EMG Sensor Based Finger Movement Detection

Mohammadbagher Fotouhi[†*], Ghazaleh Jowkar[†], Tongjue Wang[†], Juhua Hu[†], Payman Arabshahi[*], Wei Cheng[*†]
[†] *Tacoma School of Engineering and Technology*
[*] *Department of Electrical and Computer Engineering*
University of Washington, USA
{mfotouhi, tw66, juhuah, uwcheng}@uw.edu, ghazal.jowkar@gmail.com, paymana@ece.uw.edu

***Abstract*—In this article, we introduce our pilot research on finger movement detection based on the data collected by the Electromyography (EMG) sensor that is placed on the forearm to sense the muscle movements. The EMG sensor is battery powered and connects to the smartphone app through Bluetooth. The size of the EMG sensor is similar to a regular adhesive bandage for minor wound care. As it can be easily covered by sleeve, if we can successfully use them to accurately detect each individual finger movement, there will be many interesting applications such as playing a visual music instrument. We studied a machine learning model for processing the unfiltered EMG signals generated by the muscle movement when we move each individual finger. The current average accuracy of five finger movement detection is about 87%.**

***Index Terms*—Signal Processing, Feature extraction, Dimension reduction, Machine learning.**

I. Introduction

A critical component of most recent human-machine interaction (HCI) devices is Myoelectric control systems, a system that receives the Electromyography (EMG) signal originated from muscle movement. Most of the existing studies are focused on EMG signal based gesture detection [1]. Though using EMG signals for accurately detecting individual finger movements is more challenging than gesture detection, it is a necessary precursor to control a prosthetic hand for the tasks such as typing. The success of finger movement detection can also extend the EMG application to the areas such as virtual music instrument play, secure communication (sending morse code), authentication, and key pairing.

Fig. 1 shows the nano EMG sensor we used to capture the muscle signal when an individual finger is moving and the Android app for data collection. Each individual finger has two movement pattens. One is closed for half a second and open for half a second, the other is closed for one second and open for one second. We target on detecting both which finger is moving and what is the movement patten.

To achieve the goal, we conduct research in two steps: (1) Extract each movement from the raw signal, via which we can detect the movement patten, and (2) Individual finger detection from each extracted movement signal. In the rest of this article, we will introduce these research followed by the evaluation results.

II. Movement Extraction

The EMG signals are acquired from epidermal electronic systems. The collected EMG signals are noisy and distorted.

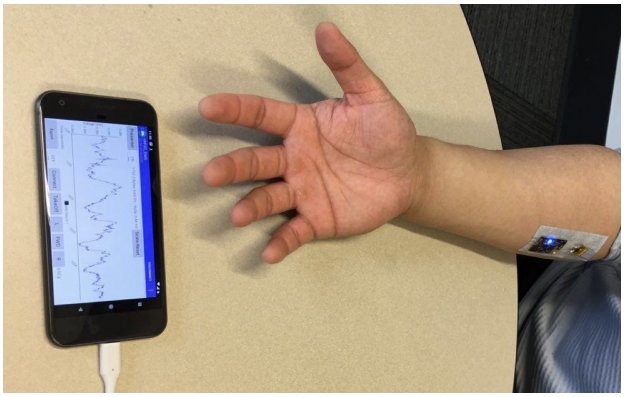

Fig. 1: EMG Sensor and Data Collection

As the signal's quality and the accuracy of extracted movements significantly affect the extracted features, which will be used for individual finger detection in the next step, we first send the signals to a high pass filter to eliminate the noise caused by electrode-skin impedance and continuous body movements. We use Butterworth high pass filter [2] with a corner frequency of 20 Hz [2]. Then, to extract each finger movements from the filtered signals, we designed a finger movement detection system, whose working flow diagram is shown in Fig. 2.

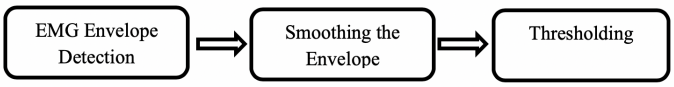

Fig. 2: Movement Extraction Work Flow

The EMG Envelope Detection step returns the upper envelopes of the filtered EMG sequence. The envelope is the magnitude of the analytic signal computed by Hilbert function. We, then, have the signal without any ripple. There are still some sharp variations in the envelope of the EMG signal. It can cause an error in duration calculations. Therefore, the signal envelope is smoothed in the second step. The Smooth function will smooth the data in the column vector by using a moving average filter. In the step of Thresholding, the samples turn to 1 when they are higher than threshold, otherwise they turn to

0. The threshold value is heuristically set by the average value of the EMG signal sequence. By multiplying the thresholded sequence as a duration window, we can access to 'Closed Fingers' and 'Open Hands' samples separately. We can, then, detect the movement patten.

III. Individual finger detection

To detect which finger is moving, we first extract features from each movement signal. These features should contain the most descriptive information about the signal and their size should be reduced in dimension compared to the input signal as a whole. We extracted 17 features from the signal.

We, then, use a machine learning based classifier to identify the individual finger based on the extracted features. Six popular machine learning algorithms are examined for classification including Convolutional Neural Network (CNN) [3], Deep Neural Network (DNN) [3], k-Nearest Neighbor (KNN) [4], Logistic regression (LR) [5], Quadratic Discriminant Analysis (QDA) [6], and XGBoost [7]. We focus on finding out a selection of classification system elements (i.e., feature set, classifier, window characteristics, dimensionality reduction method) for the best performance of individual finger detection.

We performed three rounds of data collection on our testers, giving us three data sets to study and develop a model specified to our testers. These data were first prepossessed for movement extraction then used to train and test a series of classification systems, each of them consists of a different combination of system element choices. Considering the computation power difference of the finger detection devices, we designed a single-layer classifier and a two-layer classifier for less powerful systems and powerful systems, respectively.

1) Single Layer Classifier: By using StratifiedKFold to split the whole data set with a fixed random seed (random state = 1), we ran fivefold cross-validation on each model. We applied Principal Component Analysis (PCA), which is used for dimension reduction, to transform the data set into a new subspace, which makes it more separable. After components are received, we feed the data to the classifiers to do cross-validation. Note that we removed the features that are zeroed out before running PCA. We compared the result of different classifiers with/without using PCA in Table I.

TABLE I: Single Layer Classifier Performance Comparison

5 class classifier	5 fold cross validation total accuracy(%)	
Classifier	Without Using PCA	Using PCA
XGBoost	75.4	77.8
LR	78.03	79.5
CNN	79.6	79.03
DNN	79.9	79.03
QDA	70	79.1
KNN	70.2	69.9

By using PCA, we can combine the information to make them more separable and thus yield better results in classification [8]. Through a series of evaluations, the results and the distribution of the data showed us that the quality and the distribution of the data generally have more impacts on the classification's accuracy than different classifiers.

2) Two-layer classifier: From the confusion matrix of different classifiers and the evaluation results, we observed that the Middle finger has the lowest detection accuracy. To further improve the performance, we designed a two-layer classifier. We have a binary classifier between the Middle finger and the rest of the fingers at the first layer. At the second layer, we have a 4-class classifiers for detecting Thumb, Index, Ring, and Pinkie. TableII shows the average detection accuracy among the three data-set while using different classifiers. We can observe that the best two-layer classifier is DNN with PCA at binary layer and CNN without PCA at 4-class layer. The best average accuracy among three data sets is about 87%

TABLE II: Two-Layer Classifier Performance Comparison

Binary Classifier	5 fold cross validation		4 class classifier	5 fold cross validation	
Classifier	Without Using PCA	Using PCA	Classifier	Without Using PCA	Using PCA
XGBoost	82.8	83.06	XGBoost	84.8	86.23
LR	85.7	85.8	LR	86.7	85.8
CNN	86	84.3	CNN	87.2	86.9
DNN	80.4	87.3	DNN	80.4	85.9
QDA	77.5	75.6	QDA	80.3	83.1
KNN	79.3	79.4	KNN	81.2	80.9

IV. Conclusion and future work

In this article, we briefly introduced our pilot work on detecting finger movement via using nano EMG sensors. We designed an effective system to detect individual finger movement and its movement patten. The current best model is a two-layer classifier, where DNN with PCA at binary layer and CNN without PCA at 4-class layer. Since the classifiers' performance vary along with the change of the quality of received signal, in the future, we will work on improving the quality of the captured signals. Besides, we will focus on using voting models, such as Ensemble, to improve the detection accuracy.

References

[1] W.-H. Yeo, Y.-S. Kim, J. Lee, A. Ameen, L. Shi, M. Li, S. Wang, R. Ma, S. H. Jin, Z. Kang *et al.*, "Multifunctional epidermal electronics printed directly onto the skin," *Advanced Materials*, vol. 25, no. 20, pp. 2773–2778, 2013.

[2] C. Castellini and P. van der Smagt, "Surface emg in advanced hand prosthetics," *Biological cybernetics*, vol. 100, no. 1, pp. 35–47, 2009.

[3] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[4] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[5] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in neural information processing systems*, 2002, pp. 841–848.

[6] A. Silva and A. Stam, "Discriminant analysis." 1995.

[7] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.