Improving Gesture Recognition By Bidirectional Temporal Convolutional Netwoks

Haoyu Chen[1], Yue Zhang[1], Dalin Zhou[2], and Honghai Liu[2]

[1] College of Computer Science and Technology,
Zhejiang University of Technology,
Hangzhou, China
chyhave@163.com, zhangyuemessi@163.com
[2] School of Computing, University of Portsmouth,
Portsmouth, United Kingdom
dalin.zhou@port.ac.uk, honghai.liu@port.ac.uk

Abstract. Surface electromyography(sEMG) based gesture recognition as an important role in Muscle-Computer interface has been researched for decades. Recently, deep learning based method has had a profound impact on this field. CNN, RNN and RNN-CNN based methods were studied by many researchers. Motivated by Bidirectional Long short-term memory(Bi-LSTM) and Temporal Convolutional Networks(TCN), we propose 1D CNN based networks called Bidirectional Temporal Convolutional Networks(Bi-TCN). The positive order signal and reverse order sEMG signal are feed to our networks to learn the different representation of the same sEMG signal. We evaluate proposed networks on two benchmark datasets, Ninapro DB1 and DB5. Our networks yields 90.74% prediction accuracy on DB1 and 90.06% prediction accuracy on DB5. The results demonstrate our networks is comparable to the state-of-the-art works.

Keywords: sEMG, Deep Learning, CNN, TCN, LSTM, Gesture Recognition

1 Introduction

Human-Computer interface system requires natural and intuitive method of interaction and can't cause too much confusion for the human users. Surface Electromyography(sEMG) naturally becomes an important role for muscle-computer interaction, which can be easily collected by non-invasion electrodes and contain abundant information about human intention of movement. Therefore, sEMG based gesture recognition has been widely researched in many fields, such as Virtual Reality(VR) interaction [27], Assistive system [9], etc.

Gesture recognition as a pattern recognition, supervised machine learning methods are widely researched, such as Support Vector Machine(SVM) [17] [2], K-nearest Neighbor(KNN) [15], Linear Discriminant Analysis(LDA) [2], Hidden Markov Model(HMM) [7], Random Forest(RF) [20], etc. These conventional methods require extracting features before classifying. Thus, a plenty of handcraft features were researched.

In recent years, Deep learning has been demonstrated remarkable capabilities in many fields, such as image classification, natural language processing(NLP) and object detection, etc. Similarly, deep learning based methods also give gesture recognition a new perspective. Unlike traditional machine learning method which requires a mount of human-selected, handcraft features. Deep learning based method can learn to extract appropriate feature by itself. Specifically, an increasing number of Convolutional Neural Networks(CNN) or Recurrent Neural Networks(RNN) based methods were researched to improve gesture recognition. More recently, 1D CNN have been considered the more appropriate approach to address mutli-variate time sequence such as sEMG and EEG.

In this work we proposed a TCN [5] based networks named Bi-TCN. Unlike CNN and CNN-RNN [22] [26] based method, TCN consider sEMG based gesture recognition as a multivariate time sequence classification and use 1D casual convolution. Motivated by Bi-LSTM, proposed networks also requires different order of sEMG signal. In the recognition period, the positive order of sEMG signal and reverse order of the same sEMG signal are feed to our networks. We validation our method on two benchmark datasets, Ninapro DB1 [3] and DB5 [18]. The results demonstrate our method is comparable to state-of-the-art works.

The organization of this paper is given as follow: Section 2 provides an overview of the related gesture recognition approaches. In Section 3, we provide the detail about proposed Bi-TCN architecture. The description of datasets is given in Section 4. In section 5, we describe the detail of the experiment. The results and discussion are presented in Section 6. Finally, we draw conclusions for this paper and give future work in Section 7.

2 Related Work

Same as other classification task, sEMG based Gesture recognition can be divided into two categories. The first is traditional machine learning based method and the second is deep learning based method.

For conventional machine learning method, feature extracting is required before classifying. There are three categories of these features: Time domain, Frequency domain and Time-Frequency domain. Atzori [3] extract Root mean square(RMS), Histogram (HIST), marginal Discrete Wavelet Transform(mDWT) as features. By combining these features with different conventional classifiers, the results shows that random forest with combined features perform the best. Ali H. Al-Timemy [1] use time domain-auto regression(TD-AR) feature and SVM to classify sEMG. They achieve 98% accuracy over ten intact subjects for the classification of 15 classes finger movement and 90% accuracy over six amputee subjects with 12 classes finger movement.

Recently, deep learning based method was widely studied. For most Deep Learning(DL) based methods, handcraft features is no more required. Hongfeng

Chen [6] *et al*. proposed a new feature extracted by CNN, called CNNfeat. By comparing CNNfeat with 25 traditional features, the result demonstrated that CNNfeat outperform the traditional features. Wentao Wei [25] *et al*. use "divide-and-conquer" strategy to split sEMG image into equal-sized stream. Then, they use CNN learns representation of each stream and feed these learned feature to a fusion classification networks. In order to learn to extract spatial-temporal information of SEMG, Yu Hu *et al*. [13] combine CNN and RNN as a new classification model. The average accuracy achieved on Ninapro DB1 and DB2 are 87.2% and 82.2%, respectively. Elahe Rahimian *et al*. [19] proposed 1D CNN based architecture named XceptionTime, they achieve SOTA performance with 92.3% prediction accuracy on Ninapro DB1.

On the other hand, some researchers combine traditional features with deep learning method to learn multi-view classification. Wentao Wei*et al*. [24] combine classical sEMG features to train a multi-view classifer. They select 3 classical handcraft features and use these features to train their multi-view networks. Finally, they achieve 88.2%, 83.7%, 64.3%, 90%, 64%, 88.3% prediction accuracy on Ninapro DB 1, 2, 3, 5, 6 , 7 without IMU, respectively.

3 Proposed Method

In many existing works [25] [24], sEMG based gesture recognition was regarded as image classification task. These methods may be not appropriate for the loss of temporal information hidden in the sEMG signal. To consider the temporal information in sEMG signal, some researchers combine LSTM with CNN, like [22] [26]. By this way, the networks can learn the spatial-temporal information of the sEMG signal. But this method will lead to parameter explosion and pressure on real-time calculation. To address above problem, a new networks architecture are presented name Temporal convolutional networks(TCN). TCN [5] has been proved to be a more appropriate method to address multivariate time series classification task by using a new technology called casual convolution and 1D CNN.

Temporal Convolutional Networks based on two principles: The first is the future information can not be leaked to the past, and the second is that neural network produces an output of the same length as the input. To satisfy these two principles, casual convolution is introduced. More detail, casual convolution can be implemented simply by padding the time sequence and discarding the over-padding value at the end after convolution operation. Padding value can be calculated as follow:

$$padding = (k - 1) * d \tag{1}$$

where k is kernel size, d is dilation factor

However, the long sequence challenges the simple casual convolution. We need stack casual convolution for many layers to look back history for very long time sequences. however, this operation will leads to a large increase in parameters and over-fitting. To solve this problem, the dilated convolution is introduced which can be calculated as follow:

$$f(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d_i} \tag{2}$$

Where k is kernel size,d is dilation factor,$s - d_i$ is the direction of the past.

To summarized, the temporal convolution is equal to 1D casual CNN + 1D dilation CNN. By using these strategies, TCN can extract information of multi-variate time sequences with a few layer, meanwhile, it will not destroying temporal attribute. The basic Temporal Convolution Networks is showed in fig1.

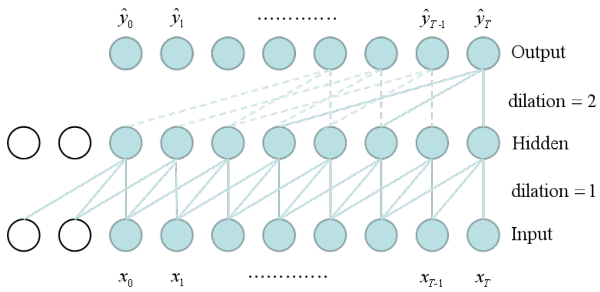


Fig. 1. Temporal Convolutional Networks.

In this work, motivated by Temporal Convolutional Networks(TCN) and Bidirectional Long Short-Term Memory(Bi-LSTM), we propose Bidirectional Temporal Convolutional Networks(Bi-TCN) which contains a pair parallel TCN as showed in fig2. The structure of two TCN is the same but the input is different. Like Bi-LSTM, one of the TCN's input is normal sEMG signal, and the other's input is reverse order of the same sEMG signal.

The first layer of proposed TCN is Batchnorm. then, followed by four casual convolution layers, the dilation rate of these convolution is set to 1, 1, 2, 2 respectively and the kernel size of all convolutional layers is 3. After each casual convolution layer, we apply PReLU activation function, Bachnorm and Dropout. In this work, we set the possibility of Dropout layer to 0.5. In the first two casual convolution layers, we apply SE-Module [12]. Basic structure of SE-Module is showed in fig3. This module can make model pay more attention on key channels. In the last two convolutional layers, we apply residual connect [11], which solves the degradation problem of deep neural networks very well by allowing layers to learn modifications to the identity mapping. The kernel size of residual layers is set to 1 with padding zero. After four casual convolutional layers, we summed the feature according to time dimension. the final layer of proposed TCN is one layer MLP which predicts the classification result. All parameters of model are initialized by using Xavier method [10].

Our method requires two stages to train. In the first stage, we train two TCN separately. Firstly, sEMG signal are feed to one TCN. Then, the extracting

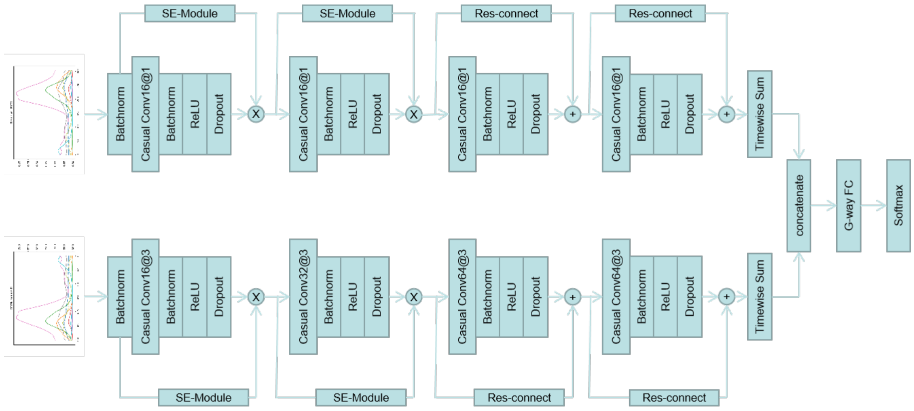


Fig. 2. Bidirectional Temporal Convolutional Networks.

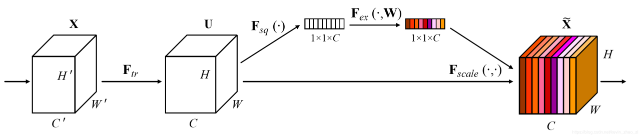


Fig. 3. SE-Module.

feature are summed according to time dimension. Finally, the summed feature are feed to on layer fully connected to train and classification. Similarly, we train the other TCN with the reverse order sEMG signal. After the first stage, we discard the fully connected layer of both two TCN and freeze its parameters.

In the second stage of train, the positive and reverse order sEMG signal are simultaneously feed to mentioned TCN. Then, the extracting feature of two TCN are summed according to time dimension and concatenated. Finally, the summed feature are feed to final fully connected layer to train. In this way, our proposed networks combine two views of the sEMG signal.

4 Datasets

To validate our proposed model, a benchmark database Ninapro is used in this paper. The Ninapro database have 7 datasets, we select the first and the fifth dataset as our validation datasets .

4.1 Ninapro DB1

The first Database of Ninapro is recorded by Otto Bock MyoBock 13E200 with 10 wireless electrodes (channels) at a sampling rate of 100Hz. DB1(Ninapro database1) [3] comprise 52 gestures from 27 intact subjects, each movement include 10 repetition, and each repetition lasted 5s. After recording, a generalized likelihood ratio algorithm is applied offline to correct the stimulation. As previous works [3] [25] [24], we apply 1st 1Hz butter-worth low-pass to filter this signal, and use repetition 1, 3, 4, 6, 8, 9, 10 as train set, and remain as test set.

4.2 Ninapro DB5

DB5 [18] include same gesture as DB1 from 10 intact subjects, each movement include 6 repetitions. The acquired equipment of DB5 is double MYO band. One of the MYO band is placed close to elbow, and the other is placed close to hand. The MYO band have 8 electrodes and the sample ratio of MYO band is 200Hz. We also apply 1st 1Hz butter-worth low-pass to filter it. We use the repetition of 1, 3, 4, 6 for training, and remain for testing.

For above datasets, we extract features by applying sliding window with 200ms window size, because the time of classification for real-time recognition should below 300ms [14]. Besides, The window step of DB1 is 10ms and of DB5 is 100ms. Like many previous works [4] [19] [24], we discard the rest gesture. Additionally, the gesture in above two datasets can be divide into three categories. Exercise A: 12 basic movements; Exercise B: 8 isometric and isotonic hand configurations and 9 basic movements of the wrist; Exercise C: 23 grasping functional movements. Fig4 show a part of these gestures. The other information of above datasets is also showed in table1.

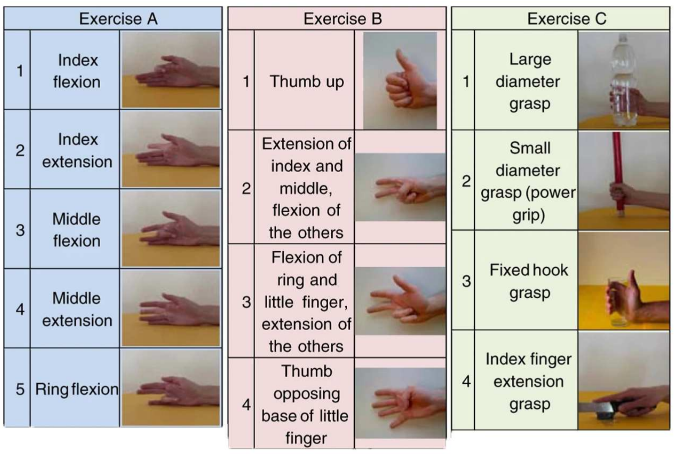


Fig. 4. Gestures.

Table 1. Specification of used Databases.

Database	DB1	DB5
Number of gestures	52	52
Number of trials	10	6
Intact subjects	27	10
Number of channel	10	16
Sampling ratio	100Hz	200Hz
Repetition for training	1,3, 4, 6, 8, 9, 10	1, 3, 4,6
Repetition for testing	2, 5, 7	2, 5

5 Experiment Setup

Because of different sampling ratio of two datasets, there are different setting during training model. For the DB1, the filter number of convolutional layers is set to 16, 32, 64, 64, respectively. The final FC layer of Bi-TCN is comprised by one MLP layer. Batchnorm is applied before final MLP. For the DB5, the filter number of convolutional layers is set to 32, 64, 128, 128, respectively. The final FC layer of Bi-TCN is comprised by two MLP layer. Output size of first MLP layer is 128. Similarly, batchnorm is applied before each MLP layer. The reason is that the length of DB5 data is twice than DB1's which require larger capacity of networks. We use Adam as our optimizer and ReducLRPlateau as learning scheduler during both training stages. The parameters of ReduceLRPlateau is set as: $model =' min', factor = 0.5, patient = 5, eps = 1e-8$. In all stages, the training epoch for DB1 is 50, and for DB5 is 80. In the first stage, the learning rate is set to 0.01. In the second stage, the initial learning rate is set to 0.001. We use CrossEntropyLoss as our loss function, which can be represented as follow:

$$H(p,q) = -\sum \sum_{i=1}^{k} p(x_i) log(q(x_i)) \tag{3}$$

We also used label-smoothing [16] for slowing down over-fitting. The label-smoothing can be calculated as follow:

$$q_i = \begin{cases} 1-\varepsilon & if\ i = y \\ \varepsilon/(K-1) & otherwise \end{cases} \tag{4}$$

Where ε is a hyper parameters, K is number of gestures. We set ε equal to 0.1.

RBF-SVM, LDA and Random Forest are selected as conventional classifiers. Traditional machine learning requires extracting feature before training and testing. Root mean square(RMS) is selected as our validation feature.

The gesture recognition accuracy is calculated as given below:

$$Accuracy = \frac{Correct\ classification}{Total\ test\ sample} * 100\% \tag{5}$$

We run all experiments for 5 times and report the average result. All experiments were implemented in workstation with INTEL 9700k processor, 16GB RAM and 2070super, 8GB GPU.

6 Result And Discussion

The classification results can be seen in 5, SVM with RBF kernel yields the best performance among the traditional classifiers, with 81.05% prediction accuracy for DB1 and 71.22% prediction accuracy for DB5, but it's still inferior to the performances of deep learning method. Our proposed TCN yield 90.16%±0.33% recognition accuracy on DB1 and 88.00%±0.34% on DB5. Comparing with proposed TCN, Bi-TCN improves average accuracy from 90.16% to 90.74%±0.26% on DB1, with a increasing of 0.58%. On DB5, it improves average accuracy from 88.00% to 90.06%±0.15%, with a increasing of 2.06%.

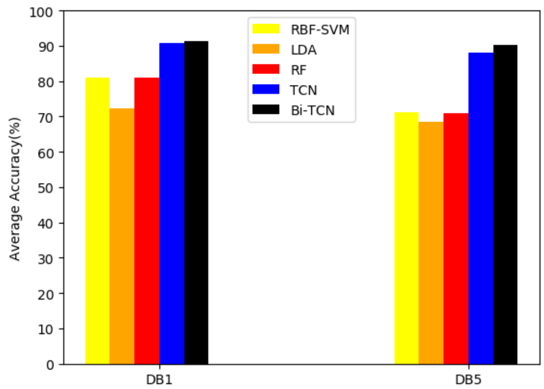
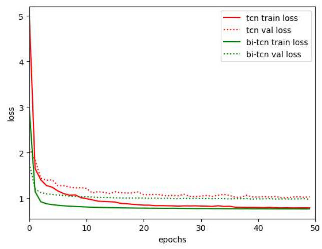
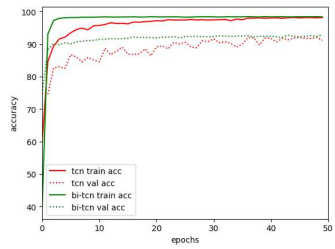


Fig. 5. Performances comparsion of different classifier.

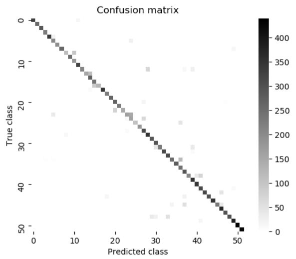
The train loss and test loss of Bi-TCN can be seen in fig6 and fig7. From the results, we can clearly see the loss of Bi-TCN are lower than proposed TCN. In the confusion matrix, some incorrect classification of proposed TCN are correctly classified by Bi-TCN. This phenomenon demonstrate that through combine positive order and reverse order of the same sEMG signal, our Bi-TCN can extract more information hidden in the signal than proposed TCN. On the other hand, as showed in the confusion matrix, the misclassification occurs more often between adjacent classes. This is because the gesture of adjacent classes are more similar and the sEMG signal of these gesture are also similar, which makes classification more difficult. That's phenomenon indicates we should pay more attention to the classification of similar gesture to further improve the accuracy of gesture recognition.



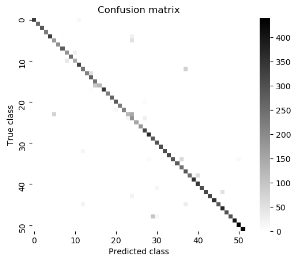
(a) Loss.



(b) Accuracy.

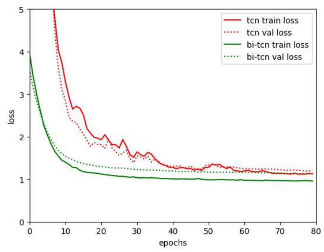


(c) TCN confusion matrix.

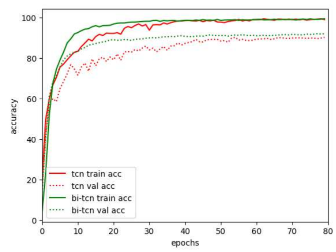


(d) Bi-TCN confusion matrix.

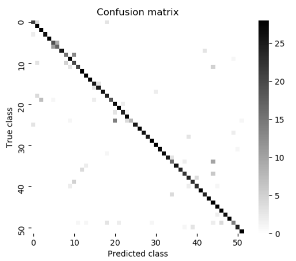
Fig. 6. Results of DB1 subject 1.



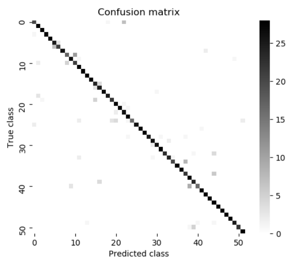
(a) Loss.



(b) Accuracy.



(c) TCN confusion matrix.



(d) Bi-TCN confusion matrix.

Fig. 7. Results of DB5 subject 1.

We also compare our method with related works. The result can bee seen in the table2, our method achieve best result on DB5. For DB1 our method's prediction accuracy is lower than SOTA, but it's worth noting that our TCN only have 32k parameters and Bi-TCN have 72k parameters, while SOTA method have 410k parameters. The less parameters means less computation power needed, which will improve real-time performance. In [23] which also use TCN framework, they achieve 89.76% recognition accuracy on 53 movements but with 300ms window. It's worth know that with decreasing of window size, the accuracy will also decrease, thus, our proposed method is outperformance than general TCN methods. On DB5, our method achieve 90.06% recognition accuracy on 52 movement, while SOTA get 90% prediction accuracy on 41 movements.

Table 2. Gesture recognition accuracy of the proposed TCN and Bi-TCN compared with the accuracy of existing works.

Database	Model	Accuracy	Parameters
DB1	Multi-view [24]	88.2%	-
	TCN [23]	89.76%(53 movements, 300ms)	85k
	XceptionTime [19]	**92.3**%	413k
	Proposed TCN	**90.16%±0.33%**	**32k**
	Proposed Bi-TCN	**90.74%±0.26%**	**72k**
DB5	TL CNN [8]	68.98%(18 movements)	-
	LCNN [26]	71.66%	-
	Stacking Ensemble Learning [21]	72.09%(40 movements)	-
	Multi-view [24]	90%(41 movements)	-
	Proposed TCN	**88.00%±0.34%**	**121k**
	Proposed Bi-TCN	**90.06%±0.15%**	**257k**

7 Conclusions

In this paper, we proposed Bidirectional Temporal Convolutional Network(Bi-TCN) to improve the performance of sEMG based gesture recognition. Our proposed networks is consist of two same structure of Temporal Convolutional Networks(TCN). This new TCN method requires two training stages. In the first stage, two TCN are trained by using reverse order sEMG signal and positive order sEMG signal, respectively. Then, we combine the feature extracted by two TCN to train a final classifier. By validating our network on two benchmark dataset, the results demonstrate our method can achieve 90.74%±0.26% and 90.06%±0.15% recognition accuracy on DB1 and DB5 respectively which is comparable to the SATO method. Moreover, our networks have less parameters than previous' which will improve the ability of real-time gesture recognition and reduce the requirements on equipment.

Despite of the good performance of Bi-TCN,it is difficult to ensure that the features learned by TCN and Reverse-TCN are different. So our future work will focus on find appropriate approaches to avoid duplication of learned features. Furthermore, we also will consider utilizing Graph Neural Networks(GNN). The reason is that the muscle groups of the hand are different when execute different gestures and this relation of the muscle groups can be regarded as a relation graph.

References

1. Ali H Al-Timemy, Guido Bugmann, Javier Escudero, and Nicholas Outram. Classification of finger movements for the dexterous hand prosthesis control with surface electromyography. *IEEE journal of biomedical and health informatics*, 17(3):608–618, 2013.
2. Ahmet Alkan and Mücahid Günay. Identification of emg signals using discriminant analysis and svm classifier. *Expert systems with Applications*, 39(1):44–47, 2012.
3. Manfredo Atzori, Arjan Gijsberts, Claudio Castellini, Barbara Caputo, Anne-Gabrielle Mittaz Hager, Simone Elsig, Giorgio Giatsidis, Franco Bassetto, and Henning Müller. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific data*, 1(1):1–13, 2014.
4. Manfredo Atzori, Arjan Gijsberts, Simone Heynen, Anne-Gabrielle Mittaz Hager, Olivier Deriaz, Patrick Van Der Smagt, Claudio Castellini, Barbara Caputo, and Henning Müller. Building the ninapro database: A resource for the biorobotics community. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 1258–1265. IEEE, 2012.
5. Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
6. Hongfeng Chen, Yue Zhang, Gongfa Li, Yinfeng Fang, and Honghai Liu. Surface electromyography feature extraction via convolutional neural network. *International Journal of Machine Learning and Cybernetics*, 11(1):185–196, 2020.
7. Joyce Chiang, Z Jane Wang, and Martin J McKeown. A hidden markov, multivariate autoregressive (hmm-mar) network framework for analysis of surface emg (semg) data. *IEEE Transactions on Signal Processing*, 56(8):4069–4081, 2008.
8. Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771, 2019.
9. Cheikh Latyr Fall, Gabriel Gagnon-Turcotte, Jean-François Dubé, Jean Simon Gagné, Yanick Delisle, Alexandre Campeau-Lecours, Clément Gosselin, and Benoit Gosselin. Wireless semg-based body–machine interface for assistive technology devices. *IEEE journal of biomedical and health informatics*, 21(4):967–977, 2016.
10. Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research*, 9:249–256, 2010.
11. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

12. Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
13. Yu Hu, Yongkang Wong, Wentao Wei, Yu Du, Mohan Kankanhalli, and Weidong Geng. A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition. *PloS one*, 13(10), 2018.
14. Bernard Hudgins, Philip Parker, and Robert N Scott. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 40(1):82–94, 1993.
15. Ernest Nlandu Kamavuako, Jakob Celander Rosenvang, Ronnie Horup, Winnie Jensen, Dario Farina, and Kevin B Englehart. Surface versus untargeted intramuscular emg based classification of simultaneous and dynamically changing movements. *IEEE Transactions on neural systems and rehabilitation engineering*, 21(6):992–998, 2013.
16. Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4696–4705, 2019.
17. Mohammadreza Asghari Oskoei and Huosheng Hu. Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE transactions on biomedical engineering*, 55(8):1956–1965, 2008.
18. Stefano Pizzolato, Luca Tagliapietra, Matteo Cognolato, Monica Reggiani, Henning Müller, and Manfredo Atzori. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PloS one*, 12(10), 2017.
19. Elahe Rahimian, Soheil Zabihi, Seyed Farokh Atashzar, Amir Asif, and Arash Mohammadi. Xceptiontime: Independent time-window xceptiontime architecture for hand gesture classification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1304–1308. IEEE, 2020.
20. Erik Scheme and Kevin Englehart. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of Rehabilitation Research & Development*, 48(6), 2011.
21. Shu Shen, Kang Gu, Xin-Rong Chen, Ming Yang, and Ru-Chuan Wang. Movements classification of multi-channel semg based on cnn and stacking ensemble learning. *IEEE Access*, 7:137489–137500, 2019.
22. Runze Tong, Yue Zhang, Hongfeng Chen, and Honghai Liu. Learn the temporal-spatial feature of semg via dual-flow network. *International Journal of Humanoid Robotics*, 16(04):1941004, 2019.
23. Panagiotis Tsinganos, Bruno Cornelis, Jan Cornelis, Bart Jansen, and Athanassios Skodras. Improved gesture recognition based on semg signals and tcn. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1169–1173. IEEE, 2019.
24. Wentao Wei, Qingfeng Dai, Yongkang Wong, Yu Hu, Mohan Kankanhalli, and Weidong Geng. Surface-electromyography-based gesture recognition by multi-view deep learning. *IEEE Transactions on Biomedical Engineering*, 66(10):2964–2973, 2019.
25. Wentao Wei, Yongkang Wong, Yu Du, Yu Hu, Mohan Kankanhalli, and Weidong Geng. A multi-stream convolutional neural network for semg-based gesture recognition in muscle-computer interface. *Pattern Recognition Letters*, 119:131–138, 2019.

Bidirectional Temporal Convolutional Networks 13

26. Yuheng Wu, Bin Zheng, and Yongting Zhao. Dynamic gesture recognition based on lstm-cnn. In *2018 Chinese Automation Congress (CAC)*, pages 2446–2450. IEEE, 2018.
27. Ji Won Yoo, Dong Ryul Lee, Yon Ju Sim, Joshua H You, and Cheol J Kim. Effects of innovative virtual reality game and emg biofeedback on neuromotor control in cerebral palsy. *Bio-medical materials and engineering*, 24(6):3613–3618, 2014.