configurations for IFV as [8], and the performance using IFV or Caffe with SVM is very similar to those reported in [8].

The classification confusion matrices for the three datasets are shown in Figure 2. It can be seen that some classes obtained excellent accuracy while some are less accurate. For example, the classification accuracy of the fifth (cotton) class of the KTH-TIPS2 dataset is only 41.4%, which is the lowest among all the 11 classes. 37.3% cotton images are misclassified as the eighth (linen) class. This high misclassification rate can be explained by the high degree of visual similarity between the two classes. Example images are shown in Figure 3, and it can be seen that the images from different classes look very similar while those of the same class show considerable variations. This accuracy of 41.4% is however much higher than using SVM with IFV+Caffe, which is only 19.4%. In addition, with SVM, 20.7% cotton images are misclassified as wool, while with our method, this misclassification rate decreases to 7.8%. These differences indicate the advantage of our sub-

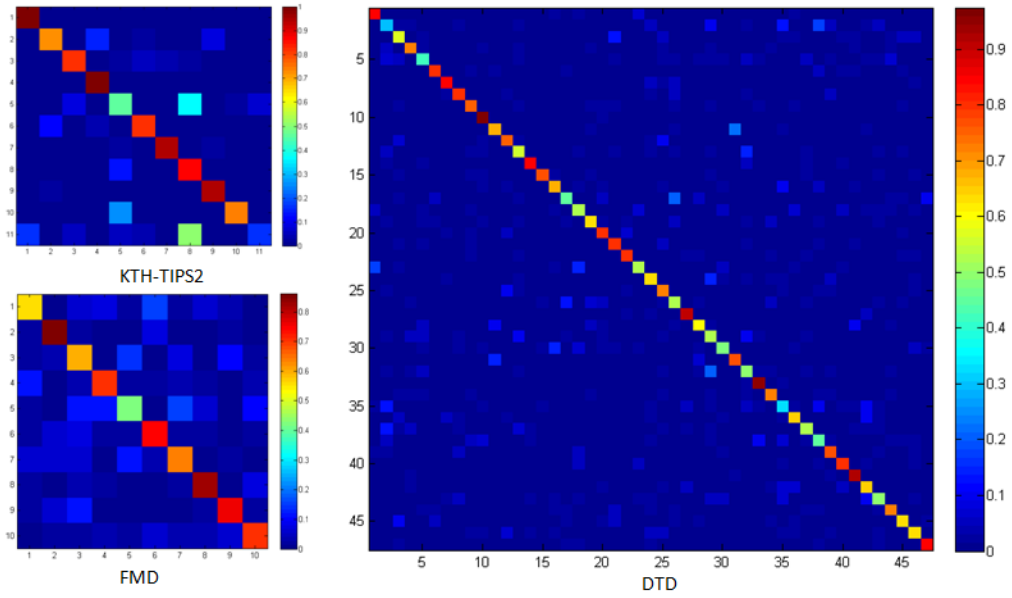

Figure 2. Confusion matrices of the three datasets.

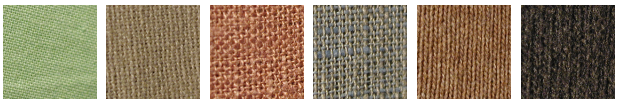

Figure 3. Images of the *cotton* (left two), *linen* (middle two), and *wool* (right two) classes from KTH-TIPS2, showing images of different samples at the scale 4.

categorization model in reducing the influence of intra-class variation and inter-class ambiguity.

Some example images that are misclassified by multiclass SVM but accurately classified by our method are shown in Figure 4. In many cases, such correction occurs when SVM produces similar probabilities for multiple classes. Our fused subcategory probabilities then add preference to the correct class. For example, for the first image shown from the KTH-TIPS2 dataset, SVM assigns the highest probability to the wood class. With our method, higher within-subcategory probabilities and contribution levels are obtained for subcategories of the correct brown bread class, compared to those of the wood class. The combined effects then counteract the lower probability of brown bread predicted by SVM and derive the accurate classification. As another example, for the first image shown from the FMD dataset, the highest SVM probability is assigned to the paper class. With our method, higher contribution levels and cluster qualities are allocated to subcategories of the correct foliage class, and higher between- and within-subcategory probabilities are obtained for these subcategories as well, compared to those of the paper class. As a result, this image could be accurately classified.

5.2.2 Component Analysis

Figure 5 illustrates the effects of the various components of our method. We compare the following approaches: (i) SVM: using multiclass SVM only for classification; (ii) Betw: for subcategory probabilities, only the between-subcategory distinctiveness is used; (iii) Within: only the within-subcategory representativeness is computed; (iv) Contr: for subcategory fusion, only the contribution levels are used as fusion weights; (v) Qual: only the cluster quality is incorporated as weights; and (vi) Ours: the complete sub-categorization method we proposed. In all these compared approaches, IFV+Caffe descriptors are used and the same training/testing setup is conducted. The results show that the various components have different effects on different datasets. For example, for the KTH-TIPS2 dataset, using the between-subcategory probability alone provides much lower performance than using the within-subcategory probability alone; while both probabilities when using alone produce similar results for the FMD dataset. This observation also suggests that although at the moment we use common parameter settings for all datasets, we can actually tune the parameters for individual databases based on the effects of different components for different datasets. Such tuning should further improve the classification performance, and we will investigate this in the future.

We further compare with other approaches for subcategory generation: (i) K-means clustering; (ii) SC: spectral clustering based on Euclidean distances; and (iii) LSVM:

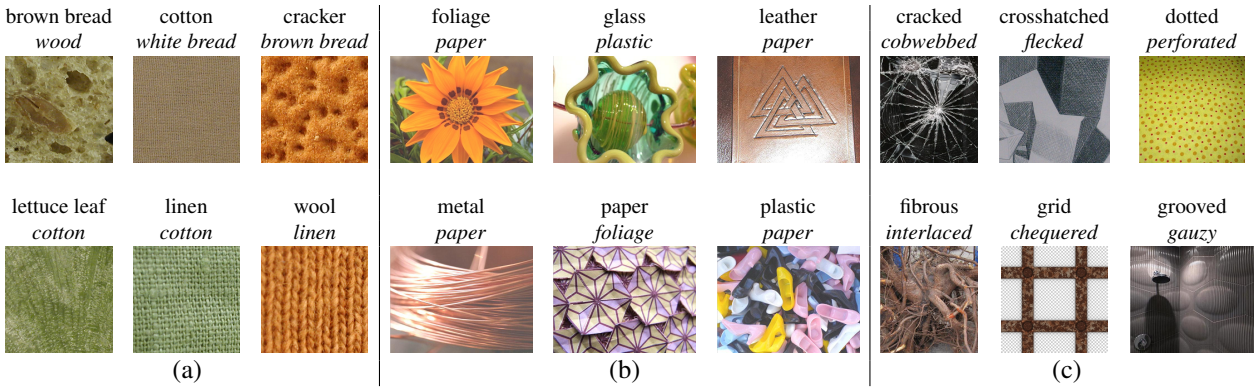

Figure 4. Example images that are correctly classified by our proposed method but misclassified by SVM, from (a) KTH-TIPS2, (b) FMD, and (c) DTD. The tags on top of each image indicate the image label (upper) and the label predicted by SVM (lower).

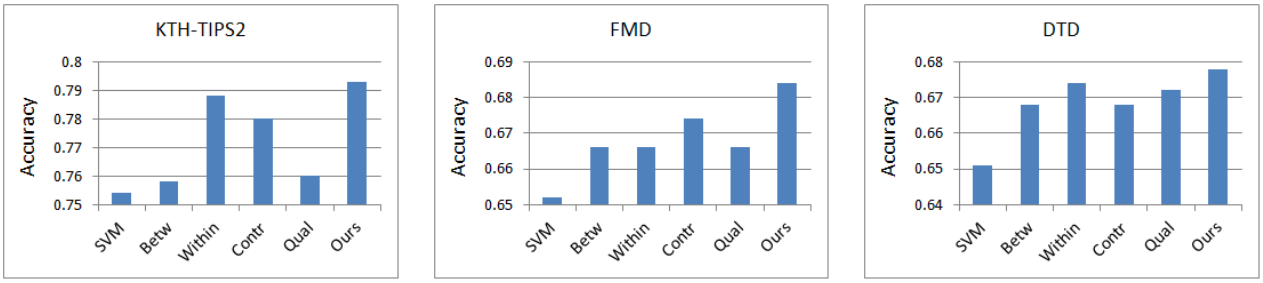

Figure 5. Classification accuracies evaluating the various components of our proposed method.

using latent SVM to obtain the subcategories while iteratively optimizing the classification objective [17, 10]. K-means and SC are used only to replace the subcategory generation component of our method. LSVM is used to directly obtain the classification results with the subcategories initialized by our subcategory generation method. As shown in Figure 6, our subcategory generation method provides better classification results than K-means and SC. The only difference between our method and SC is the way of computing the affinity matrix, and the performance difference indicates the advantage of our LLC-based sparse representation formulation. The LSVM approach can be considered as classifying based on the between-subcategory distinctiveness only, even though the subcategory assignments are iteratively refined based on the classification outputs. The advantage of our method demonstrates the benefit of integrating the multiclass, between- and within-subcategory probabilities with weighted fusion. We also suggest that better classification results using LSVM could possibly be obtained by improving the mining of training data, which is however not the focus of our current study.

Figure 7 shows the visual results of our subcategory generation method. For each dataset, we choose a class that we obtain large performance improvement over SVM to present the visual analysis. It can be seen that images of different subcategories tend to exhibit different textures (e.g. periodicity, directionality and randomness) even though they belong to the same class. For example, images of the gauzy class in the DTD dataset can contain smooth or stripped surfaces, and the resultant subcategories capture this difference in texture. Such differences can however be harder to identify for the FMD dataset compared to KTH-TIPS2 and DTD, since the images in FMD are generally more complex and the textures are often difficult to describe or categorize. Nevertheless, our visual analysis helps to verify that our method can discover the subcategory structure in the data and is thus effective in enhancing the texture classification performance by explicitly modeling the subcategory-level classification.

6. Conclusions

In this paper, we have presented a sub-categorization model for texture classification. The training images of each class are first clustered into subcategories with a locality-constrained subspace clustering method. We then design two subcategory-level probabilities to quantify the probability of a test image belonging to each subcategory, with between-subcategory distinctiveness and within-subcategory representativeness. Finally, a weighted fusion method is designed to fuse the subcategory probabilities

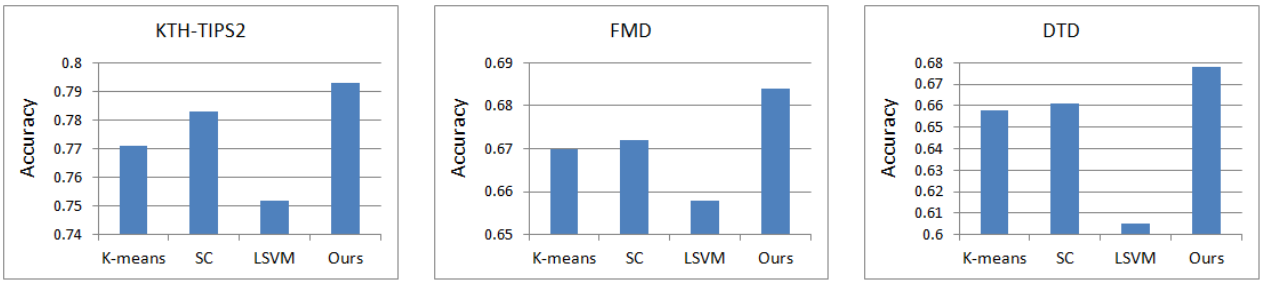

Figure 6. Classification accuracies evaluating the various ways of obtaining the subcategories.

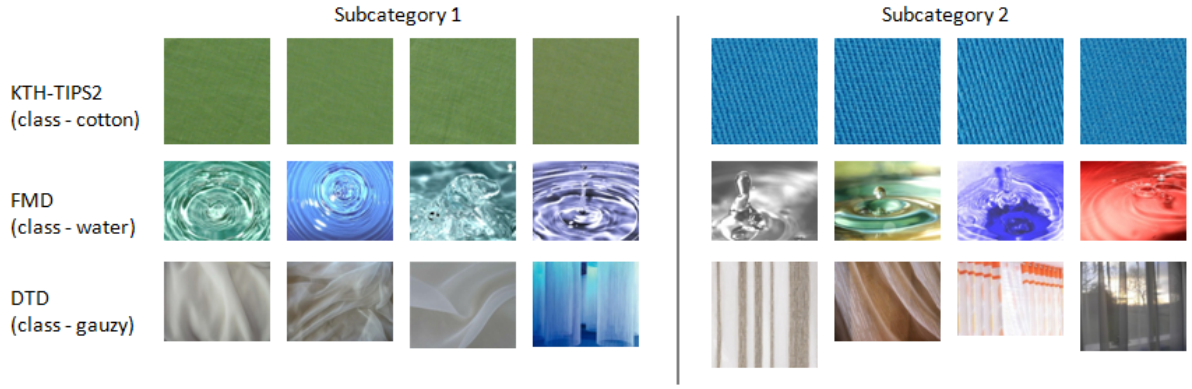

Figure 7. Visualization of subcategory generation results, showing two subcategories for each dataset with four example images.

with contribution levels and cluster qualities to derive the class-level probabilities. We have incorporated IFV and Caffe as the texture descriptors, and applied our method on three challenging datasets: KTH-TIPS2, FMD and DTD. We have shown that our method outperforms the state-of-the-art for all three datasets.

Acknowledgement

This work was supported in part by Australian Research Council (ARC) grants. Heng Huang was partially supported by US NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152, and NSF-DBI 1356628.

References

[1] O. Aghazadeh, H. Azizpour, J. Sullivan, and S. Carlsson. Mixture component identification and learning for visual recognition. *ECCV*, pages 115–128, 2012. 1

[2] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. *ECCV*, pages 663–676, 2010. 1

[3] N. Bhushan, A. Rao, and G. Lohse. The texture lexicon: understanding the categorization of visual texture terms and their relationship to texture images. *Cognitive Science*, 21(2):219–246, 1997. 1

[4] L. Bourdev, S. Maji, and J. Malik. Describing people: a poselet-based approach to attribute classification. *ICCV*, pages 1543–1550, 2011. 1

[5] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. *ICCV*, pages 1597–1604, 2005. 2, 4

[6] C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intelli. Syst. Technol.*, 2:1–27, 2011. 3

[7] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. *BMVC*, pages 1–12, 2011. 4

[8] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. *CVPR*, pages 3606–3613, 2014. 1, 2, 4, 5

[9] M. Crosier and L. D. Griffin. Using basic image features for texture classification. *Int. J. Comput. Vis.*, 88(3):447–460, 2010. 1

[10] S. K. Divvala, A. A. Efros, and M. Hebert. How important are "deformable parts" in the deformable parts model? *ECCV*, pages 31–40, 2011. 7

[11] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: a deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 4

[12] J. Dong, W. Xia, Q. Chen, J. Feng, Z. Huang, and S. Yan. Subcategory-aware object classification. *CVPR*, pages 827–834, 2013. 1

[13] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. 4

[14] E. Elhamifar and R. Vidal. Sparse subspace clustering. *CVPR*, pages 2790–2797, 2009. 2

[15] S. Escalera, D. M. J. Tax, O. Pujol, P. Radeva, and R. P. W. Duin. Subclass problem-dependent design for error-correcting output codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6):1041–1054, 2008. 1

[16] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *CVPR*, pages 1778–1785, 2009. 1

[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010. 7

[18] J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. *CVPR*, pages 3818–3825, 2014. 2

[19] C. Gu, P. Arbelaez, Y. Lin, K. Yu, and J. Malik. Multi-component models for object detection. *ECCV*, pages 445–458, 2012. 1

[20] M. Hoai and A. Zisserman. Discriminative sub-categorization. *CVPR*, pages 1666–1673, 2013. 1

[21] H. Ji, X. Yang, H. Ling, and Y. Xu. Wavelet domain multifractal analysis for static and dynamic texture classification. *IEEE Trans. Image Process.*, 22(1):286–299, 2013. 1

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 2, 4

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, pages 1–9, 2012. 4

[24] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Describing visual attributes for face verification and image search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(10):1962–1977, 2011. 1

[25] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1265–1278, 2005. 1

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

[27] F. Liu and R. W. Picard. Periodicity, directionality, and randomness: wold features for image modeling and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(7):722–733, 1996. 1

[28] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. *ICML*, pages 663–670, 2010. 2

[29] L. Liu, P. Fieguth, G. Kuang, and H. Zha. Sorted random projections for robust texture classification. *ICCV*, pages 391–398, 2011. 1

[30] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. *ECCV*, pages 347–360, 2012. 2

[31] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. J. Comput. Vis.*, 43(1):7–27, 2001. 1

[32] G. Patterson and J. Hays. Sun attribute database: discovering, annotating, and recognizing scene attributes. *CVPR*, pages 2751–2758, 2012. 1

[33] F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. *ECCV*, pages 143–156, 2010. 2, 4

[34] Y. Quan, Y. Xu, Y. Sun, and Y. Luo. Lacunarity analysis on image patterns for texture classification. *CVPR*, pages 160–167, 2014. 1

[35] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson. Recognizing materials using perceptually inspired features. *Int. J. Comput. Vis.*, 103(3):348–371, 2013. 1, 5

[36] L. Sharan, R. Rosenholtz, and E. H. Adelson. Material perception: what can you see in a brief glance? *Journal of Vision*, 9(8):784, 2009. 2, 4

[37] G. Sharma, S. ul Hussain, and F. Jurie. Local higher-order statistics (lhs) for texture categorization and facial analysis. *ECCV*, pages 1–12, 2012. 1

[38] L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. *CVPR*, pages 1233–1240, 2013. 1

[39] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004. 3

[40] R. Timofte and L. J. V. Gool. A training-free classifiation framework for textures, writers, and materials. *BMVC*, pages 1–12, 2012. 1, 5

[41] M. Varma and R. Garg. Locally invariant fractal features for statistical texture classification. *ICCV*, pages 1–8, 2007. 1

[42] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *CVPR*, pages 3360–3367, 2010. 3

[43] S. Wang, X. Yuan, T. Yao, S. Yan, and J. Shen. Efficient subspace segmentation via quadratic programming. *AAAI*, pages 519–524, 2011. 2

[44] Y. Xu, H. Ji, and C. Fermuller. Viewpoint invariant texture description using factal analysis. *Int. J. Comput. Vis.*, 83(1):85–100, 2009. 1, 4

[45] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories. *Int. J. Comput. Vis.*, 73(2):213–238, 2007. 1

[46] M. Zhu and A. Martinez. Subclass discriminant analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1274–1286, 2006. 1

[47] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? *BMVC*, pages 80.1–80.11, 2012. 1