Intra-/Inter-user Adaptation Framework for Wearable Gesture Sensing Device

Kosuke Kikui
Keio University
Yokohama, Japan
kosuke@imlab.ics.keio.ac.jp

Yuta Itoh
Tokyo Institute of Tech./RIKEN AIP
Yokohama, Japan
yuta.itoh@c.titech.ac.jp

Makoto Yamada
Kyoto University/RIKEN AIP
Kyoto, Japan
myamada@i.kyoto-u.ac.jp

Yuta Sugiura
Keio University, Yokohama, Japan
sugiura@keio.jp

Maki Sugimoto
Keio University, Yokohama, Japan
maki@imlab.ics.keio.ac.jp

ABSTRACT

The photo reflective sensor (PRS), a tiny distant-measurement module, is a popular electronic component widely used in wearable user-interfaces. An unavoidable issue of such wearable PRS devices in practical use is the need of user-independent training to have high gesture recognition accuracy. Each new user has to re-train a device by providing new training data (we call the inter-user setup). Even worse, re-training is also necessary ideally every time when the *same* user re-wears the device (we call the intra-user setup). In this paper, we propose a domain adaptation framework to reduce this training cost of users. Specifically, we adapt a pre-trained convolutional neural network (CNN) for both inter-user and intra-user setups to maintain the recognition accuracy high. We demonstrate, with an actual PRS device, that our framework significantly improves the average classification accuracy of the intra-user and inter-user setups up to 87.43% and 80.06% against the baseline (non-adapted) setups with the accuracy 68.96% and 63.26% respectively.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g., HCI): User Interfaces

Author Keywords

Photo Reflective Sensor; Domain Adaptation; CNN

INTRODUCTION

The photo reflective sensors (PRSs) are small, inexpensive electronic components that can measure the distance to an object [3]. A PRS consists of an infrared (IR) LED and an IR photo diode. The LED emits IR light to an object and the diode measures the intensity of the reflected IR light from the object. Due to the flexibility of sensor arrangements, we can design various wearable user interfaces with PRSs. For example, CheekInput by Yamashita et al. attaches PRSs to a head mounted display (HMD) and measures the change in the face shape for gesture input [10]. More specifically, the user places the index finger on the chin and moves the finger according to pre-defined gestures (e.g., Line and Circle). This pushes the skin around the finger, and this makes the distance from the skin to the PRS sensor installed on the HMD changes.

These PRS based wearable devices employ machine learning methods such as support vector machine (SVM) to identify gestures. However, the classification accuracy of the device can be significantly degraded if a user data is not included in the training set. In particular, if PRSs are used for HMD, unlike other gesture interfaces such as hand and body motion sensors, gathered data are highly user-dependent since the style of user's input (e.g., facial expression, gesture etc.) is different for each user [5]. Moreover, even if the same user re-wears a PRS device, the input data distribution can be different due to a slight shift of wearing conditions. Thus, to prevent the

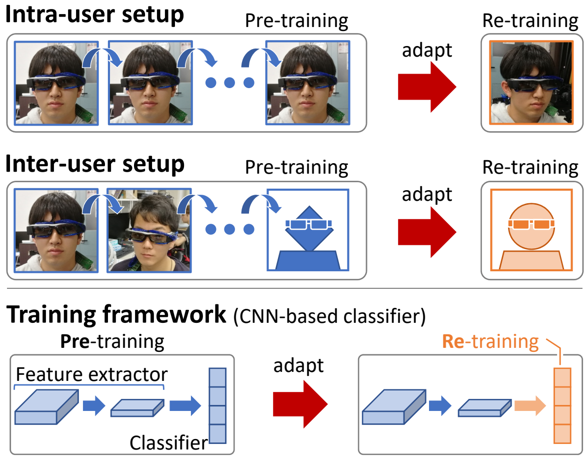


Figure 1. Schematic illustrations of our adaptation framework for wearable gesture sensing devices using PRSs. We aim to reduce the re-training cost of users by applying a domain adaptation methodology. The intra-user setup is to re-train the device after re-wearing by the same user, while the inter-user setup is to adapt the device to a new user.

ISWC '18, October 8–12, 2018, Singapore, Singapore

DOI: https://doi.org/10.1145/3267242.3267256

performance degradation caused by user shift is an important problem in practice. (See Fig. 1).

One simple solution to the problem is to build a machine learning model for a new user or a new setup from scratch. However, since we need to collect *large* labeled training dataset for building machine learning models when the recognition performance degrades, this approach is not useful in practice.

In this paper, we thus provide a domain adaptation based technique [1] to reduce the cost of rebuilding machine learning models. More specifically, we employ the convolutional neural networks (CNN) [4] as a base classifier and propose a simple yet effective fine-tuning based domain adaptation framework for wearable gesture sensing devices. Since the proposed approach can build a reasonable model from a small number of labeled data by utilizing the pre-trained CNN model. Thus, we can minimize the training cost of a new user or the same user in a re-wearing situation.

Contributions

Our main contributions include the following: We propose a user adaptation framework for wearable devices using PRSs. We demonstrate that the classification accuracy of both the intra-user adaptation (87.43%) and the inter-user adaptation (80.06%) with our method are significantly higher than that of the method without adaption (68.96%, 63.26%).

RELATED WORK

We briefly review existing PRS devices and domain adaptation.

Photo Reflective Sensor Devices

PRSs are frequently used in the human computer interaction (HCI) and the ubiquitous computing communities for human behavior recognition by utilizing a part of the skin as an input.

An eyewear device AffectiveWear detects different facial expressions by measuring the skin deformation via 17 RPSs attached on the device [5]. Anusha et al. proposed a method to recognize aerial gestures made around an HMD that are detected by several PRSs [9]. Tanaka et al. measure the movement of the user's jaw by measuring cheek motion via a PRS [8]. CheekInput is an eyewear device embedded with multiple PRSs to measure the change in wrinkle shape of the skin during touch, and it recognizes the direction of pulling of the forehead or cheeks or to recognize gestures respectively [10].

In general, all those devices require training data collection for each individual user thus potentially suffers the adaption problem. With our adaptation framework, we aim to reduce the training cost necessary for using those wearable devices.

Domain Adaptation

A standard machine learning algorithm assumes that the training and test distributions are the same. However, in practice, the training and the test datasets measured by wearable devices can be different due to reinstallation of devices. Domain adaptation is a machine learning framework to handle such distribution changes [1]. The key idea of domain adaptation is to adapt the model trained on the training set to the test set.

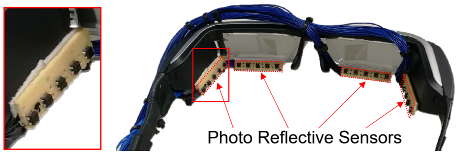


Figure 2. CheekInput [10]. (right) The device with 20 PRSs attached on the frame of an HMD. (left) A zoomed image of an array of 5 PRSs.

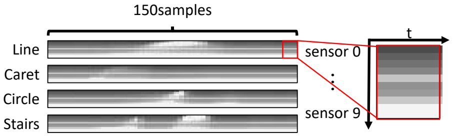


Figure 3. Examples of snapshot of time-series input data for CNNs. For the detail of each gesture type, see Figure 5. In this paper, different sensor data in the y axis, time series values in the x axis.

A widely used *supervised* domain adaptation technique for deep learning algorithms including convolutional neural networks (CNN) [2, 4, 7] is fine-tuning. Specifically, we keep some of the pre-trained model parameters trained by a large number of training data and re-train the rest of layers by test data. Thanks to this, we can adapt a pre-trained training model to fit test instances.

METHOD

We propose a *supervised* domain adaptation method for a PRS device, and demonstrate it on CheekInput [10], which recognizes gestures of fingers touching the user's cheek. CheekInput has 20 PRSs at the bottom of an head-mounted display (HMD) as shown in Figure 2. Each side of CheekInput has 10 PRSs and we used the left side for data collection.

Convolutional Neural Network (CNN)

As a base classifier, we employ the convolutional neural network (CNN) [4]. Figure 4 shows the schematic diagram of our single-layer CNN used in this paper. The single-layer CNN consists of a convolution layer and a max-pooling layer. The convolution filter size is 5×10 and the stride is 1. The max-pooling kernel size is 2×2 and the stride is 2.

For the input of CNN, we use a 2D snapshot which is obtained by transforming the PRSs time-series data (10 sensors × 150 frames) to an image. Figure 3 shows examples of time-series snapshots from CheekInput where each snapshot consists of RPS sensors (y axis) and their time-series values (x axis). The sensor values range from 0 to 1023.

Data Adaptation by Fine Tuning

We aim to reduce the data collection cost under intra-user setup (e.g., a user re-wears the device) and the inter-user setup (e.g., a user not included in the training data). More specifically, we first construct a base CNN classifier using a large number of training data (pre-trained CNN), and then we apply a domain adaptation technique for the pre-trained CNN. Since it is natural to assume that the convolution layer (i.e., feature extraction) is the same for all users, we only tune the fully connected layer of the network. To this end, we use the model parameter of the convolution layer of the pre-trained CNN,

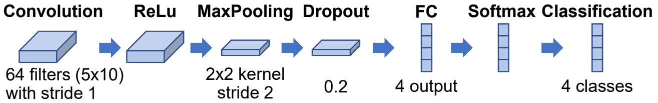


Figure 4. CNN structure. Input data size is 10 sensors $\times$ 150 frames. Filter num is 64, filter size of the convolutional layer is $[5, 10]$ and the stride is 1, and kernel size of maxpooling layer is $[2, 2]$ and the stride is 2. In the fully connected layer, all the inputs are connected 4 output.

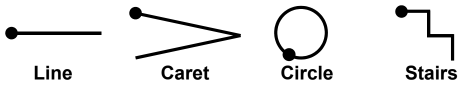


Figure 5. Four kinds of gestures we collected. The black point is start point of the gestures.

and then train the fully connected layer using a small number of labeled new user data.

EXPERIMENT

In this section, we evaluated our adaptation framework using CheeckInput [10].We used MATLAB to construct a CNN model and trained the model by momentum stochastic gradient descent (MSGD). We then evaluated the accuracy of the learned classifiers.

Time-series Gesture Data Collection

With CheekInput, we defined 4 cheek gestures: Line, Caret, Circle, and Stairs as shown in Figure 5. All gestures are performed on the left cheek and PRSs measure the change in the face shape. We asked 6 subjects to perform the 4 gesture types 30 times for each. The recording period of each gesture was 5 sec (150 frames at 30 Hz) and the subjects are instructed to complete given time-series gestures within that duration.

Each subject performs two data collection trials (A and B) while re-wearing the device between the trials. In each trial, the subjects perform displayed gestures. The instruction is displayed for 5 seconds, and the subjects perform the cheek gesture to their left cheek during that time. Note that we randomized the order of the displayed gestures for each. This process was repeated 30 times, and 120 time-series instances in total were collected for each trial for each subject (i.e., 240 instances per subject). Moreover, we introduced about 3 min break between the 2 trials for each subject to simulate a situation where a user re-wears the same PRS device, which could introduce the shift of input data distribution.

To see how the training data size affects the classification accuracy, we calculate the classification accuracy in each experiment by decreasing the number of training data used for CNN re-training. Since each gesture in each trial has 30 instances, we re-trained with 1, 3, 6, 10, 15, 20, and 24 instances, and then tested with the rest to calculate the average classification accuracy (Table 1).

Intra-user Adaptation

In the intra-user adaptation setup, for each user, we made a pre-trained CNN trained with data of a trial (trial A or B), and then we tuned the pre-trained CNN only with the other trial. Specifically, we used total 108 gesture instances (4 gestures × 27 instances) of a trial (trial A or B) for pre-training and other 12 gesture instances for validation, whereas we used gesture instances from the other trial (trial B or A) for the re-training and test (See Fig. 6). With validation data, we calculate the validation error. For training CNN, we set the initial learn rate 1e$-$5 and the batch size 60. For comparison, for each data split, we also trained a CNN model using only a new user data.

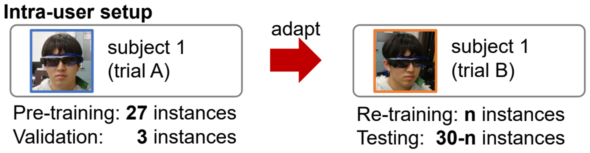


Figure 6. The intra-user setup conducted pre-training with 27 instances in the trial A(B), re-training with n instances in the trial B(A) (n=24, 20, 15, 10, 6, 3 or 1). In the testing, (30 - n) instances are used.

Re-training data	24	20	15	10	6	3	1
Final test data	6	10	15	20	24	27	29

Table 1. Split combination of re-training and final test data used in the both intra-/inter-user setups. Since each trial has 30 instances for each gesture, we re-trained with 1, 3, 6, 10, 15, 20, and 24 instances, and then tested with the rest to calculate the average classification accuracy.

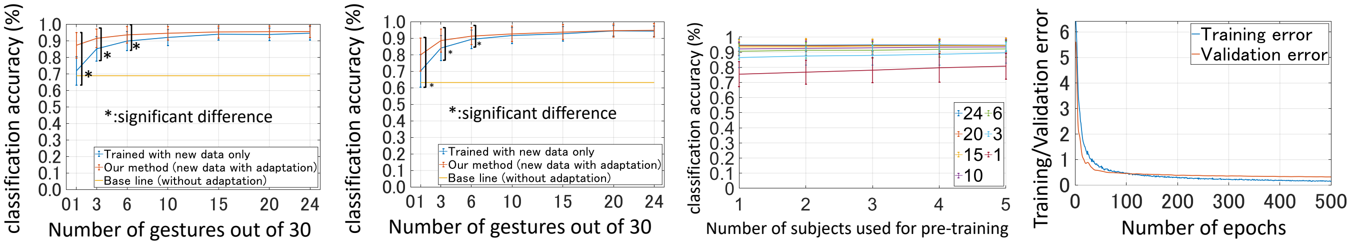
After calculating the classification accuracy of each method, we gather all classification accuracy of all subjects for each split size for each method (7 pairs. See Figure 7). We then apply the McNemar test for each pair of the same split data size to compare the two classification models [6]. This results in total 7 tests, we thus also applied Bonferroni correction on each p-value. As the result, we found that our method significantly improves the classification accuracy when we only used 1, 3, and 6 instances of the re-wearing trial for fine tuning. Using only 3 instances achieves over 90% of the classification accuracy. At the same time, using more instances for fine tuning made no differences with solely training the CNN from scratch with new instances only.

Note that we also considered if simply using the pre-trained CNN works for the other trial after re-wearing the device. The classification accuracy in such setup was overall mere 68.96%. Thus re-training is still a necessary step for a PRS device.

Inter-user Adaptation

In the inter-user adaptation, we use a pre-trained CNN trained by the instances of 5 subjects and re-train it with the remaining subject's data with data split as shown in Table 1. In total, 5 subjects $\times$ 4 gestures $\times$ 27 instances $=$ 540 gesture instances were used for pre-training, and 60 instances were used for validation (Fig. 7). For training a CNN, we experimentally set the initial learn rate 1e$-$5 and the batch size 300.

The average classification accuracy of each method is shown in Figures 7(a)(b). When the number of test gestures is large (e.g., the number of gestures is 24), the average classification accuracy of fine-tuned CNN and the CNN trained with test data are almost the same (*with* fine-tuning (94.38%) and *without* fine-tuning (94.86%). On the other hand, when the number of test gestures is small (i.e, fine-tuned with only 1 instance), the fine-tuned CNN outperformed the CNN trained by only test data (80.06% vs. 70.29%). The baseline method gets only 63.26% average classification accuracy.



(a) Intra-user.

(b) Inter-user.

(c) Performance w.r.t. # of users.

(d) Loss function.

Figure 7. (a)(b): Results of intra-user and inter-user adaptation. Red-line is the average classification accuracy of our method. Blue-line is trained with new data only (without pre-train). Yellow-line is baseline (without adaptation). (c): The number of subjects used for pre-training effect to accuracy. The more the number of participants increased, the more the average of classification accuracy increased. (d): The loss function of CNN pre-training.

DISCUSSION AND FUTURE WORK

Difference between Intra-/Inter-user Adaptation

When many gesture data was used for re-training, the classification accuracy of inter-user adaptation and intra-user adaptation was almost the same. However, when re-trained with less data, the classification accuracy of intra-user adaptation was higher than that of the inter-user adaptation. This result could come from the fact that the pre-trained network in the intra-user setup is less diverse (i.e., the feature extraction part can be similar for each data trials) than that of the inter-user setups since there is only one user in the intra-user setup.

The Number of Subjects Used for Pre-training

In the inter-user adaptation, the number of participants used for pre-training affects to the classification accuracy (Figure 7(c)). That is, if the number of pre-training subjects used for pre-training, we tend to get high classification accuracy. Thus, we need to collect many subject's data to improve classification accuracy re-train with less gesture data.

Continuous Recognition for Practical Use

In this paper, our gesture recognition is conducted offline and the detection was manually done. For a practical, automated gesture detection, an option is to detect the start of a gesture by a time point at which the differential value has changed, and identify when a fixed time has elapsed from the start.

Evaluation with other PRS devices and layouts

Our framework is generally applicable to other PRS user-interfaces using time-series input. Thus, it is worth to investigate how our framework works on other devices. Basically, if the number of layers increases, we tend to get better performance. However, it also increases computational time. For wearable devices, real-time processing is necessary. We will in future investigate the classification accuracy and the computational time, and find a good trade off between them.

CONCLUSION

We proposed an adaption framework for wearable user-interfaces using PRSs. By tuning the CNN-based classifier pre-trained with users, we demonstrate that our adaption framework significantly improves the classification accuracy in both the intra-/inter-user setups. The evaluation showed that even using only single data of each gesture from the same user in the re-training achieves the classification accuracy of 87.43% in the re-wearing situation. The same single data collection with different users (i.e., the inter-user setup) also achieved the average classification accuracy of 80.06%. Overall our adaptation framework showed a potential that it could reduce the re-training cost while maintaining the classification accuracy.

REFERENCES

1. John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In *EMNLP*.
2. Jeff Donahue et al. 2013. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. (2013).
3. Takashi Kikuchi et al. 2017. EarTouch: Turning the Ear into an Input Surface. In *MobileHCI*.
4. Yann LeCun et al. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation* 1, 4 (1989), 541–551.
5. Katsutoshi Masai et al. Facial Expression Recognition in Daily Life by Embedded Photo Reflective Sensors on Smart Eyewear. In *IUI*.
6. Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* 12, 2 (01 Jun 1947), 153–157.
7. Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).
8. Hidekazu Tanaka et al. 2011. Chewing Jockey: Augmented Food Texture by Using Sound Based on the Cross-modal Effect. In *SIGGRAPH Asia 2011 Emerging Technologies (SA '11)*. Article 18, 1 pages.
9. Anusha Withana et al. 2015. zSense: Enabling Shallow Depth Gesture Recognition for Greater Input Expressivity on Smart Wearables. In *CHI*.
10. Koki Yamashita and et al. 2017. CheekInput: Turning Your Cheek into an Input Surface by Embedded Optical Sensors on a Head-mounted Display. In *ACM VRST*. ACM, 19:1–19:8.