Social Touch Recognition Based on Support Vector Machine and T-Distributed Stochastic Neighbour Embedding as Pre-processing

Ahmed Khudhair Abbas, Adil Ibrahim Khalil and Sameera A. Abdulkader
Collage of education for pure science, Diyala University, Diyala, Iraq
E-mail: Dr.ahmed.k.abbas@uodiyala.edu.iq

Abstract. Today, one of important field is social touch gesture recognition for touch modality, which can lead to highly efficient and realistic human–robot interaction. Touch is an important interaction modality in social interaction, for instance touch can communicate emotions and can intensify emotions communicated by other modalities. In this paper, the touch gesture recognition is performed using a dataset that previously measured for numerous subjects that perform various social gestures. This dataset is dubbed as the corpus of social touch (CoST), where touches were performed on a mannequin arm. The T-Distributed Stochastic Neighbor Embedding (T-SNE) algorithm is used to reduce the dimensions of the input data. The T-SNE algorithm was used as a preprocessing stage before classification operations. The output of the T-SNE is fed to the support vector machine (SVM). The performance of the proposed systems was evaluated using leave-one-subject-out cross-validation method. The range of recognition results 31.6% to 81.4%, Mean = 61.7% and Standard Deviation = 10.05%. The proposed method can recognize gestures in nearly real time after acquiring a minimum number of frames (629 ms). which is comparable with the results of Jung et al.

Keywords: gesture classification; gesture recognition; social touch; T-SNE, SVM.

1. Introduction
One of the basic interpersonal methods to communicate emotions is through touch. Social touch classification is one of leading research which has great potential for more improvement[1, 2]. Social touch classification can be beneficial in many scientific applications such as robotics, human-robot interaction, etc. One of the most demanding, and yet simple question in the area is how to identify the type (or class) of the touch which affect the robot by analyzing the social touch gesture [3]. Each person has ability to interact with the environment and other persons via touch sensors speared over human soma. These touch sensors provide us the important information about objects that deal with it such as size, shape, position, surface and its movement. The touch considers as simplest and most straightforward of all sensors in human body, and by touch the human can contact environment and other people. Therefore the touch system play main role in human life from early days[4]. Touch gestures consider very important way for human relationship. Small gesture can express strong emotion, from the comforting experience of being touched by one’s spouse, to the discomfort caused by a touch from a stranger [5, 6].
Human can transfer significant emotions through touch language. This ability can apply on robot by using artificial skin equipped with sensors[7,8]. The study of social touch recognition, depend on the idea of the human ability to communicate emotions between them via touch [9]. The understanding of how human able to elicit significant information from social touch, helps designers to develop algorithms and method to represents that response of that the robot in correct form when it interacts with human [10]. To help robot to interpret and understand the human gestures through interaction with human, the gesture patterns must be recognized in a correct form. The precise recognition spur robot to response to human and express its internal state and artificial emotions in positive action. To ensure high ratio of recognition accuracy, sensor devices must measure the touch pressure at high spatial and pressure resolutions [11]. Therefore the robots must equipped with some sensor devices that have the ability to elicit emotions and facial expressions similar to human behavior [12]. To make the robot partnership with the society and interact with human in effective manner, we must prepare the critical requirement such as reliable method for control, perception, ways of learning and response in correct emotion [13].

The paper is organized as follows: Section 2, Background for dataset and relate works. Section 3 describes the theory of SVM and T-SNE algorithm. Section 4 explain the methodology of algorithm. Section 5, illustrate the result and discussion and Section 6, includes a list of conclusions.

2. Background
This section introduces the CoST dataset, describes the previous studies related social touch recognition.

1.1. Dataset
The dataset used in this study dubbed Corpus of Social Touch CoST. It comprises from 14 different touch gesture selected depending on, Yohanan’s touch dictionary [14]. The gestures set is shown in Table 1, [15]. This list of gestures is elected because it is similar to interaction of human with artificial arm. The data collected via 31 participants, where each one gives paper contain some information about the procedure of data collection and asked them to use their right hand to interact with mannequin arm, while their left hand is use to press the keyboard. The participants were asked to perform 14 gestures on an 8 × 8 array of pressure sensor grid wrapped around a mannequin arm. Each touch gesture can be performed in three levels of variations: gentle, normal, and rough. In addition, the participants asked to repeat each gesture 6 times. Therefore, each subject implement 252 gesture instance. For all participants, total recorded data 7812, but some data is lost after data pre-processing, the remaining data samples are 7805. Before participant began to perform gestures, each of them sees a video that contain example for a person who perform all 14 gestures in three variations (i.e. gentle, normal and rough). These gestures were performed on the mannequin arm depending on the demonstration of each gesture that shown in Table 1, Therefore during the actual data collection, each participant shown the name of the gesture not the gesture definition, where the instruction display on a PC monitor for them. The pressure sensors can't sense the movement of the mannequin arm, so the gestures utilize movement of the arm itself for example, push, lift, and swing were neglected. The instruction of gesture was pseudo-randomized, so arranged there in three blocks [16,17].
### Table 1: Gesture definition adapted from [14]

| Gesture label | Gesture Definition |
|---------------|-------------------|
| Grab          | Grasp or seize the arm suddenly and roughly. |
| Hit           | Deliver a forcible blow to the arm with either a closed fist or the side or back of your hand. |
| Massage       | Rub or knead the arm with your hands. |
| Pat           | Gently and quickly touch the arm with the flat of your hand. |
| Pinch         | Tightly and sharply grip the arm between your fingers and thumb. |
| Poke          | Jab or prod the arm with your finger. |
| Press         | Exert a steady force on the arm with your flattened fingers or hand. |
| Rub           | Move your hand repeatedly back and forth on the arm with firm pressure. |
| Scratch       | Rub the arm with your fingernails. |
| Slap          | Quickly and sharply strike the arm with your open hand. |
| Squeeze       | Firmly press the arm between your fingers or both hands. |
| Stroke        | Move your hand with gentle pressure over arm, often repeatedly. |
| Tap           | Strike the arm with a quick light blow or blows using one or more fingers. |
| Tickle        | Touch the arm with light finger movements. |

1.2. **Related works**

Social touch gesture recognition is considered as an interesting topic for researchers in the last years. That is due to that the interaction between Human and machine has a lot of applications in human life. An important part of this researches working on designing and preparing the artificial skin equipped with arrays of sensors that cover robot body. This section introduces a survey about the previous studies.

Jung, 2014 [17], Used the Gaussian Bayesian classifiers, and SVM algorithms for gesture recognition. These two algorithms were applied to Corpus of Social Touch (CoST). The CoST includes 14 distinct touches (Grab, Hit, Massage, Pat, Pinch, Poke, Press, Rub, Scratch, Slap, Squeeze, Stroke, Tap, Tickle). Touches were recorded via pressure sensor. The CoST data set consists of 7805 touch gesture samples performed by 31 (24 males, 7 females) subjects. The age of the subjects was between 21 to 62 years ($M = 34$, $SD = 12$). Each subject performs 3 variations (normal, gentle and rough) for each touch and repeats each touch 6 times. The features set that used as an input to the recognition systems were consisted of 28 features (Mean pressure, Maximum pressure, Pressure variability, Mean pressure per column (8 features), Mean pressure per row (8 features), Contact area (2 features), Peak count (2 features), Displacement (4 features), Duration). The result of touch gesture recognition for Bayesian classifiers depending on these features was from 26% to 74% ($M = 53\%, SD = 11\%$). While for the SVM the gesture recognition rates were from 22% to 63% ($M = 46\%, SD = 9\%$).
Jung et al. 2014 [15], Classified the gestures using Bayesian classifier and the SVM methods to create a baseline. The features set that used as an input to recognition systems consists of 28 features (Mean pressure, Maximum pressure, Pressure variability, Mean pressure per column (8 features), Mean pressure per row (8 features), Contact area (2 features), Peak count (2 features), Displacement (4 features), Duration. For the Bayesian method, the results range was from 24% to 75% (Mean = 54% and SD = 12%). While for the SVM, the results range was from 32% to 75% (Mean 53% and SD = 11%).

Balli Altuglu & Altun, 2015 [18], Split CoST dataset as 65% for training and 35% for testing. The Random Forests (RF) algorithm was used as the tool for touch gesture recognition. They achieved a classification rate between 26% to 95%. The mean classification was 55.6% for 14 social touches depending on the sequential forward floating search. The features employed in the model included pressure data, an image feature, a Hurst exponent, Hjorth parameters, and autoregressive model coefficient features. These features were chosen by applying a sequential floating forward search (SFFS) algorithm. The total features became 42 features. Also test their method on a second dataset, Human-Animal Affective Robot Touch (HAART). The dataset consists of 7 different touch gestures (no touch, constant, pat, rub, scratch, stroke, tickle) and satisfies classification ratio between 60% to 70% based on 133 selected features.

Gaus et al., 2015 [19], Studied the ability to differentiate the social touch gestures category using two methods, Random Forest, and Boosting. The recognition methods were applied to two types of data set. The first one was the HAART. Which consisted of 7 distinct gestures performed by 7 participants. The HAART dataset split into 4 subjects for training and 3 subjects for testing. The second data set was called the Corpus of Social Touch (CoST). Which consisted of 14 distinct gestures performed by 31 participants. The CoST dataset split 21 subjects for training and 10 subjects for testing. The two datasets are available by the Social Touch Gesture Challenge 2015. To test the performance of the proposed methods five different set of high-level features were extracted. The extracted features Included Statistical distribution (SD) of pressure surface, Binary Motion History (BMH), Motion statistical distribution (MSD), Spatial Multi-Scale Motion History Histogram (SMMHH) on touch dynamic, Local Binary Pattern on Three Orthogonal Place (LBPTOP) on touch dynamic. The result of touch gestures recognition was 67% correct classification ratio (CCR) for the HAART and 59% (CCR) for the CoST testing dataset. Hughes et al. 2015 [20], Used Deep autoencoders as a DNN, and Hidden Markov Model (HMM). They classified the CoST and the HAART gesture sets. That used in Social Touch Gestures Challenge for ICMI 2015 to achieve high accuracy for gesture recognition. The gesture-level features were used. Seven distinct features extracted form data set depending on, maximum value of pressure through a touch, the area of pressure on the sensor, and how many gestures repeated at the touch. The CoST dataset was split into 21 subjects for training and 10 subjects for testing. The gesture recognition accuracy was 56% for Cost data set. While HAART dataset was split to 4 subjects for training and 3 subjects for testing, the gesture recognition accuracy was 71%. Ta, Johal, Portaz, Castelli, & Vaufreydaz, 2015 [21], introduced an improved gesture recognition method and applied it on the CoST data set after applying it to HAART data set directly. They depended on three categories of features that includes 273 features divided as Global features, consist of 40 features, represent overall statistics of the gesture. Channel-based features consisted of 192 features, describe the spatial relationship between different channels. The sequence of average pressure features, consisted of 41 features, used the sequence of average pressure over all channels for each frame. The SVM and Random Forest (RF) methods used as a classifier for this two data set. A 3-fold Cross-validation was used to evaluate the performance of the proposed methods. The recognition accuracy for CoST dataset were 60.51% and 60.81% for SVM and RF, respectively. While for HAART data set the recognition, accuracy was 68.52% and 70.91 for SVM and RF, respectively. Jung et al. 2017 [16], Illustrated the step of dataset collection of the CoST in details. The CoST data set consists of 7805 sample of 14 distinct social touch gestures. The dataset was collected by 31 participants, each performed all gestures in three levels: gentle, normal and rough on a pressure sensor.
grid wrapped around a mannequin arm and repeating each gesture 6 times. 54 features were extracted from the dataset include Mean pressure, Maximum pressure, Pressure variability, Mean pressure per row, Mean pressure per column, Contact area per frame, Temporal peak count, Travel distance, Duration of the gesture, Pressure distribution, Spatial peaks, Derivatives, Variance over channels, Direction of movement, Magnitude of movement, Periodicity. The data was split into training and testing set. Leave-one-subject out cross validation was used to evaluate the accuracy of gesture recognition. Four methods of machine learning were applied to the data set to compare their performance and results. The Bayesian classifier achieved 57% (SD = 11%) correct classification ratio (CCR). The Decision tree algorithm introduced 48% (SD = 10%) CCR. SVM with RBF kernel 60% (SD = 11%) CCR and for the NN 59% (SD= 12%) CCR was achieved.

3. Support Vector Machine (SVM)
Support Vector Machine (SVM's) is the popular machine learning method and it is considered as one of the powerful and widespread algorithms used in classification algorithms. In the nonlinear model, the best choice for SVM is Radial Basis Function (RBF) kernel for the following reasons. The RBF is the kernel which transfers the nonlinear samples into multidimensional space. While the linear kernel, can't deal with the samples case when the relationship between attributes and class labels is nonlinear. So, we can consider the linear kernel is a special case of RBF. Thus, the linear kernel with a special parameter has the same amplitude of the RBF kernel with some parameters (C, γ). And the RBF kernel with the certain type of parameters has the same performance of the sigmoid kernel. The RBF kernel used the number of hyperparameters less than the polynomial kernel, this feature makes it less complex from polynomial kernel [22,23].

The performance of model is very sensitive to the change in the value of the γ parameter. If γ is too large, the radius of the area of the effect of the support vectors only consists the support vector itself without the amount of regularization of C that able to prevent overfitting. While setting the value of γ parameter to very small, the model is too constrained and cannot capture the complexity or “shape” of the data. The region of the effect of choosing support vector would consist all training samples [24,25].

When the value of C is set to very large some intermediate values of γ performing equally. It is not necessary to regularize by limiting the number of support vectors. The radius of the RBF kernel alone acts as a good structural regularize. In practice, though it might still be interesting to limit the number of support vectors with a lower value of C so as to get models that uses less memory and that are faster to predict [24, 26].

3.1. Cross-validation and grid-search
The best procedure to get good values of (C, γ), is the "grid-search approach". During cross-validation, various pairs of (C, γ) values are tried and the one with the best cross-validation accuracy is picked out. We found that trying exponentially growing sequences of C and γ is a practical method to determine a good value of parameters (for example, C = 2^−5, 2^−3, . . . , 2^15 and γ = 2^−15, 2^−13, . . . , 2^5). De facto, a lot of advanced methods that can save computational cost. But, there are the following two reasons why we depend on the simple grid-search approach [27, 28]. Firstly, we may not feel safe to use methods which prevent doing an exhaustive parameter search by approximations or heuristics. Secondly, the time required to find good parameters by grid-search is not much more than the time that consumed by advanced methods since there are only two parameters. In addition, The C and γ are independent, so the grid-search can be easily parallelized. Most advanced methods are iterative processes, e.g. walking along a path, which can be hard to parallelize. Since doing a complete grid-search may still be time-consuming, Therefore the using a rough grid first. Then determine a “better” region on the grid, a finer grid search on that region can be conducted [23, 29].

For problems with thousands or more data points the grid search approach works very well. For very large data sets a feasible approach is to randomly choose a subset of the data set, conduct grid-search on them, and then do a better-region-only grid-search on the complete data set [23,25].
A lot of nonlinear dimensionality reduction methods are used to preserve significant information of data in low dimensional form have been proposed. More familiar method is t-Distributed Stochastic Neighbor Embedding (T-SNE) [30,31].

3.2. T-distributed stochastic neighbour embedding (t-sne)

The T-SNE is one of the non-linear dimensional reduction algorithms which is used to eliminate the high dimensional data to low dimensional form. Make the data set more appropriate for human observation. By using the T-SNE algorithm, we have the ability to plot fewer exploratory data analysis, capturing high dimensional data easily and put it local structure [32]. By applying the T-SNE we can identify patterns in the data by determining the observed clusters depending on the similarity between data points with multiple features. However, without mainly a data exploration and visualization techniques, we cannot build any inference by only depending on the T-SNE output. However, when we use the output of T-SNE as a feature or input for another classification algorithms, the role of T-SNE was very well [33,34].

The T-SNE algorithm is preferable over another dimensional reduction algorithm because the strong gradient of T-SNE recreate the dissimilar data points which modeled by a small pairwise distance in the low-dimensional representation. While in another dimensional reduction algorithm we can't get this representation. In addition, most dimensional reduction methods, when the strength of the repulsion between very dissimilar data points is proportional to their pairwise distance in the low-dimensional map, which may cause dissimilar data points to move much too far away from each other. While the T-SNE introduces strong repulsions between dissimilar data points that are modeled by small pairwise distances, these repulsions do not go to infinity [34].

4. Methodology

Each frame of data is recorded with an 8x8 grid sensor; it can be assumed as a small image. Thus, the consecutive frames can be represented like a video. In image processing, every image must have a fixed resolution (e.g. All input images must be 200x200 pixels). However, it may not be true for video processing. We can have fixed resolution but with variable video length. For example, in our approach the resolution is 8x8 and the length is (L) which varies for every sample, L is vary for each sample because each subject performs the social gestures differently. The data samples in the CoST dataset do not have fixed length. For example, poking gesture which may take only 10-time steps (It is very low compared to other which may be more than 400-time steps). One of the main efforts for our proposed method is to find a proper L. Selecting an efficient L is a trade-off between accuracy (larger L) and high-speed classification (smaller L). In other words, larger L means more information and potentially more accuracy. On the other hand, smaller L means the fewer number of frames in real implementation and therefore, faster classification. No matter how fast our computational resources are. We still need to wait for the human to perform the touch gesture. The 8x8xL sample is needed to be flattened in order to be processed by the SVM. It means that the samples are converted from a 3D tensor into the 1D vector. Since the dimension of the input are so high which are not efficient to be trained by the SVM in both computationally and performance wise, we have to apply a pre-processing operation to reduce the dimension. The preprocessing that is implemented on the CoST dataset is based on feature extraction which is manually designed. However, we selected a preprocessing approach which does not need manual feature extraction. T-Distributed Stochastic Neighbor Embedding (T-SNE) is one of the most successful approaches which is mainly used for dimension reduction.

Here is an example why a preprocessing is necessary. The most efficient frame length for the SVM model is 85 needs to process sample as big as 8x8x85= 5440 which is not plausible for us. Therefore, we reduce dimension using T-SNE from 8x8x85 to a reasonable number (e.g. 100) which we find later user grid search in parameter space.

However, such a big dimension is also unnecessary for T-SNE for two reasons. Firstly, it will increase the computational load on the T-SNE. Secondly, the frames in each sample are not drastically
changing. Consecutive frames are most likely to be similar to small change. The data is recorded with 135 Hz (frame per second). This means that the time difference between two consecutive frames are 7.5 Milliseconds. So, we calculated the average of, e.g., 10 frames and build single frame with the average value of the sensors. So each (8x8x10 frames become 8x8x1 frame). Therefore, the 8x8x85 frames are converted to new frames of 8x8x9 frames (the last frame is average of 5 original frames). Therefore, we have introduced a very simple processing approach even before T-SNE with a simple average over k number of frames. The k will be determined using Leave-One-Subject-Out Cross-Validation. This average provides a simple temporal abstraction over k frames as the described in the following equation:

\[ Y_{i,j} = \frac{1}{k} \sum_{p=1}^{k} Z_{i,j}^p \text{ } & i,j = 1,2,\ldots,8 \]  

(1)

Where \( Z \) is the original frame, \( k \) is the number of frames to compute their average and \( Y \) is the resulted average frame.

The only drawback in this method is that if the number of frames for the averaging is increasing, then we will lose dynamic of gesture. The candidate values for Average-filter can be 1 (means no averaging over the frame), 5, 10, 20, 50 … 85. When we set the test average filter = 85 which means that we average all 85 frames and put it in one frame.

For the SVM input data, the T-SNE is used to reduce the its dimension. Even after using the average filter the dimensions of input are still high (8x8x9 = 576). The T-SNE uses the Principle Component Analysis (PCA) as pre-processing to reduce the dimension significantly. However, we should set the desired dimension as another hyper parameter. This will be the final dimension which is used by the SVM. If we set input dimension for SVM to 2 it means that our original frames size eventually reduces the 5440 frames (in case of L=85) to 2.

The candidate value for the reduced dimension can be 2, 5, 10, 20, 50 and "no dimension reduction".

The "no dimension reduction" option makes it clear whether the dimension reduction using T-SNE is helpful. Therefore, we started with minimum dimension up to maximum dimension (i.e. "no dimension reduction").

To summarize, our approach has 3 hyperparameters which are, the frame length (L), average filter (k) and reduced dimension (d). The maximum dimension depends on the frame length and the average filter. That means we will not use T-SNE for dimension reduction. The formula for the maximum dimension (MAX_dim), if \( L = \) Frame Length, and \( k = \) Average-filter as following: -

\[ \text{MAX}_\text{dim} = 64 \times \text{floor}(L/k) \]  

(2)

Where (64 is the 8x8 frame dimensions and the floor is to always choose the smaller integer value out of the equation). For example, if we set frame length to 85, and average filter to 10 and after trying all dimensions for SVM now we need to calculate MAX_dim results from the two other hyperparameters. According to Eq. (2),

\[ \text{MAX}_\text{dim} = 64 \times \text{floor}(L/k) \]  

= 64 x floor (85 / 10)  

= 64 x floor (8) = 8 x 64 = 512.

So, if we set dimension for SVM as 512 then, in this situation T-SNE will not do any dimension reduction because the given dimension to it and desired dimension will be same. Therefore, T-SNE will not use. So, by setting this number we show that why dimension reduction can be helpful.

After the T-SNE prepares the data for SVM. The SVM algorithm is used to recognize the social touch. We use libsvm library package: This library applies the Radial Basis Function (RBF) kernel in the SVM as the following equation:

\[ K(\text{X}_i, \text{X}_j) = \exp(- \gamma | \text{X}_i - \text{X}_j |^2) \]  

(3)
Where X is the features, after T-SNE preprocessing. The libsvm library has two hyperparameters, C, and gamma. Gamma (a positive number which is a hyper-parameter) determines the shape of the RBF kernel. Another hyper-parameter is C, which is the penalty for error. Usually, to find the best gamma and C, a grid search is the good solution. Candidate Gamma value are \(2^{-10}, 2^{-9}, 2^9, 2^{10}\) and candidate C values are \(2^{-10}, 2^{-9}, 2^9, 2^{10}\).

To sum up, our hybrid approach consists of; averaging, T-SNE and SVM have 5 hyperparameters of frame length (L), average filter (k), reduced dimension (d), C and Gamma.

5. Results

In this paper, we have used the T-SNE algorithm to reduce the dimensions of input data. The T-SNE algorithm is used as pre-processing step before running the classification methods. The output from T-SNE is used as an input to the SVM. The libsvm library\(^1\) is used to write the code of this model. Three hyper-parameters effects on performance of this model. The first parameter is the frame length, as mentioned in the previous model, has a significant effect on recognition accuracy. The value for frame length is set to 85. This frame length choosing after tested various value of frame length. We tested other frame lengths such as 10, 50, 150 which lead to a poor performance. Secondly, the average window (filter) size also has an effect on the result of recognition accuracy. The best size for filter among the candidates was 85 which means the whole gesture should be averaged out as a single frame. Finally, the maximum dimension of input for the SVM which is set to a range from 2 to 64. The best value of maximum dimension is 64. This value used to determine the value of parameter.

The LibSVM library limits the parameter search to two factors of gamma and C. A grid search is the good solution to determine the optimal value for these parameters. Therefore, we set the value of Gamma to \(2^{-10}, 2^{-9}, \ldots, 2^9, 2^{10}\) and C to \(2^{-10}, 2^{-9}, \ldots, 2^9, 2^{10}\). This values of gamma and C, which represents the recommended candidates to search. The search for the optimal parameters of the SVM is depicted in the Figure 1.

The results of the leave-one-subject-out cross-validation for all subject’s range are from 31.6% to 81.4%, Mean = 61.7% and Standard Deviation = 10.05%. as presented in Figure 2 and 3. From Figure 3, we can see that the classes of Stroke and Scratch represents the least accurately classified labels. Figure 4 explain confusion matrix for the results obtained by SVM with the T-SNE algorithm for gesture recognition. The results are presented as accumulated of the leave-one-subject-out cross-validation for all subjects. Table 2 illustrates the comparison between the proposed algorithm and other existing classification algorithms that applied to the same data set (CoST). The proposed algorithm improves the Correct Classification Ratio (CCR).

---

\(^1\) https://www.csie.ntu.edu.tw/~cjlin/libsvm/
**Figure 1.** Heat map of the results for the parameters of the SVM method for the gesture recognition. The optimal results are marked with the red circle. Each block in the grid is corresponded by average of leave-one-subject-out cross validation over all subjects.

**Figure 2.** The recognition accuracy for each participant when applied SVM with the T-SNE algorithm on CoST data set.
Figure 3. The accuracy of SVM with the T-SNE algorithm in predicting each gesture class.

Figure 4. The results of our proposed SVM with the T-SNE algorithm for gesture recognition is presented as accumulated confusion matrix of the leave-one-subject-out cross-validation for all subjects.
Table 2. Comparison of features from other existing classification methods applied to the same dataset (14 touch gesture by Piezoresistive fabric Pressure Sensor)

| No. | Reference | Preprocessing | # Features | Classify Method          | Mean % | S.D. % |
|-----|-----------|---------------|------------|--------------------------|--------|--------|
| 1   | [Jung, 2014; Jung et al., 2014] | Yes           | 28         | Bayesian classifier       | 54     | 12     |
|     |           |               |            | SVM                      | 53     | 11     |
| 2   | [van Wingerden et al., 2014]      | Yes           | 45         | Neural Network           | 54     | 15     |
| 3   | [Balli Al屯glu et al., 2015]      | Yes           | 42         | Random Forests (RF)      | 55.6   | 13     |
| 4   | [Gaus et al., 2015]               | Yes           | 5 set      | Random Forests (RF)      | 59     |        |
|     |           |               |            | Boosting                 | 58     |        |
| 5   | [Hughes et al., 2015]             | Yes           | 7          | Deep Autoencoders        | 66     |        |
| 6   | [Ta et al., 2015]                 | Yes           | 273        | SVM                      | 60.5   |        |
|     |           |               |            | Random Forests (RF)      | 60.8   |        |
| 7   | [Jung et al., 2017]               | Yes           | 54         | Decision tree algorithm  | 57     | 11     |
|     |           |               |            | SVM                      | 48     | 10     |
|     |           |               |            | Neural Network           | 60     | 11     |
| 8   | Our Proposed Method               | Yes           | 64         | SVM and T-SNE            | 61.7   | 10.8   |

6. Conclusion
In this paper, we have proposed systems to recognize the social touch gesture for CoST dataset. The CoST dataset 14 different social touches. We used a Support Vector Machine method to classified gesture and T-SNE foe dimension reduction as pre-proceeding. The CoST dataset is selected to train our SVM due to the variety of classes. The system used to classify touch gesture depend on frame length equal to 85. This means the system needs to 629 milliseconds to recognize the touch. Five hyper parameters factors effect on the performance of Support Vector Machine include of frame length (L), average filter (k), reduced dimension (d), C and Gamma. To find the best gamma and C, a grid search is a good solution. When the range of grid search of Gamma and C factors increase, we get the best result. However, on the other hand, the time increase and vice versa. In future we can use Cost dataset to reorganize the subject who perform the gesture. The results of the proposed SVM system show that better performance comparing with state-of-the-art results based on the leave-one-subject-out cross-validation is obtained. Where the recognition accuracy satisfies this system better from all method used by another system.
References

[1] A. Flagg and K. MacLean, "Affective touch gesture recognition for a furry zoomorphic machine," in Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction, 2013, pp. 25-32.

[2] F. R. Ortega, N. Rishe, A. Barreto, F. Abyarjoo, and M. Adjouadi, "Multi-Touch Gesture Recognition Using Feature Extraction," in Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering, ed: Springer, 2015, pp. 291-296.

[3] J. Chang, K. MacLean, and S. Yohanan, "Gesture recognition in the haptic creature," Haptics: Generating and Perceiving Tangible Sensations, pp. 385-391, 2010.

[4] M. J. Hertenstein, R. Holmes, M. McCullough, and D. Keltner, "The communication of emotion via touch," Emotion, vol. 9, p. 566, 2009.

[5] J.-J. Cabibihan and S. S. Chauhan, "Physiological responses to affective tele-touch during induced emotional stimuli," IEEE Transactions on Affective Computing, vol. 8, pp. 108-118, 2017.

[6] M. M. Jung, L. van der Leij, and S. M. Kelders, "an exploration of the Benefits of an animallike robot companion with More advanced Touch interaction capabilities for Dementia care," Frontiers in ICT, vol. 4, p. 16, 2017.

[7] A. Flagg, D. Tam, K. MacLean, and R. Flagg, "Conductive fur sensing for a gesture-aware furry robot," in Haptics Symposium (HAPTICS), 2012 IEEE, 2012, pp. 99-104.

[8] M. A. Hoepflinger, C. D. Remy, M. Hutter, L. Spinello, and R. Siegwart, "Haptic terrain classification for legged robots," in Robotics and Automation (ICRA), 2010 IEEE International Conference on, 2010, pp. 2828-2833.

[9] K. E. MacLean, S. Yohanan, Y. S. Seifdgar, M. K. Pan, E. Croft, and J. McGrenere, "Emotional Communication and Implicit Control through Touch," 2012.

[10] D. S. Tawil, D. Rye, and M. Velonaki, "Touch modality interpretation for an EIT-based sensitive skin," in Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 3770-3776.

[11] F. Naya, J. Yamato, and K. Shinozawa, "Recognizing human touching behaviors using a haptic interface for a pet-robot," in Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on, 1999, pp. 1030-1034.

[12] L. Cañamero and J. Fredslund, "I show you how I like you can you read it in my face?[robotics]," IEEE Transactions on systems, man, and cybernetics-Part A: Systems and humans, vol. 31, pp. 454-459, 2001.

[13] U. Martinez-Hernandez and T. J. Prescott, "Expressive touch: Control of robot emotional expression by touch," in 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2016, pp. 974-979.

[14] S. Yohanan and K. E. MacLean, "The role of affective touch in human–robot interaction: Human intent and expectations in touching the haptic creature," International Journal of Social Robotics, vol. 4, pp. 163-180, 2012.

[15] M. M. Jung, R. Poppe, M. Poel, and D. K. Heylen, "Touching the Void--Introducing CoST: Corpus of Social Touch," in Proceedings of the 16th International Conference on Multimodal Interaction, 2014, pp. 120-127.

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

[17] M. M. Jung, "Towards social touch intelligence: developing a robust system for automatic touch recognition," in Proceedings of the 16th International Conference on Multimodal Interaction, 2014, pp. 344-348.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.
[19] Y. F. A. Gaus, T. Olugbade, A. Jan, R. Qin, J. Liu, F. Zhang, et al., "Social touch gesture recognition using random forest and boosting on distinct feature sets," in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, 2015, pp. 399-406.

[20] D. Hughes, N. Farrow, H. Profita, and N. Correll, "Detecting and Identifying Tactile Gestures using Deep Autoencoders, Geometric Moments and Gesture Level Features," presented at the Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, Seattle, Washington, USA, 2015.

[21] K. Nakajima, "http://neuralnetworksanddeeplearning.com/chap6.htmlhttp://cs231n.github.io/," 2015.

[22] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," 2003.

[23] C. L. Reed, R. J. Caselli, and M. J. Farah, "Tactile agnosia: Underlying impairment and implications for normal tactile object recognition," Brain, vol. 119, pp. 875-888, 1996.

[24] S. Q. Fleh, O. Bayat, S. Al-Azawi, and O. N. Ucan, "A Systematic Mapping Study on Touch Classification," International Journal of Computer Science and Network Security, vol. 18, pp. 7-15, 2018.

[25] P. Doliotis, A. Stefan, C. McMurrough, D. Eckhard, and V. Athitsos, "Comparing gesture recognition accuracy using color and depth information," in Proceedings of the 4th international conference on PErvasive technologies related to assistive environments, 2011, p. 20.

[26] D. H. Jeong, C. Ziemkiewicz, W. Ribarsky, R. Chang, and C. V. Center, "Understanding principal component analysis using a visual analytics tool," Charlotte visualization center, UNC Charlotte, 2009.

[27] M. Law, "A simple introduction to support vector machines," Lecture for CSE, vol. 802, 2006.

[28] saurabh.jaju, "https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/," ed, 2017.

[29] H. Abdi and L. J. Williams, "Principal component analysis," Wiley interdisciplinary reviews: computational statistics, vol. 2, pp. 433-459, 2010.

[30] saurabh.jaju, "https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/" 2017.

[31] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM transactions on intelligent systems and technology (TIST), vol. 2, p. 27, 2011.

[32] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579-2605, 2008.

[33] s. albawi, o. bayat, s. al-azawi, and O. n. ucan, "Social Touch Gesture Recognition Using Convolutional Neural Network," Computational intelligence and neuroscience Journal, 2018.

[34] L. van der Maaten and G. Hinton, "User's Guide for t-SNE Software," 2015.