Pattern Recognition Letters 119 (2019) 131–138

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface

Wentao Wei[a], Yongkang Wong[b], Yu Du[a], Yu Hu[a], Mohan Kankanhalli[c], Weidong Geng[a,*]

[a] College of Computer Science and Technology, Zhejiang University, Zheda Road No.38, Hangzhou, 310027, China
[b] Smart Systems Institute, National University of Singapore, 21 Heng Mui Keng Terrace, 119613, Singapore
[c] School of Computing, National University of Singapore, 13 Computing Drive, 117417, Singapore

ARTICLE INFO

Article history:
Available online 6 December 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Surface electromyography
Muscle-computer interface
Gesture recognition
Deep learning
Convolutional neural network

ABSTRACT

In muscle-computer interface (MCI), deep learning is a promising technology to build-up classifiers for recognizing gestures from surface electromyography (sEMG) signals. Motivated by the observation that a small group of muscles play significant roles in specific hand movements, we propose a multi-stream convolutional neural network (CNN) framework to improve the recognition accuracy of gestures by learning the correlation between individual muscles and specific gestures with a “divide-and-conquer” strategy. Its pipeline consists of two stages, namely the multi-stream decomposition stage and the fusion stage. During the multi-stream decomposition stage, it first decomposes the original sEMG image into equal-sized patches (streams) by the layout of electrodes on muscles, and for each stream, it independently learns representative features by a CNN. Then during the fusion stage, it fuses the features learned from all streams into a unified feature map, which is subsequently fed into a fusion network to recognize gestures. Evaluations on three benchmark sEMG databases showed that our proposed multi-stream CNN framework outperformed the state-of-the-arts on sEMG-based gesture recognition.

1. Introduction

Muscle-computer interface (MCI) enables human users to naturally interact with computers using sEMG signals. It has been successfully applied in prosthesis control [1], robot control [2], virtual reality [3], and human-machine interaction [4]. The key issue of MCI is how to accurately recognize human gestures or postures via sEMG signals within the desired observational latency.

The conventional framework of sEMG-based gesture recognition system is usually composed of signal detection, signal preprocessing and segmentation, feature extraction, and gesture classification. The features used for classification are usually hand-crafted by human experts, and play an important role in effective and precise gesture recognition. For instance, Phinyomark et al. [5] evaluated 37 time domain and frequency domain sEMG features for gesture recognition and selected 7 features based on observation of scatter plots and mathematical properties. Doswald et al. [6] proposed a modified version of Phinyomark’s feature set by adding statistical features of auto-regressive residue and Hilbert–Huang transform. To recognize gestures, the extracted sEMG features are fed into classifiers, such as linear discriminant analysis (LDA) [7], hidden Markov model (HMM) [8], Gaussian mixture models (GMM) [9] and support vector machine (SVM) [10,11].

Deep learning is a revolutionary machine learning approach that has been successfully applied in many fields, such as image classification [12], speech recognition [13], video classification [14] and human activity recognition [15]. Inspired by its recent success, deep learning based approaches are recently studied to solve sEMG-based gesture recognition problems. For example, Atzori et al. [16] proposed a CNN structure based on LeNet to recognize hand movements in NinaPro database. Geng et al. [17] employed CNN to classify 8 hand movements using instantaneous high-density sEMG images. Compared to conventional classifiers which require hand-crafted sEMG features, a major advantage of CNN is that it is an end-to-end classifier which can automatically learn features from training data without assistance from human experts.

The sEMG signals originating from one muscle can generally be considered to be statistically independent of signals from neighbouring muscles [18], and certain muscles play more important roles in certain hand movements [19]. It means that for each specific gesture, only part of sEMG channels (electrodes) or a small group of muscles have a strong correlation with it, not all sEMG channels. Based on instantaneous sEMG images of 8 hand gestures [17], we observed that only part of sEMG images areas were ac-

[*] Corresponding author.
E-mail address: gengwd@zju.edu.cn (W. Geng).

https://doi.org/10.1016/j.patrec.2017.12.005
0167-8655/© 2017 Elsevier B.V. All rights reserved.

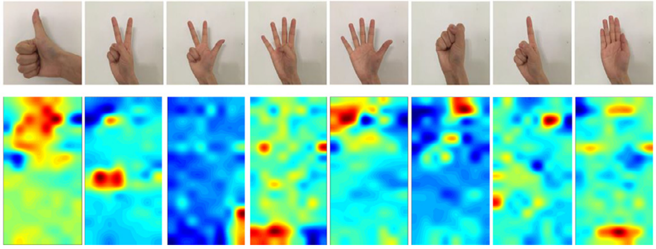

Fig. 1. Instantaneous sEMG images of various hand gestures in CapgMyo DB-a during contraction [17], where the first row are gestures and the second row are the corresponding sEMG images.

tivated during muscle contraction of specific gesture, as shown in Fig. 1. This is consistent with the observation reported in [19].

Motivated by these characteristics in sEMG image, we propose a two-stage multi-stream CNN approach for sEMG-based gesture recognition by a "divide-and-conquer" strategy, aiming at higher recognition accuracy of gestures by learning the correlation between certain muscles for each specific gesture. During the multi-stream decomposition stage, a convolutional neural network is designed to independently learn representative features from each stream, which is built upon a patch that is equally divided from the original sEMG image based on the layout of electrodes on muscles. During the fusion stage, the learned features from all streams are fed into a fusion network to achieve improved gesture recognition accuracy.

The remaining part of this paper is organized as follows. In Section 2, we review the related gesture recognition approaches. In Section 3, we introduce the divide-and-conquer learning mechanism and our proposed multi-stream framework. The detailed description and discussion of our experiments and experimental results on three databases are presented in Section 4. Finally, in Section 5, we conclude with a discussion and delineate our future work.

2. Related works

Gesture recognition is the technical core in natural human computer interaction, Cheok et al. [20] presented a comprehensive review on it, in which gesture recognition approaches can be broadly categorized into vision-based approaches and sensor-based approaches. Vision-based approaches enable gesture recognition through RGB images [21] or 3D depth channels [22] captured by monocular camera [23], stereo-camera [24] or Kinect sensor [21]. There is also growing interest in using multi-modal data for vision-based gesture recognition [25,26]. Sensor-based approaches perform gesture recognition using motion, position and velocity information of hand movements captured by various types of sensors and instruments [20], including inertial measurement unit (IMU) [27,28], bend-sensing data glove [29,30], WiFi [31,32] and sEMG electrodes [33,34].

The sEMG-based gesture recognition is a key technology in MCI. Compared with other gesture recognition approaches, the advantages of MCI include: (1) MCI is robust against occlusion compared with vision-based method (2) MCI is more suitable for outdoor use compared with data glove.

From the point of view of classifier, there are two categories for sEMG-based gesture recognition in MCI. The first one is to employ classical classifiers. Atzori et al. [33] evaluated various classifiers for sEMG-based gesture recognition on their proposed sparse multi-channel NinaPro database, among them random forests achieved the best gesture recognition performance. Amma et al. [34] used high-density sEMG (HD-sEMG) signals recorded by 192 electrodes arranged in a 8×24 array to classify 27 finger gestures, they achieved the recognition accuracy of 90.4% using root mean square features and a naive Bayes classifier. Khushaba et al. [35] employed LDA to evaluate their proposed temporal-spatial features on both sparse multi-channel and high-density sEMG databases.

The second category is based on deep learning. For instance, Atzori et al. [16] proposed a modified version of LeNet to classify hand movements in the NinaPro database, the average classification accuracy was $66.59 \pm 6.40\%$. Geng et al. [17] applied CNN-based approach to gesture recognition using instantaneous sEMG images, and the achieved recognition accuracies were 77.8% on NinaPro database, 96.8% on csl-hdemg database and 99.5% on CapgMyo database. Other CNN-based studies for sEMG-based gesture recognition mainly focused on improving the scalability of classifier through deep-domain adaptation [36] or self-recalibrating classification [37].

The work in this paper falls into the second category, which is based on deep learning. Being different with the existing single-stream deep learning approaches, we take into consideration the observation that a small group of muscles play significant roles in specific gestures [19], and accordingly propose a multi-stream divide-and-conquer CNN framework for a higher accuracy of gesture recognition in MCI.

3. Proposed multi-stream CNN framework

In this paper, we formulate the sEMG-based gesture recognition problem as a CNN based image classification problem, and use sEMG image $\boldsymbol{x} \in \mathbb{R}^{W \times H}$ as the input of CNN, where W and H are the width and height of $\boldsymbol{x}$, respectively. The formation of $\boldsymbol{x}$ depends on the type of sEMG signals. For HD-sEMG (e.g., CapgMyo database [17] and csl-hdemg database [34]), which is often recorded at a high sampling rate, gesture recognition is based on instantaneous sEMG images and majority voting is employed to determine the gesture label of a sequence of sEMG images. Thus, the size of $\boldsymbol{x}$ (i.e., W and H) is exactly same as the size of the electrode array used to collected HD-sEMG. For sparse multi-channel sEMG (e.g., NinaPro database [33]) which is usually acquired at lower sampling rate, instantaneous sEMG images and simple majority voting may not sufficiently capture temporal information among multiple frames, so a time window is employed to sample sEMG signals, and sEMG signals recorded by C channels within a L-frame time window are converted to a sEMG image $\boldsymbol{x} \in \mathbb{R}^{L \times C}$.

There has been growing interest in using multi-stream CNN for vision-based gesture recognition. For example, Molchanov et al. [38] used a 3D CNN for gesture recognition using intensity and depth data, in which the classifier consisted of two streams including a high-resolution network and a low-resolution network. Pigou et al. [39] used 2 streams of CNN for gesture recognition using Microsoft Kinect, one for extracting hand features and one for extracting upper body features.

Divide-and-conquer is an effective learning strategy that has been successfully applied to classical classifiers such as multi-layer perceptron (MLP) [40,41] and SVM [42]. Most of the existing divide-and-conquer learning approaches perform divide-and-conquer on sample space [40–42], which partition the input instances into small subsets and recursively conquer them. In contrast, Guo et al. [43] proposed a divide-and-conquer classification approach on feature-space. Specifically, the original feature-space was first decomposed into several subspaces via linear transformation, then the decomposed subspaces were used to train local classifiers, and finally, the outputs of local classifiers were fused together to obtain the final classification results.

Motivated by the divide-and-conquer classification approach proposed by [43] and multi-stream CNN in vision-based gesture recognition [38,39], we embed a divide-and-conquer mechanism

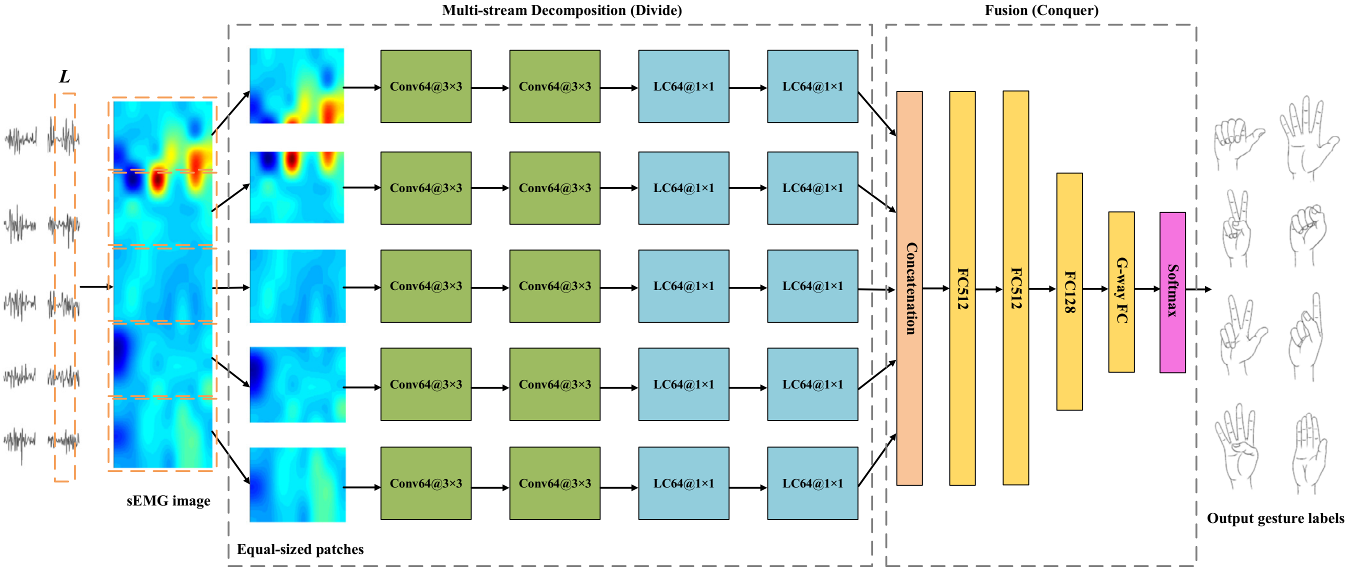

Fig. 2. Conceptual diagram of our proposed multi-stream divide-and-conquer framework. The input of the framework are sEMG signals recorded by C channels within a L-frame time window ($L = 1$ for HD-sEMG). The multi-stream CNN and the fusion network are in gray dashed boxes. Conv, LC and FC denote convolution layer, locally-connected layer and fully-connected layer, respectively. The number after the layer name denotes the number of filters, and the numbers after @ denote convolution kernel size.

in a multi-stream CNN for the sEMG-based gesture recognition pipeline. Fig. 2 shows the conceptual diagram of our proposed multi-stream divide-and-conquer framework. The pipeline is composed of two stages, including the multi-stream decomposition (divide) stage and the fusion (conquer) stage.

During the multi-stream decomposition stage, the original sEMG image $\boldsymbol{x} \in \mathbb{R}^{W \times H}$ is first decomposed into M equal-sized patches $\{\boldsymbol{x}'_i \in \mathbb{R}^{w' \times h'}\}_{i=1}^{M}, M > 1$ based on the layout of electrodes on muscles. For the convenience of presentation, we illustrated only five patches (i.e., $M = 5$) in Fig. 2. Then, each patch $\boldsymbol{x}'_i$ is used as the input of a single-stream CNN. Each stream consists of four layers. The first two layers are convolutional layers, each of them consists of 64 2D filters of 3×3 with the stride of 1 and a zero padding of 1. The two convolutional layers are followed by two locally-connected layers, each of them consists of 64 non-overlapping 2D filters of 1×1. We apply ReLU non-linearity functions [12] followed by batch normalization after each convolution layer, locally-connected layer. There is dropout with a probability of 0.5 after the last locally-connected layers of each stream.

During the fusion stage, the learned feature maps of all M streams, $\{\boldsymbol{s}'_i \in \mathbb{R}^{w' \times h'}\}_{i=1}^{M}$, are fused into a unified feature map $\boldsymbol{s} \in \mathbb{R}^{W \times H}$, which has the same size as the original sEMG image. Then batch normalization is applied to $\boldsymbol{s}$, and subsequently fed the normalized $\boldsymbol{s}$ into a fusion network for gesture recognition. The fusion network consists of four fully-connected layers. The first two hidden layers consist of 512 units respectively. The third hidden layer consists of 128 units. Finally, there is a G-way fully-connected layer and a softmax classifier at the end of the fusion network, where G equals to the number of gestures. ReLU non-linearity functions [12] followed by batch normalization are applied after each fully-connected layer, and there is dropout with a probability of 0.5 after the first and second fully-connected layers of the fusion network.

4. Experiments and results

4.1. Experimental setup

We carried out evaluations on three benchmark sEMG databases, including NinaPro database [33], csl-hdemg database [34] and CapgMyo database [17].

The proposed multi-stream CNN were implemented with MxNet [44]. The CNN was trained using stochastic gradient descent (SGD) with a batch size of 1000 and a weight decay of 0.0001 for all experiments. The weights of the CNN were initialized as described in [45]. For experiments on NinaPro database and CapgMyo database, there were a total of 28 epochs in each fold of the cross-validation, the learning rate was initialized at 0.1 and was divided by 10 after the 16th and 24th epochs. For experiments on csl-semg database, there were a total of 10 epochs in each fold of the cross-validation, the learning rate was initialized at 0.1 and was divided by 10 after the 4th and 8th epochs. During the training phase, we applied the same pre-training strategy as proposed in [17]. Specifically, the CNN was pre-trained on the union of the training sets of all subjects in each round.

In our experiments, the observational latency for gesture recognition is set to 300 ms, as the response time of an real-time system should be kept below 300 ms to avoid a time delay perceived by the user [7,46]. Thus, for evaluations on NinaPro database which used a time window to sample the raw sEMG signals, we used time windows of 50 ms, 100 ms, 150 ms and 200 ms. For evaluations on csl-hdemg database and CapgMyo database which used majority voting to determine gesture label, we used majority voting windows ranging from 1 ms to 300 ms.

4.2. Evaluations on NinaPro database

The NinaPro database [33] consists of three sub-databases. In this paper we evaluated our proposed multi-stream CNN by recognition of 52 hand gestures in the first sub-database (denoted as DB-1). NinaPro DB-1 contains 10-channel sparse multi-channel sEMG signals recorded from 27 subjects. Each hand gesture was recorded with 10 trials at a sampling rate of 100 Hz. Details of this database can be found in [33].

For experiments on NinaPro database we followed the same cross-validation scheme as described in previous studies [16,17,33]. Specifically, for each subject, the CNN was trained on approximately two thirds of the trials and tested on the remaining trials. Previous studies on NinaPro database used a 1st order 1 Hz low-pass Butterworth filter as the data preprocessing procedure [17,33], the same configuration was applied here.

Table 1
Experimental results on NinaPro DB-1 using 200 ms time windows. Results in **bold** entries indicate best performance.

Experiments	Number of streams	Classification accuracy (%)
Channel-wise-MS	10	**85.0%**
Hybrid-MS	3	81.6%
20 × 10-SS	1	82.9%
20 × 10 × 1-SS	1	83.3%
Frame-wise-MS	20	79.4%
Adjacent-frame-MS	19	79.2%

Table 2
Comparison with existing works on NinaPro DB-1, our proposed method is denoted as "multi-stream CNN".

Method	Window length (ms)	Classification accuracy (%)
Random forests [33]	200	75.3%
Single-stream CNN [17]	200	77.8%
Single-stream CNN [16]	150	66.59 ± 6.40%
Multi-stream CNN	200	85.0%
Multi-stream CNN	150	84.4%
Multi-stream CNN	100	83.4%
Multi-stream CNN	50	81.7%

We converted the sEMG signals recorded by each individual channel within a L-frame time window to a $L \times 1$ sEMG image. Each single-stream CNN was trained on sEMG images recorded by each individual channel. The available 10 channels yielded 10 streams. The above-mentioned multi-stream (MS) strategy was denoted as channel-wise-MS hereafter.

According to [33], eight electrodes were equally-spaced around the forearm, two electrodes were placed on the main activity spots of the flexor digitorum superficialis and of the extensor digitorum superficialis. Therefore, we converted sEMG signals recorded by the 8 equally-spaced electrodes within a L-frame time window to a $L \times 8$ sEMG image, and sEMG signals recorded by electrodes placed on the main activity spots of the flexor digitorum superficialis and of the extensor digitorum superficialis within a L-frame time window to a $L \times 1$ sEMG image, respectively. Thus, this strategy yielded a total of three streams, which was denoted as hybrid-MS. In this experiment we used $L = 20$, which is equivalent to a 200 ms time window.

Additionally, we also evaluated two single-stream approaches and two multi-stream approaches, they are described as:

- **20 × 10-SS:** converting sEMG signals recorded by all 10 electrodes within a 20-frame time window to a single-channel image of 20 × 10, and then used it as the input of a single-stream (SS) CNN;
- **20 × 10 × 1-SS:** converting sEMG signals recorded by all 10 electrodes within a 20-frame time window to a multi-channel image of 20 × 10 × 1, where the depth of the image equals to 20, and used it as the input of a single-stream (SS) CNN;
- **frame-wise-MS:** converting each frame of sEMG signals recorded by all 10 electrodes to a single-channel image of 1 × 10, and then used it as the input of each stream in a multi-stream (MS) CNN, a total of 20 streams;
- **adjacent-frame-MS:** converting every two adjacent frames of sEMG signals recorded by all 10 electrodes to a single-channel image of 2 × 10, and then used it as the input of each stream in a multi-stream (MS) CNN, a total of 19 streams.

For experiments on single-stream CNN, the CNN is the same as each individual stream in our proposed multi-stream CNN. Same configurations were applied to experiments on the other two databases (i.e., csl-hdemg and CapgMyo).

Table 1 shows the average classification accuracies on NinaPro DB-1 using a time window of 200 ms. It shows that the channel-wise-MS strategy, which trained each stream on sEMG images recorded by each individual channel, outperformed single-stream CNN approaches (i.e., 20 × 10-SS and 20 × 10 × 1-SS). We also noticed that for gesture recognition using single-stream CNN, converting sEMG signals to multi-channel image which contains temporal information in its depth (i.e., 20 × 10 × 1-SS) is better than converting sEMG signals to single-channel 2D image (i.e., 20 × 10-SS).

Table 2 shows the comparison with existing works on NinaPro DB-1. The author of NinaPro database achieved 75.3% classification accuracy using random forests [33] and 66.59 ± 6.40% classification accuracy using single-stream CNN [16]. The single stream CNN approach with instantaneous sEMG images [17] achieved 77.8% classification accuracy with majority voting. Using the channel-wise-MS strategy, the classification accuracy achieved by multi-stream CNN using a 200 ms time window was 85%, which was currently the highest one on NinaPro using the intra-subject cross-validation scheme described in [33].

We noticed that the CNN in [17] was trained on instantaneous sEMG images, thus the CNN can not not make full use of temporal information among multiple consecutive frames. For a fair comparison, an additional experiment was performed using the CNN implementation provided in [17] to classify multi-channel images that contained temporal information in their depth, which are the same as the input images of 20 × 10 × 1-SS. The classification accuracy reached 83.5% using a 200 ms time window, which was still 1.5 percentage point lower than the result achieved by our proposed divide-and-conquer multi-stream strategy (i.e., channel-wise-MS).

4.3. Evaluations on csl-hdemg database

The csl-hdemg database [34] contains HD-sEMG signals of 27 finger gestures performed by 5 subjects. The database consists of 5 recording sessions, and each gesture was recorded with 10 trials in each recording session. The sEMG signals were recorded at a sampling rate of 2048 Hz using an electrode array with 192 electrodes, arranged in a 8 × 24 grid with an inter-electrode distance of 10 mm. As every eighth channel in csl-hdemg does not contain meaningful data due to bipolar recordings [34], 168 channels (7 × 24) were used in our evaluation.

For experiments on csl-hdemg, we adopted the session-wise leave-one-out cross-validation scheme described in [17,34]. Specifically, for each recording session, each of the 10 trials was used in turn as the test set and the remaining 9 trials were used as the training set.

The evaluation on csl-hdemg was based on instantaneous sEMG images. We divided the original 7 × 24 instantaneous sEMG image into several equal-sized patches to train our proposed multi-stream CNN. We evaluated the following variants of input sEMG image patches:

- **7 × 24-SS:** using the original 7 × 24 sEMG image as the input of a single-stream CNN, this is the baseline by a single-stream CNN;
- **7 × 1-patch-wise-MS:** equally dividing the original 7 × 24 sEMG image into 24 non-overlapping patches of size 7 × 1, a total of 24 streams;
- **7 × 2-patch-wise-MS:** equally dividing the original 7 × 24 sEMG image into 12 non-overlapping patches of size 7 × 2, a total of 12 streams;
- **7 × 4-patch-wise-MS:** equally dividing the original 7 × 24 sEMG image into 6 non-overlapping patches of size 7 × 4, a total of 6 streams;
- **7 × 8-patch-wise-MS:** equally dividing the original 7 × 24 sEMG image into 3 non-overlapping patches of size 7 × 8, a total of 3

Table 3
Classification accuracies (%) achieved on csl-hdemg database. Results in **bold** entries indicate best performance.

Experiments	Number of streams	Size of input patches per stream	Single frame	The length of voting window (ms)	
				150	300
7×24-SS	1	7×24	89.5%	93.1%	95.1%
7×1-patch-wise-MS	24	7×1	89.5%	92.9%	94.9%
7×2-patch-wise-MS	12	7×2	89.6%	93.0%	94.9%
7×4-patch-wise-MS	6	7×4	89.8%	93.1%	95.0%
7×8-patch-wise-MS	3	7×8	**90.3%**	**93.6%**	**95.4%**
1×24-patch-wise-MS	7	1×24	90.0%	93.4%	95.2%
2×24-patch-wise-MS	6	2×24	90.2%	**93.6%**	**95.4%**

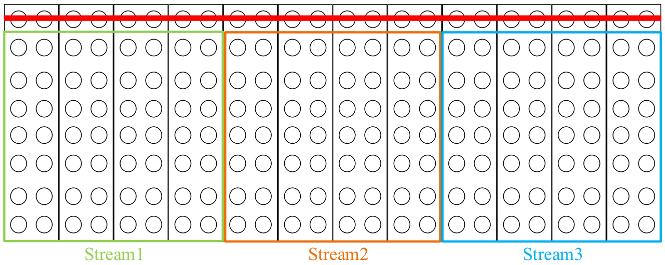

Fig. 3. Outline of the multi-stream decomposition strategy of **7 × 8**-patch-wise-MS, where the original sEMG image was equally divided into three patches of size **7 × 8**. The red strikethrough indicated that every eighth channel was ignored in our experiments. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

streams (shown in Fig. 3, where different streams are in different colours);

- **1 × 24-patch-wise-MS:** equally dividing the original 7×24 sEMG image into 7 non-overlapping patches of size 1×24, a total of 7 streams;
- **2 × 24-patch-wise-MS:** equally dividing the original 7×24 sEMG image into 6 overlapping patches of size 2×24 with 50% overlapping area between every two adjacent patches, a total of 6 streams.

For data preprocessing, we applied the same signal segmentation approach as described in [17,34]. To remove the noise in the signal, the 1st order 1 Hz low-pass Butterworth filter was utilized as that in NinaPro [33].

Table 3 shows the average single frame accuracies, and majority voting accuracies over 150ms and 300ms of various experiments on csl-hdemg. Compared to the baseline by a single-stream CNN (i.e., 7×24-SS), our proposed multi-stream CNN achieved better classification performance in three out of the six multi-stream decomposition strategies (i.e., 7×8-patch-wise-MS, 1×24-patch-wise-MS and 2×24-patch-wise-MS).

Two multi-stream decomposition strategies were recommended in this paper. The first one is 7×8-patch-wise-MS, which equally divided the original 7×24 sEMG image into 3 non-overlapping patches, each patch is of size 7×8. The resulting gesture recognition accuracy is 90.3% based on a single frame of HD-sEMG, and 93.6%, and 95.4% using simple majority voting over 150 ms and 300 ms, respectively. The second one is 2×24-patch-wise-MS, which equally divided the original 7×24 sEMG image into 6 overlapping patches, each patch is of size 2×24. Its performance on gesture recognition is very close to that in 7×8-patch-wise-MS.

Amma et al. [34] reported the recognition accuracy of 90.4% using a naive Bayes classifier with features extracted from the entire segment over each trial. However, from the point of view of MCI, existing works seldom perform gesture recognition over the entire trial due to the constraint of maximal response time (i.e., 300 ms) for real-time system [7,46]. From Table 3 we can see that using majority voting over 300 ms, our proposed method achieved an accuracy of 95.4%, which is even higher than the per-trial accuracy reported in [34].

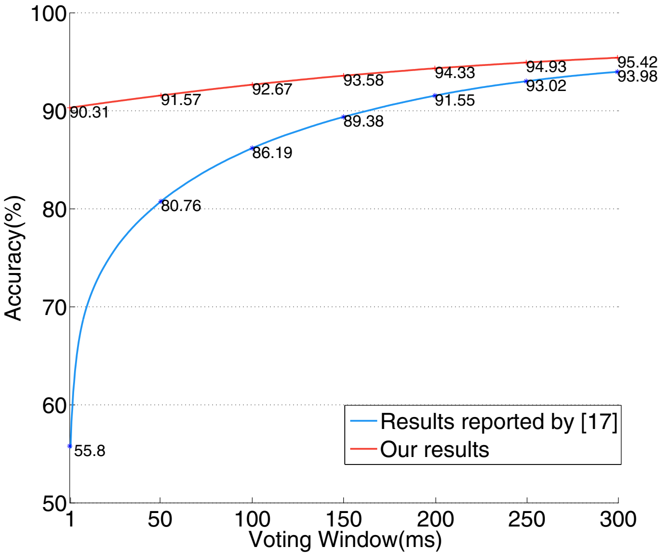

Fig. 4. Performance comparison with reported results in [17] on csl-hdemg.

To further validate the effectiveness of our proposed multi-stream CNN on smaller time windows, we compare results achieved by our proposed approach (i.e., 7×8-patch-wise-MS) with results achieved in [17] using different voting windows ranging from 1 ms to 300 ms. As shown in Fig. 4, the performance gap between our approach and the method proposed in [17] was significant. This indicates that our proposed method is more suitable for real-time MCI with less observational latency.

4.4. Evaluations on CapgMyo database

We evaluated our proposed recognition framework on DB-a of CapgMyo [17], which contains sEMG signals of 8 hand gestures performed by 18 subjects, each gesture was recorded 10 trials. The sEMG signals were recorded at a sampling rate of 1000 Hz. The sensors used in building CapgMyo consists of 8 electrode arrays, each of them has the size of 8×2. The 8 electrode arrays were fixed around the right forearm, arranged in a grid of 8×16 (shown in Fig. 5).

For experiments on CapgMyo DB-a, the CNN were trained on half of the trials and tested on the remaining trials, which was same as that described in [17]. Following the classification procedure described in [17], we used the instantaneous sEMG images to train our CNN, and employed majority voting to calculate accuracies over 40 ms and 150 ms, respectively.

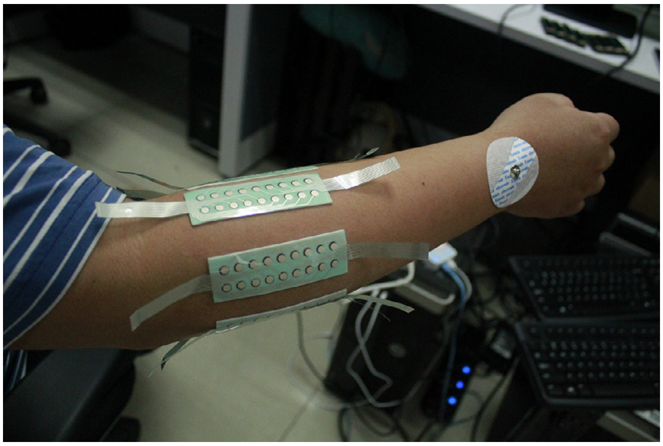

Fig. 5. The layout of electrode arrays in CapgMyo database.

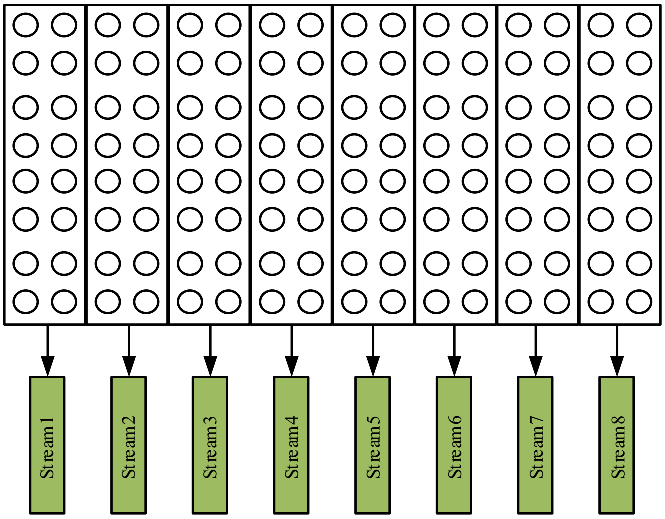

Fig. 6. Outline of the multi-stream strategy used in **8 x 2**-patch-wise-MS, where each stream was trained on sEMG images recorded by each individual electrode array.

We evaluated the following variants of input sEMG image patches for each stream based on the size and layout of electrode arrays:

- **8 x 16-SS:** using the 8 × 16 sEMG image recorded by all 8 electrode arrays to train a single-stream CNN;
- **8 x 2-array-wise-MS:** a total of 8 streams, each stream is built upon the 8 × 2 sEMG image recorded by each individual electrode array (shown in Fig. 6);
- **8 x 4-array-wise-MS:** a total of 4 streams, each stream is built upon the 8 × 4 sEMG image recorded by every two electrode arrays;
- **8 x 8-array-wise-MS:** a total of 2 streams, each stream is built upon the 8 × 8 sEMG image recorded by every four electrode arrays;
- **1 x 16-patch-wise-MS:** a total of 8 streams, each stream is built upon the 1 × 16 patch divided from the original 8 × 16 sEMG image;
- **2 x 16-patch-wise-MS:** a total of 7 streams, each stream is built upon the 2 × 16 patch divided from the original 8 × 16 sEMG image, and there is 50% overlapping area between the input patches of every two adjacent streams.

Table 4 shows the average single frame accuracies, and the majority voting accuracies over 40 ms and 150 ms achieved by various experiments on CapgMyo. Three multi-stream decomposition strategies (i.e., 8 × 2-array-wise-MS, 8 × 4-array-wise-MS and 8 × 8-array-wise-MS) outperformed the baseline result achieved by a single-stream CNN (i.e., 8 × 16-SS). Among them, the strategy that trained each stream on 8 × 2 sEMG images recorded by each individual electrode array (i.e., 8× 2-array-wise-MS) achieved the highest gesture recognition accuracy, and it is recommended here as the best configuration.

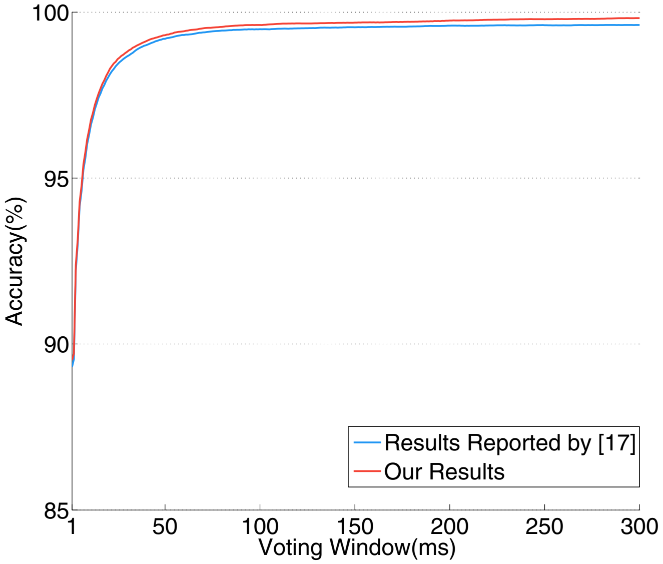

Fig. 7. Performance comparison with reported results in [17] on CapgMyo DB-a.

Fig. 7 compares results achieved by our proposed approach (i.e., 8 × 2-array-wise-MS) with those achieved by [17] using various voting windows ranging from 1ms to 300 ms on CapgMyo. Our approach achieved the recognition accuracies of 89.5% based on single frame of instantaneous sEMG image, 99.1%, 99.7% and 99.8% over majority voting of 40 ms, 150 ms and 300 ms, respectively. By comparison, the recognition accuracies in [17] were 89.3% based on single frame of instantaneous sEMG image, 99.0%, 99.5% and 99.6% by majority voting over 40 ms, 150 ms and 300 ms, respectively.

5. Discussion and conclusion

Aiming at a higher accuracy of sEMG-based gesture recognition, we proposed a multi-stream divide-and-conquer framework. Its CNN for each stream was trained independently based on equal-sized patches divided from the original sEMG image during the multi-stream decomposition stage, and fused together to obtain the final classification result during the fusion stage, thus the correlation between individual muscles and specific gestures can be thoroughly learned. The resulting accuracy of sEMG-based gesture recognition was accordingly improved, which shows the effectiveness of our proposed method.

We evaluated different multi-stream decomposition strategies using the desired time windows or majority voting windows in MCI [7,46] on three sEMG databases. On NinaPro database which contains sparse multi-channel sEMG, the highest gesture recognition accuracy using 200 ms time windows is 85.0%, which was achieved by the strategy that trained each stream on sEMG images recorded by each individual channel (i.e., channel-wise-MS). On the other hand, the hybrid-MS strategy did not achieve satisfactory performance, which indicates that even for equally-spaced electrodes, considering sEMG signals recorded by each channel independently is better than considering sEMG signals recorded by all channels as a whole.

Table 4
Classification accuracies (%) achieved on CapgMyo DB-a. Results in **bold** entries indicate best performance.

Experiments	Number of streams	Size of input patches per stream	Single frame	The length of voting window (ms)	
				40	150
8 × 16-SS	1	8 × 16	89.2%	99.0%	99.5%
8 × 2-array-wise-MS	8	8 × 2	**89.5%**	**99.1%**	**99.7%**
8 × 4-array-wise-MS	4	8 × 4	89.4%	**99.1%**	99.6%
8 × 8-array-wise-MS	2	8 × 8	89.4%	**99.1%**	99.6%
1 × 16-patch-wise-MS	8	1 × 16	84.2%	98.3%	99.1%
2 × 16-patch-wise-MS	7	2 × 16	87.9%	98.8%	99.3%

On csl-hdemg database which contains HD-sEMG, the highest gesture recognition accuracy was achieved by the strategy that trained each stream on 7 × 8 patches (i.e., 7 × 8-patch-wise-MS). The accuracy on a 300 ms voting windows reached 95.4%, which is much higher than the per-trial accuracy reported in [34].

On the other HD-sEMG datbase named CapgMyo, the highest gesture recognition accuracy was achieved by the strategy that trained each stream on 8 × 2 sEMG images recorded by each individual electrode array (i.e., 8 × 2-array-wise-MS), which is higher than that reported in [17]. On the other hand, the 1 × 16-patch-wise-MS and 2 × 16-patch-wise-MS strategies achieved the lowest classification accuracies among all experiments, which means the sEMG images recorded by one circle of 16 electrodes around forearm contain less movement information than the 8 × 2 sEMG images recorded by each electrode array. A possible explanation to this is that as the electrode arrays used in CapgMyo were manually fixed around the right forearm, they were not strictly equally spaced, thus performing convolution on sEMG images recorded by one or two circles of electrodes around forearm could not correctly capture the spatial correlation among different electrodes.

Our proposed multi-stream divide-and-conquer framework may be improved by the following approaches: (1) exploring more sophisticated fusion algorithms for multi-stream fusion (2) expansion to multi-view multi-label gesture recognition (3) applying temporal models. Firstly, during the fusion stage of our current work, the learned features produced by all streams are fused into a unified feature map by a simple concatenation operation. In the future we will explore more sophisticated fusion algorithms. Secondly, each gesture can be decomposed into a group of sub-movements, and sEMG signals recorded in different experimental sessions can be regarded as multi-view data. Following recent progress in multi-view classification [47] and heterogeneous representation learning [48], in the future we will expand current work to multi-view multi-label gesture recognition, and carry out further research in this area. Thirdly, there is a limitation of CNN for sEMG-based gesture recognition, because sEMG signals are in essence temporal data sequence. To overcome this limitation, we will adopt novel temporal models that have been applied to vision-based gesture recognition, such as sequentially-supervised long short term memory (SS-LSTM) [21], to improve current work.

Acknowledgments

This work was supported by grants from the National Key Research and Development Program of China (No. 2016YFB1001302), the National Natural Science Foundation of China (No. 61379067), and the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centre in Singapore Funding Initiative.

References

[1] C. Cipriani, F. Zaccone, S. Micera, M.C. Carrozza, On the shared control of an EMG-controlled prosthetic hand: analysis of user-prosthesis interaction, IEEE Trans. Robot. 24 (1) (2008) 170–184.

[2] B. Wang, C. Yang, Q. Xie, Human-machine interfaces based on EMG and Kinect applied to teleoperation of a mobile humanoid robot, in: Proceedings of the World Congress on Intelligent Control and Automation, 2012, pp. 3903–3908.

[3] F. Muri, C. Carbajal, A.M. Echenique, H. Fernández, N.M. López, Virtual reality upper limb model controlled by EMG signals, J. Phys. Conf. Ser. 477 (1) (2013) 12041–12048.

[4] J. Cheng, X. Chen, Z. Lu, L. Wang, M. Shen, Key-press gestures recognition and interaction based on sEMG signals, in: Proceedings of the International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction, 2010, pp. 1–4.

[5] A. Phinyomark, P. Phukpattaranont, C. Limsakul, Feature reduction and selection for EMG signal classification, Expert. Syst. Appl. 39 (8) (2012) 7420–7431.

[6] A. Doswald, F. Carrino, F. Ringeval, Advanced processing of sEMG signals for user independent gesture recognition, in: Proceedings of the Mediterranean Conference on Medical and Biological Engineering and Computing, 2013, pp. 758–761.

[7] K. Englehart, B. Hudgins, A robust, real-time control scheme for multifunction myoelectric control, IEEE Trans. Biomed. Eng. 50 (7) (2003) 848–854.

[8] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, J. Yang, A framework for hand gesture recognition based on accelerometer and EMG sensors, IEEE Trans. Syst. Man Cybern. A Syst. Humans 41 (6) (2011) 1064–1076.

[9] Z. Ju, H. Liu, A generalised framework for analysing human hand motions based on multisensor information, in: Proceedings of the IEEE International Conference on Fuzzy Systems, 2012, pp. 1–6.

[10] M.A. Oskoei, H. Hu, Support vector machine-based classification scheme for myoelectric control applied to upper limb, IEEE Trans. Biomed. Eng. 55 (8) (2008) 1956–1965.

[11] A.H. Al-Timemy, G. Bugmann, J. Escudero, N. Outram, Classification of finger movements for the dexterous hand prosthesis control with surface electromyography, IEEE J. Biomed. Health. Inform. 17 (3) (2013) 608–618.

[12] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Annual Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.

[13] O. Abdel-Hamid, A.R. Mohamed, H. Jiang, L. Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition, IEEE ACM Trans. Audio Speech Lang. Process. 22 (10) (2014) 1533–1545.

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, F. Li, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[15] J. Yang, M.N. Nguyen, P.P. San, X. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2015, pp. 3995–4001.

[16] M. Atzori, M. Cognolato, H. Müller, Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands, Front. Neurorobot. 10 (2016) 9.

[17] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, J. Li, Gesture recognition by instantaneous surface EMG images, Sci. Rep. 6 (2016) 36571.

[18] G.R. Naik, D.K. Kumar, H. Weghorn, M. Palaniswami, Subtle hand gesture identification for HCI using temporal decorrelation source separation BSS of surface EMG, in: Proceedings of the Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, 2007, pp. 30–37.

[19] Y.Y. Huang, K.H. Low, H.B. Lim, Objective and quantitative assessment methodology of hand functions for rehabilitation, in: Proceedings of the IEEE International Conference on Robotics and Biomimetics, 2008, pp. 846–851.

[20] M.J. Cheok, Z. Omar, M.H. Jaward, A review of hand gesture and sign language recognition techniques, Int. J. Mach. Learn. Cyb. (2017) 1–23.

[21] P. Wang, Q. Song, H. Han, J. Cheng, Sequentially supervised long short-term memory for gesture recognition, Cogn. Comput. 8 (5) (2016) 982–991.

[22] C. Yang, D.K. Han, H. Ko, Continuous hand gesture recognition based on trajectory shape information (in press), Pattern. Recognit. Lett. (2017). https://doi.org/10.1016/j.patrec.2017.05.016.

[23] Y. Zhou, G. Jiang, Y. Lin, A novel finger and hand pose estimation technique for real-time hand gesture recognition, Pattern. Recognit. 49 (2016) 102–114.

[24] K. Liu, N. Kehtarnavaz, Real-time robust vision-based hand gesture recognition using stereo images, J. Real Time. Image. Pr. 11 (1) (2016) 201–209.

[25] J. Wu, J. Cheng, Bayesian co-boosting for multi-modal gesture recognition, J. Mach. Learn. Res. 15 (1) (2014) 3013–3036.

[26] D. Wu, L. Pigou, P.J. Kindermans, N.D.H. Le, L. Shao, J. Dambre, J.M. Odobez, Deep dynamic neural networks for multimodal gesture segmentation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 38 (8) (2016) 1583–1597.

[27] D. Iyer, F. Mohammad, Y. Guo, E.A. Safadi, B.J. Smiley, Z. Liang, N.K. Jain, Generalized hand gesture recognition for wearable devices in iot: application and implementation challenges, in: Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition, 2016, pp. 346–355.

[28] Y. Zhang, W. Liang, J. Tan, Y. Li, Z. Zeng, Pca & hmm based arm gesture recognition using inertial measurement unit, in: Proceedings of the International Conference on Body Area Networks, 2013, pp. 193–196.

[29] J.S. Kim, W. Jang, Z. Bien, A dynamic gesture recognition system for the korean sign language (ksl), IEEE Trans. Syst. Man. Cybern. B Cybern. 26 (2) (1996) 354–359.

[30] M. Simão, P. Neto, O. Gibaru, Natural control of an industrial robot using hand gesture recognition with neural networks, in: Proceedings of the Annual Conference of the IEEE Industrial Electronics Society, 2016, pp. 5322–5327.

[31] H. Abdelnasser, M. Youssef, K.A. Harras, Wigest: A ubiquitous wifi-based gesture recognition system, in: Proceedings of the IEEE Conference on Computer Communications, 2015, pp. 1472–1480.

[32] Q. Pu., S. Gupta, S. Gollakota, S. Patel, Whole-home gesture recognition using wireless signals, in: Proceedings of the Annual International Conference on Mobile Computing & Networking, 2013, pp. 27–38.

[33] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.G.M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, H. Müller, Electromyography data for non-invasive naturally–controlled robotic hand prostheses, Sci. Data. 1 (2014) 140053.

[34] C. Amma, T. Krings, J. Böer, T. Schultz, Advancing muscle-computer interfaces with high-density electromyography, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 929–938.

[35] R.N. Khushaba, A.H. Al-Timemy, A. Al-Ani, A. Al-Jumaily, A framework of temporal-spatial descriptors based feature extraction for improved myoelectric pattern recognition, IEEE Trans. Neural. Syst. Rehabil. Eng. (99) (2017). 1–1

[36] Y. Du, W. Jin, W. Wei, Y. Hu, W. Geng, Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation, Sensors. 17 (3) (2017) 458.

[37] X. Zhai, B. Jelfs, R.H.M. Chan, C. Tin, Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network, Front. Neurosci. 11 (2017) 379.

[38] P. Molchanov, S. Gupta, K. Kim, J. Kautz, Hand gesture recognition with 3d convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 1–7.

[39] L. Pigou, S. Dieleman, P.J. Kindermans, B. Schrauwen, Sign language recognition using convolutional neural networks, in: Proceedings of the Workshop at the European Conference on Computer Vision, 2015, pp. 572–578.

[40] H. Fu, Y. Lee, C. Chiang, H. Pao, Divide-and-conquer learning and modular perceptron networks, IEEE Trans. Neural Netw. 12 (2) (2001) 250–263.

[41] D. Frosyniotis, A. Stafylopatis, A. Likas, A divide-and-conquer method for multi-net classifiers, Pattern. Anal. Appl. 6 (1) (2003) 32–40.

[42] C. Hsieh, S. Si, I. Dhillon, A divide-and-conquer solver for kernel support vector machines, in: Proceedings of the International Conference on Machine Learning, 1, 2014, pp. 566–574.

[43] Q. Guo, B.W. Chen, S. Rho, W. Ji, F. Jiang, X. Ji, S.Y. Kung, Efficient divide-and–conquer classification based on parallel feature-space decomposition for distributed systems, IEEE Syst. J. PP (99) (2015) 1–7.

[44] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, Z. Zhang, Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems, Advances in Neural Information Processing Systems, Workshop on Machine Learning Systems, 2015.

[45] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[46] B. Hudgins, P. Parker, R.N. Scott, A new strategy for multifunction myoelectric control, IEEE Trans. Biomed. Eng. 40 (1) (1993) 82–94.

[47] M. Kan, S. Shan, X. Chen, Multi-view deep network for cross-view classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4847–4855.

[48] P. Yang, Q. Tan, Y. Zhu, J. He, Heterogeneous representation learning with separable structured sparsity regularization, Knowl. Inf. Syst. (2017) 1–24.