Emotional Video Scene Retrieval Using Multilayer Convolutional Network

Hiroki Nomiya, Shota Sakaue, Mitsuaki Maeda and Teruhisa Hochin

Abstract In order to retrieve impressive scene from a video database, a scene retrieval method based on facial expression recognition (FER) is proposed. The proposed method will be useful to retrieve interesting scenes from lifelog videos. When an impressive event occurs, a certain facial expression will be observed in a person in the video. It is, therefore, important for the impressive scene retrieval to precisely recognize the facial expression of the person. In this paper, we try to construct accurate FER models by introducing a learning framework on the basis of multilayer convolutional network using a number of facial features defined as the positional relations between some facial feature points. The effectiveness of the proposed method is evaluated through an experiment to retrieve emotional scenes from a lifelog video database.

1 Introduction

Owing to recent development of multimedia recording devices such as video cameras and smart phones, people can easily record their daily lives as video data. Since various services to post and view the videos on the Internet are available for free, we can find and/or provide a number of interesting videos at any time. However, some people have a large amount of private video data which cannot be accessed by general public. It will be difficult to retrieve interesting scenes from such private video databases.

In order to solve this issue, we propose a video scene retrieval method to find some impressive scenes from a video database. The proposed method does not require the video database to be public. It thus can be used for private video databases such

H. Nomiya (✉) · S. Sakaue · M. Maeda · T. Hochin
Department of Information Science, Kyoto Institute of Technology, Goshokaido-cho,
Matsugasaki, Sakyo-ku, Kyoto 606-8585, Japan
e-mail: nomiya@kit.ac.jp

T. Hochin
e-mail: hochin@kit.ac.jp

R. Lee (ed.), *Applied Computing and Information Technology*,
Studies in Computational Intelligence 695,
DOI 10.1007/978-3-319-51472-7_8

as lifelog video databases [1]. The proposed method detects the impressive scenes using facial expression recognition (FER). This is because a certain facial expression will be observed in a person in the video when an impressive event occurs. The performance of the video scene retrieval is thus largely dependent on the accuracy of FER.

FER has been applied to video scene detection [2, 3]. Most of existing FER techniques manually define their own facial features and discriminate facial expression using them. The facial feature is one of the core elements in the FER and dominates the recognition performance. It is, however, not easy to manually select good facial features because a variety of very subtle and complex movements of several facial parts will be observed in the appearance of a facial expression.

There is an impressive video scene retrieval method on the basis of an FER method which tries to solve this problem by introducing evolutionary facial feature creation [4]. In this method, useful facial features are generated by combining several arithmetic operations for the positions of facial feature points using genetic programming. This method has an advantage that the facial features are automatically generated. However, it is generally difficult to generate complex facial feature because there are almost infinite combinations of facial features.

The FER technique in the proposed method also utilizes the positional relation between some facial features. Since generating facial features is quite difficult because of the combination problem, the proposed method predefines a number of facial features and selects some useful facial features by introducing a feature selection method. In order to enhance the FER accuracy, we introduce a learning framework on the basis of multilayer convolutional network. Convolutional network is widely used for such as image recognition and speech recognition [5] and currently known as a component of deep learning [6].

The performance of the proposed method is evaluated through an experiment to retrieve emotional scenes from a lifelog video database. We focus on the retrieval of the video scenes with smiles because some interesting events will occur in such scenes and many people will want to retrieve them. The retrieval accuracy of the proposed method is compared with that of the aforementioned retrieval method [4].

The remainder of this paper is organized as follows. Section 2 shows the facial features. Section 3 explains the feature selection method. Section 4 describes the FER method using the facial features. Section 5 introduces the method to detect emotional scenes on the basis of the result of the FER. Section 6 evaluates the performance of the proposed method through an experiment. Section 7 give some consideration about the experimental result. Finally, Sect. 8 concludes this paper.

2 Facial Feature

The proposed method uses a number of facial features computed on the basis of positional relation of several salient points on a face called facial feature points.

2.1 Facial Feature Points

We use 59 facial feature points as shown in Fig. 1. They consist of salient points on left and right eyebrows (10 points), left and right eyes (22 points), a nose (9 points), a mouth (14 points), and left and right nasolabial folds (4 points). They are obtained by using a publicly available software called Luxand FaceSDK (version 4.0) [8].

2.2 Facial Features

The feature value used in the proposed method is computed as the cosine of the angle ($\cos\theta$) between two line segments formed by three facial feature points p_i, p_j, and p_k as shown in Fig. 2 [9]. Figure 3 shows an example of the facial feature when p_j and p_k are the end points of the mouth and p_i is the center point of the left eye.

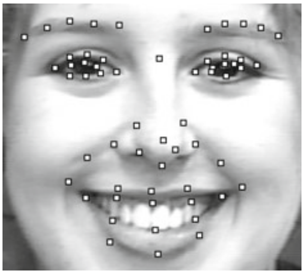

Fig. 1 Facial feature points denoted by white squares (this facial image is from Cohn-Kanade AU-Coded Facial Expression Database [7])

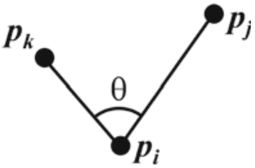

Fig. 2 A facial feature

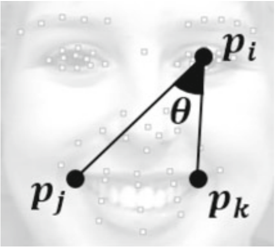

Fig. 3 An example of a facial feature

The facial feature value $f_{i,j,k}$ computed from p_i, p_j, and p_k is defined as Eq. (1).

$$f_{i,j,k} = \frac{X_{ji}X_{ki} + Y_{ji}Y_{ki}}{\sqrt{X_{ji}^2 + Y_{ji}^2}\sqrt{X_{ki}^2 + Y_{ki}^2}} \tag{1}$$

Here, X_{ji}, X_{ki}, Y_{ji}, and Y_{ki} are defined as Eq. (2).

$$\begin{aligned} X_{ji} = x_j - x_i,\ X_{ki} = x_k - x_i \\ Y_{ji} = y_j - y_i,\ Y_{ki} = y_k - y_i \end{aligned} \tag{2}$$

where x_a and y_a are the x- and y-coordinates of the facial feature point p_a ($a \in \{i, j, k\}$), respectively.

3 Feature Selection

A total of 97527 possible facial features can be defined from 59 facial feature points.[1] Therefore, using all the possible facial features leads to high computational cost. In addition, there are a lot of useless or redundant facial features. For the purpose of efficient FER, we introduce feature selection to select a small number of useful facial features.

The feature selection is performed on the basis of the usefulness of each facial feature. The usefulness is defined as the variance ratio of the between-class variance to the within-class variance. Equation (3) shows the usefulness Z_i of the ith facial feature.[2]

$$Z_i = \frac{V_i^B}{V_i^W} \tag{3}$$

Here, V_i^B and V_i^W are the ith between-class variance and within-class variance, respectively. They are computed from the facial feature values of training samples. V_i^B and V_i^W are defined by Eqs. (4) and (5), respectively.

$$V_i^B = \sum_{j=1}^{C} \frac{N_j}{N}(\mu_{j,i} - \mu_j)^2 \tag{4}$$

[1] One point which is a common end point of two line segments (i.e., p_i shown in Fig. 2) can be selected from 59 facial feature points. Then, two points can be selected from remaining 58 facial feature points. The number of possible facial features is thus $59 \times {}_{58}C_2 = 97527$.

[2] For example, the 1st facial feature corresponds to $f_{1,2,3}$ and the 2nd one is $f_{1,2,4}$. The 97527th facial feature is $f_{57,58,59}$.

$$V_i^W = \frac{1}{N} \sum_{j=1}^{C} \sum_{k=1}^{N_j} (\phi_{i,j,k} - \mu_{j,i})^2 \tag{5}$$

In the above equations, C is the number of facial expressions and corresponds to the number of classes in classification problem. The number of training samples belonging to the ith class is denoted by N_i. The total number of training samples is denoted by N (i.e., $\sum_{i=1}^{C} N_i = N$). $\phi_{i,j,k}$ is the ith facial feature value of the kth sample belonging to the jth class. $\mu_{j,i}$ is the mean value of the ith facial feature values of all the samples belonging to the jth class. μ_i is the mean value of the mean values of the ith facial feature (i.e., $\mu_i = \frac{1}{C} \sum_{j=1}^{C} \mu_{j,i}$).

A larger between-class variance makes the classification problem easier since this indicates that the centroid of each class is distant from each other. On the other hand, a smaller within-class variance is more effective since the samples belonging to the same class are close to each other. Therefore, the facial feature having high usefulness value will be useful to discriminate facial expressions.

4 Facial Expression Recognition

The proposed FER model is constructed based on multilayer convolutional network using the selected facial features. Considering the tradeoff between accuracy and efficiency, we use two layers. The convolutional network for a single layer consists of convolution and pooling.

Similar to the image recognition using convolutional network, we represent the facial feature values of M^2 selected facial features as an $M \times M$ matrix. The M^2 selected facial feature means the top M^2 facial features of the usefulness. The adjacent facial features (i.e., the nth selected facial feature and the $(n+1)$th one) are sometimes similar to each other (for example, one is calculated from the facial feature points p_1, p_2, and p_3 while the other is calculated from the ones p_1, p_2, and p_4). We represent the selected facial features as a matrix so that diverse facial features are used in the convolution and pooling steps since we believe that the diversity of facial features contributes to higher recognition accuracy.

At the first layer, the convolution operation yields a total of L_1 convolution matrices having the same size as F in accordance with Eq. (6).

$$K_1(i,j,k,l) = \sum_{a=1}^{S} \sum_{b=1}^{S} w_1(a,b,l) F(i, j+a, k+b) \tag{6}$$

$$(1 \le i \le N, 1 \le j \le M, 1 \le k \le M, 1 \le l \le L_1)$$

Here, $K_1(i,j,k,l)$ is the (j,k) entry of the lth convolution matrix produced for the ith training sample. $w_1(a,b,l)$ is the (a,b) entry of the lth weight matrix. A weight matrix is an $S \times S$ matrix and S is the patch size. Each entry of the weight matrix is initialized by a random value generated based on a normal distribution (we experimentally set the mean value to 0 and the standard deviation to 0.1). $F(i, j, k)$ is the $(j \times M + k)$th (selected) facial feature value of the ith training sample. Note that the subscripts of K, L and w mean the layer number.

After the convolution operation, a pooling operation is applied to the convolution matrices. As a result of the pooling operation, a total of L_1 pooling matrices are produced according to Eq. (7).

$$P_1(i,j,k,l) = \frac{1}{T^2} \sum_{a=1}^{T} \sum_{b=1}^{T} K_1'(i,(j-1)T + a,(k-1)T + b,l) \tag{7}$$

$$\left(1 \le i \le N, 1 \le j \le \frac{M}{T}, 1 \le k \le \frac{M}{T}, 1 \le l \le L_1\right)$$

where, K_1' is defined as Eq. (8).

$$K_1'(i,j,k,l) = RL(K_1(i,j,k,l) + \beta_1(l)) \tag{8}$$

In the above equation, β_1 is an L_1-dimensional bias vector. We initialize the bias vector such that the values of all the entries are 0.1. The function RL is called ReLU (rectified linear unit) function and defined as $RL(x) = \max\{0, x\}$. Note that the subscripts of P and β mean the layer number, and that the size of each pooling matrix is $\frac{M}{T} \times \frac{M}{T}$.

At the second layer, the convolution operation produces a total of L_2 convolution matrices using the pooling result of the first layer as defined in Eq. (9).

$$K_2(i,j,k,l) = \sum_{a=1}^{S} \sum_{b=1}^{S} \sum_{c=1}^{L_1} w_2(a,b,c,l)P_1(i,j + a,k + b,c) \tag{9}$$

$$\left(1 \le i \le N, 1 \le j \le \frac{M}{T}, 1 \le k \le \frac{M}{T}, 1 \le l \le L_2\right)$$

where, $w_2(a, b, c, l)$ is the (a, b) entry of the weight matrix defined for the cth pooling matrix generated in the first layer and for the lth convolution matrix generated in the second layer. We initialize w_2 by the same way as used to initialize w_1. Then, the pooling operation is performed according to Eq. (10).

$$P_2(i,j,k,l) = \frac{1}{T^2} \sum_{a=1}^{T} \sum_{b=1}^{T} K_2'(i,(j-1)T + a,(k-1)T + b,l) \tag{10}$$

$$\left(1 \le i \le N, 1 \le j \le \frac{M}{T^2}, 1 \le k \le \frac{M}{T^2}, 1 \le l \le L_2\right)$$

where, K_2' is defined as Eq. (11).

$$K'_2(i,j,k,l) = RL(K_2(i,j,k,l) + \beta_2(l)) \tag{11}$$

Note that weight vector β_2 is L_2-dimensional. We initialize β_2 using the same way as in the initialization of β_1.

The output (i.e., a set of pooling matrices) of the second layer is passed to a fully-connected layer. The layer consists of U units and their output values are represented as an $N \times U$ matrix Λ defined as Eq. (12).

$$\Lambda = RL(\bar{P_2} w_\Lambda + \beta_\Lambda) \tag{12}$$

Here, $\bar{P_2}$ is a matrix defined by converting P_2 into a matrix whose number of rows is N and that of columns is $(\frac{M}{T^2} \times \frac{M}{T^2} \times L_2)$.[3] w_Λ is a weight matrix whose number of rows is $(\frac{M}{T^2} \times \frac{M}{T^2} \times L_2)$ and that of columns is U. β_Λ is an $N \times U$ bias matrix that each row is the same U-dimensional bias vector. When x is a matrix, $RL(x) = X$ such that $X_{ij} = \max\{0, x_{ij}\}$, where X_{ij} and x_{ij} are the (i,j) entries of X and x, respectively. Note that we initialize w_Λ and the bias vector for β_Λ using the same way as used to initialize w_1 and β_1, respectively.

Using the outputs of all units often leads to overfitting. We therefore introduce "dropout" to prevent overfitting [10]. By introducing dropout, several units are dropped at random during training phase. We experimentally set the rate of dropout to 0.5. This means that the outputs of a half of units are ignored when Λ is computed.

Finally, the readout layer is constructed based on the outputs of the fully-connected layer. The output Q of the readout layer is defined as Eq. (13).

$$Q = SM(\Lambda w_Q + \beta_Q) \tag{13}$$

In the above equation, SM is the softmax function defined by Eq. (14).

$$SM(\chi) = \begin{pmatrix} \frac{e^{\chi_{11}}}{\sum_{i=1}^{C} e^{\chi_{1i}}} & \cdots & \frac{e^{\chi_{1C}}}{\sum_{i=1}^{C} e^{\chi_{1i}}} \\ \vdots & \ddots & \vdots \\ \frac{e^{\chi_{N1}}}{\sum_{i=1}^{C} e^{\chi_{Ni}}} & \cdots & \frac{e^{\chi_{NC}}}{\sum_{i=1}^{C} e^{\chi_{Ni}}} \end{pmatrix} \tag{14}$$

where, χ is an $N \times C$ matrix and χ_{ij} is the (i,j) entry of χ. w_Q is a $U \times C$ weight matrix. β_Q is an $N \times C$ bias matrix that each row is the same C-dimensional bias vector. Note that we initialize w_Q and the bias vector for β_Q using the same way as used to initialize w_1 and β_1, respectively.

The output of the readout layer Q is represented as an $N \times C$ matrix. It indicates the possibility of each facial expression for each training sample. For example, the (i,j) entry of Q corresponds to the possibility that the person in the ith training sample expresses the jth facial expression. Therefore, the proposed FER model can predict the facial expression (i.e., class label) of the training examples by Eq. (15).

[3] The $(i, 1)$ entry of $\bar{P_2}$ is $P_2(i, 1, 1, 1)$, and the $(i, 2)$ entry of it is $P_2(i, 1, 1, 2)$, and so on. The $(i, \frac{M}{T^2} \times \frac{M}{T^2} \times L_2)$ entry of $\bar{P_2}$ is $P_2(i, \frac{M}{T^2}, \frac{M}{T^2}, L_2)$.

$$q(i) = \operatorname*{argmax}_{j} Q(i,j) \tag{15}$$

where, $q(i)$ is the predicted class label for the ith training sample and $Q(i,j)$ is the (i,j) entry of Q.

The goal for the construction of the FER model is to optimize the parameters such as weights and biases. To do this, we use Adam algorithm [11]. It is the algorithm for first-order gradient-based optimization of stochastic objective functions. We use the cross entropy as the objective function.

After the training (i.e., the optimization of the parameters), the recognition of the facial expression of unseen samples (test samples) can be performed. This is done simply by replacing the facial feature values of training samples in F (see Eq. (6)) with those of test samples.

5 Emotional Scene Detection

The emotional scenes are detected from a video according to the predicted class label for each frame image. The class label is predicted by the FER model described in Sect. 4. The emotional scenes with a certain facial expression are determined by using the frame images having the corresponding class labels. We make use of the emotional scene detection method proposed in [4].

At the first step of the emotional scene detection, each frame image having the corresponding class label is regarded as a single emotional scene. Then, neighboring emotional scenes are integrated into a single emotional scene. The integration process is repeated until no more emotional scenes can be integrated. The resulting scenes are output as the emotional scenes of the facial expression.

The algorithm of the emotional scene detection is shown in Algorithm 1. Since the emotional scene detection algorithm can find the emotional scenes for a single facial expression, it is required to perform the emotional scene detection C times when there are C kinds of facial expressions in a video.

6 Experiment

6.1 Experimental Settings

6.1.1 Data Set

As the data set to evaluate the proposed method, we prepared six lifelog video clips by six subjects termed A, B, C, D, E, and F. All the subjects are male university students.

Algorithm 1 Emotional scene detection.

```
Notations:
  • $E_i^c$: The $i$-th emotional scene in which the facial
    expression $c$ appears.
  • $first(E_i^c)$: Frame number of the beginning frame in $E_i^c$.
  • $last(E_i^c)$: Frame number of the ending frame in $E_i^c$.
  • $length(E_i^c)$: Length of $E_i^c$. It is equivalent to
    $last(E_i^c) - first(E_i^c) - 1$.
  • $\#int(E_i^c)$: Number of emotional scenes integrated into
    $E_i^c$.
  • $\#nonemo(E_i^c)$: Number of nonemotional frames in $E_i^c$.
    Note that a nonemotional frame means that the facial
    expression appears in that frame is different from $c$.
  • $dist(E_i^c, E_j^c)$: The distance between $E_i^c$ and $E_j^c$ $(i < j)$.
    It is equivalent to $first(E_j^c) - last(E_i^c) - 1$.
Initialize:
    For each frame image having the class label $c$, perform the following initialization according
    to Equation (16):
        $$first(E_i^c) = last(E_i^c) = c_i,\ \#int(E_i^c) = 0,$$
        $$\#nonemo(E_i^c) = 0,\ length(E_i^c) = 1,\ (1 \leq i \leq M_c) \tag{16}$$
    where, $c_i$ is the frame number of the $i$-th emotional frame in the video. $M_c$ is the number
    of emotional frames. An emotional frame is the frame having the class label $c$. That is, each
    emotional scene consists of a single emotional frame.
Procedure:
  1: Find $i^*$ in accordance with Equation (17):
        $$i^* = \underset{i}{\operatorname{argmin}}\ dist(E_i^c, E_{i+1}^c)$$
        $$s.t.\ dist(E_i^c, E_{i+1}^c) \leq \frac{length(E_i^c) - \#nonemo(E_i^c)}{\#int(E_i^c) + 1}$$
        $$\wedge\ dist(E_i^c, E_{i+1}^c) \leq \frac{length(E_{i+1}^c) - \#nonemo(E_{i+1}^c)}{\#int(E_{i+1}^c) + 1} \tag{17}$$
  2: If there is no $i^*$ that satisfies Equation (17), finish the procedure and output current emotional
     scenes. Otherwise, proceed to step 3.
  3: Integrate $E_{i^*+1}^c$ into $E_{i^*}^c$ by updating $E_{i^*}^c$ as follows:
        $$last(E_{i^*}^c) \leftarrow last(E_{i^*+1}^c),\ \#int(E_{i^*}^c) \leftarrow \#int(E_{i^*}^c) + 1,$$
        $$\#nonemo(E_{i^*}^c) \leftarrow \#nonemo(E_{i^*}^c) + first(E_{i^*+1}^c) - last(E_{i^*}^c) - 1$$
     Note that $length(E_{i^*}^c)$ is also updated due to the update of $last(E_{i^*}^c)$.
  4: Delete $E_{i^*+1}^c$ and renumber the subscripts of $E_i^c$ so that the emotional scenes become
     $E_i^c, \cdots, E_{M_c-1}^c$.
  5: $M_c \leftarrow M_c - 1$ and return to step 1.
```

Table 1 Number of samples

Data set	#Samples
A	1585
B	2004
C	1616
D	1361
E	1730
F	1436

These video clips contain the scenes of playing cards recorded by web cameras. A single web camera recorded a single subject's frontal face. This experimental setting is due to the limitation of FaceSDK that it can detect the facial feature points of a single frontal face. While card games are suitable for stably recording frontal faces, a player of most of card games tries to keep a poker face. We thus chose the card games such as Hearts in which the players could clearly express the emotion.

The size of each video is 640 × 480 pixels and the frame rate is 30 frames per second. Considering the high frame rate, we selected frames from each video after every 10 frames in order to reduce the computational cost. The number of samples (i.e., frames) in each video clip is shown in Table 1. Note that the videos recorded are not shown in this paper because of privacy reasons.

The facial expressions observed in most of the emotional scenes in the video clips were smiles. Thus, we set the value of C (described in Sect. 3) to 2 intending to detect the emotional scenes with smiles, that is, to discriminate smiles and other facial expressions. The ratio of the emotional frames to all the frames varies from 16.6% to 29.6%. A subject is smiling in 24.6% of the frames in the video clip on average.

A two-fold cross validation was used in this experiment by dividing each video clip into the first and second halves. The one was used for the training and the other was used for the test.

6.1.2 Parameter Settings

In the process of the construction of the FER model, several parameters are required for training multilayer convolutional network. We experimentally determined the values of these parameters. The value of each parameter is shown in Table 2.

As described in Sect. 4, we used Adam algorithm for the parameter optimization. This is an iterative algorithm and the number of iterations should be determined. We set the value of iterations to 700 considering the result of a preliminary experiment.

In Adam algorithm, there are four parameters to be experimentally determined [11]. The exponential decay rates β_1 and β_2 are set to 0.9 and 0.999, respectively. Note that they have no relation to β_1 and β_2 described in Sect. 4. The learning rate α and the constant for stability ϵ are set to 10^{-4} and 10^{-8}, respectively.

Table 2 Parameters for FER

Parameter	Value
M	12
S	5
L_1	32
T	2
L_2	64
U	1024

6.2 Experimental Result

The recall, precision, and F-measure of the emotional scene detection are computed for the evaluation of the accuracy of the proposed method. The recall, precision, and F-measure are defined by Eqs. (18), (19), and (20), respectively.

$$recall = \frac{|T \cap \hat{T}|}{|T|} \tag{18}$$

$$precision = \frac{|T \cap \hat{T}|}{|\hat{T}|} \tag{19}$$

$$F{-}measure = \frac{2 \cdot recall \cdot precision}{recall + precision} \tag{20}$$

where, T is the correct set of emotional frames. One of the authors determined whether each frame was emotional or not prior to the experiment. $\hat{T}$ is the set of emotional frames detected by the proposed method.

We compared the scene detection accuracy of the proposed method with that of an existing method based on an evolutionary facial feature creation [4]. The number of facial features to be selected was set to six for the existing method because it was reported in [4] that six facial features were sufficient for this method. The recall, precision, and F-measure of the proposed method and the existing method are shown in Fig. 4.

7 Consideration

The proposed method outperformed the existing method for four subjects out of six ones in F-measure. In particular, Subject D's F-measure is improved well due to the large improvement in recall. The facial expression of Subject D is relatively weak compared with the other subjects. This makes the detection of his smile more difficult and leads to lower recall in the emotional scene retrieval. There are some people

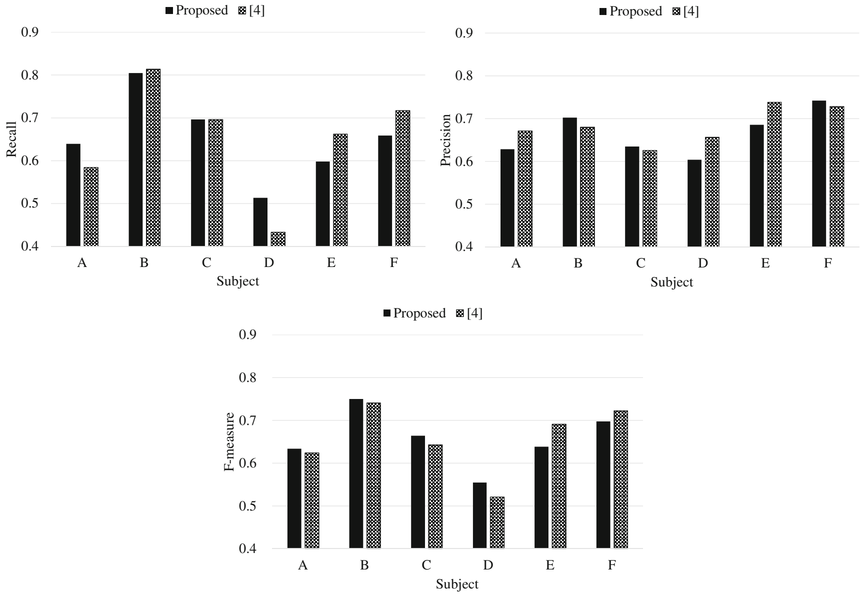

Fig. 4 Emotional scene detection accuracy (recall, precision, and F-measure)

who rarely express strong facial expressions. This result indicates that the proposed method could be suitable for them.

On the other hand, the F-measure of the proposed method for Subjects E and F is lower than that of the existing method. In the proposed method, there are many parameters to be experimentally determined as shown in Table 2, while the number of parameters in the existing method is relatively small. The accuracy for these subjects may be ameliorated by improving the parameter settings.

8 Conclusion

An emotional video scene retrieval method was proposed in this paper. The detection of emotional scenes is performed on the basis of FER. In order to accurately discriminate the facial expressions, we proposed an FER model constructed using several useful facial features and multilayer convolutional network.

The effectiveness of the proposed method was evaluated through an emotional scene detection experiment using some lifelog video clips. The detection accuracy of the proposed method was compared with an existing method. The detection accuracy (F-measure) of the proposed method for two-thirds of the subjects was higher than that of the existing method. The experimental result showed that the proposed method could be effective for the case that accurately detecting emotional scenes was relatively difficult due to weak facial expressions.

Since there are many parameters to be optimized in the proposed method, developing an effective parameter tuning method is required for the improvement of accuracy. This is included in the future work. In the experiment, we focus on only smiles. Evaluating the proposed method using a data set containing a wide variety of facial expressions is also the future work.

Acknowledgements This research is supported by Japan Society for the Promotion of Science, Grant-in-Aid for Young Scientists (B), 15K15993.

References

1. T. Datchakorn, T. Yamasaki, and K. Aizawa, "Practical Experience Recording and Indexing of Life Log Video," Proc. of the 2nd ACM Workshop on Continuous Archival and Retrieval of Personal Experiences, pp. 61–66, 2005.
2. D. Datcu and L. Rothkrantz, "Facial Expression Recognition in Still Pictures and Videos Using Active Appearance Models: A Comparison Approach," Proc. of the 2007 International Conference on Computer Systems and Technologies, pp. 1–6, 2007.
3. G. Fanelli, A. Yao, P.-L. Noel, J. Gall, and L. V. Gool, "Hough Forest-based Facial Expression Recognition from Video Sequences," Proc. of the 11th European Conference on Trends and Topics in Computer Vision, pp. 195–206, 2010.
4. H. Nomiya and T. Hochin, "Emotional Scene Retrieval from Lifelog Videos Using Evolutionary Feature Creation," Studies in Computational Intelligence, Vol. 612, pp. 61–75, 2015.
5. Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time Series," The Handbook of Brain Theory and Neural Networks, MIT Press, pp. 255–258, 1998.
6. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature Vol. 521, pp. 436–444, 2015.
7. T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive Database for Facial Expression Analysis," Proc. of the 4th IEEE International Conference on Automatic Face and Gesture Recognition, pp. 46–53, 2000.
8. Luxand Inc., Luxand FaceSDK 4.0, http://www.luxand.com/facesdk [September 11, 2016] (current version is 6.1).
9. H. Nomiya, S. Sakaue, and T. Hochin, "Recognition and Intensity Estimation of Facial Expression Using Ensemble Classifiers," Proc. of 15th International Conference on Computer and Information Science, pp. 825–830, 2016.
10. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research, Vol. 15, No. 1, pp. 1929–1958, 2014.
11. D. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," Proc. of the 3rd International Conference for Learning Representations, arXiv:1412.6980, 2015.