Turkish Journal of Science & Technology
16(1), 163-177, 2021

Novel Fuzzy Kernels Based Local Binary Pattern And Local Graph Structure Methods

Serkan METİN[1*], Şengül DOĞAN[2]
[1] Department Of Management Information Systems, Faculty of Social Sciences and Humanities, Malatya Turgut Özal University, Malatya, Turkey
[2] Department of Digital Forensics Engineering, Technology Faculty, Firat University, Elazig, Turkey
[*1] serkan.metin@ozal.edu.tr, [2] sdogan@firat.edu.tr

(Geliş/Received: 11/02/2021; Kabul/Accepted: 15/02/2021)

Abstract: Local descriptors are the most effective textural image recognition methods. Local descriptors generally consist of two phases. These are binary feature coding and histogram extraction phases, and they often use the signum function for the binary feature extraction. In this article, new fuzzy-based mathematical kernels are proposed for binary feature encoding in local descriptors. Fuzzy kernels consist of membership degree calculation and coding these membership degrees. In order to calculate membership degrees, four fuzzy sets are utilized. The proposed fuzzy kernels are considered as binary feature-extraction functions, and a novel textural image recognition architecture is created using these fuzzy kernels. These architecture phases are; (1) binary feature coding with fuzzy kernels, (2) calculating lower and upper images, (3) histogram extraction, (4) feature reduction with maximum pooling, (5) classification. In the classification phase, a quadratic kernel-based support vector machine (SVM) classifier is utilized. The presented fuzzy kernels are implemented on the Local Binary Pattern (LBP) and Local Graph Structure (LGS). 16 novel methods are presented using fuzzy kernels for each descriptor. In this article, LBP and LGS are used, and 32 novel fuzzy-based methods are proposed to improve recognition capability. 3 facial images and 3 textural image datasets are used to evaluate the methods' performance. The experimental results clearly illustrate that the fuzzy kernels based LBP and LGS methods have high facial and textural image recognition capability.

Keywords: Fuzzy coding, local binary pattern, local graph structure, texture recoginiton, face recognition, biometrics.

Yeni Bulanık Çekirdeklere Dayalı Yerel İkili Desen ve Yerel Grafik Yapısı Yöntemleri

Öz: Yerel tanımlayıcılar, en etkili dokusal görüntü tanıma yöntemleridir. Yerel tanımlayıcılar genellikle iki aşamadan oluşur. Bunlar ikili özellik kodlama ve histogram çıkarma aşamalarıdır ve genellikle ikili özellik çıkarımı için işaret işlevini kullanırlar. Bu makalede, yerel tanımlayıcılarda ikili özellik kodlaması için yeni bulanık tabanlı matematiksel çekirdekler önerilmiştir. Bulanık çekirdekler, üyelik derecesi hesaplaması ve bu üyelik derecelerinin kodlanmasından oluşur. Üyelik derecelerini hesaplamak için dört bulanık küme kullanılır. Önerilen bulanık çekirdekler, ikili özellik çıkarma işlevleri olarak kabul edilir ve bu bulanık çekirdekler kullanılarak yeni bir dokusal görüntü tanıma mimarisi oluşturulur. Bu mimari aşamalar; (1) bulanık çekirdekli ikili özellik kodlaması, (2) alt ve üst görüntülerin hesaplanması, (3) histogram çıkarma, (4) maksimum havuzlama ile özellik azaltma, (5) sınıflandırma. Sınıflandırma aşamasında, ikinci dereceden çekirdek tabanlı bir destek vektör makinesi (SVM) sınıflandırıcısı kullanılır. Sunulan bulanık çekirdekler, Yerel İkili Model (LBP) ve Yerel Grafik Yapısı (LGS) üzerinde uygulanmaktadır. Her tanımlayıcı için bulanık çekirdekler kullanılarak 16 yeni yöntem sunulmaktadır. Bu makalede, LBP ve LGS kullanılmış ve tanıma yeteneğini geliştirmek için 32 yeni bulanık tabanlı yöntem önerilmiştir. Yöntemlerin performansını değerlendirmek için 3 yüz görüntüsü ve 3 dokusal görüntü veri kümesi kullanılır. Deneysel sonuçlar, bulanık çekirdeklere dayalı LBP ve LGS yöntemlerinin yüksek yüz ve dokusal görüntü tanıma kapasitesine sahip olduğunu açıkça göstermektedir.

Anahtar kelimeler: Bulanık kodlama, yerel ikili desen, yerel grafik yapısı, doku tanıma, yüz tanıma, biyometri.

1. Introduction

Texture image recognition is used in many areas such as the face, iris, facial expression, perceptual hashing, wood, palm, etc. [1-3]. These methods generally consist of pre-processing, feature extraction, and classification stages, and local descriptors are commonly used in the feature extraction phase [4]. The first known local descriptors are LBP. LBP uses 3 x 3 non-overlapping blocks and signum function to extract textural features [5, 6]. LBP generally consists of two main stages, and these are binary feature coding and histogram extraction. This histogram is utilized as a feature of the image. LBP is often used in the literature because it provides many advantages in textural image analysis. The advantages of LBP are as follows [7-9].

1- LBP can be easily programmed.

[*] Corresponding author: serkan.metin@ozal.edu.tr. ORCID Number of authors: [1] 0000-0003-1765-7474, [2] 0000-0001-9677-5684

2- LBP is well discriminator.
3- LBP has a short execution time.
4- LBP can be easily applied in real-time applications.

These advantages indicate that LBP is an effective textural image operator. Therefore, many LBP-like local patterns have been proposed in the literature. When 8-bit features are extracted by using LBP, 4-bit features are extracted by using Center Symmetric Local Binary Pattern (CSLBP), and 9 bits are extracted by using Local Quadruple Pattern (LQPAT) [10, 11]. Therefore, feature sets of 16, 256, and 512 dimensions are obtained using these textural operators. Recognition and classification operations are performed by using the acquired features. These methods consist of binary pattern coding and histogram extraction sections as LBP. In addition, there are graph-based micropatterns in the literature. The first known method is LGS (Local Graph Structure) method. Several methods such as SLGS (Symmetric LGS), ELGS (Extended LGS), VLGS (Vertical LGS), VSLGS (Vertical Symmetric LGS) were presented after the high performance achieved by the LGS method [12-14]. The majority of the LGS and LBP-based methods use the signum function, and they only aim to improve the recognition performance by changing pattern shapes [15].

In this article, fuzzy-based binary feature extraction functions are proposed to improve the performance of local descriptors. These are called fuzzy kernels in this article, and the technical contribution of the proposed method is as follows.

- 4 novel binary feature extraction functions are presented using 4 fuzzy sets.
- A novel textural image recognition architecture is proposed in this article. 16 different feature extraction methods were obtained for each descriptor using this architecture.
- In this article, fuzzy kernels are applied to the LBP and LGS. The main purpose of using LBP and LGS is to be widely used in the literature. The effect of fuzzy on LBP and LGS are analyzed in detail.
- 6 databases were used to test the proposed methods' performance, and these databases consist of textural and facial images. The performances of the proposed methods are tested in both facial and textural image recognition using these databases.

2. Related Works

Related work has been mentioned in this part of the article. These are the local descriptors previously proposed in the literature. These are LBP and LGS. These methods are mentioned respectively. In the proposed method, the mathematical structure of these descriptors is changed [12-14].

2.1. Local Binary Pattern

LBP is the first local descriptor, and it was presented in 1996. This method consists of two main phases, and these are binary feature coding and histogram extraction. LBP uses 3 x 3 size blocks to extract textural features of the images. This is the main descriptor, and it has several variations in the literature. The LBP is summarized graphically in Fig. 1 [5].

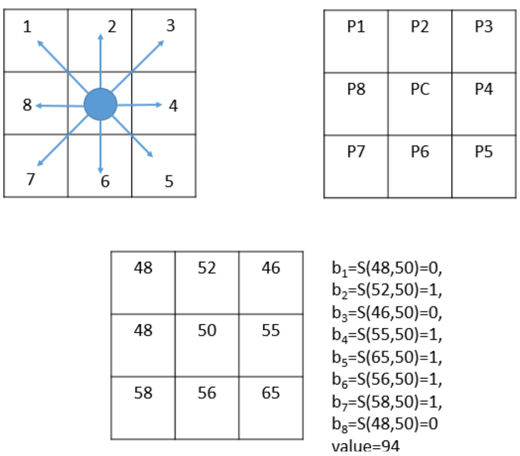


Figure 1. Graphical outline of LBP.

As shown in Fig. 1, LBP uses the signum function to extract binary features. The mathematical definition of the signum function is given in Eq. 1.

$$S(a,b) = \begin{cases} 0, a-b<0 \\ 1, a-b\geq 0 \end{cases} \tag{1}$$

Also, a mathematical description of the LBP is given in Eq. 2-6.

$$LBP^1_{i,j} = S(p_{i,j}, p_{i+1,j+1}) \text{ x } 128 + S(p_{i,j+1}, p_{i+1,j+1}) \text{ x } 64 \tag{2}$$

$$LBP^2_{i,j} = S(p_{i,j+2}, p_{i+1,j+1}) \text{ x } 32 + S(p_{i+1,j+2}, p_{i+1,j+1}) \text{ x } 16 \tag{3}$$

$$LBP^3_{i,j} = S(p_{i+2,j+2}, p_{i+1,j+1}) \text{ x } 8 + S(p_{i+2,j+1}, p_{i+1,j+1}) \text{ x } 4 \tag{4}$$

$$LBP^4_{i,j} = S(p_{i+2,j}, p_{i+1,j+1}) \text{ x } 2 + S(p_{i+1,j}, p_{i+1,j+1}) \tag{5}$$

$$LBP_{i,j} = LBP^1_{i,j} + LBP^2_{i,j} + LBP^3_{i,j} + LBP^4_{i,j} \tag{6}$$

where, p is the pixel of the image, S is signum function, $BP^k_{i,j}, k = \{1,2,3,4\}$ 2-bit values of the LBP and $LBP_{i,j}$ defines LBP value of the 3 x 3 size of block [16, 17].

2.2. Local Graph Structure

LGS is a graph-based descriptor. It is similar to LBP. The main purpose of the LGS is to extract dominant feature using graphs. This operator uses 3 x 4 size of block and signum function like LBP to extract feature. In Fig. 2, LGS is summarized as graphically [13, 18].

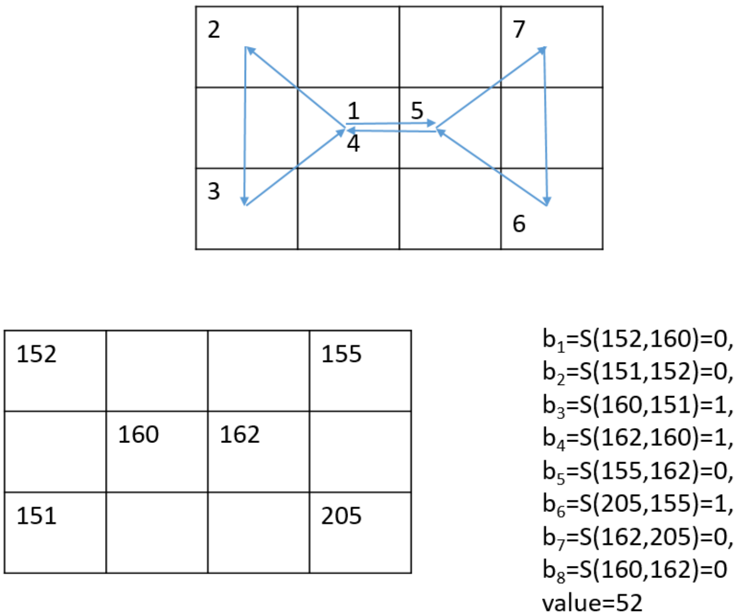


Figure 2. Graphical outline of LGS.

Mathematical definition of LGS is given in Eq. 7-11.

$$LGS^1_{i,j} = S(p_{i,j}, p_{i+1,j+1}) \text{ x } 128 + S(p_{i,j+1}, p_{i,j+2}) \text{ x } 64 \tag{7}$$

$$LGS^2_{i,j} = S(p_{i,j}, p_{i,j+2}) \text{ x } 32 + S(p_{i+1,j+2}, p_{i+1,j+1}) \text{ x } 16 \tag{8}$$

$$LGS^3_{i,j} = S(p_{i,j+3}, p_{i,+1j+2}) \text{ x } 8 + S(p_{i+2,j+3}, p_{i,j+3}) \text{ x } 4 \tag{9}$$

$$LGS^4_{i,j} = S(p_{i+1,j+2}, p_{i+2,j+3}) \text{ x } 2 + S(p_{i+1,j+1}, p_{i+1,j+2}) \tag{10}$$

$$LGS_{i,j} = LGS^1_{i,j} + LGS^2_{i,j} + LGS^3_{i,j} + LGS^4_{i,j} \tag{11}$$

Where $LGS^k_{i,j}, k = \{1,2,3,4\}$ 2-bit values of the LGS and $LGS_{i,j}$ defines LGS value of the 3 x 4 size of block [12, 19].

3. The Fuzzy-Based Binary Feature Extraction Functions

4 fuzzy functions are used in this article to extract binary features. These functions consist of two phases. These are;

1. Calculation membership degree,
2. Coding.

The presented fuzzy functions are explained below.

3.1. Triangle Function

One of the basic fuzzy sets is the triangle and is widely used in the literature. This set is given below.

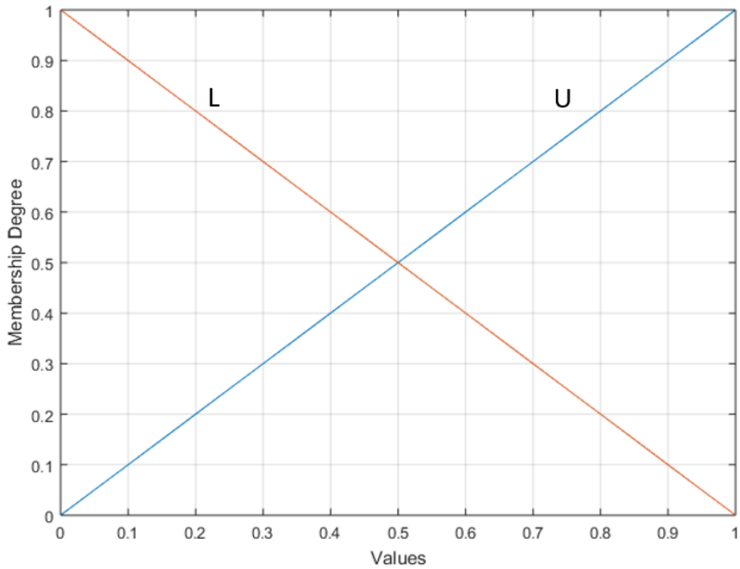


Figure 3. The used triangle fuzzy set.

The used triangle fuzzy set is shown in Fig. 3. The function obtained by using this set is called as tri(.) in this article, and the mathematical definition of the tri(.) is given in Eq. 12-13.

$$L = \begin{cases} 0, & x > 0.5 \\ 1, & x \leq 0.5 \end{cases} \tag{12}$$

$$U = \begin{cases} 1, & x \geq 0.5 \\ 0, & x < 0.5 \end{cases} \tag{13}$$

3.2. Trapezoid Function

Trapezoid function one of the widely used fuzzy sets. Therefore, this set is used in this article. A function is created by using these sets, and it is called as trap(.) in this article. The fuzzy set used to create this function is shown in Fig. 4.

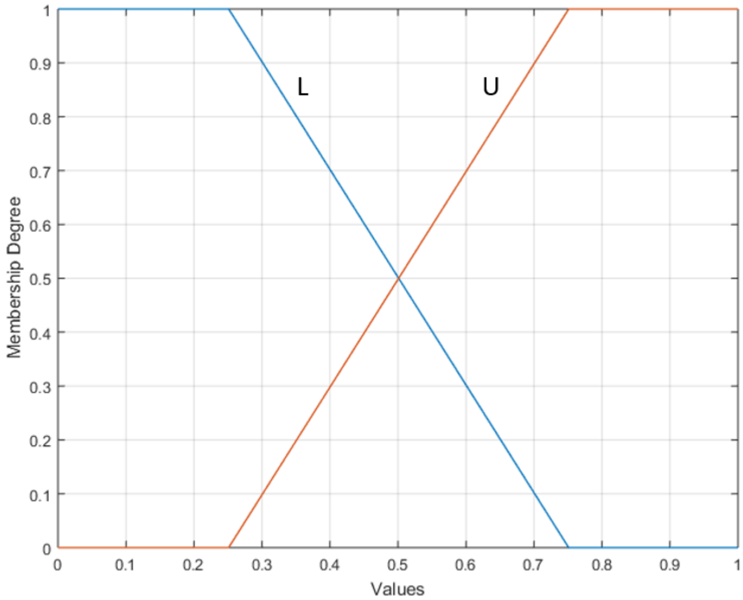


Fig. 4. The used trapezoid fuzzy set.

The mathematical definition of the trap(.) function is given in Eq. 14-15.

$$L = \begin{cases} 1, & x \leq 0.25 \; and \; 0.5 < x \leq 0.75 \\ 0, & 0.25 < x \leq 0.5 \end{cases} \tag{14}$$

$$U = \begin{cases} 1, & x \geq 0.5 \\ 0, & x < 0.5 \end{cases} \tag{15}$$

3.3. Stair Function

In this section, a novel fuzzy set is obtained by modifying the trapezoid fuzzy set, and this function is called a stair. The trapezoid fuzzy set used by stair function is shown in Fig. 5.

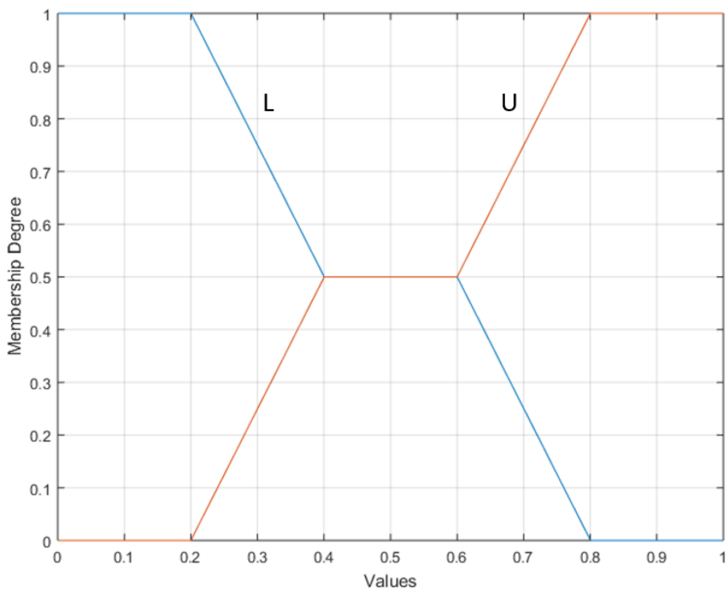


Figure 5. The used stair fuzzy set.

The mathematical definition of the stair(.) function is given Eq. 16-17.

$$L = \begin{cases} 1, & x \leq 0.6 \\ 0, & x > 0.6 \end{cases} \quad (16)$$

$$U = \begin{cases} 1, & x \geq 0.4 \\ 0, & x < 0.4 \end{cases} \quad (17)$$

3.4. Butterfly Function

Here, this function is called a butterfly because it looks like a butterfly shape. This fuzzy set is a modified version of the trapezoid fuzzy set. This function is called as but(.), and the graphical outline of this set is shown in Fig. 6.

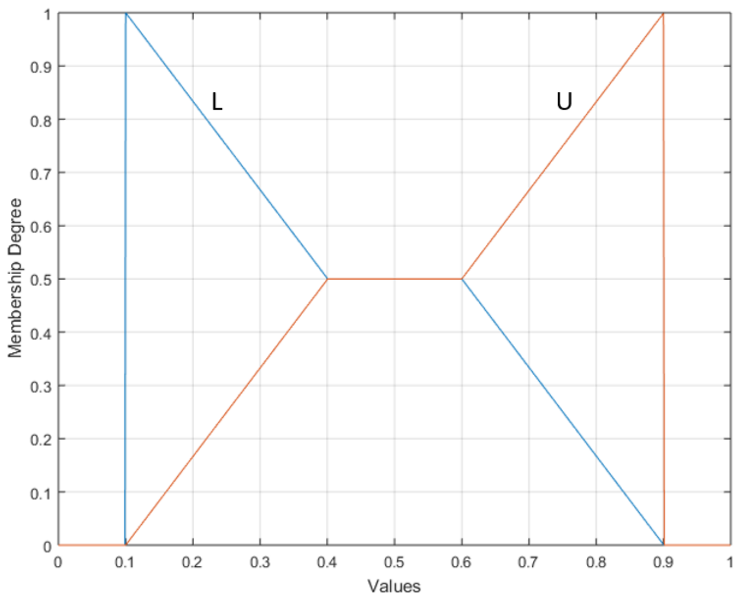


Figure 6. The fuzzy set of butterfly function.

The mathematical definition of the but(.) is given in Eq. 18-19.

$$L = \begin{cases} 1, 0.1 < x \leq 0.6 \\ 0, x \leq 0.1 \ and \ x \geq 0.9 \end{cases} \quad (18)$$

$$U = \begin{cases} 1, 0.4 < x \leq 0.9 \\ 0, x < 0.4 \ and \ x > 0.9 \end{cases} \quad (19)$$

4. The Proposed Fuzzy-Based Descriptors

In this paper, 4 novel fuzzy kernels are presented to increase the recognition rate. The proposed kernels are applied to descriptors that are widely used in the literature and have multiple variations. There are LBP and LGS. Also, a novel textural image recognition architecture for LBP and LGS is proposed in this article, and 32 novel methods are presented using this architecture. This architecture consists of lower and upper image calculation using fuzzy kernels, histogram extraction, using maximum pooling for feature reduction and classification. The general block diagram of the presented architecture is shown in Fig. 7 [20].

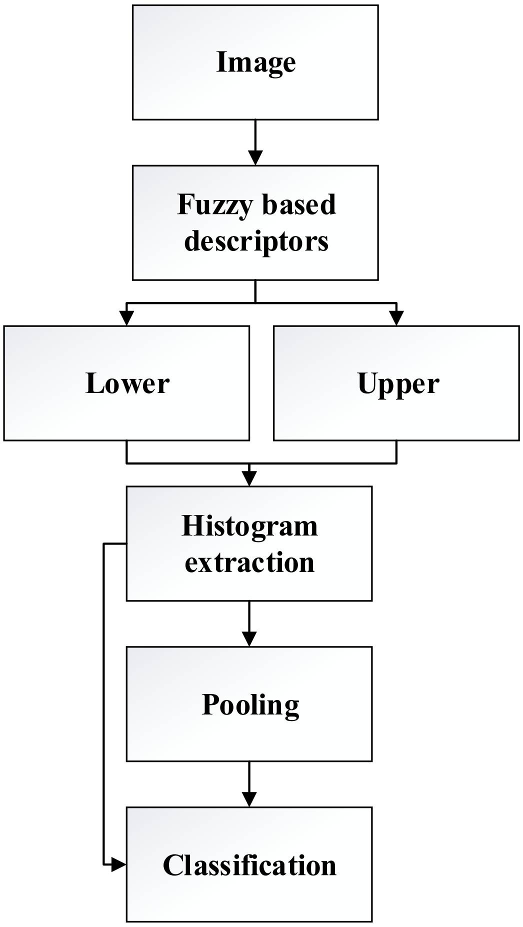


Figure 7. Block diagram of the proposed fuzzy architecture.

The steps of the presented architecture are given below.
Step 1: Load image.
Step 2: Choose LBP or LGS pattern.
Step 3: Calculate differences of neighborhood pixels.
Step 4: Select one of the fuzzy function to extract binary features of the image.
Step 5: Extract histograms.
Step 6: Reduce feature using pooling. This step is used for the pooled method.
The pseudocode of the used maximum pooling is given Algorithm 1.

Algorithm 1. Maximum pooling

Input: The feature vector is f with the size of 512.
Output: Pooled feature vector is pf with the size of 256.
1: counter=1;
2: **for** i=1 to 512 steps by 2 **do**
3: $block_1 = f_i$
4: $block_2 = f_{i+1}$
5: $pf_{counter} = \max(block)$
6: $counter = counter + 1$
7: **endfor**

Step 7: Classify these features.

The methods given above are outlined in the sub-sections of this chapter, and abbreviation of the methods and images are listed in Table 1.

Table 1. Abbreviation of the methods and images.

No	Name	Abbreviation	Dimension
1	Lover image	L	M x N
2	Upper image	U	M x N
3	Fuzzy LBP	FLBP	512
4	Fuzzy LBP of L	FLBPL	256
5	Fuzzy LBP of U	FLBPU	256
6	Pooled fuzzy LBP	PFLBP	256
7	Fuzzy LGS	FLGS	512
8	Fuzzy LGS of L	FLGSL	256
9	Fuzzy LGS of U	FLGSU	256
10	Pooled fuzzy LGS	PFLGS	256

In Table 1, M and N are the width and height of the cover image.

4.1. Fuzzy LBP

LBP is the most frequently used and most variant local descriptor in the literature. The general steps of the proposed fuzzy LBP methods are as follows.

Step 1: Load image.

Step 2: Divide the image into 3 x 3 size blocks.

Step 3: Calculate differences array using Eq. 20.

$$f_1 = I_{i,j} - I_{i+1,j+1}, f_2 = I_{i,j+1} - I_{i+1,j+1} ,$$
$$f_3 = I_{i,j+2} - I_{i+1,j+1} , f_4 = I_{i+1,j} - I_{i+1,j+1}$$
$$f_5 = I_{i+1,j+2} - I_{i+1,j+1}, f_6 = I_{i+2,j} - I_{i+1,j+1},$$
$$f_7 = I_{i+2,j+1} - I_{i+1,j+1}, f_8 = I_{i+2,j+2} - I_{i+1,j+1} \tag{20}$$

f vector defines differences of neighborhood pixels.

Step 4: Normalize these differences.

$$x = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{21}$$

where x vector defines normalized differences.

Step 5: Calculate the lower and upper values of this block. These are *tri(.),tra(.),stair(.)* and *but(.)*

$$[L, U] = fuzzy\ function(x) \tag{22}$$

Where *fuzzy function(x)* defines *tri(x),trap(x),stair(x)* or *but(x).*

Step 6: Extract histogram of L and U.

Step 7: In FLBP (Fuzzy LBP), combine histograms of the L and U, use histogram of L for FLBPL, use histogram of U FLBPU, and finally use maximum pooling for PFLBPB (Pooled FLBP).

4.2. Fuzzy LGS

LGS is the first well-known graph-based texture operator in the literature and uses the signum function for binary feature extraction. New fuzzy LGSs are proposed using fuzzy functions in this article. The proposed fuzzy LGS steps are as follows.

Step 1: Load image.

Step 2: Divide the image into 3 x 4 size overlapping blocks.

Step 3: Calculate differences vector using Eq. 23.

$$f_1 = I_{i+1,j+1} - I_{i,j}, f_2 = I_{i,j} - I_{i+2,j} ,$$
$$f_3 = I_{i+2,j} - I_{i+1,j+1} , f_4 = I_{i+1,j+1} - I_{i+1,j+2}$$
$$f_5 = I_{i+1,j+2} - I_{i,j+3}, f_6 = I_{i,j+3} - I_{i+2,j+3},$$
$$f_7 = I_{i+2,j+3} - I_{i+1,j+2}, f_8 = I_{i+1,j+2} - I_{i+1,j+1} \quad (23)$$

Step 4: Normalize differences using Eq. 14.
Step 5: Calculate L and H images using Eq. 15.
Step 6: Extract histogram of the L and U.
Step 7: Use the selected feature.
5. Experimental Results and Discussions
In this article, six databases were used to test the performance of fuzzy coding-based methods. Three of these databases are face datasets, and the other three are texture datasets. The properties of the used datasets are given in Table 2 [21-24].

Table 2. The attributes of the facial and textural databases for performance evaluation.

No	Database Name	Classes	Samples	Total	Resolution	Format	Type
1	AT&T	30	10	300	92 x 112	JPEG	Face
2	Face94	30	10	300	180 x 200	JPEG	Face
3	AR	31	10	310	768 x 576	RAW	Face
4	Outex TC 00000	24	20	480	128 x 128	RAS	Texture
5	Outex TC 00001	24	88	2112	64 x 64	RAS	Texture
6	Outex TC 00013	68	20	1360	128 x 128	BMP	Texture

Performance analyzes were carried out using the above 6 different databases. Accuracy (Acc) was used to obtain the experimental results. The mathematical definition of Acc is given in Eq. 24 [25].

$$Acc\ (\%) = \frac{Number\ of\ true\ predicted\ samples}{Number\ of\ total\ samples} \text{ x } 100 \quad (24)$$

In the final phase, the quadratic kernel Support Vector Machine (SVM) [26, 27] was used to classify the methods. The attributes of the SVM are listed in Table 3.

Table 3. Attributes of the SVM

Cross-validation folds	10
Kernel function	Quadratic
Kernel scale	Automatic
Box constraint level	1
PCA	Disable
Multiclass method	One vs All
Standardize data	True

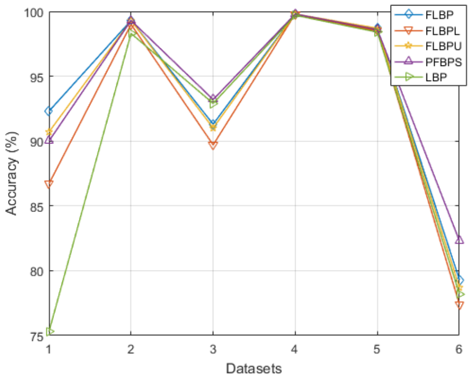
5.1. Experiments of LBP and Fuzzy coding based LBPs

In this article, 16 novel fuzzy LBP methods are proposed using 4 fuzzy kernels. In this section, the fuzzy LBP method was evaluated, and the experiments of the LBP were discussed. These experiments are listed in Table 4.

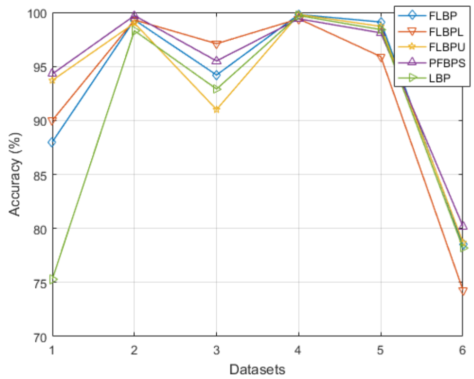
Table 4. Performance analysis of LBP methods

		FLBP	FLBPL	FLBPU	PFLBP	LBP
Triangle	1	92.3	86.7	90.7	90.0	75.3
	2	99.3	99.0	99.3	99.3	98.3
	3	91.3	89.7	91.0	93.2	92.9
	4	99.8	99.8	99.8	99.8	99.7
	5	98.7	98.5	98.7	98.6	98.4
	6	79.3	77.4	78.7	82.3	78.2
Trapezoid	1	92.3	90.0	93.7	94.3	75.3
	2	99.3	99.3	99.0	99.7	98.3
	3	91.3	97.1	91.0	95.5	92.9
	4	99.8	99.4	99.8	99.4	99.7
	5	98.7	95.9	98.7	98.1	98.4
	6	79.3	74.3	78.7	80.2	78.2
Stair	1	92.3	84.0	89.0	93.0	75.3
	2	99.3	97.7	96.3	98.0	98.3
	3	91.3	94.8	91.3	96.8	92.9
	4	99.8	99.8	99.8	99.8	99.7
	5	98.7	97.9	98.2	98.7	98.4
	6	79.3	76.4	75.9	81.0	78.2
Butterfly	1	93.3	90.3	88.0	92.7	75.3
	2	98.0	98.0	98.0	98.0	98.3
	3	94.8	94.2	91.0	94.2	92.9
	4	100	100	99.8	100	99.7
	5	99.5	99.0	99.2	99.2	98.4
	6	81.8	79.3	78.6	82.1	78.2

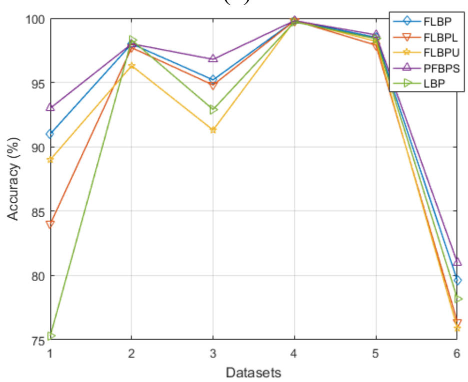
In Table 4, the best values are shown in bold font. If FLBP, FLBPL, FLBPU, and PFLBP are evaluated, the best 11 values in FLBP, 4 best values in FLBPL, 5 best values in FLBPU, 14 best values in PFLBP, and 2 best values in LBP are obtained. It has been observed that the proposed methods have a positive effect on the face and textural image recognition phase. The Fig.8. show the comparison of fuzzy kernels graphically.



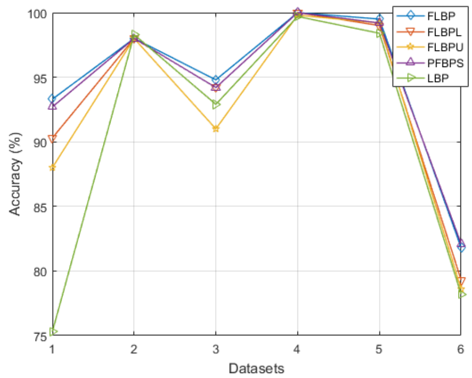
(a)



(b)



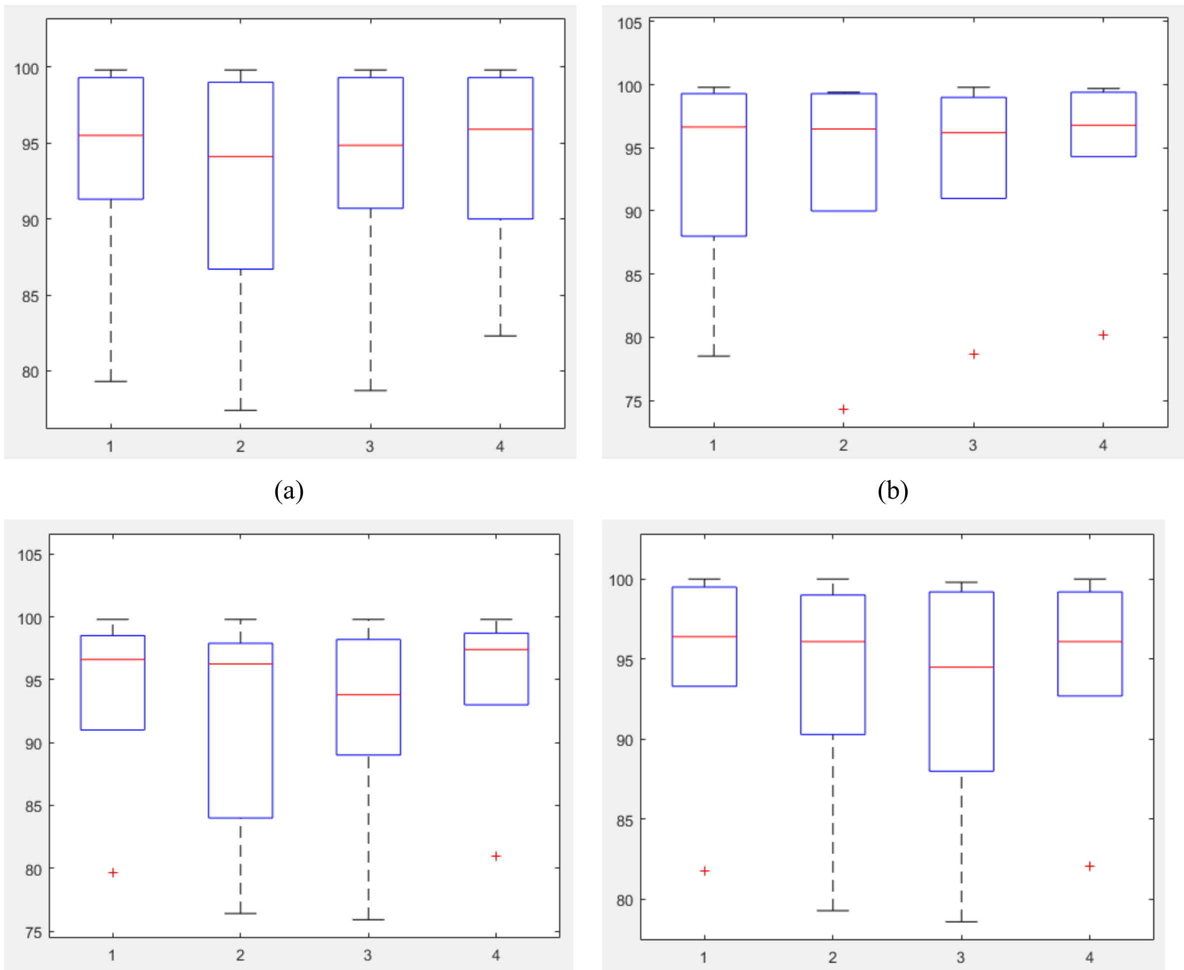
(c)



(d)

Figure 8. Performance comparison of the fuzzy functions based LBPs and LBP (a) triangle function based LBPs and LBP (b) trapezoid function based LBPs and LBP (c) stair function based LBPs and LBP (d) butterfly function LBPs and LBP.

As shown in Fig. 8, it is generally observed that the butterfly kernel-based FLBP method is more successful than the other methods. In addition, boxplot analyzes of triangle, trapezoid, stair, and butterfly functions were performed. The results are shown in Fig. 9.



(a) (b)

(c) (d)

Figure 9. Boxplot analysis of fuzzy coding based LBPs (a) Triangle function based LBPs (b) Trapezoid function LGSs (c) Stair function based LGSs (d) Butterfly function based LGSs.

The performance of the 4 fuzzy functions used is evaluated. The average performance of fuzzy coding based LBP methods for the 6 databases used is given in Table 5.

Table 5. The average accuracy of the fuzzy function for LBP.

Triangle	Trapezoid	Stair	Butterfly
93.05	93.45	92.93	93.71

The average performance of the Signum kernel LBP method for 6 databases is calculated as 90.47%. As a result, all fuzzy kernels have a positive effect on the recognition of the LBP.

5.2. Experiments of LGS and Fuzzy coding based LGSs

LGS is the first known, frequently used, and most variant graph-based descriptor in the literature. Experimental results of proposed fuzzy-based LGS methods are given in this section.

Table 6. Performance analysis of LGS methods

		FLGS	$FLGS_L$	$FLGS_U$	PFLGS	LGS
Triangle	1	**95.7**	94.3	92.0	93.7	92.3
	2	**100.0**	**100.0**	**100.0**	**100.0**	99.0
	3	**97.4**	96.8	97.1	97.1	96.8
	4	**100.0**	**100.0**	99.6	99.6	99.7
	5	**98.9**	98.6	98.4	98.8	98.0
	6	79.8	79.7	78.5	**80.3**	79.6
Trapezoid	1	**95.0**	83.0	81.3	93.0	92.3
	2	**100.0**	99.7	99.7	**100.0**	99.0
	3	97.4	95.8	94.2	98.1	96.8
	4	**100.0**	99.8	99.6	**100.0**	99.7
	5	**98.8**	98.2	95.5	98.1	98.0
	6	77.8	**80.1**	70.4	76.3	79.6
Stair	1	89.3	90.7	**91.0**	87.7	92.3
	2	**100.0**	**100.0**	**100.0**	**100.0**	99.0
	3	97.1	94.5	94.5	**97.4**	96.8
	4	**100.0**	99.8	**100.0**	**100.0**	99.7
	5	98.5	98.5	97.7	**98.7**	98.4
	6	78.1	76.3	76.8	**82.6**	79.6
Butterfly	1	**96.3**	93.3	91.3	95.3	92.3
	2	**100.0**	**100.0**	**100.0**	**100.0**	99.0
	3	97.4	95.8	94.8	**97.7**	96.8
	4	**100.0**	**100.0**	**100.0**	**100.0**	99.7
	5	99.0	98.2	98.5	**99.1**	98.0
	6	**79.9**	79.0	77.1	78.5	79.6

In Table 6, the best values are shown in bold font. If FLGS, FLGSL, FLGSU, and PFLGS are evaluated, the best 15 values in FLGS, 6 best values in FLGSL, 6 best values in FLGSU, and 13 best values in PFLGS are obtained. Performance comparisons of the fuzzy kernel-based LGS methods and LGS are shown in Fig. 10. According to the results of databases, the best score is achieved for the butterfly kernel-based FLGS method.

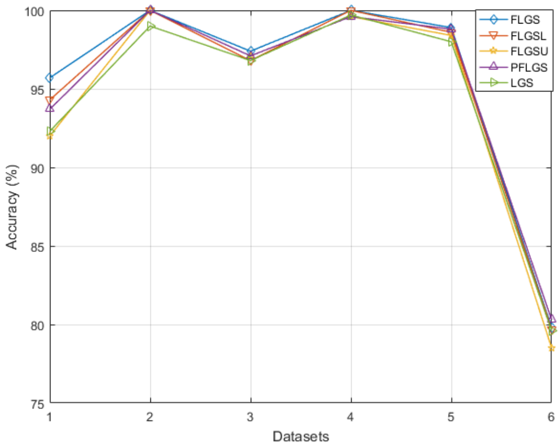
In addition, boxplot analyzes of triangle, trapezoid, stairs, and butterfly functions were performed, and the results of these are shown in Fig. 11.

The performance of the 4 fuzzy functions is evaluated. The average performance of fuzzy kernel-based LGS methods for the 6 databases is listed in Table 7.

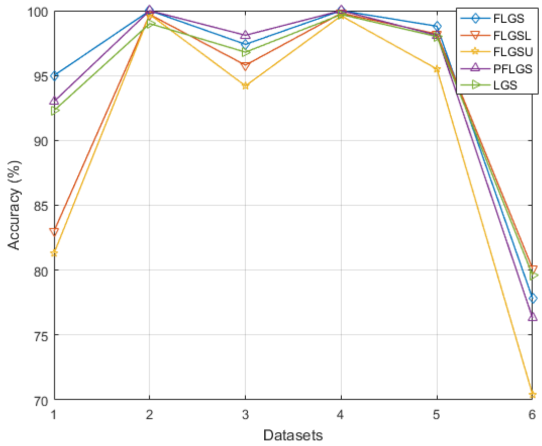
Table 7. The average accuracy of the fuzzy function for LGS methods.

Triangle	**Trapezoid**	**Stair**	**Butterfly**
94.85	92.99	93.72	94.63

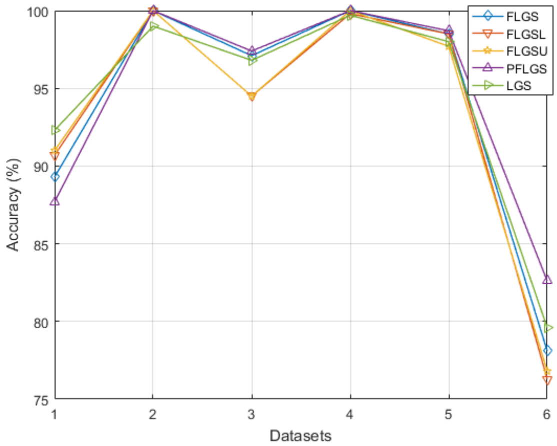
The average performance of Signum kernel LGS method for 6 databases is calculated as 94.23%. It has been observed that the triangle and butterfly function has a positive effect on LGS.



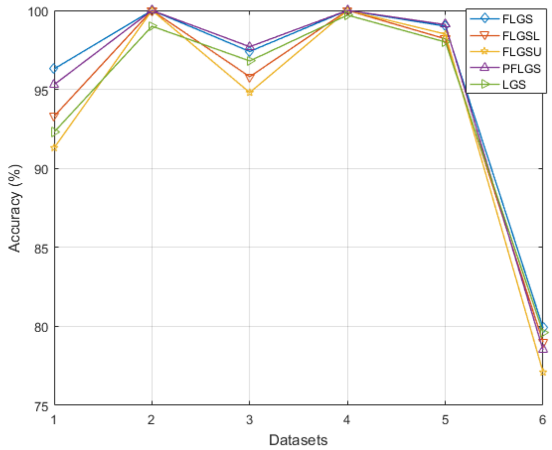
(a)



(b)

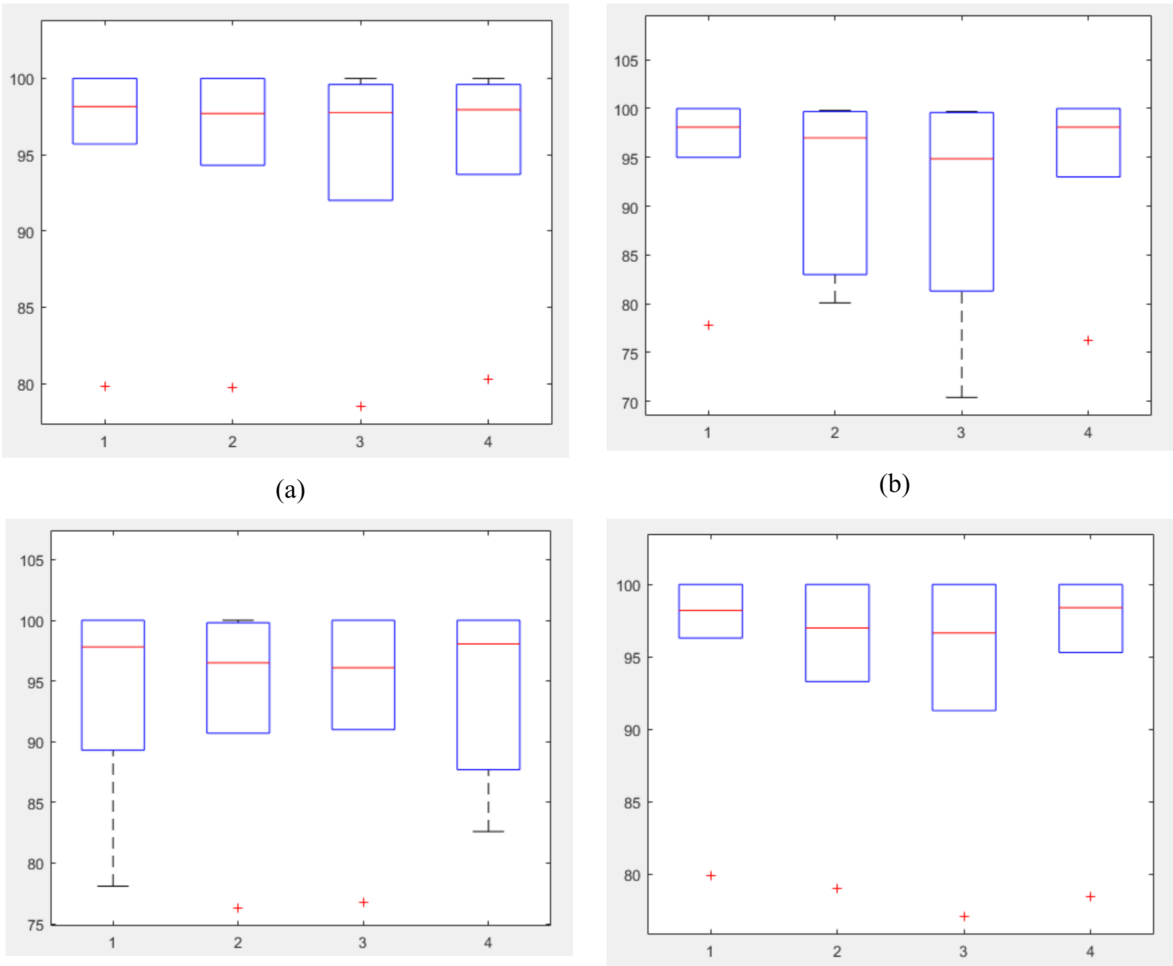


(c)



(d)

Figure 10. Performance comparison of the fuzzy functions based LGSs and LGS (a) triangle function based LGSs and LGS (b) trapezoid function based LGSs and LGS (c) stair function based LGSs and LGS (d) butterfly function LGSs and LGS.



(a) (b)

(c) (d)

Figure 11. Boxplot analysis of fuzzy kernel-based LGS methods (a) Triangle function based LGSs (b) Trapezoid function LGSs (c) Stair function based LGSs (d) Butterfly function based LGSs.

6. Conclusions and recommendations

In this study, a new fuzzy-based texture image recognition architecture is presented. In this architecture, binary feature extraction is performed using fuzzy-based functions instead of signum function. Fuzzy-based functions are design using fuzzy sets and rules. Triangle and trapezoid fuzzy sets are widely used in the literature. Stair and butterfly fuzzy sets are presented as modifications of the trapezoid fuzzy set in this study. Fuzzy kernels are proposed using these fuzzy sets. The proposed fuzzy kernels are applied to LBP and LGS methods, and 16 new descriptors are defined for each method. The performance of the proposed methods is tested using 3 face databases (AT & T, Face94, AR) and 3 texture databases (Outex TC 00000, Outex TC 00001, Outex TC 00013). The average accuracy rate is calculated as 90.47 with signum kernel-based LBP. In fuzzy kernel-based LBP, the average accuracy rates for the triangle, trapezoid, stair, butterfly are assessed as 93.05, 93.45, 92.93, 93.71, respectively, for all databases used in the study. The average accuracy rate is 94.23 with signum kernel-based LGS. In fuzzy kernel-based LGS, the average accuracy rates are 94.85, 92.99, 93.72, 94.63, respectively. Also, the best methods are butterfly kernel FLBP and FLGS methods. When the results are evaluated, all fuzzy kernels are observed to improve the recognition rate of LBP. The recognition rates of the triangle and butterfly fuzzy kernels increase in LGS. At the same time, experimental results and discussions clearly show that fuzzy coding has a positive effect on both methods. It is shown in this article that new fuzzy coding methods can be developed using different fuzzy kernels in future studies, and these methods can be applied to other local descriptors in the literature.

References

[1] Li J, Sang N, Gao C. LEDTD: Local edge direction and texture descriptor for face recognition. Signal Processing: Image Communication. 2016;41:40-5.

[2] Nguyen K, Fookes C, Jillela R, Sridharan S, Ross A. Long range iris recognition: A survey. Pattern Recognition. 2017;72:123-43.

[3] Lekdioui K, Messoussi R, Ruichek Y, Chaabi Y, Touahni R. Facial decomposition for expression recognition using texture/shape descriptors and SVM classifier. Signal Processing: Image Communication. 2017;58:300-12.

[4] Qin C, Chen X, Luo X, Zhang X, Sun X. Perceptual image hashing via dual-cross pattern encoding and salient structure detection. Information Sciences. 2018;423:284-302.

[5] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence. 2002;24:971-87.

[6] Ojala T, Pietikäinen M, Harwood D. A comparative study of texture measures with classification based on featured distributions. Pattern recognition. 1996;29:51-9.

[7] Ojala T, Valkealahti K, Oja E, Pietikäinen M. Texture discrimination with multidimensional distributions of signed gray-level differences. Pattern Recognition. 2001;34:727-39.

[8] Zhou H, Wang R, Wang C. A novel extended local-binary-pattern operator for texture analysis. Information Sciences. 2008;178:4314-25.

[9] Fathi A, Naghsh-Nilchi AR. Noise tolerant local binary pattern operator for efficient texture analysis. Pattern Recognition Letters. 2012;33:1093-100.

[10] Pei H, Yanqing S, Chaowei T, Siman Z. Center-symmetric local binary pattern based on weighted neighbor contribution. Optik. 2016;127:11599-606.

[11] Chakraborty S, Singh SK, Chakraborty P. Local quadruple pattern: A novel descriptor for facial image recognition and retrieval. Computers & Electrical Engineering. 2017;62:92-104.

[12] Bashier HK, Hoe LS, Hui LT, Azli MF, Han PY, Kwee WK, et al. Texture classification via extended local graph structure. Optik. 2016;127:638-43.

[13] Abdullah MFA, Sayeed MS, Muthu KS, Bashier HK, Azman A, Ibrahim SZ. Face recognition with symmetric local graph structure (slgs). Expert Systems with Applications. 2014;41:6131-7.

[14] Rakshit RD, Nath SC, Kisku DR. Face identification using some novel local descriptors under the influence of facial complexities. Expert Systems with Applications. 2018;92:82-94.

[15] Abusham EE, Bashir HK. Face recognition using local graph structure (LGS). International Conference on Human-Computer Interaction: Springer; 2011. p. 169-75.

[16] Ahonen T, Hadid A, Pietikäinen M. Face recognition with local binary patterns. European conference on computer vision: Springer; 2004. p. 469-81.

[17] Tuncer T, Dogan S. Pyramid and multi kernel based local binary pattern for texture recognition. Journal of Ambient Intelligence and Humanized Computing. 2020;11:1241-52.

[18] Tuncer T, Dogan S, Ertam F, Subasi A. A novel ensemble local graph structure based feature extraction network for EEG signal analysis. Biomedical Signal Processing and Control. 2020;61:102006.

[19] Dong S, Yang J, Chen Y, Wang C, Zhang X, Park DS. Finger Vein Recognition Based on Multi-Orientation Weighted Symmetric Local Graph Structure. Ksii Transactions on Internet & Information Systems. 2015;9.

[20] Vipparthi SK, Nagar SK. Local extreme complete trio pattern for multimedia image retrieval system. International Journal of Automation and Computing. 2016;13:457-67.

[21] Samaria FS, Harter AC. Parameterisation of a stochastic model for human face identification. Proceedings of 1994 IEEE workshop on applications of computer vision: IEEE; 1994. p. 138-42.

[22] Libor Spacek's Facial Image Database, Face94 Database, http://cswww.essex.ac.uk/mv/allfaces/faces94.html (accessed June 1, 2018).

[23] Martinez AM, Kak AC. Pca versus lda. IEEE transactions on pattern analysis and machine intelligence. 2001;23:228-33.

[24] Ojala T, Maenpaa T, Pietikainen M, Viertola J, Kyllonen J, Huovinen S. Outex-new framework for empirical evaluation of texture analysis algorithms. Object recognition supported by user interaction for service robots: IEEE; 2002. p. 701-6.

[25] Tuncer T, Dogan S, Pławiak P, Acharya UR. Automated arrhythmia detection using novel hexadecimal local pattern and multilevel wavelet transform with ECG signals. Knowledge-Based Systems. 2019;186:104923.

[26] Vapnik V. The support vector method of function estimation. Nonlinear Modeling: Springer; 1998. p. 55-85.

[27] Vapnik V. The nature of statistical learning theory: Springer science & business media; 2013.