Evaluation of LBP and Deep Texture Descriptors with a New Robustness Benchmark

Li Liu[1](✉), Paul Fieguth[2], Xiaogang Wang[3],
Matti Pietikäinen[4], and Dewen Hu[5]

[1] College of Information System and Management,
National University of Defense Technology, Changsha, China
liuli_nudt@nudt.edu.cn

[2] Department of Systems Design Engineering,
University of Waterloo, Waterloo, Canada
pfieguth@uwaterloo.ca

[3] Department of Electronic Engineering,
Chinese University of HongKong, Shatin, China
xgwang@ee.cuhk.edu.hk

[4] The Center for Machine Vision Research,
University of Oulu, Oulu, Finland
mkp@ee.oulu.fi

[5] College of Mechatronics and Automation,
National University of Defense Technology, Changsha, China
dwhu@nudt.edu.cn

Abstract. In recent years, a wide variety of different texture descriptors has been proposed, including many LBP variants. New types of descriptors based on multistage convolutional networks and deep learning have also emerged. In different papers the performance comparison of the proposed methods to earlier approaches is mainly done with some well-known texture datasets, with differing classifiers and testing protocols, and often not using the best sets of parameter values and multiple scales for the comparative methods. Very important aspects such as computational complexity and effects of poor image quality are often neglected.

In this paper, we propose a new extensive benchmark (RoTeB) for measuring the robustness of texture operators against different classification challenges, including changes in rotation, scale, illumination, viewpoint, number of classes, different types of image degradation, and computational complexity. Fourteen datasets from the eight most commonly used texture sources are used in the benchmark. An extensive evaluation of the recent most promising LBP variants and some non-LBP descriptors based on deep convolutional networks is carried out. The best overall performance is obtained for the Median Robust Extended Local Binary Pattern (MRELBP) feature. For textures with very large appearance variations, Fisher vector pooling of deep Convolutional Neural Networks is clearly the best, but at the cost of very high computational complexity. The sensitivity to image degradations and computational complexity are among the key problems for most of the methods considered.

Keywords: Local binary pattern · Deep learning · Performance evaluation · Texture classification

B. Leibe et al. (Eds.): ECCV 2016, Part III, LNCS 9907, pp. 69–86, 2016.
DOI: 10.1007/978-3-319-46487-9_5

1 Introduction

Texture is a ubiquitous and fundamental characteristic of the appearance of virtually all natural surfaces. Texture classification plays an important role in the fields of computer vision and pattern recognition, including biomedical image analysis, industrial inspection, analysis of satellite or aerial imagery, document image analysis, face analysis and biometrics, object recognition, material recognition and content based image retrieval.

The texture classification problem is conventionally divided into two subproblems of feature extraction and classification. It is generally agreed that the extraction of powerful texture features is of greater importance to the overall success of a texture classification strategy and, consequently, most research focuses on the feature extraction part, with extensive surveys [1,2]. Nevertheless it remains a challenge to design texture features which are computationally efficient, highly discriminative and effective, and robust to the imaging environment, including changes in illumination, rotation, view point, scaling, occlusion, and noise level.

A texture image or region obeys some statistical properties and exhibits repeated structures. Therefore, dense orderless statistical distribution of local texture features have been dominating the texture recognition literature since 1990s. The study of texture recognition has inspired many of the early representations of images. The idea of representing texture using the statistics of local features have led to the development of "textons" [3,4], the popular "Bag-of-Words (BoW)" models [5–9] and their variants such as the Fisher Vector [10]. Within the BoW framework, texture images are represented as histograms by pooling over a discrete vocabulary of discriminative and robust local features [4,6]. Important local texture descriptors include filter banks such as Gabor wavelets [11], LM filters [4], MR8 filters [6], raw pixel intensity-based features such as Local Binary Pattern (LBP) [5], Patch descriptors [8], random features [9], sparse descriptors such as SPIN [7], SIFT [1] and RIFT [7], and others [1,2]. Alternatives to simple histogram pooling have been proposed, such as Fisher Vectors (FVs) [12].

LBP [2,5] has emerged as one of the most prominent texture features and a great many new variants continue to be proposed. LBP's strengths include avoiding the time consuming discrete vocabulary pretraining stage in the BoW framework, its overall computational simplicity, its monotonic gray-scale invariance, its flexibility, and ease of implementation.

Recently, methods based on deep convolutional networks have emerged as a promising alternative to conventional "manually designed" features such as LBP. Important examples includes FV-CNN [13,14], obtained by Fisher Vector pooling of a Convolutional Neural Network (CNN) filter bank pretrained on large-scale datasets such as ImageNet, ScatNet (Scattering Convolution Networks) [15,16], PCANet [17] and RandNet [17]. When comparing these to LBP, only basic single resolution LBP methods have been normally considered [18] and no systematic performance evaluation has been carried out.

However, there has been a proliferation of LBP-related methods, so any comparison against a relatively small set cannot be considered an exhaustive investigation against the LBP strategy. Furthermore recent LBP studies show that the use of multi-scale information, for example, can significantly improve the performance of LBP variants, therefore it is highly pertinent to perform a more comprehensive performance evaluation and fair comparison of LBP approaches against novel challengers from the deep learning domain. The tests performed in this paper seek to explore and assess four criteria:

Computational complexity is an important factor in designing computer vision systems systems for real-world applications, particularly for portable computing systems (*e.g.*, smart phones, smart glasses) with strict low power constraints. Many papers emphasize primarily recognition accuracy, where we feel the need to balance this perspective with computational complexity as well.

Multiscale variations have been proposed for most LBP variations in their respective original works, but usually limited to three scales. Since the spatial support of a texture descriptor influences its classification performance, for fair comparison we propose to implement multiscale and rotational-invariant formulations of *each* LBP method up to nine scales, following the multiscale analysis approach proposed by Ojala *et al.* [5].

A large number of texture classes is one aspect complicating many texture analysis problems, together with the associated dynamics within a class (intra-class variations), such as variations in periodicity, directionality and randomness, and the external dynamics due to changes in the imaging conditions including variations in illumination, rotation, view point, scaling, occlusion and noise. Despite this complexity, most existing LBP variants have been evaluated only on small texture datasets with a relatively small number of texture classes, such as certain popular benchmark Outex test suites [5]. Experimental results based on datasets with small intraclass variations can be misleading; there are more challenging texture datasets with many texture classes or large intraclass variations, such as UIUC [7], UMD [19], CUReT [8] and KTHTIPS2b [20], DTD [21], ALOT [22] and Outex_TC40 [23], however, the performance of many LBP variants in these more challenging datasets is unknown. There is therefore significant value in performing a large scale empirical study on such challenging texture datasets.

Robustness to poor image quality, due to noise, image blurring and random image corruption, is usually neglected in the performance evaluation of texture operators. However any feature which performs only under idealized circumstances is almost guaranteed to disappoint in practice, therefore we are proposing an ensemble of robustness tests to better assess the generalizability of a given strategy away from its training setting. Noise can be severe in many medical (ultrasound, radiography), astronomical, and infrared images. The two main limitations in image accuracy are blur and noise, both of which we will test.

The main contributions of this paper are to propose a new challenging benchmark for a fair evaluation of different descriptors in texture classification, presenting a performance evaluation of the most promising LBP variants, and comparing to recent well-known texture features based on deep convolutional networks. In order to establish a common software platform and a collection of datasets for easy evaluation, we plan to make both the source code and datasets available on the Web.

2 Local Binary Pattern Methods Under Comparison

Local Binary Pattern (LBP). The original LBP [24] characterizes the spatial structure of a local image texture pattern by thresholding a 3×3 square neighborhood with the value of the center pixel and considering only the sign information to form a local binary pattern. A circular symmetric neighborhood is suggested, where locations that do not fall exactly at the center of a pixel are interpolated [5]. The LBP operator was extended to multiscale analysis to allow any radius and number of pixels in the neighborhood. A rotation invariant version $\mathrm{LBP}_{r,p}^{ri}$ of $\mathrm{LBP}_{r,p}$ was obtained by grouping together those LBPs that are actually rotated versions of the same pattern. Observing that some LBP patterns occur more frequently than others, the *uniform* LBP $\mathrm{LBP}_{r,p}^{u2}$ preserves only these frequent patterns, grouping all remaining ones. $\mathrm{LBP}_{r,p}^{riu2}$ is the combination of $\mathrm{LBP}_{r,p}^{ri}$ and $\mathrm{LBP}_{r,p}^{u2}$ [5].

Median Binary Pattern (MBP). Instead of using only the gray value of the center pixel for thresholding, MBP uses the local median. MBP also codes the value of the center pixel, resulting in a doubling in the number of LBP bins.

Local Ternary Pattern (LTP). LTP was proposed by Tan and Triggs in [25] to tackle the image noise in uniform regions. Instead of binary code, the pixel difference is encoded by three values according to a threshold T. LTP is capable of encoding pixel similarity modulo noise using the simple rule that any two pixels within some range of intensity are considered similar, but no longer strictly invariant to gray scale transformations.

Noise Resistant Local Binary Pattern (NRLBP). In a similar strategy to LTP, Ren *et al.* [26] proposed to encode small pixel difference as an uncertain bit, and then to determine its value based on the other bits of the LBP code. The main idea of NRLBP is to allow multiple LBP patterns to be generated at one pixel position, however NRLBP requires a lookup table of size 3^p for p neighboring pixels, which limits the neighborhood size.

Novel Extended Local Binary Pattern (NELBP). NELBP [27] is designed to make better use of the nonuniform patterns instead of discarding them. NELBP classifies and combines the "nonuniform" local patterns based on analyzing their structure and occurrence probability.

Local Binary Pattern Variance (LBPV). Guo *et al.* [28] proposed LBPV to incorporate local contrast information by utilizing the variance as a locally adaptive weight to adjust the contribution of each LBP code. LBPV avoids the quantization pretraining used in [5].

Noise Tolerant Local Binary Pattern (NTLBP). With similar motivations as NELBP [27], Fathi and Naghsh-Nilchi [29] proposed NTLBP that not only

Table 1. Summary of texture datasets used in our experimental evaluation. $\Theta_1 = \{5^\circ, 10^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}$, $\Theta_2 = \{0^\circ, 5^\circ, 10^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ\}$

Texture Dataset	Texture Classes	Sample Size (pixels)	# Images /Class	# Train /Class	# Test /Class	# Images in Total	Train/Test Predefined?	Instances Categories?	Description
Outex_TC10	24	128 × 128	180	20	160	4320	Yes	Instances	rotation changes (0° angle for training and angles in Θ_1 for testing)
Outex_TC12_000	24	128 × 128	200	20	180	4800	Yes	Instances	illumination variations, rotation changes
Outex_TC12_001	24	128 × 128	200	20	180	4800	Yes	Instances	(0° angle for training and angles in Θ_2 for testing)
CUReT	61	200 × 200	46	46	92	5612	No	Instances	illumination changes, small rotations, shadowing, pose changes
Brodatz	111	215 × 215	9	3	6	999	No	Instances	lack of intraclass variations
BrodatzRot	111	128 × 128	9	3	6	999	No	Instances	rotation changes, lack of intraclass variations
UIUC	25	320 × 240	40	20	20	1000	No	Instances	strong scale, rotation and viewpoint changes, nonrigid deformations
UMD	25	320 × 240	40	20	20	1000	No	Instances	strong scale, rotation and viewpoint changes
KTH-TIPS2b	11	200 × 200	432	324	108	4752	Yes	Categories	illumination changes, small rotation changes, large scale changes
DTD	47	Not Fixed	120	80	40	5640	No	Categories	Attribute-based class, many texture categories per class
ALOT	250	384 × 256	100	50	50	25000	No	Instances	strong illumination changes, large number of classes, rotation changes
Outex_TC40_A	294	128 × 128	180	80	100	52920	Yes	Instances	rotation changes, large number of classes
Outex_TC40_B	294	128 × 128	180	80	100	52920	Yes	Instances	illumination changes, rotation changes, large number of classes
Outex_TC40_C	294	128 × 128	180	80	100	52920	Yes	Instances	illumination changes, rotation changes, large number of classes

Datasets for Noise Robustness Evaluation

Texture Dataset	Texture Classes	Sample Size (pixels)	# Images /Class	# Train Images in Total	# Test Images in Total	Description
Outex_TC11n	24	128 × 128	20	480 (20 ∗ 24)	480 (20 ∗ 24)	Training: illuminants (inca), Rotations (0°)
Outex_TC23n	68	128 × 128	20	1360 (20 ∗ 68)	1360 (20 ∗ 68)	Testing: Training images injected with Gaussian Noise
Outex_TC11b	24	128 × 128	20	480 (20 ∗ 24)	480 (20 ∗ 24)	Training: illuminants (inca), Rotations (0°)
Outex_TC23b	68	128 × 128	20	1360 (20 ∗ 68)	1360 (20 ∗ 68)	Testing: Training images blurred by Gaussian PSF
Outex_TC11s	24	128 × 128	20	480 (20 ∗ 24)	480 (20 ∗ 24)	Training: illuminants (inca), Rotations (0°)
Outex_TC23s	68	128 × 128	20	1360 (20 ∗ 68)	1360 (20 ∗ 68)	Testing: Training images injected with Salt-and-Pepper
Outex_TC11c	24	128 × 128	20	480 (20 ∗ 24)	480 (20 ∗ 24)	Training: illuminants (inca), Rotations (0°)
Outex_TC23c	68	128 × 128	20	1360 (20 ∗ 68)	1360 (20 ∗ 68)	Testing: Training images with Random Pixel Corruption

uses nonuniform patterns but also tolerates noise by using a circular majority voting filter and a scheme to regroup the nonuniform LBP patterns into several different classes.

Pairwise Rotation Invariant Cooccurrence Local Binary Pattern (PRICoLBP). Borrowing from Gray Level Cooccurrence Matrices (GLCM) [30], Qi *et al.* [31] proposed PRICoLBP to encapsulating the joint probability of pairs of LBPs at relative displacements. PRICoLBP incorporates two types of context: spatial cooccurrence and orientation cooccurrence. The method aims to preserve the relative angle between the orientations of individual features. The length of the feature vector may limit the applicability of PRICoLBP.

Multiscale Joint encoding of Local Binary Pattern (MSJLBP). Instead of considering cooccurrences of LBPs at different locations as in PRICoLBP [31], MSJLBP [32] was proposed to jointly encode the pairwise information of LBPs at the same centered location but from two different scales.

Completed Local Binary Pattern (CLBP). CLBP was proposed by Guo *et al.* [33] to combine multiple LBP type features (CLBP_S, CLBP_M and CLBP_C) via joint histogramming for texture classification. The image local differences between a center pixel and its neighbors are decomposed into two complementary components: the signs and the magnitudes (CLBP_S and CLBP_M). The center pixels, representing image gray level, were also regarded to have discriminative information and are converted into a binary code by global thresholding.

discriminative Completed Local Binary Pattern (disCLBP). Guo *et al.* [34] proposed a three-layered learning model, estimating the optimal pattern subset of interest by simultaneously considering the robustness, discriminative power and representation capability of features. This model is generalized and can be integrated with existing LBP variants such as conventional LBP, rotation invariant patterns, CLBP and LTP to derive new image features.

Table 2. Classification results (%) for various LBP variants on the Outex_TC10 and Outex_TC12 (Outex_TC12_000 and Outex_TC12_001) test suites as a function of neighborhood size (the number scales used for multiscale analysis). For each method, the highest classification accuracies are highlighted in bold for each dataset. LEP filtering support is 65 × 65. Some results (○) are not provided for efficiency reasons.

	Test Suite	**Outex_TC10 (Rotation Invariance)**									**Outex_TC12 (Illumination and Rotation Invariance)**								
No.	Method	3×3	5×5	7×7	9×9	11×11	13×13	15×15	17×17	19×19	3×3	5×5	7×7	9×9	11×11	13×13	15×15	17×17	19×19
1	LBP^{riu2}	84.71	93.44	97.21	98.91	99.01	99.38	99.56	99.66	**99.69**	64.97	82.07	86.79	89.64	89.12	89.72	90.81	91.39	**92.14**
2	MBP^{riu2}	80.21	87.40	89.92	92.47	94.24	94.90	95.16	95.21	**95.29**	63.18	73.01	79.71	83.66	84.57	85.09	85.69	86.22	**86.69**
3	LTP^{riu2}	92.94	97.14	98.54	99.32	99.53	99.74	99.84	99.84	**99.92**	73.59	86.46	90.88	92.08	92.35	92.78	93.25	93.77	**94.28**
4	NRLBP^{riu2}	89.79	93.78	96.67	97.01	**98.07**	97.81	95.60	95.05	93.44	71.35	83.00	87.05	88.92	89.57	**90.20**	88.78	87.48	86.76
5	NELBP	83.52	93.88	97.08	98.70	98.88	98.93	99.48	99.53	**99.64**	69.02	85.34	88.72	89.91	89.59	90.10	91.30	92.15	**93.55**
6	NTLBP	84.24	91.88	96.15	98.10	98.88	99.19	**99.35**	99.32	99.24	67.06	82.21	88.28	91.61	92.71	93.63	94.88	**95.27**	95.23
7	PRICoLBP$_g$	—	—	—	—	—	**94.48**	—	—	—	—	—	—	—	—	**92.53**	—	—	—
8	MSJLBP	—	—	**96.67**	—	—	—	—	—	—	—	—	**95.47**	—	—	—	—	—	—
9	*dis*CLBP	89.30	97.47	98.93	99.79	**99.95**	○	○	○	○	75.22	89.80	94.40	96.00	**96.10**	○	○	○	○
10	LEP	—	—	—	—	—	—	—	—	**81.90**	—	—	—	—	—	—	—	—	**81.46**
11	CLBP	96.72	98.67	99.35	99.45	99.51	99.51	99.51	99.53	**99.58**	91.54	94.48	95.67	**95.78**	95.49	95.39	95.43	95.43	95.42
12	ELBP	96.41	99.38	99.66	**99.71**	99.71	99.66	99.64	99.56	99.53	92.08	97.37	**97.57**	97.08	96.52	96.10	96.06	96.05	96.03
13	BRINT	91.88	96.95	98.52	99.04	99.32	99.32	99.30	**99.40**	99.35	87.48	94.29	96.28	97.16	97.29	97.53	97.71	97.96	**98.13**
14	MRELBP	—	98.44	—	99.69	—	99.79	—	**99.82**	—	—	96.24	—	99.03	—	99.56	—	**99.57**	—
15	LBPV^{riu2}	91.30	94.35	97.24	98.49	98.93	99.22	**99.27**	99.14	99.11	76.88	86.76	92.72	93.34	93.92	93.81	93.92	**94.03**	94.00
16	CLBPHF	87.42	94.61	98.20	99.01	99.56	99.69	**99.71**	99.71	99.69	78.39	90.29	93.34	94.10	94.07	94.07	94.39	94.61	**94.80**
17	LBPD	—	—	**98.78**	—	—	—	—	—	—	—	—	**96.67**	—	—	—	—	—	—
18	RILPQ	—	—	—	—	—	**99.58**	—	—	—	—	—	—	—	—	**97.43**	—	—	—

Extended Local Binary Pattern (ELBP). ELBP is proposed by Liu *et al.* [35] to combine several LBP–related features: pixel intensities and differences from local patches. The intensity-based features consider the intensity of the central pixel (CI) and those of its neighbors (NI); differences are computed by radius and by angle. ELBP reflects the combination of radial differences (RD) and two intensities.

Binary Rotation Invariant and Noise Tolerant Texture descriptor (BRINT). Similar to CLBP [33] and ELBP [35], BRINT [36] combines three individual descriptors BRINT_S, BRINT_M and BRINT_C. Unlike CLBP and ELBP, where only rotation invariant uniform patterns are considered, BRINT uses all of the rotation invariant patterns. In BRINT, pixels are sampled in a circular neighborhood, but keeping the number of bins in a single-scale LBP histogram constant and small, such that arbitrarily large circular neighborhoods can be sampled and compactly encoded. BRINT has low feature dimensionality and noise robustness.

Median Robust Extended Local Binary Pattern (MRELBP). In order to jointly capture microtexture and macrotexture information, Liu *et al.* [37] built on the NI, RD and CI of ELBP [35] but with nonlocal–median pixel sampling, significantly outperforming ELBP, especially in situations of noise, image blurring and random image corruption. Moreover, MRELBP is fast to compute and has much lower feature dimensionality.

Completed Local Binary Pattern Histogram Fourier Features (CLBPHF). Ahonen *et al.* [38] proposed the LBP Histogram Fourier features (LBPHF) to achieve rotation invariance globally by first computing a uniform LBP histogram over the whole image, and then constructing rotationally invariant features from the DFT transform of the histogram. Later in [39], LBPHF is combined CLBP [33] to further improve its distinctiveness and results CLBPHF.

Local Energy Pattern (LEP). Zhang *et al.* [40] proposed LEP for texture classification, where multiscale and multiorientation Gaussian-like second order derivative filters are used to filter the original image. LEP encodes the relationship among different feature channels using an N-nary coding scheme, rather than binary. One downside of the LEP is that pretraining is required.

Local Binary Pattern Difference (LBPD). Covariance Matrices capture correlation among elementary features of pixels over an image region. Ordinary LBP features cannot be used as elementary features, since they are not numerical variables in Euclidean spaces. To address this problem, Hong *et al.* [41] developed COV-LBP. First the LBPD, a Euclidean space variant, was proposed, reflecting how far one LBP lies from the LBP mean of a given image region. Secondly, the covariance was found of a bank of discriminative features, including LBPD.

Rotation Invariant Local Phase Quantization (RILPQ). LPQ [42] is generated by quantizing the Fourier transform phase in local neighborhoods, such that histograms of LPQ labels computed within local regions are used as a texture descriptor similar to LBP, leading to a tolerance to image blur. LPQ was generalized with a rotation invariant extension to RILPQ [43].

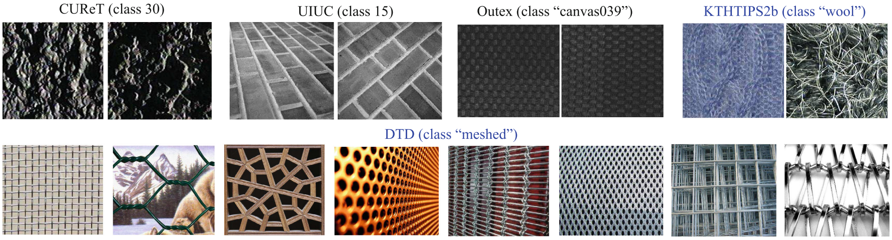

Fig. 1. Datasets such as CUReT, UIUC and Outex addressed the problem of instance-level identification. KTH-TIPS2b addressed the problem of category-level material recognition. The DTD dataset addresses a very different problem of category-level attribute recognition, *i.e.* describing a pattern using intuitive attributes. In DTD, many visually very different texture categories appear in the same attribute class, which makes the classification problem very challenging.

2.1 Recent Non-LBP Deep Learning Approaches

FV-CNN. Deep convolutional neural networks (CNN) have demonstrated their power as a universal representation for recognition. However, global CNN activations lack geometric invariance, which limits their robustness for recognizing highly variable images. Cimpoi *et al.* [13,14] propose an effective texture descriptor FV-CNN, obtained by first extracting CNN features from convolutional layers for an texture image at multiple scale levels, and then performing orderless Fisher Vector pooling of these features.

ScatNet. Despite significant progress, there is still little insight into the internal operation and behavior of deep CNN models. Arguably one instance that has led to a clear mathematical justification is the multistage architectures of ConvNet [13,44], and specifically in the wavelet convolutional scattering network (ScatNet) [15,16] where the convolutional filters are predefined as wavelets, hence no learning process is needed. ScatNet has been extended to achieve rotation and scale invariance [45].

PCANet and **RandNet.** Motivated by ScatNet, Chan *et al.* [17] proposed a simple deep learning network, PCANet, based on cascaded / multistage principal component analysis (PCA), binary hashing, and histogram pooling. The authors also introduced RandNet, a simple variation of PCANet, which shares the same topology as PCANet, but in which the cascaded filters are randomly selected, not learned.

Table 3. Performance comparison for LBP variants tested on a number of texture datasets in terms of classification scores (%) and computational complexity (including feature extraction time and feature dimensionality). All results in Part I are obtained with a NNC classifier, with the exception of SVM for the DTD results. Results for PCANet and RandNet on DTD are also obtained with SVM. For each dataset, the highest score is shadowed, and those scores which are within 1 % of the highest are boldfaced. For each method, the total number of highlighted scores are given in the "# Bold" column. In the "Time" column, the reported time does not include the extra training time for those methods labeled with (⋆). The (◇) label in the LBPD method means that although LBPD has low feature dimensionality, it is pretty time consuming in the classification stage since it requires an affine invariant metric in the NNC classification.

		Results on Fourteen Texture Datasets													Computation Cost		Noise Robustness			
	Method	Outex_TC10	Outex_TC12	CUReT	Brodatz	BrodatzRot	UMD	UIUC	KTHTIPS2b	DTD	ALOT	Outex_TC40_A	Outex_TC40_BC	# Bold	Feature Extraction Time	Feature Dimension	Outex_TC23n $\sigma = 5$	Outex_TC23b $\sigma = 1$	Outex_TC23s $\rho = 15\%$	Outex_TC23c $\upsilon = 20\%$
No.	# Classes	(24)	(24)	(61)	(111)	(111)	(25)	(25)	(11)	(47)	(250)	(294)	(294)		(ms)		(68)	(68)	(68)	(68)
	Part I: Evaluation the performance of representative LBP methods.																			
1	MRELBP [37]	**99.82**	**99.58**	**97.10**	90.86	81.92	**98.66**	94.73	**68.98**	44.89	**97.28**	96.20	78.97	6	416.6	800	**79.2**	**85.8**	**99.9**	**96.9**
2	CLBP [33]	**99.45**	95.78	**97.33**	**92.34**	84.35	**98.62**	**95.75**	64.18	42.63	**96.74**	**96.98**	65.49	7	127.9	3552	5.6	36.1	2.9	2.9
3	ELBP [35]	**99.66**	97.57	**96.60**	**93.24**	**85.92**	**98.93**	94.61	64.84	39.89	**97.21**	96.18	67.70	6	114.6	2200	3.3	19.7	1.5	4.4
4	CLBPHF [39]	**99.69**	94.80	**97.05**	91.95	82.07	97.24	92.55	**68.10**	**50.21**	**96.30**	96.42	69.63	5	256.2	4580	17.5	39.1	2.9	1.5
5	*dis*CLBP [34]	**99.95**	96.10	**96.98**	**93.18**	83.77	97.53	94.24	63.83	44.47	95.01	**97.54**	74.00	4	(⋆)585.8	7796	12.3	27.1	4.4	2.6
6	LTP^{riu2} [25]	**99.92**	94.28	**96.33**	**92.41**	83.51	96.66	93.27	63.45	41.45	94.60	**96.85**	69.14	4	231.8	420	7.7	24.3	3.5	2.9
7	BRINT [36]	**99.35**	98.13	**97.02**	90.83	78.77	97.44	93.30	66.67	45.35	96.13	96.24	**81.85**	3	248.8	1296	27.4	59.1	1.5	1.6
8	LBP^{riu2} [5]	**99.69**	92.14	**97.03**	90.70	79.22	96.15	88.36	62.69	37.09	94.15	94.83	71.72	2	87.2	210	8.4	16.6	1.5	1.5
9	NELBP [27]	**99.64**	93.55	**96.85**	90.19	80.08	95.55	88.29	62.39	39.93	95.20	95.39	74.87	2	91.3	273	10.3	17.8	1.5	1.5
10	MSJLBP [32]	96.67	95.47	**97.20**	**92.94**	79.11	96.53	83.00	65.51	43.14	95.65	88.59	60.09	2	854.6	3540	4.9	14.8	3.5	2.7
11	NTLBP [29]	**99.32**	95.27	96.11	89.31	80.25	95.72	88.13	61.30	38.24	94.47	91.70	69.49	1	332.3	388	9.0	21.7	4.7	3.7
12	PRICoLBP$_g$ [31]	94.48	92.53	96.25	**92.94**	77.00	95.69	80.38	61.17	44.53	94.38	89.56	64.16	1	380.4	3540	5.6	19.6	2.1	1.5
13	LBPV^{riu2} [28]	**99.27**	93.92	95.85	87.63	75.89	93.79	81.98	59.03	36.21	91.87	92.88	73.20	1	350.7	158	15.4	15.6	1.5	1.5
14	RILPQ [43]	**99.58**	97.43	92.15	91.37	79.59	97.49	91.17	58.75	42.70	94.85	90.76	69.33	1	44.8	256	56.5	53.9	1.5	2.6
15	LBPD [41]	98.78	96.67	94.23	89.74	74.79	92.99	90.98	63.47	35.86	92.82	89.96	60.60	0	(◇)54.2	289	14.8	40.2	2.9	2.6
16	NRLBP^{riu2} [26]	98.07	89.57	94.00	87.42	75.77	93.32	81.10	58.61	37.77	87.86	89.93	61.34	0	356.9	50	9.1	20.3	2.9	5.3
17	LEP [40]	81.90	81.46	88.31	82.64	61.41	91.75	81.80	63.13	38.67	89.67	74.97	56.07	0	(⋆)1088.9	520	**76.8**	**100.0**	1.8	5.6
18	MBP^{riu2} [46]	95.29	86.69	92.09	87.25	74.57	92.41	80.89	61.49	27.73	88.23	84.90	45.46	0	215.6	420	5.2	13.5	2.5	2.6
	Part II: comparing MRELBP with deep convolutional network based approaches.																			
1	MRELBP (SVM) [37]	**99.97**	**99.77**	**99.02**	93.12	85.06	99.36	96.88	77.91	44.89	99.08	**97.15**	**77.79**	5	416.6	800	**70.5**	**69.8**	**99.1**	**95.5**
2	FV-VGGVD (SVM) [13]	80.0	82.3	**99.0**	**98.7**	**92.1**	**99.9**	**99.8**	**88.2**	**72.3**	**99.5**	93.7	71.6	8	(⋆)2655.4	65536	**71.5**	**83.6**	5.2	9.5
3	FV-VGGM (SVM) [13]	72.8	77.5	98.7	**98.6**	**88.2**	**99.9**	**99.7**	**79.9**	**66.8**	**99.4**	92.6	56.8	7	(⋆)358.8	65536	43.9	65.7	1.5	4.9
4	ScatNet (PCA) [16]	**99.69**	**99.06**	**99.66**	84.46	75.08	98.40	96.15	68.92	35.72	98.03	**94.07**	**77.93**	5	10883.7	596	31.3	53.0	1.5	1.5
5	FV-AlexNet (SVM) [13]	67.3	72.3	98.4	98.2	83.1	**99.7**	99.1	77.9	62.9	99.1	90.4	51.8	1	(⋆)238.6	32768	46.0	63.6	5.0	8.6
6	ScatNet (NNC) [16]	98.59	98.10	95.51	83.03	73.72	93.36	88.64	63.66	26.53	85.27	87.55	72.45	0	10883.7	596	45.3	41.9	1.5	2.9
7	PCANet [17] (NNC)	39.87	45.53	92.03	90.89	37.21	90.50	57.70	59.43	41.44	88.35	59.49	44.39	0	(⋆)711.8	2048	50.7	51.9	1.5	1.5
8	PCANet^{riu2} [17] (NNC)	35.36	40.88	81.48	85.76	29.96	85.67	49.80	52.15	30.11	79.77	33.25	21.80	0	(⋆)725.6	80	43.9	36.8	1.5	2.6
9	RandNet [17] (NNC)	47.43	52.45	90.87	91.14	40.84	90.87	56.57	60.67	36.66	86.94	65.28	42.55	0	711.8	2048	6.2	27.7	1.5	1.5
10	RandNet^{riu2} [17] (NNC)	43.54	45.70	80.46	85.59	30.78	87.40	48.20	56.90	26.51	73.51	45.14	25.96	0	725.6	80	5.9	20.6	1.5	1.5

Table 4. Classification scores (%) in the context of additive Gaussian noise and Gaussian blurring.

Robust to		Gaussian noise		Gaussian blur							
Dataset (Outex_)		TC11n	TC23n	TC11b				TC23b			
No	Method	$\sigma = 5$	$\sigma = 5$	$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$	$\sigma = 1.25$	$\sigma = 0.5$	$\sigma = 0.75$	$\sigma = 1$	$\sigma = 1.25$
1	MRELBP [37]	**91.5**	**79.2**	**100.0**	**100.0**	**93.8**	**75.4**	**99.9**	**97.9**	**85.8**	**61.8**
2	CLBP [33]	11.9	5.6	98.8	74.8	49.6	23.1	86.6	55.4	36.1	21.2
3	ELBP [35]	9.4	3.3	98.3	71.5	38.5	21.5	86.2	39.9	19.7	11.0
4	CLBPHF [39]	20.6	17.5	99.6	81.3	47.9	29.4	85.4	59.2	39.1	25.1
5	*dis*CLBP [34]	25.2	12.3	100.0	70.2	39.4	20.8	95.6	51.0	27.1	14.1
6	LTP^{riu2} [25]	13.7	7.7	96.9	58.3	27.3	13.7	77.3	43.1	24.3	13.3
7	BRINT [36]	*61.9*	*27.4*	*100.0*	*97.1*	*80.4*	*44.6*	*100.0*	*79.5*	*59.1*	*39.1*
8	LBP^{riu2} [5]	17.7	8.4	94.2	46.5	24.6	12.7	72.4	30.3	16.6	9.7
9	NELBP [27]	19.2	10.3	94.0	47.7	28.3	17.1	73.3	32.0	17.8	10.5
10	MSJLBP [32]	17.7	4.9	96.0	46.0	26.0	11.9	74.9	28.9	14.8	8.9
11	NTLBP [29]	24.0	9.0	96.3	49.0	33.1	19.4	80.1	35.7	21.7	14.1
12	PRICoLBP$_g$ [31]	15.4	5.6	98.1	50.0	26.5	14.4	81.1	32.5	19.6	11.3
13	LBPV^{riu2} [28]	27.1	15.4	96.9	52.1	22.3	17.1	73.9	34.3	15.6	8.3
14	RILPQ [43]	*82.9*	*56.5*	*100.0*	*99.2*	*76.7*	*45.8*	*100.0*	*76.0*	*53.9*	*37.2*
15	LBPD [41]	24.6	14.8	99.4	85.8	65.2	45.4	87.7	56.0	40.2	30.6
16	NRLBP^{riu2}[26]	21.7	9.1	93.3	46.0	20.0	9.2	63.2	36.3	20.3	8.8
17	LEP [40]	**91.9**	**76.8**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**99.8**
18	MBP^{riu2} [46]	12.1	5.2	85.4	29.0	18.5	11.9	58.7	22.5	13.5	10.6
19	FV-VGGVD (SVM) [13]	**93.1**	**71.5**	**100.0**	**100.0**	**96.5**	**89.8**	**99.6**	**94.1**	**83.1**	**71.8**
20	FV-VGGM (SVM) [13]	81.5	43.9	100.0	99.0	87.3	60.8	96.5	87.7	65.7	42.4
21	ScatNet (PCA) [16]	60.2	31.3	100.0	94.8	80.0	64.6	97.7	72.4	53.0	41.1
22	FV-AlexNet (SVM) [13]	81.5	46.0	100.0	98.8	87.7	60.4	97.1	82.8	63.6	43.4
23	ScatNet (NNC) [16]	77.1	45.3	100.0	91.7	68.5	40.2	92.7	60.4	41.9	24.0
24	PCANet [17]	74.0	50.7	100.0	100.0	86.0	56.9	100.0	99.2	51.9	31.0
25	PCANet^{riu2}[17]	62.7	43.9	100.0	88.8	52.5	32.5	100.0	64.6	36.8	25.7
26	RandNet [17]	15.3	6.2	100.0	78.1	56.5	37.4	96.2	40.4	27.7	19.4
27	RandNet^{riu2} [17]	14.8	5.9	97.8	64.2	42.1	33.3	81.1	37.2	20.6	18.9

3 Experimental Setup

We conducted experiments on the *fourteen* texture datasets shown in Table 1. These datasets are derived from the eight most commonly used texture sources: Outex [23], CUReT [8], Brodatz [47], UIUC [7], UMD [19], KTHTIPS2b [20], ALOT [22] and DTD [21]. The experimental setup on the three test suites **Outex_TC10**, **Outex_TC12_000** and **Outex_TC12_001**, which were designated by Ojala *et al.* [5] for rotation and illumination invariant texture classification, was kept exactly as in [5].

Following Ojala *et al.* we created **Outex_TC40_A**, **Outex_TC40_B** and **Outex_TC40_C** [5] for large-scale texture classification. Each dataset contains 294 texture classes, with training data acquired under illuminant "inca" and rotations 0°, 30°, 45° and 60°, and tested with rotations 5°, 10°, 15°, 75° and 90°. The test images in **A** are from illumination "inca", the same as the training images, and thus simpler than datasets **B** and **C**, with testing data from illumination types "Horizon" and "TL84", respectively.

For **CUReT**, we use the same subset of images as in [8,9]. For **Brodatz** [47] we use the same dataset as [1,7,48]. The **BrodatzRot** dataset is generated from Brodatz by rotating each sample at a random angle, helping to test rotation invariance. The challenging **UIUC** dataset [7] contains images with strong scale, rotation and viewpoint changes in uncontrolled illumination environment. The **UMD** dataset [19] is similar to UIUC with higher resolution images but exhibits less nonrigid deformations and stronger illumination changes. We resize images in ALOT to obtain lower resolution (384 × 256). ALOT is challenging as it represents a significantly larger number of classes (250) compared to UIUC and UMD (25) and has very strong illumination changes (8 levels of illumination), albeit with less dramatic viewpoint changes.

Generalizing the texture recognition problem to a recognition of surface material, **KTH-TIPS2b** [20] has four physical samples for each class, imaged under 3 viewing angles, 4 illuminants, and 9 different scales. A quite different database, **DTD** contains textures in the wild, collected from the web and organized according to a list of 47 attribute categories inspired from human perception, with a single category containing rather different textures, as shown in Fig. 1. This dataset aims at supporting real-world applications where the recognition of texture properties is a key component.

To evaluate the robustness with respect to random noise, we considered Gaussian noise, image blurring, salt-and-pepper noise, and random pixel corruption, the same noise types tested in [49]. We use only the noise-free texture images for training and test on the noisy data, as summarized in Table 1. The test suites are based on Outex_TC11n and Outex_TC23n, which have 24 and 68 texture classes, respectively. The noise parameters include Gaussian noise standard deviation σ, Gaussian blur standard deviation σ, Salt-and-Pepper noise density ρ, and pixel corruption density υ.

Implementation Details. For the evaluated methods, we use the original source code if it is publicly available, and for the remainder we have developed our own implementation. To ensure fair comparisons, the parameters of each method are fixed across all the datasets, since it is difficult and undesirable to tune the parameters of each method for each evaluation. In most cases we use the default parameters suggested in the original papers. For ScatNet, we used the same feature presented in [15]. For PCANet and RandNet, we used the parameter settings suggested for texture classification in [17].

For most of the tested LBP methods, multiscale variations had been proposed in the original work, but usually limited to three scales. Since the spatial support of a texture descriptor influences its classification performance, for fair comparison we implemented multiscale and rotational invariant formulations of *each* LBP method up to nine scales, following the multiscale analysis approach proposed by Ojala *et al.* [5], representing a texture image by concatenating histograms from multiple scales.

Each texture sample is preprocessed, normalized to zero mean and unit standard deviation. For CUReT Brodatz, BrodatzRot, UIUC, UMD and ALOT, half of the class samples were selected at random for training and the remaining half for testing, and all results are reported over 100 random partitionings of training and testing sets. For KTHTIPS2b, we follow the training and testing scheme of [50]: training on three samples and testing on the remainder. For DTD, we follow Cimpoi *et al.* [13,21], where 80 images per class were randomly selected as training and the rest 40 as testing. All results for DTD are reported over 10 random partitionings of training and testing sets, following [13]. There have been some proposals to use more sophisticated classifiers, such as support vector machines (SVM), SVM ensembles, decision trees, or random forests. However, in this work our focus was on the distinctiveness and robustness of various LBP variants, rather than on the impact of the classifier. Therefore, unless otherwise stated, we limit our study to using the nearest neighbor classifier (NNC) and keep the other components as similar as possible.

4 Experimental Results

4.1 Overall Results

Table 2 evaluates the multiscale and rotational-invariant formulations of each LBP method up to nine scales. We can observe a general trend of performance increase with neighborhood size, with most LBP methods achieving a best performance beyond three scales, clearly indicating the necessity of using larger areas of spatial support for LBP feature extraction. Based on the results in Table 2, in our following experiments we use that neighborhood size which gives the highest score for each LBP method.

The main results for RoTeB are summarized in Table 3, including a comprehensive evaluation of all methods on fourteen benchmark datasets with varying difficulty, computation complexity comparison (including feature extraction time and feature dimensionality), with detailed noise robustness evaluation presented in Tables 4 and 5.

The most robust method is MRELBP [37] which gives the best overall performance, considering the trade off between classification accuracy, computational complexity and robustness to several types of noise. Generally MRELBP even performs better than the recent well-known deep convolutional networks based approach — ScatNet [45]. Keep in mind that the expensive computational cost of ScatNet is a severe drawback. The MRELBP benefits from its sampling scheme the spatial domain spanned by which is much larger than by many other LBP variants. This is likely to result in better discrimination capability. More importantly, instead of applying the standard thresholding to the raw pixel values, MRELBP applies it to the local medians, which works surprisingly robustly.

For the noise-free results of Table 3, we can clearly observe the best performing methods as CLBP [33], ELBP [35], MRELBP [37], CLBPHF [39], ScatNet (PCA) [15,16] and FV-CNN [13]. Among these six methods, clearly the feature extraction time of ScatNet is much more longer than others and represents a significant drawback. The feature dimensionality of CLBP, ELBP, and CLBPHF are relatively high, with the FV-CNN at an extremely high feature dimension. A serious shortcoming of PCANet and RandNet is their lack of rotation invariance.

If the textures have very large within-class appearance variations, due to view and scale variations and combined texture categories as in DTD, then the FV-CNN methods cleraly perform the best. Nevertheless, from the Outex results it can be observed that FV-CNN is relatively weak on rotation invariance, despite FV-CNN methods using data augmentation to explore multiscale information. Moreover, FV-CNN is computationally expensive, making it unfeasible to run in real-time embedded systems with low-power constraints. Interestingly, CLBPHF [39] works rather well for DTD, perhaps because it is more insensitive to large texture appearance variations than the other LBP descriptors. The 50.21 % of CLBPHF on DTD is much higher than the scores given by MR8, LM filters and Patch features, close to 52.3 % of BoW encoding of SIFT features reported in [14].

Finally, from Table 3, the best scores on datasets Outex_TC10, Outex_TC12 and CUReT are 99.95 %, 99.58 % and 99.66, nearly perfect scores even with simple NNC classification. Especially for Outex_TC10, thirteen methods give scores higher than 99 %, leaving essentially no room for improvement. Because of that saturation, and because most LBP variants have not been evaluated in recognizing a large number of texture classes, we prepared the new Outex_TC40 benchmark test suite with 294 texture classes, where the results are significantly more spread out.

4.2 Noise Robustness

Noise robustness results are shown in Tables 4 and 5. The training images were all noise free which makes the problem very hard. From Table 3 the overall best results (without noise) were given by CLBP, CLBPHF, ELBP, MRELBP, ScatNet (PCA) and FV-CNN, however with the exception of MRELBP, all of them perform poorly in noisy situations, especially when the noise level is high. The results in both tables are consistently strong: MRELBP has exceptional noise tolerance that could not be matched by any of the other tested methods, clearly driven by the nonlinear, regional medians captured by MRELBP.

From the random noise and blur tests of Table 4 the best performing methods are LEP, MRELBP and FV-CNN, due to the filtering built into each of these methods. Although RILPQ is specifically designed to address image blur, it is outperformed by LEP, MRELBP and FV-CNN in that context.

Table 5 presents the results for salt-and-pepper noise and random pixel corruption respectively. As the noise level increases, with few exceptions the performance of most of the LBP methods reduces to random classification. MRELBP stands out exceptionally clearly, performing very well (above 90 %) up to 30 % random pixel corruption, difficult noise levels where MRELBP offers strong performance, but where not a single other method delivers acceptable results.

Table 5. Classification scores (%) in the context of random salt and pepper noise with density ρ and randomly corrupted pixels. In the latter case we corrupted a certain percentage of randomly chosen pixels from each of the images, replacing their values with independent samples from a uniform distribution. The corrupted pixels are randomly chosen for each test image, with the locations unknown to the algorithm.

Robustness to		Salt and Pepper Noise										Random Corrupted Pixels									
Dataset		Outex_TC11s (24 classes)					Outex_TC23s (68 classes)					Outex_TC11c (24 classes)					Outex_TC23c (68 classes)				
Noise Parameter		Noise density ρ					Noise Density ρ					Percentage of corrupted pixels υ					Percentage of corrupted pixels υ				
No.	Method	5 %	15 %	30 %	40 %	50 %	5 %	15 %	30 %	40 %	50 %	5 %	10 %	20 %	30 %	40 %	5 %	10 %	20 %	30 %	40 %
1	MRELBP [37]	**100.0**	**100.0**	**100.0**	**85.8**	**50.2**	**100.0**	**99.9**	**94.0**	**54.6**	**19.2**	**100.0**	**100.0**	**100.0**	**99.6**	**90.6**	**99.6**	**99.2**	**96.9**	**89.8**	**57.5**
2	CLBP [33]	17.3	8.3	4.2	4.2	4.2	7.6	2.9	1.5	1.6	1.5	61.9	26.5	4.2	4.2	4.2	28.7	5.7	2.9	3.0	2.9
3	ELBP [35]	40.4	4.6	4.2	4.2	4.2	16.2	1.5	1.5	1.5	1.5	60.6	31.9	13.1	7.1	4.2	25.4	11.3	4.4	2.9	2.9
4	CLBPHF [39]	14.4	4.2	4.2	4.2	4.2	6.0	2.9	1.6	1.4	1.5	50.0	11.9	4.2	4.2	4.2	23.2	3.4	1.5	0.3	1.5
5	*dis*CLBP [34]	11.0	8.3	4.2	4.2	4.2	5.1	4.4	1.5	1.5	1.5	57.5	24.6	4.2	4.2	4.2	21.5	5.8	2.6	1.5	1.5
6	LTP^{riu2} [25]	9.0	6.3	8.3	8.3	8.3	4.3	3.5	2.9	2.9	2.9	60.0	23.3	4.4	4.2	4.2	21.1	6.6	2.9	2.6	2.9
7	BRINT [36]	30.8	7.1	6.0	4.4	4.2	15.9	1.5	1.5	1.3	1.5	89.0	53.5	17.5	7.3	4.2	62.1	20.4	1.6	1.5	1.5
8	LBP^{riu2} [5]	31.7	4.2	4.2	4.4	4.2	11.8	1.5	1.5	1.5	1.5	51.5	8.3	4.2	4.2	4.2	21.3	6.0	1.5	1.5	1.5
9	NELBP [27]	27.3	4.2	4.2	4.2	4.2	12.2	1.5	1.5	1.5	1.5	51.5	11.3	4.2	4.2	4.2	25.3	5.6	1.5	1.5	1.5
10	MSJLBP [32]	14.2	8.3	4.4	4.2	4.2	7.1	3.5	1.5	1.5	1.5	32.3	16.7	7.5	4.2	4.2	14.4	5.6	2.7	1.7	1.2
11	NTLBP [29]	74.4	22.1	4.8	5.0	6.3	40.5	4.7	3.8	2.6	2.7	82.5	45.6	11.9	4.2	4.2	49.8	22.9	3.7	1.5	1.5
12	PRICoLBP$_g$ [31]	9.6	5.2	4.2	4.2	4.2	4.2	2.1	1.5	1.5	1.5	31.7	10.0	4.2	4.2	4.2	9.0	3.0	1.5	1.5	1.5
13	LBPV^{riu2} [28]	4.6	4.2	4.2	4.2	4.2	2.8	1.5	1.5	1.5	1.5	17.7	5.0	4.2	4.2	4.2	4.0	1.5	1.5	1.5	1.5
14	RILPQ [43]	15.0	4.2	4.2	4.2	4.2	3.2	1.5	1.5	1.5	1.5	62.7	37.7	11.7	7.5	5.4	27.9	8.7	2.6	2.0	1.6
15	LBPD [41]	25.2	8.3	4.2	4.2	4.2	10.3	2.9	1.5	1.5	0.1	32.3	18.1	7.1	4.2	4.2	12.5	6.5	2.6	1.5	1.5
16	NRLBP^{riu2} [26]	8.8	8.1	8.3	4.2	5.0	2.1	2.9	2.6	1.5	1.5	72.5	41.3	21.9	12.7	7.7	25.1	10.1	5.3	2.6	1.5
17	LEP [40]	14.0	5.0	4.6	4.0	4.2	10.1	1.8	1.7	1.4	1.5	86.5	64.0	24.2	12.3	7.1	65.8	28.4	5.6	2.7	1.5
18	MBP^{riu2} [46]	31.0	8.3	4.2	4.2	4.2	17.0	2.5	1.5	1.5	1.5	45.0	18.8	8.1	4.2	4.2	23.5	8.0	2.6	1.6	1.5
19	FV-VGGVD (SVM) [13]	21.0	12.1	6.0	6.5	4.2	10.3	5.2	2.3	1.5	1.8	63.5	51.5	23.1	11.7	10.0	34.3	19.1	9.5	4.4	2.8
20	FV-VGGM (SVM) [13]	6.2	5.6	3.1	4.2	3.8	2.	1.5	3.5	1.8	2.1	34.4	15.8	12.1	9.2	13.3	10.4	7.8	4.9	5.2	5.0
21	ScatNet (PCA) [16]	4.6	4.2	4.2	4.2	4.2	1.4	1.5	1.5	1.5	1.5	29.2	12.1	5.2	4.2	4.2	3.5	1.5	1.5	1.5	1.5
22	FV-AlexNet (SVM) [13]	10.4	6.7	6.6	4.2	4.2	2.8	5.0	4.3	3.1	1.5	44.8	29.2	11.2	9.0	6.0	13.5	7.0	8.6	4.1	2.8
23	ScatNet (NNC) [16]	4.2	4.2	4.2	4.2	4.2	1.5	1.5	1.5	1.5	1.5	56.0	9.8	4.2	4.2	4.2	14.6	3.2	2.9	1.7	1.5
24	PCANet [17]	14.6	6.7	4.4	4.4	4.8	1.6	1.5	1.5	1.5	1.5	70.6	32.5	11.9	10.4	7.7	19.1	5.1	1.5	1.5	1.5
25	PCANet^{riu2} [17]	4.8	4.4	4.6	4.6	4.0	2.4	1.5	1.5	1.5	1.5	49.2	20.2	6.5	5.4	4.8	19.9	6.2	2.6	1.4	1.5
26	RandNet [17]	4.8	4.2	4.2	4.2	4.2	1.5	1.5	1.5	1.5	1.4	10.7	4.9	4.2	4.2	4.2	3.4	1.7	1.5	1.5	1.5
27	RandNet^{riu2} [17]	4.2	4.2	4.2	4.2	4.3	1.5	1.5	1.5	1.5	1.5	11.5	4.2	4.0	4.7	5.5	2.6	1.7	1.5	1.6	1.5

4.3 Computational Complexity

Feature extraction time and dimensionality (Table 3) are two key factors determining the computational cost of LBP methods. The stated computation times are the average time spent by each method to generate its multiscale features. All of the methods were implemented in MATLAB 2010b on 2.9 GHz Intel quad core CPU with 16 GB RAM. The feature extraction time was measured as the average over 480 images of size 128 × 128. Note that the reported time does not include the training time for those methods labeled with ($\star$) in Table 3. The reported feature dimensionality is the final dimensionality of each method given to the NNC classifier.

ScatNet is the most computationally expensive method for feature extraction, followed by FV-VGGVD. Its time cost for feature extraction is 125 times that of LBP^{riu2} and 26 times of MRELBP. Compared with LBP^{riu2}, most of the remaining methods do not introduce much computation overhead at the feature extraction stage. In terms of feature dimensionality, FV-CNN is extreme,

Table 6. Summary of various LBP methods used in our experimental study. Different schemes for parameters (r, p) are defined. Scheme 1: $(1, 8)$, $(2, 16)$, $(r, 24)$ for $3 \leq r \leq 9$; Scheme 2: $(r, 8), r = 1, \cdots, 9$; Scheme 3: $(1, 8)$, $(r, 24)$ for $2 \leq r \leq 9$; Scheme 4: $(2, 8)$; Scheme 5: $(1, 8)$, $(3, 8)$ and $(5, 8)$; Scheme 6: $(r, 8), r = 2, 4, 6, 8$. "Partial" in the "Noise Robust?" column means "robust to random Gaussian white noise and blur but highly sensitive to salt and pepper and random pixel corruption". Those with ($\star$) in the "Optimal Operator Size" column represent the size of the receptive field, meaning much larger input image size is required. In the "Relative Performance" column, we consider the classification performance of LBP as baseline and use ★ and X to represent better and worse than baseline respectively.

No.	Method	(r, p) Scheme	Encoding Scheme	Needs Training?	Optimal Operator Size	Feature Extraction	Feature Dimension	Noise Robust?	Rotation Invariant?	Monotonic Illumination Invariant?	Relative Performance
1	LBP^{riu2}	Scheme 1	*riu2*		19 × 19	Very fast	210	No	Yes	Yes	Baseline
2	MRELBP	Scheme 6	*riu2*		17 × 17	Fast	800	Yes	Yes	Yes	★★★
3	CLBP	Scheme 1	*riu2*		9 × 9	Fast	3552	No	Yes	Yes	★★
4	ELBP	Scheme 1	*riu2*		7 × 7	Fast	2200	No	Yes	Yes	★★
5	CLBPHF	Scheme 1	*u2*		19 × 19	Fast	4580	Partial	Yes	Yes	★★
6	*dis*CLBP	Scheme 1	*Reported*	√	11 × 11	Moderate	7796	No	Yes	Yes	★
7	LTP^{riu2}	Scheme 1	*riu2*		19 × 19	Fast	420	No	Yes	No	★
8	BRINT	Scheme 3	*ri*		19 × 19	Fast	1296	Partial	Yes	Yes	★
9	NELBP	Scheme 1	*Reported*		19 × 19	Very fast	273	No	Yes	Yes	Similar
10	MSJLBP	Scheme 5	*Reported*		7 × 7	Moderate	3540	No	Somewhat	Yes	X
11	NTLBP	Scheme 1	*Reported*		17 × 17	Fast	388	No	Yes	Yes	X
12	PRICoLBP$_g$	Scheme 4	*Reported*		13 × 13	Fast	3540	No	Somewhat	Yes	X
13	LBPV$^{riu2}_{r,p}$	Scheme 1	$riu2$		15 × 15	Moderate	158	No	Yes	Yes	X
14	RILPQ	PreFiltering	*Original*		13 × 13	Fast	256	Partial	Yes	Yes	★
15	LBPD	PreFiltering	*Original*		7 × 7	Fast	289	Partial	Yes	Yes	X
16	NRLBP^{riu2}	Scheme 2	*riu2*		11 × 11	Fast	50	No	Yes	No	XX
17	LEP	PreFiltering	*ri*	√	32 × 32	Fast	520	Partial	No	No	XX
18	MBP^{riu2}	Scheme 1	*riu2*		19 × 19	Fast	420	No	Yes	No	XX
19	PCANet	Multistage filtering, binarizing	*Original*	√	5 × 5	Moderate	2048	Partial	No	No	XXX
20	PCANet^{riu2}		*riu2*	√	5 × 5	Moderate	80	Partial	No	No	XXX
21	RandNet		*Original*		5 × 5	Moderate	2048	No	No	No	XXX
22	RandNet^{riu2}		*riu2*		5 × 5	Moderate	80	No	No	No	XXX
23	ScatNet	Repeating filtering, nonlinear, pooling	N/A		32 × 32	Very slow	596	Partial	Yes	Yes	★★
24	AlexNet+FV		N/A	√	163 × 163($\star$)	Moderate	32768	Partial	No	No	★★★
25	VGG-M+FV		N/A	√	139 × 139($\star$)	Moderate	65536	Partial	No	No	★★★
26	VGG-VD+FV		N/A	√	252 × 252($\star$)	Slow	65536	Partial	No	No	★★★

with the dimensionality of *dis*CLBP, CLBPHF, CLBP, PRICoLBP, MSJLBP, PCANet and RandNet also relatively high.

We provide Table 6 to summarize the properties of all evaluated methods including recommended operator size, feature dimensionality, robustness to image variations, tolerance of image noise and computational complexity. In order to establish a common software platform and a collection of datasets for easy evaluation, we plan to make both the source code and datasets available online.

5 Conclusions

A total of 27 methods were applied to 14 datasets, designed to test and stress an exceptional range of class types, image sizes, and disturbance invariance. The best overall performance is obtained for the MRELBP when distinctiveness, robustness and computational complexity are all taken into consideration. If the textures have very large within-class appearance variations, the FV-CNN methods clearly perform the best, however at a cost of high computational complexity. The problem of very high computational complexity should be solved to make them useful, especially in real-time embedded systems with low-power constraints. Furthermore, excellent results are obtained with FV-CNN for most test sets, but lack some robustness to noise and rotations. The role of FV is important and should be considered also with LBP methods in future studies.

In general, both micro- and macro-structures are important for texture description, since most LBP variants achieve their best performance beyond three scales, and a combination of multiple complementary texture descriptors turns out to be more powerful than a single descriptor. In general, LBP noise robustness improves when a prefiltering step is involved; however it does not necessarily guarantee good discriminability (*e.g.* LEP) and robustness to other noise types (*e.g.* salt and pepper).

It is possible that a classic CNN network could learn how to explore the properties of textured images more efficiently when trained on a very large texture dataset (similar to *ImageNet*). Unfortunately, to the best of our knowledge, such a database does not exist. We believe that a truly important question is to determine what makes a ***good large scale*** texture dataset. We have started to build such a dataset.

Based on our study, the work on CNNs for texture recognition mainly focuses on the domain transferability of CNNs. For texture, it is possible that simple networks might be enough to achieve similar or better results on texture datasets. Instead of devoting to design more and more complex networks, we feel that designing simple and efficient networks is important for problems such as mobile computing. Therefore, in the future, it would also be of great interest to study how to utilize effective LBP type computations with deep learning architectures.

Acknowledgments. This work has been supported by the National Natural Science Foundation of China under contract number 61202336 and by the Open Project Program of the National Laboratory of Pattern Recognition.

References

1. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. Int. J. Comput. Vis. **73**(2), 213–238 (2007)
2. Pietikäinen, M., Hadid, A., Zhao, G., Ahonen, T.: Computer Vision Using Local Binary Patterns. Springer, London (2011)
3. Julesz, B., Bergen, J.: Textons, the fundamental elements in preattentive vision and perception of textures. Bell Syst. Tech. J. **62**(6), 1619–1645 (1983)
4. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. Int. J. Comput. Vis. **43**(1), 29–44 (2001)
5. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
6. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. Int. J. Comput. Vis. **62**(1–2), 61–81 (2005)
7. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1265–1278 (2005)
8. Varma, M., Zisserman, A.: A statistical approach to material classification using image patches. IEEE Trans. Pattern Anal. Mach. Intell. **31**(11), 2032–2047 (2009)
9. Liu, L., Fieguth, P.: Texture classification from random features. IEEE Trans. Pattern Anal. Mach. Intell. **34**(3), 574–586 (2012)
10. Sanchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the fisher vector: theory and practice. Int. J. Comput. Vis. **105**(3), 222–245 (2013)
11. Manjunath, B., Ma, W.: Texture features for browsing and retrieval of image data. IEEE Trans. Pattern Anal. Mach. Intell. **18**(8), 837–842 (1996)
12. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15561-1_11
13. Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3828–3836, June 2015
14. Cimpoi, M., Maji, S., Kokkinos, I., Vedaldi, A.: Deep filter banks for texture recognition, description, and segmentation. Int. J. Comput. Vis. **118**(1), 65–94 (2016)
15. Sifre, L., Mallat, S.: Combined scattering for rotation invariant texture analysis. In: Proceedings of the European Symposium Artificial Neural Networks, April 2012
16. Bruna, J., Mallat, S.: Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1872–1886 (2013)
17. Chan, T., Jia, K., Gao, S., Lu, J., Zeng, Z., Ma, Y.: PCANet: a simple deep learning baseline for image classification? IEEE Trans. Image Process. **24**(12), 5017–5032 (2015)
18. Fernández, A., Álvarez, M., Bianconi, F.: Texture description through histograms of equivalent patterns. J. Math. Imaging Vis. **45**(1), 76–102 (2013)
19. Xu, Y., Yang, X., Ling, H., Ji, H.: A new texture descriptor using multifractal analysis in multiorientation wavelet pyramid. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 161–168 (2010)
20. Mallikarjuna, P., Fritz, M., Targhi, A., Hayman, E., Caputo, B., Eklundh, J.O.: The kth-tips and kth-tips2 databases. http://www.nada.kth.se/cvap/databases/kth-tips/

21. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014
22. Burghouts, G., Geusebroek, J.: Material specific adaptation of color invariant features. Pattern Recogn. Lett. **30**(3), 306–313 (2009)
23. Ojala, T., Mäenpää, T., Pietikänen, M., Viertola, J., Kyllonen, J., Huovinen, S.: Outex-new framework for empirical evaluation of texture analysis algorithms. In: Proceedings of the 16th International Conference Pattern Recognition, pp. 701–706 (2002)
24. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on feature distributions. Pattern Recogn. **29**(1), 51–59 (1996)
25. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans. Image Process. **19**(6), 1635–1650 (2010)
26. Ren, J., Jiang, X., Yuan, J.: Noise-resistant local binary pattern with an embedded error-correction mechanism. IEEE Trans. Image Process. **22**(10), 4049–4060 (2013)
27. Zhou, H., Wang, R., Wang, C.: A novel extended local-binary-pattern operator for texture analysis. Inform. Sci. **178**(22), 4314–4325 (2008)
28. Guo, Z., Zhang, L., Zhang, D.: Rotation invariant texture classification using LBP variance (LBPV) with global matching. Pattern Recognit. **43**(3), 706–719 (2010)
29. Fathi, A., Naghsh-Nilchi, A.: Noise tolerant local binary pattern operator for efficient texture analysis. Pattern Recognit. Lett. **33**(9), 1093–1100 (2012)
30. Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **3**(6), 610–621 (1973)
31. Qi, X., Xiao, R., Li, C., Qiao, Y., Guo, J., Tang, X.: Pairwise rotation invariant co-occurrence local binary pattern. IEEE Trans. Pattern Anal. Mach. Intell. **36**(11), 2199–2213 (2014)
32. Qi, X., Qiao, Y., Li, C., Guo, J.J.: Multiscale joint encoding of local binary patterns for texture and material classification. In: Proceedings of British Machine Vision Conference (BMVC) (2013)
33. Guo, Z., Zhang, L., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. IEEE Trans. Image Process. **9**(16), 1657–1663 (2010)
34. Guo, Y., Zhao, G., Pietikäinen, M.: Discriminative features for texture description. Pattern Recognit. **45**(10), 3834–3843 (2012)
35. Liu, L., Zhao, L., Long, Y., Kuang, G., Fieguth, P.: Extended local binary patterns for texture classification. Image Vis. Comput. **30**(2), 86–99 (2012)
36. Liu, L., Long, Y., Fieguth, P., Lao, S., Zhao, G.: BRINT: binary rotation invariant and noise tolerant texture classification. IEEE Trans. Image Process. **23**(7), 3071–3084 (2014)
37. Liu, L., Lao, S., Fieguth, P., Guo, Y., Wang, X., Pietikainen, M.: Median robust extended local binary pattern for texture classification. IEEE Trans. Image Process. **25**(3), 1368–1381 (2016)
38. Ahonen, T., Matas, J., He, C., Pietikäinen, M.: Rotation invariant image description with local binary pattern histogram fourier features. In: Salberg, A.-B., Hardeberg, J.Y., Jenssen, R. (eds.) SCIA 2009. LNCS, vol. 5575, pp. 61–70. Springer, Heidelberg (2009)
39. Zhao, Y., Ahonen, T., Matas, J., Pietikäinen, M.: Rotation invariant image and video description with local binary pattern features. IEEE Trans. Image Process. **21**(4), 1465–1477 (2012)

40. Zhang, J., Liang, J., Zhao, H.: Local energy pattern for texture classification using self-adaptive quantization thresholds. IEEE Trans. Image Process. **22**(1), 31–42 (2013)
41. Hong, X., Zhao, G., Pietikainen, M., Chen, X.: Combining LBP difference and feature correlation for texture description. IEEE Trans. Image Process. **23**(6), 2557–2568 (2014)
42. Ojansivu, V., Heikkilä, J.: Blur insensitive texture classification using local phase quantization. In: Elmoataz, A., Lezoray, O., Nouboud, F., Mammass, D. (eds.) ICISP 2008. LNCS, vol. 5099, pp. 236–243. Springer, Heidelberg (2008). doi:10.1007/978-3-540-69905-7_27
43. Ojansivu, V., Rahtu, E., Heikkilä, J.: Rotation invariant local phase quantization for blur insensitive texture analysis. In: IEEE International Conference on Pattern Recognition (ICPR), pp. 1–4 (2008)
44. Krizhevsky, A., Ilya, S., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
45. Sifre, L., Mallat, S.: Rotation, scaling and deformation invariant scattering for texture discrimination. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1233–1240, June 2013
46. Hafiane, A., Seetharaman, G., Zavidovique, B.: Median binary pattern for textures classification. In: Proceedings of the 4th International Conference on Image Analysis and Recognition, pp. 387–398 (2007)
47. Brodatz, P.: Textures: A Photographic Album for Artists and Designers. Dover Publications, New York (1966)
48. Liu, L., Fieguth, P., Kuang, G., Clausi, D.: Sorted random projections for robust rotation invariant texture classification. Pattern Recogn. **45**(6), 2405–2418 (2012)
49. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2009)
50. Caputo, B., Hayman, E., Mallikarjuna, P.: Class-specific material categorization. In: Internaltional Conference on Computer Vision (ICCV), pp. 1597–1604 (2005)