An Emotional Recognition System for Facial Expressions with Surface Common Features

Maierdan Maimaitimin, Keigo Watanabe, and Shoichi Maeyama
Department of Intelligent Mechanical Systems
Okayama University
Okayama, Japan
merdan-m@usmm.sys.okayama-u.ac.jp, {watanabe, maeyama}@sys.okayama-u.ac.jp

Received: 14 October 2016 / Revised: 20 November 2016 / Accepted: 14 December 2016 / Published online: 26 December 2016 © IJSMM2016

Abstract—In this paper, the recognition problem of facial expressions, which is based on 3D surface common features, is addressed by using a deep learning structure. A face in 3D is captured by a 3D sensor, where the raw data is provided in a point cloud structure. A geometric attribute map that is a surface common feature is obtained from such 3D point cloud data. Then, a set of maps are fed into a convolution neural network (CNN), which is pre-trained by an auto-encoder in previous work. The CNN is used to predict the activity and arousal parameters of each part on the face. At the last layer of the whole network, such parameters are used to predict the current facial expression. Note here that as the database, there are six different facial expressions such as angry, fear, happy, etc., captured from 30 peoples in 230 frames, and it is more than 40 thousand sets in total. As a result, the CNN with pre-training on surface common features outperforms the hand-crafted descriptors in the same experimental condition.

Index Terms—Surface common features, deep learning, facial expressions.

I. INTRODUCTION

The facial expressions are response to internal emotional states, intentions, or social communications of humans [1], [2]. Automatic analyzing and understanding the facial expressions is of increasing importance in AI research filed because it provides the emotions, health status and affect of humans. This information can be used for controlling the interaction in HRI. For instance, by analyzing the human emotions, an intelligent system is able to support the user by offering help. In the last few years, many research have been proposed to recognize the facial expressions. Ying-li et al. [3] proposed a facial expression analysis system based on both permanent facial features and transient facial features, and it achieved average recognition rates of 96 percent. Some other approaches employ geometric and appearance based features. For example, Littlewort et al. [4] used a filter bank consisting of 72 complex-valued Gabor filters of eight orientations and nine spatial frequencies to extract features used afterwards to detect the AUs using support vector machine classifiers.

However, human emotions are complicated, thus the emotional expressions are not easy interpretation and inference for the interaction. Since some emotional facial expressions have a very similar physical performance in 2D image data source, it can be a difficult task even for human to distinguish a difference between sad and anger in images.

For improving the recognition accuracy of emotional facial expressions, some 3D point cloud data are applied as the analyzing source. The 3D point cloud data are recognized as the geometric projection of the object in digital form, which are the most direct information captured from the real world, but being limited in a sense. These points in the point cloud space are independent of each other, so that those points are meaningless to a logical system. In our previous research, we developed a new method, the surface common feature (SCF) descriptor. The method can obtain the surface geometrical features, such as the “curvatures,” “edges,” “smooth plane,” and “rough plane,” directly from the point cloud data.

The SCF is applied on the point cloud data of human emotional facial expressions. Also a pre-trained convolutional autoencoder is applied as front-end network to predict the intermediate values, valence and arousal, from the SCF facial data. By implementing a back-end neural network, the intermediate values are used in this layer to classify the data to emotions. In this particular work, we have achieved average facial expression recognition rates of 93 percent [5].

Some relevant tools and related work are discussed in section 2. The current method, which includes facial data processing, deep learning structure, is discussed in sections 3. Finally, a comparative result is listed in section 4.

II. RELATED WORK

A. Surface Common Features

In previous research, we proposed a 3D surface analyzing method, i.e., surface common feature descriptor [5].

A surface condition can be expressed by its surface normal vectors \( \hat{n} \) [10]. In particular, two adjacent normal vectors are formed to express a minimum surface condition by calculating the distance vector \( \hat{D} \). The processing of point cloud data with \( m \) points is calculated as the following steps:
Calculate all point vectors \( \tilde{p}_i \). The point vector is started from the origin point O to each point \( p_i \), where \( i = m \).

- Each of two adjacent point vectors \( \tilde{p}_i \) and \( \tilde{p}_{i+1} \) are used to generate a tangent vector \( \tilde{T}_i \).
- A normal vector \( \tilde{n} \) is considered as the cross product of two adjacent tangent vectors \( \tilde{T}_i \) and \( \tilde{T}_{i+1} \).
- The difference between two adjacent normal vectors is defined as the distance vector \( D \), which can represent the bending level between the two estimated minimum surfaces by its size \( |D| \).

After this calculation, a point cloud data can be converted to a SCF map.

![Fig. 1 A simplified valence-arousal space in this case](image)

**B. Emotion Modeling**

The valence-arousal space (VAS) is used to classify the emotions by the valued labels, where the simplified version is shown in Fig. 1. This map discrete is created based on the psychological model of Russell's circumplex model [6]. This value labels of the classification model can lead to a robust emotional state representation that is continuous in principle.

![Fig. 2 A convoluting operation](image)

**C. Convolution**

The convolution of a dataset is produced by applying a \( N \times N \) matrix filter to the \( M \) dimensional dataset, and it produces a new matrix which will be of size \((N-m+1) \times (N-m+1)\). A filter is weight-matrix where \( N \) is an odd number, so that the matrix has a unique center. The convolution can be imagined as the result of moving a filter across the \( M \) dimensional dataset that replaces each data with some functions of its neighborhood. And the computation to the output is shown in the following equations:

\[
x^{'j}_{ij} = \sum_{i=0}^{m-1} w_{ab} y^{j-1}(i+a) \times (j+b)
\]

(1)

In a deep learning structure [13], convolutions are capable of transforming data in many useful and concrete ways, such as emphasizing edges and computing gradients of hue and value in image processing, as shown in Fig. 2. Because of the combination of subsampling layer, it is an elegant way of enforcing a sparse code required to deal with the over complete representations of convolutional architectures.

**D. Autoencoder**

An autoencoder is a well-known machine learning technique. Usually, it is used as a pre-training methods for the entire deep belief network. It can obtain the initial parameters of a neuron by inputting the feature itself. An autoencoder takes an input \( \{x_i\}_{i=1}^{m} \) and first maps it with an encoder to a hidden representation \( \{y_i\}_{i=1}^{m} \) through a deterministic mapping.

The improved stacked autoencoder [7], [8] is shown in Fig. 3, where the original data \( \{x_i\}_{i=1}^{m} \) are encoded as \( \{y_i\}_{i=1}^{m} \) by sigmoid function to the code layer, and it uses the same sigmoid function but transmitted parameters \( \{a, b\} \) to decode back to the original input. By using the gradient descent algorithm to minimize the reconstruction error, the most optimized parameters \( \{a, b\} \) can be obtained. To consider the data size in our case, the second encode layer is necessary. The output of the first code layer \( \{y_i\}_{i=1}^{m} \) is used as the input in the second code layer, \( \{v_i\}_{i=1}^{m} \) is obtained by the same way in the first code layer but with the another parameter set \( \{a, b\} \). By
III. CURRENT METHODS

A. SCF Training Data

The Cohn-Kanade [9] facial expression database is used as our teaching models to the actors, and the Kinect device is used to capture the real time SCF dataset. In this time, six basic emotional facial expressions are captured in 230 frames from 30 actors. To reduce the computing load, the face area is separated in six parts, “left eye,” “left cheek,” “left mouth corner,” “right eye,” “right cheek,” and “right mouth corner.”

B. Data Label

The literature descriptions of emotional facial expressions are encoded based on the VAS [6]. Such as the literature descriptions of eyes in emotion “Angry” is “bulging,” then the high level performance flag of “Arousal” is assigned as 1, and the eye line in emotion “Angry” is also described as “lowered,” then the low level performance flag of the “Valence” is assigned as 1. So the “angry” is encoded as \[\{1,0,0,0,0,1\}\], which is shown in Fig. 5, and others so on. In fact, this encoding system allows us to separate the performance values more than 10 levels, but as a simple task in this case, it is not necessary.

C. Front-end Convolution Autoencoder

As shown in Fig. 4, a front-end convolution autoencoder is assigned to predict the VAS values from the SCF data of each facial part. In the all of convolution layers, the convolution filters are pre-trained by the autoencoder [11, 12]. And only the first convolution layer is powered by the deep autoencoder. As shown in blue box in Fig. 4, they represent the output feature maps of each convolution layer. The visualized filters are shown in orange box. In this case, we have six front-end convolution autoencoder structures, depending on six facial parts.

D. Back-end Neural Network

To predict an emotion, the VAS values of six parts are fed to a back-propagation neural network. And the labels of the emotions are encoded simply, such as "Angry" = \(\{1, 0, 0, 0, 0, 0\}\); “Sad” = \(\{0, 1, 0, 0, 0, 0\}\); “Fear” = \(\{0, 0, 1, 0, 0, 0\}\); “Calmness” = \(\{0, 0, 0, 1, 0, 0\}\); “Happy” = \(\{0, 0, 0, 0, 1, 0\}\) and “Surprise” = \(\{0, 0, 0, 0, 0, 1\}\). In this network, we have...
36 neurons in the input layer, and 18 neurons in the first hidden layer and 6 neurons in the output layer.

IV. RESULT

The reconstruction results are shown in Fig. 6. The Fig. 7 is the confusion matrix of front-end network, i.e., it shows the cheek SCF data are not performing very well in the network. But the final predictions are still shown extremely well, as shown in Fig. 8.

V. CONCLUSION

As we have learned from the reconstruction results, the specialized convolution autoencoder performed very well in SCF facial data. From the confusion matrix we were able to assume that, because of the low activities of the cheek in human emotional facial expressions, it was not considered very valuable in the network. So even the intermediate value of “cheek” were not predicted very well in the front-end convolution autoencoder, but the final predictions of full network were still acceptable.

REFERENCES

[1] M. Pantic, A. Pentland, and A. Nijholt, “Human computing and machine understanding of human behavior: A survey,” in Proc. of Artificial Intelligence for Human Computing, 2007, pp. 47–71.
[2] JF. Cohn and P. Ekman, “Measuring facial action,” in Handbook of Nonverbal Behavior Research Methods in the Affective Sciences, New York, USA: Oxford Univ. Press, 2008, pp. 9–64.
[3] J. Whitehill, Z. Serpell, YC. Lin, A. Foster, and JR. Movellan, “The faces of engagement: Automatic recognition of student engagement from facial expressions,” IEEE Trans. on Affective Computing, vol. 5, no. 1, pp. 86–98, 2014.
[4] M. S. Bartlett, G. Littlewort, and M. Frank, “Recognizing facial expression: machine learning and application to spontaneous behavior,” In Proc. of Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 2, 2005, pp. 568–573.
[5] M. Maierdan, K. Watanabe, and S. Meyama. “Surface-common-feature descriptor of point cloud data for deep learning,” In Proc. of Int. Conf. on Mechatronics and Automation (ICMA), 2016 pp. 525–529.
[6] J. Posner, J. A. Russell, and B. S. Peterson, “The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology,” Developmental and Psychopathology, 2005.
[7] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” Journal of Machine Learning Research, vol. 11, no. 3, pp. 3371–3408, 2010.
[8] J. Masci, U. Meier, D. Cirean, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in Proc. of Artificial Neural Networks and Machine Learning–ICANN 2011, vol. 6791 LNCS, no. PART 1, pp. 52–59, 2011.
[9] T. Kanade, J. F. Cohn, and Y. L. Tian, “Comprehensive database for facial expression analysis,” in Proc. of Int. Conf. on Automatic Face and Gesture Recognition, 2000, pp. 46–53.
[10] D. OuYang and H.Y. Feng, “On the normal vector estimation for point cloud data from smooth surfaces,” CAD Computer-Aided Design, vol. 37, no. 10, pp. 1071–1079, 2005.
[11] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface reconstruction from unorganized points,” ACM SIGGRAPH Computer Graphics, vol. 26, no. 2, pp. 71–78, 1992.
[12] Y. Bengio, “Learning deep architectures for AI,” Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.
[13] P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” Unsupervised and Transfer Learning Challenges in Machine Learning, vol. 7, pp. 37–50, 2012.