

EMG-Based Hand gesture recognition for myoelectric prosthetic hand control

BOUCETTA Lakdhar Nadjib , CHEMLOUL Bilal , REBAI Karima
Electrical Engineering and Computer Science Department, ENST, Algiers-Algeria

Em ail: l_boucetta@enst.dz , b_chemloul @enst.dz , Karima.rebai@enst.dz

Abstract – The work presented in this paper aims to contribute to the development of a deep learning-based approach to recognize hand movements using surface electromyography signals (sEMG). This will be used subsequently for the incarnation of these movements using a prosthetic hand.

The developed classification mechanism combines the proprieties of the convolutional neural network (CNN) known by its ability to extract EMG signal features with the properties of the Long short-term memory (LSTM) cells that extract and learn sequential properties of the time series phenomena. The developed approach could achieve an accuracy rate of 98.8% using the "sEMG for Basic Hand movements Data Set" dataset. To our knowledge, the obtained result is one of the highest accuracy values among all researches done using this dataset.

To develop a complete prosthetic hand system, a prosthetic hand prototype was designed. It was controlled by the hand recognition unit in order to embody the six movements (cylindrical, hook, lateral, palmar, spherical, and peripheral) available in the dataset used.

Index Terms—Hand gesture recognition, sEMG signals, Deep learning, Convolutional neural networks (CNN), Long Short- Term Memory (LSTM), Prosthetic hand.

I. INTRODUCTION

For decades, millions of people have suffered from limb amputations like hands. This often caused a functional disability and psychological issues that deprived them to perform many essential vital functions.

Faced with this intractable situation, it was necessary to search for solutions to replace the

amputated organs with artificial ones that would help them perform their daily tasks.

With the development of the technology, the efficiency of prosthetic limbs has increased to witness the peak of its development in the current era with the development of many control technologies that allow prosthetic limbs to be obtained with a greater number of movements, a larger functional area, greater comfort, and a more natural appearance.

Myoelectric prostheses are one of the most effective prosthesis technologies created for people with upper limb amputations. They are externally powered artificial limbs dominated by electrical signals generated by muscles called Electromyogram signals (EMG). Controlling schemes of myoelectric hand prostheses can be split into two categories: non-pattern recognition and pattern recognition-based methods. The non-pattern recognition methods (e.g., on/off controllers and/or finite state machines) provide pre-defined control commands based on a sequence of muscle contractions. In this work, the pattern recognition-based approach or the so-called pattern recognition control will be discussed.

The pattern recognition-based control is the extensively studied myoelectric control strategy, in which it is assumed that there is a set of distinguishable and repeatable patterns in EMG signals during different types of movements. The patterns are learned by pattern recognition algorithms that consist of two major steps: feature extraction and classification. Depending on these two steps, the classification of the surface electromyography signals (sEMG) can be done using one of the following three methods:

- Conventional approach: Features extraction and classical classifier.
- Hand-crafted features in deep learning algorithms.

- End to end approach.

II. RELATED WORKS

The classification of EMG signals can be done with different approaches, as mentioned previously. In this section, we will present the two last approaches as they are based on deep learning techniques.

In recent years, deep learning has emerged as one of the most popular Artificial Intelligence techniques, yielding state-of-the-art results for a range of supervised and unsupervised tasks. It is part of a more prominent family of machine learning methods dealing with algorithms inspired by the structure and function of the brain called Artificial Neural Networks (ANN). The superior performance of deep learning resides in the ability to incorporate the feature extraction step, i.e., the ability to extract high-level features from raw data using statistical learning on large datasets. This approach produces effective representations of the input space, based on powerful discriminative features, different from the traditional techniques based on non-deep ML, built on hand-crafted features or rules designed by expert researchers.

This considerable increase in the popularity of deep learning has provided a new perspective to analyze sEMG signals for hand gesture recognition. Currently, classify hand gestures using deep learning can be made using two methods. The first one is based on the separation between feature extraction and classification stages. This method is not widely used because deep architectures have the enough capacity to learn good feature representations on their own.

Typically, the most architecture used in the **Hand-crafted features with deep learning algorithms** approach is the Multilayer Perceptron (MLP) [1]. This architecture does not have the sufficient capacity to learn features such as Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN). For this, it requires a set of input extracted characteristics. Zhang et al. [1] proposed a real-time hand gesture recognition model using sEMG. A feed-forward artificial neural network (ANN) was used for this purpose with a sliding window approach to segment data during feature extraction. The proposed model achieved an average recognition rate of 98.7%.

The second approach is the end-to-end learning of the relationship between raw inputs and classes, which is the most progressive approach in the last years. This approach bypasses the intermediate steps used in traditional pipeline designs. Many researchers used the convolutional neural network (CNN) for myoelectric pattern recognition purposes. In [2], authors compare the success rate of four different CNN architectures applied on the UCI dataset [3], with different signal inputs (raw signals, Fourier transform of the signal; the root means square, and the Empirical mode decomposition (EMD)). In another study [4], Ulysse Côté-Allard et al. proposed the application of a convolutional neural network (CNN) augmented using transfer learning techniques for two datasets of 18 and 17 participants recorded using the Myo armband device. The proposed CNN architecture achieved an average accuracy of 97.81% on seven hand/wrist gestures of the second data set.

In another way, some researchers proposed to consider the sequential character of the EMG signal and developed approaches that exploit Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM). M. Simão et al. [5] proposed the use of an LSTM recurrent neural network to classify gestures of [6]. They were able to recognize hand gestures with an accuracy greater than 90%.

Xia et al. proposed in [7], the development of a CNN-LSTM architecture to classify the surface electromyography signals (sEMG), this architecture was composed of successive convolution, pooling, and LSTM layers. It is essential to notice that the inputs of this model are time-frequency vectors. This architecture achieved an accuracy rate of 90%. In another work, Nahid et al. [8] proposed a comparative study of classifying different hand gestures of two well-known surface Electromyogram (sEMG) data sets, Rami Khusaba EMG repository and UCI Machine Learning Repository [3] using transfer learning and CNN-LSTM neural network architectures. The combination of CNN and LSTM has achieved state-of-the-art accuracy for these data sets, 99.72% for the UCI Machine Learning Repository and 99.83% for the Rami Khusaba EMG repository.

In the light of these studies, it was decided to develop an architecture that combines the Convolutional Neural Network layers (CNN) and Long Short-Term Memory layers (LSTM) since it incorporates the ability to extract features provided by CNN layers and also the ability to learn the given time-dependent features provided by LSTM layers.

The particularity of our proposed architecture compared to most works cited above which use the CNN-LSTM architecture resides in the use of one-dimensional convolutional layers instead of the two-dimensional ones as done in [7], [8], this choice is motivated by the fact that inputs are of a one-dimensional nature. The second particularity is the wrapping of the CNN layers in time distributed layers, which allow applying the same layer to several inputs and produce one output per input to get the result in time. This will be useful as these CNN layers are followed by LSTM layers, which are known for their ability to process time-series data.

III. MATERIALS AND METHODS

A. EMG Data Collection

In this work, and due to our inability to provide an adequate amount of data to train the CNN-LSTM architecture, it was decided to use the "sEMG for Basic Hand movements Data Set".

This open-source dataset is got from UC Irvine Machine Learning Repository (UCI) [3]. It contains signals of different people grasping different objects. Three surface EMG electrodes attached to the forearm are used for getting signals, two of them in Flexor Capri Ulnaris and Extensor Capri Radialis and the reference electrode in the middle, in order to gather information about the muscle activation. For the data collection, five healthy subjects (two males and three females) of the same age approximately (20 to 22-year-old) were asked to perform six movements each for about 6 seconds with 30 trials [3]. The following movements are the most commonly used in daily life:

1. **Spherical (S):** for holding spherical tools.
2. **Tip (T):** for holding small tools.
3. **Palmar (P):** for grasping with palm facing the object.
4. **Lateral (L):** for holding thin, flat objects.
5. **Cylindrical (C):** for holding cylindrical tools.
6. **Hook (H):** for supporting a heavy load.

The data were collected at a sampling rate of 500 Hz, using as a programming kernel the National Instrument's (NI) LabVIEW [3]. The signals were band-pass filtered using a Butterworth band-pass filter with low and high cutoff 15Hz and 500Hz

respectively and a notch filter at 50Hz to eliminate line interference artifacts.

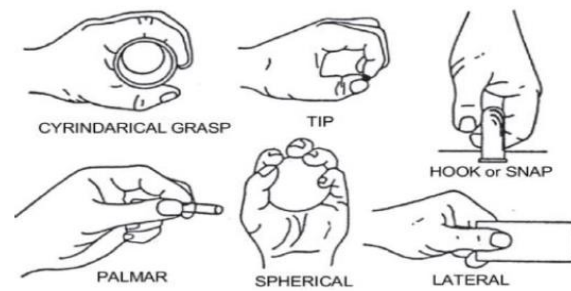


Fig. 1. Gestures performed in "sEMG for Basic Hand movements Data Set," [3]

B. Preprocessing

To focus only on segments where the muscle is contracted, we applied the sliding window approach. This approach serves to decompose the signal into segments where possible muscle activities of value for feature extraction might exist. This improves the classification performance.

The windowing technique is divided into types: adjacent and overlapping windowing. The signal is divided into windows, with the end of the previous window is associated with the following window in the adjacent windowing technique. In the overlap windowing, pieces of the past and next window are overlapped, as illustrated in *fig. 2*.

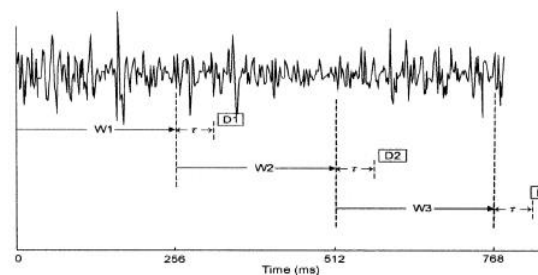


Fig. 2. Signal windowing

In this study, signals were segmented with a 300 ms window and 60 ms of overlap between windows. In the end, a data matrix of 89991x150x2 was got, so 89.991 window samples were generated with a data length of 150.

After splitting the dataset into a set of 300 ms windows, the final step of processing was devoted to organizing this data into training, validation, and testing sets. The training samples represent 75% of the dataset, while the remaining 25% was divided into

the validation and test samples with proportions of 10% and 15%, respectively.

C. Classification using the CNN-LSTM architecture

The architecture proposed in this work is a hybrid architecture containing convolution layers used for extracting features from the sEMG signal, combined with recurrent layers that support sequence prediction.

The model (*fig.3*) is made up of two one-dimensional convolutional layers (1D-CNN), where each one is followed by a max-pooling layer. The number of filters used in convolutional layers is 96, with a kernel size of 3. The approach used to implement this model consists of dividing each window of 150-time steps into subsequences of 30-time steps, allowing the CNN layers to process them using the time-distributed layer.

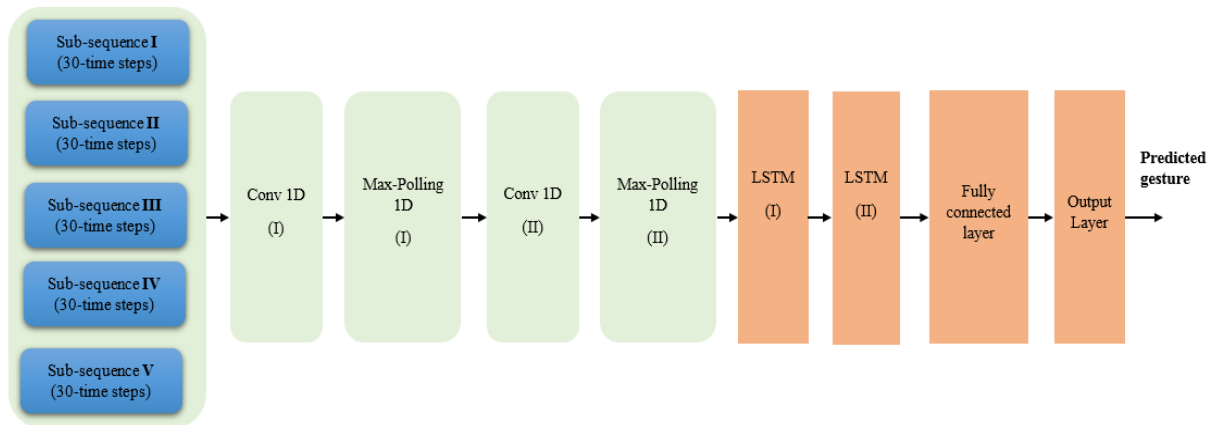


Fig. 3. CNN-LSTM architecture proposed

IV. EXPERIMENT AND RESULTS

In deep learning applications, even after finding the appropriate architecture, it is still necessary to look for optimal hyper-parameters that improve the algorithm's performance. Generally, there are several possible techniques to achieve this, such as performing a grid search where several combinations of parameters are tested and comparing the results. The approach used in this paper is based on testing the model with only one parameter at a time and keep the others parameters constants. This gave us the advantage of seeing the effect of each parameter on model learning. At the end of each test, the hyper-parameter value provided the best result is kept for the subsequent tests.

The entire convolutional layers are wrapped in the Time Distributed layer, which applies the same layer to several inputs and produces one output per input to get the result in time; this allows the same CNN model to read in each of the five subsequences of the window.

The feature vectors extracted by convolutional layers will then be the input of two stacked recurrent networks, where the first one is a many-to-many LSTM layer with 80 nodes. As for the second, it is a many-to-one LSTM layer with 64 nodes. Dropout layers with the probability of 0.3 and 0.25 follow each of the LSTM layers, respectively. In the end, there is a fully connected layer containing 32 nodes, followed by a Softmax layer with six nodes that correspond to the number of classes.

In most cases, the process of searching for optimal hyper-parameter values begins with testing the architecture with random parameter values. This allowed us to get an idea of the model's performance to improve it in the phase of hyper-parameter tuning.

For this, the first step made was to test the CNN-LSTM model with parameters indicated in *Table I*.

Table I. The initial hyper-parameter values used for the model training

Activation function	Tanh
Optimizer	Adam
Learning rate	0.001
Number of epochs	100
Batch size	16

To assess the model's performance during the training phase, we check after each epoch if there is an improvement in the validation accuracy. If so, then weights and biases will be stored. At the end of 100 epochs, the model that provides the best validation accuracy will be used for the test phase. This can be seen as a kind of early stopping.

The model evaluation comes out with the following accuracy curve (fig.4) and the confusion matrix (fig.5).

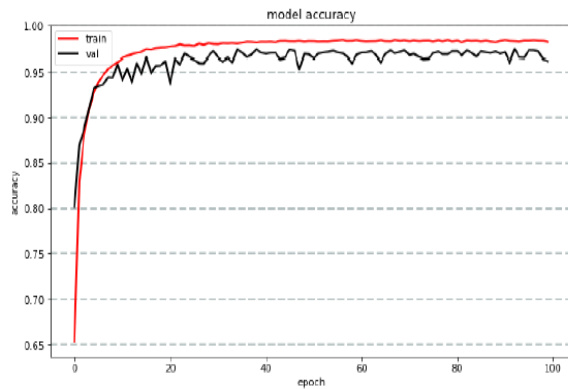


Fig. 4: Model accuracy curve using the training and validation data.

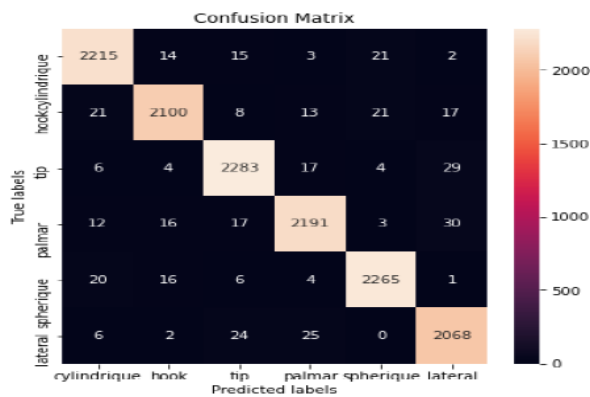


Fig. 5. Confusion matrix

From the resulting accuracy curve, we note that the CNN-LSTM model tested using arbitrary parameter values gave satisfactory results. The accuracy was 98.38% during the training phase and 97.2% in the testing phase.

In addition, we can also observe from the confusion matrix that the classification errors are almost evenly distributed across all classes, which means that the classification has no bias in favor of a class.

Despite the promising results achieved, it remains possible to improve the accuracy values, reduce the loss error, and reduce the training period. This is

through testing the model with different hyper-parameter values to get the optimal model.

For this, four hyper-parameters were tested namely, the activation function, learning rate, number of epochs, and batch size.

A. Final structure

The final structure of the developed CNN-LSTM model is the same as that presented in fig.6.

As a final evaluation of the results, the CNN-LSTM model was again trained using the best set of parameters shown in Table II.

Table II: Hyper-parameters used in the final model

Activation function	Relu
Optimizer	SGD
Learning rate	0.2
Number of epochs	100
Batch size	128

Results obtained after the evaluation of this model are represented by the accuracy curve (fig.6) and the confusion matrix (fig.7).

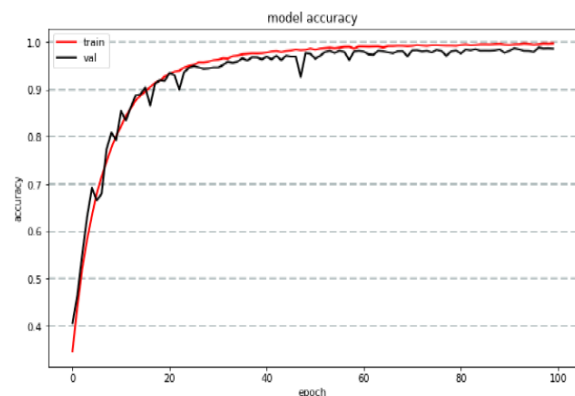


Fig. 6. Model accuracy curve in the train and validation phase

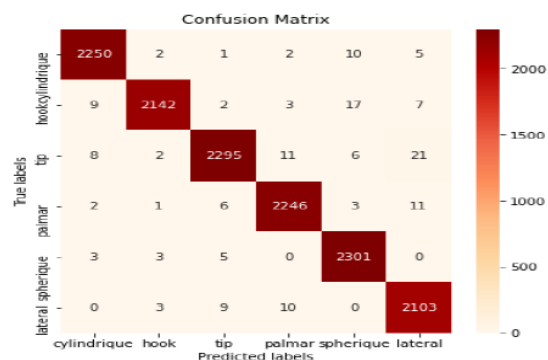


Fig. 7. Confusion matrix

The model evaluation comes out with an accuracy of 98.8%, which is an excellent result. By looking at *fig. 6* and *fig. 7*, some observations can be extracted. At first, the accuracy values in the training and validation phases are very close, which confirms the absence of the overfitting problem. The second remark to note from the confusion matrix is that the classification errors are almost equally distributed in all classes, which confirms the above observation regarding the unbiasedness of the model for any of the classes.

V. HAND PROSTHESIS SYSTEM

Prosthetic hands are artificial devices designed for hand amputees to provide them with certain functions of human hands. Depending on the application of these prostheses and some other criteria, there are several types of control. Perhaps the most important and widely used is one that uses the EMG signal.

In the previous section, a mechanism was developed for the recognition of hand gestures based on the CNN-LSTM model. Subsequently, the embodiment of these hand movements using a myoelectric prosthesis.

The final project architecture comprises five units (*fig.8*): Data unit, hand gesture recognition unit, control unit, actuation unit, and also the printed prosthetic hand. In the data unit, data can be either signals captured in real-time from the patient limb using an EMG sensor, or pre-existing signals, as done in this project. The hand gesture recognition unit is a sub-system that classifies hand gestures based on

EMG signals provided by the previous unit; this unit was described previously.

The control unit receives the classification result of the EMG signal and sends the corresponding command signal to the actuation unit. This latter consists of four servo motors. According to the command signal provided by the control unit, servo motors move to certain angles, pulling the printed prosthetic hand's fingers to a specific position to form the required gesture. Figure (*fig.8*) shows system architecture and its units.

To avoid confusion, we will use the expression "**Printed prosthetic hand**" to refer only to the mechanical structure. In contrast, we will refer to the system comprising (control unit + actuator unit + printed prosthetic hand) by the expression "**prosthetic hand**".

Figures (*a-f (fig.9)*) show the different movements performed by the developed prosthetic hand.

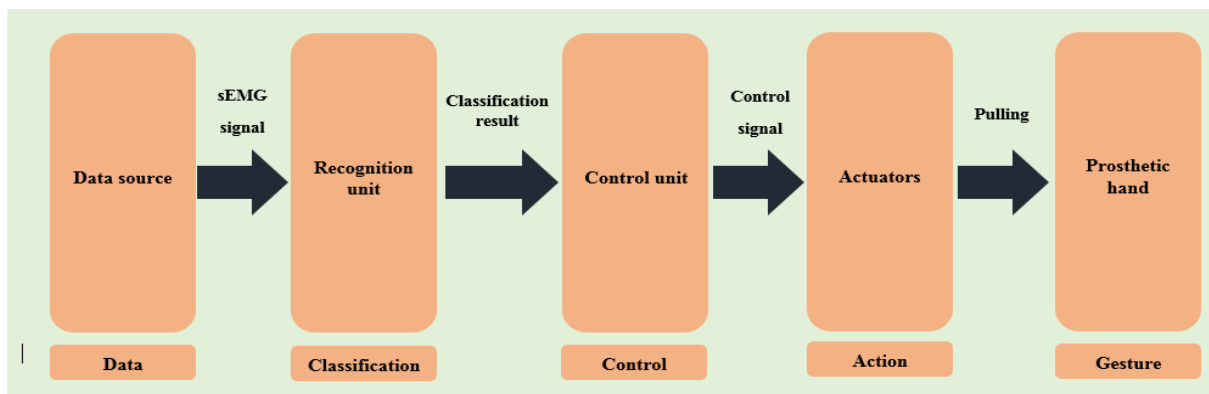


Fig. 8. Hand prosthesis system



Fig. 9. Hand gestures realization

VI. DISCUSSION

When developing a control mechanism for a prosthetic hand based on gesture recognition techniques, two major objectives must be met: achieving high accuracy in the shortest possible time. The test of the complete system represented by the classification algorithm (CNN-LSTM) and the prosthetic hand system showed excellent results, with an overall accuracy of 98.8%, and the ability to classify each movement with approximately equal accuracy. On the other hand, the classification algorithm took 0.08 seconds to classify the 150 ms EMG signal, which is a good value. This value can be improved if more powerful calculation tools are used.

The test of the prosthetic hand system allowed us to compare the movements performed by the natural hand with those of the prosthetic hand. This comparison showed satisfactory results in the latter's ability to embody the desired gestures, although some shortcomings were recorded, mainly due to the structure of the hand which lacks some of the characteristics of the natural hand, making it very difficult to embody certain gestures.

Regarding the time needed for the prosthetic hand to embody these movements, this time varied with the

movement performed, but in general, the average time was equal to 2 seconds. This value is considered reasonable given the characteristics of the electrical and mechanical components used.

VII. CONCLUSION

In this paper, we proposed to develop a hand gesture recognition system using sEMG signals to control a prosthetic hand. This recognition algorithm is based on deep learning techniques that combine the convolutional neural network and the recurrent neural network, which have shown promising results in this type of application. Where the first one is effective in feature extraction and is robust against the noise that overwhelms this type of data. While the LSTM cells can represent features of time dependencies. The developed CNN-LSTM model was tested on the dataset **sEMG for Basic Hand movements Data Set**, which resulted in obtaining a high classification accuracy of 98.8%, this value is one of the highest accuracy values obtained among all researches done using this dataset. This hand gesture recognition model was used subsequently to control a prosthetic hand prototype. Although these satisfactory results, different aspects can be investigated in the future:

- Integrating an sEMG acquisition unit into the system to produce a myoelectric prosthetic hand.

- Improving the model performances either by using the current architecture itself or by developing another architecture while considering the properties of the EMG signal.

- Improving the functional side of the prosthetic hand by increasing the number of degrees of freedom.

Pattern Recognition (icIVPR), (2020), DOI:
10.1109/ICIEVicIVPR48672.2020.9306613.

VIII. REFERENCES

- [1]. Zhang, & Yang, Jaemo & Qian, Jinwu, "Real-Time Surface EMG Pattern Recognition for Hand Gestures Based on an Artificial Neural Network", (2019), DOI: 19. 3170. 10.3390/s19143170.
- [2]. Bakırcıoğlu, Kaan & ÖZKURT, Nalan, "Classification of Emg Signals Using Convolution Neural Network". International Journal of Applied Mathematics Electronics and Computers, (2020), DOI: 10.18100/ijamec.795227.
- [3]. Sapsanis, Christos, Tzes, Anthony & Georgoulas, sEMG for Basic Hand movements. UCI Machine Learning Repository, (2014).
- [4]. 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", IEEE International Conference on Systems, Man, and Cybernetics (SMC), (2017), DOI: 10.1109/SMC.2017.8122854.
- [5]. M. Simão, P. Neto, O. Gibaru, "EMG-based online classification of gestures with recurrent neural networks", Pattern Recognition Letters, (2019), DOI: 10.1016/j.patrec.2019.07.021.
- [6]. Simão, Miguel, Neto, Pedro, & Gibaru, Olivier. UC2018 DualMyo Hand Gesture Dataset (1.0-alpha) [Dataset].Zenodo,(2018),<https://doi.org/10.5281/zenodo.1320922>
- [7]. Xia P, Hu J, Peng Y, "EMG-Based Estimation of Limb Movement Using Deep Learning with Recurrent Convolutional Neural Networks". Artif Organs, (2017), DOI: 10.1111/aor.13004. Epub.
- [8]. N. Nahid, A. Rahman, and M. A. R. Ahad, "Deep Learning-Based Surface EMG Hand Gesture Classification for Low-Cost Myoelectric Prosthetic Hand," 2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision &