Contents lists available at ScienceDirect

Defence Technology

journal homepage: www.keaipublishing.com/en/journals/defence-technology

A novel facial emotion recognition scheme based on graph mining

Alia K. Hassan [a], Suhaila N. Mohammed [a, b, *]

[a] Department of Computer Science, University of Technology, Baghdad, Iraq
[b] Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

ARTICLE INFO

Article history:
Received 19 July 2019
Received in revised form
11 September 2019
Accepted 24 December 2019
Available online 30 December 2019

Keywords:
Emotion recognition
Facial landmarks
Graph mining
gSpan algorithm
Binary cat swarm optimization (BCSO)
Neural network

ABSTRACT

Recent years have seen an explosion in graph data from a variety of scientific, social and technological fields. From these fields, emotion recognition is an interesting research area because it finds many applications in real life such as in effective social robotics to increase the interactivity of the robot with human, driver safety during driving, pain monitoring during surgery etc. A novel facial emotion recognition based on graph mining has been proposed in this paper to make a paradigm shift in the way of representing the face region, where the face region is represented as a graph of nodes and edges and the gSpan frequent sub-graphs mining algorithm is used to find the frequent sub-structures in the graph database of each emotion. To reduce the number of generated sub-graphs, overlap ratio metric is utilized for this purpose. After encoding the final selected sub-graphs, binary classification is then applied to classify the emotion of the queried input facial image using six levels of classification. Binary cat swarm intelligence is applied within each level of classification to select proper sub-graphs that give the highest accuracy in that level. Different experiments have been conducted using Surrey Audio-Visual Expressed Emotion (SAVEE) database and the final system accuracy was 90.00%. The results show significant accuracy improvements (about 2%) by the proposed system in comparison to current published works in SAVEE database.

1. Introduction

The notion of emotion recognition is an active research topic in pattern recognition and artificial intelligence fields due to its diverse applications in different areas like smart systems and human robot interaction [1]. In general, emotion recognition procedure consists of two steps which involve significant features extraction and classification. Feature extraction defines a set of independent attributes which can describe the emotion of facial expression. While in classification, the features are categorized into one of various emotion classes like happy, sad, disgust, anger, surprise, etc. [2]. Feature selection technique and classifier design are one of the main stepping stone to a robust emotion recognition system. A well-developed emotion classification algorithm sometimes cannot produce high level of accuracy due to worthless selected features [3].

Distinctive methods have been constantly developed to extract features for emotion recognition task like geometric based and appearance based methods. Geometric features like facial landmarks and the locations of the decisive parts of a face, are important for understanding the affective conveyed by facial expressions [4]. Recently, several emotion recognition systems were proposed based on facial landmarks to extract distinguishable features. Mohseni et al. [5] developed a method for automatic facial expression recognition based on plotting a face model graph on the facial image using facial landmarks points and extracting features by measuring the ratio of the facial graph sides. The authors used Support Vector Machine (SVM) for classification and achieved accuracy of about 87.7% on 280 images from MMI database. Haq et al. [6] employed Facial Marker Position (MP) and Marker Angle (MA) as features and SVM as a classifier. The achieved result was 88% on Surrey Audio-Visual Expressed Emotion (SAVEE) database. Mohammed et al. [7] used landmarks points to generate distance features which are in turn clustered using modified k-Means algorithm. Neural Network (NN) was then employed for classification purpose. The authors achieved accuracy of 90.26% using 870 images

* Corresponding author. Department of Computer Science, University of Technology, Baghdad, Iraq.
E-mail addresses: 110018@uotechnology.edu.iq, suhailan.mo@sc.uobaghdad.edu.iq (A.K. Hassan), suhailan.mo@sc.uobaghdad.edu.iq (S.N. Mohammed).

Peer review under responsibility of China Ordnance Society

https://doi.org/10.1016/j.dt.2019.12.006

from Cohen-Kanade (CK) dataset. Munasinghe [8] used facial landmarks to extract distance features from points which are nearest form eyes and nose. The same feature vector is calculated for the neutral emotion and the difference between emotional vector and natural one is considered as final features to identify emotions using random forest classifier. Testing of the method was carried out using CK + database and average 90% success rate was achieved for four emotions only.

On the other hand, the use of graph representations has recently gained popularity in pattern recognition and machine learning fields. The main advantage of graphs is that they offer a powerful way to represent structured data [9]. Although different works used Elastic Bunch Graph Matching (EBGM) algorithm like [10,11] for face recognition purpose, but in these works, the entire graph is utilized for recognition purpose and the authors didn't attempt to mine the facial graphs by finding substructures that may change during some emotion expression. Thus, a large number of features are generated with the previously proposed works and many of these features may be irrelevant. By taking graph mining as a starting point, a novel method is proposed in this paper for facial emotion detection by utilizing the graph mining techniques to reduce the size of the final extracted features and improve accuracy. The rest of the paper is organized as follows: the theoretical background of graph mining concepts and cat swarm intelligence are reviewed in Section 2. The proposed methodology is discussed in Section 3. The simulation results and experiments are demonstrated in Section 4. Finally, the work conclusions and future works are shown in Section 5.

2. Theoretical background

2.1. Graph mining concepts

Graphs become increasingly important in representing complicated structures, such as images, chemical compounds, biological networks, the Web, and XML documents [9]. Many important properties of graphs can be employed in the representation of such domains. Graph mining algorithms can, therefore, be leveraged for analyzing different properties of graphs [12].

A graph is a theoretical construct composed of a set of points (called vertices) connected by lines (called edges). The vertices of a graph denote discrete pieces of information, while the edges of a graph denote the relationships between those pieces [13]. A graph database D is a collection of n different graphs, $G_1 = (N_1, A_1) \ldots G_n = (N_n, A_n)$, such that the set of nodes in the ith graph is denoted by N_i, and the set of edges in the ith graph is denoted by A_i. Each node $p \in N_i$ is associated with a label denoted by $l(p)$ [14].

2.1.1. Matching and distance computation

Graph matching is the process of evaluating the structural similarity among graphs [9]. To check whether a match exist between a pair of graphs, a one-to-one correspondence must be established between the nodes of these graphs, such that their labels match, and the edge presence between corresponding nodes also match. In other words, the distance between such two graphs is zero. Therefore, the problem of distance computation between a pair of graphs is at least as hard as that of graph matching. Matching graphs are also said to be isomorphic [14].

2.1.2. Frequent sub-graph mining

Frequent sub-graphs are useful at characterizing, classifying and clustering different set of graphs [9]. Frequent Sub-graph Mining (FSM) is the process of discovering sub-graphs that occur often in a database of graphs. In this context, often means that the graph in question occurs more than some arbitrary threshold (minimum support) number of times. Since FSM does not rely on training data, frequent sub-graph mining is considered a type of unsupervised learning [13].

Graph-based Substructure PAtterN (gSpan) is one of FSM algorithms. gSpan avoids generating the redundant sub-graphs by restricting the number of redundant sub-graph candidates that must be examined. This algorithm is one of the first depth-first FSM algorithms, which can be characterized by the following concepts [13]:

1) gSpan Encoding. A gSpan Encoding is an ordered set of edges, where the order is determined by how the edges are traversed during depth-first traversal. Each edge in a gSpan Encoding is a five-tuple made up of the unique indices of the endpoints (first and second elements), the labels of these endpoints (third and fifth elements, respectively), and the edge label (fourth element).
2) Rightmost expansion. Edges are expanded in gSpan encoding via rightmost expansion such that a new edge may be added only if one of its endpoints lies on the rightmost path. Adding edges in this way ensures the gSpan encoding validity.
3) Lexicographic Order. The gSpan encodings can be given a linear ordering; that is, they can be ordered from smallest to largest. This ordering is based on finding edges that are common between two gSpan encodings. This encoding was introduced to recognize graphs that are isomorphic to each other, so that the algorithm does not waste time doing things it has already done.
4) Minimum gSpan Encoding. The Minimum gSpan Encoding (MgSE) is simply the smallest of all the possible gSpan encodings (the lowest one in the lexicographic ordering). MgSE is the key to candidate pruning in gSpan because the MgSE allows gSpan to recognize when it is about to generate a sub-graph that it has already considered. The use of the MgSE for candidate pruning makes gSpan much more efficient than previous Apriori-based algorithms.

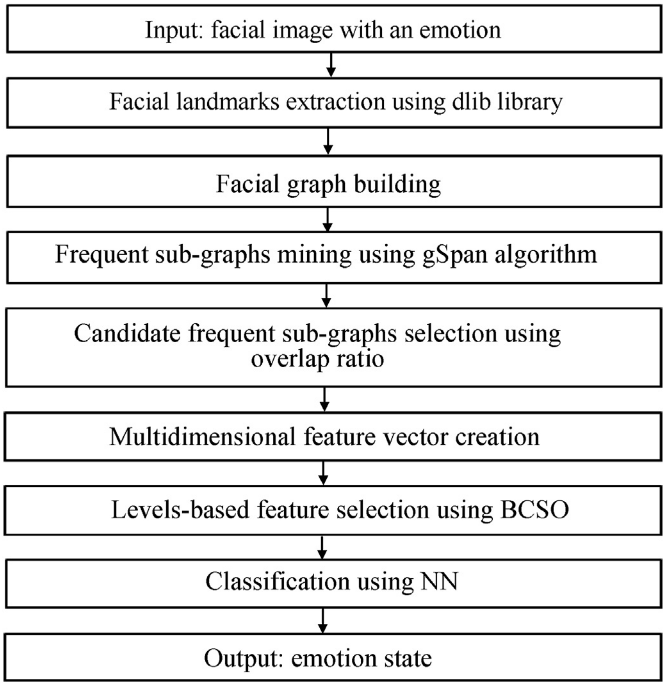
Algorithm 1 illustrates the main steps of gSpan Frequent Subgraph Mining algorithm while Algorithm 2 shows the procedure of gSpan_Expansion part of gSpan algorithm.

```
Algorithm 1 gSpan Frequent Sub-graph Mining [13]
Input     D: Graph database, Minsupp: Minimum support
Output    S: Set of frequent sub-graphs
Step1     Set S←All frequent one edge sub-graphs in D
Step2     Sort S in lexicographic order
Step3     Set N←S
Step4     Foreach n in N do gSpan_Expansion (D, n, Minsupp, S)
          Remove n from D
          Next
          Return S
Algorithm 2 gSpan_Expansion [13]
Input     D: Graph database, n: Sub-graph, Minsupp: Minimum support
Output    S: Set of frequent sub-graphs
Step1     If n doesn't meet condition of MgSE Then Return S
Step2     Add n to S
          Foreach single edge rightmost expansion of n (e) Do
          If (support(e) ≥ Minsupp) Then gSpan_Expansion (D, e, Minsupp)
          Next
```

2.2. Cat swarm optimization

Nowadays, researchers in pattern recognition and machine learning fields are increasingly recognizing the significance of swarm optimization for data dimensionality reduction and their effect in classification process accuracy. Many swarm optimization algorithms have been used for feature selection task, for example, Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). Recently, a considerable number of new swarm-inspired algorithms have emerged like Binary Cat Swarm Optimization (BCSO) algorithm [15].

Chu et al.[16] proposed that cats in nature have two modes of behavior: seeking mode and tracing mode. In seeking mode, cats move near its original position slowly. In the tracing mode, the cats move with theirs own velocities for every dimension. The main advantage BCSO which make it better than other swarm-inspired algorithms, it is greatly outperforms these algorithms in terms of accuracy of the obtained results and speed of convergence. The version of cat swarm which can be used for feature selection purpose is called binary because the initial position of each cat in the population can hold only 0s and 1s values [17] (see Ref. [15] for more details about BCSO).


Input: facial image with an emotion

Facial landmarks extraction using dlib library

Facial graph building

Frequent sub-graphs mining using gSpan algorithm

Candidate frequent sub-graphs selection using overlap ratio

Multidimensional feature vector creation

Levels-based feature selection using BCSO

Classification using NN

Output: emotion state

Fig. 1. The general framework of the proposed system.

3. Proposed methodology

The general framework of the proposed system is shown in Fig. 1. The facial landmarks points are extracted from each input image. Using landmarks points, facial graph is then built and the proper weights are assigned on each edge of the graph. To find the frequent sub-graphs in each emotional class, gSpan algorithm is employed. However; a huge number of frequent sub-graphs will be resulted and only related sub-graphs must be picked out. The candidate related sub-graphs are selected based on their overlapping with other classes. Binary encoding is built to form the final multidimensional feature vector. The classification stage is performed in levels and BCSO is used in the proposed system to select the suitable feature subset for each level of classification.

3.1. Facial landmarks extraction

To build a graph on the facial region, facial landmarks are extracted to localize the location of key regions of the face, such as: eyebrows, eyes, nose, mouth, and jawline. dlib library which was implemented by Kazemi and Sullivan [18], is used to extract landmarks points from the facial image. Kazemi and Sullivan detected facial landmarks with a two-step process:

Step 1: Localize the face in the image (face detection) using OpenCV's built-in Haar cascades.

Step 2: Detect locations of the important facial structures on the face region using ensemble method.

Dlib library can estimate locations of 68 landmark points' coordinates that represent the key facial structures of the face. The indexes of these points can be visualized on Fig. 2(a). However; to avoid building very large graph, only points that may be affected with change during expressing an emotion are used (in total 20 points are used) as shown in Fig. 2(b) in which the selected points are marked with blue circles.

3.2. Facial graph building

To build the facial graph using extracted facial landmarks, point's indices are used as vertices' labels while edges labels are

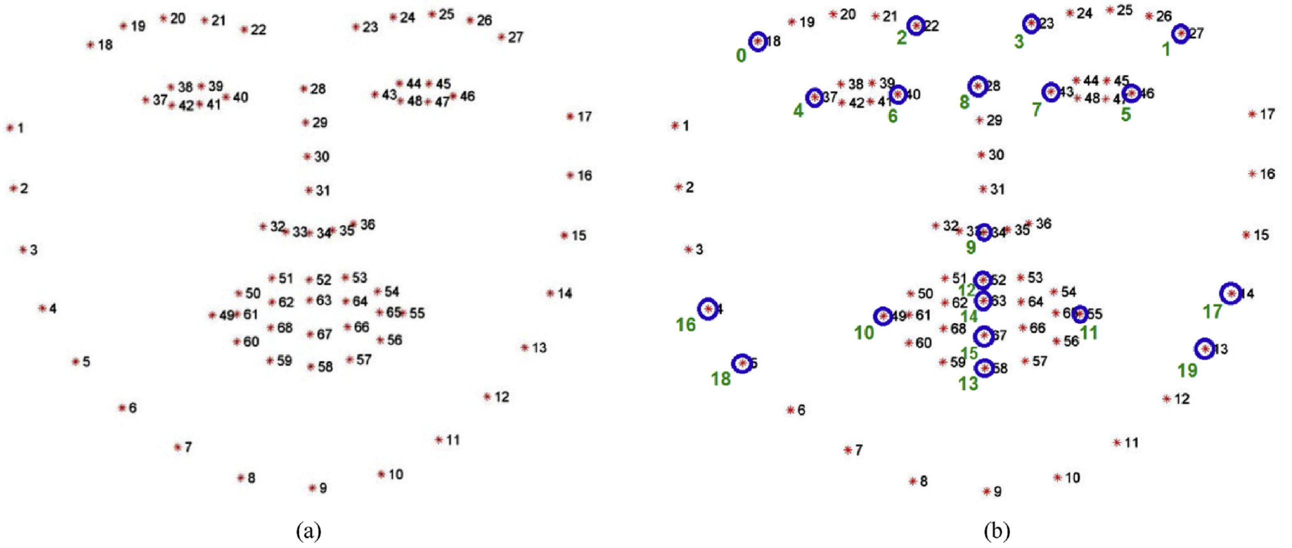


Fig. 2. Facial landmarks' locations, (a) points detected with dlib library, (b) points used in the proposed system.

determined as follows:

Step1: For each pair of points p_1 (x_1, y_1) and p_2 (x_2, y_2), the distance between these points' coordinates are calculated using Euclidian distance as in Eq. (1) [19].

$$D_{p1p2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

Step2: To avoid camera zooming differences and scaling variations among different faces, normalization process is done by dividing each computed distance by the maximum distance Max_D in that face as in Eq. (2). After normalization step, all distances in the face will be within the range [0–1].

$$D^{'}_{p1p2} = \frac{D_{p1p2}}{\mathrm{Max}_D} \tag{2}$$

where D_{p1p2} and $D^{'}_{\mathrm{p1p2}}$ are the distance between p_1 and p_2 before and after normalization respectively, Max_D is the maximum generated distance in the given face region.

Step3: In order to apply the gSpan algorithm, it is necessary to categorize the continues values of the distances in each graph. The range of distances is divided into sub-ranges with length 0.5 and each sub-range is assigned a label from 1 to 20 to cover all possible distance values as shown in Table 1.

After assigning a label to each node and computing the label of each edge, a fully connected undirected graph becomes ready for the successive steps of mining. Fig. 3 shows an example of a graph which is built for a sample face from the used database.

3.3. Frequent sub-graphs generation

To extract discriminable features that represent the common change in the facial graph, the frequent sub-graphs within each class of emotions need to be mined first. Algorithm 1 is employed for this purpose. The algorithm takes the graph dataset of each class and minimum support value as input, while the output is the frequent sub-graphs for this set of graphs. Since the algorithm may produce too many frequent sub-graphs, a parameter (N) is added as a limit for the maximum number of the generated sub-graphs and its value can be determined by experiment.

3.4. Candidate frequent sub-graphs selection

In this stage, only sub-graphs which make small overlap with other emotions will be considered as candidate sub-graphs from each emotion. The candidate set of frequent sub-graphs are selected as follows:

$$\text{Overlap }(i) = \frac{\text{Number of graphs in } D \text{ that contain } \; i \text{ (except graphs that produce } i)}{\text{Total number of graphs in } D} \tag{3}$$

Table 1
Distance sub-ranges and the assigned labels used in the generated facial graph.

Sub-range	Label	Sub-range	Label	Sub-range	Label	Sub-range	Label	Sub-range	Label
[0.0–0.05)	1	[0.05–0.1)	2	[0.1–0.15)	3	[0.15–0.2)	4	[0.2–0.25)	5
[0.25–0.3)	6	[0.3–0.35)	7	[0.35–0.4)	8	[0.4–0.45)	9	[0.45–0.5)	10
[0.5–0.55)	11	[0.55–0.6)	12	[0.6–0.65)	13	[0.65–0.7)	14	[0.7–0.75)	15
[0.75–0.8)	16	[0.8–0.85)	17	[0.85–0.9)	18	[0.9–0.95)	19	[0.95–1]	20

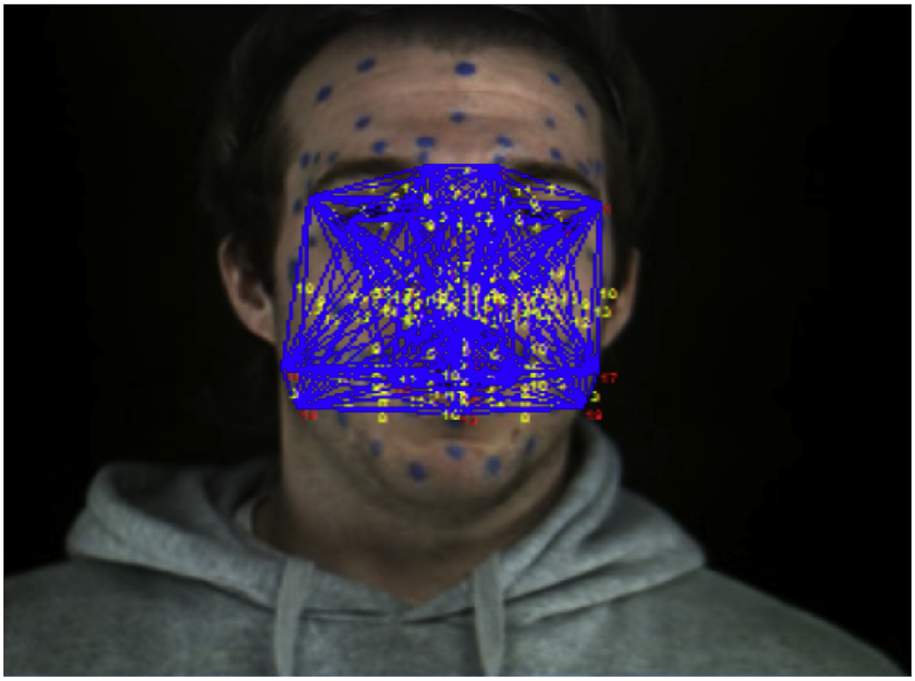


Fig. 3. An example of a graph built for a sample face from the used database.

Step1. The overlap ratio that each sub-graph within each emotion made with other emotion graphs is calculated. The overlap ratio of sub-graph (i) can be calculated as in Eq. (3):

Step2. The sub-graphs are then sorted in ascending order according to their overlap ratio.

Step3. Find the best pair of M sub-graphs that provide smallest overlap ratio.

Step4. The process is continued by finding best three sub-graphs with smallest overlap ratio and best four and so on until the overlap ratio will be zero or its value doesn't improve.

3.5. Multidimensional feature vector creation

For the purpose of being able of applying machine learning and optimization algorithms, the final candidate feature vector of sub-graphs must be converted to multidimensional vector-space representation. Binary encoding is used for this purpose, in which each graph in the database is searched for a match with the given sub-graph. If the match is found, then the sub-graph will be encoded with value 1 in multidimensional vector else it will be represented with 0 as in Eq. (3).

$$f_i = \begin{cases} 1 \text{ if a match is found for sub} - \text{graph } i \\ \quad 0 \quad \text{otherwise} \end{cases} \tag{4}$$

3.6. Levels-based feature selection

Levels-based classification is used in the proposed method such that in each level, only one emotion is classified from the remaining ones. Six levels of classification are used to classify a given facial image to one of the seven basic emotions as shown in Fig. 4.

BCSO with feed forward neural networks as an objective function is utilized to select the proper set of features in each level as shown in Fig. 5. The fitness function used in BCSO is the classification accuracy η which can calculate as [20]:

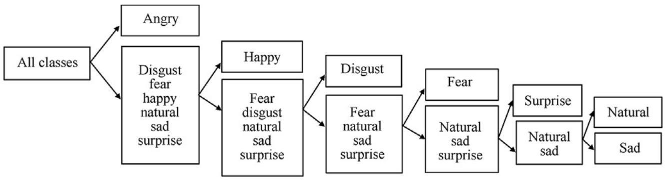


Fig. 4. Levels of binary classification.

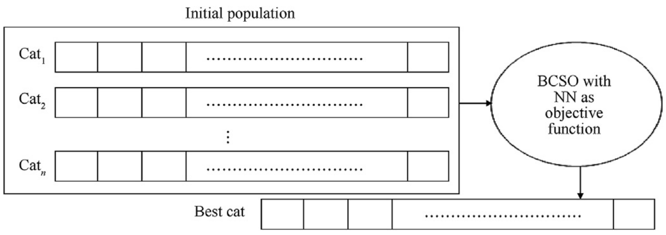


Fig. 5. Feature selection scenario.

$$\eta = \frac{corr}{n} \tag{5}$$

where *corr* gives the number of samples which correctly classified and n is the total number of data samples. To find *corr* value, feed forward neural network are employed; such that in each iteration of BCSO, NN is trained and then tested to compute the fitness value of each cat in the population.

BCSO is employed for feature selection task due to its convergence speed since it works by applying two modes of operations (tracing mode and seeking mode) which make it suitable for feature selection task that proposed in this paper, especially that neural network is used as a fitness function as shown in Fig. 5. Neural networks need training and testing for different architectures to give the best accuracy for a given set of features. This scenario of optimization needs time to coverage and since BCSO offers speed, we utilized it.

The main reason for selecting NN as objective function is the final feature vector may be partially incomplete due to using support value less than 100% in gSpan algorithm which means not all frequent sub-graphs exist in all graphs for a given emotion class. NN is machine learning method which has the ability to deal with partially incomplete data.

3.7. Classification stage

The final stage in the proposed system is the classification stage. Six NNs are built; each NN is trained and tested with the selected features which are picked out with BCSO in each level of classification. The database is divided into training and testing sets, where 80% of the database is used for training the NN and selecting the best configuration to achieve the proper weights in each level of classification.

4. Experimental results

4.1. Database description

The proposed system is trained and tested on Surrey Audio-Visual Expressed Emotion (SAVEE) database. SAVEE includes recordings of seven different emotions for four male actors, totally 480 samples. The sentences are in British English utterances and chosen from the standard corpus of TIMIT. The data were recorded in a visual media lab with high quality audio-visual equipment, processed and labeled. To check the quality of performance, the recordings were evaluated by ten subjects under audio, visual and audio-visual conditions [21].

In the proposed method, each video in the database is framed and the middle frame is used since it holds the apex state of the emotion. Totally 480 frames are used for system evaluation purpose.

4.2. Results

Experiments are conducted to determine the optimal parameters' values of the proposed method. To mine the frequent sub-graphs in each emotional class, gSpan algorithm is applied with different *N* and support values. If *N* is small the system may miss important sub-graphs from being generated which lead to increase overlapping among emotion classes and reduce the final accuracy. While when *N* is high, a chance will be given for more sub-graphs to be generated and reduce the overlap among classes. However; many of these sub-graphs are with low effectiveness. This will increase the analysis time during frequent sub-graphs dimension reduction and when selecting the optimal feature subset for each level of classification, as well. On the other hand, if support is small then gSpan will generate sub-graphs that may not exist for a large number of graphs within the emotion class and in effect reducing the accuracy. While, if support is high then gSpan will generate few sub-graphs and prevents many important sub-graphs from being generated. In a nutshell, if N is very small and support is very high, then gSpan may miss many important sub-graphs by generating a few number of sub-graphs (sub-graphs with support equal or more than the predefined support value) and then a combination consists of a small number of less effective sub-graphs will be selected. This will reduce the accuracy of the proposed system. While If N is very high and support is very small, then gSpan will generate a large number of sub-graphs in which most of them not exists in all graphs within a given emotion class and then a combination consists of high number of weak sub-graphs are selected. This will also decrease the accuracy of the proposed system. When $N = 4000$ and support = 70% the system gives the best overlap values, for this reason these value are adopted in the experiments. Table 2 shows the first ten smallest overlap ratio achieved when applying gSpan with support = 70% within each class of emotion along with sub-graph number that generate it.

In candidate sub-graphs selection step, the value for M is determined by trial and error and the best obtained value is 20. The total number of generated sub-graphs is reduced from about 2800 thousands sub-graphs to 57 sub-graphs for the seven basic emotions. Tables 3–9 show the final selected sub-graphs for different emotion classes while Table 10 demonstrates the overlap ratio that selected sub-graphs make for each emotion. Based on Table 2, the overlap ratio for emotion anger and emotion disgust is quite not stable. This refers to the fact that the graphs of anger and disgust emotions contain frequent sub-graphs with varying overlapping ratio and this affect the overlapping ratio of the final sub-graphs combination of those classes as shown in Table 10 in which they

Table 2
The first ten smallest overlap ratio achieved when applying gSpan algorithm with support = 70% within each class of emotion.

#	Angry		Disgust		Fear		Happy		Natural		Sad		Surprise	
	Sub-graph No.	Overlap Ratio	Sub-graph No.	Overlap Ratio	Sub-graph No.	Overlap Ratio	Sub-graph No.	Overlap Ratio	Sub-graph No.	Overlap Ratio	Sub-graph No.	Overlap Ratio	Sub-graph No.	Overlap Ratio
1	85	3.13%	464	31.67%	2863	18.13%	2060	16.88%	516	7.92%	643	11.67%	1569	18.13%
2	86	3.13%	289	37.29%	2864	18.13%	2061	16.88%	1073	7.92%	644	11.67%	1570	18.13%
3	1256	5.83%	288	37.29%	2870	18.13%	2062	16.88%	1748	7.92%	654	11.67%	1632	18.13%
4	1257	5.83%	436	37.50%	2871	18.13%	2063	16.88%	315	8.13%	655	11.67%	1633	18.13%
5	273	6.04%	435	37.50%	2878	18.13%	2524	16.88%	1457	8.13%	323a1	12.08%	2995	18.54%
6	1244	6.46%	580	37.50%	2879	18.13%	2525	16.88%	872	8.13%	3232	12.08%	2996	18.54%
7	1245	6.46%	592	37.5%	2940	18.54%	2528	16.88%	1493	8.54%	1119	12.29%	3058	18.54%
8	81	6.67%	463	39.17%	2941	18.54%	2529	16.88%	1495	8.54%	1120	12.29%	3059	18.54%
9	274	6.67%	207	39.58%	2947	18.54%	2530	16.88%	1496	8.54%	1108	12.29%	1858	19.38%
10	82	6.67%	462	39.58%	2948	18.54%	2531	16.88%	1492	8.54%	1109	12.29%	2601	19.38%

Table 3
Final selected sub-graphs for angry emotion.

Sub-Graph Number	gSpan Encoding	
Sub-Graph1	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 3, v_4 = 8$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,5,3), (0,4,0,9,8)
Sub-Graph2	Vertices	$v_0 = 3, v_1 = 4, v_2 = 6, v_3 = 7, v_4 = 8, v_5 = 11$
	Edges	(0,1,3,8,4), (1,2,4,4,6), (0,5,6,5,3), (1,3,4,9,7), (3,0,7,3,3), (3,4,7,3,8), (1,5,4,14,11)
Sub-Graph3	Vertices	$v_0 = 1, v_1 = 5, v_2 = 16, v_3 = 14, v_4 = 18, v_5 = 13, v_6 = 15$
	Edges	(0,1,1,3,5), (0,2,1,20,16), (2,3,16,10,14), (2,4,16,3,18), (4,5,18,9,13), (5,6,13,2,15)
Sub-Graph4	Vertices	$v_0 = 3, v_1 = 4, v_2 = 6, v_3 = 9, v_4 = 8$
	Edges	(0,1,3,8,4), (1,2,4,4,6), (2,0,6,5,3), (0,3,3,9,9), (3,4,9,7,8)
Sub-Graph5	Vertices	$v_0 = 7, v_1 = 8, v_2 = 9, v_3 = 14$
	Edges	(0,1,7,3,8), (1,2,8,7,9), (2,0,9,7,7), (2,3,9,3,14)

Table 4
Final selected sub-graphs for disgust emotion.

Sub-Graph Number	gSpan Encoding	
Sub-Graph1	Vertices	$v_0 = 3, v_1 = 4$
	Edges	(0,1,3,8,4)
Sub-Graph2	Vertices	$v_0 = 1,v_1 = 5, v_2 = 18, v_3 = 16, v_4 = 7, v_5 = 8, v_6 = 6, v_7 = 9$
	Edges	(0,1,1,3,5), (1,2,5,18,18), (2,3,18,3,16), (3,0,16,20,1), (3,4,16,15,7), (4,5,7,3,8), (5,6,8,3,6), (6,7,6,7,9)
Sub-Graph3	Vertices	$v_0 = 4, v_1 = 7, v_2 = 8$
	Edges	(0,1,4,9,7), (1,2,7,3,8)
Sub-Graph4	Vertices	$v_0 = 1, v_1 = 5, v_2 = 18, v_3 = 13, v_4 = 16, v_5 = 7, v_6 = 8$
	Edges	(0,1,1,3,5), (0,2,1,20,5), (2,3,18,9,13), (2,4,18,3,16), (4,5,16,15,7), (5,6,7,3,8)
Sub-Graph5	Vertices	$v_0 = 11, v_1 = 15$
	Edges	(0,1,11,4,15)
Sub-Graph6	Vertices	$v_0 = 1,v_1 = 5, v_2 = 3, v_3 = 18$
	Edges	(0,1,1,3,18), (1,2,5,5,3), (1,3,5,18,18), (3,0,18,20,1)
Sub-Graph7	Vertices	$v_0 = 6, v_1 = 8, v_2 = 7,v_3 = 9$
	Edges	(0,1,6,3,8), (1,2,8,3,7), (1,3,8,7,9)

Table 5
Final selected sub-graphs for fear emotion.

Sub-Graph Number	gSpan Encoding	
Sub-Graph1	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 7, v_5 = 8, v_6 = 18, v_7 = 9$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (3,4,5,4,7), (4,0,7,6,1), (4,5,7,3,8), (5,3,8,6,5), (3,6,5,18,18), (6,0,18,20,1), (6,7,18,10,9)
Sub-Graph2	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 18, v_5 = 16, v_6 = 7, v_7 = 8, v_8 = 14$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (3,4,5,18,18), (4,0,18,20,1), (4,5,18,3,16), (5,0,16,20,1), (0,6,1,6,7), (6,7,7,3,8), (7,0,8,8,1), (0,8,1,13,14)
Sub-Graph3	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 6, v_5 = 4, v_6 = 7, v_7 = 8$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (3,4,5,9,6), (4,5,6,4,4), (0,6,1,6,7), (6,7,7,3,8)
Sub-Graph4	Vertices	$v_0 = 0, v_1 = 19, v_2 = 17, v_3 = 18, v_4 = 1, v_5 = 5, v_6 = 7, v_7 = 9, v_8 = 16$
	Edges	(0,1,0,20,19), (1,2,19,3,17), (2,3,17,17,18), (3,4,18,20,1), (4,5,1,3,5), (5,3,5,18,18), (4,6,1,6,7), (3,7,18,10,9), (3,8,18,3,16)
Sub-Graph5	Vertices	$v_0 = 0, v_1 = 5, v_2 = 1, v_3 = 7, v_4 = 8, v_5 = 18, v_6 = 16, v_7 = 6, v_8 = 19$
	Edges	(0,1,0,14,5), (1,2,5,3,1), (2,3,1,6,7), (3,4,7,3,8), (4,1,8,6,5), (4,2,8,8,1), (2,5,1,20,18), (5,6,18,3,16), (0,7,0,6,6), (0,8,0,20,19)
Sub-Graph6	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 6, v_4 = 4, v_5 = 5, v_6 = 18, v_7 = 14, v_8 = 16, v_9 = 7$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (2,3,3,6,6), (3,4,6,4,4), (0,5,1,3,5), (5,6,5,18,18), (6,0,18,20,1), (6,7,18,9,14), (6,8,18,3,16), (0,9,1,6,7)
Sub-Graph7	Vertices	$v_0 = 0, v_1 = 19, v_2 = 17, v_3 = 11$
	Edges	(0,1,0,20,19), (1,2,19,3,17), (2,3,17,5,11)
Sub-Graph8	Vertices	$v_0 = 0, v_1 = 19, v_2 = 14, v_3 = 18, v_4 = 5, v_5 = 16, v_6 = 1, v_7 = 7$
	Edges	(0,1,0,20,19), (1,2,19.8,14), (2,3,14,9,18), (3,4,18,18,5), (3,5,18,3,16), (5,6,16,20,1), (6,7,1,6,7)
Sub-Graph9	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 7, v_4 = 8, v_5 = 6, v_6 = 18$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,6,7), (3,4,7,3,8), (4,0,8,8,1), (4,5,8,3,6), (0,6,1,20,18)
Sub-Graph10	Vertices	$v_0 = 0, v_1 = 19, v_2 = 13, v_3 = 15$
	Edges	(0,1,0,20,19), (1,2,19,8,13), (1,3,19,8,15)
Sub-Graph11	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 7, v_5 = 8, v_6 = 18, v_7 = 9$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (3,4,5,4,7), (4,0,7,6,1), (4,5,7,3,8), (5,3,8,6,5), (3,6,5,18,18), (6,0,18,20,1), (6,7,18,10,9)

Table 6
Final selected sub-graphs for happy emotion.

Sub-Graph Number	gSpan Encoding	
Sub-Graph1	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 7, v_5 = 8, v_6 = 18, v_7 = 9, v_8 = 14$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (0,4,1,6,7), (4,5,7,3,8), (5,0,8,8,1), (0,6,1,20,18), (6,7,18,10,9), (6,8,18,9,14)
Sub-Graph2	Vertices	v0 = 1, v1 = 2, v2 = 3, v3 = 7, v4 = 8, v5 = 18, v6 = 5, v7 = 6, v8 = 4
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,6,7), (3,4,7,3,8), (4,0,8,8,1), (0,5,1,20,18), (5,6,18,18,5), (6,7,5,9,6), (7,8,6,4,4)
Sub-Graph3	Vertices	$v_0 = 0, v_1 = 19, v_2 = 15, v_3 = 17, v_4 = 18, v_5 = 1, v_6 = 7, v_7 = 8, v_8 = 16$
	Edges	(0,1,0,20,19), (1,2,19,8,15), (1,3,19,3,17), (3,4,17,17,18), (4,5,18,20,1), (5,6,1,6,7), (6,7,7,3,8), (4,8,18,3,16)
Sub-Graph4	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 4, v_4 = 6, v_5 = 7$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (2,3,3,9,4), (2,4,3,6,6), (0,5,1,6,7)
Sub-Graph5	Vertices	$v_0 = 0, v_1 = 4, v_2 = 1, v_3 = 5, v_4 = 7, v_5 = 8, v_6 = 18, v_7 = 6, v_8 = 19$
	Edges	(0,1,0,3,4), (1,2,4,14,1), (2,3,1,3,5), (2,4,1,6,7), (2,5,1,8,8), (2,6,1,20,18), (0,7,0,6,6), (0,8,0,20,19)
Sub-Graph6	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 18, v_5 = 16, v_6 = 7, v_7 = 8, v_8 = 14$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (3,4,5,18,18), (4,0,18,20,1), (4,5,18,3,16), (5,0,16,20,1), (0,6,1,6,7), (6,7,7,3,8), (7,0,8,8,1), (0,8,1,13,14)
Sub-Graph7	Vertices	$v_0 = 0, v_1 = 19, v_2 = 17, v_3 = 11$
	Edges	(0,1,0,20,19), (1,2,19,3,17), (2,3,17,5,11)
Sub-Graph8	Vertices	$v_0 = 0, v_1 = 19, v_2 = 14, v_3 = 12, v_4 = 18, v_5 = 1, v_6 = 5, v_7 = 7, v_8 = 8, v_9 = 16$
	Edges	(0,1,0,20,19), (1,2,19,8,14), (2,3,14,1,12), (2,4,14,9,18), (4,5,18,20,1), (5,6,1,3,5), (6,4,5,18,18), (5,7,1,6,7), (7,8,7,3,8), (4,9,18,3,16)
Sub-Graph9	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 18, v_5 = 16, v_6 = 7, v_7 = 8, v_8 = 9, v_9 = 6$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (3,4,5,18,18), (4,0,18,20,1), (4,5,18,3,16), (5,0,16,20,1), (0,6,1,6,7), (6,7,7,3,8), (7,0,8,8,1), (6,8,7,7,9), (8,9,9,7,6)
Sub-Graph10	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 18, v_5 = 16, v_6 = 7, v_7 = 8, v_8 = 6$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (3,4,5,18,18), (4,5,18,3,16), (5,0,16,20,1), (0,6,1,6,7), (6,7,7,3,8), (7,0,8,8,1), (7,8,8,3,6)
Sub-Graph11	Vertices	$v_0 = 1, v_1 = 2, v_2 = 3, v_3 = 5, v_4 = 7, v_5 = 8, v_6 = 18, v_7 = 9, v_8 = 14$
	Edges	(0,1,1,10,2), (1,2,2,4,3), (2,0,3,7,1), (0,3,1,3,5), (0,4,1,6,7), (4,5,7,3,8), (5,0,8,8,1), (0,6,1,20,18), (6,7,18,10,9), (6,8,18,9,14)

Table 7
Final selected sub-graphs for natural emotion.

Sub-Graph Number	gSpan Encoding	
Sub-Graph1	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 1, v_6 = 5, v_7 = 18, v_8 = 16,$ $v_9 = 10, v_{10} = 14, v_{11} = 12, v_{12} = 9, v_{13} = 11$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6.6,0), (2,3,6,3,8), (3,1,8,6,4), (3,4,8,3,7), (4,1,7,9,4), (4,5,7,6,1), (5,6,1,3,5), (6,4,5,4,7), (5,7,1,20,18), (7,8,18,3,16), (4,9,7,12,10), (9,10,10,4,14), (10,11,14,1,12), (11,12,12,3,9), (12,2,9,7,6), (11,13,12,4,11)
Sub-Graph2	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 1, v_6 = 5, v_7 = 9, v_8 = 11, v_9 = 14, v_{10} = 10$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,1,8,6,4), (3,4,8,3,7), (4,1,7,9,4), (4,5,7,6,1), (5,6,1,3,5), (6,4,5,4,7), (4,7,7,7,9), (7,2,9,7,6), (2,8,6,12,11), (8,1,11,14,4), (8,9,11,4,14), (9,10,14,4,10)
Sub-Graph3	Vertices	$v_0 = 0, v_1 = 4,v_2 = 6, v_3 = 8, v_4 = 9$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,1,8,6,4), (3,4,8,7,9)
Sub-Graph4	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 3$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,6,3)
Sub-Graph5	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 1$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,11,1)
Sub-Graph6	Vertices	$v_0 = 0, v_1 = 5$
	Edges	(0,1,0,14,5)
Sub-Graph7	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 1, v_6 = 5, v_7 = 18, v_8 = 16, v_9 = 10, v_{10} = 14, v_{11} = 12, v_{12} = 9, v_{13} = 11$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,1,8,6,4), (3,4,8,3,7), (4,1,7,9,4), (4,5,7,6,1), (5,6,1,3,5), (6,4,5,4,7), (5,7,1,20,18), (7,8,18,3,16), (4,9,7,12,10), (9,10,10,4,14), (10,11,14,1,12), (11,12,12,3,9), (12,2,9,7,6), (11,13,12,4,11)

Table 8
Final selected sub-graphs for sad emotion.

Sub-Graph Number	gSpan Encoding	
Sub-Graph1	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 9, v_6 = 12, v_7 = 14, v_8 = 11,v_9 = 19$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,4,8,3,7), (4,5,7,7,9), (5,1,9,9,4), (5,2,9,7,6), (5,6,9,3,12), (6,7,12,1,14), (7,2,14,10,6), (7,4,14,10,7), (7,8,14,4,11), (8,1,11,14,4), (8,2,11,12,6), (6,9,12,8,19)
Sub-Graph2	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 9, v_6 = 12, v_7 = 14, v_8 = 10, v_9 = 19$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,4,8,3,7), (4,5,7,7,9), (5,1,9,9,4), (5,2,9,7,6), (5,6,9,3,12), (6,7,12,1,14), (7,2,14,10,6), (7,4,14,10,7), (7,8,14,4,10), (8,4,10,12,7), (6,9,12,8,19)
Sub-Graph3	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 9, v_6 = 11, v_7 = 18, v_8 = 1, v_9 = 5, v_{10} = 16$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,4,8,3,7), (4,5,7,7,9), (5,1,9,9,4), (5,2,9,7,6), (2,6,6,12,11), (6,1,11,14,4), (6,7,11,12,18), (7,8,18,20,1), (8,9,1,3,5), (9,7,5,18,18), (8,10,1,20,16), (10,7,16,3,18)
Sub-Graph4	Vertices	$v_0 = 0, v_1 = 4, v_2 = 3, v_3 = 6$
	Edges	(0,1,0,3,4), (1,2,4,9,3), (1,3,4,4,6)
Sub-Graph5	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 9, v_6 = 19, v_7 = 16, v_8 = 1, v_9 = 18$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,4,8,3,7), (4,5,7,7,16), (5,1,9,9,4), (5,2,9,7,6), (0,6,0,20,19), (6,7,19,17,16), (7,8,16,20,1), (8,9,1,20,18), (9,7,18,3,16)
Sub-Graph6	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 9, v_6 = 12, v_7 = 14, v_8 = 10, v_9 = 16, v_{10} = 1, v_{11} = 5, v_{12} = 18$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,4,8,3,7), (4,5,7,7,9), (5,1,9,9,4), (5,2,9,7,6), (5,6,9,3,12), (6,7,12,1,14), (7,2,14,10,6), (7,4,14,10,7), (4,8,7,12,10), (4,9,7,15,16), (9,10,16,20,1), (10,11,1,3,5), (11,12,5,18,18), (12,10,18,20,1)
Sub-Graph7	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 9, v_6 = 16, v_7 = 1, v_8 = 5, v_9 = 18$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0) (2,3,6,3,8), (3,4,8,3,7), (4,5,7,7,9), (5,1,9,9,4), (5,2,9,7,6), (5,6,9,10,16), (6,7,16,20,1), (7,8,1,3,5), (8,9,5,18,18), (9,6,18,3,16)
Sub-Graph8	Vertices	$v_0 = 0, v_1 = 4, v_2 = 6, v_3 = 8, v_4 = 7, v_5 = 9, v_6 = 12, v_7 = 14, v_8 = 11,v_9 = 19$
	Edges	(0,1,0,3,4), (1,2,4,4,6), (2,0,6,6,0), (2,3,6,3,8), (3,4,8,3,7), (4,5,7,7,9), (5,1,9,9,4), (5,2,9,7,6), (5,6,9,3,12), (6,7,12,1,14), (7,2,14,10,6), (7,4,14,10,7), (7,8,14,4,11), (8,1,11,14,4), (8,2,11,12,6), (6,9,12,8,19)

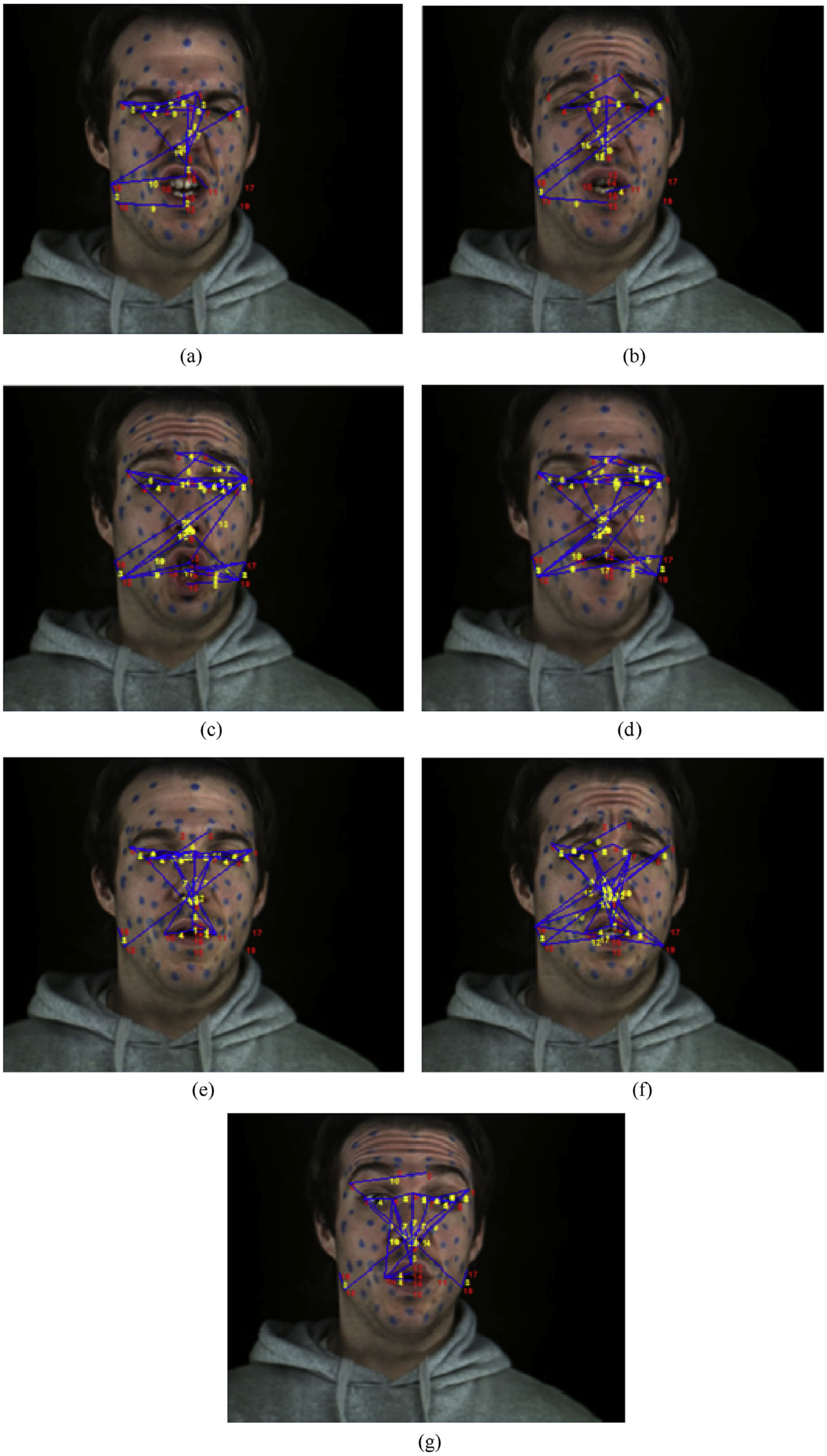
Table 9
Final selected sub-graphs for surprise emotion.

Sub-Graph Number	gSpan Encoding	
Sub-Graph1	Vertices	$v_0 = 1, v_1 = 5, v_2 = 7, v_3 = 8, v_4 = 6, v_5 = 9, v_6 = 10, v_7 = 12$
	Edges	(0,1,1,3,5), (1,2,5,4,7), (2,0,7,6,1), (2,3,7,3,8), (3,1,8,6,5), (3,4,8,3,6), (4,5,6,7,9), (5,2,9,7,7), (5,3,9,7,8), (4,6,6,10,10), (6,7,10,4,12)
Sub-Graph2	Vertices	$v_0 = 1, v_1 = 5, v_2 = 7, v_3 = 8, v_4 = 9, v_5 = 6, v_6 = 12, v_7 = 18$
	Edges	(0,1,1,3,5), (1,2,5,4,7), (2,0,7,6,1), (2,3,7,3,8), (3,1,8,6,5), (2,4,7,7,9), (4,1,9,9,5), (4,5,9,7,6), (4,6,9,3,12), (0,7,1,20,18)
Sub-Graph3	Vertices	$v^0 = 1, v_1 = 5, v_2 = 7, v_3 = 8, v_4 = 6, v_5 = 4, v_6 = 9, v_7 = 12$
	Edges	(0,1,1,3,5), (1,2,5,4,7), (2,0,7,6,1), (2,3,7,3,8), (3,1,8,6,5), (3,4,8,3,6), (4,5,6,4,4), (4,6,6,7,12), (6,2,9,7,7), (6,3,9,7,8), (4,7,6,9,12)
Sub-Graph4	Vertices	$v_0 = 0, v_1 = 3, v_2 = 19, v_3 = 17$
	Edges	(0,1,0,10,3), (0,2,0,20,19), (2,3,19,3,17)
Sub-Graph5	Vertices	$v_0 = 1, v_1 = 5, v_2 = 7, v_3 = 8, v_4 = 6, v_5 = 10, v_6 = 12, v_7 = 14, v_8 = 15$
	Edges	(0,1,1,3,5), (1,2,5,4,7), (2,0,7,6,0), (2,3,7,3,8), (3,1,8,6,5), (3,4,8,3,6), (4,5,6,10,10), (5,6,10,4,12), (5,7,10,4,14), (5,8,10,4,15)
Sub-Graph6	Vertices	$v_0 = 1, v_1 = 5, v_2 = 7, v_3 = 8, v_4 = 6, v_5 = 9, v_6 = 10, v_7 = 18$
	Edges	(0,1,1,3,5), (1,2,5,4,7), (2,0,7,6,1), (2,3,7,3,8), (3,1,8,6,5), (3,4,8,3,6), (4,5,6,7,9), (5,2,9,7,7), (5,3,9,7,8), (1,6,5,14,10), (0,7,1,20,18)
Sub-Graph7	Vertices	v0 = 1, v1 = 5, v2 = 7, v3 = 8, v4 = 6, v5 = 9, v6 = 4, v7 = 18, v8 = 16
	Edges	(0,1,1,3,5), (1,2,5,4,7), (2,0,7,6,1), (2,3,7,3,8), (3,1,8,6,5), (3,4,8,3,6), (4,5,6,7,9), (5,2,9,7,7), (5,3,9,7,8), (5,6,9,9,4), (0,7,1,20,18), (7,8,18,3,16)
Sub-Graph8	Vertices	$v_0 = 1, v_1 = 5, v_2 = 7, v_3 = 8, v_4 = 6, v_5 = 9, v_6 = 10, v_7 = 12$
	Edges	(0,1,1,3,5), (1,2,5,4,7), (2,0,7,6,1), (2,3,7,3,8), (3,1,8,6,5), (3,4,8,3,6), (4,5,6,7,9), (5,2,9,7,7), (5,3,9,7,8), (4,6,6,10,10), (6,7,10,4,12)

Table 10
Overlap ratio of final selected sub-graphs for each emotional class.

Emotion	Angry	Disgust	Fear	Happy	Natural	Sad	Surprise
Overlap Ratio	0.00%	0.00%	1.04%	1.25%	3.96%	3.33%	0.21%

BCSO is utilized to select proper features on each level of classification with following configuration settings.
Number of cats = 3, Counts of dimensions to change = 0.2, Seeking memory pool = 3, Mixture ratio = 0.3, Mutation ratio = 0.2, Number of iterations = 10, Velocity max = 50.


(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig. 6. Final Selected sub-graphs, (a) angry emotion, (b) disgust emotion, (c) fear emotion, (d) happy emotion, (e) natural emotion, (f) sad emotion, (g) surprise emotion.

Table 11
The recognition accuracy for different levels with different emotion combinations in each level.

Level6	Level5	Level4	Level3	Level2	Level1	
–	–	–	–	–	98.75%	Angry
–	–	–	98.06%	95.48%	95.21%	Disgust
–	–	94.67%	86.87%	93.81%	93.33%	Fear
–	–	–	–	97.62%	95.83%	Happy
97.78%	92.92%	86.33%	92.22%	90.95%	92.29%	Natural
97.78%	90.83%	89%	93.89%	90.24%	92.92%	Sad
–	97.92%	88.67%	93.06%	95.24%	92.71%	Surprise
97.78%	97.92%	94.67%	98.06%	97.62%	98.75%	Best Accuracy
90.00%						Final Accuracy

Table 12
Number of feature selected using BCO within each level of classification.

Levels	Level1	Level2	Level3	Level4	Level5	Level6
Number of selected features	31	27	27	30	30	28

Table 13
A Comparison between current research works on SAVEE database and the proposed system.

Authors	Methodology	Accuracy
Haq et al. (2015) [6]	MP and MA features and SVM classifier	88%
Li et al. (2017) [22]	240- dimensional visual features	87.5%
Thattil et al. (2019) [23]	Viola–Jones face recognition algorithm and facial image emotion classification by CNN (AlexNet)	77.4%
The proposed graph mining based method	gSpan frequent sub-graphs mining, BCSO and NN	90.00%

gave zero overlap ratio despite other emotions gave some overlapping. Fig. 6 illustrates locations of the final selected sub-graphs in face region within each emotion.

Table 11 shows the recognition accuracy achieved when applying BCSO for different levels with different emotion combinations in each level. As can be observed, emotions except for fear can be recognized quite well by using binary classification. The final accuracy achieved by the system for all levels is 90.00% due to the error occurs in each level of classification.

Table 4 shows class precision for different data-algorithm combinations. As can be observed, all emotions except for sadness can be recognized quite. well by using unimodal and multimodal datasets. Anger is best recognized by applying the W-SMO algorithm on the LBP-TOP dataset, happiness is best recognized by applying a neural network on the. combined audio-visual dataset etc.

Table 12 shows the number of selected features in each level of classification. As depicted in the table, Level1 needs the maximum number of features because in this level, angry class must be separated from all other emotions and thus requires more features. Level4 and Level5 work with 30 features which refer to the difficulty of separating natural and sad emotions from remaining ones. While Levels 2, 3 and 6 required the less number of features with accuracy more than 97%.

When comparing the achieved results with other current research works on SAVEE database, it shown that an improvement of about 2% is achieved on the accuracy of facial emotion recognition using the proposed graph mining method as shown in Table 13.

5. Conclusions and future works

Facial emotion recognition finds many real life applications such as human robot interaction, security, medicine, education etc. A novel method for emotion classification has been proposed in this paper. Graph theory and mining concepts were employed for finding frequent sub-graphs in each emotional class through using gSpan algorithm. The selection of best candidate sub-graphs using overlap ratio metric helps in reducing number of redundant sub-graphs and focusing the work on the useful ones only. On other hand, using binary classification in six levels has the great impact on increasing the final accuracy of the system. Experimental results have clearly showed the effectiveness of the proposed system with accuracy of about 90.00% using SAVEE database. As a future work, graph clustering techniques such as k-medois can be employed to cluster the facial graphs within each emotion into k different clusters and then applying gSpan on each cluster instead of each class. This may help in further reducing of the overlap ratio among classes. In addition, fuzzy logic can be used instead of crisp binary encoding of sub-graphs by allowing each sub-graph to have a membership degree between 0 and 1 on each emotion according to the number of matched nodes and edges. The usage of fuzzy encoding will help in avoiding the problem of partial incomplete vector-space representation of the encoded sub-graphs.

References

[1] Dagar D, Hudait A, Tripathy HK, Das MN. Automatic emotion detection model from facial expression. In: International conference on advanced communication control and computing technologies (ICACCCT), ramanathapuram; 2016. p. 77–85.

[2] Anurag D, Ashim S. A comparative study on different approaches of real time human emotion recognition based on facial expression detection. In: 2015 international conference on advances in computer engineering and applications. Ghaziabad; 2015. p. 483–7.

[3] Ghosh S, Kundu D, Paul G. A fuzzy logic approach in emotion detection and recognition and formulation of an odor-based emotional fitness assistive system. Int J Synth Emot (IJSE) 2015;6:14–34.

[4] Liu C, Ham J, Postma E, Midden C, Joosten B, Goudbeek M. Representing affective facial expressions for robots and embodied conversational agents by facial landmarks. Int J Soc Robot 2013;5:619–26.

[5] Mohseni S, Zarei N, Ramazani S. Facial expression recognition using anatomy based facial graph. In: IEEE international conference on systems, man, and cybernetics. San Diego, CA: SMC; 2014. p. 3715–9. 2014.

[6] Haq S, Asif MA, Jan T, Ahmad N, Khan Y. Audio-visual emotion classification using filter and wrapper feature selection approaches. Sind Univ Res J (Sci Ser) 2015;47 –1:67–72.

[7] Mohammed SN, George LE, Ayad H. The effect of classification methods on facial emotion recognition accuracy. Br J Appl Sci Technol 2016;14:1–11.

[8] Munasinghe MI. Facial expression recognition using facial landmarks and random forest classifier. In: 2018 IEEE/ACIS 17th international conference on computer and information science. Singapore: ICIS; 2018. p. 423–7.

[9] Cook DJ, Holder. LB mining graph data. John Wiley and Sons; 2007.

[10] Hanmandlu M, Gupta D, Vasikarla S. Face recognition using elastic Bunch graph matching. In: IEEE applied imagery pattern recognition workshop. Washington, DC: AIPR; 2013. p. 1–7. 2013.

[11] López FA, Higinio MM, José AG. A neural network framework for face recognition by elastic Bunch graph matching. Recent Advances on Electroscience and Computers 2015:75–81. 978-1-61804-290-3.

[12] Aggarwal CC, Wang H. Managing and mining graph data. Springer; 2010.

[13] Samatova NF, Hendrix W, Jenkins J, Jenkins K, Chakraborty A. Practical graph mining with R. CRC Press; 2014.

[14] Aggarwal CC. Data mining the textbook. Springer; 2015.

[15] Brezocnik L, Fister I, Podgorelec V. Swarm intelligence algorithms for feature selection: a review. Appl Sci 2018;8:15–21.

[16] Chu SC, Tsai P, Pan JS. Cat swarm optimization. In: Yang Q, Webb G, editors. PRICAI 2006: trends in artificial intelligence. PRICAI 2006. Lecture notes in computer science. Berlin, Heidelberg: Springer; 2006. p. 854–8.

[17] Sharafi Y, Khanesar MA, Teshnehlab M. Discrete binary cat swarm optimization algorithm. In: 3rd IEEE international conference on computer, control and communication, IC4; 2013. p. 1–6.

[18] Kazemi V, Sullivan J. One millisecond face alignment with an ensemble of

regression trees. In: The IEEE conference on computer vision and pattern recognition. CVPR; 2014. p. 1867–74.
[19] Weir MD, Hass J, Giordano FR. Thomas calculus. Pearson Education; 2008.
[20] Herout A. Pattern recognition, recent advances. TECH Education and Publication; 2010.
[21] Surrey audio-visual expressed emotion (SAVEE) database home page. http://kahlan.eps.surrey.ac.uk/savee/. [Accessed 7 March 2019].
[22] Li Y, He Q, Zhao Y, Yao H. Multi-modal emotion recognition based on speech and image. Advances in multimedia information processing. PCM; 2017.
[23] Avots E, Sapinski T, Bachmann M, Kaminska D. Audiovisual emotion recognition in wild. Mach Vis Appl 2019;30:975–85.