LBP-and-ScatNet-based Combined Features For Efficient Texture Classification

Vu-Lam Nguyen, Ngoc-Son Vu, Hai-Hong Phan, Philippe-Henri Gosselin

Noname manuscript No.
(will be inserted by the editor)

LBP-and-ScatNet-based Combined Features For Efficient Texture Classification

Vu-Lam Nguyen
· **Ngoc-Son Vu**
· **Hai-Hong Phan**
· **Philippe-Henri Gosselin**

Received: date / Accepted: date

Abstract In this paper, we propose a micro-macro feature combination approach for texture classification. The two disparate yet complementary categories of features are combined. By this way, Local Binary Pattern (LBP) plays the role of micro-structure feature extractor while the scattering transform captures macro-structure information. In fact, for extracting the macro type of features, coefficients are aggregated from three different layers of the scattering network. It is a handcrafted convolution network which is implemented by computing consecutively wavelet transforms and modulus non-linear operators. By contrast, in order to extract micro-structure features which are rotation-invariant, relatively robust to noise and illumination change, the completed LBP is utilized alongside the biologically-inspired filtering (BF) preprocessing technique. Overall, since the proposed framework can exploit the advantages of both feature types, its texture representation is not only invariant to rotation, scaling, illumination change but also highly discriminative. Intensive experiments conducted on many texture benchmarks such as CUReT, UIUC, KTH-TIPS2b, and OUTEX show that our framework has a competitive classification accuracy.

Keywords Image texture · Image classification · Image texture analysis · Wavelet Transforms · Scattering Transforms

V.L. Nguyen E-mail: lam.nguyen@ensea.fr
· N.S. Vu E-mail: son.vu@ensea.fr
· H.H. Phan E-mail: thi-hai-hong.phan@ensea.fr
· P.H. Gosselin E-mail: philippe-henri.gosselin@ensea.fr
ETIS - ENSEA/ Université de Cergy-Pontoise/ CNRS UMR 8051
95000-Cergy, France
Tel.:+33 01 30 73 66 10
Fax: +33 01 30 73 66 27

1 Introduction

Texture provides important clues for identifying materials and objects, especially when shape is not available. A wide range of applications such as industrial inspection, image retrieval, medical imaging, remote sensing, object and facial recognition can be developed depend upon analyzing the textures of their surfaces. Hence, texture analysis includes segmentation, shape extraction, synthesis, and classification is an active field.

There are variety texture analysis approaches have been proposed. They can be ranged from a simple to sophisticated computation strategy methods. Simple yet efficient feature extraction approaches can be a long list. They can be 1) Local Binary Pattern (LBP) method [1] and its variants, 2) the representation based on co-occurrence matrix [2],(3) the filter-based approaches such as works in [3,4], 4) the wavelet transform method as works in [5–7], 5) the texton dictionary-based [8–10], 6) the use of bidirectional features [11, 12], and so on. In addition, many sophisticated computation strategy methods have been introduced to improve the feature robustness and performance. Scattering Network (ScatNet) [13], and Convolutional neural network (CNNs) with Fisher vector CNN (FV+CNN) delegation [14] belong to this category.

Among those approaches, the LBP-family can be considered as a popular feature method which extracts well local micro-structure information from images. Ojala *et al.* [15] first introduced the LBP method in 1996, then a multi-resolution version [1] in 2002. After that, several extensions on LBP have been conducted. In 2007 Tan *et al.* extended LBP to 3-valued codes to become the local ternary pattern (LTP) [16]. Liao *et al.* proposed dominant LBP (DLBP) [17] which combines the most frequently occurred patterns with the Gabor filter responses for features. Later Guo *et al.* introduced completed LBP (CLBP) [18], which merges three components the sign (CLBP_S), magnitude (CLBP_M), and center pixel intensity (CLBP_C) together to form features. This enhances discriminative power compared to the original version. Variance in LBP (LBPV) [19] is used to encode local contrast information without requiring a quantization process, rotation invariance is implemented by estimating principal orientations and aligning LBP histograms. By constructing a cascading spatial pyramid of LBP, Qian *et al.* [20] introduced pyramid transformed LBP (PLBP), the robustness of PLBP was compared with those of other LBP variants in this works. Further, Liu *et al.* suggested extended LBP [21] by a combination of pixel intensities and local differences. In this way, the pixel intensity part is divided into a central pixel's component and neighbor's component. Likewise, the local difference consists of two components: radial differences and an angular difference. At the end, those four were combined to form features. In addition, Zhao *et al.* in [22] presented Local Binary Pattern Histogram Fourier features (LBP-HF) which implements rotation invariance by computing discrete Fourier transforms of local binary pattern (LBP) histograms. In [23], moreover, Guo *et al.* presented a three-layered learning framework in which LTP and CLBP were used as raw features to train and select the discriminative features.

Contrary to the micro-structure descriptors of LBP family, several broader range feature methods have been developed. Bovik *et al.* applied the Gabor filters to compute the average filter responses for features [3]. Mallat proposed the multi-resolution wavelet decomposition method [24], which generates coefficients from the high-low (HL), low-high (LH), and low-low (LL) channels for subsequent classification tasks. Porter *et al.* [25] removed the high-high (HH) wavelet channels and combined the LH and HL wavelet channels to obtain rotation invariance wavelet features. Haley *et al.* [26] calculated isotropic rotation invariance features from Gabor filter responses. More recent, scattering transform is considered as a high performance approach based on cascading wavelet transform layers [13] compared to previous wavelet-based methods. With the aim of leveraging both fine details and broader-range multi-path of signals, we propose a novel framework which combines LBP features and those of scattering transform for texture classification. A preprocessing algorithm, the biologically-inspired filtering(BF) [27], as well as an efficient PCA classifier are also used. Our hand-crafted descriptor has the accuracy which is close to those of the CNN state-of-the-art (FV-CNN [14]) on the three out of four well-known texture datasets (UIUC[28], CUReT[29], OUTEX[30], KTH-TIPS2b[31]). FV-CNN is the Fisher Vector pooling of a Convolutional Neural Network (CNN) filter bank.

This paper is the extended version of two previous publications [32,33]. Compared to these two papers, this new version adds a number of new experiments and a substantial amount of new discussion. The rest of paper is organized as follows. After discussing about related work in Section 2, the proposed approach is introduced in Section 3. Section 4 presents experimental results, and conclusions are drawn in Section 5.

2 Related Work

In this section, Biologically Inspired Filtering (BF) preprocessing technique, which is used to enhance LBP-feature discriminative power, is presented alongside Local Binary Pattern (LBP), Completed LBP (CLBP) review. Besides, scattering transform, the global features used to compensate for those of local LBP, is briefly reviewed. At the end of this section, PCA classifier used in the proposed framework is presented.

2.1 Biologically Inspired Filtering

Biologically Inspired Filtering (BF) [27] imitates the human retina mechanism to extract more detail information of a given image when being used as a preprocessing step. It enhances performance of different features in terms of discriminative power for texture classification, including CLBP.

In general, BF consists of two steps:

Step 1: given an image I_{in}, it is first filtered by a band-pass Difference of Gaussians (DoG). $I_{bf} = DoG \star I_{in}$, with $DoG = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$. $\star$ is the convolution operator, σ_1, and σ_2 are the standard deviations of the low pass filters.

Step 2: the filter responses are then decomposed into two "maps" corresponding to the image details alongside two sides of the image edge:

$$I_{bf}^{+} = \begin{cases} I_{bf}(p) & if \quad I_{bf}(p) \geq \epsilon \\ 0 & otherwise \end{cases}, \qquad I_{bf}^{-} = \begin{cases} |I_{bf}(p)| & if \quad I_{bf}(p) \leq -\epsilon \\ 0 & otherwise \end{cases}. \quad (1)$$

The term bf refers to "Biologically-inspired Filtering", p is the considered pixel, ϵ is a threshold. In features extraction step, instead of using the input image I_{in}, features are first extracted from two images, I_{bf}^{-} and I_{bf}^{+}, and are then concatenated.

2.2 Brief Overview of the LBP and CLBP

The LBP method, proposed by Ojala *et al.* [1], encodes the pixel-wise information in images. LBP encoding is:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \quad s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (2)$$

where g_c represents the gray value of the center pixel whereas g_p ($p = 0, ..., P-1$) denotes the gray value of the neighbor pixel on a circle of radius R, and P is the total number of the neighbors. A given texture image is then represented by a histogram of LBP codes. Ojala *et al.* also introduced a rotation invariant complement called uniform patterns LBP^{riu2} which have less than two "one-to-zero or vice versa" transitions.

Guo *et al.* [18] recently proposed CLBP descriptors, by which image local differences are decomposed into two complementary components, the signs (s_p) and the magnitudes (m_p): $s_p = s(g_p - g_c)$, $m_p = |g_p - g_c|$ where g_p, g_c and $s(x)$ are defined as in (1). Two operators called CLBP-Sign ($CLBP\_S$) and CLBP-Magnitude ($CLBP\_M$), respectively, are proposed to encode them, where the $CLBP\_S$ is equivalent to the conventional LBP, and the $CLBP\_M$ measures the local variance of magnitude. The $CLBP\_M$ is defined as follows:

$$CLBP\_M_{P,R} = \sum_{p=1}^{P-1} t(m_p, c)2^p, \; t(x, c) = \begin{cases} 1, x \geq c \\ 0, x < c \end{cases} \quad (3)$$

where threshold c is the mean value of m_p of the whole image. CLBP-Center ($CLBP\_C$) operator extracts the local central information as $CLBP\_C_{P,R} = t(g_c, c_I)$ where threshold c_I is set as the average gray level of the whole image. The overall descriptor is obtained by combining the three operators $CLBP\_S$, $CLBP\_M$ and $CLBP\_C$.

2.3 Review of Scattering Transform

The Scattering transform [34] is calculated by using Scattering Network (ScatNet). It is actually a handcrafted deep convolution network, in which cascade of wavelet transform and modulus non-linearities operators are consecutively computed to form the network layers. As illustrated in (Fig. 1), each $|W_m|$ outputs invariant scattering coefficients $S_m x$ and a next layer of covariant wavelet modulus coefficients $U_{m+1}x$, which is further transformed by the subsequent wavelet-modulus operators.

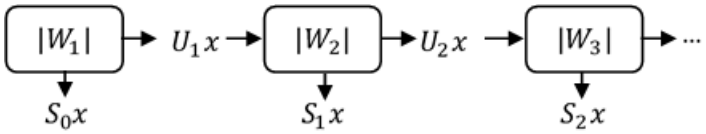

Fig. 1 Scattering network formed by wavelet-modulus cascading

The average $S_m x$ carries the low frequencies of $U_m x$ while it loses all the high frequencies. High frequencies are captured by roto-translation convolutions with wavelets. $|W_m|$ transforms $U_m x$ into the average $S_m x$ and a new layer $U_{m+1}x$ of wavelet amplitude coefficients: $|W_m|(U_m x) = (S_m x, U_{m+1}x)$. Repetitively computing this wavelet modulus transform would generate multiple layers of scattering invariant coefficients. For $m = 0$ $U_0 x = x$, in case the network has three layers, the scattering feature vector (Sx) would be a concatenation of three $S_i x$ coefficients such that $Sx = (S_0 x, S_1 x, S_2 x)$. A filter bank of low-pass and high-pass filters for implementing Morlet wavelet (W_m operator) is illustrated in Fig. 2.

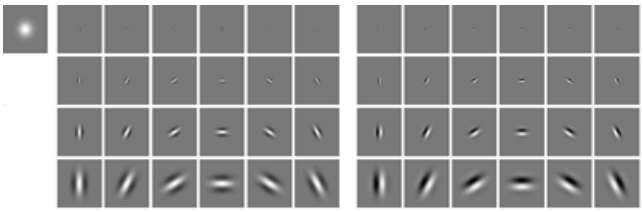

Fig. 2 Complex Morlet wavelets with Gaussian kernel (top left corner), different scales (along rows) and orientations (along columns). The real and imaginary parts are shown on the left and on the right, respectively..

2.4 PCA Classifier

A generative classifier called Principal Component Analysis (PCA) [13] was proved to have decent performance for ScatNet in case of small training dataset. PCA Classifier is described as follows.

Given a test image X, $\widetilde{S}X$ denotes the scattering transform of X and its dilated version $\mathfrak{D}_{\mathrm{j}}X$.

$$\widetilde{S}X = \left(\sum_{0\leq j<H} 1\right)^{-1} \sum_{0\leq j<H} \widetilde{S}\mathfrak{D}_{\mathrm{j}}X. \tag{4}$$

The representation of $\widetilde{S}X$ used at test time is therefore a Scattering Transform.

Let $P_{U_c}\widetilde{S}X$ denotes the orthogonal projection of $\widetilde{S}X$ in the scattering space U_c of a given class c. The principal components space U_c is approximately computed from the singular value decomposition (SVD) of the matrix of centered training sample $\widetilde{S}\mathfrak{D}_{\mathrm{j}}X_{c,i} - \mu_c$ with all possible samples i dilated by 2^j for a given class c. The PCA classification computes the class $\hat{c}(X)$ base on the minimum distance $\left\|(Id - P_{V_c})(\widetilde{S}X - \mu_c)\right\|$ from $\widetilde{S}X$ to the space $\mu_c + U_c$, (Fig. 3)

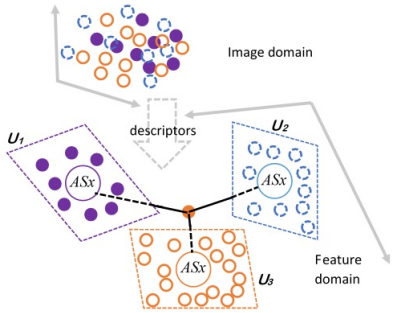

Fig. 3 PCA-classifier classifies a test image X based on the minimum distance from Scattering Transform $\widetilde{S}X$ to subspace $\mu_c + U_c$.

3 Proposed Method

A motivation for our proposal is that LBP features and those of scattering transform provide complementary information. The CLBP is used to capture small and fine details information from texture by encoding the pixel-wise structure in a certain radius. It is usually of from 1 to 7 pixel-distances from a center. However, it does not take into account the wider range pixel relationship that takes place beyond the coverage of its radii. In this section, we propose a framework which complements the CLBP features with the broader

range, multi-scale, multi-path counterparts, the scattering transform of texture image. This section is organized as follows. First, we present how BF preprocessing technique used to enhance the robustness of CLBP features. Second, ScatNet implementation is presented in details. Third, we describe our proposed framework for extracting texture features. Finally, the framework general settings with a PCA classifier for texture classification are detailed.

3.1 Utilizing BF+CLBP

BF preprocessing technique[27] with following properties would improve performance of LBP features while preserve the computation efficiency:

- Robust to illumination: the band-pass DoG filter removes the low frequency illumination.
- Rotation insensitivity: the used filter is isotropic and discards all orientation information, and so is independent to rotation.
- Low computational time.

We therefore, use BF to improve performance of the micro features by calculating the CLBP code on the BF-maps (I_{bf}^{+} , I_{bf}^{-}, Fig. 4) instead of the input image.

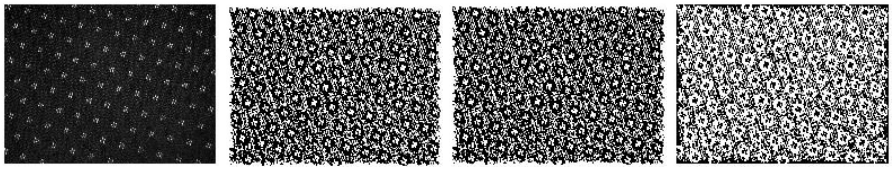

Fig. 4 BF processing chain: I_{in}, I_{bf} , I_{bf}^{+} , I_{bf}^{-} (left to right).

3.2 ScatNet Configuration

ScatNet performs scattering transform which captures a wide range of signals for features. The configuration proposed in [13] is used for our framework. That is a 3-layer handcrafted convolution network with the first layer implemented two operators consecutively, 1) a 2D-Morlet-wavelet transform followed by 2) a non-linear modulus operator. The latter generates signal magnitudes for the next layers. While the second and third layers are computed by a 3-D wavelet transform of signal magnitudes which have been calculated from the previous layer (Fig. 5). The 3-D wavelet transform is implemented by a spatial 2D-wavelet transform and a 1D-wavelet transform along the rotation angles. The wavelet transform utilized here is Morlet wavelet.

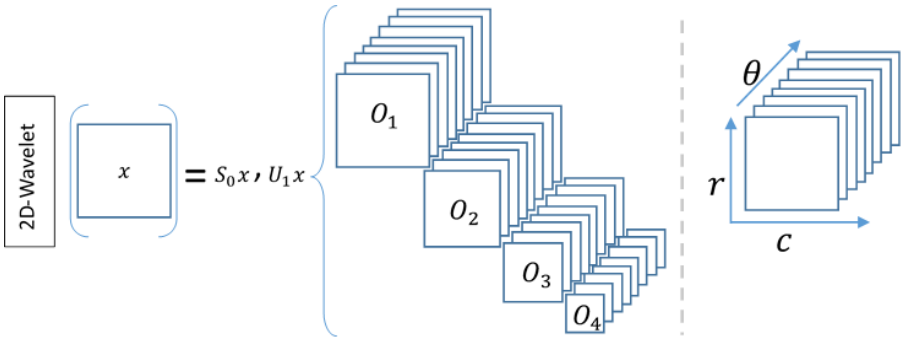

Fig. 5 A spatial 2D-wavelet on input image x outputs a coefficient S_0x, and U_1x is then grouped into orbits (signals with the same scale and different orientations, left).
3D-wavelet on each orbit (right) is computed by a spatial 2D-wavelet followed by an 1D-wavelet transform along the orientation variables θ.

3.3 Proposal Framework

The overall idea is illustrated in Fig. 6. We do not utilize the BF preprocessing technique for ScatNet because this network itself contains band pass filters whose functionality is similar to those of BF.

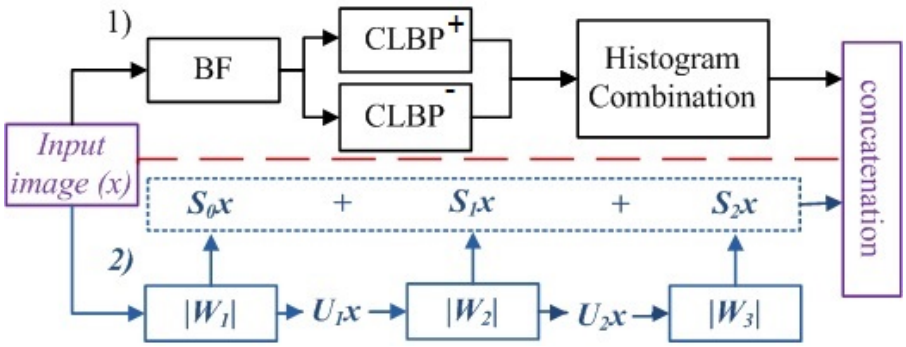

Fig. 6 Integrated Descriptor consists of two components separated by the dashed line:
1) BF takes one image as input, and generates two outputs I_{bf}^{+} and I_{bf}^{-} which are further used as CLBP's inputs, namely BF+CLBP component.
2) ScatNet Component: scattering representation is computed by a cascade of wavelet-modulus operators $|W_m|$. Every $|W_m|$ has one input and two outputs which are the scattering coefficients $S_m x$ and the next layer wavelet modulus coefficients $U_{m+1}x$. The latter is used for further transformation.
Final features are the concatenation of those generated by the components.

For a given input image x, we first apply BF+CLBP and the Scattering Transform on the image independently, then aggregate their outputs (features) to form final features. These features capture well both local and global structure information of texture images. Specifically, BF splits the input image x

into I_{bf}^{+} and I_{bf}^{-} (Fig. 4), BF+CLBP representation of image x is the combination of CLBP features of I_{bf}^{+} and I_{bf}^{-}, namely BF+CLBP features. Similarly, the ScatNet features are gained by an average value of the its coefficients (S_0x, S_1x, S_2x). These mean values reflect the signal strength of an image. An important drawback comes up. The mean values of ScatNet coefficients are incompatible with CLBP histograms, one of them conflicts the other due to their future extraction schemes. A measure for the aforementioned conflict is that both types of features are then normalized before being combined. Finally, an aggregation of those forms our integrated features.

We also discovered that the PCA classifier, which is used for ScatNet in [13], has a higher performance than the nearest neighbor classifier for our descriptor in term of classification accuracy.

It should be noted that we focus mainly upon building a descriptor based on the Scattering Transform rather than techniques of Scatnet [13] such as multi-scale average, and multi-scale training to augment classification results.

3.4 Framework Settings

The framework is centered around two types of descriptors. The first type are LBP descriptors extracted by using three-scale CLBP. Different schemes of radius-neighborhood pairs $(R_1, P_1) + (R_2, P_2) + (R_3, P_3)$ are chosen. For example, $(1, 8) + (2, 16) + (3, 24)$ *or* $(1, 16) + (2, 16) + (3, 16)$ *or* $(1, 8) + (3, 16) + (5, 24)$ *or* $(1, 16) + (3, 16) + (5, 16)$ *or* $(3, 16) + (5, 16) + (7, 16)$, the "+" denotes the concatenation histograms of the CLBP codes. CLBP features dimension change according to number of neighbors. P=8,16,24 the CLBP descriptors are sequentially the 200-dimensional,648-dimensional, and 1352-dimensional features vectors. The experiments use three various scales so the feature vectors would be those multiplied by three. When BF preprocessing technique is applied, the dimensions of these vectors will be double because BF_CLBP features of an image are the concatenation of features of two other images extracted by BF.

The second type of texture descriptors are ScatNet [13]. The features extracted from a hand-crafted 3-layer convolutional network. In the framework, 3 various configurations are used depend upon the size of images in datasets, those are the 3-scale, 4-scale, and 5-scale of 8-orientation filter bank, i.e. the smaller scale configuration is used for dataset with smaller image size. The feature vector dimensions of those three network configurations are 108, 208, and 340 respectively.

When both types are combined, the final descriptors capture both micro and macro structure information of data, the dimensionality of the final representation varies according to the schemes selected, ranging from 2,308 to 2,540 for CLBP+ScatNet, and from 4,508 to 4,740 for BF+CLBP+ScatNet.

The PCA classifier from ScatNet library [13] is utilized for classification. This is an affine space classifier which takes feature vectors as input to build the affine spaces. These spaces have dimensions (lower than those of input feature

vectors) as parameters which are chosen from 80 to 150 in the experiments. The dimensionality of input features is reduced using principal component analysis to build the spaces as mentioned. Next section will discuss about experimental validation and its analysis.

4 Experimental Validation

This section evaluates the proposed method for classifying texture data. First, parameter settings and datasets are presented. Second, we evaluate the results and compare with those of well-known state-of-the-art. Finally, we analysis the proposed framework and its complexity. In the experiments, we used source code of CLBP, BF, ScatNet, VLFeat library and MatConvNet shared by Guo *et al.*, [18], Vu *et al.* [27], Mallat's group [13], and Vedaldi *et al.* [35,36] respectively to generate texture classification results on 4 benchmarks: UIUC, CUReT, KTH_TIPS2b, and Outex.

4.1 Experimental Parameters and Dataset Settings

We analyze the effectiveness of our method by doing experiments on four popular texture databases, and their testing protocols are strictly followed. The PCA classifier [13] is used to produce texture classification results.

Parameters for the BF preprocessing technique are: $\sigma_1 = 1.25, \sigma_2 = 6, \epsilon = 0.15$ as recommended in [27].

Arguments of ScatNet for the Scattering Transform are selected such that scaling number is J=3 for datasets with resolution of images at 150×150 or below, J=4 for image size 300×300 or smaller, and J=5 otherwise. Here, the principle is that the smaller image size the smaller scaling number chosen as recommended in [13]. Orientations of filter bank is set to 8 (L=8), number of ScatNet Layers equal 3 (M=3). Since if the layer number exceeds this threshold, both feature dimension and feature extraction time increase with little improvement in classified accuracy.

Experiments were conducted on datasets, with samples of those represented in Fig. 7, and the summary is in table 1.

Table 1 Summary of Datasets for the experiments

Dataset	Images	Image size	Classes	Splits
CURet	5612	200×200	61	100
UIUC	1000	640×480	25	100
Outex TC10	4320	128×128	24	pre-defined
Outex TC12	9120	128×128	24	pre-defined
KTH-TIPS2b	4752	200×200	11	pre-defined

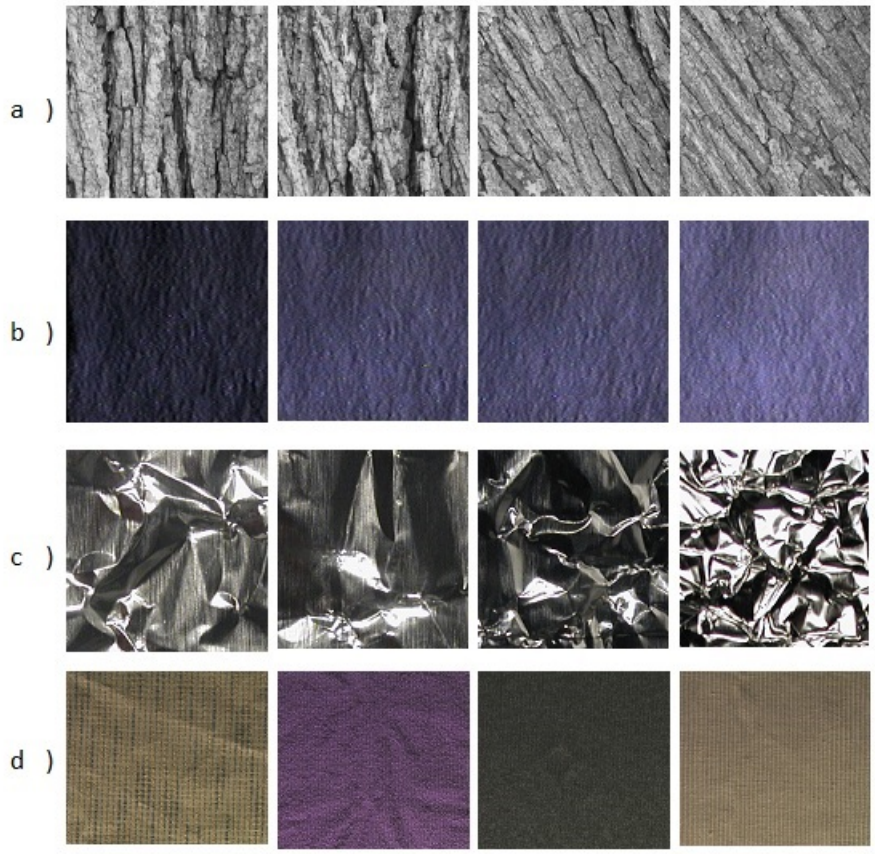

Fig. 7 Images in the same class from a) UIUC, b) CUReT, c) KTH_TIPS2b, d) Outex datasets.

UIUC [28] has 25 classes of texture, each class contains 40 different images which has resolution of 640×480 include changes of viewing angle, scale, and illumination conditions. The mean classification accuracy, 100 random splits between training and testing with a half of samples (20 images) per class chosen for training the rest for testing, is reported.

CUReT [29] database contains 61 texture classes, 205 images per class, acquired at different viewpoints, illumination, and orientations. There are 118 images shot from a viewing angle of less than 60 degrees. Large images are cropped to (200×200) across all texture classes. Cropped regions are then converted to grey scale. We follow the common training and testing scheme, i.e. a subset 46 images from 118 each class are chosen for training, the remains of unseen data are used for testing . Splits are implemented 100 times independently, the average accuracy over 100 randomly partitions are reported.

The material databases **KTHTIPS2b** [31], with 3 viewing angles, 4 illuminants, and 9 different scales, produce 432 images per class, with the image

size of 200×200 and 11 classes in total. Regarding the KTHTIPS2b databases, we follow the common testing and training protocols. Only unseen data is used for testing, with three out of four samples used for training and the remaining for testing.

OUTEX [30] database contains textural images which are captured from a wide variety of real material surfaces. We consider the two commonly used test suites, Outex_TC_00010 (TC10) and Outex_TC_00012 (TC12), containing 24 classes with up to 200 texture images per class. This database is built by taking images under three different illuminations ("horizon", "inca", and "t184") with resolution of images at 128×128. Standard protocols with predefined training and testing sets are exploited for this dataset.

4.2 Classification Results

Intensive experiments were conducted on four texture datasets, the results are compared with well-known and state-of-the-art of those, we chose the highest results reported by relevant articles for the comparison. We divide our experiments into two main parts. In experiment #1), a combination of CLBP and ScatNet is used while experiment #2) BF is added to enhance the robustness of CLBP. Our method reaches state-of-the-art on CUReT [29], is close to the recent frontier (FV-SIFT+ FC+FV-VD) reported in [14] on UIUC [28], KTH-TIPS2b [31], and is comparable to state-of-the-art on three testing suites of Outex [30]. Details are shown in Table 2,3,5,4. It is worth noticing that our novel framework has a better classification accuracy than the ones it inherits from on the experimented datasets while we do not use the multi-scale training technique of ScatNet [13].

Our method produces handcrafted features without expensive learning stage as the convolutional neural network based methods, FV-SIFT+ FC+FV-VD [14] does. Even though, ours gets a better classification results than those of the nineteen-layer network [14] on Outex [30].

Details about classified accuracy on experimented databases as follow:

Results on UIUC Database: This section discusses more detail the classification results obtained on the UIUC dataset. There are two experiments actually drawn on this dataset.

i) CLBP+ScatNet: We get around 2% classification enhancement on this dataset (from 95.90% to 97.50%) comparing to the its original version.

ii) BF+CLBP+ScatNet has got similar improvement in classification accuracy, increasing from 95.20% to 98.69%. When BF preprocessing techniques added to CLBP, the accurate rate rises roundly 1% while the feature vectors of the CLBP codes also rise double.

From above analyses, one can conclude that local CLBP/BF+CLBP features combined with broader range features (ScatNet) will improve the overall performance on UIUC dataset. This is because these two kinds of features are complementary in such a way that they can be robust to the changes of viewing angle, scale, and illumination conditions.

Results from Table 2 shows that our accurate rate on this dataset is similar to that of works in [37], roundly 1% lower than the recent frontier FV-SIFT+ FC+FV-VD [14] of this dataset while beating all others.

Table 2 Integrated Descriptor with BF+CLBP+ScatNet single scale: classification accuracy(%) and comparing with those of well-known methods on UIUC dataset

Method	Accuracy(%)
CLBP_S/M	95.90
CLBP_S/M/C	96.02
BF+CLBP_S/M	96.48
BF+CLBP_S/M/C	96.28
ScatNet	95.20
CLBP_S/M+ScatNet	97.20
CLBP_S/M/C+ScatNet	97.50
BF+CLBP_S/M+ScatNet	98.63
BF+CLBP_S/M/C+ScatNet	98.69
FV-SIFT+ FC+FV-VD[14]	99.90
Zhang *et al.*[37]	98.70
WMFS[37]	98.60
VZ-joint[10]	97.83
OTF[38]	97.40
MFS[39]	92.74
Hayman *et al.*[40]	92.00

Results on CUReT database: The experiments on CURet dataset get similarly interesting observations as those of UIUC. The combined versions have about 1% improvement in comparison with their original versions. As illustrated in table 3, the correct classification rate of our method closely reaches State-of-the-art on this dataset. Once again, the small fined structure features (CLBP) can compensate for scattering transform features for being robust to the combinations of viewing and illumination directions of texture in CUReT.

Results on OUTEX dataset: In case of Outex database, experiment conducted on 2 test suites, Outex_TC10 and Outex_TC12. It is consistent in the analyses of results obtained on the other databases, we figure out two interesting observations:

i)The texture classification accurate rate are at 99.87%, 98.43%, and 99.63% for Outex_TC10, Outex_TC12_00, and Outex_TC12_01 respectively, they are consistently higher than those of the original features.

ii) Our results are much better than those of FV-SIFT+ FC+FV-VD[14], which can be seen as the frontier on many datasets other than OUTEX, the data suite is mainly used to test the rotation invariance of features. One can observe that the two rotation invariance features, ScatNet and CLBP, are combined together in our novel framework, results in the features which are also rotation invariant yet more robust. Details can be seen in (Table 4).

Results on KTHTIPS2b: As can be seen from Table 5, our proposal descriptors get a significant improvement over their original versions, the en-

Table 3 Comparing proposal classification accuracy(%) with those of well-known methods on CUReT

Method	Accuracy(%)
CLBP_S/M	98.60
CLBP_S/M/C	98.56
BF+CLBP_S/M	98.65
BF+CLBP_S/M/C	98.57
ScatNet	99.10
CLBP_S/M+ScatNet	99.53
CLBP_S/M/C+ScatNet	99.50
BF+CLBP_S/M+ScatNet	99.74
BF+CLBP_S/M/C+ScatNet	99.51
FV-SIFT+ FC+FV-VD[14]	99.70
Broadhurst[41]	99.22
VZ-MR8[9]	98.61
Multi-scale BIF[42]	98.60
Hayman *et al.*[40]	98.46
VZ-joint[10]	97.66
Zhang *et al.*[37]	95.30
DLBP[17]	92.77

Table 4 Comparing proposal classification accuracy(%) with those of well-known methods on OUTEX dataset

Method	TC10	TC12	
		t184	horizon
CLBP_S/M	99.30	96.40	97.05
CLBP_S/M/C	99.20	96.20	97.25
BF+CLBP_S/M	99.60	97.29	97.99
BF+CLBP_S/M/C	99.50	97.13	98.40
ScatNet	98.39	96.23	98.38
CLBP_S/M+ScatNet	99.40	98.45	98.80
CLBP_S/M/C+ScatNet	99.51	98.66	98.94
BF+CLBP_S/M+ScatNet	99.82	98.22	99.40
BF+CLBP_S/M/C+ScatNet	99.87	98.43	99.63
NTLBP[43]	99.24	96.18	94.28
DLBP[17]	99.10	93.20	90.04
COV-LBPD[44]	98.78	95.72	97.62
LTP[16]	98.54	92.59	89.17
LBP [1]	97.70	87.30	86.40
MSJLBP[45]	96.67	95.21	95.74
PRICoLBP[46]	94.48	92.57	92.50
VZ-MR8[9]	93.59	92.55	92.82
NRLBP[47]	93.44	86.13	87.38
VZ-Path [10]	92.00	91.41	92.06
MBP[48]	89.92	95.18	95.55
FV-SIFT+ FC+FV-VD[14]	80.00	82.16	82.44

hancement in classification accuracy is approximately 6% for CLBP+ScatNet, 10% for BF+CLBP+ScatNet, and up to 15% when comparing to CLBP or ScatNet features alone. Also, our classification result on this database is ranked closely to FV-SIFT+ FC+FV-VD [14].

Table 5 Integrated Descriptor with BF+CLBP+ScatNet multi-scale: proposal classification accuracy(%) on KTHTIPS2b dataset and comparing with those of well-known methods

Method	Accuracy(%)
CLBP_S/M	63.30
CLBP_S/M/C	63.20
BF+CLBP_S/M	67.68
BF+CLBP_S/M/C	68.31
ScatNet	67.26
CLBP_S/M+ScatNet	69.15
CLBP_S/M/C+ScatNet	68.80
BF+CLBP_S/M+ScatNet	78.09
BF+CLBP_S/M/C+ScatNet	77.71
FV-SIFT+ FC+FV-VD[14]	81.50
MSJLBP[45]	65.51
ELBP[21]	64.84
MWLD[49]	64.70
COV-LBPD[44]	63.47
LTP[16]	62.12
PRICoLBP[46]	61.17
LBP [1]	60.35
MBP[48]	60.29
VZ-Path [10]	60.70
NTLBP[43]	58.78
NRLBP[47]	57.00
VZ-MR8[9]	55.7

108	0	0	0	0	0	0	0	0	0	0
0	61	0	0	0	7	0	0	40	0	0
0	1	95	0	7	0	0	2	0	1	2
0	1	1	103	1	2	0	0	0	0	0
0	0	0	0	39	2	0	54	5	6	2
0	8	24	7	0	56	2	0	11	0	0
3	0	3	0	0	1	96	0	0	5	0
0	0	0	0	5	0	0	96	0	0	7
0	8	1	0	0	6	0	0	92	1	0
0	0	1	0	0	0	0	0	0	107	0
0	0	11	0	4	0	14	7	0	0	72

Table 6 Confusion Matrix BF+CLBP on KTH-TIPS-2b

4.3 Framework experimental analysis and complexity

This section we first show the supplement of CLBP to ScatNet in the proposed framework, features generated from the framework are more tolerant to the variations in scale, pose and illumination with KTH-TIPS2b dataset when CLBP and ScatNet combined. Then, we represent the complexity of our method (Table 9). Finally, feature dimension reduction and its relevant accuracy are illustrated (Fig. 9)

Table 6, 7 , and 8 are confusion matrices of classification of one split on KTH-TIPS2b dataset with BF+CLBP, ScatNet, and the combined features respectively. The tables show that our framework leverages CLBP and ScatNet

108	0	0	0	0	0	0	0	0	0	0
15	44	1	2	0	8	14	0	21	2	1
1	0	101	0	1	0	0	3	0	2	0
5	12	1	77	0	0	2	0	11	0	0
0	0	2	0	64	0	4	30	0	7	1
3	11	7	12	0	38	19	0	9	4	5
19	0	0	0	0	0	89	0	0	0	0
0	1	0	0	6	1	0	99	0	0	1
3	13	2	10	0	2	12	0	62	4	0
11	0	0	0	0	0	1	0	0	96	0
11	3	25	0	0	0	10	0	0	3	56

Table 7 Confusion Matrix Scatnet on KTH-TIPS-2b

108	0	0	0	0	0	0	0	0	0	0
0	97	0	0	0	8	0	0	3	0	0
0	0	98	0	8	0	0	1	0	1	0
0	0	0	108	0	0	0	0	0	0	0
0	0	1	0	65	0	0	30	0	7	1
0	17	12	8	0	64	1	0	3	0	3
0	0	0	0	0	0	108	0	0	0	0
0	0	0	0	2	0	0	103	0	0	3
0	0	1	0	0	0	0	0	107	0	0
0	0	1	0	0	0	0	0	0	107	0
0	0	0	0	0	0	19	5	0	0	84

Table 8 Confusion Matrix of the proposal method on KTH-TIPS-2b

to reduce the misclassification cases, confusion matrix of the framework (table 8) shows a smaller miss-class numbers than those of the other two (table 6, 7). For example, the case of class 4 (Cork) and 7 (Lettuce Leaf), illustrated in Fig. 8, are classified incorrectly by both CLBP and ScatNet. However, the proposed framework does 100% correctly in these classes. This shows that CLBP and ScatNet can complement each other.

(Fig. 8) illustrates misclassification cases of the original features while the proposed method classify these images without error. However, BF+CLBP mis-classifies 2 left images: **cork** (class 4) is classified incorrectly into cracker (class 6), brown bread (class 2), corduroy (class 3), and cotton (class 5). While **lettuce leaf** (class 7) is classified incorrectly into aluminium foil (class 1). ScatNet mis-classifies 2 right images: **cork** (class 4) is misclassified into aluminium foil (class 1), brown bread (class 2), corduroy (class 3), lettuce leaf (class 7), and white bread (class 9). **lettuce leaf** (class 7) is misclassified into aluminium foil (class 1).

Among those texture feature methods mentioned in this paper, network-based approaches have high classification accuracy and obviously high complexity in comparison with others. We will present the complexity of our works, and compare it with a well-known state-of-the-art CNN, the FV-SIFT+ FC+FV-VD[14], (table 9). Feature extraction time (Fea. Ex. Time) of methods in second, is calculated on the same laptop with Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz, 2501 Mhz, 4 Core(s), 8 Logical Processor(s) and

Fig. 8 BF+CLBP mis-classifies (2 left images), ScatNet mis-classifies (2 right images), the proposed method classifies these images without error.

8Gb of RAM. We measure the feature extraction time by the average computation time of 500 images from KTH-TIPS2b dataset. Our framework feature extraction time depend of ScatNet parameters (scales), the smaller scale the faster feature extraction. In general, our approach takes a little longer time to extract features in comparison with FV-SIFT+ FC+FV-VD[14]. However, our method has the feature dimension which is much lower, this leads to a shorter classification time.

Table 9 Framework complexity and comparison, feature extraction time in seconds (Fea. Ex. Time(seconds)), feature dimension (Fea. Dimension)

Method	Fea. Ex. Time(second)	Fea. Dimension
Ours	0.7073	4,608
FV-SIFT+ FC+FV-VD[14]	0.5431	65,536

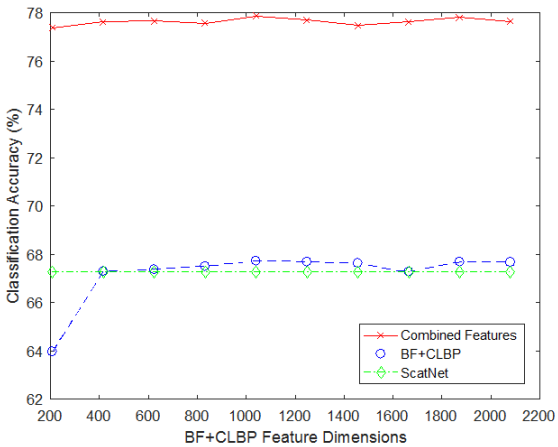

Fig. 9 Classification accuracy on KTH-TIPS-2b dataset in which dimension reduction of BF+CLBP-feature-vectors are done before concatenating with those of ScatNet.

Fig. 9 illustrates Classification accuracy on KTHTIPS2b dataset with dimension reduction of BF+CLBP feature vectors. There is a slight decrease in classification accuracy (from 68.38% to 67.28%) when dimension of BF+CLBP feature vectors are reduced from 3888 to 416 using PCA. However, the accurate rates of the combined method are stable around 77.63% with feature vector dimension ranging from 416 to 2288 (the feature vector dimension without dimension reduction is 4096), the accuracy peaks at 77.87% (the accurate rate without dimension reduction is 78.09%) with the feature vector dimension of 1248. Therefore, applying feature vector dimension reduction of BF+CLBP features before concatenating with those of ScatNet can be considered as a trade-off between classification accuracy and feature dimension.

5 Conclusion

In this paper, we have proposed a novel framework which generates handcrafted features for texture classification. It takes full advantages of the BF preprocessing technique [27], the local LBP features, and the global ones extracted from ScatNet. Experiments show our proposal enhances distinctiveness of texture while preserving the robustness to variations in illumination, rotation, and scale. Overall, ScatNet and LBP are not concurrent, but complementary while the preprocessing technique makes the LBP descriptors more robust. Our modest improvements in terms of classification accuracy have not reached the recent frontiers but are in the high ranking on the experimental databases, to the best of our knowledge. Future study can be drawn on the same domain with scale variation tolerance by using multi-scale training techniques.

References

1. T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.
2. R. M. Haralick, K. Shanmugam *et al.*, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.
3. A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55–73, 1990.
4. B. S. Manjunath and W.-Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, 1996.
5. S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
6. A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1186–1191, 1993.

7. D. Charalampidis and T. Kasparis, "Wavelet-based rotational invariant roughness features for texture classification and segmentation," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 825–837, 2002.
8. T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.
9. M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. Journal of Computer Vision*, vol. 62, no. 1-2, pp. 61–81, 2005.
10. ——, "A statistical approach to material classification using image patch exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2032–2047, 2009.
11. O. G. Cula and K. J. Dana, "Compact representation of bidirectional texture functions," in *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, 2001.
12. ——, "3d texture recognition using bidirectional feature histograms," *International Journal of Computer Vision*, vol. 59, no. 1, pp. 33–60, 2004.
13. L. Sifre and S. Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1233–1240.
14. M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, "Deep filter banks for texture recognition, description, and segmentation," *International Journal of Computer Vision*, vol. 118, no. 1, pp. 65–94, 2016.
15. T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.
16. X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Analysis and Modeling of Faces and Gestures*. Springer, 2007, pp. 168–182.
17. S. Liao, M. W. Law, and A. C. Chung, "Dominant local binary patterns for texture classification," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1107–1118, 2009.
18. Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, Jun. 2010.
19. ——, "Rotation invariant texture classification using lbp variance (lbpv) with global matching," *Pattern recognition*, vol. 43, no. 3, pp. 706–719, 2010.
20. X. Qian, X.-S. Hua, P. Chen, and L. Ke, "Plbp: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition*, vol. 44, no. 10, pp. 2502–2515, 2011.
21. L. Liu, L. Zhao, Y. Long, G. Kuang, and P. Fieguth, "Extended local binary patterns for texture classification," *Image and Vision Computing*, vol. 30, no. 2, pp. 86–99, 2012.
22. G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen, "Rotation-invariant image and video description with local binary pattern features," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1465–1477, 2012.
23. Y. Guo, G. Zhao, and M. PietikäInen, "Discriminative features for texture description," *Pattern Recognition*, vol. 45, no. 10, pp. 3834–3843, 2012.
24. S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
25. R. Porter and N. Canagarajah, "Robust rotation-invariant texture classification: wavelet, gabor filter and gmrf based schemes," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 144, no. 3, pp. 180–188, 1997.
26. G. M. Haley and B. Manjunath, "Rotation-invariant texture classification using a complete space-frequency model," *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 255–269, 1999.
27. N. Vu, T. P. Nguyen, and C. Garcia, "Improving texture categorization with biologically-inspired filtering," *Image Vision Comput.*, vol. 32, no. 6-7, pp. 424–436, 2014.

28. S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, 2005.
29. K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Trans. Graph.*, vol. 18, no. 1, pp. 1–34, 1999.
30. T. Ojala, T. Menp, M. Pietikinen, J. Viertola, J. Kyllnen, and S. Huovinen, "Outex - new framework for empirical evaluation of texture analysis algorithms," in *16th Int. Conf. on Pattern Recognition*, 2002, pp. 701–706.
31. B. Caputo, E. Hayman, M. Fritz, and J.-O. Eklundh, "Classifying materials in the real world," *Image Vision Comput.*, vol. 28, no. 1, pp. 150–163, 2010.
32. V.-L. Nguyen, N.-S. Vu, and P.-H. Gosselin, "A scattering transform combination with local binary pattern for texture classification," *14th International Workshop on Content-based Multimedia Indexing*, 2016.
33. V.-L. Nguyen, N.-S. Vu, H.-H. Phan, and P.-H. Gosselin, "An integrated descriptor for texture classification," *23rd International Conference on Pattern Recognition*, 2016.
34. S. Mallat, "Group Invariant Scattering," *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.
35. A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2010.
36. A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.
37. Y. Xu, X. Yang, H. Ling, and H. Ji, "A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid," in *CVPR*, 2010, pp. 161–168.
38. Y. Xu, S. Huang, H. Ji, and C. Fermüller, "Combining powerful local and global statistics for texture description," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 2009, pp. 573–580.
39. Y. Xu, H. Ji, and C. Fermüller, "Viewpoint invariant texture description using fractal analysis," *International Journal of Computer Vision*, vol. 83, no. 1, pp. 85–100, 2009.
40. E. Hayman, B. Caputo, M. Fritz, and J. Eklundh, "On the significance of real-world conditions for material classification," in *European Conference on Computer Vision*, 2004.
41. R. E. Broadhurst, "Statistical estimation of histogram variation for texture classification," in *Proc. Intl. Workshop on texture analysis and synthesis*, 2005, pp. 25–30.
42. M. Crosier and L. D. Griffin, "Using basic image features for texture classification," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 447–460, 2010.
43. A. Fathi and A. R. Naghsh-Nilchi, "Noise tolerant local binary pattern operator for efficient texture analysis," *Pattern Recognition Letters*, vol. 33, no. 9, pp. 1093–1100, 2012.
44. X. Hong, G. Zhao, M. Pietikäinen, and X. Chen, "Combining LBP difference and feature correlation for texture description," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2557–2568, 2014.
45. X. Qi, Y. Qiao, C. Li, and J. Guo, "Multi-scale joint encoding of local binary patterns for texture and material classification," in *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*, 2013.
46. X. Qi, R. Xiao, C.-G. Li, Y. Qiao, J. Guo, and X. Tang, "Pairwise rotation invariant co-occurrence local binary pattern," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 11, pp. 2199–2213, 2014.
47. J. Ren, X. Jiang, and J. Yuan, "Noise-resistant local binary pattern with an embedded error-correction mechanism," *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 4049–4060, 2013.
48. A. Hafiane, G. Seetharaman, and B. Zavidovique, "Median binary pattern for textures classification," in *Image Analysis and Recognition, 4th International Conference, ICIAR 2007, Montreal, Canada, August 22-24, 2007, Proceedings*, 2007, pp. 387–398.
49. J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao, "WLD: A robust local image descriptor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1705–1720, 2010.