EMG-based HCI Using CNN-LSTM Neural Network for Dynamic Hand Gestures Recognition

Qiyu Li * **Reza Langari** **

* *Texas A&M University, College Station, TX 77843 USA (e-mail: qiyu.li@tamu.edu).*
** *Texas A&M University, College Station, TX 77843 USA (e-mail: rlangari@tamu.edu)*

Abstract: Human-computer interaction(HCI) has a broad range of applications. Many HCI systems are based on bio-signal analysis and classification. The surface electromyographic(sEMG) signal is one of the most used signals that are formed by muscle activation although the details are rather complex. The applications of sEMG signals are referred to as myoelectric control since the dominant use of this signal is to activate a device even if (as the term control may imply) feedback is not always used in the process. With the development of deep neural networks, various deep learning architectures are used for sEMG-based gesture recognition with many researchers having reported good performance. Nevertheless, challenges remain in accurately recognizing sEMG patterns generated by gestures produced by the hand or the upper arm. For instance, one of the difficulties in hand gesture recognition is the influence of limb positions. Several papers have shown that the accuracy of gesture classification decreases when the limb position changes even if the gesture remains the same. Prior work by our team has shown that dynamic gesture recognition is in principle more reliable in detecting human intent, which is often the underlying idea of gesture recognition. In this paper, a Convolutional Neural Network (CNN) with Long Short-Term Memory or LSTM (CNN-LSTM) is proposed to classify five common dynamic gestures. Each dynamic gesture would be performed in five different limb positions as well. The trained neural network model with high recognition performances is then used to enable a human subject to control a robotic arm.

Keywords: Myoeletric Control, Neural Network, Human Computer Interaction, sEMG Signal, Gesture Recognition

1. INTRODUCTION

Human-computer interaction (HCI) is an important technology in conjunction with intelligent mechatronic devices(Ozdemir et al. (2020)). A number of biosignals are used in HCI applications as the linkage between computers and humans. Surface electromyography (sEMG) signal has found its niche in this realm as it is evident in the literature (Sun et al. (2020), Yasen M (2019)). sEMG is generated by various ions that are present in the muscles during flexion and contraction movements(Shanmuganathan (2020)). With the development of deep learning methods, various of different deep neural networks are applied to sEMG signal analysis. For example, An attention-based Bidirectional Convolutional Gated Recurrent Unit (Bi-CGRU) model(Xie et al. (2020)) has demonstrated a 88.73% accuracy on hand gestures. However, while good performance using various models has been achieved, we still face challenges in sEMG-based gesture recognition. The research conducted by (Mukhopadhyay and Samui (2020)) shows that limb position has a significant impact on gesture recognition.

According to (Mitra and Acharya (2007)), gestures can be divided into static and dynamic categories. A static motion is represented by repeated and constant multi-dimensional sEMG features. In contrast, a dynamic motion is expressed by a temporal sequence of multiple sEMG features that vary during the respective motion. A previous member of our team, (Shin (2016)), proposed a sequence-based pattern recognition model to classify dynamic sEMG hand gestures. Based on his research, the classification of dynamic gestures achieved higher accuracy compared to the result of static gestures when limb position was changed. He also explored the potential for dynamic gestures classification at different limb positions and showed high recognition accuracy at different limb positions by using cross-validation but did not demonstrate a real-time recognition system for dynamic gestures at different limb positions. The present work uses some dynamic gestures but at more common limb positions than those used in Shin's work. A real-time system then recognizes all the gestures at all different limb positions. Because the limb positions used are common in people's daily lives, the proposed approach enhances the potential for gesture recognition to handle real-world problems.

In this paper, five dynamic gestures are chosen. A Convolutional Neural Network (CNN) with Long Short-Term Memory, or LSTM (CNN-LSTM) is used to classify the

Manuscript 162 submitted to 2022 Modeling, Estimation and Control Conference.
Received April 23, 2022.

resulting dynamic gestures so as to train an HCI system involving interaction with a robotic arm designed based on a trained CNN-LSTM model in Section 2. Then, two experiments involving the model itself and the HCI system are presented in Section 3. Section 4 shows and analyses the results of experiments from section 3, then gives the conclusion.

2. METHOD

In this section, a CNN-LSTM neural network is introduced for dynamic gestures recognition. Then based on this neural network, we will describe an HCI system as the application of myoelectric control.

2.1 CNN-LSTM Neural Network

Convolutional Neural Network The convolutional neural network known as CNN is commonly used in image processing (Naranjo-Torres et al. (2020), Huang et al. (2020)) due to its strong feature extraction function. In the convolutional layer, a kernel containing a batch of parameters connects the current layer and input data of the previous layer. The kernel has a smaller size than the input matrix. Every time, the kernel would move along a specific direction in a predefined step on the input data until it moved through the whole input matrix. In the training process, the parameters in the kernel would update constantly(Huang and Chen (2019)). Therefore, the special local information from the input data would be captured. There are three kinds of convolutional layers1D, 2D, 3D (where D means dimension). 1D convolutional layer shown in Figure. 1, is commonly used in time series data since the kernel only moves along the time axis which preserves the important temporal information. For a complete CNN architecture shown in Figure. 2, there are batch normalization layer and drop out layer following the convolutional layer.

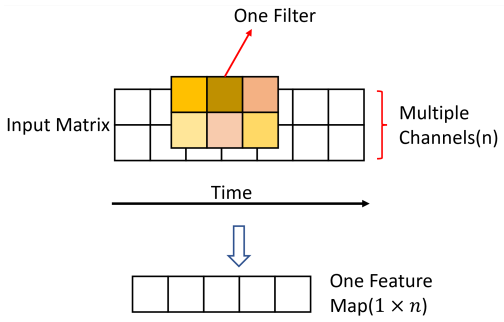

Fig. 1. 1D Convolution with Multiple Channels

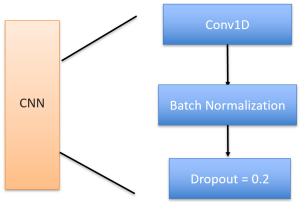

Fig. 2. CNN Architecture

Long Short-Term Memory Neural Network Long Short-Term Memory neural network known as an LSTM neural network is a type of Recurrent Neural Network (RNN). Unlike RNNs that face *vanishing* problem when the input data is long. The LSTM architecture(Hochreiter and Schmidhuber (1997)) was proposed to handle "long-term dependencies"(Yu et al. (2019)). An LSTM unit includes three gates that enable the network to handle long-term dependencies as shown in Figure. 3.

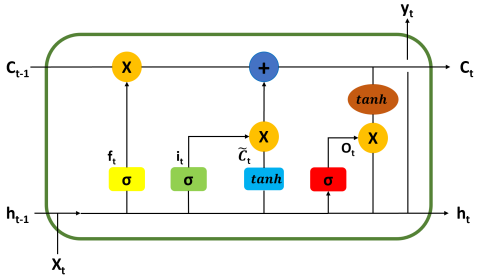

Fig. 3. LSTM Unit

The value of the gate controls the flow of information.The range of the value is between 0 to 1. When the gate approaches 0, the information through that gate would be deleted, and when the gate is close to 1, the information through the gate would be saved. For example, if the information is relevant, 115 the forget gate would be close to 0, and the input gate would be close to 1. Therefore, the new information would be stored in the current cell while the old information is forgotten. Consequently, the entire network can easily learn long-term dependencies between sequences(Wang et al. (2020)).

CNN-LSTM Neural Network Based on the previous introduction, the CNN LSTM in Figure 4 is used in the present study. This type of hybrid neural network has already been employed in many types of research(Cai and Zhu (2021), Islam et al. (2020)). The hybrid CNN-LSTM neural network is employed for sEMG-based dynamic gesture recognition where the gestures are performed in different limb positions in this paper. The input is the raw EMG sequences of multiple channels. The CNN would extract local features of the raw data, which is then input into the LSTM neural network where the temporal information is used. Finally, a dense layer with neurons which is the same as the number of gestures categories, and a softmax layer are followed. Note that the notion of a dense layer refers to a type of layer in neural network. This does not relate to the data scale. The neurons of one dense layer are connected with every neurons of its previous layer.

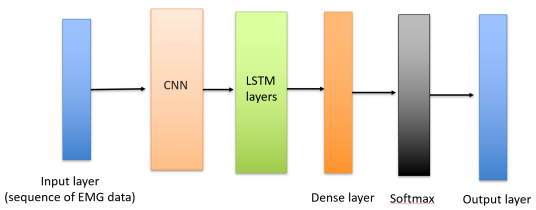

Fig. 4. CNN-LSTM Nueral Network

Manuscript 162 submitted to 2022 Modeling, Estimation and Control Conference.
Received April 23, 2022.

2.2 HCI system

The CNN-LSTM model is determined in the classification process, the real-time recognition system for five gestures is designed as follows. a MYO armband [1] shown in Figure. 5 is used for raw EMG data collection. It has eight EMG sensors with a 135 sampling rate of 200hz. In this research, we only consider the influence of arm positions. Therefore, only right handed human subjects were recruited and each human subject wears the armband on their right forearm.

Fig. 5. MYO Armband

In the real-time recognition system, a fist recognition is set as a trigger before collecting the signals of dynamic gestures. This motion could be considered as a trigger mechanism that informs the programming to capture the gesture signals.For the trigger mechanism, a specific value as a threshold is set to identify if the sum of the absolute values of each channel is higher than the threshold, which indicates muscle activity.After the fist motion, the armband vibrates. Then gesture signals would be recorded in the following 3 seconds. Compared to no trigger mechanism, the system with the fist offers the programming a clear start to capture pieces of signals and less noise. Based on the real-time recognition system, a HCI system(as shown in Figure.6) allows people to use dynamic gestures to control a robot arm.

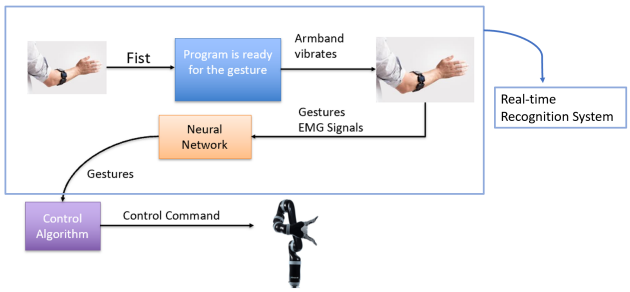

Fig. 6. HCI System

To test our HCI system, a task is designed. Once the task is fixed, we would have specific trajectories for completing the task by using the HCI system. As an example, a task with four trajectories is designed. Each trajectory corresponds to a set of control commands: control commands 1(C1), control commands 160 2(C2), control commands 3(C3) and control commands 4(C4). If duplicate commands are sent to the robot, the task will fail. For instance, if C3 is sent twice, the robot would never complete the task even if it finally receives all four commands. Since the classification accuracy is not 100%, there will be wrong recognition results which lead to wrong commands. To avoid the issue, a control algorithm for the task is shown in Figure. 7. Each gesture corresponds to a control command. The mechanism is to produce a certain order for the commands. If there is a wrong classification because the output which represents the command and i do not match, no command would be sent. Human subjects need to repeat the gestures in the order list until the robot moves, then perform 170 the next motions. A state machine explains how the HCI system runs and the state changes of the robot arm as shown in Figure. 8. 0 represents a wrong command and 1 represents the correct command; q0 is an initial state for the task that the robot arm is in a home position. The state would change every time the correct commands are generated. On the contrary, the state would remain in its current status and does not change if a wrong command is generated. When the first correct command is generated, the robot arm will move to the first position, and the system will be at the q1 state. Then the system would be at the q2 state after the second correct command is generated. Next, the system would be at the q3 state after generating the third correct command. The last state would be q4 after the fourth correct command is generated. The control algorithm could be extended when more trajectory pieces are needed for a task.

```
i = 1
if fist:
{
if motion == gestures 1 and i == 1:
   control command 1
   i = i + 1
if motion == gestures 2 and i == 2:
   control command 2
   i = i + 1
if motion == gestures 3 and i == 3:
   control command 3
   i = i + 1
if motion == gestures 1 and i == 4:
   control command 4
   i = i + 1
 if i == 5:
    print("task finished")
}
```

Fig. 7. Control Algorithm

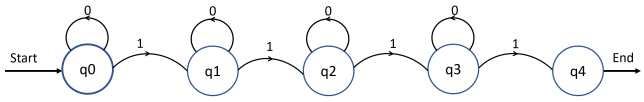

Fig. 8. Finite State Machine for The Task

[1] The manufacturer of MYO armband is Thalmic Labs

Manuscript 162 submitted to 2022 Modeling, Estimation and Control Conference.
Received April 23, 2022.

3. EXPERIMENTS SET

Two experiments are conducted in this research. The first one is to measure the performance of the gesture recognition system and the second one is to test the HCI system.

3.1 First Experiment: Gesture Recognition

The first experiment was designed to test the performance of the CNN-LSTM model to classify the five dynamic gestures shown in Figure. 9. For each gesture, the human subject was asked to perform these at five different limb positions.

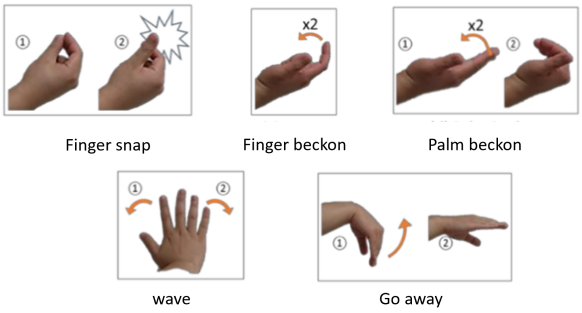

Fig. 9. Five Dynamic Gestures. Adapted from "Myoelectric Human Computer Interaction Using Reliable Temporal Sequencebased Myoelectric Classification for Dynamic Hand Gestures" by Shin, 2016, Doctoral dissertation, Texas A & M University.

The training dataset is from 7 human subjects. Every gesture at each arm position was repeated 10 times. And to increase the data size, augmentation is used here that shuffles the order of 8 signal channels shown in Figure. 10. The final dataset is six times of the original one. Then the new dataset is used to train the CNN-LSTM neural network.

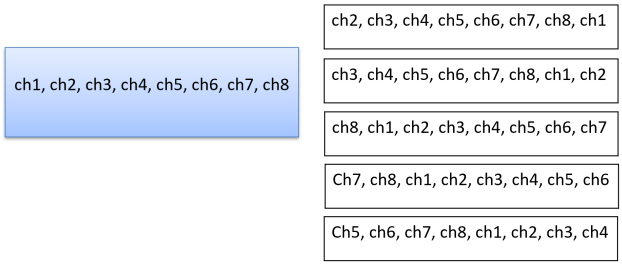

Fig. 10. Augmentation

After the CNN-LSTM model was trained by the above training dataset, the model was applied to the real-time gesture recognition system mentioned before to test its accuracy. The test dataset was from 12 human subjects. We chose 7 people for the training dataset since it would not be too insufficient for the training where the validation accuracy could reach over 90%, and small tests on subjects perform well. We used 12 people for the test. Human subjects for the test have two parts, one part includes people who contributed to the training dataset before, one part includes people who are not in the training dataset. So this may give us a view to see if its contribution to the training dataset would influence the accuracy of a subject test. Every gesture at each arm position would be repeated 10 times. The recognition results would be recorded.

3.2 Second Experiment: HCI System

The second experiment is to evaluate the performance of the HCI system. To measure the performance, we design a go and grasp task. A water bottle is put at a certain point where it is reachable by the robot arm. The robot arm starts from its home position, approaches the bottle, and closes its gripper to grab it. Then the robot moves back to its home position with the bottle. The experiment is set up as shown in Figure 11.

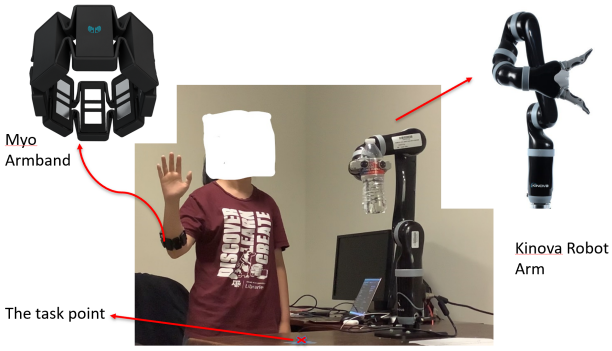

Fig. 11. Task Setup

Human subjects repeat the task three times and the time of every task is recorded. In addition, a joystick is used as a comparison device since the user can send a direct command to the robot arm compared to the HCI system which required time to recognize a gesture and perform the required command. Human subjects need to finish the same go and grasp task via the joystick three times and their time is recorded. Human subjects are given ten minutes to get familiar with the joystick before the task. If the water bottle were to be knocked down by the manipulator during one task, the task execution would continue and the time to reset the bottle would be included in the task execution time

4. RESULTS AND DISCUSSION

4.1 Results for First Experiment

This section presents the results obtained by applying the first experiment on 12 human subjects. There are five gestures and each gesture had five arm positions. The human subjects repeated each gesture at each arm position ten times. The total amount of test gestures is 3000. The overall accuracy was 84.2%, which shows the comprehensive condition for these 3000 gesture data. To show the influence of limb positions on each gesture, the accuracy for each position is in the Table 1.

Manuscript 162 submitted to 2022 Modeling, Estimation and Control Conference.
Received April 23, 2022.

Table 1. Accuracy in Each Limb Position

Position 1	Position 2	Position 3	Position 4	Position 5
82%	82.14%	81.68%	84.5%	85.66%

From Table 1, the accuracies vary in a small range among different arm positions. To get more details of the accuracies, the distribution of accuracy at different positions for each different gestures is given in Figure 12.

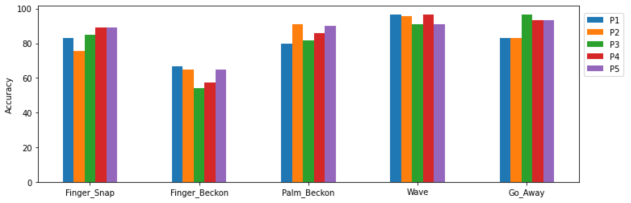

Fig. 12. Accuracy Distribution

To further explore the accuracies of the model, we analyzed the same gesture at the same arm positions performed by different subjects. In Figure. 13, we represent the performance of the finger snap gesture from two subjects. The accuracies of the finger snap for the two subjects are slightly different. As is shown in Figure. 13, the accuracies of S1 on P1, P2 and P3 are higher than these of S2. In addition, the accuracies of S1 on P2 and P5 are equal to these of S2. Therefore, the overall accuracy of S1 is 98% which is higher than 84% of S2.

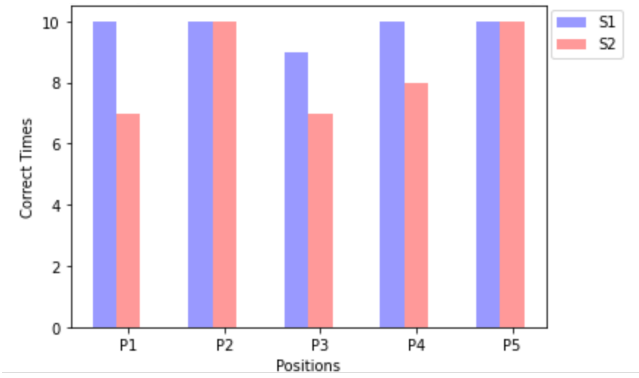

Fig. 13. Finger Snap at Five Different Limb Positions for Two Different Subjects

4.2 Results for Second Experiment

In this section, we represent, analyze and compare the performance of myoelectric control and joystick control mode. Each human subject repeated three times on each control mode. The average time of task of 12 human subjects for two control modes is in the Table 2.

Table 2. Average Time for Joystick Control and Myoelectric Control

Subject	Myoelectric Control(s)	Joystick Control(s)
1	41.6	37.0
2	36.2	35.6
3	36.6	52.7
4	44.1	26.6
5	57.3	30.06
6	43.2	43.4
7	48.6	43.0
8	40.6	23.0
9	50.6	31.62
10	36.87	41.9
11	28.4	55.8
12	36.4	37.91

4.3 Discussion

For the first experiment in previous sections, Table 1 gives a brief idea of how much limb position impacts the gesture classification accuracy. The average accuracy of the 12 human subjects at each limb position is close; the largest difference is 3.98%, which means the model could classify five dynamic gestures effectively when the limb positions change. Figure. 12 provides the details for accuracies of every gesture at different limb positions. As we can see, there is no major difference between accuracies at five different limb positions for each gesture. This further proves that the model is relatively position-invariant for the five dynamic gestures. But the Figure. 12 also indicates that the dynamic gesture itself could influence the accuracy where the accuracy of the wave gesture is over 90% and the accuracy of finger beckon is around 70%. Additionally, Figure. 13 shows that different human subjects have an impact on accuracy. In short, the model is relatively position-independent, but the accuracy is slightly influenced by the dynamic gesture selection and human subjects' performing proficiency.

For the second experiment in section 3, most subject spent more time on myoelectric control compared to joystick control. But for some subjects like subject 6, the time for the two control mode is similar. In addition, the myoelectric control time is much smaller than joystick control for subject 3. The majority results of the human subjects are fit the assumption that myoelectric control would use longer time than the joystick control. But time differences are not huge. Considering the extra processing time of executing the motions, catching the motions, wrong classification and recognition, the differences are acceptable. Moreover, the fixed trajectory make it impossible for the myoelectric control algorithm here to complete other task unless the commands are modified according to the specific requirements. But this help the control system perform stable in every task. During the experiments, myoelectric control is able to grab the bottled water successfully without knocking it down. But it happened when using joystick control. Because the trajectory of myoelectric control method is fixed. And the path by using joystick changes every time. In addition, the myoelectric control mode is easier to start than the joystick control. Human subjects do not need to learn the rules and practice for the task. Based on the discussion before, myoelectric control establish a potential in human computer interaction.

Manuscript 162 submitted to 2022 Modeling, Estimation and Control Conference.
Received April 23, 2022.

5. CONCLUSION

This paper proposed a CNN-LSTM neural network to classify five dynamic gestures. Based on the daily life habits of most people, five limb positions were chosen for each gesture. In the tests, motions were performed in the pre-defined limb positions. The model is able to recognize every gesture when it is under different limb positions, although the differences of accuracies on different positions still exist as shown in Table 1. After the proposed model is applied successfully to reduce the effect of limb positions for gesture recognition, an HCI system is designed based on the trained model. To use the system, human subjects need to wear a MYO armband for EMG signals collection. The gesture data would be input into the system and be recognized by the trained model. Then a command would be sent according to the corresponding relationship between gestures and commands. This system enables human subjects to manipulate a robot arm using pre-defined gestures.

Future work needs to concentrate on the following directions: the tiger mechanism in the real-time recognition system needs to be removed, the classification accuracy needs to be improved and the classification system needs to be more robust. Many neural networks achieve good results on classification work, like the Bidirectional LSTM neural network. In addition, methods like transfer learning which enable us to employ the knowledge from large public datasets, may improve the accuracy. However, the specific method needs to be further determined.

ACKNOWLEDGEMENTS

Thank you to my advisor Dr. Langari who encourages me to try and gives helpful guidance

REFERENCES

Cai, Z. and Zhu, Y. (2021). A hybrid CNN-LSTM network for hand gesture recognition with surface EMG signals. In X. Jiang and H. Fujita (eds.), *Thirteenth International Conference on Digital Image Processing (ICDIP 2021)*, volume 11878, 20 – 28. International Society for Optics and Photonics, SPIE. doi:10.1117/12.2601074.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Huang, D. and Chen, B. (2019). Surface emg decoding for hand gestures based on spectrogram and cnn-lstm. In *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, 123–126. doi:10.1109/CCHI.2019.8901936.

Huang, L., He, M., Tan, C., Jiang, D., Li, G., and Yu, H. (2020). Jointly network image processing: multi-task image semantic segmentation of indoor scene based on cnn. *IET Image Processing*, 14(15), 3689–3697. doi: https://doi.org/10.1049/iet-ipr.2020.0088.

Islam, M.Z., Islam, M.M., and Asraf, A. (2020). A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. *Informatics in Medicine Unlocked*, 20, 100412. doi: https://doi.org/10.1016/j.imu.2020.100412.

Mitra, S. and Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3), 311–324. doi:10.1109/TSMCC.2007.893280.

Mukhopadhyay, A.K. and Samui, S. (2020). An experimental study on upper limb position invariant emg signal classification based on deep neural network. *Biomedical Signal Processing and Control*, 55, 101669. doi: https://doi.org/10.1016/j.bspc.2019.101669.

Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R.J., Fredes, C., and Valenzuela, A. (2020). A review of convolutional neural network applied to fruit image processing. *Applied Sciences*, 10(10). doi: 10.3390/app10103443.

Ozdemir, M.A., Kisa, D.H., Guren, O., Onan, A., and Akan, A. (2020). Emg based hand gesture recognition using deep learning. In *2020 Medical Technologies Congress (TIPTEKNO)*, 1–4. doi: 10.1109/TIPTEKNO50054.2020.9299264.

Shanmuganathan, V., Y.H.K.M.e.a. (2020). R-cnn and wavelet feature extraction for hand gesture recognition with emg signals. 16723–16736. doi: https://doi.org/10.1007/s00521-020-05349-w.

Shin, S. (2016). Myoelectric human computer interaction using reliable temporal sequence-based myoelectric classification for dynamic hand gestures. URL `https : / /hdl .handle .net /1969 .1 /174271`.

Sun, Y., Xu, C., Li, G., Xu, W., Kong, J., Jiang, D., Tao, B., and Chen, D. (2020). Intelligent human computer interaction based on non redundant emg signal. *Alexandria Engineering Journal*, 59(3), 1149–1157. doi: https://doi.org/10.1016/j.aej.2020.01.015. A Special Section on: Computational Methods in Engineering and Artificial Intelligence.

Wang, Y., Wu, Q., Dey, N., Fong, S., and Ashour, A.S. (2020). Deep back propagation–long short-term memory network based upper-limb semg signal classification for automated rehabilitation. *Biocybernetics and Biomedical Engineering*, 40(3), 987–1001. doi: https://doi.org/10.1016/j.bbe.2020.05.003.

Xie, B., Meng, J., Li, B., and Harland, A. (2020). Gesture recognition from bio-signals using hybrid deep neural networks. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 493–499. doi: 10.1109/ICAICA50127.2020.9182510.

Yasen M, J.S. (2019). A systematic review on hand gesture recognition techniques. *PeerJ Computer Science*, 5:e218. doi:https://doi.org/10.7717/peerj-cs.218.

Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31(7), 1235–1270. doi:10.1162/neco$_{a0}$1199.

Manuscript 162 submitted to 2022 Modeling, Estimation and Control Conference.
Received April 23, 2022.