Recognition of Social Touch Gestures Using 3D Convolutional Neural Networks

Nan Zhou and Jun Du(✉)

University of Science and Technology of China, Hefei, Anhui,
People's Republic of China
francis7999@outlook.com, jundu@ustc.edu.cn

Abstract. This paper investigates on the deep learning approaches for the social touch gesture recognition. Several types of neural network architectures are studied with a comprehensive experiment design. First, recurrent neural network using long short-term memory (LSTM) is adopted for modeling the gesture sequence. However, for both handcrafted features using geometric moment and feature extraction using convolutional neural network (CNN), LSTM cannot achieve satisfactory performances. Therefore, we propose to use the 3D CNN to model a fixed length of touch gesture sequence. Experimental results show that the 3D CNN approach can achieve a recognition accuracy of 76.1 % on the human-animal affective robot touch (HAART) database in the recognition of social touch gestures challenge 2015, which significantly outperforms the best submitted system of the challenge with a recognition accuracy of 70.9 %.

Keywords: Deep learning · Social touch gesture · 3D CNN

1 Introduction

In recent years there has been an increasing interest on human-robot interaction studies that use touch modality. In social human-robot interaction, the correct interpretation of touch gestures provides additional information about affective contents in touch, and can be used together with audio-visual cues to improve affect recognition performance [1]. Some well-known robots, such as AIBO (1999), Paro (2001), Nao (2006) and Reeti (2011) are equipped with touch sensors. Some researchers have investigated skin-like sensing, i.e. lots of sensors spreading all over the robot body [2–4].

In the international conference on multimodal interaction (ICMI) last year, the Recognition of Social Touch Gestures Challenge 2015 was launched. In this challenge, organizers provided participants with pressure sensor grid datasets of various touch gestures, namely Corpus of Social Touch (CoST) database and Human-Animal Affective Robot Touch (HAART) database [5]. Many conventional classification approaches were adopted, including support vector machine (SVM) [6], logistic regression [7], random forest [6, 8, 9], and multiboost [8].

Recently, the deep learning techniques are successfully used in many research areas. Convolutional neural networks (CNNs) [10] are one type of feedforward neural networks, which make promising results especially for the computer vision area.

T. Tan et al. (Eds.): CCPR 2016, Part I, CCIS 662, pp. 164–173, 2016.
DOI: 10.1007/978-981-10-3002-4_14

CNN receives raw images as the inputs, uses trainable kernels to extract features and pooling layers to down-sample feature maps, and makes output feature maps highly invariant to specific input transformation. Researchers have found that with appropriate parameters and regularization terms, CNNs can outperform methods with manually extracted features [11–13]. Recurrent neural networks are another type of deep neural networks with directed circles between units which make it "deep in time". RNNs can model the temporal actions by changing their outputs through time. But this simple RNN is suffering from problems like gradient vanishing and explosion, easy to lose the track of long term connections [14]. However, by using RNN with the long short-term memory (LSTM) structure, the gradient vanishing problem can be alleviated [15].

In this study, we investigate on the deep learning approaches for the social touch gesture recognition which are rarely mentioned for 2015 challenge. Several types of neural network architectures are studied with a comprehensive experiment design. First, recurrent neural network using long short-term memory (LSTM) is adopted for modeling the gesture sequence. However, for both handcrafted features using geometric moment and feature extraction using CNN, LSTM cannot achieve satisfactory performances. Therefore, we propose to use the 3D CNN to model a fixed length of touch gesture sequence. Experimental results show that the 3D CNN approach can achieve a recognition accuracy of 76.1 % on the human-animal affective robot touch (HAART) database in the recognition of social touch gestures challenge 2015, which significantly outperforms the best submitted system of the challenge with a recognition accuracy of 70.9 %.

The remainder of the paper is organized as follows. In Sect. 2, LSTM with geometric moment features (denoted as GM-LSTM) is introduced. In Sect. 3, LSTM with CNN-based feature extraction (denoted as LRCN) is presented. In Sect. 4, 3D CNN is elaborated. In Sects. 5 and 6, we report experimental results and analysis. Finally we conclude the paper in Sect. 7.

2 GM-LSTM

Geometric moments represent geometric features of an image and are invariant to rotation, transition and scaling, which are also called invariant moments [16]. In image processing, geometric moments can be used as important features to represent objects. The zeroth-order moment, $\mu_{0,0}$ and the first-order moments, $\mu_{1,0}$ and $\mu_{0,1}$ are given by

$$\mu_{0,0} = \sum_{x=0}^{w} \sum_{y=0}^{h} I(x, y) \tag{1}$$

$$\mu_{1,0} = \frac{1}{\mu_{0,0}} \sum_{x=0}^{w} \sum_{y=0}^{h} xI(x, y) \tag{2}$$

$$\mu_{0,1} = \frac{1}{\mu_{0,0}} \sum_{x=0}^{w} \sum_{y=0}^{h} yI(x, y) \tag{3}$$

where w and h represent the width and height of an image, respectively. $I(x, y)$ is the intensity of pressure at (x, y). Higher-order moments are calculated from the following equation

$$\mu_{i,j} = \frac{1}{\mu_{0,0}} \sum_{x=0}^{w} \sum_{y=0}^{h} (x - \mu_{1,0})^i (y - \mu_{0,1})^j I(x, y) \tag{4}$$

In our implementation, the zeroth, first, second and third order moments for each frame are calculated, resulting in a 10-dimension feature vector.

In GM-LSTM architecture, the input layer receives a sequence (432 frames for each sample) of GM feature vectors while the output layer has 7 units, each corresponding to a type of gesture labels. The activation function of output layer is the softmax function, and the loss function is cross-entropy error function. The hidden layers are represented by the LSTM layers as shown in Fig. 1.

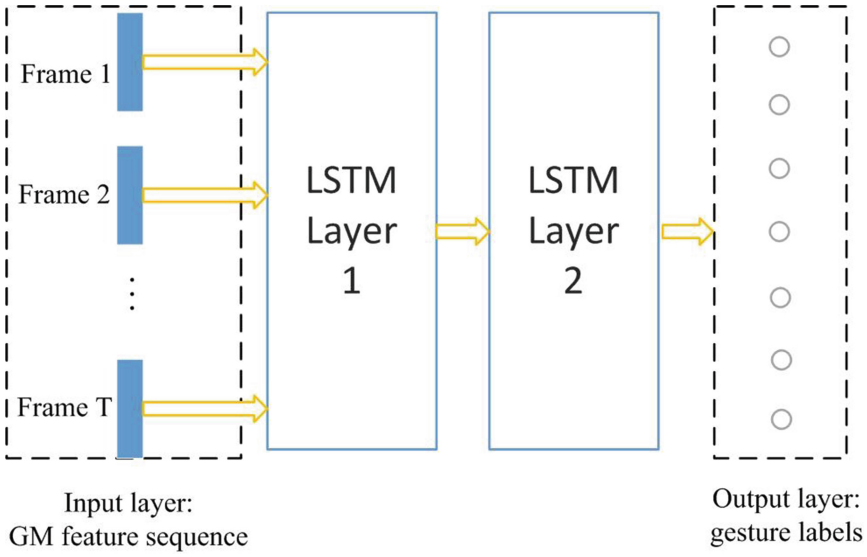

Fig. 1. A diagram of GM-LSTM architecture.

3 LRCN

Donahue et al. proposed long-term recurrent convolutional networks (LRCNs) to deal with problems like activity detection, image captioning and video captioning in 2014. It performed well for many datasets, becoming a type of leading deep learning methods [17]. LRCNs are one kind of deep neural networks that can make end-to-end classification of videos. It uses CNN to extract hierarchical features for each frame, and LSTM to model the feature sequence. Then the parameters of the whole network are updated using backpropagation through time (BPTT) algorithm.

As a combination of CNN and LSTM, LRCN has its own architecture including the input layer, several convolutional layers and pooling layers for CNN, several LSTM layers, and the output layer, as shown in Fig. 2. For our task, the input layer receives

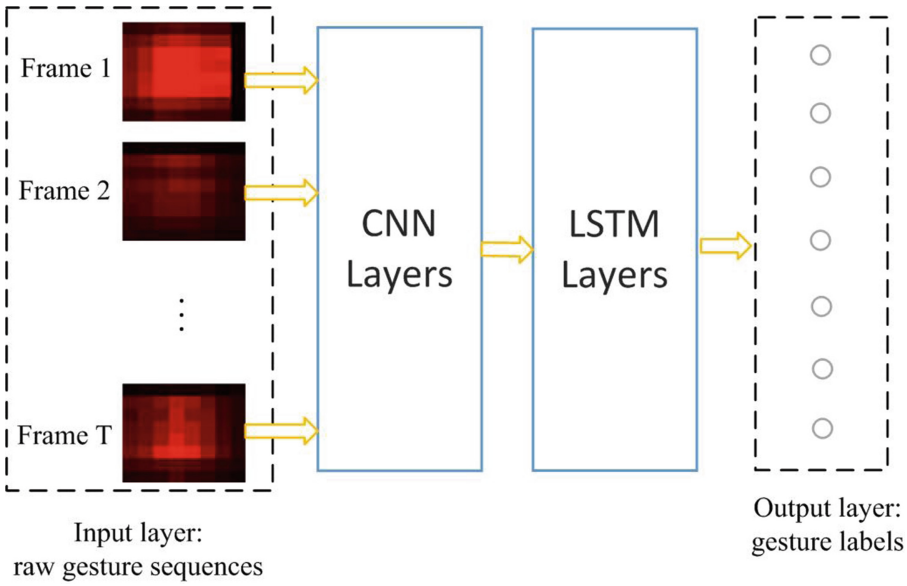

Fig. 2. A diagram of LRCN architecture.

raw input of gesture samples, namely a 8 × 8 image for each frame with the pixel value ranging from 0 to 1023. The output layer has 7 units, each corresponding to a type of gesture labels. The activation function of output layer is the softmax function, and the loss function is cross entropy error function.

4 3D CNNs

Ji et al. proposed 3D convolutional neural networks to tackle the problem like human action detection in videos and achieved excellent performance, indicating the superiority of 3D CNNs compared with other approaches [18].

In 2D CNNs, convolutions are applied on the 2D feature maps to compute features from the spatial dimensions only. When applied to video analysis problems, it is desirable to capture the motion information encoded in multiple contiguous frames. To this end, the proposed 3D CNNs can simultaneously compute features from both spatial and temporal dimensions. By the use of 3D CNNs, contiguous frames in a gesture sequence are first stacked up, reshaped into a 3D cube. Then 3D convolutions with 3D kernels are applied on the cube. That's why feature maps of 3D CNN are relevant to both spatial and temporal information, and can capture motion information. The value at position (x, y, z) of the j-th feature map in the i-th layer is given by

$$v_{ij}^{xyz} = \tanh\left(b_{ij} + \sum_m \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \quad (5)$$

where $\tanh(\cdot)$ is the activation function, b_{ij} is the value at position (i, j) of the bias vector, w_{ijm}^{pqr} is the value at (p, q, r) of the m-th convolution kernel, P_i and Q_i are the height and width of the kernel, and R_i is the temporal length of the kernel.

The overall architecture of 3D CNNs is shown in Fig. 3. Apart from the 3D convolutional layers and max pooling layers, another main difference from the GM-LSTM and LRCN is the fully connected layers are adopted before the output layer.

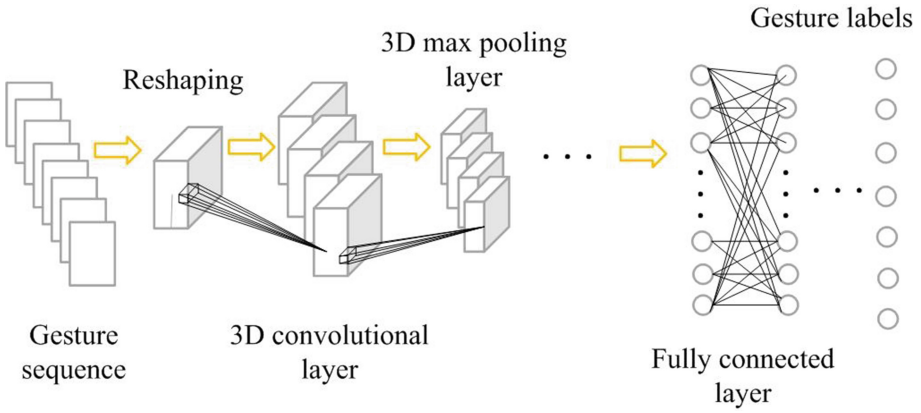

Fig. 3. A diagram of 3D CNN architecture.

5 Experiments and Results

Our experiments are conducted on the HAART dataset of the recognition of social touch gestures challenge 2015. The main purpose of HAART dataset design is to find methods of recognizing human emotion by gestures. The sampling rate of HAART dataset is 54 Hz and the time duration is 8 s. The number of participants is 10. The number of gestures types is 7. The size of sensor grid is 10×10, which was trimmed to 8×8 to match that of CoST dataset in the challenge [5]. Other details can refer to [5].

5.1 GM-LSTM and LRCN Experiments

In the GM-LSTM experiments, two LSTM layers with 50 units for each layer are good configurations to balance the model capability and the over-fitting problem. After 500 epochs of training, the test set accuracy is 65.3 %.

In the LRCN experiments, we can achieve a fair performance when using three convolutional layers and two 128-unit LSTM layers. The first convolutional layer has 4 feature maps, and its kernel size is 3×3. The second convolutional layer has 8 feature maps, its kernel size is 3×3, and the second convolutional layer is followed by an average pooling layer. The third convolutional layer has 16 feature maps, and its kernel size is 2×2. The test set accuracy at the 500-th epoch is 60.6 % (Table 1).

Table 1. The accuracy of test set at the 500-th epoch in GM-LSTM and LRCN experiments.

Classifier	Accuracy (%)
GM-LSTM	65.3
LRCN	60.6

5.2 3D CNN Experiments

The input layer of 3D CNN receives a raw gesture sequence with 432 frames. For each frame, the size of the image is 8 × 8 with the pixel value ranging from 0 to 1023. The output layer has 7 units, each corresponding to a type of gesture label. The activation function of output layer is the softmax function, and the loss function is cross entropy error function. The number of training epochs is 600.

5.2.1 Experiments on the Number of Feature Maps

The number of feature maps in every convolutional layer determines the dimension of features extracted and is important for recognition performance. In this set of experiments, the number of convolutional layers was fixed to 4, and we found that when the number configuration of feature maps for 4 convolutional layers was set as 16-32-64-128, 3D CNNs achieved the best performance, compared with 8-16-32-64 and 32-64-128-256 setting, as shown in Table 2. The kernel sizes in every convolutional layer were all set to be 3 × 3×3.

Table 2. The accuracy of test set in feature map configuration experiments

Feature map configuration	Accuracy (%)
8-16-32-64	72.9
16-32-64-128	75.1
32-64-128-256	68.1

Table 3. The accuracy of test set for different numbers of fully connected layers.

The number of fully connected layers	Accuracy (%)
1	66.9
2	71.3
3	72.5
4	70.9

5.2.2 Experiments on the Number of Fully Connected Layers

The number of fully connected layers determines the complexity and generalization capability of 3D CNNs. The optimal number can vary substantially in different tasks. In this set of experiments, the number of units in every fully connected layer was 256 (Table 3).

Table 4. The accuracy of test set in the dropout value experiments

Dropout value	Accuracy (%)
0(no dropout)	68.9
0.2	64.5
0.5	71.3

5.2.3 Experiments on the Dropout

When training deep neural networks, if the dataset is not large enough, we should use dropout as a trick to prevent over-fitting. Dropout was proposed by Hinton et al. in 2012. It randomly screens some weights of the units in hidden layers and improves neural networks by preventing co-adaptation of feature detectors [19]. Our experiments achieved a relatively good performance when the dropout value was 0.5, compared with no dropout case and the dropout value of 0.2. In these experiments, the number configuration of feature maps for 4 convolutional layers was set as 8-16-32-64, and the number of units in every fully connected layer was 256 and the number of fully connected layers was 3 (Table 4).

5.3 Overall Comparison

The final configuration of 3D CNN is as follows. The number of convolutional layers is 4. The number configuration of feature maps for 4 convolutional layers is 16-32-64-128. The kernel sizes in every convolutional layer are all set to be $3 \times 3 \times 3$. The pooling sizes in four convolutional layers are $5 \times 1 \times 1$, $3 \times 2 \times 3$, $3 \times 2 \times 2$, $3 \times 2 \times 2$, respectively. The number of fully connected layers is 3. The number of units in the fully connected layers is 1024. The dropout value is 0.5 and learning rate is 0.001. The initialization method is he_normal [20], and the batch size is 20.

With this configuration, we achieved the best test set accuracy of 76.1 %, which is significantly better than the first ranked result (70.9 % in [6]) in the challenge. And 3D CNN becomes the state-of-art method for social touch gestures recognition task. Table 5 shows an overall performance comparison with other approaches.

Table 5. An overall performance comparison with other approaches.

Approach	Classifier	Accuracy (%)
Proposed	3D CNN	**76.1**
Proposed	GM-LSTM	65.3
Proposed	LRCN	60.6
[6]	Random forest	70.9
[6]	SVM	68.5
[7]	Logistic regression	67.7
[8]	Random forest	66.5
[8]	Multiboost	64.5
[9]	Random forest	61.0

6 Discussion

GM-LSTMs and LRCNs did not perform well on the touch gesture recognition task. It might be partially due to that there were not enough training data for building LSTM layers, which usually needed longer sequence data compared with the fullly connected layers. And LRCNs even performed worse than GM-LSTMs, indicating that the use of CNN for feature extraction is not always helpful if the sequence model (LSTM) is not well enough.

3D CNNs did perform well on the touch gesture recognition task, which implied that 3D CNNs were really good at extracting motion information from contiguous frames, even on a small data set compared with other approaches.

As we can see in Table 6, two types of gestures—constant and no touch—have better accuracies with few false recognitions. The rub gesture is similar to scratch and they are easily incorrectly classified to each other. The stroke is similar to rub. The tickle has the lowest accuracy and is often incorrectly classified to scratch. All those problems are apparently related with the way the gesture samples were collected and the interconnections between gestures. And we can make more studies in the future.

Table 6. The confusion matrix for HARRT dataset

Counts	Constant	No touch	Pat	Rub	Scratch	Stroke	Tickle
Constant	34	1	0	0	0	0	0
No touch	0	36	0	0	0	0	0
Pat	0	0	29	2	2	1	2
Rub	0	0	0	27	2	5	2
Scratch	0	0	1	6	26	1	2
Stroke	0	0	1	5	1	27	2
Tickle	0	0	0	2	20	2	12

7 Conclusion

This paper analyzed different neural network structures including 2D CNNs, 3D CNNs and LSTMs. GM-LSTMs, LRCNs and 3D CNNs are compared on the social touch gestures recognition task. The conclusion is that the 3D CNN approach achieves the best result, beyond the first ranked result in the challenge—70.9 %, and 3D CNN is the state-of-art method for touch gestures classification task. In comparing 3D CNN experiments with GM-LSTM and LRCN experiments, we observe that LSTMs require larger dataset and 3D CNNs are more robust to the data size. Also we analyzed the similarities of gestures based on confusion matrix.

In the future, on one hand, we will further verify the effectiveness on large gesture databases. On the other hand, we will apply 3D CNNs to other tasks involving temporal information.

References

1. Balli Altuglu, T., Altun, K.: Recognizing touch gestures for social human-robot interaction. In: Proceedings of the 2015 ACM International Conference on Multimodal Interaction (ICMI), pp. 407–413 (2015)
2. Cooney, M.D., Nishio, S., Ishiguro, H.: Recognizing affection for a touch-based interaction with a humanoid robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1420–1427 (2012)
3. Billard, A., Bonfiglio, A., Cannata, G., Cosseddu, P., Dahl, T., Dautenhahn, K., Mastrogiovanni, F., Metta, G., Natale, L., Robins, B., et al.: The ROBOSKIN project: challenges and results. In: Padois, V., Bidaud, P., Khatib, O. (eds.) Romansy 19–Robot Design, Dynamics and Control. CISM International Centre for Mechanical Sciences, pp. 351–358. Springer, Vienna (2013)
4. Knight, H., Toscano, R., Stiehl, W.D., Chang, A., Wang, Y., Breazeal, C.: Real-time social touch gesture recognition for sensate robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3715–3720 (2009)
5. Jung, M.M., Cang, X.L., Poel, M., MacLean, K.E.: Touch challenge '15: recognizing social touch gestures. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICMI), pp. 387–390 (2015)
6. Ta, V.-C., Johal, W., Portaz, M., Castelli, E., Vaufreydaz, D.: The Grenoble system for the social touch challenge at ICMI 2015. In: Proceedings of the 2015 ACM International Conference on Multimodal Interaction (ICMI), pp. 391–398 (2015)
7. Hughes, D., Farrow, N., Profita, H., Correll, N.: Detecting and identifying tactile gestures using deep autoencoders, geometric moments and gesture level features. In: Proceedings of the 2015 ACM International Conference on Multimodal Interaction (ICMI), pp. 415–422 (2015)
8. Falinie, Y. Gaus, A., Olugbade, T., Jan, A., Qin, R., Liu, J., Zhang, F., Meng, H., Bianchi-Berthouze, N.: Social touch gesture recognition using random forest and boosting on distinct feature sets. In: Proceedings of the 2015 ACM International Conference on Multimodal Interaction (ICMI), pp. 399–406 (2015)
9. Altuglu, T.B., Altun, K.: Recognizing touch gestures for social human-robot interaction. In: Proceedings of the 2015 ACM International Conference on Multimodal Interaction (ICMI), pp. 407–413 (2015)
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
11. Yu, K., Xu, W., Gong, Y.: Deep learning with kernel regularization for visual recognition. In: NIPS, pp. 1889–1896 (2008)
12. Ahmed, A., Yu, K., Xu, W., Gong, Y., Xing, E.: Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5305, pp. 69–82. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88690-7_6
13. Mobahi, H., Collobert, R., Weston, J.: Deep learning from temporal coherence in video. In: ICML, pp. 737–744 (2009)
14. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
16. Teh, C.-H., Chin, R.T.: On image analysis by the methods of moments. IEEE Trans. Pattern Anal. Mach. Intell. **10**(4), 496–513 (1988)

17. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description (2014). CoRR, abs/1411.4389
18. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013)
19. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors (2012). CoRR, abs/1207.0580
20. He, K., et al.: Delving deep into rectifiers: surpassing human-level performance on imagenet classification. arXiv preprint arXiv:1502.01852 (2015)