Circular Mean Filtering For Textures Noise Reduction

M. H. Shakoor*(C.A.) and F. Tajeripour

Abstract: In this paper, a special preprocessing operations (filter) is proposed to decrease the effects of noise of textures. This filter using average of circular neighbor points (C-mean) to reduce noise effect. Comparing this filter with other average filters such as square mean filter and square median filter indicates that it provides more noise reduction and increases the classification accuracy. After applying filter to noisy textures some Local Binary Pattern (LBP) variants are used for feature extraction. The Implementation part for noisy textures of Outex, UIUC and CUReT datasets shows that using proposed filter increases the classification accuracy significantly. Furthermore, a simple and new technique is proposed that increases the speed of c-mean filter noticeably.

Keywords: Circular Mean Filter, Local Binary Pattern, Noise Robust, Texture Classification.

1. Introduction

Texture classification plays an important role in image processing and computer vision. It has been widely used in many applications. Some areas such as fabric defect detection [1, 2], remote sensing [3], medical image processing [4], face recognition [5], and image retrieval [6] are related to texture classification. Texture analysis methods are usually divided into structural, statistical, model based and frequency based methods.

The structural approaches, such as topological texture descriptors [7], morphological operators [8, 9] and filtering techniques [10] are used for description of patterns and their placement. The statistical method including some methods such as co-occurrence matrix [11] and local binary patterns [12]. These methods are based on extracting the statistical properties of textures. Third group is model-based methods such as hidden markov [13], autocorrelation [14] and autoregressive [15] models. Finally, the fourth analyzing methods are frequency-based or transform approaches such as some Gabor and wavelet based algorithms [16-18].

Recently most of these researches focus on statistical analysis of texture. Statistical methods are not only simple but also they can be used for all types of textures. Also they can be combined with other types of texture analysis approaches to increase the performance of texture classification [19-21]. One of the important statistical methods for texture feature extraction is Local Binary Pattern (LBP). LBP is introduced by Ojala et al. [22]. It is a simple operator to describe local texture patterns, and it has achieved high performance for classification results on representative texture datasets [23]. The main goal of proposing LBP was related to texture classification. But LBP has been used for many other applications [24-27].

Some versions of LBP were proposed [28, 29]. $\mathrm{LBP}_{P,R}^{riu2}$ has become the most popular since it decreases the feature number significantly and obtains high discriminative ability. Ojala et al. divided local patterns into two groups; uniform and non-uniform patterns. They proved that the uniform-patterns have more discriminative information than non-uniform patterns. Also noise decreases the percentage of uniform patterns. After that, some extended LBPs [30, 31] were introduced that could produce more features for each texture than $\mathrm{LBP}_{P,R}^{riu2}$. In these methods, the number of features was increased exponentially when the neighbor points of LBP grew. There are many other methods that are proposed based on LBP.

Such as all other local operators, LBP is sensitive to noise. Therefore some noise robust LBP methods are introduced. Improved Local Binary Pattern (ILBP) is proposed by Jin et al. [32]. It is similar to LBP but in ILBP the average value of the whole neighborhood including the center is used instead of center point. Tan and Triggs [33] proposed Local Ternary Pattern (LTP) to quantize the difference between a pixel and its neighbor points into three levels. In LTP it is necessary to set threshold value properly. Liao et al. proposed dominant LBP (DLBP) [34]. It used the most frequently patterns to extract more discriminative features. It selected the 80 % of the most frequently appeared local

Iranian Journal of Electrical & Electronic Engineering, 2015.
Paper first received 18 Apr. 2015 and in revised form 01 Aug. 2015.
* The Authors are with the Computer Engineering Department, School of Computer and Electrical Engineering, Shiraz University, Shiraz, Iran.
E-mails: mhshakoor@gmail.com and tajeri@shirazu.ac.ir.

patterns from histogram of LBP and remain patterns that contains noise non-uniform are removed from features. Median Binary Patterns (MBP) [35] is proposed by Hafiane et al. MBP is robust to noise because it used median gray value of the neighborhood points instead of the center point. Another form of LBP that is resistant to noise is fuzzy local binary pattern FLBP [36] or soft LBP [37]. In FLBP each pixel may contribute to several bins in the histogram of possible patterns by different membership values. The FLBP is a very time consuming method. Fathi et al. [38] proposed a noise tolerant method (NTLBP) that combined a circular majority voting and a new LBP version and regrouped the non-uniform LBP in order to better performance. Ren et al. [39] proposed a much more efficient Noise Resistant Local Binary Pattern (NRLBP) approach. The NRLBP method restores some of the image local structures that are cropped by noise. This method is very time consuming and it cannot generalize to larger scales neighborhoods and it is efficient only for small neighborhood such as R = 1 and P = 8. Lui et al. [40] proposed Binary Rotation Invariant and Noise Tolerant (BRINT) texture classification method that uses mean of neighbor points for LBP. BRINT decreases the effects of noise by using the mean value of some sequential neighbor points instead of each neighbor points. In other words it declines the noise value from the neighbor points. In BRINT method multi resolution LBP is used to increasing the classification accuracy. Some methods such as Completed Robust Local Binary Pattern (CRLBP) [41] used Weighted Local Gray Level (WLG) to reduce the effect of noise on the center point of LBP. This method used average of center and neighbor points of LBP instead of center point. It is similar to ILBP method [32], but CRLBP used a weight value for center point when it calculates the average of the points. The aforementioned methods are some of the most popular LBP that are resistant to noise. There are some noise robust LBPs. Kylberg et al. reviewed and compared most of them in [42].

In spite of almost all noise robust LBP methods that extend LBP to make it resistant to noise in this paper a prepossessing operation (filter) is proposed. Some of filters are general filter that are used for general images. However in this paper by applying the filters to noisy texture the noise effect is decreased. After that it is possible to extract feature from filtered texture by using LBP or any other feature extraction methods.

This paper is organized as follows: in section two LBP and some last versions of LBP are explained. Section three presents proposed methods. Experimental results and conclusion are reported in section four and five respectively.

2 Brief Review of Some LBP's

In this section, we provide a brief review of Local Binary Pattern (LBP) and some state-of-the-art versions of it that used for texture classification and texture analysis.

0	95	32
50	50	61
38	20	41

Image

0	1	0
1		1
0	0	0

Binary Code

1	2	4
8		16
32	64	128

Weights

0	2	0
8		16
0	0	0

LBP Code= 2+8+16=26

Fig. 1 The process of calculating the LBP code.

2.1 Local Binary Pattern (LBP)

The LBP is a descriptor for describing texture features. It provides binary codes by comparing P points of the neighbor pixels with respect to the center pixel value. It generates a binary code 0 if the value of neighbor pixel is smaller than that of the center value of patch. Otherwise, it generates a binary code 1. Then the binary codes are multiplied with the corresponding weights and the results are outlined to generate an LBP code. This value is calculated as follows in Eq. (1):

L (1)

where g_c is the pixel value of the center point and g_i is the pixel value of i-th neighboring pixel, P is the number of neighbor pixels and R is the radius.

s (2)

Fig. 1 shows the process of generating LBP code. In this figure the square neighborhood is used that is not rotation invariant. To obtain rotation invariance, the original LBP was extended to a circular symmetric neighbor set of P members on a circular region with radius R using uniform patterns [12]. The rotation invariant uniform LBP (LBP^{riu2}) can be obtained as follows (Eqs. (3)-(5)):

$$\mathrm{LBP}_{P,R}^{riu2}(x,y)=\begin{cases}\sum_{i=0}^{P-1} s(g_i-g_c) & \text{if } U(\mathrm{LBP}_{P,R})\le 2\\ P+1 & \text{otherwise}\end{cases} \tag{3}$$

$$s(g_i-g_c)=\begin{cases}1 & g_i\ge g_c\\ 0 & g_i< g_c\end{cases} \tag{4}$$

$$U(\mathrm{LBP}_{P,R})=|s(g_{P-1}-g_c)-s(g_0-g_c)|+\sum_{i=1}^{P}|s(g_i-g_c)-s(g_{i-1}-g_c)| \tag{5}$$

Riu2 reflects that the rotation invariant uniform patterns have a U value of at most 2. U is used to estimate the uniformity that corresponds to the number of spatial transitions, i.e., bitwise 0/1 changes between successive bits in the circle. Fig. 2 is an example of the local uniform patterns with different U.

Furthermore, LBP^{ri} and LBP^{u2} are two other types of LBP that are used for texture classification. LBP^{ri} is a rotation invariant method but LBP^{u2} is not. Due to high feature number, both of these methods are very time consuming methods and are not suitable for real time and fast texture processing.

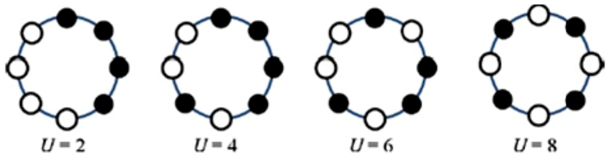


Fig. 2 An example of local binary patterns with different U.

2.2 Completed Local Binary Pattern (CLBP)

The CLBP [43] method is proposed by Gue et al. to combine the sign and magnitude codes of LBP. In CLBP the local difference is divided into the sign (S) and magnitude (M) so CLBP_S and CLBP_M are made. Also the center point of each local patch is compared with mean of image and CLBP_C is made. Both CLBP_S and CLBP_M produce binary strings so that they can be combined for texture classification. In this paper in all parts of results the CLBP refers to CLBP_S/M/C.

2.3 Local Ternary Pattern (LTP)

Local ternary patterns [33] (LTP) is the same as LBP but it uses a threshold value to quantize the difference between a pixel and its neighbors into three levels. LTP is one of the most important and powerful LBP based descriptor that extracts features from noisy texture. The higher value of noise the more value of threshold must be used. In the implementation part threshold is 10.

3 Proposed Method (Circular Mean Filter)

3.1 C-Mean Filter

Here some average filtering methods are used for noise reduction. These filters are applied to noisy textures then features are extracted from textures by using some variants of LBP and they are used for texture classification.

a) Square Mean Filter (s-mean):
Average of center and square neighbor points

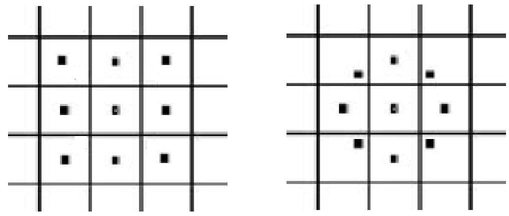
b) Square Median Filter (s-median):
Median of center and square neighbor points

c) Proposed Filter:
Circular Weight Mean Filter (c-mean (α=1,8)):
c-mean (α=1): average of center and circular neighbor points.
c-mean (α=8): average of center and circular neighbor points that uses α=8 as weight of center point.

Circular neighborhood is used in LBP for feature extraction. But in this paper it is used as neighborhood for noise reduction. Fig. 3 shows an example of a neighborhood with center 100. Fig. 3(a) illustrates the 3×3 square neighbor points and Fig. 3(b) determines the circular neighborhood with R = 1. The values of neighbor points in Fig. 3(b) are interpolated and are shown in this figure. Considering this figure it can be calculated that s-mean is 68, s-median is 50 and, c-mean values are 64 and 75 for center weight values α = 1 and α = 8 respectively.

30	40	80
50	100	60
190	50	10

42.2	40	69.3
50	100	60
124.3	50	36.4



(a) (b)

Fig. 3 An example of a part of an image around 100, (a) 3×3 square neighborhood, (b) circular neighborhood (R = 1).

Fig. 3 shows the small neighborhood 3×3 or R = 1. It is possible to use greater neighborhood such as 5×5 or 7×7 for square and R = 2 or 3 for circular neighborhood. The larger neighborhoods remove some local information and texture edges therefore it decreases the texture classification accuracy. All results of implementation part show that all of these filters increase the classification accuracy for all type of LBP that are used here (LBP_S, LBPM, LTP and CLBP).

Some methods such as CRLBP [41] uses Weighted Local Gray Level (WLG) to decline the effect of noise for LBP. CRLBP may be similar to using weight for center point and 3×3 neighborhood for s-mean when it used before CLBP method. According to CRLBP method the s-mean with α = 1 provides better accuracy than α = 8. So in this paper the s-mean with α = 1 is used for all parts of implementation. The proposed circular mean filter has some advantages rather than CRLBP. (CRLBP uses a technique which is the same as s-mean filter):

- C-mean filter reduces the noise affection and it is independent to feature extraction method. In other words it is a preprocessing method to reduce the effects of noise. After noise reduction it is possible to use any feature extraction methods.
- C-mean filter provides more uniform patterns than s-mean therefore it provides more discriminative features than s-mean for texture classification. The implementation part indicates that the accuracy of texture classification is increased when c-mean filter is used instead of s-mean.
- In spite of s-mean filter that uses square neighborhood the c-mean filter uses circular neighborhood therefore it is a rotation invariant filter.
- Implementation indicates that c-mean and s-median filters provide the best and the worst accuracy respectively. Sometimes the accuracy of s-mean reaches the c-mean.

The only advantage of using s-mean instead of c-mean is that the c-mean requires the interpolation step to calculate the neighbor points on the circle, therefore, c-mean is more time consuming method than s-mean. In the next part a simple and new technique is proposed that increases the speed of c-mean significantly.

3.2 Fast C-Mean Filter

In this part a simple technique is introduced to increase the speed of c-mean filtering. Using c-mean instead of s-mean filter is more time consuming. Because the circular mean filter requires interpolation for neighbor points on the circular patch. Also, If the image size is M×N, a 3×3 mask should be convolved M×N times with image which requires considerable time.

For 3×3 patch, 4 point values should be interpolated from neighbor points. In this part a simple technique is proposed that reduce this time significantly. According to Fig. 3(b), 4 points of circular neighborhood (corners) should be interpolated. For example the upper right value on circular neighborhood is 69.3. It is provided by weighted mean of 80, 40, 100 and 60. The weight of these points are 0.21, 0.5, 0.21 and 0.09 respectively. These weights are provided reverse to the distance of center of each circular neighbor point and center of original points. For other 3 corner points the weight values are the same but it selected according to distance from center of each neighbor points. Therefore, 4 interpolated images are provided.

If the image size is M×N, a 3×3 mask (s-mean or c-mean) should be convolved M×N times with image which requires considerable time. To reduce this time, the noisy image should be shifted one pixel to four directions. By calculating, the average of the original and 4 shifted and 4 interpolated images I_{avg} is obtained. I_{avg} is the same as the noisy image that is filtered by C-mean filter. This method increases the speed of average filtering more than 10 times.

4 Experimental Results

To evaluate the performance of the proposed methods, some comprehensive texture datasets are used: the Outex dataset [23], the UIUC [44] and the CUReT [45]. In this part only riu2 mapping method is used. In all cases the threshold value for LTP is 10 and CLBP refers to CLBP_S/M/C.

4.1 Dissimilarity Metric Method

For comparing two textures the LBP histograms of them must be compared. There are many methods for comparing two histograms, such as histogram intersection, log-likelihood ratio, and chi-square method [12]. In this paper such as many researches the chi-square method is used for classification. Eq. (9) shows chi-square method. A test sample *T* is assigned to the class of model *L* that minimizes the chi-square distance:

D (9)

where *N* is the number of bins of each histogram, and T_i and L_i are the values of the sample and the model image at the i_{th} bin respectively. The nearest neighborhood classifier with k = 1 is used for classification.

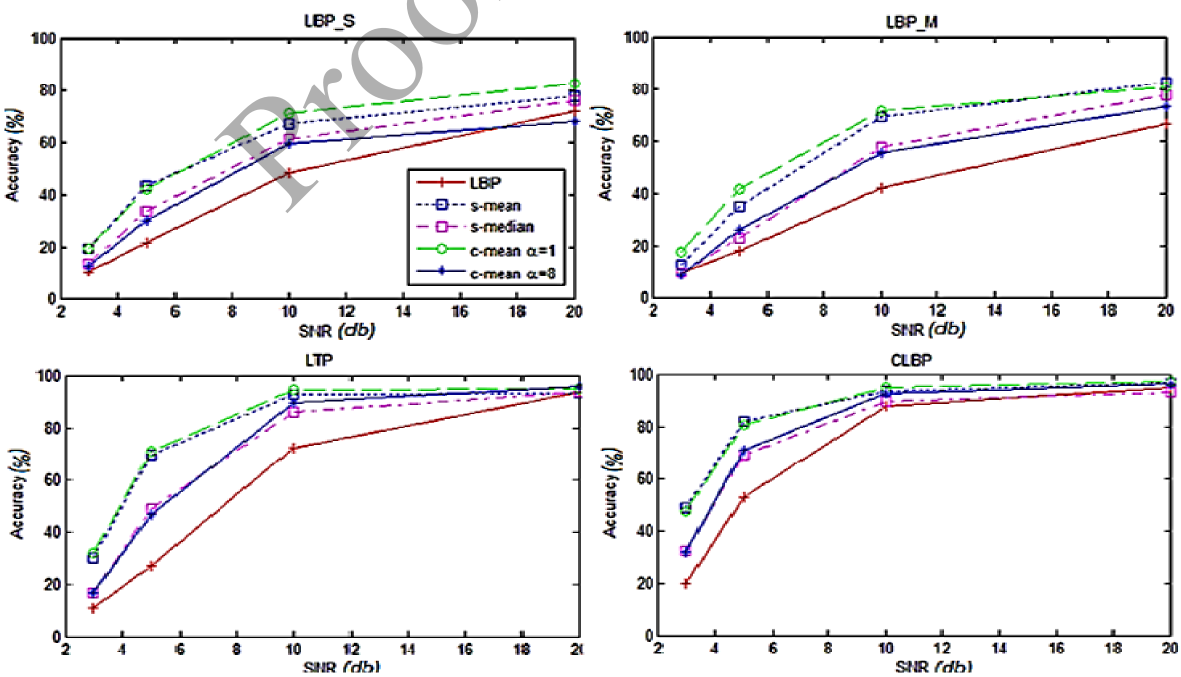


Fig. 4 Comparing the accuracy for LBP_S, LBP_M, LTP and CLBP after applying average filters for Outex (TC10) for different SNR values. (Radius = 1 and α = 1.8 and 3x3 mask).

4.2 Experimental Results on the Outex Dataset

The Outex dataset includes some test suites [23]. These suites were collected under different illumination, rotation, and scaling conditions. Outex_TC_00010 (TC10) and Outex_TC_00012 (TC12) are considered as famous two test suites in this dataset. They can be used for rotation invariant tests. These two suites have the same 24 classes of textures, which are collected under three different illuminates (horizon, inca, and t184) and nine different rotation angles (0º, 5º, 10º, 15º, 30º, 45º, 60º, 75º and 90º). Fig. 4 determines the results of using filters for noisy textures of TC10. Considering this figure it indicates that in all plots of the circular neighbor mean (c-mean) provides the best results when center weight is 1 when the weight set by 8 the accuracy declines drastically. In some plots such as CLBP the accuracy of s-mean reaches the accuracy of c-mean (α = 1). All of the plots of Fig. 4 are provided by using 3×3 (for square neighborhood) and R = 1 (for circular neighborhood) and for SNR = 3, 5, 10 and 20. Fig. 5 illustrates the results of using greater filters for TC10.

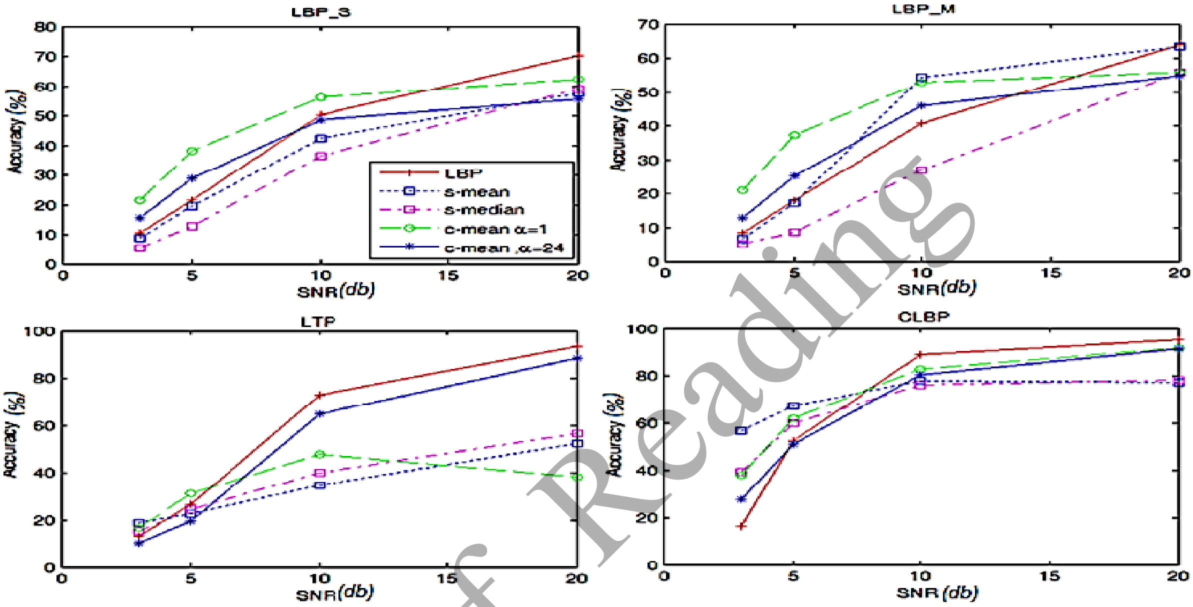


Fig. 5 Comparing the accuracy for LBP_S, LBP_M, LTP and CLBP after applying average filters for Outex (TC10) for different SNR values. (Radius = 3 and α = 1,24 and 7x7 mask).

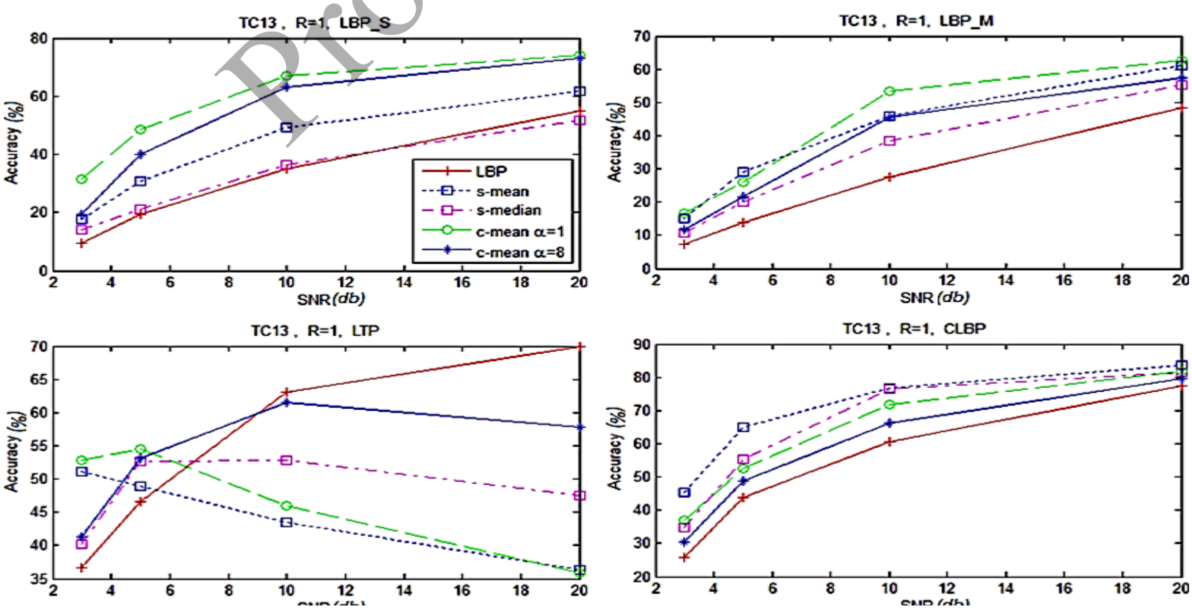


Fig. 6 Comparing the accuracy for LBP_S, LBP_M, LTP and CLBP after applying average filters for Outex (TC13) for different SNR values. (Radius = 1 and α = 1,8 and 3x3 mask).

It is the same as Fig. 4 but the size of filter is larger than Fig. 4 and for square mean 7×7 window is used and for circular mean the radius is R = 3. Also for c-mean the values of 1 and 24 are used for weight. Both of Figs. 4 and 5 extract features by using R = 1 and P = 8 for LBP_S, LBP_M, LTP and CLBP. Considering the Figs. 4 and 5 they show that in all cases the performance of large filter is lower than the performance of small filter.

Also in this paper TC13 is used. This suite includes 68 texture classes with size 128*128 and inca illumination. Fig. 6 compares the results of classification rates by applying some average filter on noisy textures for TC13.

The results of applying filter on TC13 noisy textures for LBP_S and for low noisy texture of LBP_M is the same as TC10. For CLBP and SNR < 10 for LBP_M s-mean provides the best results. Also the plot of LTP indicates that using these filters increases the accuracy for SNR < 10. While for low noisy texture these filters decreases the performance. It relates to the threshold value of LTP.

4.3 Experimental Results on the UIUC Dataset

The UIUC dataset [44] contains 25 classes with 40 images in each class. Each texture is 640*480. For implementation, each time, N = 20 images of each class are selected randomly for train and the rest of them (40-N) are used for test. This operation is run 100 times independently and the average of the results is shown in Fig. 7. These operations are carried out riu2 mapping for LBP_S, LBP_M, LTP and CLBP. R = 1 and P = 8 for feature extraction. In Fig. 7 such as previous figures the classification accuracy of LBP_S, LBP_M, LTP and CLBP for SNR = 3, 5, 10 and 20 are shown. For square filters 3x3 neighborhood is used and for c-mean the circular neighborhood with R = 1 is selected.

Considering the plots it is shown that c-mean whit α = 1 provides the best results for LBP_S, LTP and CLBP. Only for LBP_M the s-mean has the best performance. Furthermore, these plots show that all of these filters reduce the noise and increase the classification accuracy of textures. Such as Outex, in UIUC dataset the highest accuracy is provided by CLBP. In this figure, the size of mask is 3x3. If the larger mask is used the accuracy is decreases significantly. Because the larger average mask removes some edges and local information of textures. It is true for all other datasets that are used in this paper.

4.4 Experimental Results on the CUReT Dataset

The CUReT database [45] contains 61 classes of textures. These texture images captured at different viewpoints and illumination orientations. For each class, 92 images are selected from the images that have a viewing angle of less than 60°. For this part every time N = 46 images are chosen randomly for train from each class. The remaining (92-N) images are used as test samples. The average classification rates are obtained from 100 random tests. In this dataset R = 1 and riu2 mapping is used for feature extraction and for filtering the 3x3 square neighborhood and circular neighborhood with R = 1 are used. The results of classification for LBP_S, LBP_M, LTP and CLBP are shown on plots of Fig. 8. Such as previous figures these plots are related to classification accuracy for SNR = 3, 5, 10 and 20.

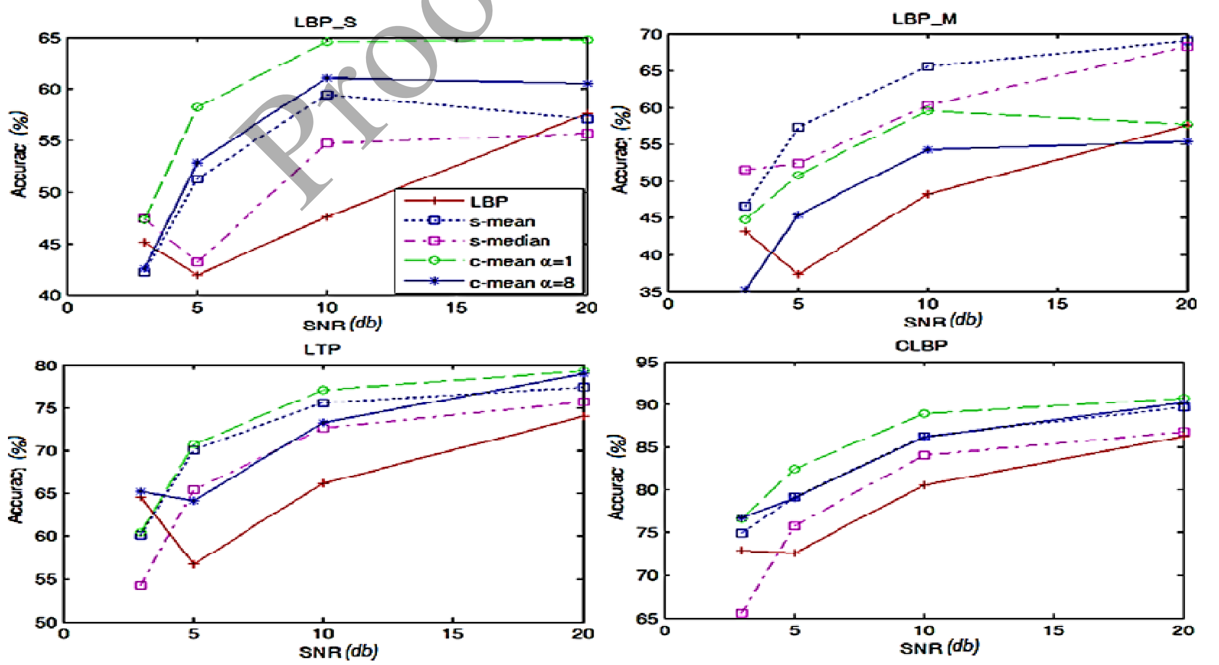


Fig.7 Comparing the accuracy for LBP_S, LBP_M, LTP and CLBP after applying average filters UIUC for different SNR values. (Radius = 1 and α = 1,8 and 3x3 mask).

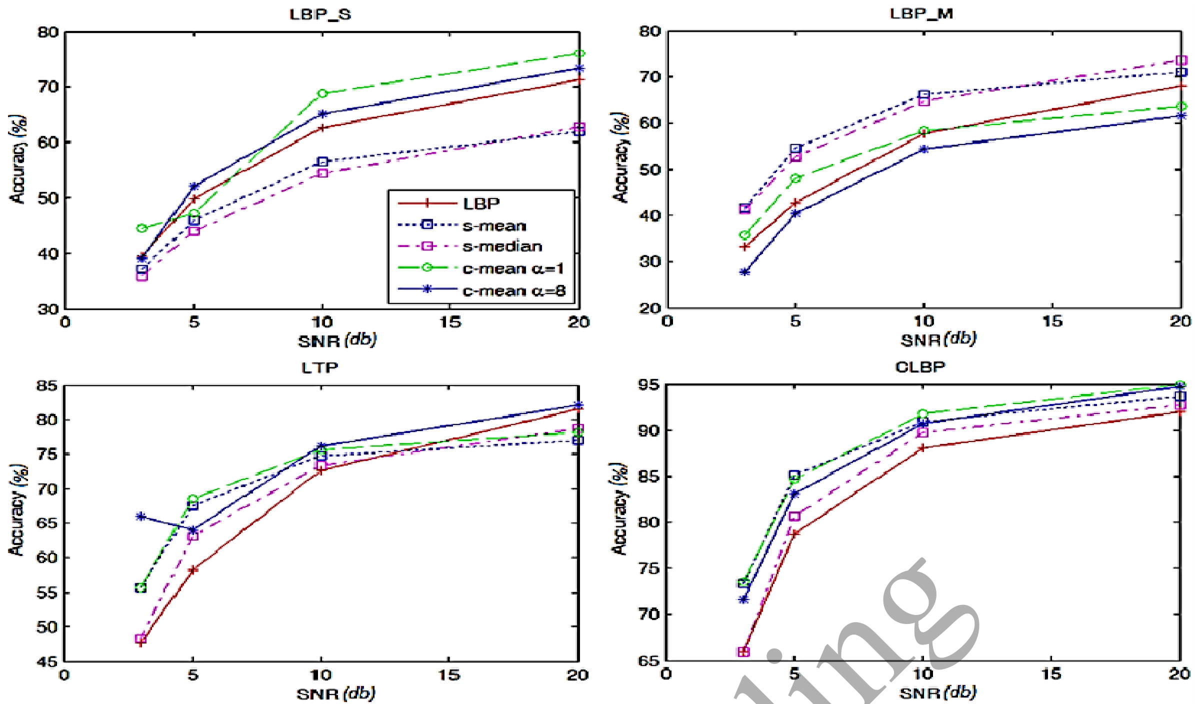


Fig. 8 Comparing the accuracy for LBP_S, LBP_M, LTP and CLBP after applying average filters CUReT for different SNR values. (Radius=1 and α =1,8 and 3x3 mask)

Considering the plots of Fig. 8 they indicate that for LBP_S, LTP and CLBP the c-mean with α = 1 provides the best accuracy. Unless for LBP_M the accuracy of s-mean is the best. According to Fig. 8 the highest performance is related to CLBP. For LBP_S both c-mean with α = 1 and 8 prepare the best performance. For LTP and CLBP the s-mean provides the second best accuracy. Such as Outex and UIUC in this dataset the performance of median filter is low. For LTP and CLBP using these filters increases the accuracy. However for LBP_S only c-mean filter increases the accuracy of classification. While for LBP_M using the c-mean filter with α = 8 decreases the performance.

5 Conclusion

In this paper some average filtering methods are used that reduce the noise affection on texture images. These filters apply to texture before any feature extraction and classification. The implementation part shows that usually circular mean filtering with α = 1 provides better performance than other filters. Also in some cases the square 3x3 filter prepares slightly better accuracy than circular mean filter. Median filter also increases the performance of classification but this filter provides lower accuracy than circular and square mean filters. Also larger filters provide lower performance than small filters. The implementation shows that c-mean (α = 1) usually provides more accurate features for texture classification than s-mean. If s-mean filter with 3x3 neighborhood is applied to noisy textures then CLBP is used for feature extraction it is the same as CRLBP. Considering the plots of CLBP in the implementation part they illustrate that for CLBP feature extraction method, the accuracy of c-mean is higher than s-mean(CRLBP) for TC10 (SNR > 10), UIUC and CUReT. Only for TC13 and low SNR of TC10, the performance of s-mean is slightly better than c-mean.

References

[1] F. S. Cohen, Z. Fan and S. Attali, “Automated inspection of textile fabrics using Textural models”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 8, pp. 803-808, 1991.

[2] F. Tajeripour, E. Kabir and A. Sheikhi, “Fabric defect detection using modified local binary patterns”, *EURASIP Journal on Advances in Signal Processing.* Vol. 8, No. 1, pp. 1-12, 2008.

[3] H. Anys and D. C. He, “Evaluation of textural and multi polarization radar features for crop classification”, *IEEE Transactions on Geoscience and Remote Sensing* Vol. 33, No. 5, pp. 1170-1181, 1995.

[4] Q. Ji, J. Engel and E. Craine, “Texture analysis for classification of cervix lesions”, *IEEE Transactions on Medical Imaging* Vol. 19, No, 11, pp. 1144-1149, 2000.

[5] B. Zhang, Y. Gao, S. Zhao and J. Liu, “Local derivative patterns versus local binary patterns: face recognition with high-order local patterns descriptor”, *IEEE Transactions on Image Processing*, Vol. 19, No. 2, pp. 533-544, 2010.

[6] S. Murala, R. P. Maheshwari and R. Bala subramanian, "Local tetra patterns: a new feature descriptor for content-based image retrieval", *IEEE Transactions on Image Processing*, Vol. 21, No. 5, pp. 2874–2886, 2012.

[7] G. Eichmann and T. Kasparis, "Topologically invariant texture descriptors", *Computer Vision, Graphics and Image Processing*, Vol. 41, No. 3, pp. 267–281, 1988.

[8] W. K. Lam and C. Li, "Rotated texture classification by improved iterative morphological decomposition", *IEEE Proceedings Vision, Image and Signal Processing*, Vol. 144, No. 3, pp. 171–179, 1997.

[9] K. F. Coco, E. O. T. Salles and M. Sarcinelli-Filho, "Topographic independent component analysis based on fractal theory and morphology applied to texture segmentation", *Signal Process*, Vol. 87, No. 8, pp. 1966–1977, 2007.

[10] T. Randen and J. Husoy, "Filtering for texture classification: a comparative study", *IEEE Trans. Pattern Analysis and Machine Intelligent*, Vol. 21, No. 4, pp. 291–310, 1999.

[11] R. M. Haralik, K. Shanmugam and I. Dinstein, "Texture features for image classification", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3, No. 6, pp. 610–621, 1973.

[12] T. Ojala, M. Pietikainen and T. T.Maenpa, "Multi resolution gray-scale and rotation Invariant texture classification with local binary patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 971–987, 2002.

[13] J. L. Chen and A. Kundu, "Rotation and gray scale transform invariant texture identification using wavelet decomposition and hidden Markov model", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 2, pp. 208–214, 1994.

[14] P. Campisi, A. Neri, C. Panci and G. Scarano, "Robust rotation-invariant texture classification using a model based approach", *IEEE Transactions on Image Processing*, Vol. 13, No. 6, pp. 782–791, 2004.

[15] R. L. Kashyap and A. Khotanzad, "A model-based method for rotation invariant texture classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 4, pp. 472–481, 1986.

[16] H. Arof and F. Deravi, "Circular neighborhood and 1-DDFT features for texture classification and segmentation", *IEEE Proceedings Vision, Image, and Signal Processing*, Vol. 145, No. 3, pp. 167–172, 1998.

[17] N. D. Kim and S. Udpa, "Texture classification using rotated wavelet filters", *IEEE Transactions on Systems, Man and Cybernetics, Part A:Systems and Humans,* Vol. 30, No. 6, pp. 847–852, 2000.

[18] M. Kokare, P. K. Biswas and B. N. Chatterji, "Rotation-invariant texture image Retrieval using rotated complex wavelet filters", *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics,* Vol. 36, No. 6, pp. 1273–1282, 2006.

[19] R. M. Haralick, K. Shanmugam and I. H. Dinstein, "Textural Features for Image Classification", *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 3, No. 6, pp. 610-621, 1979.

[20] M. Galloway, "Texture analysis using gray level run lengths", *Computer Graphics and Image Processing*, Vol. 4, No. 2, pp. 172-199, 1975.

[21] A. H. Mir, M. Hanmandlu and S. N., "Tandon, Texture analysis of CT images", *Engineering in Medicine and Biology Magazine*, Vol. 14, No. 6, pp. 781-786, 1995.

[22] T. Ojala, M. Pietikäinen and D. A. Harwood, "Comparative Study of texture Measures with Classification Based on Feature Distributions", *Pattern Recognition*, Vol. 29, No. 1, pp. 51-59, 1996.

[23] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen and S. Huovinen, "Outex - new framework for empirical evaluation of texture analysis algorithm", *in Proc. International Conference on Pattern Recognition*, Vol. 1, No. 1, pp. 701-706, 2002.

[24] T. Ahonen, A. Hadid and M. Pietikäinen, "Face recognition with Local Binary Patterns: application to face recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 12, pp. 2037-2041, 2006.

[25] G. Zhao and M. Pietikäinen, "Dynamic texture recognition using Local Binary Patterns with an application to facial expressions", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 27, No. 6, pp. 915-928, 2007.

[26] L. Nanni and A. Lumini, "Local binary patterns for a hybrid fingerprint matcher", *Pattern Recognition*, Vol. 41, No 11, pp. 3461–3466, 2008.

[27] A. Lucieer, A. Stein and P. Fisher, "Multivariate texture-based segmentation of remotely sensed imagery for extraction of objects and their uncertainty", *Int. J. Remote Sens.* Vol. 26 No. 14, pp. 2917–2936, 2005.

[28] T. Ojala, "Nonparametric Texture Analysis Using Simple Spatial Operators, with Applications in Visual Inspection", *Acta Universitatis Ouluensis*, Vol. 1, No. 105, 1997.

[29] M. Pietikäinen, T. Ojala and Z. Xu, "Rotation-invariant Texture Classification Using Feature Distributions", *Pattern Recognition*, Vol. 33, No. 1, pp. 43-52, 2000.

[30] D. Huang, Y. Wang and Y. Wang, "A Robust Method for Near Infrared Face Recognition Based

on Extended Local Binary Pattern”, *in Proc. Int. Symp. Vis. Comput.*, pp. 437-446, 2007.

[31] Y. Huang, Y. Wang and T. Tan, “Combining Statistics of Geometrical and Correlative Features for 3D Face Recognition”, *in Proc. Brit. Mach. Vis. Conf.*, Vol. 1, No. 3, pp. 879-888, 2006.

[32] H. Jin, Q. Liu, H. Lu and X. Tong, “Face detection using improved LBP under Bayesian framework”, *in Proceedings of the 3rd International Conference on Image and Graphics, ICIG 2004, Hong Kong China*, Vol. 3, No.1, pp. 306–309, 2004.

[33] X. Tan, and B. Triggs, “Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions”, *in Proc. International Workshop on Analysis and Modeling of Faces and Gestures*, pp.168-182, 2007.

[34] S. Liao, M. W. K. Law and A. C. S. Chung, “Dominant local binary patterns for texture classification”, *IEEE Trans. on Image Processing*, Vol. 18, No. 5, pp. 1107-1118, 2009.

[35] A Hafiane, G Seetharaman and B Zavidovique, “Median Binary Pattern for Textures Classification”, *in Proceedings of the 4th International Conference, ICIAR 2007, Vol. 4633 of Lecture Notes in Computer Science. Median binary pattern for textures classification* (*Montreal, Canada, 2007*), pp. 387–398, 2007.

[36] D. K. Iakovidis, E. G. Keramidas and D. Maroulis, “Fuzzy Local Binary Patterns for Ultrasound Texture Characterization”, *in Proceedings of the 5th International Conference on Image Analysis and Recognition, ICIAR 2008, Vol. 5112 of Lecture Notes in Computer Science. Fuzzy local binary patterns for ultrasound texture characterization (Povoa de Varzim Portugal*), pp. 750–759, 2008.

[37] T Ahonen, M Pietikainen, “Soft histograms for local binary patterns”, *in Proceedings of the Finnish Signal Processing Symposium, FINSIG 2007, Soft histograms for local binary patterns (Oulu, Finland, 2007),* pp. 1–4, 2007.

[38] A. Fathi and A. R. Naghsh-Nilchi, “Noise tolerant local binary pattern operator for efficient texture analysis”, *Pattern Recognit. Letters*, Vol. 33, No. 9, pp. 1093- 1100, 2012.

[39] J. Ren, X. Jiang and J. Yuan, “Noise resistant local binary pattern with an embedded error correction mechanism”, *IEEE Trans. Image Process.*, Vol. 22, No. 10, pp. 4049-4060, 2013.

[40] L. Liu, Y. Long, P. Fieguth, S. Lao and G. Zhao, “BRINT: Binary Rotation Invariant and Noise Tolerant Texture Classification”, *IEEE Trans. on Image Processing*, Vol. 23, No. 7, pp. 3071-3084, 2014.

[41] Y. Zhao, W. Jia, R.-X. Huc and H. Min, “Completed robust local binary pattern for texture classification”, *Neurocomputing*, Vol. 106, No. 4, pp. 68-76, 2013.

[42] G. Kylberg and I. M. Sintorn, “Evaluation of noise robustness for local binary pattern descriptors in texture classification”, *EURASIP Journal on Image and Video Processing*, Vol. 17, No. 1, pp. 107-114, 2003.

[43] Y. Zhao, D. S. Huang and W. Jia, “Completed Local Binary Count for Rotation Invariant Texture Classification”, *IEEE Transactions on Image Processing*, Vol. 21, No.10, pp. 4492–4497, 2012.

[44] S. Lazebnik, C. Schmid and J. Ponce, “A sparse texture representation using local affine regions”, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 27, No. 8, pp. 1265–1278, 2005.

[45] K. J. Dana, B. Van Ginneken, S. K. Nayar and J. J. Koenderink, “Reflectance and texture of real world surfaces”, *ACM Transactions on Graphics*, Vol. 18, No. 1, pp. 1-34, 1999.

Mohammad Hossein Shakoor received the B.Sc. degree in computer engineering from Shiraz University, Shiraz, Iran, in 1998 and M.Sc. degree in computer architecture from Isfahan University, Isfahan, Iran, in 2003. Currently he is pursuing Ph.D. in Artificial Intelligent of Computer engineering at Shiraz University. His research interests include texture classification, pattern recognition and computer vision.

Farshad Tajeripour received the B.Sc. and M.Sc. degrees in Electronic engineering from Shiraz University, Shiraz, Iran, in 1994 and 1997. He received Ph.D. degree in Electronic engineering from Tarbiat Modarres University, Tehran, Iran, in 2009. Currently he is an assistant professor in Shiraz University. His research interests include texture classification, pattern recognition, computer vision and video processing.