Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-023-04612-x

ORIGINAL RESEARCH

Automated facial expression recognition using novel textural transformation

Turker Tuncer[1] · **Sengul Dogan**[1] · **Abdulhamit Subasi**[2,3]

Received: 10 October 2020 / Accepted: 4 April 2023

Abstract
Facial expressions demonstrate the important information about our emotions and show the real intentions. In this study, a novel texture transformation method using graph structures is presented for facial expression recognition. Our proposed method consists of five steps. First the face image is segmented and resized. Then the proposed graph-based texture transformation is used as feature extractor. The exemplar feature extraction is performed using the proposed deep graph texture transformation. The extracted features are concatenated to obtain one dimensional feature set. This feature set is subjected to maximum pooling and principle component analysis methods to reduce the number of features. These reduced features are fed to classifiers and we have obtained the highest classification accuracy of 97.09% and 99.25% for JAFFE and TFEID datasets respectively Moreover, we have used CK + dataset to obtain comparison results and our textural transformation based model yielded 100% classification accuracy on the CK + dataset. The proposed method has the potential to be employed for security applications like counter terrorism, day care, residential security, ATM machine and voter verification.

Keywords Facial expression · Texture transformation · Textural feature extraction · Emotion detection · Facial expression recognition (FER)

1 Introduction

Facial expressions contain important information about the mood, mind, emotion and pain of people as a reaction of the brain to various situations (Farajzadeh and Hashemzadeh 2018; Yaddaden et al. 2018). The facial expression recognition (FER) is widely used in many fields such as biometrics and digital forensics to reveal the human emotion from the face image. The performance on the detection of people's mood gets intensified by the inclusion of facial areas such as face, eye, mouth and nose using advanced image processing techniques. Humans have various expressions like neutral, anger, happiness, sadness, fear, and surprise (Owusu et al. 2014; Shan et al. 2009; Zhang et al. 1996). These expressions play a major role in people's communications. In terms of digital forensics, the situations such as being excited or nervous include important information on the psychology of guilt. For this purpose, many algorithms have been developed to provide human machine interaction. The basic building blocks of these programs include face detection, facial local feature extraction, distinctive feature selection, and classification (Cohen et al. 2003; Fasel and Luettin 2003; Lien et al. 1998). The human brain reflects emotions using facial expressions and these recognition systems also classify the facial expressions (Matsugu et al. 2003; Shan et al. 2009; Virrey et al. 2019). Many facial expression recognition methods have been proposed based on expression recognition (feelings, opinion, thought), non-verbal communication (gestures and facial expression, signs), verbal communication (listener reactions), and psychological activities (pain and fatigue) (Moore and Bowden 2011; Zhao and Pietikainen 2007).

✉ Abdulhamit Subasi
abdulhamit.subasi@utu.fi; absubasi@effatuniversity.edu.tr

Turker Tuncer
turkertuncer@firat.edu.tr

Sengul Dogan
sdogan@firat.edu.tr

[1] Department of Digital Forensics Engineering, Technology Faculty, Firat University, Elazig, Turkey

[2] Institute of Biomedicine, Faculty of Medicine, University of Turku, Turku, Finland

[3] Department of Computer Science, College of Engineering, Effat University, Jeddah 21478, Saudi Arabia

Published online: 25 April 2023

The image processing and pattern recognition methods have been widely used for FER. Farajzadeh and Hashemzadeh (2018) proposed a new face recognition model using five FER image datasets. The local binary pattern (LBP) and histogram of oriented gradients (HOG) techniques are used for feature extraction. They worked in two phases using LBP and HOG methods. The HOG based method achieved better results than LBP based method. Revina and Sam Emmanuel (2018) presented a modified facial expression recognition with local directional number (LDN) pattern, dominant gradient local ternary pattern (DGLTP). In their study, two FER image datasets are used to obtain numerical results of the method. Li et al. (2018) proposed a novel end-to-end learnable local binary pattern network using LBP network for face detection. The comparison results are given in terms of equal error rate (EER) and half total error rate (HTER) in their study. Turan and Lam (2018) proposed a structure with local descriptors based on histogram. The comparison results are given in terms of accuracy. Zhang and Hua (2015) analyzed the driver fatigue recognition based on LBP and boost local binary pattern. They have used support vector machine (SVM) for classification and recognition rate as a performance measure. Ertugrul et al. (2018) proposed a model of Kinship patterns for facial expressions. In their study, they used two (smile and disgust) databases. They used accuracy rates for performance evaluation. Chao et al. (2015) presented a model using expression specific local binary pattern and class regularized locality preserving projection. In their study, recognition rate of JAFFE (Lyons et al. 1999) dataset is evaluated. Guo et al. (2018) analyzed the expression dependent susceptibility to study the face distortions and evaluated the facial expressions. The accuracy, reaction time, expression intensity, fixation duration and numbers are used to evaluate the performance. Moore and Bowden (2011) proposed local Gabor binary pattern to study the facial expression. They have used three FER image datasets and presented their experimental results. Savchenko (2021) used lightweight neural networks for FER detection. They selected the AFEW and the VGAF datasets. Vo et al. (2020) proposed a pyramid with super-resolution (PSR) method for wild FER detection. They used three datasets (FER+, RAF-DB and AffectNet). They presented the results according to confusion matrix, accuracy. Revina and Emmanual (2019) applied multi-support vector neural network to FER images. Cohn-Kanade AU-Coded Expression and JAFFE databases were used for experimental results. They used accuracy, true positive rate and false positive rate metrics to evaluate the performance. Abate et al. (2019) proposed a method for clustering facial attributes. For this purpose, they utilized principal component analysis, convolutional neural networks. The experimentations were carried out using CelebA dataset.

Facial expression recognition is a major problem for image processing and pattern recognition. In the literature, many methods have been presented to solve this problem. We have presented a texture based method to detect the FER automatically which is seen in Fig. 1. The proposed scheme is as follows:

- In the preprocessing phase, the face area of the image is segmented and resized.
- The segmented image is divided into pieces to implement exemplar method.
- The proposed texture transformation is applied on each piece to extract the features.
- The extracted features are concatenated to obtain the final feature.
- PCA and one dimensional (1D) maximum pooling are used for feature reduction.
- SVM and LDA are used for automated classification.

The major contributions of this paper are given below.

- Most of the image descriptors available in the literature such as local binary pattern (LBP) (Ojala et al. 2002), local quadruple pattern (LQPAT) (Chakraborty et al.

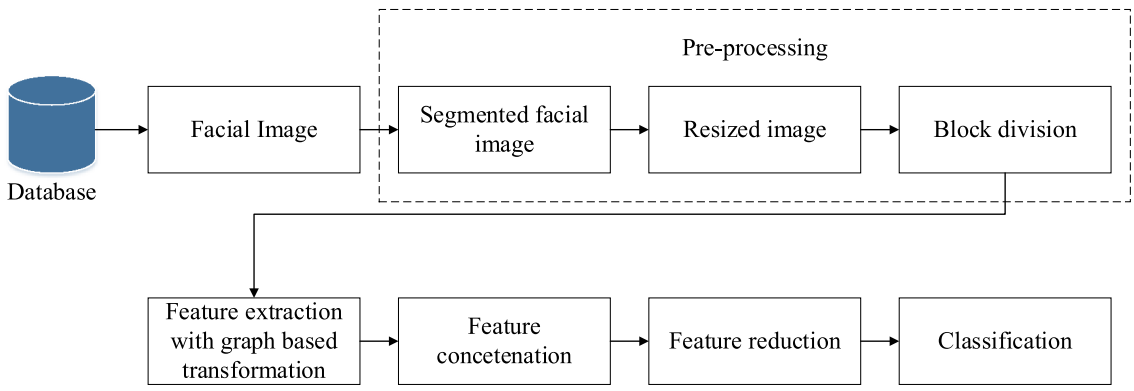


Fig. 1 Block diagram of the proposed method

2017), dual cross pattern (DCP) (Ding et al. 2016), and local ternary pattern (LTP) (Tan and Triggs 2010) use single pattern and are applied onto each block for feature extraction. Therefore, these descriptors could not achieve successful results for some problems like FER which involve heterogeneous dataset using single pattern. To solve this problem, a novel graph based texture transform is proposed. This transform uses 15 patterns from the five levels for feature extraction by utilizing the graph based structures.

- A novel cognitive and lightweight deep feature extraction network is proposed in this paper. This method is cognitive because optimization and weight updating methods are not used. The proposed method is fast as it does not involve any complex calculations.
- A novel hybrid feature reduction is proposed in this method. In here, maximum pooling and PCA are utilized together for feature reduction.

2 The proposed graph based texture transformation

The image descriptors have been widely used for face recognition and FER methods (Ahonen et al. 2006; Hernández et al. 2007; Kabir et al. 2010; Rivera et al. 2013; Sadeghi and Raie 2019). A novel graph based texture descriptor is employed using variable patterns and it extracts features up to five levels. The basic geometric shapes are utilized as patterns. The proposed transformation can be used with variable block sizes. In this study, overlapping blocks with size of 3×3, 5×5, 7×7 are utilized.

Our proposed transformation is illustrated using an example as shown below. A sample block size of 3×3. Equation (1) shows the mathematical description of the 3×3 block.

$$block = \begin{bmatrix} Im_{i,j} & Im_{i,j+1} & Im_{i,j+2} \\ Im_{i+1,j} & Im_{i+1,j+1} & Im_{i+1,j+2} \\ Im_{i+2,j} & Im_{i+2,j+1} & Im_{i+2,j+2} \end{bmatrix} \quad (1)$$

where $Im_{i,j}$ is i^{th} and j^{th} pixel value of the image. The proposed graph based transformation is explained using a 3×3 sample block and patterns for 3×3, 5×5 and 7×7 size of blocks are shown in the figures by using →, → and → arrows. Also, the signum function is used as binary feature extraction function and its mathematical description is shown in Eq. (2).

$$\mathrm{Sig}(a,b) = \begin{cases} 0, a < b \\ 1, a \geq b \end{cases} \quad (2)$$

where Sig(.,.) represents the signum function, a and b are input parameters of the signum function.

2.1 Level 1

In this paper, a graph-based texture transformation of five levels is implemented. In each level, variable shapes are used to create graphs and these graphs are utilized as patterns. In the first level, two graphs are utilized, and 8-bit binary feature values are extracted from each graph. Finally, 16-bit binary features are extracted in level 1.

To obtain the numerical results of the example, a sample block size of 3×3 is used as shown in Fig. 2.

The mathematical description and example of the level 1 are shown in Fig. 3. In level 1, local binary pattern and a graph which is similar to square shape is used.

2.2 Level 2

In this level, 4 pentagons are used for feature extraction. These pentagons are obtained with 90-degree rotation. In the level 2, 20 bit binary features are extracted. The mathematical description and example of the level 2 are shown in Fig. 4.

2.3 Level 3

8-bit binary data are extracted from level 3 which uses two shapes. The shapes used and mathematical expressions of level 3 are shown in Fig. 5.

2.4 Level 4

The triangle shapes are utilized as patterns in this level. Totally, 12 bit values are extracted and description of this level is given in Fig. 6. To extract these bits, 4 triangles at 90-degree rotations are used.

2.5 Level 5

In this level, the lines are used to extract the features. The line patterns are shown in Fig. 7. $3+3+2$ bits are obtained by applying the signum function to the pixels at the ends of the lines. The 8 bits of data are obtained using 8 lines.

Finally, 64 bit features are extracted from each block. The 8 feature images are constructed using these bits. The pseudo code of feature image creation is given Algorithm 1.

Fig. 2 Sample 3×3 block

55	49	41
58	59	44
52	60	48

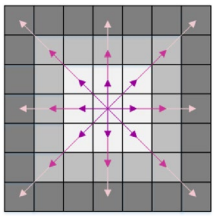
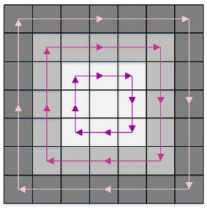
Graph	Mathematical Description	Numerical Example	
	$bit_1 = Sig\left(Im_{i+1,j+1}, Im_{i,j}\right)$	59 >= 55→1	11110111
	$bit_2 = Sig\left(Im_{i+1,j+1}, Im_{i,j+1}\right)$	59 >= 49→1	
	$bit_3 = Sig\left(Im_{i+1,j+1}, Im_{i,j+2}\right)$	59 >= 44→1	
	$bit_4 = Sig\left(Im_{i+1,j+1}, Im_{i+1,j}\right)$	59 >= 48→1	
	$bit_5 = Sig\left(Im_{i+1,j+1}, Im_{i+1,j+2}\right)$	59 >= 60→0	
	$bit_6 = Sig\left(Im_{i+1,j+1}, Im_{i+2,j}\right)$	59 >= 52→1	
	$bit_7 = Sig\left(Im_{i+1,j+1}, Im_{i+2,j+1}\right)$	59 >= 58→1	
	$bit_8 = Sig\left(Im_{i+1,j+1}, Im_{i+2,j+2}\right)$	59 >= 55→1	
	$bit_9 = Sig\left(Im_{i,j}, Im_{i,j+1}\right)$	55 >= 49→1	11000101
	$bit_{10} = Sig\left(Im_{i,j+1}, Im_{i,j+2}\right)$	49 >= 41→1	
	$bit_{11} = Sig\left(Im_{i,j+2}, Im_{i+1,j+2}\right)$	41 >= 44→0	
	$bit_{12} = Sig\left(Im_{i+1,j+2}, Im_{i+2,j+2}\right)$	44 >= 48→0	
	$bit_{13} = Sig\left(Im_{i+2,j+2}, Im_{i+2,j+1}\right)$	48 >= 60→0	
	$bit_{14} = Sig\left(Im_{i+2,j+1}, Im_{i+2,j}\right)$	60 >= 52→1	
	$bit_{15} = Sig\left(Im_{i+2,j}, Im_{i+1,j}\right)$	52 >= 58→0	
	$bit_{16} = Sig\left(Im_{i+1,j}, Im_{i,j}\right)$	58 >= 55→1	

Fig. 3 Mathematical description and examples of the Level 1

Algorithm 1. The pseudo code of feature image construction.

Input: Generated bits using graph based texture transform with size 64 bits.
Output: 8 decimal values (dv) with size of 1 x 8

1: $k = 1$;
2: **for** i=1 to 64 step by 8 **do**
3: $dv_k = 0$;
4: **for** j=0 to 7 **do**
5: $dv_k = dv_k + bit_{i+j} * 2^{7-j}$;
7: **endfor j**
8: $k = k + 1$;
9: **endfor i**

Then, the 8 feature images are constructed using these decimal values and histogram of these images are used as features.

3 The texture transformation based facial expression recognition

In this method, an exemplar facial image recognition method is presented. This algorithm uses a novel graph based texture transformation for feature extraction. The proposed method consists of preprocessing, feature extraction, feature concatenation, feature reduction and classification phases. During the preprocessing step, facial images are segmented to obtain facial area. Then, the segmented image is resized, and this image is divided into 30×30 non-overlapping blocks. The proposed texture transformation is applied to each block for feature extraction. The maximum pooling and PCA are employed for feature reduction together. In this step, 2048×12=24,576 features are reduced to 1024 features using 1D maximum pooling. The pseudo code of 1D maximum pooling is shown in Algorithm 2.

Algorithm 2. The proposed 1D maximum pooling algorithm.

Input: Concatenated feature (CF) with 24,576 dimensions.
Output: Pooled feature (PF) with 1024 dimensions.

1: $k = 1$;
2: **for** i=1 to 24576 step by 24 **do**
3: $PF(k) = \max(CF(i{:}i+23))$ // Divide features into 24 size of windows.
4: $k = k + 1$;
5: **end for i**

In the last phase, LDA and SVM classifiers are used for classification. The steps of the proposed method are given in Table 1. 128 and 1024 features are utilized as inputs for the quadratic kernel SVM and LDA respectively. To obtain 128 features, PCA is applied onto the pooled features.

In this study, LDA and SVM are utilized as classifier and attributes of them are listed in Table 2.

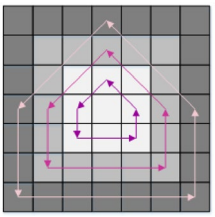
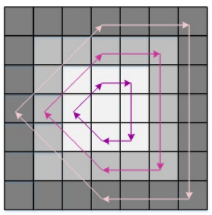
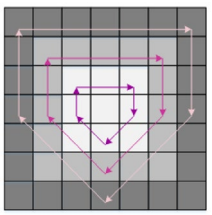
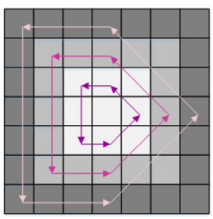
Graph	Mathematical Description	Numerical Example	
	$bit_{17} = Sig\left(Im_{i,j+1}, Im_{i+1,j}\right)$	49 >= 58→0	01110
	$bit_{18} = Sig\left(Im_{i+1,j}, Im_{i+2,j}\right)$	58 >= 52→1	
	$bit_{19} = Sig\left(Im_{i+2,j}, Im_{i+2,j+2}\right)$	52 >= 48→1	
	$bit_{20} = Sig\left(Im_{i+2,j+2}, Im_{i+1,j+2}\right)$	48 >= 44→1	
	$bit_{21} = Sig\left(Im_{i+1,j+2}, Im_{i,j+1}\right)$	44 >= 49→0	
	$bit_{22} = Sig\left(Im_{i+1,j}, Im_{i,j+1}\right)$	58 >= 49→1	11001
	$bit_{23} = Sig\left(Im_{i,j+1}, Im_{i,j+2}\right)$	49 >= 41→1	
	$bit_{24} = Sig\left(Im_{i,j+2}, Im_{i+2,j+2}\right)$	41 >= 48→0	
	$bit_{25} = Sig\left(Im_{i+2,j+2}, Im_{i+2,j+1}\right)$	48 >= 60→0	
	$bit_{26} = Sig\left(Im_{i+2,j+1}, Im_{i+2,j+1}\right)$	60 >= 58→1	
	$bit_{27} = Sig\left(Im_{i+2,j+1}, Im_{i+1,j}\right)$	60 >= 58→1	10100
	$bit_{28} = Sig\left(Im_{i+1,j}, Im_{i,j}\right)$	58 >= 59→0	
	$bit_{29} = Sig\left(Im_{i,j}, Im_{i,j+2}\right)$	55 >= 41→1	
	$bit_{30} = Sig\left(Im_{i,j+2}, Im_{i+1,j+2}\right)$	41 >= 44→0	
	$bit_{31} = Sig\left(Im_{i+1,j+2}, Im_{i+2,j+1}\right)$	44 >= 60→0	
	$bit_{32} = Sig\left(Im_{i+1,j+2}, Im_{i,j+1}\right)$	44 >= 49→0	00101
	$bit_{33} = Sig\left(Im_{i,j+1}, Im_{i,j}\right)$	49 >= 59→0	
	$bit_{34} = Sig\left(Im_{i,j}, Im_{i+2,j}\right)$	55 >= 52→1	
	$bit_{35} = Sig\left(Im_{i+2,j}, Im_{i+2,j+1}\right)$	52 >= 60→0	
	$bit_{36} = Sig\left(Im_{i+2,j+1}, Im_{i+1,j+2}\right)$	60 >= 44→1	

Fig. 4 Mathematical description and examples of the level 2

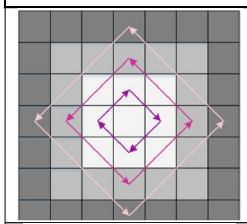
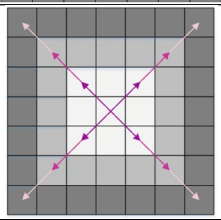
Graph	Mathematical Description	Numerical Example	
	$bit_{37} = Sig\left(Im_{i+1,j}, Im_{i,j+1}\right)$	58 >= 49 → 1	1101
	$bit_{37} = Sig\left(Im_{i+1,j}, Im_{i,j+1}\right)$	58 >= 49→1	1101
	$bit_{38} = Sig\left(Im_{i,j+1}, Im_{i+1,j+2}\right)$	49 >= 44→1	
	$bit_{39} = Sig\left(Im_{i+1,j+2}, Im_{i+2,j+1}\right)$	44 >= 60→0	
	$bit_{40} = Sig\left(Im_{i+2,j+1}, Im_{i+1,j}\right)$	60 ≥ 58→1	
	$bit_{41} = Sih\left(Im_{i+1,j+1}, Im_{i,j}\right)$	59 >= 55→1	1111
	$bit_{42} = Sig\left(Im_{i+1,j+1}, Im_{i,j+2}\right)$	59 >= 41→1	
	$bit_{43} = Sig\left(Im_{i+1,j+1}, Im_{i+2,j+2}\right)$	59 >= 48→1	
	$bit_{44} = Sig\left(Im_{i+1,j+1}, Im_{i+2,j}\right)$	59 >= 52→1	

Fig. 5 Mathematical description and examples of level 3

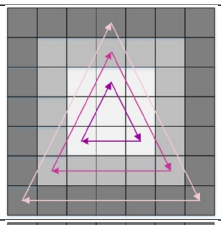
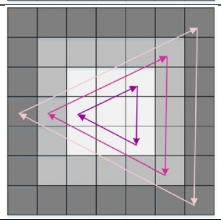
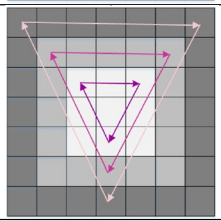
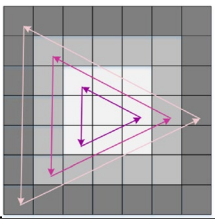
Graph	Mathematical Description	Numerical Example	
	$bit_{45} = Sig\left(Im_{i+2,j}, Im_{i,j+1}\right)$	52 >= 49→1	110
	$bit_{46} = Sig\left(Im_{i,j+1}, Im_{i+2,j+2}\right)$	49 >= 48→1	
	$bit_{47} = Sig\left(Im_{i+2,j+2}, Im_{i+2,j}\right)$	48 >= 52→0	
	$bit_{48} = Sig\left(Im_{i+1,j}, Im_{i,j+2}\right)$	58 >= 41→1	100
	$bit_{49} = Sig\left(Im_{i,j+2}, Im_{i+2,j+2}\right)$	41 >= 48→0	
	$bit_{50} = Sig\left(Im_{i+2,j+2}, Im_{i+1,j}\right)$	48 >= 52→0	
	$bit_{51} = Sig\left(Im_{i,j}, Im_{i,j+2}\right)$	55 >= 41→1	101
	$bit_{52} = Sig\left(Im_{i,j+2}, Im_{i+2,j+1}\right)$	41 >= 60→0	
	$bit_{53} = Sig\left(Im_{i+2,j+1}, Im_{i,j}\right)$	60 >= 55→1	
	$bit_{54} = Sig\left(Im_{i,j}, Im_{i+2,j}\right)$	55 >= 52→1	110
	$bit_{55} = Sig\left(Im_{i+2,j}, Im_{i+1,j+2}\right)$	52 >= 44→1	
	$bit_{56} = Sig\left(Im_{i+1,j+2}, Im_{i,j}\right)$	44 >= 55→0	

Fig. 6 Mathematical description of the Level 4 with a numerical example

4 Results

4.1 Datasets

In order to evaluate the recognition ability of the proposed texture transformation based exemplar FER method, JAFFE (Lyons et al. 1999) and TFEID (Chen and Yen 2007) datasets are used. These two public datasets are widely used for FER. The sample images of these datasets are shown in the Fig. 8 and its attributes are given as below.

JAFFE (Lyons et al. 1999) consists of 210 facial expression images of a Japanese female. TFEID (Chen and Yen 2007) dataset comprises of seven expressions (anger, disgust, fear, happiness, neutral, sadness, surprise) of 268 Taiwanese models.

4.2 Experimental setup

In the experiments, the computer with specifications listed in Table 3 was used and simulations were done using Matlab2018a.

As can be seen from this table (see Table 3), the proposed graph-based model has been implemented on a simple configured computer since it is a lightweight image classification (FER) model.

4.3 Cases

The presented graph-based textural extractor is a parametric function. For feature extraction, variable sized overlapping blocks can be used. To clearly show feature extraction ability of the presented graph-based function, 3×3, 5×5 and 7×7 sized overlapping blocks have been used. Moreover, we have considered LDA and SVM as classifiers. Hence, six cases have been created and these cases are defined in Table 4.

4.4 Performance evaluation

To measure classification ability of these cases, accuracy parameter is used. The mathematical formula of the accuracy is given in Eq. (3).

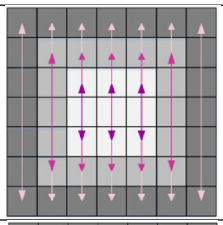
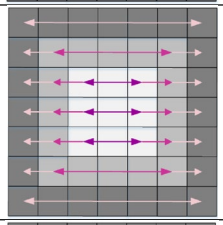
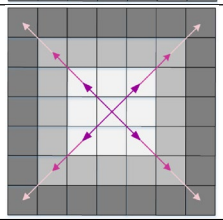
Graph	Mathematical Description	Numerical Example	
	$bit_{57} = Sig\left(Im_{i,j}, Im_{i+2,j}\right)$	55 >= 52→1	100
	$bit_{58} = Sig\left(Im_{i,j+1}, Im_{i+2,j+1}\right)$	49 >= 60→0	
	$bit_{59} = Sig\left(Im_{i,j+2}, Im_{i+2,j+2}\right)$	41 >= 48→0	
	$bit_{60} = Sig\left(Im_{i,j}, Im_{i,j+2}\right)$	55 >= 41→1	111
	$bit_{61} = Sig\left(Im_{i+1,j}, Im_{i+1,j+2}\right)$	58 >= 44→1	
	$bit_{62} = Sig\left(Im_{i+2,j}, Im_{i+2,j+2}\right)$	52 >= 48→1	
	$bit_{63} = Sig\left(Im_{i,j}, Im_{i+2,j+2}\right)$	55 >= 48→1	10
	$bit_{64} = Sig\left(Im_{i,j+2}, Im_{i+2,j}\right)$	41 >= 52→0	

Fig. 7 Mathematical description of level 5 with a numerical example

$$Acc(\%) = \frac{\#\text{True predictive images}}{\#\text{Total images}} \times 100 \tag{3}$$

Moreover, the calculated confusion matrices have been given. By using the confusion matrices, other commonly used performance evaluation parameters can be calculated.

4.5 Validation

tenfold cross validation is used in the classification phase and average success rates are evaluated. Performance obtained using the two databases (JAFFE and TFEID) is given below.

4.6 Experimental results using JAFFE database

The cases used in the experiment are implemented utilizing JAFFE dataset to classify the seven universal facial expressions. The results and confusion matrices of the cases are shown in Table 5. Besides the confusion matrices obtained for different cases are shown in Fig. 9.

4.7 Experimental results using TFEID database

The cases used in the experiment are implemented utilizing TFEID dataset to classify the 7 universal facial expressions. The results obtained are shown in Table 6. Moreover, Fig. 10 shows the confusion matrices obtained for various cases using TFEID dataset.

4.8 Execution time and computational complexity analysis

In order to evaluate the performance of the proposed method clearly, execution time and computational complexity are examined. The average execution time of the proposed method for an image is listed in Table 7 for different block sizes. In Table 7, the listed time measures are the summation of preprocessing, feature extraction, feature concatenation and feature reduction times for an image.

The space complexity of the proposed texture transformation is also calculated, and the computational complexity of the different cases is given in Table 8. It is calculated for 3×3, 5×5 and 7×7 block sizes separately.

According to Table 8, the presented model has linear time complexity. In this respect, by using this model, a mobile FER application can be implemented.

4.9 Comparisons with state-of-the-arts

In this paper, an exemplar-based FER method is proposed using a novel graph based texture transformation method. The classification rate and space complexity are considered as the performance evaluation metrics. 3×3, 5×5 and

Table 1 The steps of the proposed textural graph transformation based exemplar FER method

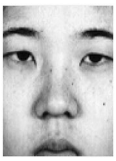
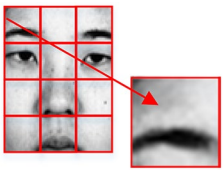
Step 1: Load the image		$W \times H$ In here, W and H represent width and hieigth of the original image
Step 2: Segment the facial area		$w \times h$ w and h define size of the face segmented image
Step 3: Resize the facial area of face image		120×90
Step 4: Divide non-overlapping regions in facial area. The exemplar method is used in this step		30×30
Step 5: Extract the features in each region by using the proposed graph based transform with variable block size	$f_i = [f_{i,1}, \ldots, f_{i,2048}]$ $i = \{1, 2, \ldots, 12\}$	f_i represents features of region with size of 2048
Step 6: Concatenate the features of each region	$f^C = \begin{bmatrix} f_{1,1} & \cdots & f_{1,2048} \\ \vdots & \ddots & \vdots \\ p_{12,1} & \cdots & p_{12,2048} \end{bmatrix}$	f^C represents concatenated features
Step 7: Reduce the feature dimension by using maximum pooling or maximum pooling + PCA	feat = 1 … k	k = 1024 or 128
Step 8: Classify the reduced features using quadratic kernel SVM or LDA using tenfold cross validation		

Table 2 The attributes of the classifiers used

Method	Attributes	Value
LDA	Regularization	Full
	PCA	Enable (Number of components = 128)
SVM	Kernel function	Quadratic
	Box constraint level	1
	Kernel scale mode	Auto
	Manuel kernel scale	1
	Multiclass method	One-vs-All
	Standardize data	True
	PCA	Disable

7×7 block sizes have been used for feature extraction. Six cases have been described using these various block sizes and classifiers. Seven widely used methods are chosen to obtain the comparison results. We have obtained the highest classification accuracy for case 3 among the seven previously presented state of art methods using the same datasets.

The proposed method is compared with 3D Gabor features + SVM (Zhang and Tjondronegoro 2011), contours + SimNet (Lee et al. 2013), meta probability codes + SVM (Farajzadeh et al. 2014), curvelet + online sequential learning machine (Uçar et al. 2016), local Fisher discriminant analysis (Wang et al. 2016), pyramid + SVM + single-branch decision tree (Ashir and Eleyan 2017), exemplar-based + SVM (Farajzadeh and Hashemzadeh 2018). The summary of comparison results obtained using various methods using the same database is listed in Table 9.

The proposed method is also compared with exemplar based SVM [1] method for each expression and the results are listed in Table 10.

Moreover, we have used a commonly used FER dataset (CK +) for increasing comparatively results. CK + (Lucey et al. 2010) contains 981 images with seven emotions. The distribution of this dataset is given as follows; anger:

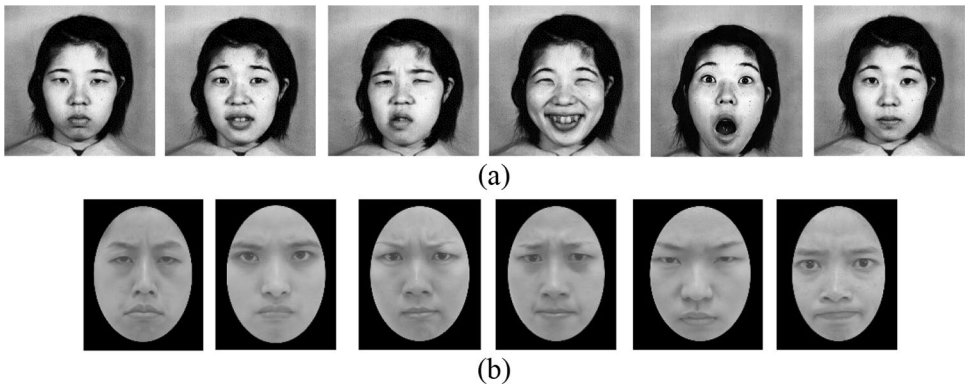


Fig. 8 Various facial expressions of two datasets: **a** JAFFE, **b** TFEID

Table 3 The specifications of the PC used in this study

Systems	Features
CPU	Intel Core i7-7700 (3.60 GHz and 4.20 GHz with Turbo boost)
RAM	16 GB
Buffer	8 M
HDD	1 TB
Operating System	Windows 10.1
Tool	MATLAB 2018a

135, contempt: 54, disgust: 177, fear: 75, happy: 207, sadness: 84 and surprise: 249. We have used 3×3 overlapping blocks our recommended graph based textural feature extractor and the calculated results have been shown in Fig. 11.

As can be seen from Fig. 11, By using LDA and SVM classifiers, 97.25% and 100% classification accuracies have been calculated respectively. Moreover, comparative results about the CK + dataset have been tabulated in Table 11.

5 Discussions

The experimental results clearly illustrate that the proposed exemplar based FER method is superior as compared to the previously presented methods. The case 3 achieved 2.44% and 0.37% higher classification rate than the best previously reported results for JAFFE and TFEID respectively.

Table 4 The attributes of the defined cases

Name	Classifier	Feature extractor
Case 1	LDA	The presented graph-based textural transformation with 3×3 sized overlapping blocks
Case 2	LDA	The presented graph-based textural transformation with 5×5 sized overlapping blocks
Case 3	LDA	The presented graph-based textural transformation with 7×7 sized overlapping blocks
Case 4	SVM	The presented graph-based textural transformation with 3×3 sized overlapping blocks
Case 5	SVM	The presented graph-based textural transformation with 5×5 sized overlapping blocks
Case 6	SVM	The presented graph-based textural transformation with 7×7 sized overlapping blocks

Table 5 Accuracy rates (%) of the cases used in the experiments utilizing JAFFE dataset

No	Expression	Case 1 Acc(%)	Case 2 Acc(%)	Case 3 Acc(%)	Case 4 Acc(%)	Case 5 Acc(%)	Case 6 Acc(%)
1	Anger	63.3	76.67	89.66	60.0	56.67	86.58
2	Disgust	90.0	93.34	96.67	93.33	93.34	93.34
3	Fear	100.0	100.0	100.0	100.0	100.0	100.0
4	Happiness	100.0	100.0	100.0	93.33	93.34	96.67
5	Neutral	96.67	96.67	96.67	100.0	100.0	96.67
6	Sadness	100.0	96.67	96.67	100.0	100.0	93.34
7	Surprise	100.0	100.0	100.0	100.0	96.67	96.67
	Average	92.85	94.76	**97.09**	92.38	91.43	94.75

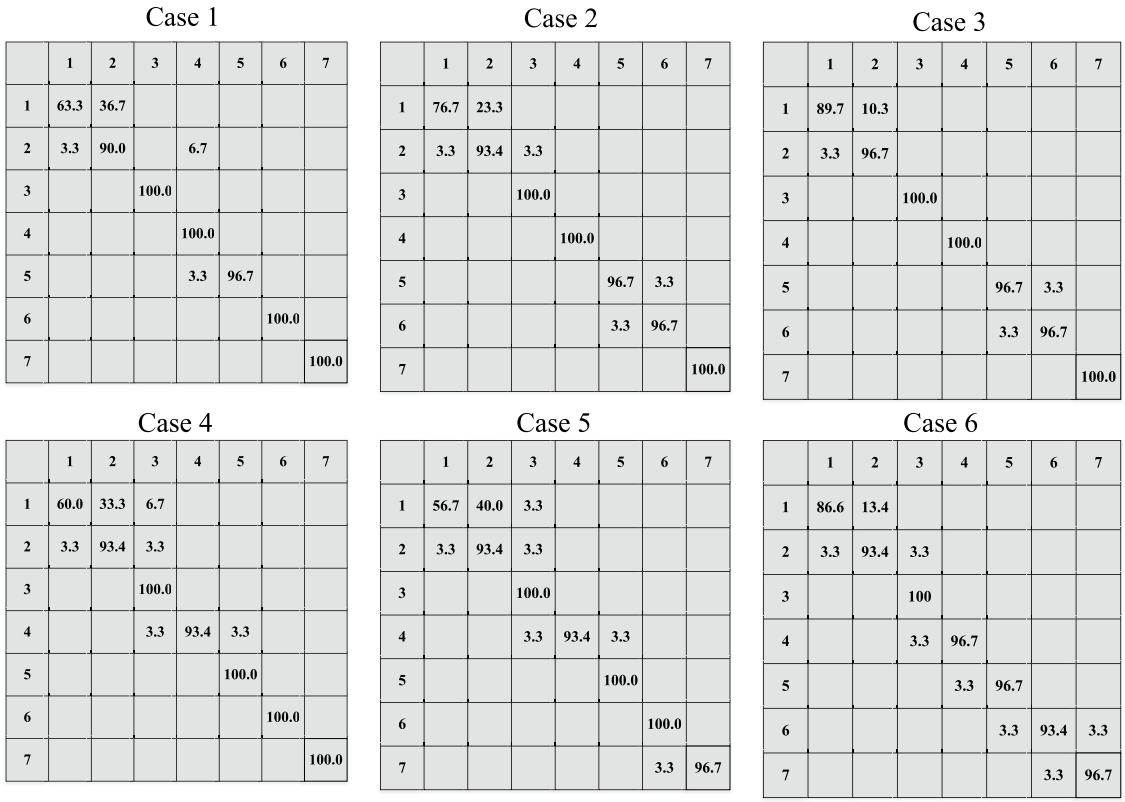


Fig. 9 The confusion matrices obtained for various cases using JAFFE dataset

Table 6 Accuracy rates (%) obtained for various cases using TFEID dataset

No	Expression	Case 1 Acc(%)	Case 2 Acc(%)	Case 3 Acc(%)	Case 4 Acc(%)	Case 5 Acc(%)	Case 6 Acc(%)
1	Anger	79.49	89.74	100.0	84.62	89.74	100.0
2	Disgust	100.0	100.0	100.0	82.05	87.17	94.87
3	Fear	97.44	97.44	97.50	97.44	97.44	97.44
4	Happiness	97.44	97.44	97.44	97.44	94.87	94.87
5	Neutral	100.0	97.44	100.0	100.0	100.0	97.44
6	Sadness	97.44	100.0	100.0	97.44	100.0	97.44
7	Surprise	100.0	100.0	100.0	100.0	100.0	100.0
	Average	95.97	97.44	**99.25**	94.14	95.60	97.43

The second performance evaluation criteria of this method is the execution time. The results of the execution time and space complexity of the proposed graph methods are given in Table 7. The space complexity $O(n^2)$ and computational complexity for each block is equal to $O(145)$ are calculated. It can be noted from the Table 8 that, the computational complexity of this method directly depends on the size of image.

The advantages of the proposed method can be summarized as follows:

– Convolutional neural networks (CNN) are widely used for FER and they achieved good accuracy rates. However, CNN involves high computational complexity and it does not achieve high success rate using small datasets. To extract features deeply, a novel graph based texture transformation is proposed in this study. 15 descriptors are utilized in five levels in the transformation. Our developed image descriptors yield higher classification accuracy and lower computational complexity as compared to the previous studies. Therefore, the graph based texture transformation method is a lightweight feature extractor for FER. Moreover, this

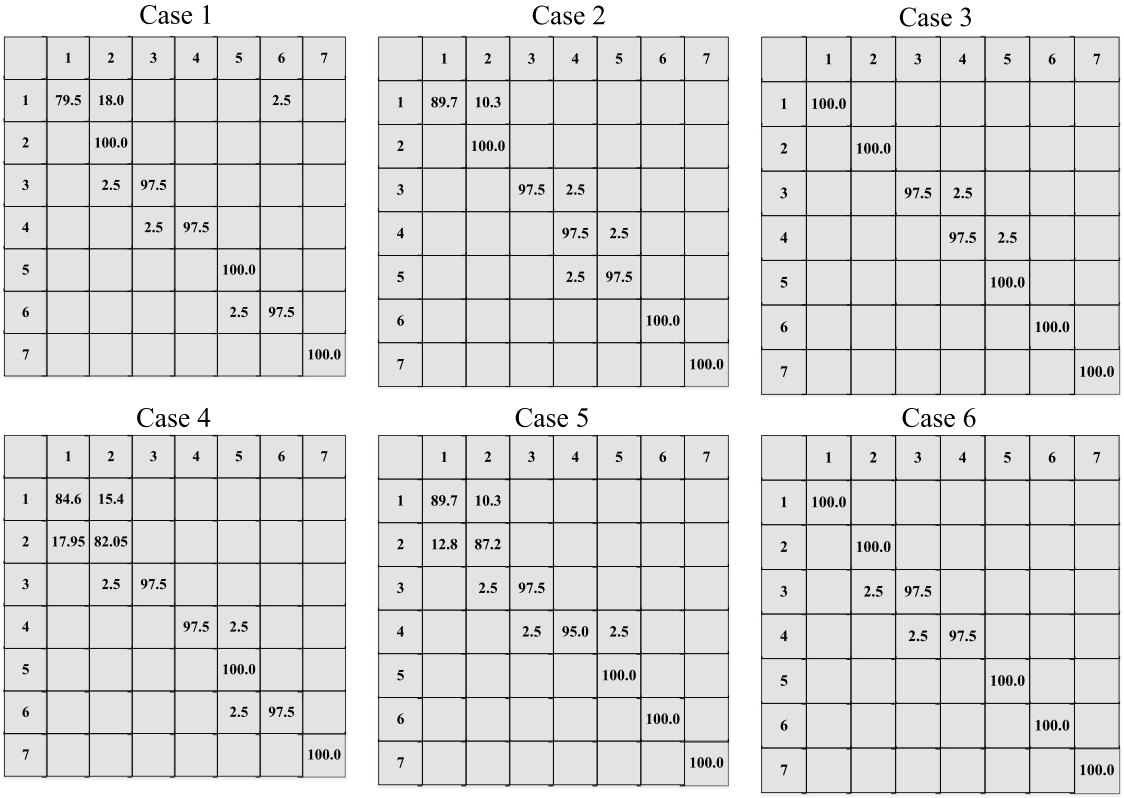
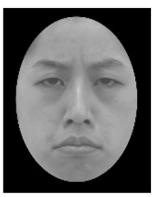


Fig. 10 The confusion matrices obtained for various cases using TFEID dataset

Table 7 Execution time (in seconds) for the proposed method with variable block sizes

Dataset	Sample image	Size of block		
		3×3 (Case 1,4)	5×5 (Case 2,5)	7×7 (Case 3,6)
JAFFE		0.274	0.237	0.220
TFEID		0.276	0.241	0.228

model is appropriate to develop mobile FER application.

- The proposed scheme is a cognitive method as it does not involve any random weight assignment, weight updating, ensemble based method or optimization algorithms.
- The developed technique is robust as we performed ten-fold cross validation.
- The proposed technique extracted distinctive features and as a result achieved high classification accuracy.
- The algorithm is computationally less intensive and can be used in real time applications.
- This graph-based model has been tested on three FER datasets and our model has attained over 97% classification accuracies for all datasets.

Table 8 The computational complexity of the proposed graph based feature extraction method for 3×3, 5×5 and 7×7 block sizes

Input: Grayscale Image (I) with size of W x H
Output: Feature vector (F) with size of 2048

	Size of block and cases		
	3×3 (Case 1,4)	5×5 (Case 2,5)	7×7 (Case 3,6)
[W,H] = size(I);	*O(1)*	*O(1)*	*O(1)*
for i = 1 to W-b **do**	*O(9(W-2)(H-2))*	*O(9(W-4)(H-4))*	*O(9(W-6)(H-6))*
for j = 1 to H-b **do**	*O(64(W-2)(H-2))*	*O(64(W-4)(H-4))*	*O(64(W-6)(H-6))*
Assign pixels	*O(64(W-2)(H-2))*	*O(64(W-4)(H-4))*	*O(64(W-6)(H-6))*
Calculate binary features	*O(8(W-2)(H-2))*	*O(8(W-4)(H-4))*	*O(8(W-6)(H-6))*
Convert decimal values to binary features	*O(2048)*	*O(2048)*	*O(2048)*
Construct 8 feature images and histogram extraction			
end for j			
end for i			
Concatenate histograms to obtain feature			
Total:	*T(W,H) = O(145WH-290W-290H + 2629)*	*T(W,H) = O(145WH-580W-580H + 4369)*	*T(W,H) = O(145WH-870W-870H + 7269)*

The limitation of the proposed technique is that, it has been developed and tested on relative small databases. In the big datasets, deep learning models (Aouayeb et al. 2021; Minaee et al. 2021; Revina and Emmanuel 2019; Savchenko 2021; Umer et al. 2021; Vo et al. 2020) have been used to get high classification performance. In this research, we have presented a feature engineering work. To get high classification results is not main objective of this work. By using simple graphs, we have investigated classification ability of the textural features on the FER image datasets.

Table 9 The overall comparison of results obtained for automated detection of FER using the same datasets

Method	JAFFE (%)	TFEID (%)
3D Gabor features + SVM	91.0	–
Contours + SimNet	–	92.8
Meta Probability Codes + SVM	86.38	92.54
Curvelet + online sequential learning machine	94.65	–
Local Fisher discriminant analysis	77.02	–
Pyramid + SVM + single-branch decision tree	91.43	89.71
Exemplar-based + SVM	92.53	98.90
The proposed Case 3 (Graph based texture transformation + 1D maximum pooling + PCA + LDA)	**97.09**	**99.25**

Bold values indicate the best results

6 Conclusions

In this paper, a novel graph based texture transformation is employed in the feature extraction phase of the automated facial expression recognition. The proposed method consists of five levels and 2048 features are extracted using the proposed graph based texture transformation. The main purpose of the proposed transformation is to obtain in-depth properties with low computational complexity using variable patterns which are constructed using basic geometric shapes. The experimental results are presented for *six* cases. We have obtained 99.25% and 97.09% success rates using TFEID and JAFFE datasets respectively for Case3. Moreover, we have used CK + FER image dataset to increase comparative results. Our graph-based textural feature extraction based classification model yielded 97.25% accuracy using LDA and 100% accuracy rate deploying SVM classifier on the CK + dataset. By using three FER image datasets, general classification capability of the presented graph-based model has been demonstrated. Our proposed framework can be used in the real world applications as it requires minimum execution times and low space complexity. The developed

Table 10 The comparison results obtained using the same datasets for different expressions

		JAFFE			TFEID		
		Recognition Rates (%)					
		Farajzadeh and Hashemzadeh's (Farajzadeh and Hashemzadeh 2018) method	The proposed method	Farajzadeh and Hashemzadeh's (Farajzadeh and Hashemzadeh 2018) method	The proposed method		
		LBP	HOG	Case 3	LBP	HOG	Case 3
1	Anger	83.3	**96.67**	89.66	**100.0**	**100.0**	**100.0**
2	Disgust	72.41	89.66	**96.67**	**100.0**	**100.0**	**100.0**
3	Fear	84.38	87.50	**100.0**	92.5	**100.0**	97.44
4	Happiness	70.97	93.55	**100.0**	92.5	**100.0**	97.44
5	Neutral	90.00	90.00	**96.67**	89.74	94.87	**100.0**
6	Sadness	83.87	90.32	**96.67**	84.62	97.44	**100.0**
7	Surprise	90.00	100.0	**100.0**	**100.0**	**100.0**	**100.0**
	Average	82.14	92.53	**97.09**	94.19	98.90	**99.25**

Bold values indicate the best results

Fig. 11 Confusion matrices of the presented graph based model using LDA and SVM classifiers

	1	2	3	4	5	6	7
1	94.07					5.93	
2		98.15				1.85	
3	0.56		99.44				
4	1.33		1.33	94.67	2.67		
5					100.0		
6	10.71					89.29	
7	0.40	1.20					98.40

(a) LDA

	1	2	3	4	5	6	7
1	100						
2		100					
3			100				
4				100			
5					100		
6						100	
7							100

(b) SVM

Table 11 Comparative results about the CK + dataset

Study	Method	The accuracy results (%)
Minaee et al. (2021)	Attentional convolutional network	98.00
Ashir et al. (2020)	Compressive sensing theory	97.80
Aouayeb et al. (2021)	Vision transformer, convolution neural network	99.80
Umer et al. (2021)	Convolution neural network	97.69
Mukhopadhyay et al. (2021)	Local binary pattern, convolution neural network	79.56
Li et al. (2021)	Efficient convolutional neural network	99.50
Connie et al. (2017)	Convolutional neural network, scale invariant feature transform	99.40
Ghazouani (2021)	Genetic programming, local binary pattern	98.00
Shi et al. (2021)	Multibranch cross-connection Convolutional neural network	98.48
Our model	Textural transformation	100.0

algorithm can be applied to detect other images like palm, hair, and vein.

Funding Open Access funding provided by University of Turku (UTU) including Turku University Central Hospital. This work was supported by Effat University with the Decision Number of Decision No. UC#9/29 April.2020/7.1-22(2)5, Jeddah, Saudi Arabia.

References

Abate AF, Barra P, Barra S, Molinari C, Nappi M, Narducci F (2019) Clustering facial attributes: narrowing the path from soft to hard biometrics. IEEE Access 8:9037–9045

Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal 28:2037–2041

Aouayeb M, Hamidouche W, Soladie C, Kpalma K, Seguier R (2021) Learning vision transformer with squeeze and excitation for facial expression recognition. arXiv preprint arXiv:210703107

Ashir AM, Eleyan A (2017) Facial expression recognition based on image pyramid and single-branch decision tree. Signal Image Video Process 11:1017–1024

Ashir AM, Eleyan A, Akdemir B (2020) Facial expression recognition with dynamic cascaded classifier. Neural Comput Appl 32:6295–6309

Chakraborty S, Singh SK, Chakraborty P (2017) Local quadruple pattern: a novel descriptor for facial image recognition and retrieval. Comput Electr Eng 62:92–104

Chao W-L, Ding J-J, Liu J-Z (2015) Facial expression recognition based on improved local binary pattern and class-regularized locality preserving projection. Signal Process 117:1–10

Chen L-F, Yen Y-S (2007) Taiwanese facial expression image database, Brain Mapping Laboratory, Institute of Brain Science. National Yang-Ming University, Taipei

Cohen I, Sebe N, Garg A, Chen LS, Huang TS (2003) Facial expression recognition from video sequences: temporal and static modeling. Comput vis Image Underst 91:160–187

Connie T, Al-Shabi M, Cheah WP, Goh M (2017) Facial expression recognition using a hybrid CNN–SIFT aggregator. International workshop on multi-disciplinary trends in artificial intelligence. Springer, pp 139–149

Ding C, Choi J, Tao D, Davis LS (2016) Multi-directional multi-level dual-cross patterns for robust face recognition. IEEE Trans Pattern Anal 38:518–531

Ertuğrul IÖ, Jeni LA, Dibeklioğlu H (2018) Modeling and synthesis of kinship patterns of facial expressions. Image vis Comput 79:133–143

Farajzadeh N, Hashemzadeh M (2018) Exemplar-based facial expression recognition. Inform Sci 460:318–330

Farajzadeh N, Pan G, Wu Z (2014) Facial expression recognition based on meta probability codes. Pattern Anal Appl 17:763–781

Fasel B, Luettin J (2003) Automatic facial expression analysis: a survey. Pattern Recogn 36:259–275

Ghazouani H (2021) A genetic programming-based feature selection and fusion for facial expression recognition. Appl Soft Comput 103:107173

Guo K, Soornack Y, Settle R (2018) Expression-dependent susceptibility to face distortions in processing of facial expressions of emotion. Vis Res. https://doi.org/10.1016/j.visres.2018.02.001

Hernández B, Olague G, Hammoud R, Trujillo L, Romero E (2007) Visual learning of texture descriptors for facial expression recognition in thermal imagery. Comput vis Image Underst 106:258–269

Kabir MH, Jabid T, Chae O (2010) A local directional pattern variance (LDPv) based face descriptor for human facial expression recognition. In: 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, IEEE, pp 526–532

Lee H-C, Wu C-Y, Lin T-M (2013) Facial expression recognition using image processing techniques and neural networks. Advances in intelligent systems and applications. Springer, Cham, pp 259–267

Li L, Feng X, Xia Z, Jiang X, Hadid A (2018) Face spoofing detection with local binary pattern network. J vis Commun Image R 54:182–192

Li M, Li X, Sun W, Wang X, Wang S (2021) Efficient convolutional neural network with multi-Kernel enhancement features for real-time facial expression recognition. J Real-Time Image Proc 18:2111–2122

Lien JJ, Kanade T, Cohn JF, Li C-C (1998) Automated facial expression recognition based on FACS action units. In: Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, IEEE, pp 390–395

Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z, Matthews I (2010) The extended cohn-kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, IEEE, pp 94–101

Lyons MJ, Budynek J, Akamatsu S (1999) Automatic classification of single facial images. IEEE Trans Pattern Anal 21:1357–1362

Matsugu M, Mori K, Mitari Y, Kaneda Y (2003) Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Netw 16:555–559

Minaee S, Minaei M, Abdolrashidi A (2021) Deep-emotion: facial expression recognition using attentional convolutional network. Sensors 21:3046

Moore S, Bowden R (2011) Local binary patterns for multi-view facial expression recognition. Comput vis Image Und 115:541–558

Mukhopadhyay M, Dey A, Shaw RN, Ghosh A (2021) Facial emotion recognition based on textural pattern and convolutional neural network. In: 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), IEEE, pp 1–6

Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal 24:971–987

Owusu E, Zhan Y, Mao QR (2014) A neural-adaboost based facial expression recognition system. Expert Syst Appl 41:3383–3390

Revina IM, Emmanuel WS (2018) Face expression recognition using LDN and dominant gradient local ternary pattern descriptors. J King Saud Univ Comput Inf Sci 33:392–398

Revina IM, Emmanuel WS (2019) Face expression recognition with the optimization based multi-SVNN classifier and the modified LDP features. J vis Commun Image R 62:43–55

Rivera AR, Castillo JR, Chae OO (2013) Local directional number pattern for face analysis: face and expression recognition. IEEE Trans Image Process 22:1740–1752

Sadeghi H, Raie A-A (2019) Histogram distance metric learning for facial expression recognition. J vis Commun Image R 62:152–165

Savchenko AV (2021) Facial expression and attributes recognition based on multi-task learning of lightweight neural networks. In: 2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY), IEEE, pp 119–124

Shan C, Gong S, McOwan PW (2009) Facial expression recognition based on local binary patterns: a comprehensive study. Image vis Comput 27:803–816

Shi C, Tan C, Wang L (2021) A facial expression recognition method based on a multibranch cross-connection convolutional neural network. IEEE Access 9:39255–39274

Tan X, Triggs W (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans Image Process 19:1635–1650

Turan C, Lam K-M (2018) Histogram-based local descriptors for facial expression recognition (FER): a comprehensive study. J vis Commun Image R 55:331–341

Uçar A, Demir Y, Güzeliş C (2016) A new facial expression recognition based on curvelet transform and online sequential extreme learning machine initialized with spherical clustering. Neural Comput Appl 27:131–142

Umer S, Rout RK, Pero C, Nappi M (2021) Facial expression recognition with trade-offs between data augmentation and deep learning features. J Amb Intel Hum Comp 13:721

Virrey RA, Liyanage CDS, Petra MIBPH, Abas PE (2019) Visual data of facial expressions for automatic pain detection. J vis Commun Image R 61:209–217

Vo T-H, Lee G-S, Yang H-J, Kim S-H (2020) Pyramid with super resolution for in-the-wild facial expression recognition. IEEE Access 8:131988–132001

Wang Z, Ruan Q, An G (2016) Facial expression recognition using sparse local Fisher discriminant analysis. Neurocomputing 174:756–766

Yaddaden Y, Adda M, Bouzouane A, Gaboury S, Bouchard B (2018) User action and facial expression recognition for error detection system in an ambient assisted environment. Expert Syst Appl 112:173–189

Zhang A, Cheng B, Acharya RS, Menon RP (1996) Comparison of wavelet transforms and fractal coding in texture-based image retrieval. In: Visual Data Exploration and Analysis III, 1996. International Society for Optics and Photonics, pp 116–126

Zhang Y, Hua C (2015) Driver fatigue recognition based on facial expression analysis using local binary patterns. Optik 126:4501–4505

Zhang L, Tjondronegoro D (2011) Facial expression recognition using facial movement features. IEEE Trans Affect Comput 2:219–229

Zhao G, Pietikainen M (2007) Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans Pattern Anal 29:915–928