Modified chess patterns: handcrafted feature descriptors for facial expression recognition

Mukku Nisanth Kartheek^{1,2} · Munaga V. N. K. Prasad¹ · Raju Bhukya²

Received: 26 April 2021 / Accepted: 5 September 2021 / Published online: 21 September 2021
© The Author(s) 2021

Abstract

Facial expressions are predominantly important in the social interaction as they convey the personal emotions of an individual. The main task in Facial Expression Recognition (FER) systems is to develop feature descriptors that could effectively classify the facial expressions into various categories. In this work, towards extracting distinctive features, Radial Cross Pattern (RCP), Chess Symmetric Pattern (CSP) and Radial Cross Symmetric Pattern (RCSP) feature descriptors have been proposed and are implemented in a 5×5 overlapping neighborhood to overcome some of the limitations of the existing methods such as Chess Pattern (CP), Local Gradient Coding (LGC) and its variants. In a 5×5 neighborhood, the 24 pixels surrounding the center pixel are arranged into two groups, namely Radial Cross Pattern (RCP), which extracts two feature values by comparing 16 pixels with the center pixel and Chess Symmetric Pattern (CSP) extracts one feature value from the remaining 8 pixels. The experiments are conducted using RCP and CSP independently and also with their fusion RCSP using different weights, on a variety of facial expression datasets to demonstrate the efficiency of the proposed methods. The results obtained from the experimental analysis demonstrate the efficiency of the proposed methods.

Keywords Facial expressions · Handcrafted feature descriptor · Pattern · Person independent · Symmetric

Introduction

Facial expressions provide valuable information about a person by reflecting his psychological characteristics, which provide an important means for effective communication [66]. Facial expressions expressed by humans are common across traditions and cultures, and they provide an immediate means to analyze the mood of a person [15]. Research suggests that about 55% of human communication happens through facial expressions only [8]. Since the last decade, the field of Facial Expression Recognition (FER) has gathered a

lot of attention from researchers because of its wide range of applications in driver mood detection, affective computing, clinical psychology and animation [35] etc. The relevance in day to day communication and advanced intelligent interactions between humans and machines are the important factors driving the study of FER [45].

The robustness of developing an FER system is hindered due to inconsistent image acquisition conditions, spontaneous expressions, ethnicity variations, illumination variations, aging factors, noise variations etc [33]. So, developing a feature descriptor that is robust to such dynamic changes is a complicated and challenging task. FER system includes image acquisition and pre-processing task, feature extraction task for extracting expression-specific features and classification task for classifying the expressions [35]. The classification task of an FER system is hugely dependent on the method used for feature extraction, as inappropriate feature extraction would degrade the performance even after using the best of classifiers. So, a proper feature extraction technique which could effectively capture expression-specific changes is essential for an FER system [66].

✉ Mukku Nisanth Kartheek
nisanthkartheek@gmail.com

Munaga V. N. K. Prasad
mvnkprasad@idrbt.ac.in

Raju Bhukya
drrajunitw@gmail.com

¹ Center for Affordable Technologies, Institute for Development and Research in Banking Technology, Hyderabad, India

² Department of Computer Science and Engineering, National Institute of Technology Warangal, Warangal, India

A person's face can depict many expressions and there are only minute differences between the various expressions conveyed by human beings. For accurate classification in FER, it is essential to capture those minute information related to specific expressions. Based on the literature studies, the texture-based feature descriptors have been proven to be effective for extracting valuable features from a facial image by manipulating the neighborhood pixel relationship for detecting and capturing minute details in an image [35,56,57]. In FER systems, the most significant problem is to extract effective and meaningful patterns from the facial images [56].

The neighboring pixels relationship and the adjacent pixels relationship are crucial to detect finer appearance changes with respect to specific expressions to detect facial expressions effectively. Accordingly, the proposed feature descriptors have been developed by considering the neighboring pixels relationship (both diagonal and four neighbors) for Radial Cross Pattern (RCP) and by considering the adjacent pixels relationship based on Knight pixel positions for Center Symmetric Pattern (CSP). In this work, for the purpose of feature extraction, the 24 neighboring pixels surrounding the center pixel in a 5×5 neighborhood are arranged into two groups, namely RCP, which extracts two feature values by comparing 16 pixels with the center pixel and CSP extracts feature one feature value from the remaining 8 pixels. The features are extracted using RCP and CSP independently and also with their fusion named Radial Cross Symmetric Pattern (RCSP). The main strength of the proposed methods is that, from the available images in the dataset, the relevant features can be extracted without requiring much training data.

The remainder of the paper is structured as follows: the related work in the field of FER is summarized in the second section, and a brief review of existing descriptors that are a basis for the proposed methods are mentioned in the third section. The proposed feature descriptors for facial feature extraction are discussed in the fourth section. In the fifth section, the datasets considered for experimental evaluation and the comparison analysis of the proposed methods with the existing methods are reported. The concluding remarks and the suggestions for further analysis and study are mentioned in the last section.

Related work

The existing techniques in FER systems are broadly classified into geometric-based methods and appearance-based methods [35]. The geometric-based methods [7,13] encode the locations, shapes, corner and contour information of main facial components such as eyes, nose and mouth. These geometric-based methods encode the characteristics that can

describe the entire facial image with a lesser number of features that are scale and rotation invariant. Despite the fact that these geometric-based methods represent facial geometry, they fail to capture minute local details such as skin's texture variations, and ridges. The facial appearance is best described by appearance-based methods, which can further be classified as global (holistic)-based methods [4,58] and local-based methods [21,27,34,35,40–42]. Holistic (global) methods such as Eigenfaces [58] and Fisher faces [4] apply projection based techniques for producing a global description of the entire facial image. As these global-based methods are aimed at representing a facial image globally, they are unsuitable for capturing finer appearance changes corresponding to various facial expressions [41]. The local appearance-based methods [17,34,35,40–42,45] are aimed at investigating the local regions to describe various features such as corners, curved and straight edges etc. Also, the local-based methods can capture micro-level texture information such as ridge details, specific skin changes and minute characteristics that are relevant to various facial expressions.

The research on local-based methods is being carried out in two directions, namely texture-based [35,45,67] and edge-based approaches [17,34,40–42]. Local Binary Pattern (LBP) [45] is the most popular texture-based method for facial feature extraction. LBP is computationally efficient and is also invariant to changes in monotonic illumination. In cases of intensity fluctuations, random noises, and non-monotonic illumination levels, LBP's feature extraction capability is affected [74]. Lai et al. [27] proposed Center Symmetric LBP (CSLBP) for greatly reducing the feature vector length of LBP. In Local Directional Pattern (LDP) [17], the directions related to top three responses from the eight obtained responses are encoded. Since Kirsch masks are applied on a local 3×3 neighborhood, the existence of noise or intensity distortions may affect the computations of Kirsch value responses. The top negative and positive Kirsch response values are encoded by the Local Directional Number (LDN) [41]. LDN is still affected by the noise present in the local neighborhood, even after preserving top 'k' positive and negative Kirsch responses. The difference in intensity values of opposing pixels in the principal directions is encoded as numbers in Local Directional Texture Pattern (LDTP) [40]. Ryu et al. [42] proposed Local Directional Ternary Pattern (LDTerP) and a multi-level approach for efficiently encoding the emotion related information. NEDP [16] considers the gradient of the center pixel as well as its neighbors for exploring a wider neighborhood to extract consistent features despite the presence of subtle distortions and random noises in a local region.

For capturing expression-specific changes, Murari et al. [35] proposed Regional Adaptive Affinitive Pattern (RADAP) that uses positional thresholds and multi-distance information for describing features that are robust to intra

class variations and illuminations. Also, XRADAP, ARADAP and DRADAP operators are obtained from RADAP by performing xor, adder and decoder operations respectively. Despite the existence of noise in a local neighborhood, Local Prominent Directional Pattern (LPDP) [33] explores local regions for extracting crucial information about edges. Micheal Revina et al. [39] proposed Multi-Directional Triangles Pattern (MDTP) for extracting the features at the locations of lips and eyes. Local Dominant Directional Symmetrical Coding Patterns (LDDSCP) [54] generates two feature values by partitioning the Kirsch response values into two symmetrical groups based on the directional information. Local Optimal Oriented Pattern (LOOP) [21] uses sorted Kirsch responses for weight assignment, rather than using sequential weights. In Center Symmetric Local Gradient Coding (CS-LGC) [66], the gradients are computed in four different directions in a center symmetric manner.

Local Directional Maximum Edge Patterns (LDMEP) [32] applied Robinson's masks in a local neighborhood for extracting both magnitude and phase information. Kas et al. [22] proposed Multi-level Directional Cross Binary Pattern (MDCBP) for texture recognition by combining both multi-radius and multi-orientation information. Durga et al. [23] proposed LBP with Adaptive Window (LBP-AW) for noise robust facial feature extraction. Alphonse et al. [2] proposed Multi-Scale and Rotation-Invariant Phase Pattern (MRIPP) for extracting blur-insensitive and rotation invariant facial features. Kumar et al. [26] proposed Weighted Full binary Tree-Sliced Binary Pattern (WFBT-SBP) for analyzing an RGB image based on inter-pixel similarity patterns. Kola et al. [24] proposed fusion of both singular values and Wavelet-based Local Gradient Coding-Horizontal and Diagonal (WLGC-HD) features for effective FER.

Szegedy et al. [53] proposed GoogleNet for object detection and classification. Duc et al. [62] proposed fusion of AlexNet and Support Vector Machine (SVM) for effective facial feature extraction. Subramanian et al. [50] proposed Meta-Cognitive Neuro-Fuzzy Inference System (McFIS) for FER. Jung et al. [20] used Convolutional Neural Network (CNN) for detecting faces and Deep Neural Network (DNN) for recognizing facial expressions from those detected faces. Shojailangari et al. [48] proposed Landmark-based Pose Invariant feature Descriptor (PID) for handling continuous head pose variations. Zhao et al. [72] proposed LBP on three orthogonal planes (LBP-TOP) for dynamic texture recognition. Shojailangari et al. [47] proposed Optical Flow-based spatial temporal feature descriptor for representing the facial expressions.

Aneja et al. [3] proposed DeepExpr, a transfer learning technique to map expressions from humans to animated characters. Zhao et al. [73] proposed an instance-based transfer learning approach with multiple feature representations. Sun

et al. [51] proposed Individual Free Representation-Based Classification (IFRBC) that utilizes Variation Training Set (VTS) and virtual VTS for remitting the side effects caused by the individual differences. Wu et al. [63] proposed Adaptive Feature Mapping (AFM) for transforming the feature distribution of testing samples into that of training samples. Li et al. [29] proposed Deep Locality Preserving Convolutional Neural Network (DLPCNN) to preserve the locality closeness by maximizing the inter-class scatters. Verma et al. [60] proposed variants of Hybrid Inherited Feature Learning Network (HiNet) for capturing the local contextual information of expressive regions. Ji et al. [18] proposed a fusion network based on intra category common and distinctive feature representation. For FER, Xie et al. [64] presented the Deep Attentive Multi-path Convolutional Neural Network (DAMCNN), which combines the Salient Expression Region Descriptor (SERD) with the Multi-Path Variation Suppressing Network (MPVS-Net).

Zeng et al. [70] proposed a framework for FER that combines both geometric and appearance-based features and utilized Deep Sparse Auto Encoders (DSAE) for recognizing the facial expressions. Drawing inspiration from human vision system, Sadeghi et al. [43] proposed a method based on gabor filters. Li et al. [28] used reinforcement learning for selection of relevant images for expression classification. Saurav et al. [44] proposed Dual Integrated Convolution Neural Network (DICNN) model for recognizing 'in the wild' facial expressions on embedded platform. Jeon et al. [25] utilized subband selective multilevel stationary wavelet gradient transform features for recognizing facial expressions. Image filter-based Subspace Learning (IFSL) is proposed by Yan et al. [65] for better capturing the facial information. Feutry et al. [10] proposed a framework to learn anonymized representation of statistical data. Zeng et al. [69] proposed a novel pattern recognition-based method for accurate segmentation of test and control lines for quantitative analysis of gold immunochromatographic strips. Minaee et al. [36] proposed Attentional Convolutional Network (ACN) model with less than ten layers for classifying emotions from facial images. Sun et al. [52] adopted Dictionary Learning Feature Space (DLFS) for training and Sparse Representation Classification for finding the emotion of query images. A novel Deep Belief Network (DBN)-based multi-task learning algorithm is proposed by Zeng et al. [68] for the diagnosis of Alzheimer's disease. There are various other CNN-based methods such as VGG [49], PCANet [5], ResNet [12] and MobileNet [14] which have shown promising results in FER.

Fig. 1 Feature extraction through CP. **a** Logical placement of chessmen in a 5×5 neighborhood. **b** Numbering scheme of chessmen followed for feature extraction through CP. **c** A 5×5 sample image patch. **d–i** Process of feature extraction through CP for a sample 5×5 image patch. **d** $R = 01110110$. **e** $B = 00110011$. **f** $K = 00110011$. **g** $R_K = 01100110$. **h** $R_B = 01100110$. **i** $K_B = 11001010$

Existing feature descriptors

In this section, the existing feature descriptors namely Chess Pattern (CP) [57], Local Gradient Coding (LGC) [55] and its variants are presented.

CP

Tuncer et al. [57] proposed CP, a local texture-based feature descriptor, developed using chess game rules for texture recognition. With reference to the center pixel (G_c) in a 5×5 neighborhood, CP logically places chessmen (Rook, Bishop and Knight) in possible positions following chess game rules. The possible positions where Rook $\{R_{1,2,\dots,8}\}$, Bishop $\{B_{1,2,\dots,8}\}$ and Knight $\{K_{1,2,\dots,8}\}$ are placed logically are numbered, which is a basis for feature extraction is shown in Fig. 1a, b. CP extracts six features in a 5×5 neighborhood. The first three extracted features are Rook (R), Bishop (B) and Knight (K). For extracting R feature, the pixel intensities in positions $\{R_{1,2,\dots,8}\}$ are compared with the pixel intensity of G_c , which is shown in Eq. (1). The signum function used for comparing two pixel intensities is shown in Eq. (2). For extracting B feature, the pixel intensities in positions $\{B_{1,2,\dots,8}\}$ are compared with the pixel intensity of G_c , which is shown in Eq. (3). For extracting

K feature, the pixel intensities in positions $\{K_{1,2,\dots,8}\}$ are compared with the pixel intensity of G_c , which is shown in Eq. (4).

The next three features extracted are Rook_Knight (R_K), Rook_Bishop (R_B) and Knight_Bishop (K_B). For extracting R_K feature, the pixel intensities in positions $\{R_{1,2,\dots,8}\}$ are compared with the pixel intensity of positions $\{K_{1,2,\dots,8}\}$, which is shown in Eq. (5). For extracting R_B feature, the pixel intensities in positions $\{R_{1,2,\dots,8}\}$ are compared with the pixel intensity of positions $\{B_{1,2,\dots,8}\}$, which is shown in Eq. (6). For extracting K_B feature, the pixel intensities in positions $\{K_{1,2,\dots,8}\}$ are compared with the pixel intensity of positions $\{B_{1,2,\dots,8}\}$, which is shown in Eq. (7). Finally, all the six extracted features are concatenated together to form a final feature vector. The process of feature extraction through CP is demonstrated through numerical example in Fig. 1c–i. Thus, CP considers all the pixels present in 5×5 neighborhood and extracts six features by considering sign information to encode texture details present in an image. As CP considers binary weights, the feature vector length of CP is $256 \times 6 = 1536$, which is very high.

$$R = \sum_{i=1}^8 \sigma(R_i, G_c) \times 2^{8-i} \quad (1)$$

$$\sigma(m, n) = \begin{cases} 1, & \text{if } m \geq n \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

$$B = \sum_{i=1}^8 \sigma(B_i, G_c) \times 2^{8-i} \quad (3)$$

$$K = \sum_{i=1}^8 \sigma(K_i, G_c) \times 2^{8-i} \quad (4)$$

$$R_K = \sum_{i=1}^8 \sigma(R_i, K_i) \times 2^{8-i} \quad (5)$$

$$R_B = \sum_{i=1}^8 \sigma(R_i, B_i) \times 2^{8-i} \quad (6)$$

$$K_B = \sum_{i=1}^8 \sigma(K_i, B_i) \times 2^{8-i} \quad (7)$$

LGC

Tong et al. [55] proposed LGC for facial feature extraction. LGC extracts texture features in a 3×3 neighborhood. LGC operator encodes the gradient information in horizontal, vertical and diagonal directions to generate an eight-bit binary number. The binary number thus formed is then converted into a decimal number, which is replaced in the place of center pixel. This process is repeated throughout the image, and all the histogram features are concatenated block wise to form final feature vector. This LGC encoding captures consistent expression-specific texture features in all possible directions. The coding formula for feature extraction through LGC operator is shown in Eq. (8). As 3×3 mask is considered for LGC, usually radius ($d = 1$) and number of neighbors ($p = 8$). To the LGC operator, some extensions are also proposed, which are LGC-HD operator, LGC-FN operator and LGC-AD operator.

$$\begin{aligned} \text{LGC}_p^d = & \sigma(h_4, h_2) \times 2^7 + \sigma(h_5, h_1) \times 2^6 \\ & + \sigma(h_6, h_8) \times 2^5 + \sigma(h_4, h_6) \times 2^4 \\ & + \sigma(h_3, h_7) \times 2^3 + \sigma(h_2, h_8) \times 2^2 \\ & + \sigma(h_4, h_8) \times 2^1 + \sigma(h_2, h_6) \times 2^0 \end{aligned} \quad (8)$$

LGC-HD

LGC-HD is also proposed by Tong et al. [55], further optimizes the LGC operator and decreases the characteristic feature vector length by considering the gradient information in horizontal and diagonal directions only. The coding formula for feature extraction through LGC-HD operator is shown in Eq. (9). In Fig. 2, feature extraction of LGC and LGC-HD for a sample 3×3 numerical example is shown. As 3×3 mask is considered for LGC-HD, usually $d = 1$ and

$$p = 6.$$

$$\begin{aligned} \text{LGC} - \text{HD}_p^d = & \sigma(h_4, h_2) \times 2^4 + \sigma(h_5, h_1) \times 2^3 \\ & + \sigma(h_6, h_8) \times 2^2 + \sigma(h_4, h_8) \times 2^1 \\ & + \sigma(h_2, h_6) \times 2^0 \end{aligned} \quad (9)$$

LGC-FN

LGC-FN operator [46,66] expands LGC by considering 5×5 neighborhood size. LGC-FN computes feature values in three directions, namely in horizontal and along two diagonal directions. The coding formula for feature extraction through LGC-FN operator is shown in Eq. (10). A sample mask considered for 5×5 operators is shown in Fig. 3a. In Fig. 3b, a sample 5×5 image patch is shown. In Fig. 3c, the computed feature value using LGC-FN is shown.

$$\begin{aligned} \text{LGC} - \text{FN} = & \sigma(h_1, h_2) \times 2^7 + \sigma(h_3, h_4) \times 2^6 \\ & + \sigma(h_5, h_6) \times 2^5 + \sigma(h_7, h_8) \times 2^4 \\ & + \sigma(h_1, h_8) \times 2^3 + \sigma(h_3, h_6) \times 2^2 \\ & + \sigma(h_2, h_7) \times 2^1 + \sigma(h_4, h_5) \times 2^0 \end{aligned} \quad (10)$$

LGC-AD

LGC-AD operator [46,66] computes feature values in four directions, namely in horizontal, vertical and along two diagonal directions. Thus, LGC-AD operator is an extension of LGC-FN operator by additionally considering vertical gradient information. The coding formula for feature extraction through LGC-AD operator is shown in Eq. (11). In Fig. 3d, the computed feature value using LGC-AD is shown.

$$\begin{aligned} \text{LGC} - \text{AD} = & \sigma(h_1, h_2) \times 2^{11} + \sigma(h_3, h_4) \times 2^{10} \\ & + \sigma(h_5, h_6) \times 2^9 + \sigma(h_7, h_8) \times 2^8 \\ & + \sigma(h_1, h_7) \times 2^7 + \sigma(h_3, h_5) \times 2^6 \\ & + \sigma(h_4, h_6) \times 2^5 + \sigma(h_2, h_8) \times 2^4 \\ & + \sigma(h_1, h_8) \times 2^3 + \sigma(h_3, h_6) \times 2^2 \\ & + \sigma(h_2, h_7) \times 2^1 + \sigma(h_4, h_5) \times 2^0 \end{aligned} \quad (11)$$

Limitations of existing descriptors

The existing feature descriptors have some limitations as follows:

- CP generates six feature values, so its feature vector (fv) length is six times the fv length of LBP. Also, it takes more computation time than traditional LBP operator.
- LGC method extracts features in a 3×3 neighborhood using three groups of horizontal pixels and three groups

Fig. 2 Feature extraction through LGC and LGC-HD. **a** Sample mask for 3×3 operators. **b** A sample 3×3 image patch. **c** LGC feature value for sample 3×3 patch. **d** LGC-HD feature value for sample 3×3 patch

Fig. 3 Feature extraction through LGC-FN and LGC-AD. **a** Sample mask considered for 5×5 operators. **b** A sample 5×5 image patch for LGC-FN. **c** Computed feature value using LGC-FN. **d** Computed feature value using LGC-AD

of vertical pixels, but in diagonal direction, only two groups of pixels are considered. As a result, the gradient information in the diagonal directions is not completely captured, which negatively impacts the recognition accuracy [66].

- In LGC-HD operator, fv length is reduced when compared to LGC, as it does not take into consideration the gradient information computed in the vertical direction.
- LGC-FN operator does not consider the gradient information in vertical information and also the characteristics of center pixel information is neglected.
- LGC-AD operator generates a fv length of 4096, which is very huge when compared to traditional LBP operator.
- Most of the existing edge-based methods generate unstable patterns in the smoother regions of an image. Also, some of the existing variants of binary patterns generate the same feature values for different image portions.
- Deep learning techniques need high computational resources and much training data for effective expression recognition.

So, by considering all these information, new feature descriptors are developed, which are discussed in detail in the next section.

Main contributions

The main contributions in this work are summarized as follows:

- Local texture-based feature descriptors, namely RCP, CSP and their fusion RCSP are proposed and applied in a 5×5 neighborhood for extracting facial features by

considering both the neighboring pixels relationship and the adjacent pixels relationship.

- RCP considers multi-radial and multi-orientation information and extracts two features by comparing the neighboring pixels with the current pixel in horizontal, vertical and diagonal directions.
- CSP extracts one feature value by comparing the adjacent Knight pixel positions in a 5×5 neighborhood. Along with horizontal and vertical pixels, CSP method also considers the pixels in four diagonal directions in contrast to LGC which compares only in two directions. Upon considering the diagonal pixels in four directions, the experimental results also showed an enhanced recognition accuracy.
- The proposed methods have been evaluated with different weights to find out the optimal recognition accuracy.
- To evaluate and validate the efficiency of the proposed methods, the experiments are conducted on a variety of facial expression datasets which include datasets captured in the lab environment, dataset in the wild and also on an animated facial expression dataset.
- The proposed methods, which are non-parametric methods, outperformed the standard existing methods proving the robustness of the proposed descriptors.

Proposed methodology

Designing a suitable and robust feature descriptor is of most relevance for any classification tasks. As like CP, LGC-FN and LGC-AD operators, in this work also 5×5 neighborhood is considered. A sample 5×5 block (T) at the pixel coordinate (r, s) is shown in Eq. (12). The center pixel is denoted

as G_c , which is shown in Eq. (13). From Fig. 1a, in CP, it is observed that in a 5×5 grid, with reference to G_c , Rook is placed in horizontal and vertical pixel positions, Bishop is placed in diagonal and anti-diagonal pixel positions and Knight is positioned in the leftover pixel positions. The same positioning of Rook, Bishop and Knight as in CP is adopted while designing our feature descriptors. In CP, the pixel positions where Rook, Bishop and Knight could be placed are numbered only in clockwise manner, and based on this number assignment feature extraction process is explained. For image processing applications, choosing the neighborhood size is crucial in the design phase of handcrafted feature descriptors. If a smaller neighborhood (3×3) is chosen, then the number of pixels considered are limited, hence the accuracy obtained may not be optimal. In general, if more pixels are involved in designing a kernel, the more accurate is the classification. However, choosing a larger neighborhood (7×7) size increases the computation time. In this work, 5×5 neighborhood is chosen to incorporate both multi-radial and multi-orientation information for exploring wider information in a local neighborhood. By extracting features in such a manner, large inter-class distinctions and low intra-class variations can be achieved.

In this work, the 24 neighboring pixels surrounding the center pixel are divided into two groups, namely RCP and CSP. The group of pixels considered for RCP is the same as mentioned in CS-LGC [66] and MDCBP [22], but the methodology used for feature extraction is different. RCP extracts features by comparing 16 pixels with G_c and (CSP) extracts features from the remaining 8 pixels. RCP is subdivided into two groups, namely RCP₁ and RCP₂, each extracting one feature value by comparing 8 pixels information with G_c . The process of feature extraction through RCP₁, RCP₂ and CSP is discussed below in the subsequent subsections. To capture better texture details, numbering assignment is done in both clockwise (for CSP) and anti-clockwise directions (for RCP). The numbering assigned to Rook, Bishop and Knight is shown in Fig. 5a. From CP, the concept of comparing the pixel intensity with G_c is adopted while designing RCP and from LGC operators, the concept of comparing vertical, diagonal and horizontal pixel information is adopted while designing CSP. Thus, the feature descriptor is modelled by considering the advantages of both CP and LGC operators.

$$T = \begin{bmatrix} r,s & r,s+1 & r,s+2 & r,s+3 & r,s+4 \\ r+1,s & r+1,s+1 & r+1,s+2 & r+1,s+3 & r+1,s+4 \\ r+2,s & r+2,s+1 & r+2,s+2 & r+2,s+3 & r+2,s+4 \\ r+3,s & r+3,s+1 & r+3,s+2 & r+3,s+3 & r+3,s+4 \\ r+4,s & r+4,s+1 & r+4,s+2 & r+4,s+3 & r+4,s+4 \end{bmatrix} \quad (12)$$

$$G_c = T_{r+2,s+2} \quad (13)$$

Overview of the proposed method

Initially, the images from the standard facial expression datasets are given as an input. Pre-processing is then done using Viola Jones algorithm [61] for extracting the facial region. For maintaining a uniformity among all “in the lab” datasets, the images have been resized into 120×120 image responses. Histogram equalization is applied on the pre-processed images for normalizing the illumination levels in an image. The proposed feature descriptors, namely RCP, CSP and their fusion RCSP are applied over an input image to get feature response maps. The feature response maps generated using the proposed methods are divided into ‘R’ non-overlapping regions, each of size $N \times N$. From each of these regions, the features are extracted. The feature vector is formed by concatenating all the features obtained from all the regions. The feature vectors are obtained for both the training and the testing images. These feature vectors are then passed on to Multi-class Support Vector Machine (SVM) classifier for expression classification. The block diagram of the proposed method is shown in Fig. 4.

Feature extraction through RCP₁

Sign component is proved to be an efficient factor in developing feature descriptors [22]. RCP₁ contains a set of 8 pixels, which includes 4 pixels corresponding to Rook, considered from the 3×3 neighborhood ($d = 1$) and the remaining 4 pixels corresponding to Bishop, considered from the 5×5 neighborhood ($d = 2$). The numbering system for Rook and Bishop follows anti-clockwise direction, following Moore’s neighborhood [35]. The pixel positions corresponding to RCP₁ is shown in Fig. 5b. Thus, considering pixels in this manner enables in better capturing the expression-specific texture information in eight directions ($0^\circ, 90^\circ, 180^\circ, 270^\circ, 45^\circ, 135^\circ, 225^\circ, 315^\circ$) respectively. The pixel intensities present in these eight positions ($R_{1,2,3,4}, B_{5,6,7,8}$) are compared with the pixel intensity of G_c . Upon comparison based on signum function, shown in Eq. (2), if the obtained result is positive, then the corresponding bit is encoded as one, else it encoded as zero. Thus, for eight pixel positions, eight corresponding values (either 0 or 1) are obtained, which are then concatenated to form an eight bit binary number is subsequently multiplied with the weight matrix (W_z). The corresponding equations for calculating fv based on RCP₁ are shown in Eqs. (14) and (15). Generally, weight matrix contains binary weights [17,35,45]. Upon using binary weights (shown in Eq. 17, the fv length of RCP₁ is 256, as like LBP. To further reduce the fv length, different weights such as fibonacci (shown in Eq. 18) [38], prime (shown in Eq. 19), natural (shown in Eq. 20), squares (shown in Eq. 21), odd (shown in Eq. 22) and even (shown in Eq. 23) have been considered. For Fibonacci weights, the sequence of first eight

Fig. 4 Block diagram of the proposed method

Fig. 5 Process of feature extraction through Modified Chess Patterns. **a** Numbering scheme of chessmen followed for feature extraction through Modified Chess Patterns. **b** Feature extraction through RCP₁. **c** Feature extraction through RCP₂. **d** Feature extraction through CSP

Fibonacci numbers are considered, which is shown in Eq. (18). Similarly, for other weights also, the sequence of first eight numbers in that particular series are considered. At each time, W_z can take any one the weight values for feature extraction. The number thus obtained after multiplying with W_z is then replaced with the value of G_c . The process of feature extraction through RCP₁ for a numerical example is demonstrated in Fig. 6a, b.

$$\begin{aligned} \text{RCP}_1 = \{ & \sigma(R_1, G_c), \sigma(R_2, G_c), \sigma(R_3, G_c), \\ & \sigma(R_4, G_c), \sigma(B_5, G_c), \sigma(B_6, G_c), \\ & \sigma(B_7, G_c), \sigma(B_8, G_c) \} \end{aligned} \quad (14)$$

$$\text{RCP}_1 = \sum (\text{RCP}_1 \times W_z) \quad (15)$$

$$z = \{\text{binary, Fibonacci, prime, natural, squares, odd, even}\} \quad (16)$$

$$W_{\text{binary}} = [1, 2, 4, 8, 16, 32, 64, 128] \quad (17)$$

$$W_{\text{fibonacci}} = [1, 1, 2, 3, 5, 8, 13, 21] \quad (18)$$

$$W_{\text{prime}} = [2, 3, 5, 7, 11, 13, 17, 19] \quad (19)$$

$$W_{\text{natural}} = [1, 2, 3, 4, 5, 6, 7, 8] \quad (20)$$

$$W_{\text{squares}} = [1, 4, 9, 16, 25, 36, 49, 64] \quad (21)$$

$$W_{\text{odd}} = [1, 3, 5, 7, 9, 11, 13, 15] \quad (22)$$

$$W_{\text{even}} = [2, 4, 6, 8, 10, 12, 14, 16] \quad (23)$$

Feature extraction through RCP₂

RCP₂ contains a set of 8 pixels, which includes 4 pixels corresponding to Bishop, considered from the 3×3 neighborhood ($d = 1$) and the remaining 4 pixels corresponding to Rook, considered from the 5×5 neighborhood ($d = 2$). The numbering system for Rook and Bishop follows anti-clockwise

direction, following Moore's neighborhood [35]. The pixel positions corresponding to RCP₂ are shown in Fig. 5c. Thus, considering pixels in this manner enables in better capturing the information in eight directions ($45^\circ, 135^\circ, 225^\circ, 315^\circ, 0^\circ, 90^\circ, 180^\circ, 270^\circ$), respectively. The pixel intensities present in these eight positions ($B_{1,2,3,4}, R_{5,6,7,8}$) are compared with the pixel intensity of G_c . Upon comparison based on signum function, shown in Eq. (2), if the obtained result is positive, then the corresponding bit is encoded as one, else it encoded as zero. Thus, for eight pixel positions, eight corresponding binary values are obtained, which are then concatenated to form an eight bit binary number is subsequently multiplied with the W_z . The number thus obtained after multiplying with W_z is then replaced with the value of G_c . The corresponding equations for calculating fv based on RCP₂ are shown in Eqs. (24) and (25). The process of feature extraction through RCP₂ for a numerical example is demonstrated in Fig. 6a, c.

$$\begin{aligned} \text{RCP}_2 = \{ & \sigma(B_1, G_c), \sigma(B_2, G_c), \sigma(B_3, G_c), \\ & \sigma(B_4, G_c), \sigma(R_5, G_c), \sigma(R_6, G_c), \\ & \sigma(R_7, G_c), \sigma(R_8, G_c) \} \end{aligned} \quad (24)$$

$$\text{RCP}_2 = \sum (\text{RCP}_2 \times W_z) \quad (25)$$

Feature extraction through RCP

RCP is obtained by concatenating both RCP₁ and RCP₂. Thus, RCP feature descriptor generates two feature values, each corresponding to RCP₁ and RCP₂ and hence, the fv length of RCP becomes 512 (in case of binary weights). The fv length becomes 110, 156, 74, 410, 130 and 146 whenever Fibonacci, prime, natural, squares, odd and even weights are utilized for feature extraction. Thus, using other weights than binary weights, even if two feature values are generated, the

Fig. 6 Example of feature extraction through Modified Chess Patterns. **a** A sample 5×5 image patch. **b** $RCP_1 = 10110011$. **c** $RCP_2 = 00111001$. **d** $CSP = 00010010$

fv length of RCP is much lesser than fv generated for one feature value (LBP, LDP) in all cases (except whenever squares weights are used for feature extraction). The corresponding equation for calculating RCP is shown in Eq. (26).

$$RCP = RCP_1 \cup RCP_2 \quad (26)$$

Feature extraction through CSP

The process of feature extraction through CSP is inspired from LGC-AD operator. The length of fv generated by LGC-AD is 4096 (very high) and also the computational complexity involved in LGC-AD is very high. From LGC-AD, the concept of comparing the horizontal, vertical and diagonal pixel information is adopted while designing the CSP operator. The numbering assignment of Knight is done in clockwise manner. CSP operator captures the pixel information in 4 diagonal directions, 2 vertical directions and in 2 horizontal directions. Thus, by comparing the pixels as shown in Fig. 5d, the fv length of CSP is 16 times lesser than the fv length of LGC-AD. The corresponding equations for calculating CSP is shown in Eqs. (27) and (28). The process of feature extraction through CSP for a numerical example is demonstrated in Fig. 6a, d.

$$CSP = \{\sigma(K_1, K_5), \sigma(K_2, K_6), \sigma(K_3, K_7), \sigma(K_4, K_8), \sigma(K_1, K_6), \sigma(K_2, K_5), \sigma(K_3, K_8), \sigma(K_4, K_7)\} \quad (27)$$

$$CSP = \sum (CSP \times W_z) \quad (28)$$

Feature extraction through RCSP

RCSP is obtained by concatenating both RCP and CSP. Thus, RCP generates two feature values, each corresponding to RCP_1 and RCP_2 and CSP generates one feature value and hence, the fv length of RCSP becomes 768 (incase of binary weights). But, as other weights are also used, the fv length becomes 165, 234, 111, 615, 195 and 219 whenever Fibonacci, prime, natural, squares, odd and even weights are

utilized for feature extraction. Thus, using other weights than binary weights, even if three feature values are generated, the fv length of RCSP is lesser than fv generated for one feature value (LBP, LDP) in all cases (except whenever squares weights are used for feature extraction). The corresponding equation for calculating RCSP is shown in Eq. (29). The different steps involved in the calculation of final feature histogram are shown in Fig. 7.

$$RCSP = RCP \cup CSP. \quad (29)$$

Experimental analysis

For experimental analysis and evaluation, a total of ten datasets namely JAFFE [31], MUG [1], Extended Cohn–Kanade (CK+) [30], OULU VIS Strong [71], TFEID [6], KDEF [11], WSEFEP [37], ADFES [59], RAF [29] and FERG [3] have been considered. All the datasets other than RAF and FERG are captured in the lab environment. RAF contains images captured in the wild and FERG is an animated facial expression dataset. Thus, a variety of facial expression datasets have been considered for experimental evaluation. Anger, Disgust, Fear, Happy, Sad and Surprise are the basic expressions and Neutral is considered as the seventh expression [9]. The number of images considered for experimental evaluation across datasets is shown in Table 1. In Table 1, An. corresponds to Anger, Co. corresponds to Contempt, Di. corresponds to Disgust, Em. corresponds to Embarrass, Fe. corresponds to Fear, Ha. corresponds to Happy, Ne. corresponds to Neutral, Pr. corresponds to Pride, Sa. corresponds to Sad and Su. corresponds to Surprise facial expressions.

Dataset description

JAFFE

JAFFE (Japanese Female Facial Expression) dataset contains 213 facial images belonging to basic six and neutral facial

Fig. 7 Steps involved in obtaining final feature vector. **a** Sample image from TFEID. **b** Obtained feature response map. **c** Dividing the response map into blocks. **d** Histograms for each block. **e** Final feature histogram

Table 1 Number of images considered for experimental evaluation across datasets

Dataset	An.	Co.	Di.	Em.	Fe.	Ha.	Ne.	Pr.	Sa.	Su.	Total
JAFFE	30	–	29	–	32	31	30	–	31	30	213
MUG	225	–	225	–	225	225	225	–	225	225	1575
CK+	135	–	177	–	75	207	369	–	84	249	1296
OULU	240	–	240	–	240	240	240	–	240	240	1680
TFEID	34	68	40	–	40	40	39	–	39	36	336
KDEF	70	–	70	–	70	70	70	–	70	70	490
WSEFEP	30	–	30	–	30	30	30	–	30	30	210
ADFES	22	21	22	21	22	22	22	20	22	21	215
RAF	867	–	877	–	355	5957	3204	–	2460	1619	15,339
FERG	9169	–	8571	–	7419	7331	6938	–	7627	8712	55,767

expressions, captured from ten Japanese female subjects. For each subject, for a particular expression, there are almost four images available in the dataset. The hair of female subjects was tied back to capture expressive regions effectively from the facial images.

MUG

Multimedia Understanding Group created the MUG dataset. This dataset contains image sequences collected from 86 subjects, of which 51 are men and the remaining 35 are women. The images are stored in .jpg format with image size being 896×896 pixels. Among those 86 subjects, the images belonging to 45 subjects are chosen for experimental analysis. For each subject, there are five images for each expression.

CK+

CK+ dataset consists of 593 image sequences collected from 123 subjects. Each sequence begins with a neutral expression and ends with the apex of an expression. For each expression, the three apex frames from each sequence are chosen, and

the neutral expression images are chosen from the onset of the image sequences [35].

OULU

The OULU-CASIA dataset has images captured from 80 subjects whose age lies in the range of 23–58 years old. The expressions are captured in both Near-Infra Red (NIR) and Visual Light Scenarios (VIS) in different illuminations such as strong, dark and weak environments. For each illumination, 480 video sequences were captured from those 80 subjects. OULU VIS strong subset has been considered for experimental analysis and evaluation. For the basic six expressions, the three peak frames from each expression are chosen, and the images for neutral expression were collected from the onset of each recording session [35].

TFEID

Taiwanese Facial Expression Image Database (TFEID), developed at the Brain Mapping Laboratory of National Yang-Ming University, has 7200 stimuli collected from forty subjects of which twenty are male and the remaining twenty are female. TFEID database has eighth expression named

contempt in addition to basic six expressions and neutral expression.

KDEF

The Karolinska Directed Emotional Faces (KDEF) dataset developed at Karolinska Institute, Sweden has 4900 images obtained from seventy subjects of which 35 are male and the remaining 35 are female, whose age lies in the range of 20–30 years old. Each subject is posed for seven expressions, and the images are collected from the subjects in five different angles namely -90° , -45° , 0° , $+45^\circ$, $+90^\circ$, respectively. KDEF has images stored in JPEG format and the images are of size 562×762 pixels.

WSEFEP

The full form of WSEFEP dataset is Warsaw Set of Emotional Facial Expression Pictures, which has 210 images collected from 30 individuals of which 14 are male and the remaining 16 are female. The subjects have given poses for all seven facial expressions.

ADFES

Amsterdam Dynamic Facial Expression Set is the full form of ADFES dataset. This dataset has three more expressions, namely contempt, embarrassment and pride apart from the basic seven expressions. 216 images are present in this dataset. This dataset has videos stored in MPEG-2 format and still pictures are also present in this dataset. The images were captured from 22 persons of which 12 are male, 10 are female whose age lies in between 18 and 25 years old.

RAF

RAF (Real-world Affective Faces) dataset has 29,672 real-world images captured in the wild environment. RAF dataset has six basic emotions and twelve compound emotions. For our experimental evaluation, the basic emotions are only considered. This dataset has 12,271 aligned facial images for training and 3068 aligned images for testing. The same number of training and testing images have been utilized for our experimental evaluation. The image size considered for experimental evaluation is 100×100 .

FERG

FERG (Facial Expression Research Group Database) has 55,767 annotated facial images from six stylized characters. The characters were developed using MAYA software. The images from each character are categorized into seven

expressions. The proposed methods have been mainly implemented on this dataset to evaluate their performance on cartoon characters. Out of 55,767 images, 48,767 are considered for training and the remaining 7000 (1000 from each expression, chosen randomly) are considered for testing purposes.

Experimental setup

Initially, the datasets are gathered and for the datasets captured in the lab, pre-processing is done using Viola Jones algorithm [61] to extract the facial region. To maintain an uniformity among the datasets, all the images are converted into grayscale images and are resized into 120×120 image responses (as like RADAP [35]). The image size considered for RAF dataset is 100×100 , whereas for FERG dataset it is 48×48 . The feature extraction capability of the proposed methods gets affected under illumination variations. Histogram equalization is chosen over other methods because this method is useful in images with backgrounds and foregrounds that are both bright or both dark. Some of the datasets used for experimental analysis contains the images obtained by the people of different races. Hence, histogram equalization is performed to normalize the illumination levels in an image. Then, RCP and CSP feature descriptors are utilized for extracting the facial features and corresponding to RCP and CSP, feature response maps are generated. Each feature response map is then partitioned into R non-overlapping regions, where size of each block is $N \times N$. The block size is empirically chosen as 8 i.e. ($N = 8$), for all the datasets. The block-wise features extracted from each feature map are concatenated together to form a final feature vector.

For “in the lab” datasets, for experimental evaluation and analysis, person independent (PI) scheme is adopted. In PI scheme, leave one subject out strategy has been followed (for all datasets except CK+), i.e., at each time, all the images from one particular subject are excluded from the training and are used for testing purposes. A ten fold PI cross validation is performed for CK+ dataset. As a result, by excluding an user in this way, the person’s independence is always ensured. In this work, for the purpose of classification, a multi-class classifier model employing γ ($\gamma - 1$)/2 binary SVM models with one-versus-one approach and a linear kernel is followed, where γ corresponds to the total number of classes. Multi-class SVM is chosen for classification purpose, as it is the most widely classifier for addressing the problem of FER in the field of pattern recognition [23,24,35]. The experiments are performed using MATLAB R2018a software on i5 processor with Windows 10 operating system and 16 GB RAM. The existing variants of binary patterns such as LBP, LDP, LDN, CSLBP, LGC, LDTP, LDTP, RADAP and CP are implemented in our setup and correspondingly the recognition accuracy is reported. The recognition accuracy is computed

Table 2 Recognition accuracy of RCP with different weights for six expressions

Dataset	Binary	fibonacci	Prime	Natural	Squares	Odd	Even
JAFFE	59.48	60.90	58.92	59.94	57.60	61.51	59.94
MUG	87.70	87.41	87.93	88.07	87.85	86.52	88.07
CK+	91.99	91.88	91.66	91.44	92.21	90.67	91.44
OULU	75.97	75.14	76.18	74.51	75.56	75.49	75.14
TFEID	95.08	94.58	95.50	95.42	95.58	94.17	95.42
KDEF	84.05	83.10	83.33	84.52	83.33	83.57	84.52
WSEFEP	87.22	87.22	88.33	89.44	87.22	86.67	89.44
ADFES	92.27	92.42	91.67	91.67	93.94	93.18	91.67

Table 3 Recognition accuracy of CSP with different weights for six expressions

Dataset	Binary	fibonacci	Prime	Natural	Squares	Odd	Even
JAFFE	62.70	63.66	61.61	62.08	61.52	62.64	62.02
MUG	86.22	86.59	85.59	86.59	86.44	86.44	86.89
CK+	90.79	90.70	91.13	91.02	90.79	90.69	91.13
OULU	73.33	73.54	73.13	72.92	73.40	73.68	73.82
TFEID	93.50	95.08	93.75	95.08	92.83	94.25	93.42
KDEF	82.86	83.57	82.14	82.38	82.62	83.33	81.90
WSEFEP	85.56	87.22	87.22	86.11	86.67	86.11	85.56
ADFES	92.42	90.91	90.91	90.91	92.42	92.42	92.42

Table 4 Recognition accuracy of RCSP with different weights for six expressions

Dataset	Binary	fibonacci	Prime	Natural	Squares	Odd	Even
JAFFE	63.08	60.00	59.97	60.06	59.97	61.11	60.06
MUG	86.59	87.63	88.00	88.07	87.59	87.26	88.07
CK+	92.21	91.99	91.88	91.99	92.42	91.77	91.99
OULU	74.23	74.30	74.31	75.10	74.30	74.86	74.38
TFEID	95.17	94.58	94.58	95	95.08	95	95
KDEF	83.81	83.57	83.33	84.05	83.57	83.33	84.05
WSEFEP	87.22	87.22	88.33	88.89	86.67	87.78	88.89
ADFES	91.67	92.42	92.42	91.67	93.18	91.67	91.67

by taking the mean of accuracy obtained from each subject. The results generated by the authors might be different from the accuracy reported in the papers, because of the different image size and block size considered for experimental evaluation. For methods other than the variants of binary patterns, the results are taken from their corresponding papers for comparison analysis.

Experiments for six expressions

The experiments for six expressions were conducted on all the datasets captured in the lab environment. The proposed methods have been implemented with different weights and the results are tabulated. In Table 2, for each dataset, the recognition accuracy comparison of RCP, in Table 3, the recognition accuracy comparison of CSP and in Table 4, the recognition accuracy comparison of RCSP with differ-

ent weights is shown. For all the tables listed below, the weight that had achieved the highest recognition accuracy is highlighted in bold. For some datasets, there are multiple weights that have achieved highest recognition accuracy. In such cases, the weight with least fv length is highlighted in bold. For example, in Table 2, for WSEFEP dataset, RCP with natural and even weights achieved highest recognition accuracy. But, RCP with natural weights is only highlighted in bold as its fv length is lesser when compared to even weights.

Among the proposed methods with different weights, CSP method with Fibonacci weights achieved highest recognition accuracy for JAFFE dataset. RCP method with natural weights achieved highest recognition accuracy for MUG, KDEF and WSEFEP datasets. RCP method with squares weights achieved highest recognition accuracy for TFEID and ADFES datasets. RCP method with prime weights achieved highest recognition accuracy for OULU-

Table 5 Comparison analysis with existing variants of binary patterns for six expressions

Method	JAFFE	MUG	CK+	OULU	TFEID	KDEF	WSEFEP	ADFS
LBP [45]	56.66	82.65	89.97	75.34	92.42	80.95	87.22	90.16
LDP [17]	52.77	82.87	90.84	72.43	93.67	80.95	86.67	90.91
LDN [41]	56.66	81.96	88.84	72.29	93.33	82.62	87.22	88.64
CSLBP [27]	53.28	83.94	90.68	71.53	94.25	81.67	84.44	87.12
LGC [55]	58.89	86.22	89.61	73.47	95.00	83.81	87.78	90.91
LDTP [40]	55.55	82.04	89.60	72.29	93.25	83.57	87.78	88.64
LDTerP [42]	58.54	80.15	85.89	68.54	90.75	81.43	81.11	84.89
RADAP [35]	57.22	83.48	90.63	75.90	94.17	82.38	87.78	91.67
CP [57]	58.74	86.23	91.34	73.98	92.92	84.04	86.67	91.67
Proposed	63.66	88.07	92.42	76.18	95.58	84.52	89.44	93.94

VIS dataset. RCSP method with squares weights achieved highest recognition accuracy for CK+ dataset. Hence, these methods are chosen for comparison analysis with the existing methods. In Table 5, the comparison analysis of the proposed methods with the existing variants of binary patterns, implemented in our environment setup is shown. In Table 6, the comparison analysis of the proposed method with the existing methods is shown.

The comparison analysis for **JAFFE** dataset with the existing variants of binary patterns is reported in the second column of Table 5. From Table 5, the proposed CSP method outperformed existing variants of binary patterns such as LDTP, LDTerP, RADAP and CP by 8.11%, 5.12%, 6.44% and 4.92% respectively. From Table 6, the proposed method could also outperform deep learning methods such as VGG19, ResNet50, IFRBC and WLGC-HD by 3.11%, 4.22%, 1.37% and 2.96% respectively. The comparison analysis for **MUG** dataset with the existing variants of binary patterns is reported in the third column of Table 5. From Table 5, the proposed RCP method outperformed existing variants of binary patterns such as LDTP, LDTerP, RADAP and CP by 6.03%, 7.92%, 4.59% and 1.84% respectively. From Table 6, the proposed method could also outperform deep learning methods such as VGG19, ResNet50, MobileNet and HiNet by 2.85%, 1.19%, 11.37% and 0.27% respectively.

The comparison results for **CK+** dataset with the existing variants of binary patterns are reported in the fourth column of Table 5. From Table 5, the proposed RCSP method outperformed existing variants of binary patterns such as LDTP, LDTerP, RADAP and CP by 2.82%, 6.53%, 1.79% and 1.08% respectively. From Table 6, the proposed method could also outperform some deep learning methods such as VGG19, ResNet50, MobileNet and HiNet by 1.11%, 3.1%, 13.72% and 1.02% respectively. The comparison analysis for **OULU-VIS Strong** dataset with the existing variants of binary patterns is reported in the fifth column of Table 5. From Table 5, the proposed RCP method outperformed existing variants of binary patterns such as LDTP, LDTerP, RADAP

Table 6 Comparison analysis with the existing methods for six expressions

Dataset	Method	Accuracy
JAFFE	VGG19 [35]	60.55
	ResNet50 [35]	59.44
	IFRBC [51]	62.29
	WLGC-HD [24]	60.7
	Proposed	63.66
MUG	VGG19 [35]	85.22
	ResNet50 [35]	86.88
	MobileNet [60]	76.7
	HiNet [60]	87.8
	Proposed	88.07
CK+	VGG16 [35]	91.31
	ResNet50 [35]	89.32
	MobileNet [60]	78.7
	HiNet [60]	91.40
	WLGC-HD [24]	72.80
OULU	Proposed	92.42
	VGG16 [60]	73.4
	ResNet50 [60]	73.1
	MobileNet [60]	60.9
	HiNet [60]	74.3
KDEF	Proposed	76.18
	ICVR [51]	76.31
	IFRBC [51]	77.98
	Proposed	84.52

and CP by 3.89%, 7.64%, 0.28% and 2.2% respectively. From Table 6, the proposed method could also outperform some deep learning methods such as VGG16, ResNet50, MobileNet and HiNet by 2.78%, 3.08%, 15.28% and 1.88%, respectively.

The comparison analysis for **TFEID** dataset with the existing variants of binary patterns is reported in the sixth column of Table 5. From Table 5, the proposed RCP method

Table 7 Recognition accuracy of RCP with different weights for seven expressions

Dataset	Binary	fibonacci	Prime	Natural	Squares	Odd	Even
JAFFE	58.09	60.36	58.04	56.22	58.00	57.56	56.22
MUG	80.25	83.62	83.68	82.48	82.54	82.03	82.48
CK+	85.98	85.96	85.88	86.02	85.98	86.11	85.95
OULU	75.71	75.24	74.87	74.76	74.67	74.76	74.76
TFEID	94.36	93.99	95.12	96.19	94.76	93.99	96.19
KDEF	81.02	80.82	80.61	80.82	81.22	80.82	80.82
WSEFEP	84.29	84.29	83.81	83.81	84.29	84.29	84.29
ADFES	93.40	91.56	92.21	92.21	94.16	94.16	92.21
RAF	72.36	71.31	72.62	71.41	72.62	72.88	72.20
FERG	99.83	98.73	98.74	99.90	99.21	98.47	99.90

Table 8 Recognition accuracy of CSP with different weights for seven expressions

Dataset	Binary	fibonacci	Prime	Natural	Squares	Odd	Even
JAFFE	61.32	62.19	61.80	61.73	60.32	60.35	61.77
MUG	80.57	81.65	81.59	80.95	81.08	81.90	81.21
CK+	86.86	86.55	86.78	86.68	86.61	85.85	86.68
OULU	72.95	73.35	72.86	72.86	73.81	72.95	72.64
TFEID	94.64	94.29	95.00	94.29	94.29	94.64	93.93
KDEF	81.43	81.22	81.43	82.45	81.02	81.02	82.45
WSEFEP	84.29	84.29	84.76	85.71	84.29	83.33	85.71
ADFES	91.56	92.21	90.91	90.91	91.56	91.56	90.91
RAF	76.04	76.80	76.66	75.75	77.02	75.13	75.75
FERG	99.99	99.71	99.97	99.51	99.87	99.10	99.16

Table 9 Recognition accuracy of RCSP with different weights for seven expressions

Dataset	Binary	fibonacci	Prime	Natural	Squares	Odd	Even
JAFFE	60.97	60.91	61.00	61.04	60.00	60.50	61.04
MUG	82.54	83.57	83.30	83.75	82.98	83.49	83.75
CK+	87.68	87.30	87.62	87.61	88.00	87.24	87.61
OULU	74.76	73.96	74.45	73.45	73.66	73.96	73.45
TFEID	94.40	93.99	94.36	95.06	94.90	94.70	95.10
KDEF	82.65	82.24	82.65	82.45	82.65	81.63	82.45
WSEFEP	83.81	84.76	84.29	85.71	83.33	85.24	85.71
ADFES	92.86	92.21	92.21	91.56	93.51	91.56	91.56
RAF	76.27	77.64	77.15	76.56	77.12	76.89	76.56
FERG	99.99	99.46	99.44	99.99	99.97	99.87	99.99

outperformed existing variants of binary patterns such as LDTP, LDTP, RADAP and CP by 2.33%, 4.83%, 1.41% and 2.66% respectively. The comparison analysis for **KDEF** dataset with the existing variants of binary patterns is reported in the seventh column of Table 5. From Table 5, the proposed RCP method outperformed existing variants of binary patterns such as LDTP, LDTP, RADAP and CP by 0.95%, 3.09%, 2.14% and 0.48% respectively. From Table 6, the proposed method could also outperform the existing methods such as ICVR and IFRBC by 8.21% and 6.54% respectively.

The comparison analysis for **WSEFEP** dataset with the existing variants of binary patterns is reported in the eighth column of Table 5. From Table 5, the proposed RCP method outperformed existing variants of binary patterns such as LDTP, LDTP, RADAP and CP by 1.66%, 8.33%, 1.66% and 2.77% respectively. The comparison analysis for **ADFES** dataset are reported in the ninth column of Table 5. From Table 5, the proposed RCP method outperformed existing variants of binary patterns such as LDTP, LDTP, RADAP and CP by 5.3%, 9.05%, 2.27% and 2.27% respectively.

Table 10 Comparison analysis with existing variants of binary patterns for seven expressions

Method	JAFFE	MUG	CK+	OULU	TFEID	KDEF	WSEFEP	ADFES
LBP [45]	53.65	76.16	83.96	65.71	92.02	78.16	82.38	87.66
LDP [17]	52.10	78.70	84.80	69.16	93.57	80.61	80.95	90.26
LDN [41]	54.87	77.85	83.35	70.89	93.51	80.75	83.81	91.56
CSLBP [27]	52.40	79.37	85.51	57.98	89.44	80.20	79.44	86.36
LGC [55]	57.59	82.03	86.71	71.13	93.43	81.02	84.29	90.91
LDTP [40]	51.32	78.70	83.08	68.86	93.15	80.81	85.71	85.71
LDTerP [42]	51.70	78.11	81.40	64.53	90.18	77.35	79.52	79.87
RADAP [35]	56.20	80.26	84.60	74.34	93.27	80.20	87.42	90.91
CP [57]	58.50	81.26	85.94	72.08	93.27	81.43	82.86	92.21
Proposed	62.19	83.75	88.00	75.71	96.19	82.65	85.71	94.16

Experiments for seven expressions

The experiments for seven expressions were conducted on all the ten datasets. In Table 7, for each dataset, the recognition accuracy comparison of RCP, in Table 8, the recognition accuracy comparison of CSP and in Table 9, the recognition accuracy comparison of RCSP with different weights is shown. Among the proposed methods with different weights, CSP method with Fibonacci weights achieved better recognition accuracy for JAFFE dataset. CSP method with natural weights achieved better recognition accuracy for WSEFEP dataset. RCP method with binary weights achieved better recognition accuracy for OULU-VIS dataset. RCP method with natural weights achieved better recognition accuracy for TFEID dataset. RCP method with odd weights achieved better recognition accuracy for ADFES dataset. RCSP method with natural weights achieved better recognition accuracy for MUG dataset. RCSP method with squares weights achieved better recognition accuracy for CK+ dataset. RCSP method with prime weights achieved better recognition accuracy for KDEF dataset. In case of RAF dataset, RCSP method with Fibonacci weights and for FERF dataset, RCSP method with natural weights achieved better recognition accuracy and hence these methods are chosen for comparison analysis with the existing methods. The comparison analysis of the proposed methods with the existing variants of binary patterns is shown in Table 10 and the comparison analysis with the existing methods is shown in Table 11.

The comparison analysis for **JAFFE** dataset with the existing variants of binary patterns is reported in the second column of Table 10. From Table 10, the proposed CSP method outperformed existing variants of binary patterns such as LDTP, LDTerP, RADAP and CP by 10.87%, 10.49%, 5.99% and 3.69% respectively. From Table 11, the proposed method could also outperform the existing methods such as VGG19, ResNet50, DLFS and PCANet by 0.77%, 5.06%, 1.39% and 3.84% respectively. The comparison analysis for **MUG** dataset with the existing variants of binary patterns is

reported in the third column of Table 10. From Table 10, the proposed RCSP method outperformed existing variants of binary patterns such as LDTP, LDTerP, RADAP and CP by 5.05%, 5.64%, 3.49% and 2.49% respectively. From Table 11, although, the deep learning methods such as VGG19, ResNet50 and HiNet achieved 1.37%, 1.83% and 3.45% more than the proposed method, the proposed RCSP method is simple and whenever natural weights are utilized, the fv length is much lesser than that of traditional binary pattern variants.

The comparison analysis for **CK+** dataset with the existing variants of binary patterns is reported in the fourth column of Table 10. From Table 10, the proposed RCSP method outperformed the existing variants of binary patterns such as LDTP, LDTerP, RADAP and CP by 4.92%, 6.6%, 3.40% and 2.06% respectively. From Table 11, the proposed method could also outperform deep learning methods such as VGG19, ResNet50, DLFS and PCANet by 9.29%, 0.69%, 4.28% and 9.16%, respectively. Although, the method HiNet achieved 0.6% more than the proposed method, our method of feature extraction is simple and easily implementable. The comparison analysis for **OULU-VIS** dataset with the existing variants of binary patterns is reported in the fifth column of Table 10. From Table 10, the proposed RCP method outperformed the existing variants of binary patterns such as LDTP, LDTerP, RADAP and CP by 6.85%, 11.18%, 1.37% and 3.63% respectively. From Table 11, the proposed method could also outperform deep learning methods such as VGG19, ResNet50, MobileNet and HiNet by 5.21%, 10.31%, 15.31% and 3.71% respectively.

The comparison analysis for **TFEID** dataset with the existing variants of binary patterns is reported in the sixth column of Table 10. From Table 10, the proposed RCP method outperformed existing variants of binary patterns such as LDTP, LDTerP, RADAP and CP by 3.04%, 6.01%, 2.92% and 2.92% respectively. The comparison analysis for **KDEF** dataset with the existing variants of binary is reported in the seventh column of Table 10. From Table 10, the proposed

Table 11 Comparison analysis with the existing methods for seven expressions

Dataset	Method	Accuracy
JAFFE	WLGC-HD [24]	58.2
	VGG19 [35]	61.42
	ResNet50 [35]	57.13
	DLFS [52]	60.80
	PCANet [5]	58.35
	Proposed	62.19
	VGG19 [35]	85.12
MUG	ResNet50 [35]	85.58
	MobileNet [60]	68.2
	HiNet [60]	87.2
	Proposed	83.75
	VGG19 [35]	78.71
CK+	ResNet50 [35]	87.31
	HiNet [60]	88.6
	DLFS [52]	83.72
	PCANet [5]	78.84
	WLGC-HD [24]	70.60
	Proposed	88.00
	VGG19 [60]	70.5
OULU	ResNet50 [60]	65.4
	MobileNet [60]	60.4
	HiNet [60]	72.0
	Proposed	75.71
	DLFS [52]	78.60
KDEF	PCANet [5]	69.59
	Proposed	82.65
	McFIs [50]	75.00
TFEID	DAM-CNN [64]	93.36
	Proposed	96.19
	GoogleNet [53]	86.48
ADFES	AlexNet+SVM [62]	88.43
	CNN [19]	87.01
	AFM [63]	92.70
	Proposed	94.16

RCSP method outperformed existing variants of binary patterns such as LDTP, LDTP, RADAP and CP by 1.84%, 5.30%, 2.45% and 1.22% respectively. From Table 11, the proposed method could also outperform the existing methods such as DLFS and PCANet by 4.05% and 13.06% respectively.

The comparison analysis for **WSEFEP** dataset with existing variants of binary patterns is reported in the eighth column of Table 10. From Table 10, although RADAP method achieved 1.71% more than the proposed method, the feature extraction using CSP is very simple, as natural weights are

Table 12 Comparison analysis with the existing methods on RAF dataset

Method	Accuracy
DLP-CNN [29]	74.20
ICID Fusion [18]	75.40
Sadeghi et al. [43]	76.23
DCNN + RLPS [28]	72.84
IFSL [65]	76.9
Proposed	77.64

Table 13 Comparison analysis with the existing methods on FERG dataset

Method	Accuracy
DeepExpr [3]	89.02
Ensemble Multi-feature [73]	97
Adversarial NN [10]	98.2
Deep Emotion [36]	99.3
LBP-AW [23]	96.7
WLGC-HD [24]	97.9
Proposed	99.99

used. Also, RADAP uses binary weights and generates six feature values, so fv length is more, when compared to the proposed method CSP, which generates one feature value. The comparison analysis for **ADFES** dataset are reported in the ninth column of Table 10. From Table 10, the proposed RCP method outperformed existing variants of binary patterns such as LDTP, LDTP, RADAP and CP by 8.45%, 14.29%, 3.25% and 1.95% respectively. From Table 11, the proposed method could also outperform deep learning methods such as GoogleNet, AlexNet, CNN and AFM by 7.68%, 5.73%, 7.15% and 1.46% respectively.

For **RAF** dataset, the comparison analysis of the proposed methods with the existing methods is shown in Table 12. From Table 12, the proposed RCSP method outperformed some of the existing methods such as DLP-CNN, ICID Fusion, DCNN + RLPS and IFSL by 3.44%, 2.24%, 4.80% and 0.74% respectively. For **FERG** dataset, the comparison analysis of the proposed methods with the existing methods is shown in Table 13. From Table 13, the proposed RCSP method outperformed the existing methods such as DeepExpr, Ensemble Multi-feature, Adversarial NN, Deep Emotion, LBP-AW and WLGC-HD by 10.97%, 2.99%, 1.79%, 0.69%, 3.29% and 2.09% respectively.

Experiments for eight expressions

The experiments for eight expressions are performed on **TFEID** dataset. Apart from basic six plus neutral expres-

Table 14 Recognition accuracy comparison for eight expressions with different weights on TFEID Dataset

Dataset	Binary	fibonacci	Prime	Natural	Squares	Odd	Even
RCP	91.14	91.14	91.14	91.32	91.17	91.80	91.32
CSP	93.85	93.85	94.13	93.54	93.85	93.85	93.54
RCSP	92.91	92.32	92.28	92.32	92.32	92.32	92.32

Table 15 Recognition accuracy comparison for ten expressions with different weights on ADFES Dataset

Dataset	Binary	fibonacci	Prime	Natural	Squares	Odd	Even
RCP	85.20	86.22	86.97	86.02	88.03	86.11	86.62
CSP	85.01	84.96	84.05	83.54	85.01	85.01	83.09
RCSP	85.41	86.57	87.02	85.41	87.12	86.62	85.41

Table 16 Comparison with existing methods for eight expressions on TFEID dataset

Method	Accuracy
LBP [45]	90.95
LDP [17]	91.17
LDN [41]	88.06
CSLBP [27]	89.45
LDTP [40]	86.29
LGC [55]	89.95
LDTeRP [42]	88.59
RADAP [35]	90.56
CP [57]	90.48
Proposed	94.13

Table 17 Comparison with existing methods for ten expressions on ADFES dataset

Method	Accuracy
LBP [45]	83.54
LDP [17]	85.25
LDN [41]	85.20
CSLBP [27]	80.61
LDTP [40]	82.68
LDTeRP [42]	76.53
LGC [55]	85.20
RADAP [35]	84.75
CP [57]	83.43
LBP-TOP [72]	77.08
OF PID [47]	80.25
Landmark PID [48]	88.73
Proposed	88.03

sions, this dataset has one more expression named contempt. For the experimental evaluation, 336 images belonging to eight expressions have been considered. The proposed methods have been tested with different weights and the results are tabulated in Table 14 for eight expressions, in Table 15 for ten expressions (for ADFES dataset). From Table 14, CSP method with prime weights achieved better recognition and hence is chosen for comparison analysis with the existing methods. The existing binary variants have been implemented in our environment and correspondingly the recognition accuracy is reported in Table 16. From Table 16, the proposed method outperformed recent methods such as RADAP and CP by 3.57% and 3.65% respectively.

Experiments for ten expressions

The experiments for ten expressions are performed on ADFES dataset. Apart from basic six plus neutral expressions, this dataset has three more expressions, namely contempt, embarrass and pride. For the experimental evaluation, 215 images belonging to ten expressions have been considered. The proposed methods have been tested with different weights and the results are tabulated in Table 15 for ten expressions. From Table 15, RCP with squares weights

achieved better recognition and hence is chosen for comparison analysis with existing methods. The existing binary variants have been implemented in our environment and correspondingly the recognition accuracy is reported in Table 17. The results reported by Shojaeilingari et al. [48] for the methods LBP-TOP, OF PID and Landmark PID are taken directly for comparison analysis in Table 17. From Table 17, Landmark PID [48] achieved 0.70% more than the proposed method. Other than Landmark PID, the proposed method outperformed recent methods such as RADAP, CP, LBP-TOP and OF PID by 3.28%, 4.6%, 10.95% and 7.78% respectively.

Deep neural network approaches are generally preferable to handcrafted methods. However, parameters such as batch size, learning rate, number of training images, image size, and number of trainable model parameters all affect the overall recognition accuracy of deep neural networks. As a result, the accuracy of deep learning algorithms may vary based on the above factors. The results reported for deep learning methods are taken from the corresponding cited papers. The main

advantage of the proposed methods is that they are easily implementable and they extract simple and relevant features within a local neighborhood by considering both the neighboring pixel's relationship and adjacent pixel's relationship for capturing the finer appearance changes with respect to specific facial expressions. Also, from the available images in the dataset itself, the proposed methods can learn and classify the test data. The experiments are performed in person independent setup to simulate a real-world scenario. From the experimental results, the proposed methods performed better on a variety of facial expression datasets and outperformed the standard existing methods proving the robustness of the proposed descriptors.

Conclusion

The main objective of FER systems is to develop feature descriptors that could accurately classify the facial expressions into various categories. Towards realizing this task, texture-based feature descriptors, namely RCP (generates two feature values) and CSP (generates one feature value) and their fusion RCSP are proposed in this work. The experiments are conducted using RCP and CSP independently and with their fusion RCSP using different weights on a variety of facial expression datasets which include datasets captured in the lab, 'in the wild' dataset and also on an animated facial expression dataset. From the experimental results, an observation has been made that proposed methods outperformed standard existing methods proving the robustness of the proposed descriptors and in most of the experiments, RCP has achieved better recognition accuracy than CSP or RCSP and using other weights than binary resulted in an enhanced performance with decreased fv length. From the experimental results, the pixels along the radial directions proved to be efficient for capturing local minute details. As a future work, using those pixel positions, novel graph based feature descriptors with low dimensions can be proposed. The proposed descriptors can be extended in future to handle pose and illumination problems, that are more prevalent in real world. Also, further research can be carried out on different weight approaches to choose the best weight for various image processing applications.

Declarations

Conflict of interest On behalf of all authors, Mukku Nisanth Kartheek states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as

long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Aifanti N, Papachristou C, Delopoulos A (2010) The mug facial expression database. In: 11th international workshop on image analysis for multimedia interactive services WIAMIS 10, pp 1–4. IEEE
2. Alphonse AS, Shankar K, Rakkini MJ, Ananthakrishnan S, Athisayamani S, Singh AR, Gobi R (2020) A multi-scale and rotation-invariant phase pattern (MRIPP) and a stack of restricted Boltzmann machine (RBM) with preprocessing for facial expression classification. *J Ambient Intell Human Comput* 20: 1–17
3. Aneja D, Colburn A, Faigin G, Shapiro L, Mones B (2016) Modeling stylized character expressions via deep learning. In: Asian conference on computer vision. Springer, pp 136–153
4. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans Pattern Anal Mach Intell* 19(7):711–720
5. Chan TH, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2015) Pcanet: a simple deep learning baseline for image classification. *IEEE Trans Image Process* 24(12):5017–5032
6. Chen LF, Yen YS (2007) Taiwanese facial expression image database. Brain Mapping Laboratory, Institute of Brain Science, National Yang-Ming University, Taipei
7. Cootes TF, Taylor CJ, Cooper DH, Graham J (1995) Active shape models: their training and application. *Comput Vis Image Underst* 61(1):38–59
8. Donato G, Bartlett MS, Hager JC, Ekman P, Sejnowski TJ (1999) Classifying facial actions. *IEEE Trans Pattern Anal Mach Intell* 21(10):974–989
9. Ekmen B, Ekenel HK (2019) From 2d to 3d real-time expression transfer for facial animation. *Multimed Tools Appl* 78(9):12519–12535
10. Feutry C, Piantanida P, Bengio Y, Duhamel P (2018) Learning anonymized representations with adversarial neural networks. [arXiv:1802.09386](https://arxiv.org/abs/1802.09386) (arXiv preprint)
11. Goeleven E, De Raedt R, Leyman L, Verschuere B (2008) The karolinska directed emotional faces: a validation study. *Cogn Emot* 22(6):1094–1118
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
13. Hong H, Neven H, Von der Malsburg C (1998) Online facial expression recognition based on personalized galleries. In: Proceedings third IEEE international conference on automatic face and gesture recognition, pp 354–359. IEEE
14. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (arXiv preprint)
15. Huang Z, Song G, Zhao Y, Han J, Zhao X (2018) Smile recognition based on support vector machine and local binary pattern. In: 2018 IEEE 8th annual international conference on CYBER technology

- in automation, control, and intelligent systems (CYBER), pp 938–942. IEEE
16. Iqbal MTB, Abdullah-Al-Wadud M, Ryu B, Makhmudkhuaev F, Chae O (2018) Facial expression recognition with neighborhood-aware edge directional pattern (nedp). *IEEE Trans Affect Comput* 11(1):125–137
 17. Jabid T, Kabir MH, Chae O (2010) Robust facial expression recognition based on local directional pattern. *ETRI J* 32(5):784–794
 18. Ji Y, Hu Y, Yang Y, Shen F, Shen HT (2019) Cross-domain facial expression recognition via an intra-category common feature and inter-category distinction feature fusion network. *Neurocomputing* 333:231–239
 19. Jung H, Lee S, Park S, Kim B, Kim J, Lee I, Ahn C (2015) Development of deep learning-based facial expression recognition system. In: 2015 21st Korea-Japan joint workshop on frontiers of computer vision (FCV), pp 1–4. IEEE
 20. Jung H, Lee S, Park S, Lee I, Ahn C, Kim J (2015) Deep temporal appearance-geometry network for facial expression recognition. [arXiv:1503.01532](https://arxiv.org/abs/1503.01532) (arXiv preprint)
 21. Kartheek MN, Prasad MVNK, Bhukya R (2020) Local optimal oriented pattern for person independent facial expression recognition. In: Twelfth international conference on machine vision (ICMV 2019), vol. 11433. International Society for Optics and Photonics, p 114330R1–8
 22. Kas M, Ruichek Y, Messoussi R et al (2020) Multi level directional cross binary patterns: new handcrafted descriptor for SVM-based texture classification. *Eng Appl Artif Intell* 94:103743
 23. Kola DGR, Samayamantula SK (2020) A novel approach for facial expression recognition using local binary pattern with adaptive window. *Multimed Tools Appl* 20:1–20
 24. Kola DGR, Samayamantula SK (2021) Facial expression recognition using singular values and wavelet-based LGC-HD operator. *IET Biom* 20:20
 25. Kumar RJR, Sundaram M (2020) Arumugam N Facial emotion recognition using subband selective multilevel stationary wavelet gradient transform and fuzzy support vector machine. *Vis Comput* 20:1–15
 26. Kumar YR, Narayanappa C, Dayananda P (2020) Weighted full binary tree-sliced binary pattern: an RGB-D image descriptor. *Heliyon* 6(5):e03751
 27. Lai CC, Ko CH (2014) Facial expression recognition based on two-stage features extraction. *Optik-Int J Light Electron Opt* 125(22):6678–6680
 28. Li H, Xu H (2020) Deep reinforcement learning for robust emotional classification in facial expression recognition. *Knowl Based Syst* 20:106172
 29. Li S, Deng W (2018) Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Trans Image Process* 28(1):356–370
 30. 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, pp 94–101. IEEE
 31. Lyons M, Akamatsu S, Kamachi M, Gyoba J (1998) Coding facial expressions with gabor wavelets. In: Proceedings of third IEEE international conference on automatic face and gesture recognition, pp 200–205. IEEE
 32. Maheswari VU, Varaprasad G, Raju SV (2020) Local directional maximum edge patterns for facial expression recognition. *J Ambient Intell Human Comput* 20:1–9
 33. Makhmudkhuaev F, Abdullah-Al-Wadud M, Iqbal MTB, Ryu B, Chae O (2019) Facial expression recognition with local prominent directional pattern. *Signal Process Image Commun* 74:1–12
 34. Makhmudkhuaev F, Iqbal MTB, Ryu B, Chae O (2019) Local directional-structural pattern for person-independent facial expression recognition. *Turk J Electr Eng Comput Sci* 27(1):516–531
 35. Mandal M, Verma M, Mathur S, Vipparthi SK, Murala S, Kumar DK (2019) Regional adaptive affinity patterns (RADAP) with logical operators for facial expression recognition. *IET Image Proc* 13(5):850–861
 36. Minaee S, Abdolrashidi A (2019) Deep-emotion: facial expression recognition using attentional convolutional network. [arXiv:1902.01019](https://arxiv.org/abs/1902.01019) (arXiv preprint)
 37. Olszanowski M, Pochwatko G, Kuklinski K, Scibor-Rylski M, Lewinski P, Ohme RK (2015) Warsaw set of emotional facial expression pictures: a validation study of facial display photographs. *Front Psychol* 5:1–8
 38. Reddy PCS, Rao PVP, Reddy PKK, Sridhar M (2019) Motif shape primitives on Fibonacci weighted neighborhood pattern for age classification. *Soft computing and signal processing*. Springer, Berlin, pp 273–280
 39. Revina IM, Emmanuel WS (2019) MDTP: a novel multi-directional triangles pattern for face expression recognition. *Multimed Tools Appl* 78(18):26223–26238
 40. Rivera AR, Castillo JR, Chae O (2015) Local directional texture pattern image descriptor. *Pattern Recogn Lett* 51:94–100
 41. Rivera AR, Castillo JR, Chae OO (2012) Local directional number pattern for face analysis: face and expression recognition. *IEEE Trans Image Process* 22(5):1740–1752
 42. Ryu B, Rivera AR, Kim J, Chae O (2017) Local directional ternary pattern for facial expression recognition. *IEEE Trans Image Process* 26(12):6006–6018
 43. Sadeghi H, Raie AA (2019) Human vision inspired feature extraction for facial expression recognition. *Multimed Tools Appl* 78(21):30335–30353
 44. Saurav S, Gidde P, Saini R, Singh S (2021) Dual integrated convolutional neural network for real-time facial expression recognition in the wild. *Vis Comput* 20:1–14
 45. Shan C, Gong S, McOwan PW (2009) Facial expression recognition based on local binary patterns: a comprehensive study. *Image Vis Comput* 27(6):803–816
 46. Shen S, Si S (2017) Facial expression recognition based on LGC in 5 × 5 neighborhood. *Intell Comput Appl* 7:47–48
 47. Shojaeilangari S, Yau WY, Nandakumar K, Li J, Teoh EK (2015) Robust representation and recognition of facial emotions using extreme sparse learning. *IEEE Trans Image Process* 24(7):2140–2152
 48. Shojaeilangari S, Yau WY, Teoh EK (2016) Pose-invariant descriptor for facial emotion recognition. *Mach Vis Appl* 27(7):1063–1070
 49. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (arXiv preprint)
 50. Subramanian K, Radhakrishnan VB, Ramasamy S (2014) Database independent human emotion recognition with meta-cognitive neuro-fuzzy inference system. In: 2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP), pp 1–6. IEEE
 51. Sun Z, Zp Hu, Wang M, Zhao S (2017) Individual-free representation-based classification for facial expression recognition. *Signal Image Video Process* 11(4):597–604
 52. Sun Z, Hu Zp, Wang M, Zhao SH (2019) Dictionary learning feature space via sparse representation classification for facial expression recognition. *Artif Intell Rev* 51(1):1–18
 53. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9

54. Tong Y, Chen R (2019) Local dominant directional symmetrical coding patterns for facial expression recognition. *Comput Intell Neurosci* 2019:1–13
55. Tong Y, Chen R, Cheng Y (2014) Facial expression recognition algorithm using LGC based on horizontal and diagonal prior principle. *Optik* 125(16):4186–4189
56. Tuncer T, Dogan S, Abdar M, Plawiak P (2020) A novel facial image recognition method based on perceptual hash using quintet triple binary pattern. *Multimed Tools Appl* 1:1–21
57. Tuncer T, Dogan S, Ataman V (2019) A novel and accurate chess pattern for automated texture classification. *Phys A Stat Mech Appl* 536:122584
58. Turk M, Pentland A (1991) Face recognition using eigenfaces. In: *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pp 586–587
59. Van Der Schalk J, Hawk ST, Fischer AH, Doosje B (2011) Moving faces, looking places: validation of the Amsterdam dynamic facial expression set (ADFES). *Emotion* 11(4):907–920
60. Verma M, Vipparthi SK, Singh G (2019) Hinet: hybrid inherited feature learning network for facial expression recognition. *IEEE Lett Comput Soc* 2(4):36–39
61. Viola P, Jones MJ (2004) Robust real-time face detection. *Int J Comput Vis* 57(2):137–154
62. Vo DM, Le TH (2016) Deep generic features and SVM for facial expression recognition. In: *2016 3rd national foundation for science and technology development conference on information and computer science (NICS)*, pp 80–84. IEEE
63. Wu BF, Lin CH (2018) Adaptive feature mapping for customizing deep learning based facial expression recognition model. *IEEE Access* 6:12451–12461
64. Xie S, Hu H, Wu Y (2019) Deep multi-path convolutional neural network joint with salient region attention for facial expression recognition. *Pattern Recogn* 92:177–191
65. Yan Y, Zhang Z, Chen S, Wang H (2020) Low-resolution facial expression recognition: a filter learning perspective. *Signal Process* 169:107370
66. Yang J, Wang X, Han S, Wang J, Park DS, Wang Y (2019) Improved real-time facial expression recognition based on a novel balanced and symmetric local gradient coding. *Sensors* 19(8):1899
67. Yang S, Bhanu B (2011) Facial expression recognition using emotion avatar image. In: *Face and gesture 2011*, pp 866–871. IEEE
68. Zeng N, Li H, Peng Y (2021) A new deep belief network-based multi-task learning for diagnosis of Alzheimer's disease. *Neural Comput Appl* 20:1–12
69. Zeng N, Wang Z, Zhang H, Kim KE, Li Y, Liu X (2019) An improved particle filter with a novel hybrid proposal distribution for quantitative analysis of gold immunochromatographic strips. *IEEE Trans Nanotechnol* 18:819–829
70. Zeng N, Zhang H, Song B, Liu W, Li Y, Dobaie AM (2018) Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing* 273:643–649
71. Zhao G, Huang X, Taini M, Li SZ, Pietikäinen M (2011) Facial expression recognition from near-infrared videos. *Image Vis Comput* 29(9):607–619
72. Zhao G, Pietikainen M (2007) Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans Pattern Anal Mach Intell* 29(6):915–928
73. Zhao H, Liu Q, Yang Y (2018) Transfer learning with ensemble of multiple feature representations. In: *2018 IEEE 16th international conference on software engineering research, management and applications (SERA)*, pp 54–61. IEEE
74. Zhou H, Wang R, Wang C (2008) A novel extended local-binary-pattern operator for texture analysis. *Inf Sci* 178(22):4314–4325

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.