Texture classification using Dense Micro-block Difference (DMD)

Rakesh Mehta and Karen Egiazarian

Tampere University of Technology, Tampere, Finland

Abstract. The paper proposes a novel image representation for texture classification. The recent advancements in the field of patch based features compressive sensing and feature encoding are combined to design a robust image descriptor. In our approach, we first propose the local features, Dense Micro-block Difference (DMD), which capture the local structure from the image patches at high scales. Instead of the pixel we process the small blocks from images which capture the micro-structure from it. DMD can be computed efficiently using integral images. The features are then encoded using Fisher Vector method to obtain an image descriptor which considers the higher order statistics. The proposed image representation is combined with linear SVM classifier. The experiments are conducted on the standard texture datasets (KTH-TIPS-2a, Brodatz and Curet). On KTH-TIPS-2a dataset the proposed method outperforms the best reported results by 5.5% and has a comparable performance to the state-of-the-art methods on the other datasets.

1 Introduction

Texture is an important attribute of an object or a material and has been widely utilized as a visual cue for image classification. A number of computer vision problems, such as material classification [1], face recognition [2], facial expression recognition [3], object detection [4] use the texture information from images. Therefore, the fundamental problem of texture classification is highly relevant. The texture classification system encounters the problems such as, the variations in scale, illuminations, rotation and the subtle difference in the different texture patterns. These challenges have not been addressed completely by the existing methods and require deeper analysis. This paper thoroughly studies above mentioned problem and proposes a method for texture classification by integrating the advancement in the field of key-points descriptors, image encoding approaches and compressive sensing.

A variety of approaches have been proposed for texture classification. A family of these algorithms represent an image using a subset of the features from an image patch. The examples of these algorithms include Co-occurring Histograms [5], Markov random Field [6], Gabor Filter Banks [7], Local Binary Patterns (LBP) [8] and Fractal Models [9]. The key idea behind these papers is that a discriminative information can be captured from the image patch. Different methods are used to capture this information from the patches, e.g. Xu et.

al. [9] use orientation histogram, LBP uses the sign of the difference of the pixel values, Gabor based methods [7] use the response of the Gabor filter banks, etc. These approaches consider texture as a local cue and global structure between the features is not taken into account. Another class of the texture classification methods are based on the Bag-of-Words (BoW) model of image representation [10], [11], [12], [13]. The BoW model encodes both, the local structure by using the local features to form the texton dictionary, and the global appearance by computing certain statistics to represent the distribution of the textons. Due to its generalized structure it has been widely applied in a number of computer vision applications.

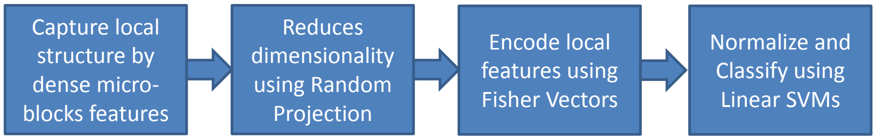
Capture local structure by dense micro-blocks features
Reduces dimensionality using Random Projection
Encode local features using Fisher Vectors
Normalize and Classify using Linear SVMs

Fig. 1. The flowchart of proposed approach.

In this paper we present a texture descriptor by combining novel patch based local features with the advancement in the BoW model. We first propose the Dense Micro-block Difference (DMD), which are based on the idea that the texture image repetitively exhibit a specific local structure which can provide discriminative information about it. Although the idea is inspired from the success of the pattern based approaches, the proposed method in practice is very different and addresses various shortcomings of these approaches. DMD are patch based features which are computed by comparing the intensities of smaller regions in it. It has been shown that the texture images being repetitive are compressible, which is further exploited in the proposed approach. The features are very fast to compute using integral image and low in dimensionality. The proposed local features are then combined with the BoW model of image representation. We use advanced encoding approach that has recently shown very promising results in the classification tasks. The combination of the proposed features with efficient encoding from the BoW model results in a robust image descriptor. Finally, the descriptor is combined with the linear SVM classifier which achieves state-of-the-art performance on a number of standard texture datasets (KTH-TIPS-2a, CUReT and Brodatz). The flowchart of the proposed approach is shown in Fig. 1.

The rest of the paper is organized as follows. In Section 2, we discuss the related work on the local features and the BoW encoding methods. In Section 3, we present the proposed approach for the texture classification. In Section 4, we study the key parameter of the descriptor and evaluate the performance of the proposed approach on three texture datasets. Finally, the paper is concluded in Section 5.

2 Related work and its analysis

Our work broadly draws inspiration from two different kind of approaches followed in texture classification. First, we call the local structure capturing features and the other is the Bag-of-Words based image representation model for feature encoding. In this section we provide a brief overview of the related work and analyse its shortcomings.

2.1 Local feature

A number of patch based local features are based on the idea that the local region of the image exhibits a certain characteristic structure and various methods have been employed to capture that structure. Among these LBP [8] has shown very promising results, and a number of modifications of LBP have been proposed [14], [15]. These features capture the local structure by taking the sign of the difference of image pixel values from the image patch in a circular geometry. The information provided by the magnitude of the difference is completely ignored, which results in the loss of the discriminative power of the features. To overcome this shortcoming, Tan et. al. [16] compared the magnitude with a predefined threshold parameter, while Guo et. al. [17] incorporated the magnitude by comparing it with a mean value of the image. The results from these approaches demonstrate that the magnitude provide a discriminative information which can be utilized in the patch based features.

Recently, Liu et. al. [18] proposed Sorted Random Projection (SRP) which utilize the compressibility of the pixel intensity difference taken from a circular geometry. Sharma et. al. [19] proposed Local Higher order Statistics (LHS), a descriptor that incorporates the high order statistics of the pixel difference from a patch. Approaches based on a pixel difference, such as LBP, SRP, LHS consider the circular geometry of the sampling points in the patch. Circular geometry can only capture the radial variation in the image patch as the difference is taken from the central pixel and its neighbours, all other directions are ignored. Although never applied in image classification, the geometry of the sampling points in the image patch has been studied in the binary key-point descriptors such as BRIEF [20], ORB [21]. Colander et. al [20] experimented with the five different kinds of geometries for sampling points and reported that random Gaussian sampling points outperforms the circular geometry and others. In ORB the sampling points are selected such that they have maximum variance in training samples and are uncorrelated.

The size of the patch is an important parameter for the patch based features. For LBP it was observed that the performance of the features improves with an increase in the patch size from 3×3 to 7×7. However, additional sampling points are required in large image patches to capture the intensity variations efficiently.

The dimensionality of these descriptors grows exponentially with the number of sampling points considered in the image patch. Therefore, using this encoding scheme, the number of sampling points cannot be increased substantially, which in turn also puts a restriction on the size of the image patch. We aim to study the encoding techniques that can overcome these shortcomings.

2.2 Encoding

The BoW model has been extensively utilized in the texture classification task [10], [11], [12], [13]. The main steps in this pipeline are: (1) Extract the features from the texture images, (2) Encode the features into an image descriptor, (3) Classify the image descriptor using a machine learning algorithm (e.g. Nearest Neighbour, Support Vector Machines (SVM)). While most of these papers use different kind of local features to capture the characteristic structure of the texture image, little importance is given to the feature encoding step. All these papers follow the encoding step of the vector quantization and hard assignment. In this step, first, the local features are extracted from the training images and, then exemplar features are chosen as the textons (using K-means clustering). These textons are used to label all the features from the training and the testing images. The local feature quantization, a common step in these approaches, is a lossy step as shown by Boiman [22]. Some attempts have been made to overcome this shortcoming. Farquhar et. al. [23] applied soft assignment using Gaussian Mixture Models. Yang, et. al. proposed sparse coding algorithm to replace the k-means which reduces the quantization error by applying less restrictive constraints. In this work we consider the Fisher kernel introduced by the Jaakola et. al. [24] and applied to image categorization by Sanchez et. al.[25]. It is an extension of BoW model as it not only encodes the zero order statistics but also the higher order statistics of the distribution of the local features.

3 Texture classification

In this section we present the proposed approach for the texture classification. First, we introduce the local features DMD which captures the intensity variation in an image patch. Next, the compressible nature of the dense feature is utilized to reduce its dimensionality. Finally, the local features are combined with the efficient Fisher encoding scheme to obtain the image descriptor.

3.1 Dense Micro-block Difference

The proposed features are based on the idea that the small patches in the texture image exhibit a characteristic structure and if captured efficiently, discriminative information can be obtained from it. Based on the ample evidence from the related work we use the intensity difference from the image patch to capture the variations in it. Furthermore, we believe that the individual pixels are more susceptible to noise and do not capture regional information, therefore we use small blocks in the image patch instead of the raw pixel values.

To encode the local structure of the patch we take the pairwise intensity differences of smaller blocks in the image patch. We address these smaller square blocks as ”micro-blocks” and their average intensity is considered for capturing variation in a patch. A image patch is usually of size 9×9 to 15×15 pixels and the micro-blocks are the smaller square region inside this patch. An illustration of the image patch and micro-blocks is provided in Fig. 2. The patch size is 21×21 and the micro-blocks are of sizes 2×2, 3×3 and 4×4. The micro-blocks pairs whose intensity difference is computed are shown color white and grey, and are connected with a line. For the sake of clarity here we show only eight micro-block pairs but in application we consider a much higher number. The high number of micro-block pairs assist in obtaining a rich and discriminative representation for a patch by capturing the variations in different directions and scales. It can be observed that the intensity difference is taken in different directions, unlike the LBP, SRP which only consider the radial direction. Furthermore, the distance between the micro-blocks is not constant, thus, the variations in patch are captured at different scales.

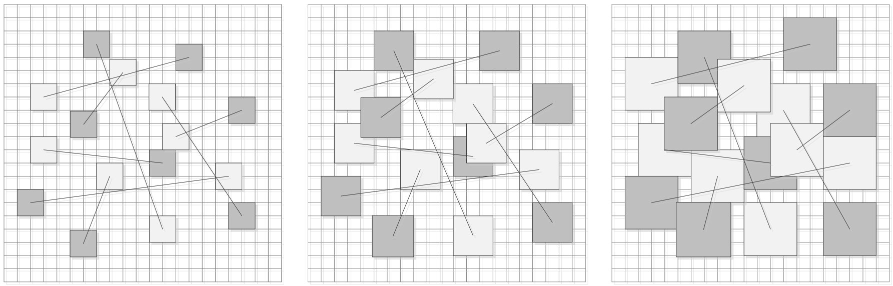

Fig. 2. The micro-blocks pairs in an image patch. Different micro-block sizes are shown (a) 2×2 (b) 3×3 and (c) 4×4.

Formally given a patch p of size $L\times L$ and two sets of image coordinates $X=\{\mathbf{x_1},\mathbf{x_2}...\mathbf{x_N}\}, Y=\{\mathbf{y_1},\mathbf{y_2}...,\mathbf{y_N}\}$ the DMD for the micro-blocks of size s is given as:

$$v(p)=\{M_s(\mathbf{x_1})-M_s(\mathbf{y_1}),M_s(\mathbf{x_2})-M_s(\mathbf{y_2}),...,M_s(\mathbf{x_N})-M_s(\mathbf{y_N})\} \quad (1)$$

where $M_s(\mathbf{x})$ is the average intensity of the pixel in micro-block located at position $\mathbf{x}=(a,b)^T$ in the patch and is given as

$$M_s(a,b)=\sum_{i=0}^{s-1}\sum_{j=0}^{s-1}p(a+i,b+j)/s^2, \quad (2)$$

and $p(a, b)$ denote the pixel intensity in patch p at location a, b.

The feature is completely specified by the following parameters: X, Y, L and s. The coordinate pair sets X and Y determine the location of sampling points in the patch and plays an important role in the design of the descriptor. Colander et. al. studied five different spatial arrangements for selecting the sampling point for keypoint matching. We follow the similar approach to Colander et. al. [20] and select the coordinates from isotropic Gaussian distribution i. e. $(X, Y) \sim$ i. i. d. $Gaussian(0, L^2/25)$. In this arrangement the coordinates are more densely distributed towards the center of the patch than towards its boundaries. Thus, larger weight is given to the center than to its boundaries, like SIFT features. The randomness in the sampling points coordinates help in capturing the variations at different scales because the distance between the sampling points $|x_i - y_i|$ is not constant. Moreover, we consider the magnitude of the difference without any thresholding, which helps in retaining the discriminative power of the features.

The computation of DMD for a image patch requires subtraction of N micro-blocks pairs. It involves the computation of the sum of the $2N$ micro-block. The micro-blocks sum can be efficiently implemented using the integral images. As summing a block using integral image requires 4 operation, for $2N$ micro-block $4 \times 2 \times N$ operations are required. Further, the computation of a feature requires N subtraction operations. The total number of operations required for a DMD feature computation is $9N$. Therefore, the complexity of the features computation is linear with the number of points considered in the image patch.

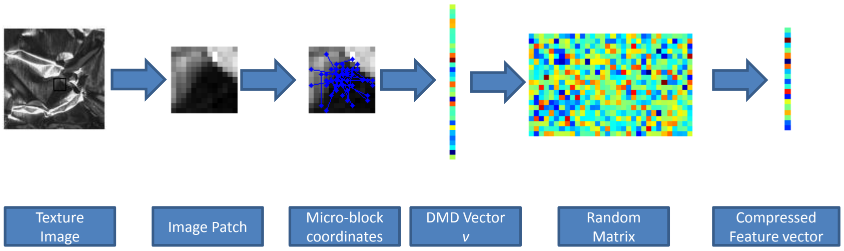

Fig. 3. Feature extraction from a patch of texture image.

3.2 Utilizing Compressibility

Using the above feature we obtain a N dimensional representation for an image patch. To efficiently capture the intensity variations in the image patch, we select a large number of sampling points which results in high dimensional features. Considering the fact that image patches are sparse in nature, we aim to take advantage of this property of the texture images. To reduce the dimensionality of the vector $v(g)$ and to make it more compressed, we utilize the Random

Projections (RP) [26]. The RP exhibit important properties of dimensionality reduction and information preservation. It is based on the idea that if the signal lies in a low dimensional manifold and is represented in a high dimensional ambient space, then, a small number of random projections of that signal preserve most of the information from it.

The random projection of the vector v is defined as:

$$d(g) = \Phi\, v(g) \tag{3}$$

where Φ is a $C \times N$ matrix, with $C << N$. With $C << N$ a loss in information is expected, however, if the signal is sparse and the matrix Φ exhibits the Restrictive Isometric Property (RIP) then the information is shown to be preserved during this transformation [27]. A number of matrices have shown to exhibit RIP property with high probability[27]. We use Gaussian random matrix as Φ (more details about it are provided in the implementation section). Fig. 3 shows feature computation and compression for a single patch from a texture image. First, a patch is selected from the image and the DMD features are extracted using the set of coordinate pairs. Then, the DMD vector is projected using a random matrix to obtain the compressed feature vector. The projection is obtained by multiplying the DMD vector with the random matrix.

The projection using random matrix reduces the dimensionality from N to C. To analyse the impact of feature projection using random matrix on texture classification, we perform a test on the KTH-TIPS-2a dataset. KTH-TIPS-2a is a commonly used texture dataset and we follow the standard test protocol, where half of the images from each class are used for training and rest half for testing. We sample the 64 micro-blocks features from image patch of size 15×15 and apply random projection on it. The reduced dimensionality C is varied from 10 to 50. The classification accuracy is shown in Table 1. It can be observed that the classification accuracy increases with dimensionality of the compressed vector upto a point after which it becomes constant. As the dimensionality increases more information is captured from the features, however after a certain point due to redundancy in the dense DMD features the accuracy stays constant. It is also interesting to note that even with a fairly low dimensionality of 10, high accuracy is achieved. Based on these results, the dimensionality of the compressed feature vector is set to 40 in all our further tests and the number of sampling points are set to 64.

Table 1. Effect of Gaussian component on the accuracy of KTH-TIPS-2a dataset.

Dimension	10	20	30	40	50	No RP
Accuracy	76.83%	77.23%	77.87%	78.54%	77.67 %	77.31%

3.3 Encoding

After computing the local features, we obtain a feature vector for each patch. Since we compute the features densely from an image, we get thousands of feature vector for an image. The features from a image have to be encoded to obtain a descriptor. Most of the early work on texture classification use the quantization/hard assignment for this purpose, however it leads to quantization error. Instead we use soft quantization by representing the feature distribution using Fisher Vector. It uses generative models for feature extraction by representing data by means of the gradient of the data log-likelihood w.r.t. the model parameters. The Fisher Vector use the Gaussian Mixture Models (GMM) to derive a probabilistic representation of the compressed features. The encoding captures the first and the second order differences between the image descriptors and the GMM centres. The higher order statistics that are learnt, provide a robust representation compared to other encoding methods such as histograms and kernel codebook.

The local features are modelled using GMM which is defined as:

$$p(\mathbf{d}|\theta) = \sum_{k=1}^{K} p(\mathbf{d}|\mu_k, \varSigma_k)\pi_k \tag{4}$$

where, $p(\mathbf{d}|\mu_k, \varSigma_k)$ is the multivariate Gaussian distribution with mean, μ_k, and covariance matrix, $\varSigma_k$, (assumed to be diagonal), π_k is the mixing coefficient of the Gaussian components and $\theta = (\pi_1, \mu_1, \varSigma_1 ... \pi_K, \mu_K, \varSigma_K)$ is the vector of the parameters for the model. K is the total number of Gaussian components assumed to be present while modelling the feature distribution. The parameters of the GMM are learned using Expectation Maximization (EM) using the features from the training samples.

Given the model, Fisher Vector is characterized by the gradient with respect to the parameter of the models. Thus, the gradient is computed with respect to the mean μ_k and the covariance $\varSigma_k$ of the GMM. It is given as:

$$\frac{\partial \log p(\mathbf{d}|\theta)}{\partial \mu_k} = h_k \varSigma_k^{-1}(\mathbf{d} - \mu_k), \tag{5}$$

$$\frac{\partial \log p(\mathbf{d}|\theta)}{\partial \varSigma_k^{-1}} = \frac{h_k}{2}(\varSigma_k - (\mathbf{d} - \mu_k)^2), \tag{6}$$

where,

$$h_k = \frac{\pi_k p(\mathbf{d}|\mu_k, \varSigma_k)}{\sum_k \pi_k p(\mathbf{d}|\mu_k, \varSigma_k)}. \tag{7}$$

The Fisher encoding is obtained by concatenating the parametric gradient for all the K components of the GMM. Thus, the length of the feature vector is 2KC, where C is the dimensionality of the compressed features. After concatenation, we apply $l2$ and power normalization [25] on the feature vectors. The $l2$ normalization helps in compensating for the fact that different images contain different amount of relevant information. The power normalization ($z \leftarrow sign(z)|z|^{\rho}$)

helps to ‘unsparsify’ the feature vector that becomes sparse when the number of Gaussian components in GMM are increased.

4 Experiments

To analyse the performance of the proposed descriptor, we conduct extensive experiments on three standard publicly available texture datasets: KTH-TIPS-2a, Brodatz and CUReT. We follow the standard protocol for testing.

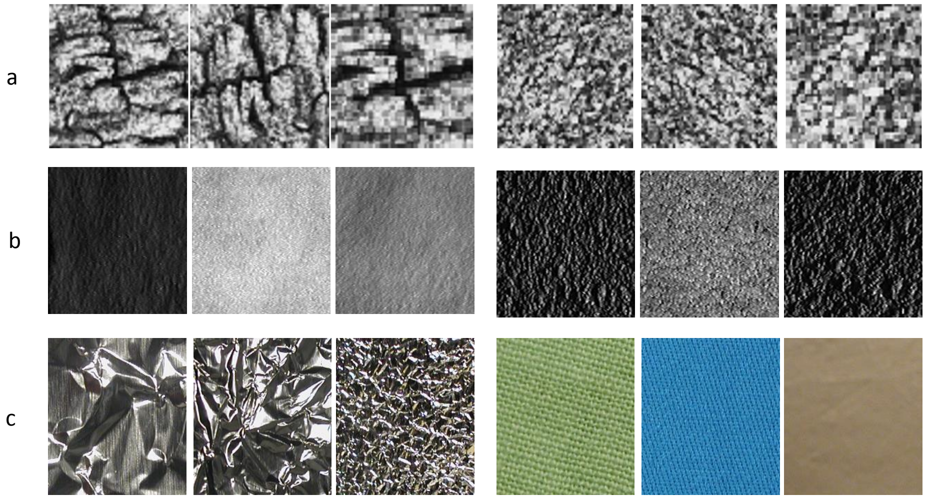

Fig. 4. The samples image from three different datasets, (a) Brodatz, (b) CUReT and (c) KTH-TIPS-2a.

The KTH-TIPS2-a texture dataset [1] contains 11 texture classes (e.g. cork, wool, linen, etc.) with 4,395 images. The images are 256 × 256 pixels in size, and they are transformed into 256 gray levels. Each texture class consists of images from four different samples. The images for each sample are taken at nine scales, under four different illumination directions, and three different poses. The variations in scales, illumination and pose makes it a challenging dataset. We use the standard testing protocol [28], [1] where at each run the three sample sets are used for training and fourth samples images for testing.

The original Brodatz dataset [29] has 32 texture classes with 16 images per class. The images are of dimension 64 × 64. To make the test more challenging, three samples are generated from each image by (1) rotating (2) scaling and (3) both rotating and scaling the original images. The resulting images are resized to 200×200 pixels, converted to grayscale and histogram normalized. Therefore, the final test set-up consists of 2048 images with 64 images in each class. Following the usual protocol in our experiment [28], we randomly select 32 images from each class for training and rest are used for testing. The accuracy is reported on 5 fold cross validation.

CUReT database [30] consist of 61 classes each containing 205 images taken under range of viewing and lighting angles. Following the usual protocol we select only 92 images per class which afford the extraction of 200 × 200 pixels foreground region of a texture. It is a challenging set for classification because of intra-class variation in appearance resulting from the different illumination conditions. In our tests we varied the number of training samples in this dataset to observe the effect of varying number of training samples on the performance.

The samples images from these datasets are shown in Fig. 4. It shows three different images from two different classes of each dataset.

First, we provide details about our implementation, then, we study the effect of certain parameters involved in the descriptor on classification performance. Finally, we compare the obtained results with state-of-the-art approaches.

4.1 Implementation details

The DMD features are extracted from the grid with a spacing of 3 pixels. It is observed that the performance of the features is maintained as long as the size of the grid does not exceed 5 pixels. With a larger grid size, the local structure is not captured efficiently and for a denser grid spacing the number of features becomes too large with no significant increase in performance. The number of sampling pairs in all our experiments is fixed to 64.

The matrix Φ is a Gaussian random matrix that is normalized to zero mean and unit variance. It is of dimension $C \times N$, where N is the dimension of the DMD vector while C is the dimension of the compressed features. The values of C is set to 40 in our experiments. The GMM parameters are estimated using 500,000 feature vectors that are randomly sampled from the training images. The center for GMM are initialized with k-mean clustering.

The parameter ρ is set to 0.5 for the power normalization. In all our experiments SVM classifier with linear kernels is used. The linear SVM requires less training time over types of kernels, during the testing it only requires a simple dot product. Another advantage of the linear kernel is that it directly operates on the feature, thus any improvement in the classification performance can be attributed to the features rather than the classifier.

4.2 Patch and Micro-block size

In the proposed approach the information is being captured at two levels, first is the patch level and the other is the micro-blocks level. When we increase the micro-block size, the overlap between them also increases as shown in Fig. 2. If the patch size is small it would lead to repetitive overlap of the micro-blocks and consequently results in redundant and correlated features. Therefore the patch size should be big enough to allow sufficient degree of spatial freedom to

Table 2. Recognition rate for KTH-TIPS-2a dataset

	Patch Size			
Micro-block size	9 × 9	11 × 11	13 × 13	15 × 15
1 × 1	73.45%	74.75%	73.99%	74.10%
2 × 2	76.62%	76.69%	77.71%	76.63%
3 × 3	76.24%	77.55%	78.04%	78.02%
4 × 4	76.91%	77.89%	77.90%	78.43%
5 × 5	75.03%	76.70%	77.58%	**78.54**%

micro-blocks. Since the patch size and micro-block size are dependent on each other, we vary these parameters jointly in our experiments. The size of the patch is varied from 9 × 9 to 15 × 15 with a step size of 2 and the micro-block size is varied from 1 to 5. The tests are performed on KTH-TIPS-2a and Brodatz dataset. The results for both these datasets are shown in Table 2 and Table 3.

It can be observed from the results that for both these datasets there is an increase in the accuracy with an increase in the patch and micro-block size. The accuracy improves with increase in the micro-block size specially for large patch sizes. The observation supports our claim that the micro-block are more efficient element for capturing the information than pixels.

Table 3. Recognition rate for Brodatz dataset

	Patch Size			
Micro-block size	9 × 9	11 × 11	13 × 13	15 × 15
1 × 1	98.96%	99.24%	99.22%	99.24%
2 × 2	98.71%	99.18%	**99.32**%	99.26%
3 × 3	98.81%	99.12%	99.18%	98.98%
4 × 4	98.30%	98.89%	98.81%	98.59%
5 × 5	96.82%	98.05%	98.20%	98.32%

4.3 Number of Gaussian Components

The number of Gaussian Components used for modelling the features distribution plays an important role in the encoding step. With more Gaussian components the distribution can be modelled in a suitable manner, but, it also leads to an increase in the dimensionality. To analyse its role, we varied the number of components from 32 to 128 with the step size of 32 and performed the classification test on the KTH-TIPS-2a dataset. The accuracy for the dataset is shown in Table 4. As expected there is an increase in the accuracy with more Gaussian components due to better modelling of local features, however after a point the accuracy stays constant. Based on the results from these experiments in all our tests we use 128 Gaussian components. The Gaussian components can be further increased, however it will also result in an increase in the dimensionality of the descriptor, without any significant increase in the performance.

Table 4. Effect of Gaussian component on the accuracy of KTH-TIPS-2a dataset.

Gaussian Component	32	64	96	128
Accuracy	76.26%	77.19%	77.89%	78.54%

4.4 Varying number of samples

We study the effect of different number of training samples on the proposed approach. To evaluate the performance, we conduct tests by varying the number of training samples in the CUReT dataset. We follow the protocol of [31], where the three different training scenarios are used. The number of training samples is set to 46, 23 and 3, while the rest of the images are used for testing. The block size for DMD is set to 13 × 13 and the micro-block size is fixed to 3 × 3. The results are shown in Table 5. The classification performance is compared with LBP, LBPV [14], LBPHF [32], MR8 [12], BIF [13] and recently proposed M-BIMF [31]. As expected, the performance drops with the less training samples. However, it is interesting to observe that accuracy decreases only by a few percents when samples are reduced from 46 to 23, however when the samples reduce to 3 from 23 the drop in accuracy is between 30 to 40 percent. It shows that with the 23 training samples the texture can be modelled considerable well, although it is not the case with 3 samples.

The proposed approach achieves the highest accuracy when the training samples are 46 and 3. When the number of samples is 23 then we achieve 93.66, which is second only to the M-BIMF features proposed recently. Even with the three samples we achieve an accuracy of 66.76 which is significantly higher than LBP, MR8, LBPHF methods.

4.5 Comparison with State-of-the-art

In this section we compare our results on the Brodatz and KTH-TIPS-2a with a number of state-of-the-art texture classification approaches. The algorithms used for comparison are LBP, LTP, Local Quantized Patterns (LQP) [33], Weber Law Descriptor (WLD) [28], Caputo et. al. [1] and Local Higher order Statistics (LHS) [19]. The LBP and LTP are computed by the binary and ternary thresholding of the pixel difference in the local circular neighbourhood and use histogram as the encoding method. LQP is also a pattern based descriptor, however the

Table 5. Recognition rate for CUReT dataset

Method	T_{46}	T_{23}	T_3
LBP	87.91	87.54	51.30
LBPV	78.83	73.71	42.03
LBP-HF	88.77	87.54	57.04
M-LBP	94.79	93.87	60.97
MR8	93.52	91.48	58.68
BIF	95.81	91.95	66.40
M-BIMF	95.62	**94.59**	65.11
Proposed	**97.32**	93.66	**66.76**

number of patterns sampled are very large, which are quantized using k-mean clustering. LHS performs the Fisher encoding of the pixel difference with LBP like geometry. WLD captures the local pattern from the gradient images. The results of all these approaches on the Brodatz and KTH-TIPS-2a dataset are shown in Table 6.

It can be observed from the results that the proposed features achieves the best results on KTH-TIPS-2a dataset. This is the highest accuracy reported on the KTH-TIPS-2a dataset to the best of our knowledge. It is interesting to note that the accuracy is still far from being perfect even for the best results. The first reason is that the variations in this dataset are much stronger than other texture datasets such as Brodatz, etc, for which nearly perfect accuracy can be achieved. Another reason for a lower accuracy on this dataset is the testing protocol for this dataset. Since the three samples are used for training and the fourth sample for testing, there is a considerable difference between the training and the testing images. The images from the three samples for two texture class are shown in Fig 4. It can be seen that there is a significant difference in the images. Thus, to perform on this dataset the algorithm should have a generalization property. The high recognition rate of the proposed algorithm shows that it also has a generalization property and can easily adapt to the variations during training and testing.

The comparison of the accuracies on KTH-TIPS-2a, shows that the LBP and LTP are inferior to the state-of-the-art descriptor LHS. LTP achieves higher accuracy than LBP because it has three quantization levels compared to two levels of LBP. Since LHS has even more quantization levels than LTP, there is a further increase in the performance from LTP to LHS. Therefore, we can infer that with more quantization levels the pixel difference is modelled in a better way, hence an improvement in performance is observed. Although DMD and LHS both have same number of quantization levels, the gain of DMD over LHS can be attributed to the fact that DMD captures the information from the patches at the multiple scales rather than the single scale that is used in LHS. Also the compressed vectors of DMD, by means of random projection, capture the inherent structure of the patch in an effective way. The proposed method

Table 6. Recognition rate for Brodatz and KTH-TIPS-2a datasets.

Method	KTH-TIPS-2a	Brodatz
WLD	64.7	97.5 ±0.6
LQP	64.2	96.9
LBP	69.8 ±6.9	87.2 ±1.5
LTP	71.3 ±5.3	95.0 ±0.8
Caputo et. al [1]	71.0	-
LHS	73.0 ±5.7	**99.3** ±0.3
Proposed	**78.5** ±4.6	**99.3** ±0.5

outperforms the LBP, LTP by 8.7%, 7.2% respectively. Compared to state-of-the-art descriptors, WLD, Caputo et. al. [1] and LHS the proposed approach shows a significant improvement of 13.8%, and 7.5% and 5.5% respectively.

For Brodatz dataset a near perfect recognition rate is shown by the LHS and DMD descriptors, which achieve more than 99%. The proposed approach gains by more than 12% over LBP and by 3% over WLD. It can be seen that for Brodatz dataset all the descriptors achieve better recognition rate compared to the KTH-TIPS-2a dataset. The variation between the training and the testing samples are not as high as the previous dataset as the samples for both are taken from similar image samples. It is easier to model the texture samples and moreover it does not require the generalization property.

An important advantage of the DMD is its speed. We compare the computation time of DMD with the SIFT features. SIFT and other gradient based features are very frequently used for texture classification. The computation time for the DMD features for an image from KTH-TIPS-2a dataset is 0.28 seconds. For the same setting the computation time of the SIFT features is 25.67 seconds on a standard computer. The SIFT features are also densely computed with a grid of 3×3 pixels. The computation times for SIFT features is almost 100 times more than the DMD features, which make the proposed feature favourable for the real time applications.

5 Conclusion

We presented a novel approach for texture classification based on block based features and Fisher Vector encoding. The proposed DMD features capture the local structure from texture images with the help of micro-blocks. These are very fast to compute, easy to implement and discriminative in nature. When combined with efficient coding technique we obtain a robust texture descriptor. The tests performed on challenging datasets demonstrated the efficiency of the proposed approach.

Acknowledgement.The research leading to this paper partially recieved funding from TUT project Big Data 83255.

References

1. Hayman, E., Caputo, B., Fritz, M., Eklundh, J.O.: On the significance of real-world conditions for material classification. In: Computer Vision-ECCV 2004. Springer (2004) 253–266
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28** (2006) 2037–2041
3. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing **27** (2009) 803–816
4. Trefnỳ, J., Matas, J.: Extended set of local binary patterns for rapid object detection. In: Proceedings of the Computer Vision Winter Workshop. Volume 2010. (2010)
5. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural features for image classification. Systems, Man and Cybernetics, IEEE Transactions on (1973) 610–621
6. Cross, G.R., Jain, A.K.: Markov random field texture models. Pattern Analysis and Machine Intelligence, IEEE Transactions on (1983) 25–39
7. Bovik, A.C., Clark, M., Geisler, W.S.: Multichannel texture analysis using localized spatial filters. Pattern Analysis and Machine Intelligence, IEEE Transactions on **12** (1990) 55–73
8. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24** (2002) 971–987
9. Xu, Y., Ji, H., Fermüller, C.: Viewpoint invariant texture description using fractal analysis. International Journal of Computer Vision **83** (2009) 85–100
10. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. Pattern Analysis and Machine Intelligence, IEEE Transactions on **27** (2005) 1265–1278
11. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. International journal of computer vision **73** (2007) 213–238
12. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. International Journal of Computer Vision **62** (2005) 61–81
13. Crosier, M., Griffin, L.D.: Using basic image features for texture classification. International Journal of Computer Vision **88** (2010) 447–460
14. Guo, Z., Zhang, L., Zhang, D.: Rotation invariant texture classification using lbp variance (lbpv) with global matching. Pattern recognition **43** (2010) 706–719
15. Mehta, R., Egiazarian, K.: Rotated local binary pattern (rlbp) - rotation invariant texture descriptor. ICPRAM (2013) 497–502
16. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. Image Processing, IEEE Transactions on **19** (2010) 1635–1650
17. Guo, Z., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. Image Processing, IEEE Transactions on **19** (2010) 1657–1663
18. Liu, L., Fieguth, P., Kuang, G., Zha, H.: Sorted random projections for robust texture classification. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 391–398

19. Sharma, G., ul Hussain, S., Jurie, F.: Local higher-order statistics (lhs) for texture categorization and facial analysis. In: Computer Vision–ECCV 2012. Springer (2012) 1–12
20. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: Computer Vision–ECCV 2010. Springer (2010) 778–792
21. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 2564–2571
22. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
23. Farquhar, J., Szedmak, S., Meng, H., Shawe-Taylor, J.: Improving" bag-of-keypoints" image categorisation: Generative models and pdf-kernels. (2005)
24. Jaakkola, T., Haussler, D., et al.: Exploiting generative models in discriminative classifiers. Advances in neural information processing systems (1999) 487–493
25. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: Theory and practice. International journal of computer vision **105** (2013) 222–245
26. Donoho, D.L.: Compressed sensing. Information Theory, IEEE Transactions on **52** (2006) 1289–1306
27. Candes, E.J., Tao, T.: Decoding by linear programming. Information Theory, IEEE Transactions on **51** (2005) 4203–4215
28. Chen, J., Shan, S., He, C., Zhao, G., Pietikainen, M., Chen, X., Gao, W.: Wld: A robust local image descriptor. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010) 1705–1720
29. Valkealahti, K., Oja, E.: Reduced multidimensional co-occurrence histograms in texture classification. Pattern Analysis and Machine Intelligence, IEEE Transactions on **20** (1998) 90–94
30. Dana, K.J., Van Ginneken, B., Nayar, S.K., Koenderink, J.J.: Reflectance and texture of real-world surfaces. ACM Transactions on Graphics (TOG) **18** (1999) 1–34
31. Pan, J., Tang, Y.Y.: Texture classification based on bimf monogenic signals. In: Computer Vision–ACCV 2012. Springer (2013) 177–187
32. Ahonen, T., Matas, J., He, C., Pietikäinen, M.: Rotation invariant image description with local binary pattern histogram fourier features. In: Image Analysis. Springer (2009) 61–70
33. ul Hussain, S., Triggs, B.: Visual recognition using local quantized patterns. In: Computer Vision–ECCV 2012. Springer (2012) 716–729