Compact color–texture description for texture classification

Fahad Shahbaz Khan\textsuperscript{a,\ast}, Rao Muhammad Anwer\textsuperscript{b}, Joost van de Weijer\textsuperscript{c}, Michael Felsberg\textsuperscript{a}, Jorma Laaksonen\textsuperscript{b}

\textsuperscript{a} Computer Vision Laboratory, Linköping University, Sweden
\textsuperscript{b} Department of Information and Computer Science, Aalto University School of Science, Finland
\textsuperscript{c} Computer Vision Center, CS Dept. Universitat Autonoma de Barcelona, Spain

1. Introduction

Classifying textures is a difficult problem in computer vision and pattern recognition. The task is to associate a class label to its respective texture category. In recent years, a variety of texture description approaches have been proposed\ [30,10,20,5,9,52,12,41]. These approaches can be divided into two categories, namely sparse and dense representations. The sparse representation works by detecting feature points either based on interest point or dense sampling strategy. Feature description is then performed on these sampling points\ [20,49]. The second strategy, dense representations, involves extracting local features for each pixel in an image\ [30,10,5]. In this paper, we investigate the problem of texture classification using dense local texture representations.

A variety of texture description approaches exist in literature\ [30,10,20,5,9,52]. One of the most successful approaches is that of Local Binary Patterns (LBP)\ [30] based image representations. Other than texture classification, LBP have been successfully employed to solve other vision problems as well, such as object detection\ [48], face recognition\ [1] and pedestrian detection\ [42]. LBP describes the neighbourhood of a pixel by its binary derivatives which are used to form a short code to describe the pixel neighbourhood. A variety of LBP variants have been proposed\ [10,47,45]. Combining multiple texture features, such as variants of LBP features, is still an open research problem. The work of Guo et al.\ [9] proposes a learning framework to combine variants of LBP features for texture classification. Tan and Triggs\ [38] propose to combine Gabor wavelets and LBP features for the problem of face recognition. In this paper, we propose to use a heterogeneous feature set by combining multiple texture description methods.

Combining multiple texture description methods have an inherent problem of high-dimensional final image representations. Recently, Elfiky et al.\ [7] proposed to use a divisive information theoretic clustering (DITC) method\ [6] to counter the problem of high-dimensionality of bag-of-words based spatial pyramid
representations. The DITC compression was shown to reduce the dimensionality of image representations without any significant loss in accuracy. Similar to the work of Elfiky et al. [7], we propose to use the DITC approach to compress the high-dimensional multi-texture representation. However, different to the work of Elfiky et al. [7], here we investigate compressing a multi-texture histogram to obtain a single heterogeneous texture representation.

Generally, state-of-the-art texture descriptors operate on grey level images thereby ignoring the color information. Color in combination with shape features has been shown to yield excellent results for object recognition [32,18,19], object detection [16] and action recognition [15]. Color description is a challenging problem due to significant variations in color caused by changes in illumination, shadows and highlights. Recent works have shown that an explicit color representation improves the performance for object recognition [18,19], object detection [16] and action recognition [15]. In this paper, we perform a comprehensive evaluation of pure color descriptors, popular in object recognition, for the task of texture classification.

Contributions: We first show that combining multiple texture description methods significantly improves the performance compared to using the single best texture method alone. We further propose to use information theoretic compression approach to compress high-dimensional multi-texture features into a compact heterogeneous texture representation. Finally, we provide a comprehensive evaluation of color features, popular in object recognition, for the task of texture classification. This paper extends our earlier work [17] for texture classification that only evaluated the contribution of color for texture recognition. Beyond the work in [17], here we investigate the problem of combining multiple local texture descriptors for robust texture description. We perform extensive experiments on four challenging texture datasets namely, KTH-TIPS-2a, KTH-TIPS-2b, FMD and Texture-10.

The results of our experiments clearly demonstrate that combining multi-texture descriptors significantly improves the performance compared to the single best method alone. We further show that multi-texture representations can be compressed efficiently without any significant loss in accuracy. Finally, our comprehensive evaluation of color features suggest that discriminative color names outperform other color descriptors for texture recognition. By combining the best color descriptor with our compact heterogeneous texture representation provides state-of-the-art results on three of the four texture datasets.

The paper is organized as follows. In Section 3 we investigate the problem of combining multiple texture descriptors. A comprehensive evaluation of pure color descriptors for texture description is provided in Section 4. In Section 5 we provide experimental results. Section 6 finishes with concluding remarks.

2. Related work

A variety of texture description approaches have been proposed in recent years [30,10,20,5,9,52,47,22]. Varma and Zisserman [41] propose a statistical approach for texture modeling using the joint probability distribution of filter responses. A multiresolution approach based on local binary patterns (LBP) is proposed by Ojala et al. [30] for gray-scale and rotation invariant texture classification. The LBP is one of the most successful approaches for texture classification with several variants existing in literature [10,47,45]. Chen et al. [5] propose a method based on Weber’s law consisting of two components namely differential excitation and orientation. An image is represented by concatenating the two components in a single representation. Ul Hussain and Triggs [12] introduce an approach that uses lookup-table based vector quantization for texture description. A set of low and mid-level perceptually inspired image features are proposed by Sharan et al. [35] for texture classification.

Combining multiple texture representations for robust classification [9,24,38,11] is an interesting problem. The work of Tan and Triggs [38] combines Gabor wavelets and LBP for the problem of face recognition. Ylioinas et al. [46] combine contrast information together with local binary patterns for improved gender classification. A combination of HOG, LBP and Gabor features is used by Li et al. [23] for gender classification. To counter the dimensionality of the proposed image representation, Partial Least Squares (PLS) is used to learn a low-dimensional representation. Hong et al. [11] propose a numerical variant of LBP which is efficient and rotation invariant. The method is combined with other cues by a covariance matrix. Guo et al. [9] propose a learning framework to fuse a variety of LBP variants such as conventional LBP, rotation invariant patterns, local patterns with anisotropic structure, completed local binary patterns and local ternary patterns. Similar to Guo et al. [9], we investigate the problem of combining multiple texture description approaches. However, instead of only combining LBP variants [9], we here investigate fusing multiple texture descriptors to obtain a single heterogeneous texture representation.

A variety of color description approaches have been proposed in the field of object and scene recognition [8,3,43,32,18,19]. Bosch et al. [3] propose to compute SIFT descriptors directly on HSV channels for image classification. A comprehensive evaluation of color descriptors is performed by Sande et al. [32]. It has been shown that using an explicit color descriptor significantly improves the performance for object recognition [18,19], object detection [16], texture recognition [17] and action recognition [15]. In this work, we perform a comprehensive evaluation of pure color descriptors, popular in object recognition, for the problem of texture classification.

3. Combining multiple texture descriptors

Here we present our framework of combining multiple texture features and obtaining a compact heterogeneous texture representation. We combine five texture descriptors namely, completed local binary patterns [10], WLD descriptor [5], binary Gabor pattern [51], local phase quantization descriptor [31] and binarized statistical features [14]. We start by providing a brief overview of the five texture descriptors used in this work.

Completed local binary patterns [10]: The completed local binary patterns (CLBP) extend the conventional LBP operator by incorporating local difference sign-magnitude transform information (LDSMT). The LDSMT further consists of two components, namely the difference sign and difference magnitude encoded by a binary code. Likewise the conventional LBP, a region is also represented by its center pixel encoded by a binary code after global thresholding. The final image representation is obtained by concatenating the three binary code maps to form a single histogram.

WLD descriptor [5]: The WLD descriptor is inspired by Weber’s Law and encodes both differential excitation and orientations at locations. The first component, differential excitation, captures the ratio between the intensity difference of a pixel with its neighbors and the intensity of the current pixel. The second component captures the gradient orientation of the current pixel.

Binary Gabor patterns [51]: The binary Gabor patterns (BGP) is a rotation invariant texture descriptor. Unlike MR8 filters [41], BGP uses pre-defined rotation invariant binary patterns and does not require a pre-training phase to learn a lexicon dictionary. Unlike LBP, where each sign is binary coded from the difference of two
single pixels, BGP adopts the difference of regions to counter the noise sensitivity problem.

**Local phase quantization** [31]: The local phase quantization (LPQ) descriptor works by quantizing the phase information of the Fourier transform and is robust to image blur. To counter the problem of heavy image blur, the approach uses short-term Fourier transform with a uniform function. A data correlation scheme is also incorporated into the descriptor which plays a crucial role in case of a sharp image. The LPQ descriptor is shown to provide excellent results for both texture and face recognition tasks.

**Binarized statistical descriptor** [14]: The binarized statistical image feature (BSIF) represents each pixel by a binary code. These binary codes are constructed by learning a set of basis vectors from natural images using independent component analysis and an efficient scalar quantization scheme. The number of basis vectors determines the length of the pixel binary codes used to construct the final histogram of an image.

In our approach, each image is represented by the five aforementioned texture description methods. The final representation is obtained by concatenating all five texture representations into a single histogram, \( H = [h_{11}, h_{12}, h_{33}, h_{14}, h_{55}] \). This multi-texture histogram is then input to the classifier for texture classification.

### 3.1. Compact multi-texture representation

The multi-texture representation has the disadvantage of being high-dimensional (more than 3k of size) for an image. This is problematic as it significantly increases the computational time and memory usage in the classification stage. Recently, Elfiky et al. [7] proposed a compression approach using the DITC algorithm [6] to counter the high-dimensionality issue of the bag-of-words based spatial pyramid representation. In this work, we also use the same underlying approach to compress the high-dimensional multi-texture representation. However, the difference with the work of Elfiky et al. [7], is that here we investigate the DITC algorithm to solve the problem of compressing multi-texture histogram to obtain a single heterogeneous texture representation.

The DITC algorithm has been shown to obtain excellent results in reducing large histograms to compact ones. The algorithm is designed to find a fixed number of clusters that minimize the loss in mutual information between clusters and the category labels of training images. The DITC algorithm works on the class-conditional distributions over the texture histograms. The class-conditional estimation is measured by the probability distributions \( p(R|h) \), where \( R = \{r_1, r_2, \ldots, r_0\} \) is the set of \( O \) classes. The DITC algorithm works by estimating the drop in mutual information \( I \) between the histogram \( H \) and the class labels \( R \). The transformation from the original texture histogram \( H \) to the new representation \( H^R = \{H_1, H_2, \ldots, H_I\} \) (where every \( H_i \) represents a group of words in the original uncompressed histogram) is equal to

\[
\Delta I = I(R; H) - I(R; H^R) = \sum_{j=1}^J \sum_{h \in H_j} p(h) \text{KL}(p(R|h), p(R|H_j)),
\]

where KL is the Kullback–Leibler divergence between the two distributions defined by

\[
\text{KL}(p_1, p_2) = \sum_{z \in \mathbb{Z}} p_1(z)\log\frac{p_1(z)}{p_2(z)}.
\]

The multi-texture histogram bins with similar discriminative power are merged together over the classes. For more details, we refer to Dhillon et al. [6] on the DITC algorithm.

### 4. Combining color and texture

There exist two main strategies namely, early and late fusion, to combine color and texture information [29,17]. Early fusion works by computing texture descriptor on the color channels. In this way, a joint color–texture representation is obtained that combines the two cues at the pixel-level. Early fusion based image representation has the advantage of being more discriminative since the two cues are combined at the pixel level. However, early fusion representations suffers from the problem of high dimensionality.

Contrary to early fusion, late fusion combines the two cues at the image level. A separate histogram is constructed for color and texture. The two visual cues are then combined by concatenating the separate histograms into a single representation. The late fusion approach has shown to provide superior results for texture recognition [29,17], object recognition [18], object detection [16] and action recognition [15]. Therefore, in this work, we use late fusion scheme for combining color and texture information. Next, we provide an overview of pure color descriptors.

#### 4.1. Pure color descriptors

Here, we provide a brief overview of the pure color descriptors, popular in object recognition, for the problem of texture description. **RGB histogram** [32]: We use the standard RGB descriptor as a baseline. The RGB histogram is constructed by combining the three histograms from the R, G and B channels. The descriptor has 45 dimensions.

**rg histogram** [32]: The rg histogram is based on the normalized RGB color model. The descriptor is 45 dimensional. It is invariant to light intensity changes and shadows.

**Opponent-angle histogram** [43]: Unlike other pure color descriptors based on the (transformed) RGB values of the image, the opponent-angle histogram is constructed based on image derivatives. The histogram has 36 dimensions.

**HUE histogram** [43]: The HUE descriptor was proposed by Weijer and Schmid [43] and consists of 36 dimensions. In this descriptor, the hue is weighted by the saturation of a pixel in order to counter the instabilities in hue.

**Transformed color distribution** [32]: The transformed color descriptor is derived by normalizing each channel of RGB histogram. The descriptor has 45 dimensions. It is invariant with respect to scale and light intensity.

**Color moments and invariants** [32]: In the work of van de Sande et al. [32], the color moment histogram is constructed by using all generalized color moments up to the second degree and the first order. The color moment invariants are constructed using generalized color moments. The color moments histogram has 36 dimensions whereas the color moment invariants has 24 dimensions.

**Hue-saturation descriptor**: The hue-saturation descriptor is invariant to luminance variations. The histogram has 36 dimensions (nine bins for hue times four for saturation).

**Color names** [44]: Most of the color descriptors discussed above are designed to achieve photometric invariance. Instead, color names descriptor aims at providing a certain degree of photometric invariance with discriminative power. The color names are used in daily life by humans to communicate color, such as “black”, “blue” and “orange”. Here, we use the 11 dimensional color names mapping learned from the Google images by van de Weijer et al. [44].

**Discriminative color descriptors** [19]: The discriminative color descriptors by Khan et al. [19] take an information theoretic approach to the problem of color description. The method works by clustering color values together based on their discriminative power with an objective function to minimize the drop of mutual
information of the final color representation. In this work, we use the three universal color representations with 11, 25 and 50 dimensions, respectively.

5. Experimental results

To validate the performance of the proposed framework, we use four challenging datasets, namely KTH-TIPS-2a, KTH-TIPS-2b, FMD and Texture-10. The KTH-TIPS-2a dataset consists of 11 texture categories with images at 9 different scales, 3 poses and 4 different illumination conditions. We use the standard protocol [4,36,5] by reporting the average classification performance over the 4 test runs. In each time, all the images from 1 sample are used for test while the images from the remaining 3 samples are used as a training set. The KTH-TIPS-2b dataset also consists of 11 texture categories. Here, for each test run, all images from 1 sample are used for training while all the images from remaining 3 samples are used for testing. The FMD dataset consists of 10 texture categories with 100 images [37,35] for each class where 50 images are used for training and 50 for testing. The Texture-10 dataset consists of 10 different texture categories [17] where 25 images per class are used for training and 15 for testing. Fig. 1 shows example images from the four texture datasets.

Throughout our experiments, we use one-versus-all SVM using the $\chi^2$ kernel [49]. Each test instance is assigned the category label of the classifier giving the highest response. The final classification score is obtained by calculating the mean recognition rate per category.

5.1. Experiment 1: combining texture features

We start by providing results for multi-texture representations. The results are presented in Table 1. For the CLBP descriptor, we use multiple radius values since it was shown to improve the performance compared to using a single radius value. On the KTH-TIPS-2a and Texture-10 datasets, CLBP provides the best performance compared to other single texture features. Among the five texture descriptors, the best results are achieved when using the BGP descriptor on the KTH-TIPS-2b dataset and WLD descriptor on the FMD dataset. In case of FMD and KTH-TIPS-2b datasets, the BSIF descriptor alone provides inferior results compared to other four texture descriptors. However, the performance still improves by 2.1% and 1.3% respectively on these datasets by adding the BSIF descriptor.

Combining the five texture representations in a single representation significantly improves the performance on all datasets. On the KTH-TIPS-2a dataset, a significant gain of 4.0% is obtained by combining multiple features compared to the single best representation. Similarly, gains of 5.6%, 7.8% and 2.4% are obtained by combining multiple texture features on the KTH-TIPS-2b, FMD and Texture-10, respectively. The results clearly suggest that different texture representations possess complementary information and should be combined to obtain a significant performance boost.

5.2. Experiment 2: compact multi-texture features

As discussed above, combining multi-texture representations improve the overall performance. However, this performance improvement comes at the price of high dimensionality. Here, we present the results obtained, using the approach described in Section 3.1, to compress the high dimensional multi-texture representation. We fix the final dimension of our multi-texture representation to 500.

Table 2 shows the results obtained on the four texture datasets. The DITC compression method reduces the dimensions from 3184 to 500 without any significant loss in accuracy. Surprisingly, on the KTH-TIPS-2a and KTH-TIPS-2b datasets, the low-dimensional compact representation improves the performance compared to the original representation. This demonstrates that the DITC method removes the redundancy while increasing the discriminative power in certain cases such as KTH-TIPS-2a and KTH-TIPS-2b datasets.

We also compared our texture compression approach with the discriminative texture feature selection method [9] on Texture-10 dataset. The method [9] learns a selection of LBP patterns based on robustness, discriminative power and representation of the features. We use the same feature representation (CLBP), having rotation invariance and a pixel neighborhood of 16, for both compression methods. The original representation is reduced to 500 using the two compression methods. The original feature representation with 8k dimensions provides a recognition rate of 71.3%. The feature selection method [9] obtains a classification rate of 70.0%. Our DITC based compression method improved the performance by providing an accuracy of 72.6%.

Additionally, we also compare the DITC compression method with conventional approaches for very low-dimensional representations. We compare with standard compression methods namely, PCA, PLS and Diffusion maps. Fig. 2 shows results obtained using different compression techniques on the FMD and Texture-10 datasets. The three compression methods, PCA, PLS and Diffusion maps provide inferior performance on both datasets. The DITC based compression method significantly outperforms other compression methods even for very compact texture representations.

5.3. Experiment 3: pure color descriptors

Here, we provide results of our comprehensive evaluation of color descriptors for texture recognition. Table 3 shows the results obtained using different color description methods on the four texture datasets. On the KTH-TIPS-2a and KTH-TIPS-2b datasets, RGB
The results are obtained using discriminative color descriptors with 50 dimensions. Similarly, on the FMD and Texture-10 datasets, the discriminative color names with 50 dimensions provide the best performance.

The results clearly demonstrate the effectiveness of using a discriminative color description approach that aims at maximizing the discriminative power while maintaining a certain degree of photometric invariance. Therefore, we select the discriminative color descriptors with 50 dimensions as an explicit color representation. In our final experiment, we combine the discriminative color descriptors with 50 dimensions as an explicit color representation. In the next section, we will discuss the results obtained using the DITC based compression technique.

5.4. Comparison with state-of-the-art

Table 4 shows a comparison with state-of-the-art approaches on four texture datasets. On the KTH-TIPS-2a dataset, the method of Sharma et al. [36] based on local-high-order statistics provides a classification accuracy of 73.0%. The approach by Lee et al. [21] based on local color vector binary patterns achieves a recognition rate of 61.7%. Our approach, while being compact, outperforms the state-of-the-art methods with a significant gain of 7.8% over

### Table 1
Classification accuracy (%) of different texture representations on four texture datasets. In all cases, combining multi-texture representations significantly improves the performance compared to the single best texture method.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dimension</th>
<th>KTH-TIPS-2a</th>
<th>KTH-TIPS-2b</th>
<th>FMD</th>
<th>Texture-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLBP [10]</td>
<td>1944</td>
<td>76.1 ± 5.6</td>
<td>61.5 ± 2.3</td>
<td>43.6</td>
<td>76.9</td>
</tr>
<tr>
<td>WLD [5]</td>
<td>512</td>
<td>68.5 ± 5.1</td>
<td>56.0 ± 2.8</td>
<td>43.8</td>
<td>74.7</td>
</tr>
<tr>
<td>BGP [51]</td>
<td>216</td>
<td>76.8 ± 4.9</td>
<td>63.3 ± 3.4</td>
<td>43.2</td>
<td>66.0</td>
</tr>
<tr>
<td>LPQ [31]</td>
<td>256</td>
<td>67.7 ± 5.6</td>
<td>54.4 ± 2.7</td>
<td>41.0</td>
<td>75.3</td>
</tr>
<tr>
<td>BSIF [14]</td>
<td>256</td>
<td>70.0 ± 5.7</td>
<td>54.3 ± 2.8</td>
<td>34.4</td>
<td>66.0</td>
</tr>
<tr>
<td>CLBP + WLD</td>
<td>2456</td>
<td>78.1 ± 4.8</td>
<td>63.7 ± 2.8</td>
<td>46.6</td>
<td>77.8</td>
</tr>
<tr>
<td>CLBP + WLD + BGP</td>
<td>2672</td>
<td>79.2 ± 5.1</td>
<td>65.1 ± 2.3</td>
<td>48.1</td>
<td>78.6</td>
</tr>
<tr>
<td>CLBP + WLD + BGP + LPQ</td>
<td>2928</td>
<td>79.9 ± 4.9</td>
<td>67.6 ± 2.6</td>
<td>49.5</td>
<td>78.9</td>
</tr>
<tr>
<td>CLBP + WLD + BGP + LPQ + BSIF</td>
<td>3184</td>
<td>80.8 ± 5.3</td>
<td>68.9 ± 2.9</td>
<td>51.6</td>
<td>79.3</td>
</tr>
</tbody>
</table>

The bold entries in the table correspond to the highest performance (number) compared to other methods.

### Table 2
Classification accuracy (%) obtained when using the original high-dimensional texture and compact texture representations. Note that the compression method reduces the dimensionality with little or no loss in accuracy.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dimension</th>
<th>KTH-TIPS-2a</th>
<th>KTH-TIPS-2b</th>
<th>FMD</th>
<th>Texture-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original texture feature</td>
<td>3184</td>
<td>80.8 ± 5.3</td>
<td>68.9 ± 1.7</td>
<td>51.6</td>
<td>79.3</td>
</tr>
<tr>
<td>Compact texture (DITC)</td>
<td>500</td>
<td>82.2 ± 5.4</td>
<td>69.0 ± 1.6</td>
<td>49.0</td>
<td>78.0</td>
</tr>
</tbody>
</table>

The bold entries in the table correspond to the highest performance (number) compared to other methods.

### Table 3
Comparison (%) of pure color descriptors on four texture datasets. Note that the best performance is obtained by using discriminative color names with 50 dimensions.

<table>
<thead>
<tr>
<th>Method</th>
<th>Dimension</th>
<th>KTH-TIPS-2a</th>
<th>KTH-TIPS-2b</th>
<th>FMD</th>
<th>Texture-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>50</td>
<td>55.5 ± 5.8</td>
<td>42.1 ± 1.8</td>
<td>20.3</td>
<td>52.3</td>
</tr>
<tr>
<td>HUE</td>
<td>30</td>
<td>54.3 ± 6.2</td>
<td>43.3 ± 2.3</td>
<td>22.2</td>
<td>52.7</td>
</tr>
<tr>
<td>Opp-angle</td>
<td>36</td>
<td>53.3 ± 6.1</td>
<td>43.1 ± 2.1</td>
<td>21.6</td>
<td>50.7</td>
</tr>
<tr>
<td>Transformed color</td>
<td>45</td>
<td>52.8 ± 5.3</td>
<td>44.8 ± 1.8</td>
<td>23.0</td>
<td>40.0</td>
</tr>
<tr>
<td>Color moments</td>
<td>30</td>
<td>54.9 ± 5.7</td>
<td>45.1 ± 3.6</td>
<td>26.0</td>
<td>50.1</td>
</tr>
<tr>
<td>Color moments inv</td>
<td>24</td>
<td>50.1 ± 5.5</td>
<td>41.0 ± 2.4</td>
<td>10.0</td>
<td>44.6</td>
</tr>
<tr>
<td>HS</td>
<td>36</td>
<td>53.6 ± 5.2</td>
<td>42.9 ± 2.9</td>
<td>26.0</td>
<td>44.6</td>
</tr>
<tr>
<td>Color names</td>
<td>11</td>
<td>56.8 ± 5.8</td>
<td>44.2 ± 1.7</td>
<td>25.6</td>
<td>56.0</td>
</tr>
<tr>
<td>Discriminative color descriptors</td>
<td>11</td>
<td>55.7 ± 5.6</td>
<td>43.9 ± 2.1</td>
<td>22.0</td>
<td>50.7</td>
</tr>
<tr>
<td>Discriminative color descriptors</td>
<td>25</td>
<td>57.4 ± 5.8</td>
<td>46.4 ± 2.2</td>
<td>25.6</td>
<td>54.0</td>
</tr>
<tr>
<td>Discriminative color descriptors</td>
<td>50</td>
<td>60.1 ± 5.7</td>
<td>48.1 ± 1.9</td>
<td>27.4</td>
<td>58.0</td>
</tr>
</tbody>
</table>

The bold entries in the table correspond to the highest performance (number) compared to other methods.
the best reported result. On the KTH-TIPS-2b dataset, the extended LBP approach [28] provides an accuracy of 58.1%. A combination of LBP and Fourier features achieves an accuracy of 54.6%. Our approach outperforms existing methods on this dataset by providing a recognition accuracy of 70.6%.

On the FMD dataset, a training-free approach by Timofte and Gool [40] obtains a recognition accuracy of 55.7%. Our approach, despite its simplicity, achieves an accuracy of 54.2%. The best results on this dataset are obtained using perceptually inspired features [35]. It is worthy to mention that our approach neither uses any ground-truth masks nor any perceptually inspired features. Such features are complementary to the approach presented in this paper and can be combined to obtain further boost in performance. Finally, on the Texture-10 dataset, our approach outperforms the color names and LBP fusion methods [17] by achieving a recognition accuracy of 82.0%.

6. Conclusion

In this paper we investigated the problem of texture recognition in images. Firstly, we have shown that fusing different texture representations significantly improves the performance compared to the single best method. To counter the high-dimensionality problem of the image representation, we proposed to use the DITC approach. Additionally, we performed a comprehensive evaluation of pure color descriptors, popular in image classification, for the task of texture recognition.

The results show that our compact texture representation with a dimensionality of only 500 significantly improved the performance over existing texture classification methods. Among the color descriptors, the discriminative color descriptors provide the best results. Finally, we fused the discriminative color descriptors with our compact texture representation and showed that it can achieve state-of-the-art performance.

In this work, we used a simple late fusion technique to combine the color and texture features. Future work includes investigating sophisticated fusion approaches to combine the color and texture descriptions. A further comparison of the DITC approach with other compression approaches [13,34] can provide a further insight on its applicability to other computer vision applications.

Acknowledgements

This work has been supported by SSF through a grant for the project CUAS, by VR through a grant for the project ETT, through the Strategic Area for ICT research ELLIIT, CADICS and The Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, 251170). We also acknowledge the grants 255745 and 251170 of the Academy of Finland, SSP-14183 of EIT ICT Labs, and the D2I SHOK project. The project TIN2013-41751 of Spanish Ministry of Science. The calculations were performed using computer resources within the Aalto University School of Science “Science-IT” project.

References


