Automatic Feature Extraction for Social Touch Classification

Songha Ban[O] Ngoc Doan Ruben Kole Michon Zeegers Gökçe Kuscu
Department of Cognitive Science and Artificial Intelligence, Tilburg University
s.ban@uvt.nl, n.t.n.doan@uvt.nl, r.koe@uvt.nl, m.j.zeegers@uvt.nl, g.kusku@uvt.nl

Abstract

Social touch is an important interaction in both human development and the field of human-robot interaction. In this paper we will examine whether it is possible to automatically extract significant features from the gesture data of the Corpus of Social Touch (CoST), which should eventually improve the accuracy of social touch classification. We experimented with three different feature extraction methods: Convolutional LSTM Autoencoder, Principal Components Analysis (PCA), and Sparse Random Projection (SRP). By evaluating the classification performance with a CNN model, SRP turned out to extract the best features by recording the training accuracy of 92.5% and the test accuracy of 61.88%.

1. Introduction

Social touch has a significant influence on human development [1], and it plays an increasing role in the field of human-robot interaction [2]. As previous studies [3][4] show the impacts of touch interactions on human's emotions and attitudes [3][4], robots being capable of interacting with humans through social touch will contribute to the development of human-computer interaction.

To build a foundation for this, one of the most important tasks is to recognize social touch gestures and to interpret them. In their introduction of the Corpus of Social Touch (CoST), Jung et al. [5] conducted experiments to classify gestures, which resulted in the highest test accuracy of 60%. Prior to the experiments, 54 features including mean pressure, contact area, traveled distance, and duration were extracted using handpicked feature engineering. The criteria on which these features were deemed significant, however, appear rather arbitrary. It is highly likely that invaluable information that can differentiate one gesture from another has been accidentally discarded during this process.

Hence, the goal of this study is to examine the scenarios where machines automatically extract features from the raw gesture data. We hypothesize that automatic feature learning can minimize the errors in humans' manual feature engineering, which are mostly because of the lack of relevant domain expertise. Moreover, once automatic feature extraction becomes a reality, it will certainly speed up the workflow to classify gestures and make applications to recognize social touch in real-time a possibility.

2. Methodology

2.1. The Data Set

The data set we used is the Corpus of Social Touch (CoST), which was introduced by [6]. This corpus contains 7805 gesture captures of 14 different social touch gestures, which were performed either gently, normally or roughly on an 8 x 8 pressure sensor grid attached to a plastic arm of a mannequin. The number of frames of each gesture capture varies in the range of 10 to 1747, and each frame consists of 64 channels ranging between 0 and 1023.

2.2. Preprocessing

Every sample was padded with zeros to have 1747 frames, which was the maximum number of frames among the entire data. This reshaped the data into the same size, which makes it easier to train machine learning models with existing libraries. In addition, feature extraction algorithms can obtain complex temporal and spatial information such as patterns of pressure over time given the whole sequences of pressure data.

Before zero-padded, the whole data matrix was incremented by 1. This put the pressure values in the range of 1 to 1024, which distinguish them from empty values that were padded as 0s. Then we divided all values by 1024 so that they are on a scale between 0 and 1. Each sample was of size 1747 x 64. Yet the data was very sparse because the average length of gesture capture from the original data was 191.78. A majority of frames (around 1200) in most samples were zeros. In order to create classification labels, we ignored the variants and used one-hot encoding [7] to embed the gesture labels into vectors where the size of each label is 14. Prior to all experiments, the data set was split into a training set (70%), a validation set (10%), and a test set (20%) as commonly used in practice.

2.3. Feature Extraction

The goal of this study is to let the machine learn meaningful features from raw data to improve accuracy of touch classification. We tried following three different models to reduce dimension and extract latent features: Convolutional LSTM Autoencoder (ConvLSTM-AE), Principal Component Analysis (PCA), and Sparse Random Projection (SRP).

2.3.1. ConvLSTM-AE

ConvLSTM shows good performance in capturing spatio-temporal correlations [8]. We assumed that taking an approach similar to video processing [9] would work for gesture capture data, and therefore started with a simple autoencoder with ConvLSTM layers to extract latent representations of the data. Encoding part of the network consists of two ConvLSTM layers: one with 16 filters of size 3 x 3 returning a sequence, and the other with 32 filters of size 3 x 3 returning a single frame. Latent representation of size 8 x 8 x 32 is repeated 1747 times before decoding. A decoder consists of a ConvLSTM layer with

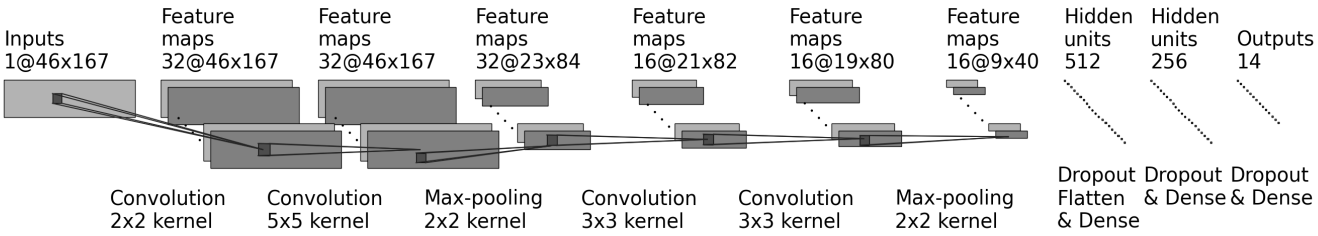

Figure 1. CNN classification model

16 filters of size 3 x 3 and one with 1 filter to generate the input data. The model had 93,640 trainable parameters, and used AdaDelta for an optimizer with learning rate 1.0 and loss function of mean squared error.

2.3.2. PCA

Principal component analysis (PCA) is usually used for dimension reduction by extracting the dominant patterns in the data matrix [10]. In this study, as the average length of the original samples was 192, we decided the number of components as 6400 to reduce the dimension around half as small (100 x 64). The output of PCA was then used as an input for a CNN classifier, which is described further in the classification section, to evaluate the performance.

2.3.3. SRP

According to the Johnson-Lindenstrauss lemma, any high dimensional data can be converted into a lower dimensional space through random projection [11]. We used Sparse Random Projection (SRP) implemented in Scikit-learn library [12] to extract a dense matrix from the sparse preprocessed data. The number of components, which is the dimensionality of the target projection space, was adjusted by the number of samples and the bound given by the Johnson-Lindenstrauss lemma as in (1), where ε refers to the squared distances ratio distortion. Density, which is ratio of non-zero component in the random projection matrix was calculated by 1/sqrt(number of features) as recommended by Li et al. [13]. The output of SRP (size 7682) was put into the CNN classifier to compare the results with other feature extraction methods.

$$n\ components \geq \frac{4\log(n\ samples)}{\varepsilon^2 - \varepsilon^3} \quad (1)$$

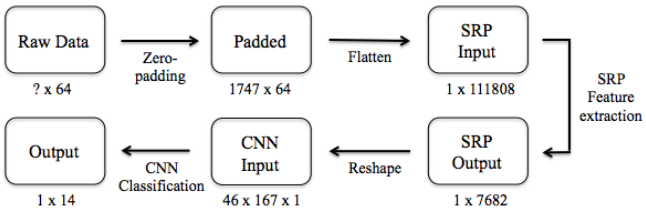

Figure 2. Overall architecture of SRP-CNN

2.4. Classification

Performance of classification shows how well the features reflect characteristics of the data. In order to evaluate the feature extractors introduced in the previous section, we built a baseline model using Convolutional neural network (CNN) [14] to classify the data into 14 different social gesture classes. First, we trained 50 to 100 epochs to see the loss trend, and for the actual classifier for better accuracy, we trained 250 epochs. The model architecture can be found in Figure 1, which was determined after several attempts with different filters and layers. We dropped out 25% of the units after the last convolution layer and 50% of the units after each dense layer. The number of trainable parameters was 3,117,294, and AdaDelta algorithm was used as an optimizer with learning rate 1.0. The batch size was 64, and the loss function was categorical cross entropy as following.

$$L(y, \hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}\left(y_{ij} * \log\left(\hat{y}_{ij}\right)\right) \quad (2)$$

Table 1. Results of feature extractions

	ConvLSTM-AE	PCA	SRP
Output size	2048	6400	7682
Training hours	20	4	2
Training accuracy (%)	-	98.45	92.50
Test accuracy (%)	-	29.08	61.88

3. Results

ConvLSTM-AE was not able to finish the training successfully with very slow decrease in loss only until 10 iterations and no changes after that. In addition, training took too much time (2 hours per iteration with GeForce GTX 1080 graphic cards). These happened because the size of input and output data was too sparse and big. Considering the time and cost efficiency and performance, we decided to move on to the other algorithms that are more efficient and are able to project the data well enough.

PCA took 4 hours to fit and transform each sample into 6400 components. For the classification task, the model was over-fitted as test accuracy stayed at 29.08% while the training accuracy went up to 98.45%.

SRP showed the best performance and time efficiency. It took 2 hours to fit, and the combination with CNN was also better than PCA with the test accuracy of 61.88% (M=62%, SD=13%).

For SRP-CNN model, we computed accuracies per variant on test set, which resulted in 58% for gentle gestures, 68% for normal gestures, and 60% for rough gestures. A confusion matrix for this model can be found in Figure 3, and the overall architecture is provided in Figure 2.

4. Discussion

SRP-CNN model provided the best result on the test set with the accuracy of 61.88%. Different accuracies upon variants show that the model classifies normal gestures more accurately than gentle or rough gestures. Based on Table 2 and a confusion matrix in Figure 3, we were able to find out that gentle and rough gestures are more likely to be misrecognized as other similar gestures. For example, rough grab can be misclassified

as squeeze, and rough rub and gentle scratch can be misclassified as one another. These gestures have similarities in nature, which are also confusing to humans, and they take a large portion of the misclassified cases. We can infer from the result that features extracted from SRP are containing enough information similar to human perception to recognize gestures up to certain level.

Table 2 compares the result from our model with five other models experimented in the previous study by Jung et al. [5]. For rough gesture, SVM models [5] tend to perform better than our model by 2%. Nevertheless, our model produced higher overall accuracy, and for gentle and normal gesture, it performed better with remarkable differences. Normal gestures usually represent each gesture more clearly than other variants, and better performance with a big gap on normal gesture means that the quality of features is good enough even when compared with the handpicked features from [5]. From this result, we could conclude that automatic feature extraction through SRP can lead to better performance than manual feature engineering.

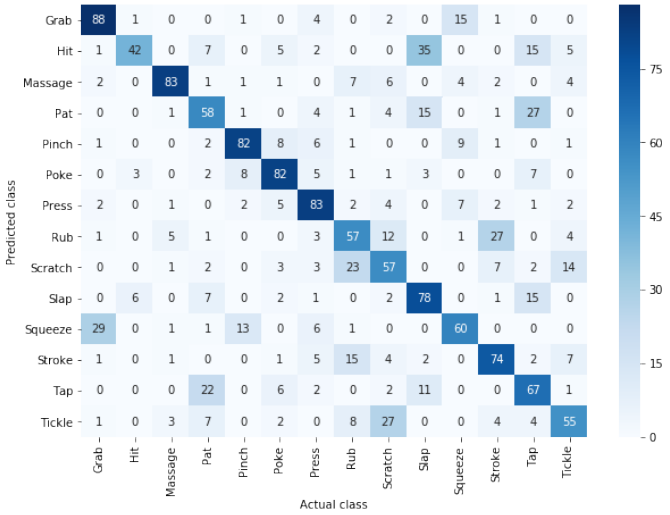

Figure 3. Confusion matrix of the result from SRP-CNN on the test set

Table 2. Accuracies per variant from SRP-CNN and models from study [5]

	All	Gentle	Normal	Rough
SRP-CNN	**.62**	**.58**	**.68**	.60
Bayesian	.57	.52	.59	.58
Decision tree	.48	.43	.49	.52
SVM linear	.59	.54	.60	**.62**
SVM RBF	.60	.54	.60	**.62**
Neural Network	.59	.52	.58	.59

5. Conclusion

The experiment conducted in this study shows that SRP can automatically extract meaningful features from raw CoST data for better classification of social touch gestures. This not only improves efficiency by skipping manual feature extraction, but also opens up the potential to be applied to a variety of topics, which makes machine learning more approachable. If more studies are conducted in the future to improve classification and even interpretation of the gestures, the spectrum of robots to interact with humans will be much broader.

References

[1] C. J. Cascio, D. Moore, and F. McGlone, "Social touch and human development," *Developmental Cognitive Neuroscience*, vol. 35, pp. 5–11, 2019.

[2] T. Hirano *et al.*, "Communication cues in a human-robot touch interaction," 2016, pp. 201–206.

[3] H. Cramer, N. Kemper, A. Amin, B. Wielinga, and V. Evers, "'Give me a hug': The effects of touch and autonomy on people's responses to embodied social agents," *Journal of Visualization and Computer Animation*, vol. 20, pp. 437–445, 2009.

[4] S. J. Yohanan, "The Haptic Creature : social human-robot interaction through affective touch," 2012.

[5] M. M. Jung, Mannes Poel, R. W. Poppe, and D. K. J. Heylen, "Automatic recognition of touch gestures in the corpus of social touch," *Journal on multimodal user interfaces*, vol. 11, pp. 81–96, 2017.

[6] M. M. Jung, R. Poppe, M. Poel, and D. K. J. Heylen, "Touching the void – introducing CoST: Corpus of social touch," pp. 120–127, 2014.

[7] Pau Rodríguez, M. A. Bautista, Jordi Gonzàlez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image and Vision Computing*, vol. 75, pp. 21–31, 2018.

[8] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. WOO, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., pp. 802–810, 2015.

[9] Viorica Patraucean, Ankur Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," 2015.

[10] Svante Wold,K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, pp. 37–52, 1987.

[11] Dimitris Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, pp. 671–687, 2003.

[12] Lars Buitinck *et al.*, "API design for machine learning software: experiences from the scikit-learn project," pp. 108–122, 2013.

[13] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," pp. 287–296, 2006.

[14] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," MIT Press, pp. 255–258, 1998.