Proportional Myoelectric Control in a Virtual Reality Environment

Iliana Loi[1*], Angeliki Grammatikaki[1], Panagiotis Tsinganos[1], Efe Bozkir[2], Dimitris Ampeliotis[3], Konstantinos Moustakas[1], Enkelejda Kasneci[2] and Athanassios Skodras[1]

[1] Department of Electrical and Computer Engineering, University of Patras, Patras, Greece

[2] Department of Computer Science, University of Tübingen, Tübingen, Germany

[3] Department of Digital Media and Communication, Ionian University, Argostoli, Greece

*Corresponding author: loi@ceid.upatras.gr

2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP) | 978-1-6654-7822-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IVMSP54334.2022.9816252

***Abstract*—Translating input modalities such as hand interactions, speech, and eye tracking in virtual reality offers an immersive user experience. Especially, it is crucial to track the user's hand gestures, since they can help in translating user intentions into actions in virtual environments. In this work, we developed a virtual reality application which incorporates electromyography-based deep learning methods for recognizing and estimating hand movements in an online fashion. Our application automates all user controls, providing an immense potential for rehabilitation purposes.**

I. Introduction

Virtual reality (VR) applications can offer interactive and immersive 3D environments that can be used for a variety of purposes such as athlete training [1], educational training [2], industrial training [3], rehabilitation like prosthetic training [4] or even combining aspects like gaming and rehabilitation together [5]. In order to increase the realism and user experience in these applications, it is necessary to track and recognize hand gestures to convey the user intentions to the computer and perform the corresponding tasks [6]. Conventional methods for hand movement tracking include translating the hand input through VR controllers' signals, data gloves [7] and stereo cameras [8], or electromyographic (EMG) signal acquisition. The latter has gradually been drawing scientific attention, since EMGs contain a large amount of information and is widely used for human motor intention prediction [9], [10]. It is possible to use sophisticated sensors like armbands to record EMG data, which can be used as input to deep neural networks for gesture recognition and tracking [9], [11]. Except for gesture recognition, EMG signals can be used for continuous motion estimation (also known as proportional myoelectric control [10]), which in the case of upper limb control, refers to hand and finger joint prediction. Approaches based on deep learning are widely utilized for this task as well [12], [13], [14].

In this work, we developed a puzzle-like VR application which combines EMG-based techniques to recognize and estimate grasp movements that can be used in simple tasks such as grabbing and relocating objects. In our application, EMG signals are recorded via an armband and the hands' continuous motion and gestures are estimated via deep neural networks in an online fashion, unlike previous works, which are mostly testing their deep learning models offline [15]. Moreover, our application integrates two machine learning networks for gesture recognition and estimation, whereas similar works focus in developing models for either gesture recognition or joint angle estimation (e.g. [16], [12]). Since the full control of the application depends on the armband input, and not the VR controllers, this application can be easily deployed for people with hand dysfunctionalities or amputations, in order to help them exercise and rehabilitate underused muscles as well as to investigate arm fatigue issues.

II. Related Work

A. EMG-based Gesture Recognition

Gesture classification based on electromyography has enormous potential, as EMG signals capture the electrical activity of the muscles that eventually result in hand gestures [11]. Recent works in this field utilize deep learning models for EMG-based gesture recognition using either their own recorded datasets [15] or publicly available ones like Ninapro [17] and Myo's Dataset [18], [19]. In the case of recording data for either neural network training or inference, non-invansive methods to measure EMG data (surface EMG - sEMG) are used by most works, meaning that muscle signals are recorded from the surface of the skin using sensor devices like armbands [16], [20], [21]. Moreover, convolutional neural networks (CNNs) are preferred over other deep learning models, since they are able to extract informative, representative, and transferable features from EMG signals [12]. For example, one of the first works in this field, AtzoriNet [22], a CNN-based network, achieved a $66.6 \pm 6.4\%$ test accuracy on 50 moves of Ninapro DB1. Following works, aimed at increasing the performance, such as [23], where a modified version of the CNN in [22] achieved an improvement of 3% test accuracy on the same dataset, and [24] where a multi-stream CNN peaked with 84.4% on Ninapro DB1. In a subsequent work, the authors of [24] achieved a better performance with 87.4% accuracy when training with data from DB1, by introducing a multi-view CNN [25]. Furthermore, in [26] a test accuracy of 82.54% was achieved by training a CNN model with raw EMG values on DB1. Nevertheless, when trained with a smaller move set such as in [20], [15], where a data of six and seven simple hand gestures was used respectively, similar CNN architectures can reach a testing accuracy over 99%. This is due to the fact that the models are exposed to smaller portion of data with less variability; therefore, resulting in higher accuracy. More recently, [27] achieved test accuracy of 95.77% using a hybrid CNN and Bidirectional Long Short-Term Memory (BiLSTM) based architecture named EMGHandNet, which outperforms the multi-view CNN in [25].

B. EMG-based Finger Joint Angle Estimation

Estimating hand joint kinematics via EMG-based techniques is of interest to many. Similar to EMG-based hand gesture recognition, deep learning approaches are utilized such as CNNs and recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTM). The latter are widely used in human motion estimation and prediction [28], since they can model both short and long-term temporal dependencies in time-series. In [13], a simple LSTM architecture was trained using EMG data from 5 subjects of Ninapro's DB2 to estimate hand joint kinematics of grasp movements. This LSTM model performs better in terms of Pearson's Correlation Coefficient (CC), Root Mean Square Error (RMSE) and Normalized Root Mean Square Error (NRMSE) compared to other methods, such as Radial Basis Function Neural Network (RBF), which endorses the use of LSTM architectures for synchronous and proportional myoelectric control [13]. Moreover, a hybrid CNN-LSTM model was developed in [12] to investigate both temporal and spatial EMG information to be exploited in joint kinematics estimation. This model is also an alternative to conventional architectures such as RBF- and random forest (RF)-based approaches. More recently, [14] estimated the same grasp movements of DB2 as in [13], using a large-scale temporal convolutional network (LS-TCN) and discussed the superiority of LS-TCN over a simple temporal convolutional network. In particular, LS-TCN has a limited number of layers and a larger convolutional kernel size, which improve the model accuracy, and overcome the problem of losing temporal features from the EMG data that is caused when the depth of the model is increased. Similar processing methods for EMG signal filtering, smoothing, and feature extraction (e.g., RMS, MAV) as described in Section II-A, were also applied.

III. Methods

A. Application Overview

Our VR application is a more complex version of the children's toy "shape sorting cube". Using items of different shapes and sizes, users try to solve a puzzle-like task. In particular, users have to relocate randomly spawning objects from one workbench to the other, where the destination workbench have holes fitting to the shapes of the objects that they have to place. The target item (e.g., ball) is transferred to the target position when users perform the relevant gesture (e.g., power sphere grasp - See Fig. 1[1] for an example). After moving the target item to the correct hole and position, user score is increased. The objects that are used in the application are selected based on the movements our networks recognize and estimate. The key element of this application is the simultaneous functioning of two neural networks that are used to map the users' hand input control into the game: a CNN, which classifies gestures of users and a BiLSTM model that carries out finger joint angle estimation. Both predictions are based on sEMG signals measured via an armband.

The application was developed using Unity3D [29] game engine (version 2019.3.14f1) on a Windows 10 64-bit system used with the following specifications: AMD Ryzen 7 3700X, 3.60 GHz CPU, NVIDIA GeForce RTX 3060 Ti 4GB GPU and 48 GB, 931 MHz RAM. Furthermore, the EMG signal acquisition was conducted using the MindRove armband (MindRove, https://www.mindrove.com), which captures sEMG signals with 1×8 array of equally spaced around the forearm electrodes. The training of both CNN and BiLSTM networks was carried out in Python and the final models were imported in Unity in the Open Neural Network Exchange (ONNX) [30] format, which provides an open source format for deep learning and machine learning models.

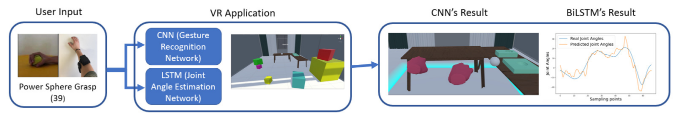

Fig. 1. A general overview of the functionality of the application. The user is performing movement No. 39 - power sphere grasp. The CNN model recognizes the performed move and the target item, i.e. a virtual ball, moves towards its corresponding hole/place. Simultaneously, the BiLSTM estimates the finger joint angles.

B. Dataset

We used DB1 of the Ninapro [17] dataset for training the EMG-based gesture recognition network, which contains sEMG data from 27 intact subjects, while performing 52 different hand movements (+1 one rest/idle movement). The data were recorded using Ottobock (Otto Bock HealthCare GmbH, https://www.ottobock.com), a device with 10 electrodes (10 sEMG channels) with a frequency of 100Hz. Furthermore, for training the EMG-based finger joint angle estimation network, we used Ninapro's DB9, a database added to Ninapro in 2020 [31]. This includes harmonized and calibrated kinematics data of Ninapro DB1. The acquisition of hand kinematics data was conducted using a 22-sensor CyberGlove II data glove (CyberGlove Systems LLC, https://www.cyberglovesystems.com), i.e., a motion capture data glove instrumented with 22 joint-angle sensors [31].

To refine the training process in terms of time and accuracy, we minimized the size of the input dimension, which corresponds to distinct number of movements. Thus, 6 out of 52 Ninapro's DB1 movements, which are depicted in Fig. 2, were selected in order to evaluate the performance of our neural networks in terms of recognition and estimation of different gestures. It is also possible to carry out comparisons between our LSTM architecture and previous works [13], as they used the same movement subset for training.

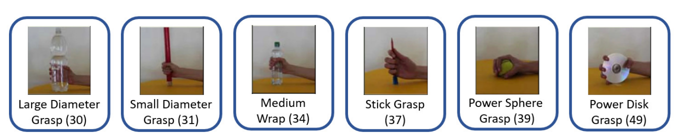

Fig. 2. Movements 30, 31, 34, 37, 39, and 49 of Ninapro's DB1 52 move set, which were used for training. Images are reproduced from [17].

C. CNN for EMG-based Gesture Recognition

For EMG-based gesture recognition, we used the CNN proposed in [23], a re-implementation of the model proposed in [22]. The architecture was modified to fit our inputs and outputs. The implementation was carried out using the Keras API for Python and the neural network consists of 4 hidden

[1]Full-sized figures can be found at https://drive.google.com/drive/folders/154Og1xZC__KDXrrSbpBKUmMWgJZjMRHd?usp=sharing

layers with ReLU activation functions and an output layer with Softmax activation function, as illustrated in Fig. 3. The input dimension of the model is $15 \times 8 \times 1$ (height $\times$ width $\times$ depth), meaning an EMG image whose height is equal to the length of the sliding window (15 samples in our case) and whose width matches the number of MindRove's EMG channels/electrodes. The output of this network is a 1×6 vector, containing the probability of each gesture. However, the EMG input images of the model proposed in [23] were of dimensions $15 \times 10 \times 1$, since Ninapro's DB1 data were recorded with an armband device that has 10 electrodes. Additionally the output of that model was a probability vector of length 52. Given that in our case the input is $15 \times 8 \times 1$ and the output vector has a length of 6, the hyperparameters of the layers of the CNN had to be adjusted. The modifications included adjusting the zero-padding values of the intermediate convolutional layers and changing the number of filters in the last layer to 6.

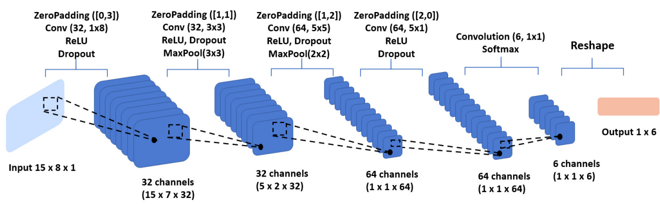

Fig. 3. The CNN architecture based on [23]. Modifications were made to the initial architecture according to our inputs and outputs.

Before training, basic filtering and augmentation methods described in [11] were used to pre-process sEMG data of Ninapro DB1, which is described as follows. (1) A 5^{th} order Butterworth low-pass filter with 1Hz cuttoff frequency was applied to the data for smoothing. Smoothing was performed in order to extract the sEMG signal's amplitude estimate, which is used for the gesture classification task. (2) For data augmentation, a sliding window of 150ms length (corresponding to 15 samples) with 60% overlapping (corresponding to 6 samples overlapping) was applied to the sEMG signals to generate training examples that have a set duration. (3) Min-max normalization was applied to use a common scale for the data used for training. Furthermore, to evaluate the performance of different training instances of the model, we used the inter-subject evaluation scheme [11], where the test dataset consists of all movements of one subject and the training dataset comprises of the data from the remaining $N-1$ subjects, where $N = \#subjects$ (leave-one-subject-out cross-validation). This procedure was iterated until the data of each subject were in the test set. This way, our network generalizes well with unseen data.

As MindRove produces raw sEMG values, that are noisy and have a more random and non-stationary nature, they are not suitable for gesture recognition [25]. Therefore, basic signal processing methods are applied on these signals before inference, which are as follows. (1) Raw EMG values are transformed to real values by multiplying with LSB $= 0.0045 * 10^{-6}V$. (2) The Root-Mean-Square (RMS) values for every 5 samples (with step size of 1) of the real EMG signals were calculated. This step was crucial since, Ninapro's DB1, which was used for training our gesture recognition network, contains also the RMS rectified version of the raw sEMG signal. (3) An anti-aliasing filter, namely a 1^{st} order low-pass Butterworth filter with 50Hz cut-off frequency, was applied on the RMS version of EMG data, followed by down-sampling at every 5 samples. This was done since MindRove has a sampling rate of 500 Hz, which is 5 times more the sampling rate of Ottobock (100 Hz), i.e. the sensors used for recording the data of DB1. (4) Min-max normalization was also applied to the data from MindRove. For the normalization step, the min-max values of the RMS rectified EMG signals produced by the armband, were used, since training data (Ninapro) have a different nature from inference data (MindRove).

D. BiLSTM for EMG-based Joint Angle Estimation

For hand joint kinematics estimation based on EMG signals, a Bidirectional LSTM was developed. As shown, in Fig. 4, each Bi-LSTM layer consists of two parallel LSTM layers to provide a forward and a backward sequence. These two loops are used to exploit both past and future information from data to make predictions [32]. The latter results in increasing prediction accuracy and, thus, is the main reason why Bi-LSTM is usually preferred over the unidirectional LSTM architecture. Our LSTM model has an input of 8 EMG features, according to output of MindRove, and consists of 3 Bidirectional LSTM layers with ReLU activation functions, each one of them containing 128 hidden units. The output layer is a fully connected layer with also 128 input units and an output of 19 joint angles. To prevent overfitting, a dropout layer is added before the output layer.

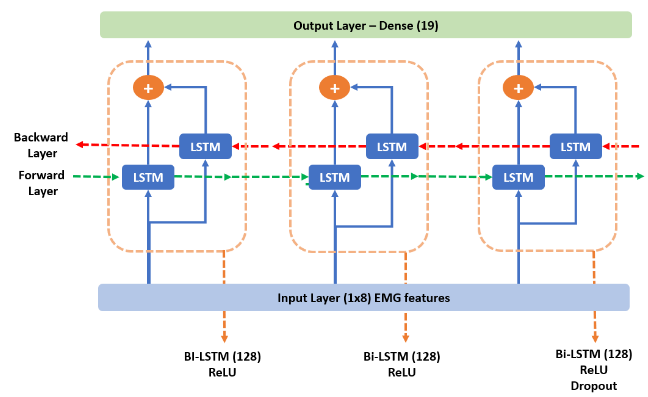

Fig. 4. A general overview of the architecture of the Bidirectional LSTM.

For training this model, DB1 and DB9 from Ninapro were used. As aforementioned, DB9 includes the harmonized and calibrated version of kinematic data of Ninapro DB1, which were acquired using a 22-sensor data glove, thus measuring 22 joint angles. Similar to [13], as we are interested only in grasp movements, we excluded joints No. 20, 21 and 22, for which the sensors are located at the wrist and outside the fingers' area; thus, predicting only 19 joint angles. Supplementary, aforementioned pre-processing methods in Section III-C were used for preparing the data of DB1 for training, and a similar inter-subject evaluation technique was used for training the BiLSTM. Nevertheless, data from only 6 subjects from DB1 that cover all subjects' characteristics as much as possible (4 males and 2 females aged between 25-31, out of which 4 were right-handed and 2 left handed) were used for training the model, just as in [13], [14], where also a small subset of subject data from Ninapro were utilized. Before feeding armband data in real-time into the BiLSTM model, raw EMG values of MindRove were also pre-processed as mentioned in Section III-C.

IV. Results

A. CNN results

Our CNN model achieves an accuracy of $80 \pm 2.36\%$, a performance which is comparable with other CNN architectures like [22], [24] and [26], which are also trained upon the full move set of DB1 of Ninapro (52 movements) and using window lengths of 150ms. Yet, our model does not achieve a performance in the vicinity of 100% accuracy like state-of-the-art approaches such as [27], [15], where a test accuracy of 95.77% and 99.59% was reported, respectively. According to Table I, one reason for this is that most models (e.g. [24], [27]) achieved higher performances using a subject-wise (intra-subject) evaluation technique, whereas, we used an inter-subject evaluation method to train a model that can generalize to new unseen data. Intra-subject evaluation achieves higher accuracy values than the inter-subject training technique, because the model has a better understanding of hand gestures due to low intra-subject signal covariance [27] and thus, it is usually preferred over aggregated data schemes. Moreover, similar CNN architectures such as [20], [15], can reach an outstanding testing accuracy, since they are trained with datasets containing a small set of movements, which results in higher accuracy due to the fact that the models are exposed to a smaller portion of data with less variability.

Another reason why our model shows poorer performance is that it uses fewer inputs than the initial model as described in [23], which may lead to lower accuracy values. In particular, the CNN proposed in [23] was trained with EMG input images of dimensions $15 \times 10 \times 1$, since the data from DB1 of Ninapro recorded with Ottobock, which has 10 electrodes. In our case, we trained the same model with smaller inputs, namely $15 \times 8 \times 1$, to match the output of MindRove (8 EMG signals instead of 10). This reduction in the dimensions of the input images results in accuracy loss, due to obtaining less informational content during training.

Nevertheless, compared with approaches that also used inter-subject evaluation techniques, our model performs significantly better than other more complex schemes, such as in deep domain adaptation approaches [33] and [34], which achieved an accuracy of 67.4% and 65.7% on Ninapro DB1, respectively, and state-of-the-art approaches like [35], where the highest test accuracy recorded is 73.35% in DB1.

B. BiLSTM results

To evaluate the performance of our BiLSTM network based on the range of real joint angles, we used the Normalized Root Mean Square Error (NRMSE) and Pearson Correlation Coefficient (CC) metrics, which are reported in Table II. From Table II, it is shown that our model achieves an overall average NRMSE of 13% and an average CC of 90%. This NRMSE value is an indication that our model performs well given that, lower NRMSE values suggest better performance. Moreover, similar architectures like the LSTM model proposed in [13] - where predictions for the same 6 movements were made based on DB2 of Ninapro - produced an average NRMSE value close to ours, $15.27\% \pm 0.02\%$. As for our CC values reported in Table II, coefficients closer to 1 indicate high correlation between real values (Ninapro data) and predicted values (BiLSTM predictions). In addition to the overall performance, preliminary tests of this model and our gesture recognition network runs online approximately at 60 FPS, indicating a real-time capability of the system.

TABLE I.

Publication	Test Accuracy	Evaluation Scheme	Dataset	#Moves
[22]	$66.6 \pm 6.4\%$	Intra-Subject	Ninapro DB1	52
[33]	67.4%	Inter-Subject	Ninapro DB1	52
[24]	84.4%	Intra-Subject	Ninapro DB1	52
[20]	99%	-	Own Dataset	6
[34]	65.7%	Inter-Subject	Ninapro DB1	52
[26]	82.54%	-	Ninapro DB1	52
[15]	99.59%	-	Own Dataset	7
[27]	95.77%	Intra-Subject	Ninapro DB1	52
[35]	73.35%	Inter-Subject	Ninapro DB1	22
Ours	$80 \pm 2.36\%$	Inter-Subject	Ninapro DB1	52

[a] Test accuracy of EMG-based hand gesture recognition approaches. Similar to previous works, by "test accuracy", we refer to testing the model offline using Ninapro Data. Moreover, in the intra-subject approach, data from one subject are used for training and validating the network, and this process repeats for all subjects available in the dataset [11]. As a result, one model is created for every subject and performance is reported accordingly.

TABLE II.

Metric	M_{30}	M_{31}	M_{34}	M_{37}	M_{39}	M_{49}	Average
NRMSE	0.13	0.10	0.12	0.13	0.12	0.13	0.13
CC	0.90	0.94	0.90	0.87	0.92	0.86	0.90

[b] The average NRMSE and CC values for estimating 19 finger joint angles for each one of the 6 movements. $M_{\#}$ indicates the number of the movement as illustrated in Fig. 2

V. Conclusion

Going beyond the state of the art, our approach is combining the use of a CNN and a BiLSTM for recognizing and estimating hand gestures, unlike previous works, which are addressing either one of the two problems. Both models have simple architectures, yet presenting good or even better performance comparing with recent works as discussed in Section IV. More specifically, our convolutional network classifies 6 specific grasp tasks with an accuracy of $80 \pm 2.36\%$, a performance that is comparable with similar works evaluated with an inter-subject way, while BiLSTM gives feedback of the joint angles measured while performing a task online. Our BiLSTM model scores a high average CC of 90% and a low average NRMSE of 13%, indicating a good performance. In this application, VR controllers are replaced by virtual hands, whose function depends on the EMG signals measured via the armband. Since the control of the application is fully automated, it can be used for rehabilitation purposes for amputees.

In future work, we plan to extend this application with movements other than grasp and add more complex tasks for users. In addition, we will carry out user studies to evaluate user experience. Lastly, we plan to include interaction with more than one input modalities such as speech recognition [8], [36] and eye tracking [37], [38].

Acknowledgment

This work is supported by German-Greek Universities cooperation program of German Academic Exchange Service (DAAD) - Project OMEGA (Project ID 57515461) 2020-2022.

REFERENCES

[1] A. Batmaz, A. Mutasim, M. Malekmakan, E. Sadr, and W. Stuerzlinger, "Touch the Wall: Comparison of Virtual and Augmented Reality with Conventional 2D Screen Eye-Hand Coordination Training Systems," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2020, pp. 184–193.

[2] H. Gao, E. Bozkir, L. Hasenbein, J.-U. Hahn, R. Göllner, and E. Kasneci, "Digital transformations of classrooms in virtual reality," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, 2021, pp. 483:1–483:10.

[3] M. Pavlou, D. Laskos, K. Risvas, and K. Moustakas, "XRSISE: An XR Training System for Interactive Simulation and Ergonomics Assessment," *Frontiers in Virtual Reality*, vol. 2, 2021.

[4] I. Phelan, M. Arden, C. Garcia, and C. Roast, "Exploring Virtual Reality and Prosthetic Training," in *2015 IEEE Virtual Reality (VR)*, 2015, pp. 353–354.

[5] M. Aulisio, D. Han, and A. Glueck, "Virtual Reality Gaming as a Neurorehabilitation Tool for Brain Injuries in Adults: A Systematic Review," *Brain Injury*, no. 34, pp. 1–9, 2020.

[6] Y. Li, J. Huang, T. Feng, W. Hong'an, and D. Guozhong, "Gesture Interaction in Virtual Reality," *Virtual Reality and Intelligent Hardware*, vol. 1, p. 9, 2019.

[7] H. Min, C. Chen, S. Huang, X. Tian, Y. Yang, and Z. Wang, "Highly-Accurate Gesture Recognition Based on ResNet with Low-Budget Data Gloves," in *AISS 2021: 2021 3rd International Conference on Advanced Information Science and System*, 2021, pp. 1–6.

[8] K. Moustakas, M. Strintzis, D. Tzovaras, S. Carbini, O. Bernier, J. Viallet, S. Raidt, M. Mancas, M. Dimiccoli, E. Yagci, S. Balci, and E. Leon, "Masterpiece: Physical Interaction and 3D Content-Based Search in VR Applications," *Multimedia, IEEE*, vol. 13, pp. 92 – 100, 2006.

[9] D. Xiong, D. Zhang, X. Zhao, and Y. Zhao, "Deep Learning for EMG-based Human-Machine Interaction: A Review," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, pp. 512 – 533, 2021.

[10] L. Bi, A. Feleke, and C. Guan, "A review on EMG-based Motor Intention Prediction of Continuous Human Upper Limb Motion for Human-Robot Collaboration," *Biomedical Signal Processing and Control*, vol. 51, pp. 113–127, 2019.

[11] P. Tsinganos, A. Skodras, B. Cornelis, and B. Jansen, *Deep Learning in Gesture Recognition Based on sEMG Signals*. CRC Press, 2018.

[12] T. Bao, S. A. R. Zaidi, S. Xie, P. Yang, and Z. Zhang, "A CNN-LSTM Hybrid Framework for Wrist Kinematics Estimation using Surface Electromyography," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[13] C. Wang, G. Weiyu, H. Zhang, L. Guo, C. Huang, and C. Lin, "sEMG-based Continuous Estimation of Grasp Movements by Long-Short Term Memory Network," *Biomedical Signal Processing and Control*, vol. 59, p. 101774, 2020.

[14] C. Chen, G. Weiyu, C. Ma, Y. Yang, Z. Wang, and C. Lin, "sEMG-Based Continuous Estimation of Finger Kinematics via Large-Scale Temporal Convolutional Network," *Applied Sciences*, vol. 11, p. 4678, 2021.

[15] M. A. Ozdemir, D. H. Kisa, O. Guren, A. Onan, and A. Akan, "EMG based Hand Gesture Recognition using Deep Learning," in *2020 Medical Technologies Congress (TIPTEKNO)*, 2020.

[16] Z. Karapinar Senturk and B. Melahat Sevgul, "Machine Learning Based Hand Gesture Recognition via EMG Data," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 10, no. 2, 2021.

[17] A. Atzori, Manfredo Gijsberts, C. Castellini, B. Caputo, A.-G. Mittaz Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography Data for Non-Invasive Naturally-Controlled Robotic Hand Prostheses," *Nature*, vol. 1, 2014.

[18] U. Côté Allard, C. L. Fall, A. Campeau-Lecours, C. Gosselin, F. Laviolette, and B. Gosselin, "Transfer Learning for sEMG Hand Gestures Recognition Using Convolutional Neural Networks," 2017, p. 1663–1668.

[19] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 4, pp. 760–771, 2019.

[20] J. Pinzón Arenas, R. Moreno, and J. Herrera, "Convolutional Neural Network for Hand Gesture Recognition using 8 Different EMG Signals," in *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, 2019.

[21] N. Tsagkas, P. Tsinganos, and A. Skodras, "On the Use of Deeper CNNs in Hand Gesture Recognition Based on sEMG Signals," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2019, pp. 1–4.

[22] M. Atzori, M. Cognolato, and H. Müller, "Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands," *Frontiers in Neurorobotics*, vol. 10, 2016.

[23] P. Tsinganos, B. Cornelis, C. Jan, B. Jansen, and A. Skodras, "Deep Learning in EMG-based Gesture Recognition," in *5th International Conference on Physiological Computing Systems*, 2018, pp. 107–114.

[24] W. Wei, Y. Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, "A Multi-Stream Convolutional Neural Network for sEMG-Based Gesture Recognition in Muscle-Computer Interface," *Pattern Recognition Letters*, vol. 119, 2017.

[25] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, and W. Geng, "Surface-Electromyography-Based Gesture Recognition by Multi-View Deep Learning," *IEEE Transactions on Biomedical Engineering*, pp. 1–1, 2019.

[26] Y. Cheng, G. Li, M. Yu, D. Jiang, J. Yun, Y. Liu, Y. Liu, and D. Chen, "Gesture Recognition Based on Surface Electromyography-Feature Image," *Concurrency and Computation: Practice and Experience*, vol. 33, 2020.

[27] N. Karnam, S. R. Dubey, A. Turlapaty, and B. Gokaraju, "EMGHandNet: A Hybrid CNN and Bi-LSTM Architecture for Hand Activity Classification using Surface EMG Signals," *Biocybernetics and Biomedical Engineering*, vol. 42, 2022.

[28] J. Martinez, M. J. Black, and J. Romero, "On Human Motion Prediction Using Recurrent Neural Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4674–4683.

[29] J. K. Haas, "A History of the Unity Game Engine," 2014.

[30] "Open Neural Network Exchange Open Source Format," https://github.com/onnx/onnx, accessed: 2022-04-09.

[31] N. Jarque-Bou, M. Atzori, and H. Müller, "Large Calibrated Database of Hand Movements and Grasps Kinematics," *Scientific Data*, vol. 7, 2020.

[32] I. Ihianle, A. Nwajana, S. Ebenuwa, R. Otuka, K. Owa, and M. Orisatoki, "A Deep Learning Approach for Human Activities Recognition From Multimodal Sensing Devices," *IEEE Access*, vol. 8, pp. 179 028–179 038, 2020.

[33] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, "Surface EMG-Based Inter-Session Gesture Recognition Enhanced by Deep Domain Adaptation," *Sensors*, vol. 17, p. 458, 2017.

[34] I. Ketyko, F. Kovács, and K. Z. Varga, "Domain Adaptation for sEMG-based Gesture Recognition with Recurrent Neural Networks," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–7.

[35] N. Zheng, Y. Li, W. Zhang, and M. Du, "User-Independent EMG Gesture Recognition Method Based on Adaptive Learning," *Frontiers in Neuroscience*, vol. 16, 2022.

[36] K. Moustakas, D. Tzovaras, L. Dybkjær, N. Bernsen, and O. Aran, "Using Modality Replacement to Facilitate Communication between Visually and Hearing-Impaired People," *Multimedia, IEEE*, vol. 18, pp. 26 – 37, 2011.

[37] E. Bozkir, P. Stark, H. Gao, L. Hasenbein, J.-U. Hahn, E. Kasneci, and R. Göllner, "Exploiting Object-of-Interest Information to Understand Attention in VR Classrooms," in *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, 2021, pp. 597–605.

[38] E. Bozkir, D. Geisler, and E. Kasneci, "Assessment of driver attention during a safety critical situation in VR to generate VR-based training," in *ACM Symposium on Applied Perception*. ACM, 2019, pp. 23:1–23:5.