



# Graph based feature extraction and hybrid classification approach for facial expression recognition

L. B. Krithika<sup>1</sup> · G. G. Lakshmi Priya<sup>1</sup>

Received: 27 September 2019 / Accepted: 7 July 2020 / Published online: 14 July 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

In the current trends, face recognition has a remarkable attraction towards favorable and inquiry of an image. Several algorithms are utilized for recognizing the facial expressions, but they lack in the issues like inaccurate recognition of facial expression. To overcome these issues, a Graph-based Feature Extraction and Hybrid Classification Approach (GFE-HCA) is proposed for recognizing the facial expressions. The main motive of this work is to recognize human emotions in an effective manner. Initially, the face image is identified using the Viola–Jones algorithm. Subsequently, the facial parts such as right eye, left eye, nose and mouth are extracted from the detected facial image. The edge-based invariant transform feature is utilized to extract the features from the extracted facial parts. From this edge-based invariant features, the dimensions are optimized using Weighted Visibility Graph which produces the graph-based features. Also, the shape appearance-based features from the facial parts are extracted. From these extracted features, facial expressions are recognized and classified using a Self-Organizing Map based Neural Network Classifier. The performance of this GFE-HCA approach is evaluated and compared with the existing techniques, and the superiority of the proposed approach is proved with its increased recognition rate.

**Keywords** Facial expression · Emotion recognition · Weighted visibility graph · Self-organizing map based neural network

## 1 Introduction

Emotion is a unique feature of human, widely facial expressions are utilized in the non-verbal communication process and also plays a significant role in the process of recognizing emotions (Kim et al. 2019). Next to the voice tone, the facial expressions are most essentially considered for emotional communication. This permits an individual person to express their emotional state, which is also an indicator of feelings. Therefore, the information on the facial expressions can be utilized in automatic systems of emotion recognition. Depending on the facial expressions, the emotional states are categorized into six states such as joy, anger, surprise, disgust, neutral and fear (Ratliff and Patterson 2008). The Human Machine Interaction (HMI) systems become more essential because of the significance of facial

expression recognition. Thus, various machine learning, as well as computer vision algorithms, are introduced for designing the HMI (Fuentes et al. 2017; Samara et al. 2019). Moreover, there are so many annotated face databases with basic expressions that are portrayed by human beings and some faces are captured in uncontrolled settings. This facial expression recognition approaches helps to classify the faces in a given image or sequence of images (Mollahosseini et al. 2016).

### 1.1 Problem description

Even though several conventional techniques are utilized for detecting and classifying the facial expressions, still they did not detect and classify the facial expressions in a flexible manner (Mollahosseini et al. 2016). The major reason for this issue lies in determining the accurate training dataset specifically for emotions like fear or sadness becomes a challenge.

✉ G. G. Lakshmi Priya  
lakshmi priya.gg@vit.ac.in

L. B. Krithika  
krithika.lb@vit.ac.in

<sup>1</sup> School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India

## 1.2 Objectives

Depending upon the problems that are identified previously, this research paper has the objectives as follows: To effectively detect and classify the facial expressions, a GFE-HCA: graph-based feature extraction and hybrid classification approach are proposed. Viola–Jones algorithm is utilized to discover the facial image (Jensen 2008). The features from facial parts are extracted by using the edge-based invariant feature transform method and then the dimensions of the extracted features are optimized by implementing the weighted visibility graph (Xie and Hu 2017). Similarly, the shape-based features are extracted using facial shape appearances. Finally, the facial expression is recognized and classified by proposing the neural network-based self-organizing classifier.

## 1.3 Organization

The subsequent sections of this work is organized as follows: Sect. 2 analyses the traditional methods utilized for detecting and recognizing the facial expressions. Section 3 elucidates the proposed method for accurate facial expression recognition. Section 4 assesses the results of both traditional and proposed systems with various performance metrics. Finally, this work is determined in Sect. 5.

## 2 Related works

This division discusses the traditional approaches that are utilized for recognizing facial expressions. Also, the merits and demerits of the existing techniques are analyzed. This section is sub-categorized into three sub-divisions such as pre-processing, feature extraction and classification. The techniques and algorithm that are utilized for these sub-divisions are discussed as follows.

### 2.1 Pre-processing

The nature of axis-symmetric faces and designed a framework for producing axis symmetrical virtual dictionary approximately to increase the precision of face recognition was considered (Xu et al. 2016). The proposed system was simple to implement and mathematically tractable. The proposed method offered better results compared to that of other pre-processing approaches.

A simple solution for emotion recognition proposed using the group of the convolutional neural network along with some image pre-processing techniques (Lopes et al. 2017). It was used in real-time applications and took less time.

A novel system to recognize the facial expression, which was depending on the ensembles of descriptors that utilized various pre-processing approaches was proposed (Lumini et al. 2017). The performance of this presented system was evaluated using two distinct datasets, such as FERET and LFW. This system performed well on both the datasets and attained increased performance rates. Also, this novel system offered better results with higher accuracy.

The authors utilized stationary wavelet transform for extracting the features to recognize the facial expression (Qayyum et al. 2017). Here the pre-processing was done using normalization histogram equalization for detecting the face more accurately. With the pre-processing step, the facial parts were determined effectively, and this helped to extract the features in an efficient way. After this, the classification process was carried out to classify the facial expressions. These results offered an increased average recognition rate for different datasets. From the results, it was noted that this approach offered better recognition results with increased accuracy compared to that of the traditional techniques.

The author introduced a novel ensemble of descriptors to recognize the facial expressions from the Patterns of the Oriented Edge Magnitudes (POEM) descriptor (Nanni et al. 2017). Here, various ensembles were developed by using the pre-processing approaches. This approach was evaluated with different datasets, such as LFW and FERET. The approach performed well on both the datasets with accurate results.

This section discussed different traditional pre-processing techniques for different facial datasets. It included main benefits like high average recognition rate and less processing time, but the need for improving the accuracy is a constant challenge.

### 2.2 Feature extraction

A novel approach was adopted for analyzing the facial expressions with the discovery of some common information from several expressions (Zhong et al. 2014). The overall expressions and the specific expressions are distinguished through the common and particular patches. For locating the distinct patches, two stages multi-task sparse learning was used. In this approach, the dominant patches were found for all the expressions. The specific patches were analyzed in the individual expression by combining both the related facial expression recognition and facial verification tasks. The main advantage of this method was patch learning which provided better performance.

An approach depending upon the multi-level Haar wavelet was used to extract the features of the appearance from the obvious face regions on dissimilar scales (Goyani and Patel 2017). Initially, the approach segmented the most informative geometric elements like eyes, eyebrows, mouth

by utilizing the Viola–Jones cascade object detector. Here, the Haar features of the segmented elements were discovered. Then the process of classification was done by utilizing the one vs all logistic regression model. The advantage of extracting the Haar features was easy computation and helped to represent the signals which were presented in low dimensions in an effective manner. But, it was difficult to recognize the expressions in dynamic images.

A new feature extraction background which helped in signifying the differences in the facial expression as a linear mixture of localized basis functions was presented (Sariyanidi et al. 2017). Here, a sparse linear model was trained with the Gabor phase shifts, which was computed from the facial video. It helped to obtain the linear basis function of the anticipated framework. The difficulties of generalization were addressed by this framework and achieved by similar learning parameters by recognizing both the posed and spontaneous micro-expressions.

The authors announced a novel approach for extracting the noticeable areas of the face with Self-Organizing Map based Neural Network classifier (Liu et al. 2017). Firstly, the normalization occurs on the prominent areas of similar position on the face. On the salient areas, the features such as LBP and HOG are extracted. The Principal Component Analysis was used to decrease the dimension of the fusion features. The peak expression frames are used in this method with the help of the salient area definitude method. The face expressions were classified using numerous classifiers. The alignment of the particular areas with several expressions in the face was done through the normalization of salient regions. Besides, there were limitations in recognition which was not performed with inaccurate landmarks, and also there was no improvement in recognition with fewer image data.

A novel approach was presented to recognize the facial expression with three steps such as feature extraction, feature optimization and emotion recognition (Mistry et al. 2016). The face representations are originally extracted using altered Local Binary Patterns. The facial expressions are separated using embedded PSO algorithm through finding the important and discriminative features. Different classifiers are used for recognizing the emotions like happy, sad, anger, surprise, fear, disgust and neutral. This approach is assessed using extended Cohn Kanade and MMI databases. The advantage of this work is computational cost and speed of the convergence.

The author introduced a novel approach to discover the specificity of the expression variation in the face (Xie et al. 2017). The specificity of the expression was signified through the triplet wise expression recognition in accordance with Action Unit (AU) and patch weight optimization. To acquire better generalization, the active AUs are detected and presented in testing samples using sparse representation-based approach. The accuracy is enhanced in this approach,

and the cross-database was better in its performance. In real-time applications, the recognition in cross-database was difficult for acquiring better results.

From this section, different traditional feature extraction techniques have discoursed. It includes some drawbacks like difficulty in acquiring better results, no improvement in recognition with fewer image data and difficulty in recognizing the expressions in dynamic images.

### 2.3 Classification

An automatic facial expression system for extracting the salient data from their video sequences was developed (Fang et al. 2014). The framework explored the parametric space, and it was tested with six machine learning techniques. An entire collection of machine learning approaches were working with the specific task of expression in face recognition. These approaches performed well in the case of face recognition with accurate results.

The face identification system was proposed corresponding to the set of three face detectors (Yu and Zhang 2015). The Face Expression Recognition (FER) challenge offered the pre-trained huge dataset, and each CNN technique was initialized randomly. The multiple CNN model was included with two systems for learning the collaborative loads of their network responses.

The authors utilized the dataset for standard expression of face, and it was made by influencing the societal images and the deep model system was trained based on the standard database (Peng et al. 2016). The specific keywords were used for image search, and it was obtained from social labelled images. The mislabelled images were removed by junk image cleaning. The spontaneous expressions were recognized by deep Convolutional Neural Network.

For emotion recognition, the author proposed a novel technique in which the recognition of the face is taken from the single image frame which used the combination of the geometric features as well as /the appearance with the help of Support Vector Machine classification (Ghimire et al. 2017a, b). By isolating the face region into the regular grid, appearance aspects for the facial expression recognitions were calculated. Also, the geometric aspects are extracted from the respective areas.

The emotion recognition technique had been evaluated on Cohn-Kanade dataset. Facial Expression Recognition and Analysis challenge (FERA) was introduced to detect the Action Units and intensity on more number of data where a broad set of videos comprising nine outlooks were generated (Valstar et al. 2017). FACS interpreted dataset was absorbed for estimating the expressions on several camera visions, extending risky postures. The problems were in estimating the intensity in expression and also in detecting head poses were interpreted in this method. The geometric features used

for the standard results improves the potential for the specific challenges.

The authors present a real-time facial expression recognition system on smart phones (Suk and Prabhakaran 2014). The system returns the recognized expression by the SVM classifiers for facial expression recognition with the dynamic features. The results of the work show that the system was effective on mobile devices with speed and accuracy.

The above study conveys that existing image processing techniques used for emotion recognition had problems such as inaccurate recognition of the emotion, less efficiency, which degrades the performance in categorizing the facial expressions. To solve these issues, the proposed work aims to introduce an efficient face recognition system by using improved feature extraction and classification techniques.

### 3 Proposed work

This section describes the overall flow of the GFE-HCA system for identifying the facial images from a certain video sequence. The Fig. 1 represents the flow of the

GFE-HCA system in which the input facial image is provided from which the foreground images will be detected using Viola–Jones algorithm (Jensen 2008). It detects the face region more accurately and faster when compared to other techniques. Thus, the Viola–Jones algorithm is utilized for extracting the facial regions followed with the extraction of facial parts such as right eye, left eye, nose and mouth, which is depicted in Fig. 2.

The steps performed after the detection of facial parts are as follows.

- The shape appearance features extracted with respect to the appearance of shape in the facial parts and the edge-based feature extraction is performed by means of edge-based invariant transform.
- Using the extracted edge-based invariant transform feature, an Eigen value-based Weighted Visibility Graph is constructed. From this constructed graph, the following features such as link density in the graph, average closeness centrality, graph entropy, average distribution weight of the graph and average degree of a graph are then extracted.

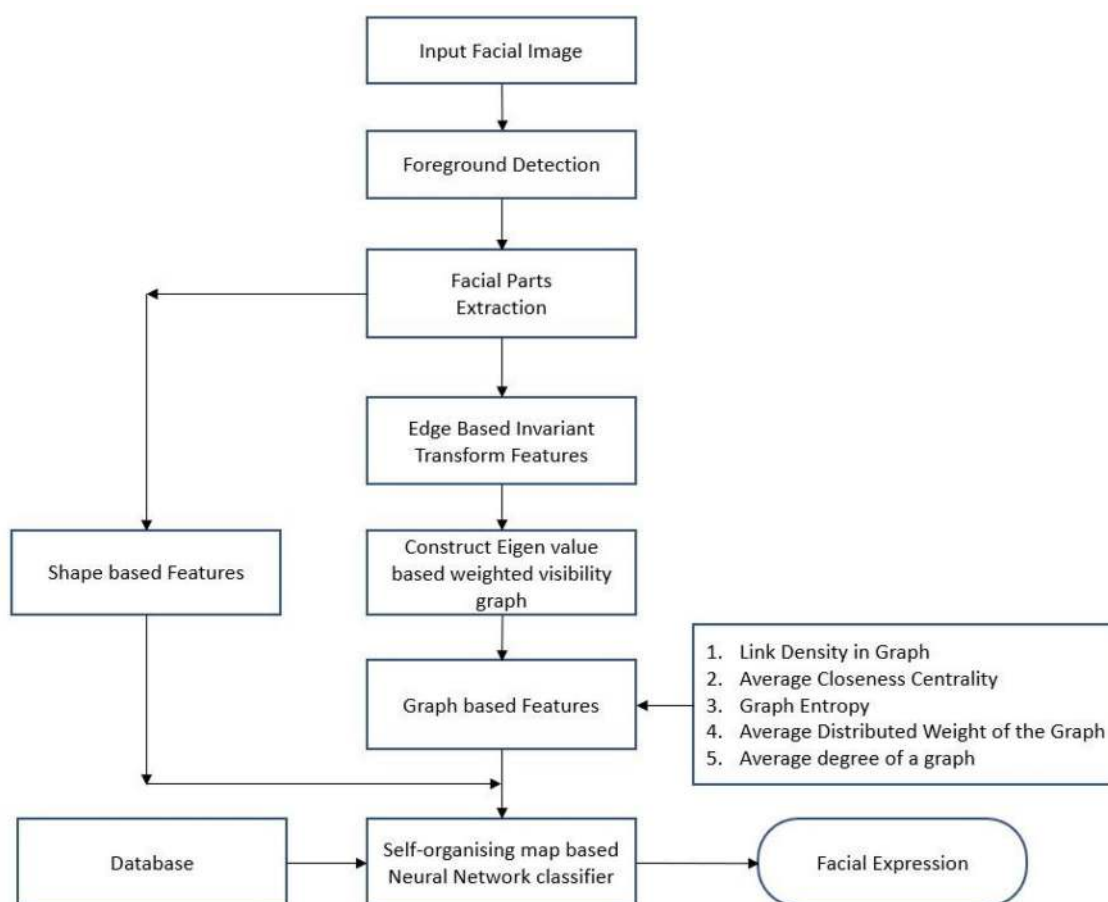
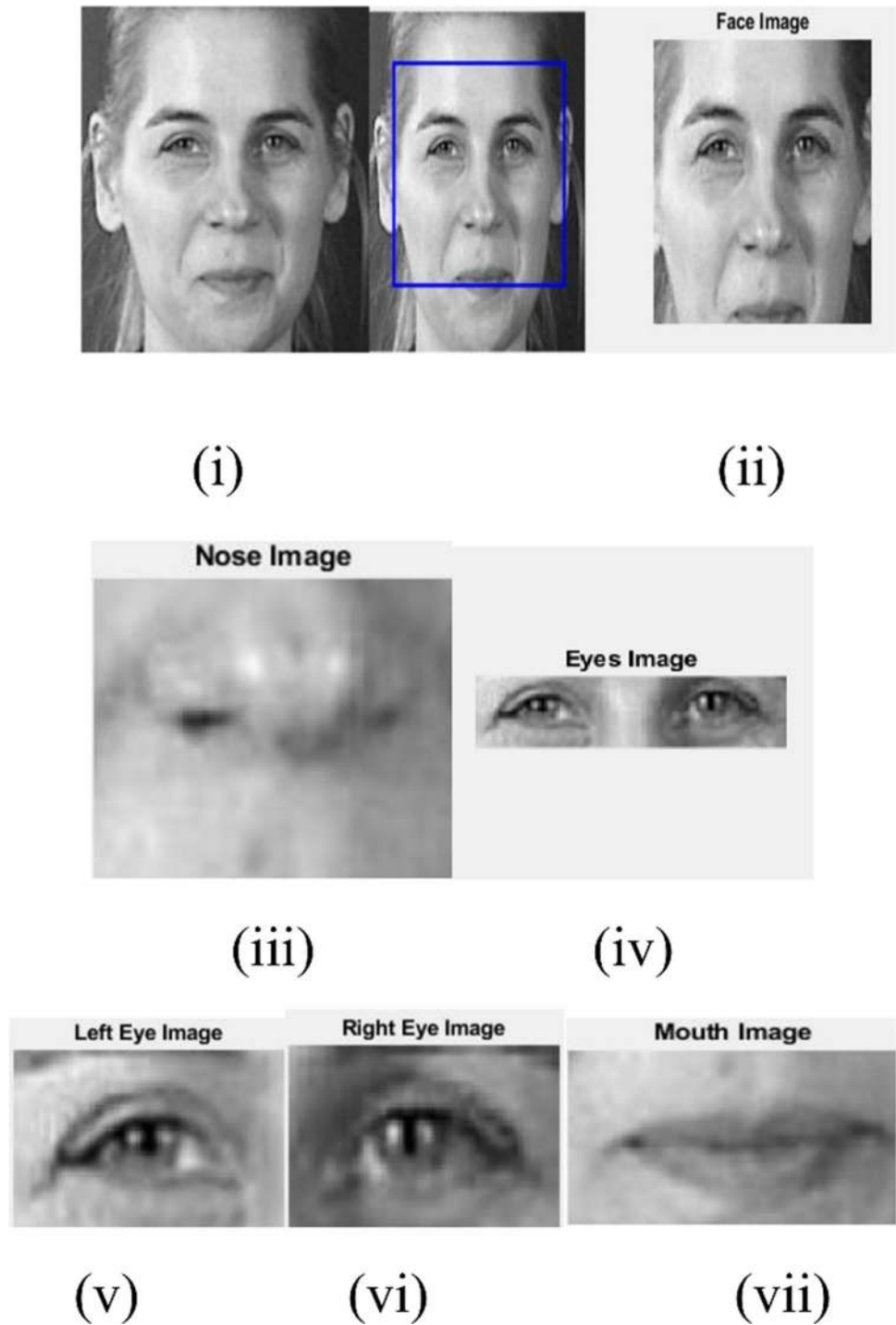


Fig. 1 Overall flow of the proposed system

**Fig. 2** (i) input facial image  
(ii) detected face image  
(iii) extracted nose image  
(iv) extracted eye image  
(v) extracted left eye image  
(vi) extracted right eye image and  
(vii) extracted mouth image



- The extracted shape-based and graph-based features will be classified with the help of the Self-Organizing Map based Neural Network Classifier. Finally, performance analysis is calculated to prove the efficiency of the proposed method.

### 3.1 Edge-based invariant transform

An edge is a point in a digital image on which the intensity deviates abruptly. The parts of the facial image are extracted by using the edge-based invariant transform. The invariant features are extracted here for predicting the uniqueness among the emotions, by suppressing the common

information of the images. The Algorithm-I for the edge-based invariant transform is provided below. In this, the face parts are extracted initially by the prediction of edge regions on applying the Canny Edge detector (Ramadass et al. 2018) mechanism. The Canny Edge detector is one of the best prevalent algorithm for the detection of edges. It only requires minimum numerical computations and also satisfies the following specifications like good detection, noise sensitivity, good localization, speed and efficiency. The filtered coefficients of the parts are then estimated and

the corners are predicted by using the filtered coefficients for each individual parts of the face. The invariant features are estimated by the detected corner in which the area, pixels and points are identified finally. The key points are calculated by the Harris point detection in the image. The invariant transform depends on the appearance of the object in a specified point of interest and is invariant to the scaling images and rotation where the invariant features denote the detection of variations among different emotions. The algorithm for edge-based invariant transform is given as follows.

---

### **Algorithm I: Edge-based Invariant Transform**

---

**Input:** Extracted Face Parts  $PF_N$

**Output:** Features  $Fea$

---

**Procedure:**

*Step1: Predict the edge region by applying the 'Canny' Edge detector,*

$$Edge_{img} = edge(PF_N, 'canny');$$

*Step2: Estimate the filtered coefficients of the face parts  $PF_N$*

*// N – Number of Face Parts.*

$$H(x, y) = \frac{1}{2\pi\rho^2} \cdot e^{-\frac{(x^2+y^2)}{2\rho^2}}$$

*where (x,y) - Indicates the index of the filtered coefficients such as row and column.*

*$\rho$  – Standard Derivation (constant)*

*Step3: Corners prediction with filtered coefficients,*

*Apply filtering for each face parts,*

$$A = \text{filter}(PF_N, H)$$

$$C = A * A;$$

$$Unwanted_A^{idx} = \max\left(0, H\left(\frac{size-1}{2} - 1\right)\right)$$

$$Unwanted_C^{idx} = \max\left(0, H\left(\frac{size-1}{2} - 1\right)\right)$$

*where,*

*size – number of the filter coefficient H.*

*$Unwanted_A^{idx}$  – Dissimilarity pixel of the face parts*

**Graph based Feature Extraction and Hybrid Classification Approach for Facial Expression Recognition**

$Unwanted_c^{idx}$  – Dissimilarity pixel of the face parts

$$Update_A = A - A(Unwanted_A^{idx})$$

$$Update_C = C - C(Unwanted_c^{idx})$$

$$Corner = (Update_A * A) - (Update_C)^2 - (k * (Update_A + A))^2$$

where,  $k = 0.04$  // kernel constant

Step4: Here invariant features are estimated with corner detection,

Rectangular Regions are detected from the  $Edge_{img}$

$$Reg = \begin{cases} [i \ j] & Edge_{img}(i, j) == 1 \\ 0 & else \end{cases}$$

$$Rowmax = \max(Reg(:, n))$$

$$Rowmin = \min(Reg(:, n))$$

$$Colmax = \max(Reg(m, :))$$

$$Colmin = \min(Reg(m, :))$$

where,

$Rowmax, Rowmin, Colmax, Colmin$  denotes the row and column value for minimum and maximum index

$n$  &  $m$  are the each row and each column of the entire image

$$Binar_{img} = Edge_{img}(Rowmax: Rowmin, Colmax: Colmin)$$

$$Area = (Rowmax - Rowmin) * (Colmax - Colmin);$$

$$Pixel = Corner(Binar_{img});$$

$$pts = harris(Corner) \quad // \text{ Detect harris features}$$

Step5: Final Features  $Fea$

$$Fea = [Area \ Pixel \ pts]$$



### 3.2 Visibility graph construction

The dimensions of extracted edge-based invariant features are improved using weighted visibility graph. The visibility graph has the property for characterizing the edge-based invariant features into graph-based features since it inherits the dynamical properties of the feature data where it was created.

The resultant network is then utilized for acquiring valuable information about the features (i.e.) from that graph that can acquire the graph-based features like link density, an average of closeness centrality, graph entropy, average distribution weight of the graph and average degree of a graph. Here, the weighted visibility graph is constructed for edge-based invariant features by considering a graph as,

$$G = (N_{ij}, E_{ij}) \quad (1)$$

where  $N_{ij}$  represents the feature data which is considered as nodes and  $E_{ij}$  represents the edges formed between the features data. The natural visibility graph algorithm is utilized for finding the links amongst the various nodes of the constructed weighted visibility graph (Zhu et al. 2015). This visibility graph depends on the idea of Euclidean plane in which every vertex denotes the location of the point and the associations among the nodes is possible only if visibility is amongst them which is depicted in Fig. 3.

The edges and the weights amongst any pair of nodes exist if they satisfy the condition depending on the visibility graph and this can be represented as given in Eq. (2),

$$p(t_z) < p(t_x) + (p(t_y) - p(t_x)) \frac{t_z - t_x}{t_y - t_x}, x < z < y \quad (2)$$

where,  $p_x = p(t_x)$  and  $p_y = p(t_y)$  are the sample data points.

$x < z < y$  denotes the index of feature values.

Here  $p(t_x)$  denotes the features points at the  $t_x$  location,  $p(t_y)$  denotes the feature points at the  $t_y$  location,  $p(t_z)$  denotes the feature values at the  $t_z$  location,  $t_x$  and  $t_y$  denotes time events,  $t_z$  is some event that exists between them, i.e.,  $t_x < t_z < t_y$ .

After this, the edge weight for the links that are formed between two nodes is determined. The preservation of the weight information helps to obtain the robust result of a complex network. This weighted complex network plays an important role in distinguishing the weak and potentially less significant edges. Here the edge invariant data can be defined by producing the weighted visibility graph. This constructed graph is denoted as  $G(V_{ij}, E_{ij}, w_{ij})$  where  $w_{ij} : E_{ij} \rightarrow R$  weighted function. Naturally, every edge of the graph are directional as a connection amongst the node  $n_x = p_x$  and node  $n_y = p_y$  is considered to have direction from  $n_x$  to  $n_y$  where  $x < y$ . Here, let us consider the absolute value of the edge weight, and it can be calculated as,

$$w = \arctan \frac{\text{eig}(p(t_y) - p(t_x))}{t_y - t_x}, x < y \quad (3)$$

where  $w$  denotes the weight of the edge amongst the nodes  $n_x$  and  $n_y$ . Here every edge weight value is considered in the radian function, and the eigenvalue transforms the weight in the linear form. The arctan is referred to an inverse trigonometric function which supports to recognize the variations in the data. Eigenvalues are the special set of values associated in a linear system where the eigenvectors and eigenvalues are used to transform the matrix given. Here, eigenvalues are used for weight updation. Next, the features from the graph such as link density, average closeness centrality, graph entropy, average distribution weight and average degree of the graph are extracted. The Algorithm II for graph-based feature extraction is illustrated as follows.



---

**Algorithm II: Feature extraction using Eigen value-based weighted visibility graph**


---

**Input:** Edge-based invariant features

**Output:** Graph-based features

---

**Procedure:**

Step1: Estimate the weight for the formed edges using equation (3)

Step2: Link Density in Graph,

Edges = size( $E_{ij}$ ) // Number of formed edges,

vertices = size( $V_{ij}$ ) // Number of formed vertices,

$$Dense_{graph} = \frac{2 * Edges}{vertices (vertices - 1)}$$

Step3: Average of Closeness Centrality

For  $a=1$ : size( $V_{ij}$ )

For  $b=1$ : size( $V_{ij}$ )

$$Dist(a,b) = \lim_{b \rightarrow 1: size(V_{ij,1})} \sqrt{(X(V_{ij}(a)) - X(V_{ij}(b)))^2}$$

End

$$D(a) = \lim_{a \rightarrow 1: size(V_{ij,1})} \sum_{b=1}^{size(V_{ij,1})} Dist(a,b);$$

$$Closeness_{central}(a) = \frac{size(E_{ij})}{D(a)}$$

End

Step4: Graph Entropy

$$Entropy = - \sum_{i=1}^n P(i) * \log(P(i))$$

where 'i' denotes each feature points

'P' - Degree Distribution of graph,

$$P = \frac{n_k}{n}$$

//  $n_k$  – number of k connected nodes to the particular  $n^{th}$  node.

// n – Number of vertices

Step 5: Average Distribution Weight of the Graph

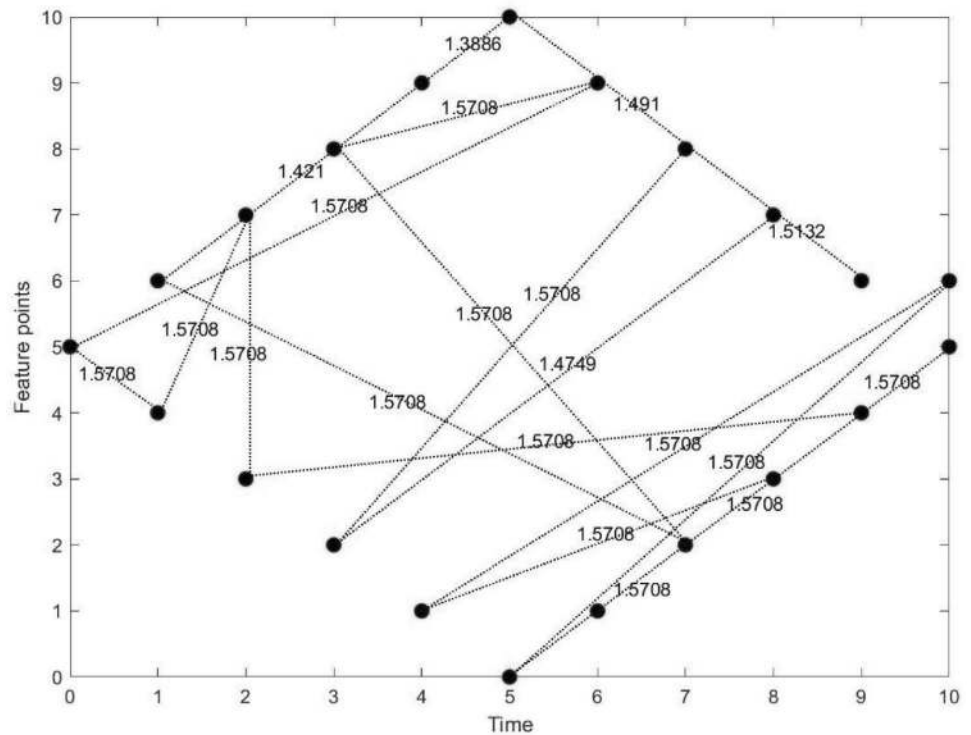
$$Weight_{avg} = \text{mean}(w)$$

Step 6: Average degree of a graph

$$deg = 2 * size(E_{i,j})$$

$$Graph_{fea} = [Dense_{graph} \ Closeness_{central} \ Entropy \ Weight_{avg} \ deg]$$


---

**Fig. 3** Visibility graph

### 3.3 Shape-based feature extraction

The graph-based features extracted are discussed in the previous sub-section. Instantly, the shape-based features are extracted using the shape appearance of the facial parts. The shape-based feature extraction is the process of extracting the facial parts like the right eye, left eye, nose and mouth.

The shape-based extraction mainly focuses on the height and width of each facial parts. Initially, the height and width of each facial parts are calculated as per the total number of face parts in the facial image, and hence the shape of the image is finally extracted. The Algorithm III for this shape-based feature extraction method is shown below.

---

### **Algorithm III: Shape-based Feature Extraction**

---

**Input:** Extracted Face Parts  $PF_N$

**Output:** Shape Features  $shape_{fea}$

---

**Procedure:**

Step 1: Calculate the height and width of each face parts,

$$Height_i = size(PF_i, 1) \forall i \rightarrow 1 \text{ to } N$$

$$Width_i = size(PF_i, 2) \forall i \rightarrow 1 \text{ to } N$$

where  $N$ - number of face parts which is 4

Step2:

$$shape_{fea} = [Height_1 \ Width_1 \ \dots \ \dots \ \dots \ Height_N \ Width_n]$$


---

### **3.4 Self-organizing map-based neural network classifier**

After the extraction of features, the features are then classified with the use of Self-organizing Map-based Neural Network Classifier. The neural network system is generally composed of highly interconnected components that process the information by their dynamic state responses to the external inputs. It is most widely utilized in the application of pattern recognition because they have the capability of generalizing as well as responding to unexpected patterns of inputs. The Self-Organizing Map is a structure of feed-forward with a single computational layer of neurons that are arranged in rows and columns. At first, in the training phase, the mean threshold for each feature of the class is estimated. Then, the features that are belonging to the same

class are processed. The pairwise distance will be computed for each feature set. After that, the density for that specified class depending on the distance among the features will be evaluated. Depending on the estimated distance, the rank for each class will be identified. For that updated rank, the gradient is being identified. Based on that gradient, the centroid computations are made. Then, in the testing phase, the test features are processed by the same operation of the threshold, pairwise distance and centroid computation. Based on the centroid, the positions are updated. The next phase is the validation phase in which the updated features in the training phase, as well as the testing phase, are applied. From this, the expected class is predicted corresponding to the test features by the trained network. The Algorithm IV for classification is given as follows.

---

**Algorithm IV: Self-organizing Map-based Neural Network Classifier**


---

**Input:** Extracted Train Features  $X_{tr}$ , Test Features  $X_{te}$  and Label  $L$

**Output:** Predicted Class  $C$

---

**Procedure:**

**% Training Phase**

Estimate the mean threshold for every class of features,

Step 1: Extract the features which belong to the same class

$$R = X_{tr}(L == N) \quad // N- \text{ number of classes, } X_{tr}\text{-training features}$$

Step 2: The mean threshold is

$$\delta = \text{avg}(\sum(R)) - \sum(\text{avg}(R))^2$$

Step 3: Calculate pairwise distance for the set of features,

$$\text{dist} = \text{pdist}(R, 'euclidean')$$

Step 4: Estimate the density for the particular class based on the distance among the features,

$$\text{density} = \frac{\text{histc}(R)}{\text{sum}(\text{dist})}$$

Step 5: Rank the classes based on the estimated distance

$$\text{Rank} = \min(\text{dist}(\max(\text{density})))$$

Step 6: Estimate the gradient for updated rank,

$$\text{Gradient} = \text{density}(\text{Rank})$$

Step 7: Centroid calculation for each class of every set of features,

$$\text{Cent} = \frac{\text{size}(\text{Gradient} == N)}{1 + \sum(\text{mean}(R(\text{Gradient}, :) - \text{mean}(R))) / \delta}$$

**% Testing Phase**

Step 8: Extract the features which belong to the same class

$$R = X_{te}(L == N) \quad // N- \text{ number of classes, } X_{te}\text{-testing features}$$

Step 9: The mean threshold is

$$\delta = \text{avg}(\sum(R)) - \sum(\text{avg}(R))^2$$

Step 10: Calculate the pairwise distance for the set of test features and estimated centroid,

$$\text{dist} = \text{pdist}(R, 'euclidean')$$

$$\text{dist2} = \text{pdist}(\text{Cent}, 'euclidean')$$

Step 11: Update the centroid based on the position,

$$\text{pos} = \min(\text{dist}, \text{dist2})$$

$$\text{Cent} = R(\text{pos}, :)$$

**% Validation Phase**

Step 12: Feed the neural network with updated features which are updated at the testing phase,

$$\text{net} = \text{feedforward}(\text{cent}, L)$$

Step 13: Predict the class corresponding test features by the trained network,  $C = \text{net}(X_{te})$

---

### 4 Performance analysis

The implementation outcomes of the GFE-HCA system is presented using numerous measures like sensitivity, specificity, accuracy, precision and recall. The proposed method is evaluated using MMI dataset (MMI Facial Expression Database 2016), which consist of 2900 videos containing high-resolution images of 75 subjects with different facial expressions of various people. The issues for analyzing the automatic human behavior are solved using a dataset which is developed here.

From MMI dataset, 175 facial expression sequences layering emotions of 33 subjects were selected. Each sequence is labelled with the following emotions, namely anger, disgust, fear, sad and happy. Furthermore, the improvement of the proposed method is presented by relating it with the existing emotion recognition technique is considered in this paper (Ghimire et al. 2017a, b).

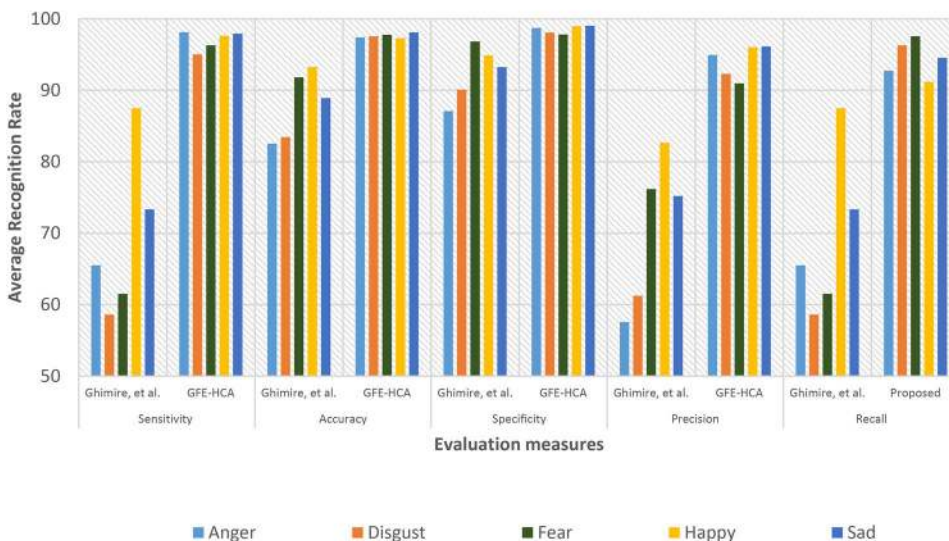
The proposed work is implemented using MATLAB version R2017b, Intel core i5-4200U system configured with 8 GB RAM and 2.5 GHz clock speed. The choice for this tool is due to the wide array of functional libraries.

**Table 1** Confusion matrix of the proposed method

	Anger	Disgust	Fear	Happy	Sad
Anger	<b>96.5</b>	1.3	1.1	0.3	0.8
Disgust	0.1	<b>96.3</b>	0.4	1.6	1.7
Fear	0.3	1.2	<b>97.5</b>	0.9	0.1
Happy	2.0	0.3	0.4	<b>96.5</b>	0.8
Sad	1.1	1.0	0.6	0.7	<b>96.6</b>

Bold values in diagonal signifies values of accurate predictions

**Fig. 4** Comparative measures of the existing and proposed method



### 4.1 Overall performance evaluation

The measures like sensitivity and specificity are utilized for evaluating the performance of classification. Sensitivity defines the capability of the classifier to correctly recognize which are given in the following equations,

$$Sensitivity = \frac{TP}{TP + FN} \tag{4}$$

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

True Positive (TP) is the number of positive cases (emotions) that are detected correctly.

True Negative (TN) is the number of negative cases (emotions) that are not detected.

False Positive (FP) is the number of positive cases which are falsely detected.

False Negative (FN) is the number of negative cases which are rejected.

Depending upon these calculations, the accurateness of the emotion recognition system is deliberated. Accuracy is defined as the closeness of a measured value to the standard values, and it is calculated as follows:

$$Accuracy = \frac{TN + TP}{(TN + TP + FN + FP)} \tag{6}$$

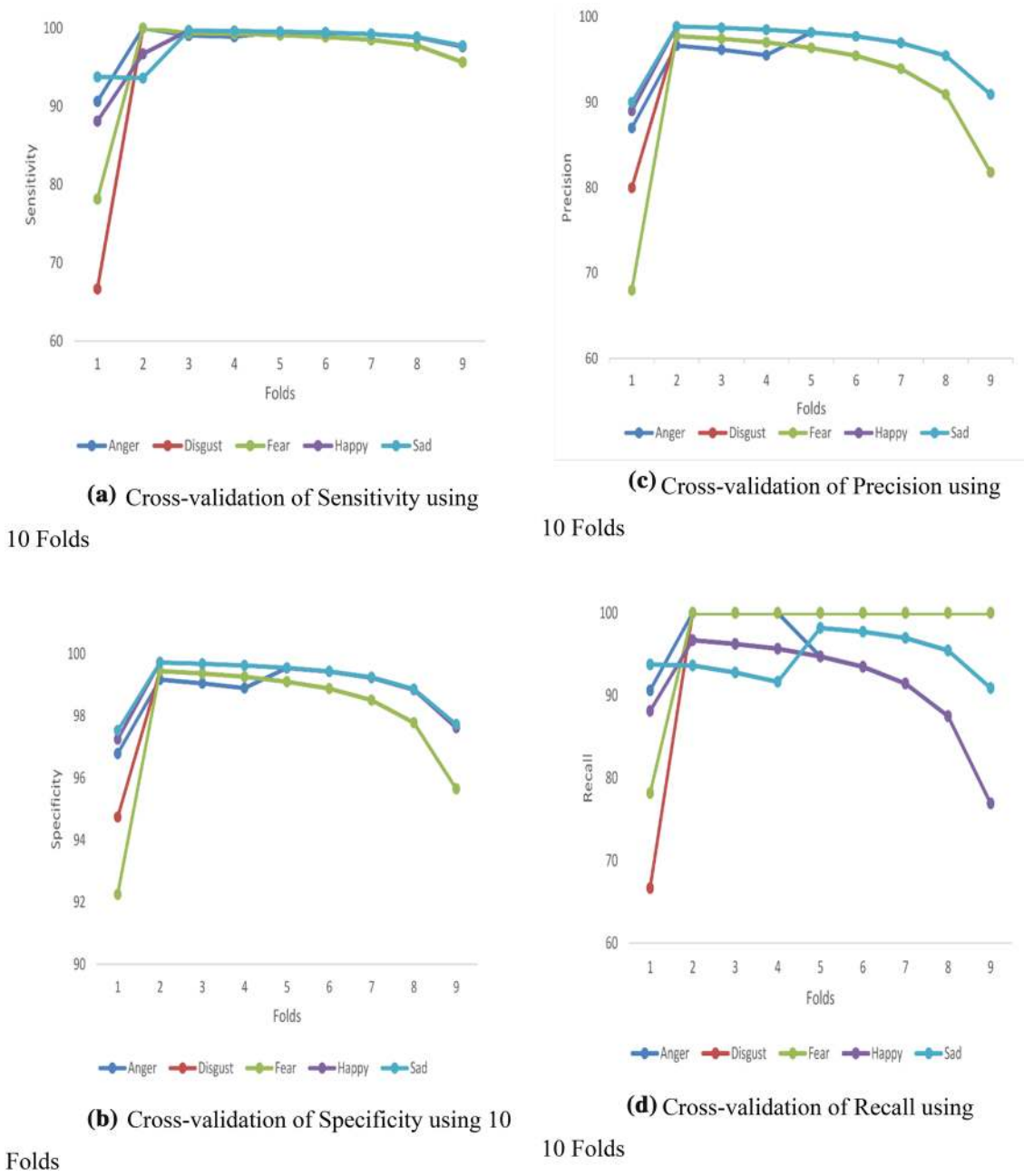
Precision is defined as the amount of the true positive outcomes beside all positive effects. This can be evaluated using the following equation as,

$$Precision = \frac{TP}{(TP + FP)} \tag{7}$$

### 4.2 Performance analysis

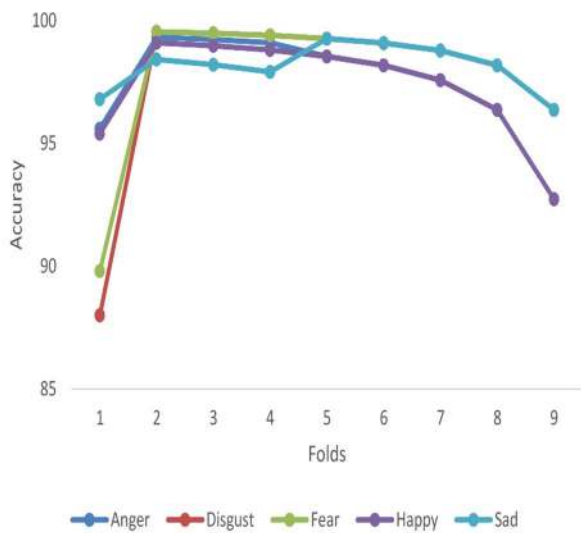
The execution of the proposed methodology is evaluated using MMI dataset. The performance metrics like sensitivity, specificity, precision, recall and accuracy are utilized for assessing the performance of the proposed methodology. The confusion matrix of the proposed system is considered for five emotions is enumerated in Table 1. The proposed approach offers improved results than the existing approach, is shown in Fig. 4.

Cross-validation technique considered for performance analysis makes sure that overfitting does not happen. K-fold cross-validation is employed where the data set is partitioned into K equal parts. During the first run, the first part is kept untouched, and the remaining K – 1 parts are used to perform model training. Testing is done with the part that was kept untouched, similarly for all the K times, one partition is kept untouched, and other parts are used for training. After the training, the untouched part is used for evaluation. This type of validation makes sure the entire data set is equally



**Fig. 5** a Cross-validation of Sensitivity using 10 Folds. b Cross-validation of Specificity using 10 Folds. c Cross-validation of Precision using 10 Folds. d Cross-validation of Recall using 10 Folds. e Cross-validation of Accuracy using 10 Folds





(e) Cross-validation of Accuracy using 10 Folds

Folds

Fig. 5 (continued)

validated. The average accuracy of our proposed work shows above 96%.

From Fig. 4, the sensitivity comparative results for the proposed approach offers increased sensitivity results to the ‘anger’, ‘happy’ and ‘sad’ emotions whereas, it offers less sensitivity to the ‘disgust’ emotion. Likewise, the proposed approach offers increased accuracy results for the emotions ‘fear’ and ‘sad’ with an overall accuracy above 94%. Similarly, the proposed method offers increased specificity results for the emotions ‘anger’, ‘happy’, and ‘sad’ emotion, whereas it gives less specificity results to ‘fear’ emotion. From the results, it is perceived that the proposed method offers increased precision results for the emotions ‘happy’ and ‘sad’ whereas it gives less precision results to ‘fear’ emotion.

The recognition rate of the proposed technique is calculated using various performance measures separately for the listed five emotions like anger, disgust, fear, sad, happy using tenfold cross-validation. The performance of the proposed method has been analyzed with sensitivity, specificity, precision, recall, accuracy using tenfold cross validation as presented in Fig. 5a–e. From the assessment, it is apparent that the proposed technique offers a better recognition rate when measured with the existing methods.

MMI dataset is complex when compared with other datasets, as per the base paper (Ghimire et al. 2017a, b) and the author applauded MMI dataset as challenging when compared with CK + and MUG datasets. Here the performance is validated and compared with the existing (Ghimire et al.

Table 2 Confusion matrix of the existing work (Khan et al. 2018)

	Anger	Disgust	Fear	Happy	Sad
Anger	<b>97.7</b>	2.3	0	0	0
Disgust	0	<b>87.17</b>	5.1	2.5	5.1
Fear	0	0	<b>92.6</b>	7.3	0
Happy	0	0	0	<b>100</b>	0
Sad	0	0	2.9	0	<b>97.05</b>

Bold values in diagonal signifies values of accurate predictions

Table 3 Confusion matrix of the existing work (Gogić et al. 2018) in PI scenario

True label	Anger	Disgust	Fear	Happy	Sad
Anger	90.91	0.00	0.00	0.00	9.09
Disgust	3.12	77.08	1.04	17.71	0.00
Fear	13.10	0.00	33.33	3.57	46.43
Happy	0.00	3.17	0.00	96.03	0.79
Sad	18.75	3.12	0.00	0.00	78.12
Surprise	3.25	0.00	0.81	0.00	1.63
Neutral					94.31

2017a, b; Khan et al. 2018; Gogić et al. 2018) methods. Weber Local Binary Image Cosine Transform (WLBI-CT) (Khan et al. 2018) extracts and integrates the frequency components of images obtained through Weber local descriptor and local binary descriptor. The confusion matrix of the WLBI-CT method is given in Table 2.

As in Gogić et al. (2018) LBF-NN uses a combination of gentle boost decision tree and neural network on the distinct facial landmarks optimized across all expressions under study. These combined feature optimization improves the rate of recognition in complex expressions.

The confusion matrix of the LBF-NN method with Person Independent (PI) scenario is given in Table 3. However, while considering the Person Dependent (PD) scenario, this method yields far better results compared to the methods discussed in this paper. Comparison of the proposed method with the existing methods (Ghimire et al. 2017a, b; Khan et al. 2018; Gogić et al. 2018) is depicted in Table 4.

### 5 Conclusion

This work aims to recognize human emotion in an effective manner by proposing a novel method. This paper introduces a GFE-HCA: Graph-based Feature Extraction and Hybrid Classification Approach for recognizing the facial expression from the given input image. Initially, the facial regions are detected in the foreground detection step, and then the

**Table 4** Average recognition accuracy comparison of proposed work with (Khan et al. 2018; Gogić et al. 2018)

Works compared	Average accuracy rate (%)
Proposed work with MMI and 5 labels	96.47
Ghimire et al. (2017a, b) with MMI dataset	77.2
WLBI-CT (Khan et al. 2018) work with MMI and 5 labels	94.44
LBF-NN (Gogić et al. 2018) in PI with MMI dataset	75.09

facial parts are extracted. The features from that detected facial parts are extracted using edge-based invariant feature. Then, the weighted visibility graph is constructed for optimizing the dimensions of the edge-based invariant features. Then the features from a graph such as link density, average closeness centrality, graph entropy, average distribution weight and average degree of the graph are extracted. Also, the shape appearance-based features from the facial parts is extracted. Finally, the Self-Organizing Map based Neural Network Classifier is introduced to recognize and classify the facial expressions. This approach is compared with the existing techniques and verified with several performance metrics such as sensitivity, specificity, precision, recall and accuracy. The evaluation results concluded that the proposed approach offers better results with increased emotion recognition rate when compared to the various existing works.

## 6 Future works

Recently Online learning platform, IoT and Autonomous vehicles are trending among the researchers. Future work concentrates on a few of the following.

- In our work, the facial image and the camera plane are in parallel, recent advancement in online learning also ventures that the work needs to be improved for at least three different angles 15°, 45° and 60°.
- Improving the model to adapt and give accurate results on resource constraint devices as in most cases of IoT devices
- Image stabilization is not guaranteed while in autonomous moving vehicles, future work should concentrate on finding a mechanism to give an accurate result.

Our future goal on the same research work is to add a feedback continuous training module that can cater to continuous training and layer addition to adapt upcoming time on time changes. One such example is how to use the same model and add a layer to give a good result on facial emotion during CoVID'19 mask applied facial images.

## References

- Fang H, Mac Parthaláin N, Aubrey AJ et al (2014) Facial expression recognition in dynamic sequences: an integrated approach. *Pattern Recogn* 47:1271–1281
- Fuentes C, Herskovic V, Rodríguez I et al (2017) A systematic literature review about technologies for self-reporting emotional information. *J Ambient Intell Human Comput* 8:593–606
- Ghimire D, Jeong S, Lee J, Park SH (2017a) Facial expression recognition based on local region specific features and support vector machines. *Multimed Tools Appl* 76:7803–7821
- Ghimire D, Lee J, Li Z-N, Jeong S (2017b) Recognition of facial expressions based on salient geometric features and support vector machines. *Multimed Tools Appl* 76:7921–7946
- Gogić I, Manhart M, Pandzic IS et al (2018) Fast facial expression recognition using local binary features and shallow neural networks. *Vis Comput*. <https://doi.org/10.1007/s00371-018-1585-8>
- Goyani M, Patel N (2017) Multi-level haar wavelet based facial expression recognition using logistic regression. *Indian J Sci Technol* 10:1–9
- Khan SA, Hussain A, Usman M (2018) Reliable facial expression recognition for multi-scale images using weber local binary image based cosine transform features. *Multimed Tools Appl* 77:1133–1165
- Kim H, Ben-Othman J, Cho S, Mokdad L (2019) A Framework for IoT-enabled virtual emotion detection in advanced smart cities. *IEEE Netw* 33:142–148
- Jensen OH (2008) Implementing the Viola-Jones face detection algorithm. MS thesis. Technical University of Denmark, DTU, DK-2800 Kgs., Lyngby, Denmark
- Liu Y, Li Y, Ma X, Song R (2017) Facial expression recognition with fusion features extracted from salient facial areas. *Sensors* 17(4):712
- Lopes AT, de Aguiar E, De Souza AF, Oliveira-Santos T (2017) Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recogn* 61:610–628
- Lumini A, Nanni L, Brahnam S (2017) Ensemble of texture descriptors and classifiers for face recognition. *Appl Comput Inf* 13:79–91
- Mistry K, Zhang L, Neoh SC et al (2016) A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition. *IEEE Trans Cybern* 47:1496–1509
- MMI Facial Expression Database (2016) <https://www.mmifacedb.eu>
- Mollahosseini A, Chan D, Mahoor MH (2016) Going deeper in facial expression recognition using deep neural networks. In: 2016 IEEE Winter conference on applications of computer vision (WACV). IEEE, pp 1–10
- Nanni L, Lumini A, Brahnam S (2017) Ensemble of texture descriptors for face recognition obtained by varying feature transforms and preprocessing approaches. *Appl Soft Comput* 61:8–16
- Peng X, Xia Z, Li L, Feng X (2016) Towards facial expression recognition in the wild: a new database and deep recognition system.

- In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 93–99
- Qayyum H, Majid M, Anwar SM, Khan B (2017) Facial expression recognition using stationary wavelet transform features. *Math Probl Eng* 2017:1–9. <https://doi.org/10.1155/2017/9854050>
- Ramadass M, Nanthini T, Prasanna Devi S (2018) Local directional ternary pattern for facial expression recognition. *J Netw Commun Emerg Technol (JNCET)* 8:320–324
- Ratliff MS, Patterson E (2008) Emotion recognition using facial expressions with active appearance models. In: Proc. of HRI. Citeseer
- Samara A, Galway L, Bond R, Wang H (2019) Affective state detection via facial expression analysis within a human–computer interaction context. *J Ambient Intell Human Comput* 10:2175–2184
- Sariyanidi E, Gunes H, Cavallaro A (2017) Learning bases of activity for facial expression recognition. *IEEE Trans Image Process* 26:1965–1978
- Suk M, Prabhakaran B (2014) Real-time mobile facial expression recognition system—a case study. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 132–137
- Valstar MF, Sánchez-Lozano E, Cohn JF, et al. (2017) Fera 2017—addressing head pose in the third facial expression recognition and analysis challenge. In: 2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017). IEEE, pp 839–847
- Xie S, Hu H (2017) Facial expression recognition with FRR-CNN. *Electron Lett* 53:235–237
- Xie W, Shen L, Yang M, Lai Z (2017) Active AU based patch weighting for facial expression recognition. *Sensors* 17:275
- Xu Y, Zhang Z, Lu G, Yang J (2016) Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification. *Pattern Recogn* 54:68–82
- Yu Z, Zhang C (2015) Image based static facial expression recognition with multiple deep network learning. In: Proceedings of the 2015 ACM on international conference on multimodal interaction, pp 435–442
- Zhong L, Liu Q, Yang P et al (2014) Learning multiscale active facial patches for expression analysis. *IEEE Trans Cybern* 45:1499–1510
- Zhu X, Lei Z, Yan J, et al (2015) High-fidelity pose and expression normalization for face recognition in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 787–796

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