

RLBP: Robust Local Binary Pattern

Jie Chen

Vili Kellokumpu

Guoying Zhao

Matti Pietikäinen

Center for Machine Vision Research
University of Oulu
Finland

Abstract

In this paper, we propose a simple and robust local descriptor, called the robust local binary pattern (RLBP). The local binary pattern (LBP) works very successfully in many domains, such as texture classification, human detection and face recognition. However, an issue of LBP is that it is not so robust to the noise present in the image. We improve the robustness of LBP by changing the coding bit of LBP. Experimental results on the Brodatz and UIUC texture databases show that RLBP impressively outperforms the other widely used descriptors (e.g., SIFT, Gabor, MR8 and LBP) and other variants of LBP (e.g., completed LBP), especially when we add noise in the images. In addition, experimental results on human face recognition also show a promising performance comparable to the best known results on the Face Recognition Grand Challenge (FRGC) face dataset.

1 Introduction

Recently, many sparse and dense descriptors (e.g., SIFT, Gabor, MR8 and LBP) have been proposed for different kinds of applications. There are several studies to evaluate their performance, e.g., [13, 14]. LBP [15] is perhaps the best performing dense descriptor and it has been widely used in various applications, such as texture classification, human detection and face recognition [18]. It has been proven to be highly discriminative and its key advantages, namely its invariance to monotonic gray level changes and computational efficiency, make it suitable for demanding image analysis tasks.

However, one issue of LBP is that it is not so robust to the noise present in images when the gray-level changes resulting from the noise are not monotonic, even if the changes are not significant [2]. To this end, we propose a new descriptor based on LBP, i.e., robust local binary pattern (RLBP). The idea is to locate the possible bit in LBP pattern changed by the noise and then revise the changed bit of the LBP pattern. The idea is very simple, but it works very well. For example, the performance of LBP decreases significantly when we add white Gauss noise in the Brodatz texture dataset [1]. However, the performance of RLBP almost does not change. We also add noise in UIUC texture [7] and FRGC face datasets [17] to testify the performance of RLBP.

1.1 Related works

Recently, many different image descriptors have been proposed. For example, Lowe introduced the sparse scale-invariant feature transform (SIFT) descriptor [11], which performs elegantly [13]. Several attempts to improve SIFT have been reported [3, 7, 10, 13, 19].

A highly popular dense image descriptor is LBP [18]. Many variants of it have been proposed recently, achieving considerable success in various tasks. Ahonen et al. exploited the LBP for face recognition [18], and Tan and Triggs proposed local ternary patterns (LTP) [21]. Zhao and Pietikäinen introduced the spatiotemporal LBP [24]. Liao et al. proposed the dominant local binary pattern (DLBP) [8], and Guo et al. proposed a completed modeling of the local binary pattern (CLBP) operator [4].

The rest is organized as follows: in Section 2, we present the details of RLBP. In Sections 3 and 4, we carry out the experiments. Section 5 concludes the paper.

2 RLBP for Image Representation

In this section, we will first review the LBP briefly and then introduce the proposed RLBP.

2.1 Local binary pattern

The basic form of LBP is illustrated in Fig. 1 (a) and (b) [15]. Specifically, we use a local neighborhood around each pixel as input (see Fig. 1 (a)) and then threshold the neighborhood pixels at the value of the central pixel. The resulting binary-valued string is weighted as follows:

$$LBP(I_c) = \sum_{i=0}^{P-1} 2^i s(I_i - I_c), \quad (1)$$

where the parameter P runs over the eight neighbors of the central pixel I_c . I_c and I_i are the gray-level values at c and i , and $s(A)$ is 1 if $A \geq 0$ and 0 otherwise.

One extension of the original LBP is the *uniform patterns*: an LBP is ‘uniform’ if it contains at most two 0-1 or 1-0 transition when viewed it as a circular bit string (e.g., 11110011 is a uniform pattern). The final texture feature employed in analysis is the histogram of the operator outputs (i.e., pattern labels) accumulated over a texture sample. In general, the histogram of uniform patterns provides better discrimination in comparison to the histogram of all individual patterns [15].

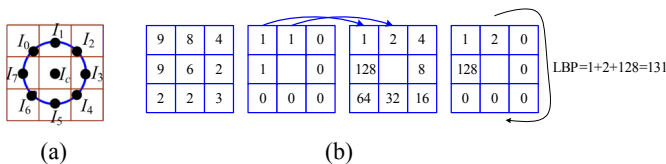


Fig. 1. LBP. (a) A pixel and its eight neighbors; (b) Basic LBP.

Let us consider the LBP in the case that $P=8$ and $R=1$, i.e., LBP(8,1), where parameter P controls the quantization of the angular space, and R determines the spatial resolution of the operator. Formally, given an image I , denote its histogram of LBP(8,1) to be H , which has 256 bins before mapping it to a uniform pattern. Let H^{ui} be the uniform pattern histogram. We have the mapping function:

$$H^{ui} = \varphi(H), \tag{2}$$

which computes the uniform pattern histogram H^{ui} for H .

Let H_i be the i -th bin of H ($i = 0, \dots, 255$) and H_k^{ui} be the k -th bin of H^{ui} ($k = 0, \dots, 58$). Let $|H_i|$ be the occurrence frequency of H_i . We have the detailed form of function $\varphi(H)$:

If the LBP pattern corresponding to H_i is a uniform pattern,

$$|H_k^{ui}| = |H_i|, \quad k=k+1 \text{ and } i=i+1; \tag{3}$$

otherwise, $|H_0^{ui}| = |H_0^{ui}| + |H_i|, \quad i=i+1$.

2.2 Robust local binary pattern

An example of how to compute RLBP from LBP is shown in Fig. 2. The neighbors of the pixel x_c would give the LBP pattern string (11010011). However, the pixel value of $x_2=124$ here is of high probability of being noisy since it results in a (101) substring (see following for details). If we change the corresponding bit of x_2 in LBP string from 0 to 1, the new LBP string of this pixel (11110011) would denote a local corner, which is a more meaningful pattern for the texture representation and classification.

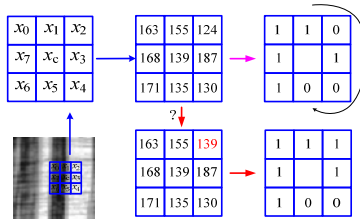


Fig. 2. Robust local binary pattern

Let us consider the LBP in the case that $P=8$ and $R=1$, i.e., LBP(8,1). In general, any neighboring three-bit substring in an eight-bit LBP pattern string is one in the set $Y = \{y_1=(000), y_2=(001), y_3=(010), y_4=(011), y_5=(100), y_6=(101), y_7=(110), y_8=(111)\}$. Following the idea of the example in Fig. 2, we assume that the cases of y_3 and y_6 are noisy and change them to a new sub-string: $y'_3=(000)$, and $y'_6=(111)$.

Formally, given an image I , let its histogram of LBP(8,1) be H , which has 256 bins before mapping it to a uniform pattern. For each bin H_i of H ($i = 0, \dots, 255$), we search all of its neighboring three-bit substrings, and map its y_3 or y_6 to y'_3 or y'_6 , respectively. We denote the mapping function as follows:

$$RH_s = \Psi(H_i), \tag{4}$$

where RH is the resulted histogram of RLBP, composed of uniform patterns, and RH_s is the s -bin, $s \in \{1, \dots, 58\}$. Let $|RH_s|$ be the occurrence frequency of RH_s . The occurrence frequency of RH_s is then changed to $|RH_s| + |H_i|$ after this mapping.

Note that the mapping function $\Psi(H_i)$ is not a one-to-one mapping. Specially, one bin H_i would map to more than one bin (for example, the LBP string 11010011 in Fig. 2 can be mapped to 11110011 or 11000011). Let T_i denote that the number of bins that H_i maps to those bins in RH by Eq. (4). Thus, Eq. (4) is revised as follows:

$$\{RH_{st} | t=1, \dots, T_i\} = \Psi(H_i), \tag{5}$$

and the new occurrence frequency of RH_{st} is computed as

$$|RH_{st}| = |RH_{st}| + \frac{1}{T_i} |H_i|, t=1, \dots, T_i. \quad (6)$$

In our case, we also use the uniform patterns of LBP as in [15]. For LBP(8,1), we map the 256 bins of LBP(8,1) to 59 bins, denoting it as LBP^{u2}(8,1). The traditional lookup table [15], considering 256 bins and the uniform pattern model, maps all the non-uniform patterns to one “miscellaneous” bin, i.e., the collection bin H_0^u in Eq. (3). The occurrence frequencies of the non-uniform pattern bins increase when the noise in the image increases. Different from the method in [15], we re-consider the non-uniform patterns during the mapping from the original 256-bin histogram to uniform pattern histogram as in Eq. (6).

Note that some resulting bins RH_{st} by Eq. (5) might not be a uniform pattern. Assuming that there are T_{i1} patterns in uniform (i.e., $\{RH_{st}|t=1, \dots, T_{i1}\}$) and $T_{i2} = T_i - T_{i1}$ patterns not in uniform (i.e., $\{RH_{st}|t=T_{i1}+1, \dots, T_i\}$), and we use RH_0 to accumulate the frequency of non-uniform patterns. Thus, Eq. (6) is revised as follows:

$$\begin{aligned} |RH_{st}| &= |RH_{st}| + \frac{1}{T_i} |H_i|, t=1, \dots, T_{i1} \\ |RH_0| &= |RH_0| + \frac{T_{i2}}{T_i} |H_i| \end{aligned} \quad (7)$$

Combining the Eqs. (5) and (7), we map a non-uniform pattern to one/some uniform pattern(s) and non-uniform pattern(s). By this way, we re-consider the non-uniform pattern which might be resulted from a uniform pattern due to the noise. Note that the input of the mapping function $\Psi(H_i)$ is non-uniform pattern, and for the uniform pattern we use the same rule as [15].

One risk for this method is that it would map a natural non-uniform pattern to a uniform pattern. However, it would not change the discrimination of the LBP^{u2} histogram. It is because all the non-uniform patterns in the full LBP(8,1) pattern (256 bins) are mapped to the same bin (i.e., H_0^u) in LBP^{u2}(8,1), and H_0^u is in general not discriminative for classification. In addition, the number of natural non-uniform patterns is limited [15].

The mapping function $\Psi(H_i)$ in Eq. (5) is the same for all the images. The mapping table can be computed beforehand. Thus, the RLBP is also computationally efficient compared to the traditional LBP.

2.3 Motivation and statistical evidence

In this section, we will first discuss motivation for our method, and then show statistical evidence. In our case, we only map the two patterns, i.e., y_3 and y_6 . Our motivation for RLBP is to develop a descriptor to be robust to point noise (e.g., Gaussian, salt & pepper or Rayleigh noise). According to the statistical results [15], the neighboring pixels in an image should be smooth (e.g., LBP=00000000) or disturbed by edges (e.g., LBP=00001111). If one pixel is a noise, the bit in LBP pattern might be changed, e.g., LBP=00001101, where the seventh bit is different from its two neighboring bits, i.e., we have a substring $y_6 = (101)$.

However, for other patterns, e.g., $y_2 = (001)$, the middle bit takes the same bit value as one of its neighboring bits at least, which might describe an edge (e.g., LBP=00001111). In addition, we only mapped those which were non-uniform patterns. In [15], all the non-uniform patterns are mapped to the same bin when building the histogram for LBP. By our method, we improve the discriminability of LBP.

We will then show some statistical evidence. For the validation dataset, four different

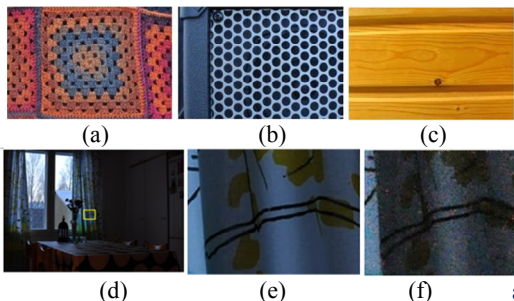


Fig. 3. Illustration of the textures. (a) - (d) illustrates the textures. (e) is a patch cropped from (d) with ISO 100 and (f) is a patch cropped from (d) with ISO 6400.

views are captured with Nikon D90 DSLR mounted on a tripod. The noise levels were affected using different ISO boost settings. Using a low ISO level results in a lower noise image, while increasing the ISO level results in images with more noise. Each view was captured using four different ISO boost levels (ISO 100, ISO 400, ISO 1600 and ISO 6400). Five images were captured for each ISO setting resulting in four views in four ISO levels. Thus, we got 80 images in total. We used uncompressed images with full resolution (4288×2848). Some examples are shown in Fig. 3.

The goal is to test how noise affects the uniform patterns (u2) of LBP. The hypothesis is that when the noise level increases, the amount of uniform patterns decreases as more patterns are mapped to the collection bin. The idea of RLBP is to map noisy patterns into uniform patterns and the hypothesis is that RLBP will be less affected by the noise.

Fig. 4 shows how the ratio of the uniform patterns in $LBP^{u2}(8,1)$ clearly decreases when the noise level increases. Here, the ratio of uniform patterns is the quotient by dividing the number of uniform patterns by the number of all LBP patterns of one image. Also it is obvious that the RLBP representation is less sensitive to noise. It is clear that the results support the hypothesis. In addition, from Fig. 4, we can find that after mapping the two patterns, i.e., y_3 and y_6 , the ratio of uniform patterns in RLBP is close to 1. It means that these two patterns take up most of the patterns in $LBP^{u2}(8,1)$ which are changed from uniform patterns to non-uniform ones due to the noise.

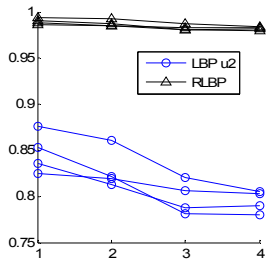


Fig. 4. The ratio of uniform patterns observed as a function of noise level for the four textures in Fig. 3 (a)- (d). For $LBP(8,1)$. Y-axis is the average of each noise level; x-axis shows the different ISO, i.e., 1=ISO100, 2=ISO400, 3=ISO1600, 4=ISO6400.

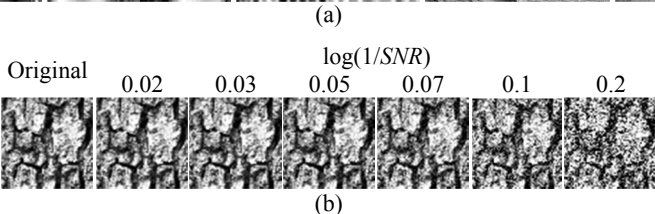
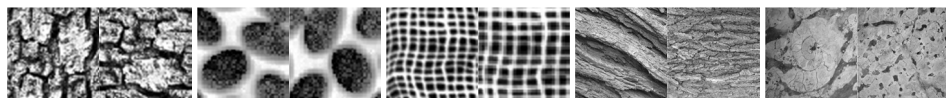


Fig. 5. (a) Example images from Brodatz and UIUC texture datasets. (b) An example from Brodatz by adding white Gaussian noise.

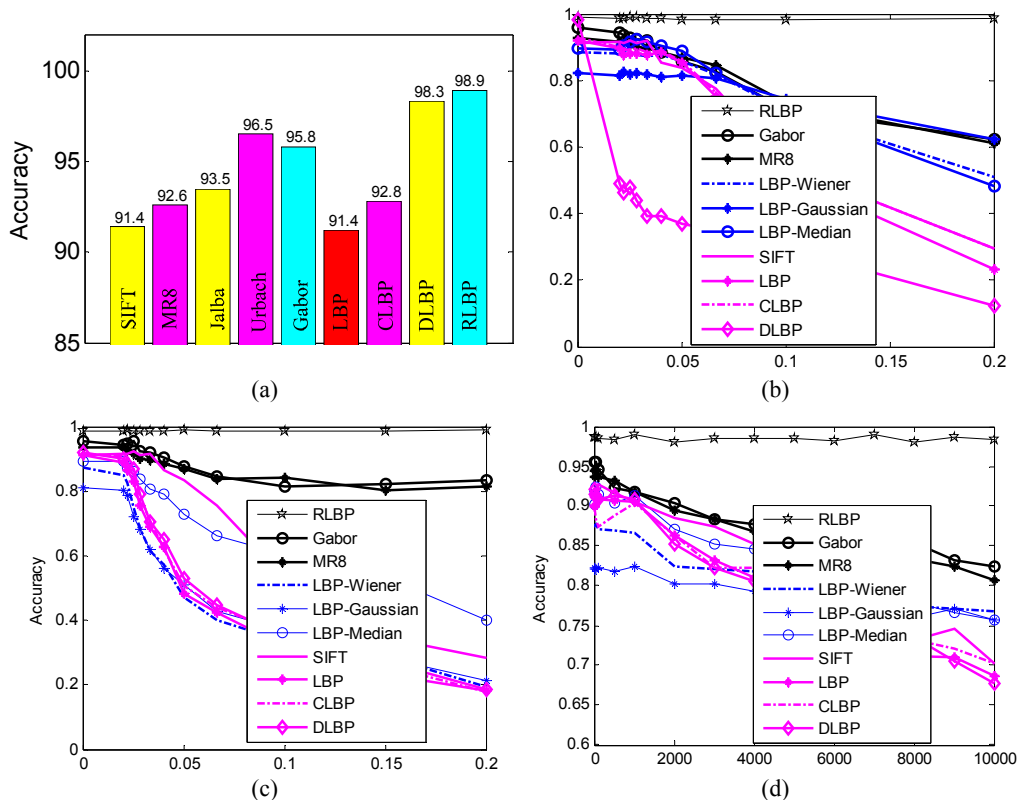


Fig. 6. Performance comparison with existing methods over Brodatz textures, (a) the original textures, (b) white Gaussian noise. (c) Salt & Pepper noise and (d) Rayleigh noise.

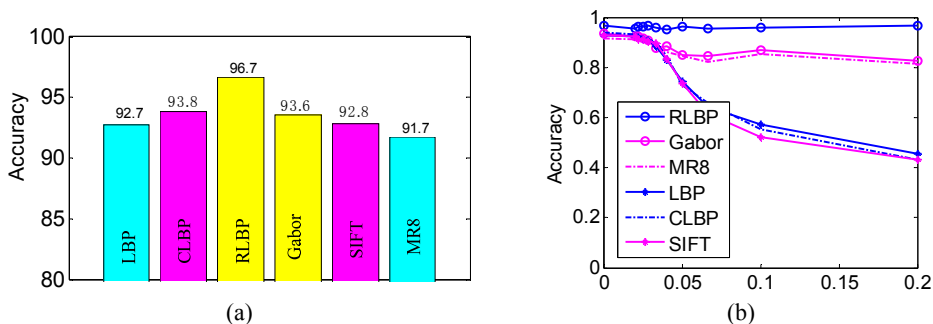


Fig. 7. Performance comparison with existing methods over UIUC textures, (a) using the original textures, (b) using the textures added with white Gaussian noise.

3 Application to Texture Classification

In this section, we use RLBP for texture classification, which plays an important role in many applications, such as robot vision, content-based access to image databases and automatic tissue recognition in medical images [12, 15, 18].

3.1 Dataset and set-ups

Experiments are carried out on two different texture databases: Brodatz [1] and UIUC [7]. Example images are shown in Fig. 5. For Brodatz32 [16], it comprises 2,048 samples, with 64 samples in each of the 32 texture categories [6, 16, 22]. UIUC has 25 classes with images under uncontrolled illumination, albedo variations, 3D shape, as well as a mixture of both. Note that we use different evaluation setups for the Brodatz and UIUC texture databases. Specifically, for the Brodatz textures we use half of the samples in each class for training and half for testing as in [6, 16, 22]. For UIUC, we use ten samples for training and the rest for testing as in [7]. We repeat these experiments ten times and report the average values.

We use the RLBP histogram feature as a representation and build a system for texture classification. As the classifier we use the K -nearest neighbour method, which has been successfully utilized in classification. In our case, $K=1$. To compute the distance between two given images I_1 and I_2 , we first obtain their RLBP histogram features H_1 and H_2 . We then measure the similarity between H_1 and H_2 by the normalized histogram intersection.

3.2 Experimental Results

Experimental results on Brodatz textures are illustrated in Fig. 6. In Fig. 6 (a), we compare our method with others on the classification task of Brodatz textures: SIFT [11], MR8 [23], Jalba [6], Urbach [22], and Manjunath [12] (i.e., Gabor) and LBP [15] and its variants, i.e., CLBP [4], DLBP[8]. Note that the results of [6] and [22] from other methods are quoted directly from the original papers. Other methods are re-implemented by us following the ideas of these papers. In addition, LBP in the following parts is referred to LBP^{u2} for short.

Results on Brodatz dataset

From Fig. 6 (a), one can find that RLBP works in a very robust way in comparison to other methods. The performance of SIFT might suffer from the image size (64×64), by which the sparse points of interest are not enough for SIFT to perform well.

Besides the performance comparison with other methods, we also carried out an experiment on the Brodatz dataset to compare the efficiency of RLBP with LBP and Gabor. The experiments are performed on a 3.10 GHz Intel Core i5-2400 processor using 8GB RAM by executing C/C++ code. To extract the features for an image of this dataset, the average time consumptions of LBP and RLBP are 0.6 milliseconds, while that of Gabor is 197 milliseconds. Note that the mapping table for RLBP is computed beforehand.

Results on Brodatz dataset with Noise

We test the robustness of RLBP by additive noise and compare its performance with existing methods. In Fig. 5 (b), we show some examples from Brodatz textures by adding white Gaussian noise. From this figure, when the value of $\log(1/SNR)$, i.e., the logarithm of the inverse of the signal-to-noise ratio (SNR), increases, the image quality degrades significantly. Especially when $\log(1/SNR)=0.2$, the image appearance changes significantly.

As can be seen from Fig. 6 (b), we compare the performances of RLBP, LBP, CLBP, DLBP, SIFT, Gabor and MR8 on the Brodatz textures with added white Gaussian noise. Here, x axis is $\log(1/SNR)$. From this figure, one can see that the performance of LBP decreases significantly when the noise strength is larger than 5%. However, the performance of RLBP changes little even when the noise strength is larger than 20%. For

the poor performance of DLBP when we added white Gaussian noise, a possible reason is that the noise may change the pattern type of a dominant pattern in the noise-free case. Therefore, the noise may possibly reshape the dominant pattern histogram in some cases by assigning some of the dominant pattern's occurrence to other types of patterns.

We also test the case by first applying a filtering and then applying LBP as shown in Fig. 6 (b). Here, we use three types of filters, i.e., Wiener, Gaussian and median filters. In Fig. 6 (c) (d), we compare the performances of RLBP with existing methods on the Brodatz textures with added salt & pepper noise and Rayleigh noise.

Results on UIUC dataset

As shown in Fig.7 (a), we compare the performance of RLBP and existing methods on the classification task of UIUC textures. Similarly, we also add white Gaussian noise on the UIUC textures as shown in Fig. 7(b). From this figure, one can find that RLBP gets very good results compared with other methods.

4 Application to Face Recognition

In this section, we use RLBP for human face recognition. Face recognition is an active research area because of both its scientific challenge and wide range of potential applications, such as human-computer interaction and biometric identity authentication. Numerous face recognition algorithms have been developed in the past two decades [25].

4.1 Dataset and setups

The dataset Face Recognition Grand Challenge (FRGC) version 2.0 is designed to promote face recognition in general with emphasis on 3D and high resolution still imagery. Meanwhile, there are six experimental protocols in FRGC and Experiments 1, 2 and 4 are designed for still images. There are 222 subjects showing 12,776 still images in the training set. For the classifier, we use Fisher Discriminant Analysis (FDA) and take Experiment 1 for evaluations. The Experiment 1 measures performance on 16,028 frontal facial images. These images are taken under controlled illumination, and both target and query set have the same number of samples (i.e., 16,028). The performance is reported as Verification Rates (VR) at 0.1% False Acceptance Rate (FAR). In our case, we only consider the still images. One example face from FRGC data set is shown in Fig.8.

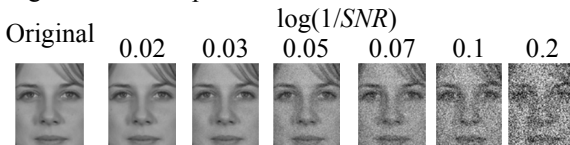


Fig. 8. An example from FRGC data set, with added white Gaussian noise.

For the face recognition framework, we use the similar method as in [20], in which Su et al. got very good results on FRGC. Specifically, they combined both global and local discriminative features for face representation and recognition. For the global features, they performed the Fourier transform for whole face images. For the local features, they performed the Gabor wavelet transform and then extracted some spatially partitioned image patch features. Different from [20], we use a different local feature, i.e., RLBP, to replace Gabor and keep the rest set-ups the same.

4.2 Experimental Results

The resulting final strong classifier combining the global and local FDA is tested on the FRGC Experiment 1. The experimental results are shown in Fig. 9 (a). Herein, we also compare the performance of the resulting classifier (we call it “*RLBP*”, and the same for “*LBP*” and “*CLBP*”) with some existing methods. Here, the results of Hwang [5], Liu [9] and Su [20] are quoted directly from the original papers. The results of LBP and CLBP are implemented by us. From this figure, one can find that *RLBP* gets the best results over this dataset. In addition, *LBP*, *CLBP* and Su [20] (i.e., Gabor) also achieve very good results.

Likewise, we also test the FRGC dataset set by adding white Gaussian noise. Some examples are shown in Fig. 8. The results of face recognition are plotted in Fig. 9 (b). One can find that *RLBP* works much better than the others, especially for the high level of noise images.

5 Conclusions

In this paper, we proposed a robust version of LBP (i.e., RLBP) by changing the coding of bits of LBP, which could otherwise be changed by noise. The experimental results are very promising although the idea is simple. Specifically, the experimental results over Brodatz texture dataset and the noisy Brodatz textures show perfect results. For example, RLBP achieves the accuracy of 98.9% for the dataset without noise and 98.6% with high levels of noise, while LBP gets the accuracy of 91.4% for the dataset without noise and 23.1% with high levels of noise. Experimental results on the UIUC texture database also show that RLBP impressively outperforms LBP. In addition, experimental results on FRGC face recognition also show a promising performance comparable to the best known results and reasonably good results for the dataset with added white Gaussian noise. For example, when $\log(1/SNR) = 0.04$, *RLBP* achieves the accuracy of 70% compared to 63% of Gabor, 48% of CLBP and 44% of LBP.

An interesting future work is to develop a multiscale RLBP because for the LBP when $P=16$ and $R=2$, and $P=24$ and $R=3$, the percentages of the uniform patterns are statistically about 66.9% and 54%, respectively, while it is 87% when $P=8$ and $R=1$. Another future work is to employ RLBP for depth images captured, e.g. by Kinect which tend to contain lots of noise.

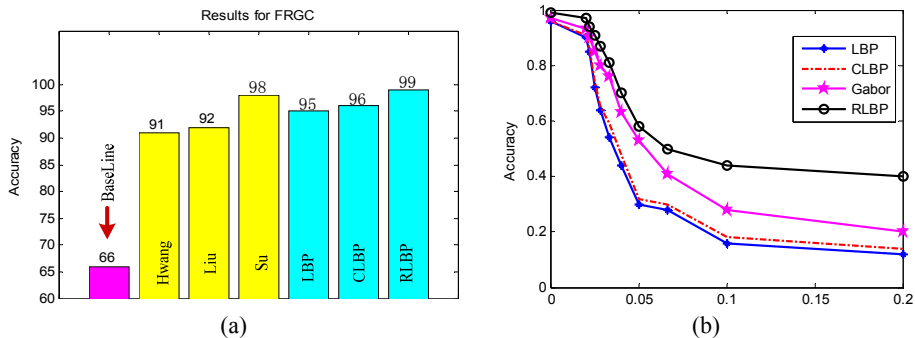


Fig. 9. Performance comparison with existing methods over FRGC face dataset, (a) using the original face samples, (b) using the samples added white Gaussian noise.

Acknowledgements

This work was supported by the FiDiPro program of Tekes, the Academy of Finland and Infotech Oulu.

References

- [1] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*, Dover Publications, New York, 1966.
- [2] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao. WLD: A Robust Local Image Descriptor. *IEEE TPAMI*, 2010.
- [3] B. Fan, F. Wu, and Z. Hu, Rotationally Invariant Descriptors using Intensity Order Pooling, *IEEE TPAMI*, 2012.
- [4] Z. Guo, L. Zhang, and D. Zhang, A Completed Modeling of Local Binary Pattern Operator for Texture Classification, *IEEE TIP*, 2010.
- [5] W. Hwang, G. Park, J. Lee, and S.C. Kee, Multiple Face Model of Hybrid Fourier Feature for Large Face Image Set, *CVPR*, 2006.
- [6] A.C. Jalba, M.H.F. Wilkinson, and J.B.T.M. Roerdink, Morphological Hat-transform Scale Spaces and Their Use in Pattern Classification, *Pattern Recognition*, 2004.
- [7] S. Lazebnik, C. Schmid, and J. Ponce. A Sparse Texture Representation Using Local Affine Regions. *IEEE TPAMI*, 2005.
- [8] S. Liao, M.W.K. Law, and A.C.S. Chung, Dominant Local Binary Patterns for Texture Classification, *IEEE TIP*, 2009.
- [9] C. Liu, Capitalize on Dimensionality Increasing Techniques for Improving Face Recognition Performance, *IEEE TPAMI*, 2006.
- [10] L. Liu, P. Fieguth, G. Kuang, H. and Zha, Sorted Random Projections for Robust Texture Classification, *ICCV*, 2011.
- [11] D. Lowe, Distinctive Image Features from Scale Invariant Key Points, *IJCV*, 2004.
- [12] B. Manjunath and W. Ma, Texture Features for Browsing and Retrieval of Image Data, *IEEE TPAMI*, 1996.
- [13] K. Mikolajczyk and C. Schmid, A Performance Evaluation of Local Descriptors, *IEEE TPAMI*, 2005.
- [14] P. Moreels and P. Perona, Evaluation of Features Detectors and Descriptors based on 3D Objects, *IJCV*, 2007.
- [15] T. Ojala, M. Pietikäinen, and T. Mäenpää, Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns, *IEEE TPAMI*, 2002.
- [16] T. Ojala, K. Valkealahti, E. Oja, and M. Pietikäinen, Texture Discrimination with Multidimensional Distributions of Signed Gray Level Differences, *Pattern Recognition*, 2001.
- [17] P. J. Phillips, P. J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, Overview of the Face Recognition Grand Challenge, *CVPR*, 2005.
- [18] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer Vision Using Local Binary Patterns*. Springer, 2011.
- [19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, ORB: An Efficient Alternative to SIFT or SURF, *ICCV*, 2011.
- [20] Y. Su, S. Shan, X. Chen, and W. Gao, Hierarchical Ensemble of Global and Local Classifiers for Face Recognition, *ICCV*, 2007.
- [21] X. Tan and B. Triggs, Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions, *Proc. IEEE Int'l Workshop on Analysis and Modeling of Faces and Gestures*, 2007.
- [22] E.R. Urbach, J.B.T.M. Roerdink, and M.H.F. Wilkinson, Connected Shape-Size Pattern Spectra for Rotation and Scale-Invariant Classification of Gray-Scale Images, *IEEE TPAMI*, 2007.
- [23] M. Varma and A. Zisserman. A Statistical Approach to Texture Classification from Single Images, *IJCV*, 2005.
- [24] G. Zhao and M. Pietikäinen, Dynamic Texture Recognition Using Local Binary Patterns with an Application to Facial Expressions, *IEEE TPAMI*, 2007.
- [25] S.Z. Li and A.K. Jain, Eds. *Handbook of Face Recognition*, 2nd edition, Springer, 2011.