Hand Gesture Recognition with sEMG Signals: A Deep Learning Approach

Tomi Kalejaiye*
Cornell Tech
ok93@cornell.edu

Inna Lin*
Cornell Tech
wl676@cornell.edu

Vini Tripathii*
Cornell Tech
ut33@cornell.edu

(* equal contribution)

Abstract

In this study we propose models for hand gesture recognition based on sEMG signal. We compare various datasets for sEMG signals in terms of signal quality and latency. For two datasets we selected, one collected with medical grade sensor and the other with consumer grade sensor, we experiment deep neural net models, CNN and LSTM, as well as traditional machine learning models. We analyze the advantage and disadvantage of each model and their suitability to different datasets. Both our traditional machine learning models and deep neural network models outperformed existing research on selected dataset. We also recognize the remaining challenge in sEMG signal processing and modeling.

1 Introduction

Currently there are nearly 400,000 upper limb amputees in the United States. In thirty years, this number is expected to double. Despite the ever-growing need, the prosthetics available today are insufficient to bridge the gap left by amputation: the unintuitive control mechanism causes many amputees to give up on using their prosthetics. Recent advances in the fields of deep learning and BCI (Brain Computer Interfaces) suggest that an EEG-EMG integrated prosthetic could provide greater and more intuitive control. In this paper we explore the use of deep and traditional algorithms in real-time decoding the six most common grasp movements from surface electromyography readings.[1] Previous studies have shown that sEMG can be effectively used to decode a variety of hand movements (including finger motions), but decoding grasps is considerably harder [1]. In this paper we study the difference between data sets and corresponding models we could use.

2 Related work

There have been a number of papers in using deep neural networks for decoding EMG signals [2]. The general pipeline for this problem is fairly standard. As EMG signals are time varying, the incoming signal is cut into windows. This windowing may sometimes be overlapping, which can be a method for data augmentation [3]. EMG signals are also multichannel, so the windowed portion of an EEG signal can be represented as a two dimensional array. The input is therefore an nxm array where n is the number of samples, and m is the number of channels. Its entries are the values at each channel, for each sample. The output for the neural network when used for classification is a

[1] While we originally intended to investigate decoding motor-imagery of electroencephalography sensor readings, due to the Corona pandemic we were not able to access Cornell University's resources. We attempted to contact other institutions that had access to EEG recordings of hand movement but due to GDPR regulations or institutional rules, we were unable to obtain access. The publicly available data sets only differentiated between left and right hand instead of hand movement, making them inapplicable for our research.

Preprint. Work in progress.

vector representing multiple classes, such as hand gestures or finger movements. Accordingly, in this paper we explored different preprocessing and feature extraction methods in conjunction with deep neural networks and traditional algorithms on four different data sets. This paper gives insight into the relative importance of sample frequency and channel number in decoding EMG data.

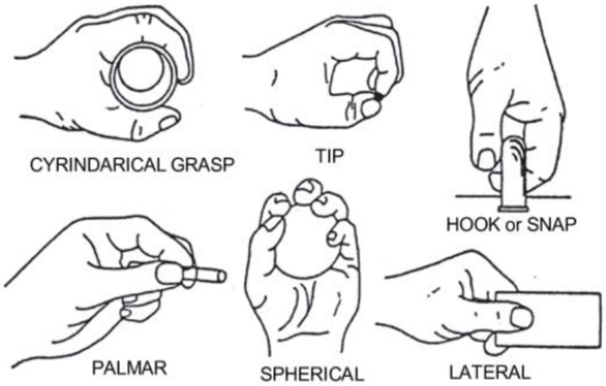
3 Data sets

3.1 Data

Over the course of this experiment we developed and tested a variety of algorithms over four very different data sets: the data set from the paper "EMG Based Control of Individual Fingers of Robotic Hand," our own collected data set and two data sets (Uptras 1 and Uptras 2) from the paper "EMG based classification of basic hand movements based on time frequency features" [4].

3.2 Uptras

The two Uptras data sets [4] are considerably larger than the previous two. Accordingly, most of the deep net were developed and trained on the later two data sets. The Uptras data set consists of two channel sEMG data collected at 500Hz with medical grade equipment. Higher sampling frequencies have been shown to have vital importance in decoding EMG data. Uptras 1 consists of six different subjects performing the six different hand grasps for thirty trials each, with each trial lasting six seconds.



The data is labeled as follows: 1 = spherical, 2 = tip, 3 = palmar, 4 = lateral, 5 = cylindrical, 6 = hook Uptras 2 has a single subject repeating the six grasp for hundred trials for three days, with each trial lasting five seconds. The motions and labels are the same. The Uptras sets had the advantage of size and higher sampling frequency but the disadvantage of less channels.

3.3 Robotic Hand

The "Robotic Hand" data set contains approximately 7000 samples of 8-channel EMG data collected with a commercial grade sensor at a sampling frequency of 200Hz. Processing of raw data involved applying algorithms to detect peaks in the wave and extracting 150 data points in case of a signal to form a sample. There are 10 features extracted per electrode (hence 80 columns): standard deviation, root mean square, minimum, maximum, zero crossings, average amplitude change, amplitude first burst, mean absolute value, waveform length, willison amplitude. The data is labelled (81st column) as follows: 1 = index finger, 2 = middle finger, 3 = ring finger, 4 = little finger, 5 = thumb, 6 = rest and 7 = victor gesture We attempted to replicate and improve the results of this data set but faced several challenges. The biggest problem was missing data: the online repository was missing nearly third of the raw data which had been accidentally deleted by the authors. It was not clear which third had been deleted, so we attempted to map our extracted features to the available featured data with mixed results. When we compared the performance of the three different classifiers on the given feature extracted data and our own extracted features, we found our features outperformed the given features.

	Our Results (Accuracy)	"Robotic Hand" (Accuracy)
Logistic Regression	87.12%	83.84%
Support Vector Machine	71.4%	51.96%
Bagging Classifier-Gradient Boosting Ensemble	94.3%	91.7%

3.4 Self Collected

Given that the results above seemed promising, but technical difficulties made it difficult to proceed, we attempted to collect our own eight channel data set based on the six most common grasps. The sensor was commercial grade with a sampling frequency of 200Hz. We collected thirty trials for grasp movement where each trial lasted five seconds (1000 data points). Unfortunately we were only able to collect data from one subject as the sensor shortly thereafter began to experience streaming issues. The labeling scheme is as follows: 1 = cylindrical 2 = hook, 3 = lateral, 4 = palmar, 5 = spherical, 6 = tip

3.5 Preprocessing

Preprocessing and feature engineering have been proven to play a critical role in EEG and EMG algorithms. There is a wealth of suggested features from existing literature that show high promise. Among the most common are Standard Deviation (STD), Root Mean Square (RMS), Waveform Length and Willison Amplitude. We attempted to recreate the ten features used in the "Robotic Hand" paper: standard deviation, root mean square, minimum, maximum, zero crossings, average amplitude change, amplitude first burst, mean absolute value, waveform length, willison amplitude. Here zero crossings were defined specifically as crossing from - to + instead of a sign change in either direction. The willison amplitude which is defined as the number of data points above a given threshold: the threshold was not specified in the paper so we approximated it through reverse engineering.

After researching other feature extraction methods [5][6][7][8] we increased the number of features to 27, including features extracted in both the time and frequency domain. The final features were: standard deviation, root mean square (this is equivalent to V-order with v=2 which is considered optimal), variance, minimum, maximum, medium, mean absolute value (also called IEMG – integrated electromyogram), waveform length, average amplitude change, the first peak magnitude, number of zero crossings, slope sign change, skewness, kurtosis, absolute value of the 3rd, 4th, and 5th temporal moments, difference absolute standard deviation value, mean frequency, median frequency, frequency of maximum power, mean power, total power, the first, second and third spectral elements, and log detector (which provides an estimate of the muscle contraction). Features were extracted over a sliding window of data.

These features were extracted for both the Uptras dataset and our own collected dataset. Prior to feature extraction, the Uptras dataset was high-pass filtered at 15Hz and notch filtered at 50 Hz to remove noise. Testing with three classification models as baseline (Logistic Regression, Support Vector Machine, Bagging Classifier-Gradient Boosting Ensemble) we found that increasing the number of features always increased the model performance, but the relative increase was dependent on the features in question. We also experimented with an alternative form of feature extraction [9], spectrogram generation. Spectrograms contain key information about change in frequency content over time. Spectrograms were generated from our self-collected data set (8-channel) and the Uptras data sets (2-channel). Spectrograms were generated for each channel individually and concatenated together. The parameters for spectrogram generation for the Uptras dataset was: perseg=75, noverlap=58, nfft=75. The input data to the spectrogram was of length 143 data points with an overlap of 133 data points.

3.6 Initial results from traditional machine learning

Before exploring neural networks based models like CNN and LSTM, we first evaluate the performance of traditional models like SVM and HMM which have previously been found to get good results. This gives us a good baseline for evaluating the performance of our deep learning based methods. Our results are shown in the table below.

Self-collect data set	Accuracy	ROCAUC
Logistic Regression	89.17%	0.9428
Support Vector Machine	16.71%	0.50
SGD Classifier	73.5%	0.841
Bagging Classifier-Gradient Boosting Ensemble	90.46%	0.935
Hidden Markov Model	92%	————

3.7 Online vs. offline

While the goal of this project is to create an online algorithm, it is easier to determine optimal parameter ranges with offline algorithms. To simulate the online performance, all window sizes were required to be less than the average human reaction time (0.25 seconds) with a buffer for computation.

3.8 Metrics

The key metric was accuracy, but for the traditional algorithms we also evaluated ROCAUC score and for the neural network models cross entropy loss plots were also considered.

4 Methods

4.1 Initial CNN

4.1.1 Overview

Our initial proposed model for classifying data from the Uptras dataset was a convolutional neural network (CNN). CNNs have seen great success with not only image classification, but other signal processing tasks as well, such as music genre classification, and speech recognition. The convolutional layers of a CNN are essentially learnable filters that can help the mode remove noise and extract useful information from signals. By taking the short time Fourier transform on windows of our sEMG data, we have created a two-dimensional time and frequency-based representation of our signal, the filtering of which is an ideal task for a CNN. Our initial architecture was based on the fairly successful results of [9] in classifying 8 channel sEMG data. While their data was taken from an 8 channel, higher end measurement apparatus, and their gesture classes were different, we believed the architecture could see some success on our dataset as well.

4.1.2 Architecture

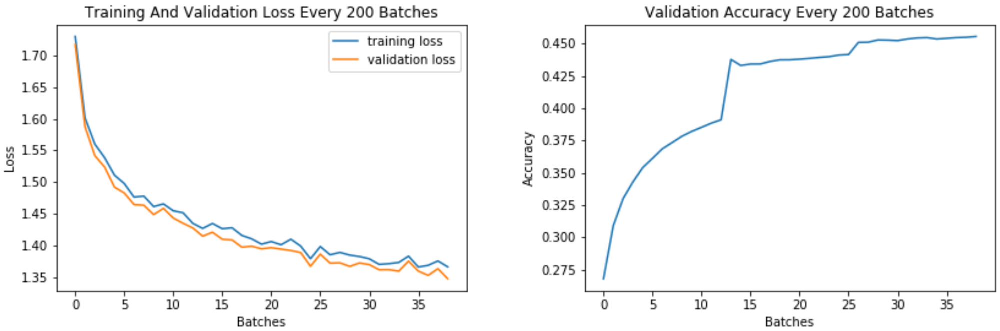
The architecture consisted of three convolutional layers followed by max pool subsampling layers. The convolutions were taken independently across each channel with the same convolution kernel. This seemed an unusual design choice, as generally with convolutional neural networks, channels have unique convolution kernels. However, considering the meaning of the channels in sEMG data, this architecture can be understood. The channels in sEMG data represent different locations of probes, but the data they read out is fundamentally the same. They are not sensitive to different frequencies, but reading from different muscles. Therefore, the convolution and subsampling are done independently on each of the two channels. Then the channels are independently put through two fully connected layers, before finally being recombined by concatenation into one vector. This vector then passes through one more fully connected layer before going to the output layer. We use dropout between all the convolution layers with p=0.5, and between the linear layers with p=0.75. The detailed architecture diagram with the sizes of the dimensions of each layer can be found in [9]. The only difference in our implementation is that since we have a higher sampling frequency, we have more frequencies in our spectrogram, and so more rows in our inputs.

Another interesting aspect of this architecture is that the convolutions are only taken across the rows of the input spectrogram, and not across columns. This is significant as the rows represent frequencies of the spectrogram whereas the columns represent sample windows. The authors of [9] do not explicitly describe their reasoning behind this. However it seems this architecture was designed under the assumption that the significant features classifying the spectrograms appear across features, and not across the time domain.

The tanh activation function used throughout for both the linear and convolution layers, as in [9], that was the activation function used, and it was reported to converge more quickly than ReLU and have similar validation accuracy. We trained the model for 5 epochs with both activation functions and did not notice a significant difference in the loss plots with both.

4.1.3 Training and Results

We trained this model on the Uptras dataset. We used Adam as the gradient descent rule, with a learning rate of 1e-3. This value was found through trial and error. During initial training however, it was found that the loss converged very quickly, (within an epoch). While decreasing the learning rate allowed the model to converge to a slightly lower loss when training, it comparatively made training take considerably longer. We ultimately decided to implement learning rate annealing to prevent the loss from converging too quickly. We used a scheduler to reduce the learning rate by a factor of 0.1 every time the loss plateaued for three epochs. We iteratively experimented with other hyperparameters (batch size, gradient rule) but decided not to change the architecture so we could get a good baseline for the model from [9] on this dataset. Below is a plot showing the training and validation loss of this model.



(a) Training and Validation Loss

(b) Validation Accuracy

Figure 1: Training and Validation Results (Initial CNN Architecture)

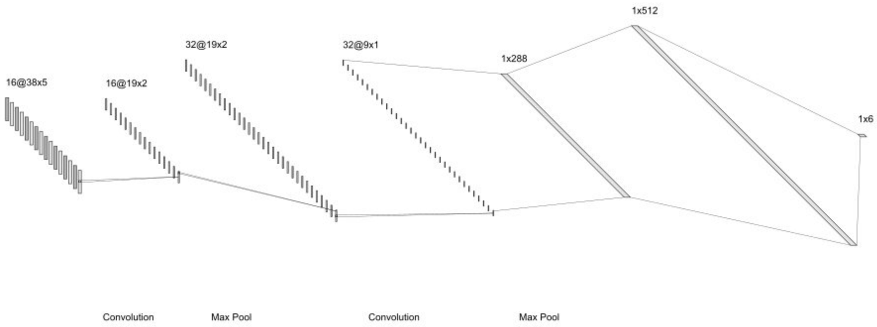
The final validation accuracy of this model was 46%. This is significantly below the accuracy reported in [9]. There are a few differences between this dataset and the dataset that the model in [9] was trained on, which might help explain its poor performance here. One significant is the difference in the number of channels. In sEMG data, each channel corresponds to a different sEMG probe location on the arm, and another muscle. As most movements don't have a significant correspondence to one muscle in their sEMG readings, a greater number of channels can help significantly in classifying movements. Another significant difference is that we are classifying a different set of gestures than in those [9]. The gestures in [9] seem to be requiring more significant flexion of the forearm muscles than our gestures do. It may be that the sEMG signals correlating to our gestures are more subtle than theirs.

4.2 Alternative CNN

We were interested in seeing if we could implement our own CNN that would have improved performance on this dataset, so we decided to design another CNN architecture and evaluate it.

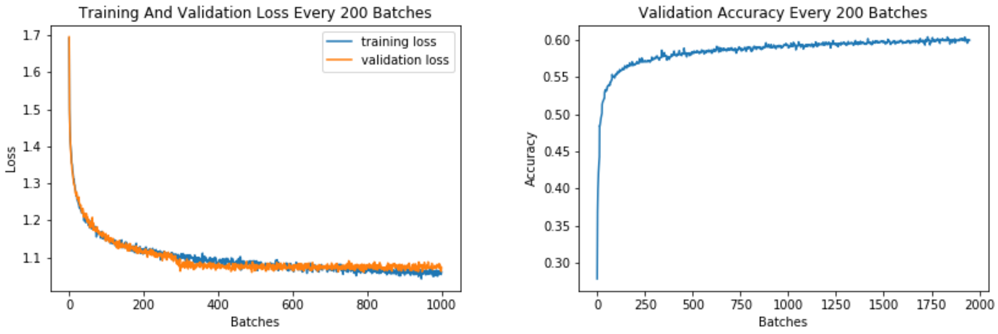
4.2.1 Architecture

Our second CNN architecture consisted of two convolution layers, however unlike the convolution kernels in the first architecture, we used more standard 3x3 convolution kernel here. It was not obvious to us that there was an advantage in only convolving across frequencies, and considered that we might be missing some usefuly correlations by not doing convolutions across the sample domain as well. We also used max pooling sub sampling layers like the last architecture, and droupout layers, finally ending with two fully connected layers here instead of 4. We used ReLU as an activation function.



4.2.2 Training Results

This model was trained also trained on the Uptras dataset, with the same hyperparameters as the previous CNN. Its training and validation loss can be seen in the figure below.



(a) Training and Validation Loss

(b) Validation Accuracy

Figure 2: Training and Validation Results (Our CNN Architecture)

The final validation accuracy of this model was 60%. While still low in comparison to the performance achieved in [9], it is a significant improvement over the previous architecture.

4.3 LSTM

4.3.1 Overview

Since the signals for each hand movement are sequencial, we utilize the Long Short-Term Memory (LSTM) Model given its hidden state structures. Besides its ability to improve the vanishing/exploding gradient problem, we can also explore the dependency of longer term memories in sEMG data. We analyzed the resulting accuracy and compared with other methods.

4.3.2 Architecture

We implemented a LSTM network on both the 2-channel and the 8-channel dataset, with sequence classification as the learning objective. We use feature extraction technique described above to extract a feature vector for each snap shot in any given sequence. For the 2-channel dataset, our best performing architecture includes a LSTM layer with 180 hidden dimension, followed by a fully connected layer that reduce the dimension down to 6, preparing for the classification task. For the 8-channel dataset, our best performing architecture includes a LSTM layer with 50 hidden dimension, followed by a fully connected layer. We have use cross entrypy loss in the training process.

For both datasets, we also apply a dropout layer after the fully connected linear layer, we experimented the dropout ratio and have found that 0.2 is the best dropout ratio for both the 8-channel and 2-channel datasets.

4.3.3 Training

In the training process, we have experimented several parameter combinations for each dataset. For the 2-channel dataset, we experimented training epochs of [100, 150, 225, 250, 300] and have found that 225 epochs generated converged results that are also as stable as later epochs. For learning rate, we experimented [1e-5, 1e-4, 1e-3, 3e-3, 5e-3, 1e-2] and have found a learning rate of 3e-3 with linear steps each 150 epochs to be the best performing. We also experimented step size for learning rate in range [no step, 100, 200, 300] and have found 100 works the best for this dataset.

For the 8-channel dataset, we experimented training epochs of [200, 300, 400, 500, 600] and have found that 500 epochs generated converged results that are also as stable as later epochs. For learning rate, we experimented [1e-5, 1e-4, 1e-3, 3e-3] and have found a learning rate of 1e-4 with linear steps each 300 epochs to be the best performing. We also experimented step size for learning rate in range [no step, 100, 200, 300] and have found 300 works the best for this dataset.

4.3.4 Results

We used LSTM on both the 2-channel and 8-channel datasets. As we described earlier, the 2-channel dataset has 54 features per snapshot but a longer length of 236 per sequence. We used 20% data as the validation data and the accuracy we achieved was 72.2%. In comparison, the 8-channel dataset has 128 features per snapshot and a shorter length of 165 per sequence. During our experiment, we found out that using higher hidden dimension generates better results for model that has longer sequence.

	Model Params	Validation Accuracy
2-channel	lr=3e-3, ep=225, hidden=180	72.2%
8-channel	lr=1e-4, ep=500, hidden=50	82.5%

We plotted the validation loss and accuracy per 5 epochs (per 10 epochs for valuation accuracy for 2-channel) as shown below. The movement recognition task is sensitive to small changes in the data points, therefore the loss and accuracy can be volatile, we are taking the average of each 20 epochs as our validation accuracy for reporting.

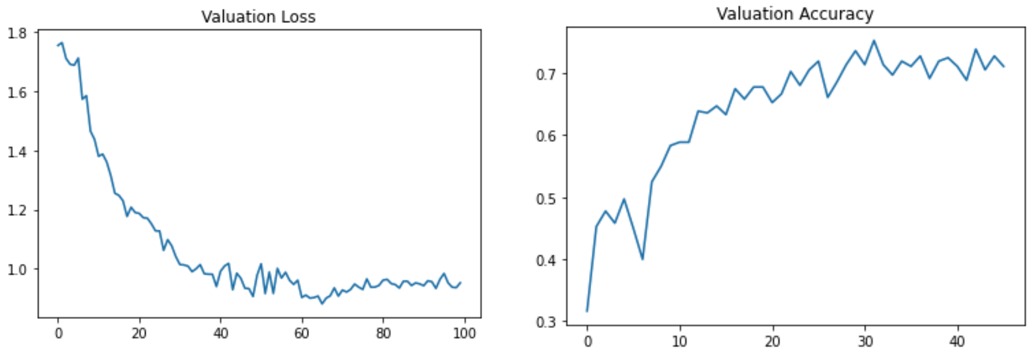
5 Analysis

5.1 LSTM Model

The LSTM model was able to capture longer-term memories of the brainwave signal and it has been proved by the out-performance compared to CNN. From our experiment we also recognize that a higher dimensional feature set improve the performance of LSTM. This is likely due to two reasons. First, the feature extract process was able to catch more sEMG signal from multiple channels. Second, a larger feature set, including the addition of more features and the inclusion of more channels, is able to feed each LSTM block more information and to ameliorate the volatility in each single channel.

5.2 LSTM and CNN

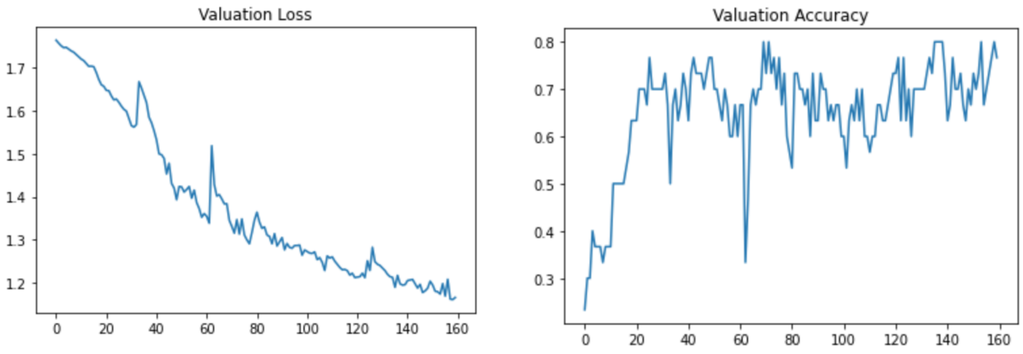
We compared LSTM and CNN model in terms of their difference in architecture, which leads to different level of suitability for sEMG data. CNN is able to perform on raw spectrogram data without



(a) Validation Loss

(b) Validation Accuracy

Figure 3: Training and Validation Results (LSTM, 2-channel)



(a) Validation Loss

(b) Validation Accuracy

Figure 4: Training and Validation Results (LSTM, 8-channel)

further processing, which is more efficient. CNN is also more suitable for smaller data set compared to LSTM, since CNN have more data points that can be used in each sequence, whereas LSTM takes in one sequence as an input. From the experiments we have conducted, LSTM outperforms CNN in bothe 2-channel and 8-channel datasets. This is likely due to two reasons. First, the sEMG data has rich memory that makes LSTM a more suitable model. Second, the feature extraction process in LSTM performs better than the feature learning process by CNN.

6 Conclusion

In this study, we have explored various available sEMG data sets, both from medical grade sensor and from consumer sensor. We have compared the use of traditional machine learning methods and deep neural networks, specifically CNN and LSTM. We have found different method outperforms on different data set and have proposed outperforming models given the characteristics of the data set. We also recognized the challenge in analyzing sEMG data from both signal collection and modeling perspective.

7 Future Work

Future work can be done in the following areas. i) data collection: more data points from each individual with 8-channel data. ii) Further improvement of the model performance by applying fused CNN-LSTM model to take advantage of both architectures.

References

[1] Tsagkas, N., Tsinganos, P., Skodras, A. (2019, July). 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) (pp. 1-4). IEEE.

[2] Naseer, N., Ali, F., Ahmed, S., Iftikhar, S., Khan, R. A., Nazeer, H. (2018, November). EMG Based Control of Individual Fingers of Robotic Hand. In 2018 International Conference on Sustainable Information Engineering and Technology (SIET) (pp. 6-9). IEEE.

[3] Roy, Y., Banville, H., Albuquerque, I., Gramfort, A., Falk, T. H., Faubert, J. (2019). Deep learning-based electroencephalography analysis: a systematic review. Journal of neural engineering, 16(5), 051001.

[4]Sapsanis, C., Georgoulas, G., Tzes, A., Lymberopoulos, D. (2013, July). Improving EMG based classification of basic hand movements using EMD. In 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 5754-5757). IEEE.

[5]Parajuli, N., Sreenivasan, N., Bifulco, P., Cesarelli, M., Savino, S., Niola, V., ... Gargiulo, G. D. (2019). Real-Time EMG Based Pattern Recognition Control for Hand Prostheses: A Review on Existing Methods, Challenges and Future Implementation. Sensors, 19(20), 4596.

[6]Bhattachargee, C. K., Sikder, N., Hasan, M. T., Nahid, A. A. (2019, July). Finger Movement Classification Based on Statistical and Frequency Features Extracted from Surface EMG Signals. In 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2) (pp. 1-4). IEEE

[7]Tkach, D., Huang, H., Kuiken, T. A. (2010). Study of stability of time-domain features for electromyographic pattern recognition. Journal of neuroengineering and rehabilitation, 7(1), 21.

[8] Phinyomark, A., Phukpattaranont, P., Limsakul, C. (2012). Feature reduction and selection for EMG signal classification. Expert systems with applications, 39(8), 7420-7431.

[9] Allard, U. C., Nougarou, F., Fall, C. L., Giguère, P., Gosselin, C., Laviolette, F., Gosselin, B. (2016, October). A convolutional neural network for robotic arm guidance using semg based frequency-features. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2464-2470). IEEE.