

Artificial Hand Simulation Using Artificial Intelligence and Machine Learning

FARHAN FUAD RUPOM¹, GAZI MUSA AL JOHAN², SHAFAITUL JANNAT³ and FARZANA FERDOUSI TAMANNA.⁴

¹Department of Computer Science and Engineering,Brac University,Mohakhali,Dhaka (e-mail: rupom.cse16@gmail.com)

²Department of Computer Science and Engineering,Brac University,Mohakhali,Dhaka(e-mail: johandipto19@gmail.com)

³Department of Computer Science and Engineering,Brac University,Mohakhali,Dhaka (e-mail: shipa96jannat@gmail.com)

⁴Department of Computer Science and Engineering,Brac University,Mohakhali,Dhaka (e-mail: tamannamou76@gmail.com)

ABSTRACT Electromyography is a unique approach for recording and analyzing the electrical activity generated by muscles, and a Myo-electric controlled prosthetic limb is an outwardly controlled artificial limb which is controlled by the electrical signals instinctively produced by the muscle system itself. Artificial Intelligence and Machine learning is very powerful in every technological field along with biomedical field. The purpose of this work is to utilize the power of Machine learning and Deep learning for predicting and recognizing hand gestures for prosthetic hand from collecting data of muscle activities. Although this technology already exists in the technological world but those are very costly and not available in developing countries. So, designing a cost effective prosthetic hand with the maximize accuracy is the major focus and objective of this work. We have used a data set recorded by MyoWare Muscle Sensor which represents uninterrupted readings from 8 sensors. We have used Deep learning with the data set for predicting the following gestures which are hand-open, hand-close, spherical grip, and fine pinch .Then we used classic machine learning algorithms which are K-nearest Neighbor (KNN), Support Vector Machine (SVM), and the combination of KNN and SVM both for feature extraction on data recorded with the 8-electrode surface EMG (sEMG) MyoWare Muscle Sensor. Using the combination of SVM and KNN We have accomplished a real time test accuracy of 96.33 percent at classifying the four gestures of our prosthetic hand. This paper also presents 3D modeling of the robotic hand and its control system using Autodesk 3D's Max software, EMG MyoWare Muscle Sensor, Machine Learning and Deep Learning.

INDEX TERMS Electromyography, Hand gestures, K-nearest Neighbor(KNN), Support Vector Machine(SVM), EMG MyoWare Muscle Sensor, Autodesk 3ds Max software, Prosthetic

I. INTRODUCTION

Due to disease, accidents, and congenital defects people from worldwide are losing their body parts daily. As the road and industrial accidents are increasing in developing countries like Bangladesh, the percentage of disable people is increasing. In a statistics review it has been known that, In the USA, there are already 2.1 million people living without their limb which they have lost in many accidents, unexpected incidents, diseases and that number is expected to double by 2050 [23]. Due to limb loss a person's life become more struggling because of facing emotional and financial lifestyle changes. But the good news is being a paralyzed or disable does not mean living without independence in this era when technological world that we are watching today where amputation or any disability can be turned into power for those persons. Therefore, the challenge of this work is to make

research and study the different possible ways of doing using artificial intelligence with prosthetic hands to improve the lives of amputees. But The costs of commercially available myoelectric hands are very high, ranging in price from 2 to 30 Lac [10]. As a result, amputee people from developing countries like Bangladesh are not able to effort this costly bionic robotic prosthesis. So our goal is to build a model in a cost effective way so that poor people can afford our model to make less their struggles in their daily life. The type of prosthesis that replaces a missing body parts such as arms, legs, hands are called an artificial limb. There are four main types of artificial limbs. They are, Transradial prosthesis, Transhumeral prosthesis,Transtibial prosthesis, Transfemoral prosthesis [20]. We have mainly focused on the transradial prosthesis as it is an artificial limb that replaces an arm missing below the elbow which is called upper limb prosthesis.

While doing this work, the challenge we have taken is to do unlimited research and studies for finding out the different and best possible ways of using Machine Learning and Deep Learning with EMG based prosthetic hands to do make betterment of the amputees. We selected the unique approach electromyogram (EMG) as it is one of the sources that can be used as a control method for a prosthetic hand and simple to use. Surface EMGs passes information about the muscular activity of for using those readings as a source of controlling process for prosthetic limbs [17]. As our prosthetic hand is assisted with a sensor system it is 3D printable, affordable, easy and not unpleasant to use. The given below ideas are based on our contributions:

- Firstly, we have used a dataset recorded by MyoWare Muscle Sensor which represents uninterrupted readings from 8 sensors.
- Secondly, for feature extraction, Furthermore, mean absolute value, Root mean square, Slope Sign Changes, Zero Crossings and other techniques are used.
- Thirdly, for feature classification SVM, KNN, and combination of KNN and SVM algorithms are used but the combination of SVM and KNN has the highest accuracy rate among other two algorithms for classifying the four different hand gestures which are hand-open, hand-close, spherical grip, and fine pinch.
- Finally, after identifying different gestures, we have used the classifiers to collect and transfer information and to control our prosthetic arm. We have also used the 3D modeling and rendering software name Autodesk 3D max for implementing our system the software simulation system.

The rest of the paper has been described in the following manner. Section discusses why we need bionic arm. Research objective is presented in section III and the section IV talks about the literature review. The whole system architecture is discussed at the section V and The EMG signal and electrodes placement, characteristics and nature of the signal described in the Section VI. System methodology is talked about in the section VII along with the algorithms and kernel which are proposed. Signal processing with different methods along with feature extraction from signal presented at Section VIII. In Section IX, described about the data-set acquisition and characteristics. System implementation with algorithms, feature selections and necessary calculations are discussed in the section X. User interfacing, visual representation and analysis of system output are described in Simulation and Analysis which is in Section XI. and Section XII shows the variability of classification rate, error percentage and accuracy rate result. Ultimately Section XIII concludes our paper.

II. PROBLEM STATEMENT

As with the pace of modern science, the biomedical systems and devices have also got advanced in technology and features, but the sector hasn't gone far enough to provide the instruments which can perform precise data classification detection and act accordingly with low cost production.

Even in this era, many people are moving around without hand or legs or other limbs where the problem should have been solved by technology. Many people losing their limbs every day through accident or medical complexity around the globe. Annually approximately 1, 85,000 individuals are decapitated [1]. It tends to result in 300 to 500 surgeries daily [1]. Insurgents approximately 1,558 individuals lost their body parts in the wars in Afghanistan and Iraq [1]. Approximately 30 percent of those with limb amputation had bipolar disorder [1]. Lifetime cost of medical care for the individuals with limb loss is \$509.275 in contrast to \$361 million for the individuals without any body part loss [1]. In 2013, clinic expenses totaled \$8.7 billion for clients' amputated [1]. Up to 55 percent of people carrying diabetes who already have lower limb amputation would undergo amputation within two or three years of the second leg [1]. Sometime within Five years, approximately half of people with surgical removal induced by cardiovascular problems would die [1]. The death rates for mammograms, pancreatic cancer and testicular cancer are significantly greater than five years [1]. Around 1988 and 2009, there were 24 percent more lacerations induced by diabetic [1]. Domestically and internationally, over one million limbs are surgically removed within every 30 seconds worldwide [1]. So as it can be realized the numerous people around the globe are in need of prosthetic limbs with low cost. Again prosthetic limbs with slightly lesser production cost cannot maintain the quality of performing gestures according to users' gestures, as the data classification and acting according to classification is not that much precise. So it's kind of an one way deal, cost can be reduced but precision of performance will be affected or the quality precision of acting according to users data can be increased but with sacrificing the option of producing it with low cost. As a result, most of the prosthetic limbs are out of the affordable price for most of the general people.

III. RESEARCH OBJECTIVE

Our research objective is to create a system where the data and information of user will be acquired and classified precisely as well as to ensure the quality of performing or movement of the prosthetic limb analyzing the data classification as precisely as possible. Most of the cost effective prosthetic limbs can't maintain the synchronization between users' data or signal acquired through EMG sensor and can't act precisely how the user wants the prosthetic hand to be performed. On the other side, which prosthetic arm or limbs do have these kind of quality in synchronization and performance, are not cost efficient or affordable for financially back lagged people. As a result, our purpose is to develop a system where the production cost will be minimized but the data classification and performance with synchronization will be maintained as far as possible. Our data-set for training will be larger and for that, the decision making system of our proposed model will be able to decide suitable action according users' new data values. We aim to implement machine learning and artificial intelligence though using

Support Vector Machine algorithm (SVM) and K-nearest neighbor algorithm (KNN) to classify the data and train the data set accordingly. Moreover, we are going to implement the software system through simulation or hardware to rectify any kind of bugs and inconvenience. According to that, the user can almost move his prosthetic hands like a real bio hand which will be produced within low cost and greater accuracy.

IV. LITERATURE REVIEW

Johnny L. G. Nielsen et al. [20] proposed a novel approach done with surface electromyogram to record information from one upper limb to force generated using the collateral limb. They measured from the right wrist in multiple degrees of freedom along with different movements. They recorded along with isometric forces in several degrees of freedom from the right wrist.

J. Carpaneto et al. [5] proposed using the EMG signal for a physical prosthetic limb. EMG-based stimulation is done by machine learning in the nervous system and the use of implantation of electrodes in the muscles. The vector machine algorithm is being used to predict different types of grip including grasp gestures using both distal and proximal upper limb tissues with EMG activity.

Katsutoshi Kurabayashi et al. [15] proposed using neural networks to monitor the electromyography and form memory material using two methods. Firstly one is EMG signal rectification and integration and the one is the cycle of the neural network. Because of light weight and compact, SMA is used compared to the EMG signal. In [5] Researchers record myoelectric response from two adult macaque monkeys recording 12 items of various shapes to distinguish among types of EMG stimulation associated with particular grasping behaviors. Netta Gurari et al. [19] recommended the use of positive guidance for foot, arm and fingers. It allows people to connect with the world. Using a manipulation method that simulates an optimal robotic prosthetic, able individuals tapped on surfaces of different hardness by using a gyroscope vibration signal were recorded. At the foot of both the tip such vibrating input works better than in the upper limb. R. Brent Gillespie et al. [10] suggested a process in using a novel form of tactile stimulation including concepts in novel control. Here the electric elbow support is being used in the form with attachment torques across the arm to feedback grasping movements. The EMG stimulation is derived from both the movements of the forearms. This will operate well with and without pressure control in several different situations, employing reticular formation and reticular formation biceps myoelectric indications to represent power, and us to accurately identify artifacts. Via sensory feedback, it significantly increasing. It is applied in 7 individuals who are able-bodied and also the outcome is effective.

Ryait et al. [26] suggested innovations throughout the field of prosthetic limbs. This paper contains three sections, the first one to illustrate EMG signals effects, the next to examine EMG signal on computer simulations, and even the third to describe various virtual hand models using EMG signals.

Ryait et al. [26] suggested innovations throughout the field of prosthetic limbs.

V. SYSTEM ARCHITECTURE

Based on the embedded micro-processor structure, the system depends on reasonable and high speed micro-controller is adequate equating for signal processing and it is convenient for the kind of project needs to develop a high-proficiency, responsiveness, and suitable controller. The block diagram demonstrate in Fig 1 a versatile controller based on reasonable elements can be used as a “Do it yourself” kit with the capacity to serve personalized hand postures which suits the lifestyle of the users.

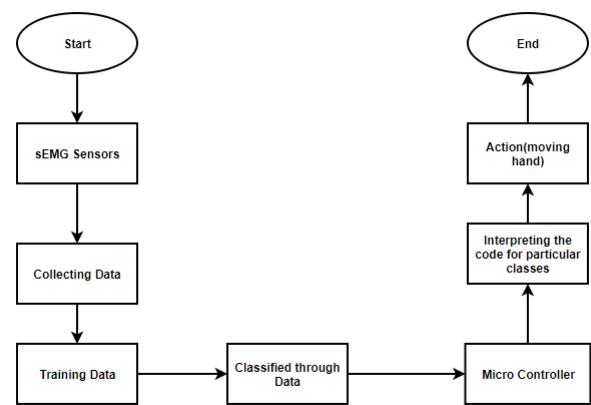


FIGURE 1: The system block diagram showing the architecture

A. PHYSICAL ARCHITECTURE

The physical architecture is the representation and layout of a project and its modules in a schema. This architecture indicates the layout of the structure or arrangement of the physical modules of a system. One of its aims to compliment the logical architecture part and project requirements. In this system the module that represents the physical architectures are EMG sensor for taking input; microprocessor, filtering and simulation processor for processing and simulation preview for increasing the accuracy.

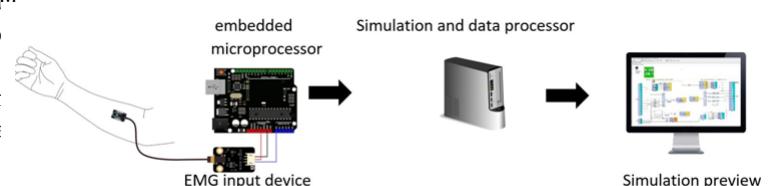


FIGURE 2: Physical Architecture

B. LOGICAL ARCHITECTURE

In order to represent the logical machine view, the MVC architecture is used all over [17]. By using this architecture, one can represent distinction among the presentation, the data and the processing. MVC gives three rudimentary fragments whose parts are illustrated :

- Model: Includes all the data for displaying [17].
- View: Encompasses presentation of data to user and interface [17].
- Controller: Controls user inputs and encompasses them to process and sends command for updating model [17].

For interactive application, this kind of architecture is perfect in order to fulfill the requirements [17].

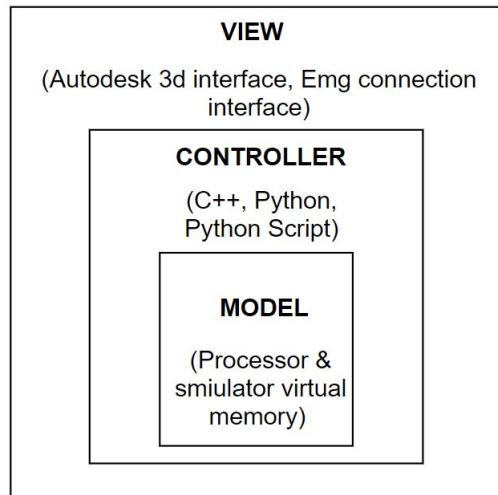


FIGURE 3: Logical Architecture

VI. DATA UNDERSTANDING

A. SEMG SIGNAL DETECTION

An accurate exposure of individual action in sEMG is an essential topic in the research of the hand's motor system, a single-threshold system is used to expose muscle timing on and off, analyzing the Root Mean Square (RMS) value of the reformed signals to thresholds whose result is based on the mean power of the framework of noise of each channel [8]. After calculating the offset from the bias voltage reference has been removed, In order to finally determine the RMS value, 100 samples are reformed to identify the expected manual intervention used to trigger the FSM transformations regulating each and every function.

B. ELECTROMYOGRAPHY AND SIGNALS

A motion starts as a signal from our brain whenever we move one of our skeletal muscles. Then The signal reaches from the spinal cord in the brain to the respective muscle's main motor neuron which is shown in Fig 4. It creates an electrical signal down the length of the muscle after activation of the motor neuron by the brain and that signal creates a polarization and depolarization known in the muscle

and known as the potential for action. That action potential expands to the muscle for activating the muscle's motor neuron cells, and the signal activates all the sarcomeres along the muscle's strands and the muscles lease. So the implication of a session on Electromyography is to record that action potential. Electromyography can be used in a various way. The most important use of Electromyography is medical diagnostic. Medical diagnostic let us know about the type of signals the muscles exhibit in case of abnormal muscle activities. Another use of EMG and controlling prosthetic arm is Rehabilitation which is the purpose of this paper and also application of this. This is also used in Bio-mechanical researches for understanding our brain controls over our muscles and this related studies [3]. Mainly two types of elec-

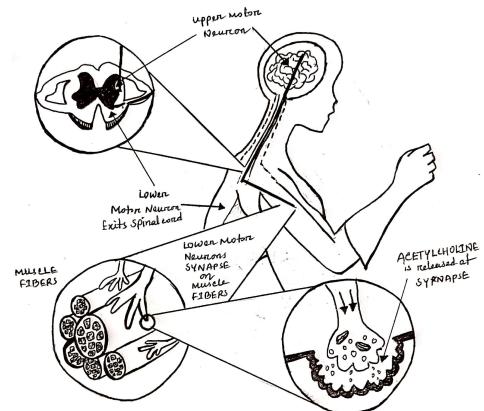


FIGURE 4: Skeleton Muscle Function

trodes are usually used for Electromyography, surface and intramuscular. Surface electrodes at Fig 5 are very common among other methods such as EKG's and EEG's and also used in this work. As they are used just by sticking to the skin, they are very simple to use. For the readings of Signals (SEMG) through skin, they have a metal conductor in the middle with some electric jelly on the other side. They don't cost the patient much discomfort so they are convenient to use. The use of this type of electrodes is as simple as putting them on and taking them off after collecting the readings. They do not need any special training, either. They can't take values from the muscles which are too far below the muscles or too far from the skin because the readings are limited to muscles that are very close to the surface of the skin. They will also take few signals from adjacent muscles. On the other hand, Intramuscular electrodes at Fig 6 cause discomfort for the patient as they have to be inserted directly into the muscle. This counts as a disadvantage of this type of electrodes. Skin conditions of some people can be swollen by getting a needle stuck in the muscle very significantly. But the advantage of this type of electrodes is they can measure values from any types of muscles as well they also don't get any signals from adjacent muscles. The suggested sEMG signal frequency

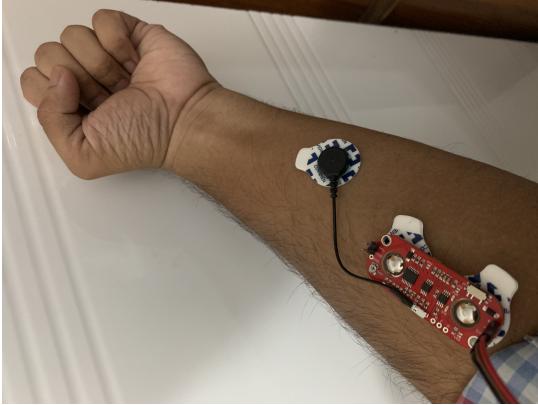


FIGURE 5: Surface electrodes

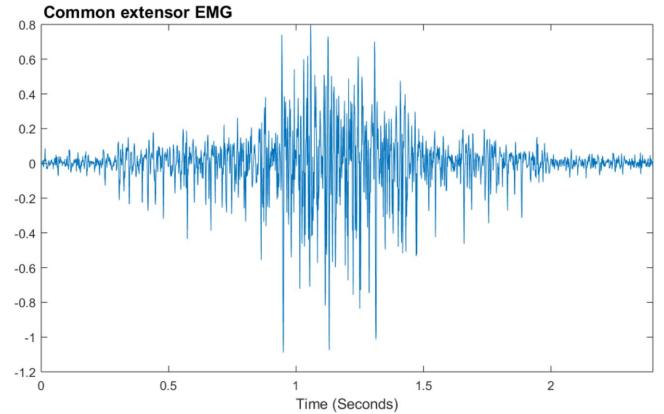


FIGURE 7: Nature of Raw EMG Sign

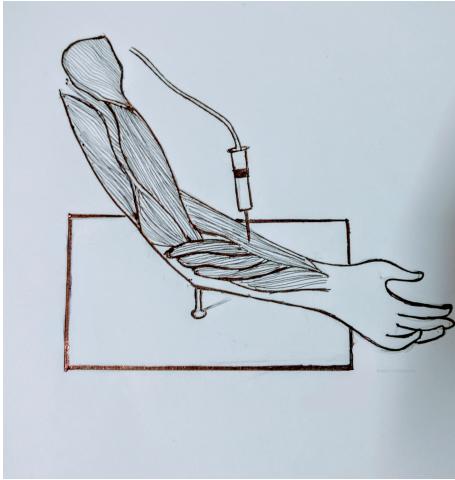


FIGURE 6: Intramuscular electrode

range is 5-500Hz which requires a sampling frequency more than or equal to 1000Hz [18]. The MyoWare Muscle Sensor is limited to 200Hz. For processing the collected signals accurately robust and adequate techniques are necessary. Prior to amplification, the amplitude range of the EMG signal is 0 mV-10 mV (± 5 mV) [25]. The raw EMG data contains external, removable noise. The EMG signal is bipolar, which indicates it extends in both positive and negative directions and focuses around zero shown in Fig 7.

VII. SYSTEM METHODOLOGY

In our proposed model, we will first collect real data through EMG signal. Then we will do pre-processing and feature extraction of those signals using different methods. After that, we will classify our data set for getting the most accurate value for motion recognizing. Finally, we will feed those values to our simulated model and will test the accuracy of our model.

A. CLASSIC MACHINE LEARNING

I SVM : A binary classifier, which determines whether or not a sample is in one class, can be described as SVMs.

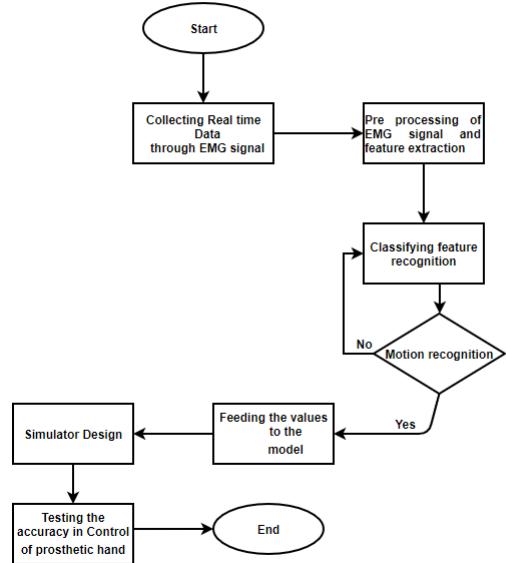


FIGURE 8: Flowchart of Bionic Robotic arm

It employs the principle of optimizing structural hazards which compromise the methodological pitfalls and the model's intricacy [24]. The construction of a decision surface is an integral part of an SVM, so as to maximize the distance from positive to negative examples. The judgment surface has a high level, since Rn components are the input vectors. For linearly separable data, it can be distinguished with an optimal separator line [24]. For linearly separable data, it can be distinguished with an optimal separator line [24]. For this the optimal hyper plane for classification can be described by [24],

$$w^T x_i + b = 0 \quad (1)$$

where w is a vector of weight and x a vector of input. The ideal hyper plane with two-dimensional linearly separable information shows an example in Fig 9. A particular set of input vectors is defined for this hyper

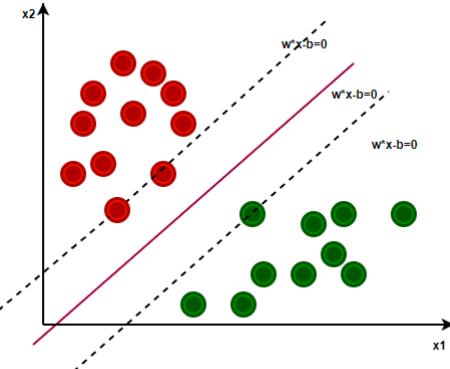


FIGURE 9: Hyper plane with distinguishable data (linear model)

plane, known as supporting vectors. This is the ideal hyper plane. Several input vectors are positioned adjacent to the optimum hyper plane. Categorization is then rendered. This is the ideal hyper plane [11]. According to the following terms for any fresh specimens' x_i [11],

$$w^T x_i + b \geq 0 \quad y_i = y_i + 1 \quad (2)$$

$$w^T x_i + b < 0 \quad y_i = y_i - 1 \quad (3)$$

Usually Support vector machine (SVM) is a straight forward as well as highly innate concept. In spite of that, when the vector spaces are inseparable, difficulties arise. Problem arises when the data point falls into the space between the ideal hyper planes and in the moment of being the data co-ordinates plotted across hyper planes. A variable i is used to calculate support vector machine calculation that is responsible for changes from the ideal position of the data point. The following optimization problem can therefore summarize the general form of an SVM briefly [6]. Over a range of marked $(x_i, y_i), i = 1 \dots l$ pairs $x_i \in R_n$ and y a SVM the solution is described to [6]:

$$\min_{w, b, \epsilon} \frac{1}{2} w^T W + C \sum_{i=1}^l \epsilon_i \quad (4)$$

C is a penalty parameter of the error that regulates the balance between the complexity of an SVM and the number of not separable points. The optimization problem includes the C function, which is a parameter that is user-defined. The user can control the ability of the SVM to generalize using this parameter. This issue is solved by means of quadratic optimization. Since the separation of data is an issue, SVM architecture requires an additional element. SVM's use a nonlinear kernel function which transforms the defined vector into a larger dimension input space of a vector. In this way, the chance of the input vectors being further separated is increased. The operation of a SVM is quite simple

after the optimal hyper plane has been built. The vector for input is mapped to the functional area. The SVM then defines the location and judgment of the function in the hyper plane. SVMs' capacity to generalize is their exclusive benefit. With the inclusion of the vector discrepancy in its optimization algorithm as a variable, SVM's can compensate for some input vector heterogeneity which further makes them a good candidate with which to classify EMG.

II SVM kernel: SVM incorporates a kernel to map the input data to a function space. Three kernels are typically used.

There are no strategies to choose a kernel feature to use for the SVM or identify the functionality of the kernel for a specific application. The RBF feature should be first preference, according to [6]. One explanation is that RBF kernel that function like sigmoid and linear kernels with certain parameters C and γ . A further justification to use the RBF kernel is that numerical computations are less daunting. The RBF kernel throughput fluctuates from 0 to 1, with a polynomial kernel vaguely resembling infinity or 0.

III KNN: K-Nearest Neighbor (KNN) plays a significant role in pattern recognition. This method identifies the homogeneous things that are close to each other. It is mainly used to solve regression and classification problems which need predictions. For ease of interpretation and low calculation time the KNN method gives a very good assumption. KNN classification is used to perform statistical analysis using a discriminant function when stable parametric estimation is quite unknown or determining is tough. This algorithm uses the Euclidean distance between two information (in n-dimensional space two of the given points) is defined by the given formula.

$$dist(x_1, x_2) = \sqrt{\left(\frac{n}{i=1}(x_1 - x_2)^2\right)} \quad (5)$$

Here, x_1 and x_2 are two information [4]. It measures the distance between x_1 and x_2 corresponding to the n attributes [4]. This Euclidean distance is used to get the closeness of the objects in the algorithm. The steps which are used in KNN algorithm are [4]:

- a) Determination of the parameters for K
- b) Sorting the distance and finding out the nearest neighbors
- c) Nearest neighbors category Y is gathered.
- d) Nearest neighbors simple majority is used

As our proposed model deals with human motion data we have used this method to get a better result. The electromyography (EMG) signals influences the artificial hand's grabbing and hand grasp opening which is achieved by multiple parameters of different movements of test subjects using K-NN rule [4]. It's possible to use overlap metric or Hamming distance. The test sample which are of green circle those should be categorized

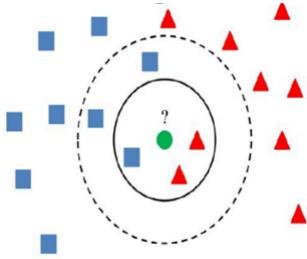


FIGURE 10: K-nearest Neighbor

either as first class blue squares or as red triangles of second class. If the value of k is 3 so there are 2 triangles and only 1 square inside the inner circle, then it is classified to the second class [13]. And if the value of k is 5 and there are 2 triangles versus 3 squares inside the outer circle then it is classified to the first class [13]. The algorithm's training phase consists primarily of collecting the training samples feature vectors and also class labels. The test sample with unknown class is represented as a vector in the feature space in the actual classification phase. Intervals from the new vector to all collected vectors are calculated and samples closest to k are chosen [13]. There are so many techniques for classifying the new vector to a particular class but one of the most used techniques is to find the prediction of the new vector to the most common class amongst the K nearest neighbors [13]. A primary drawback of using this approach to classify a new vector into a class is that the classes with the more common examples intend to control the prediction of the new vector, as they tend to appear in the nearest K neighbors when the neighbors are calculated based on their huge number [13]. But this issue can be solved by taking into account the distance of each K nearest neighbors along with new vector which needs to be classified and to predict the class of the new vector based on these distance values [13].

VIII. DATA PREPARATION

A. THEORETICAL SIGNAL PROCESSING

In this section, the different commonly used signal processing techniques for EMG signals are explained. Like,

I Noise Filtering: The purpose is therefore to eliminate any unwanted noise which contaminates the transmission. This seems to be critical because it will effect the data that we evaluate if it has not been eliminated. The majority of EMG transmissions range from 20Hz to 350Hz [17]. That indicates that a pulse with a wavelength less than 20Hz or greater than 350Hz originates from another point than a muscle [22]. The noise can come from either the electrodes activity, or a particular high frequency transmission from radio frequency propagation and mobile phones [22].

II Rectification : Every EMG value would be optimistic after rectification, since all the negative values below the

x-axis are folded on the optimistic side after rectified which is defined as full-wave rectification [17]. Another method of rectification is half-wave rectification, whereby the negative values are set to zero rather than being transformed into an absolute value [17].

III Normalization: Normalization means transferring a transmission to a level which is related to a certain amount of values. Because each and every electrode plays a part in the output of machine learning, the range of feature values should be standard [17]. Relying on the deep learning technique used, when the importance of any of the variables varies significantly, this feature will overpower through the remainder and adversely affect the output of the program [17]. That is why these functions are standardized such that all the features can make a nearly equal contribution through training [17]. The easiest way to do this is to update the set of functions in the min-max algorithm to new ranges from 0 and 1 or from -1 to 1. The standard distribution is an another option to standardize the data [17].

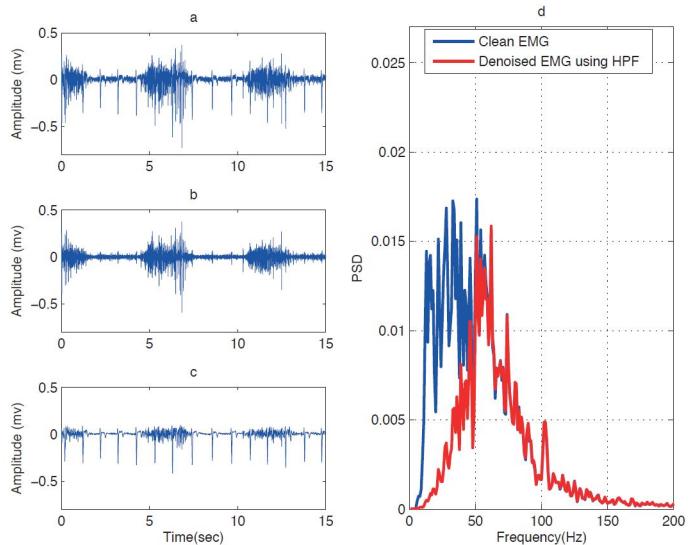


FIGURE 11: (a) Contaminated EMG, (b) denoised EMG, (c) estimated noise using HPF and (d) PSD of raw EMG and denoised EMG using HPF [2]

B. PRACTICAL DATA PROCESSING AND FEATURE EXTRACTION

It should be recognized that the test wavelength for EMG transmission depends on the number of tests taken in a second, when the testing rate is decreased by analog to digital conversion. The data has used which is tested at 200 Hz and another is at 400 Hz. First of all, data are measured at 200Hz. The EMG transmissions are also filtered as a pre-processing stage by a low-pass filter and a high-pass filter [17]. Low-pass filters are those filters, which permit the transmission to transfer in lower frequency data and then amplify the higher frequencies [17]. On the other hand, high-pass filters do the

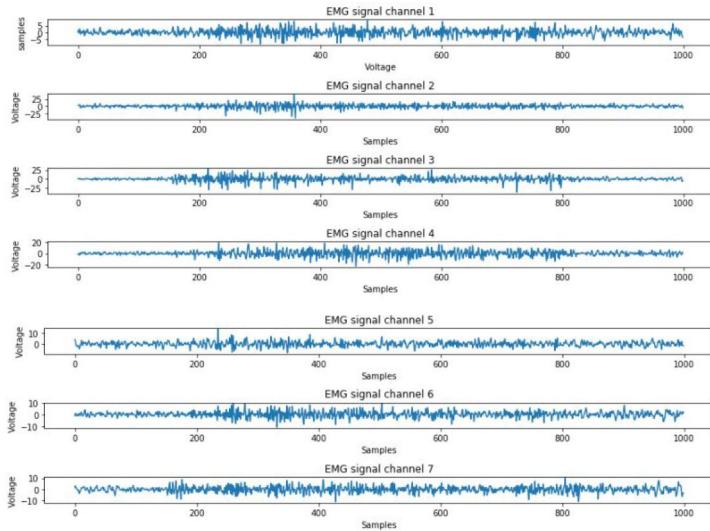


FIGURE 12: Data visualization for an EMG signal on electrodes [17]

complete reverse; lower wavelengths are eroded [17]. Researchers say in EMG signals that the intention of using a low pass filter is to suit the usual response rate of the muscle. In many other terms, a process delay is generated to imitate the electro-mechanical gap by the low-pass filter [17]. Different Feature Extraction methods have been used before passing the data into another standard machine learning algorithms [17].

FEATURE EXTRACTION

The efficiency of each pattern classification system based on almost wholly on the choice of characteristics used to classify continuous time wave shapes [8]. Given that the time-domain features are chosen due to their low computational complexity [8] and understanding that sEMG signals are nonlinear and stochastic, features are derived from time series analysis once a sEMG signal is detected and filtered suggested in [8].

I Variance : It is explained as scope of the sEMG signal's power consistency and is given by, [9]

$$\alpha^2 = \frac{1}{N-1} \sum_{n=1}^N x_n^2 \quad (6)$$

Where x_n is the nth data segment of sEMG signal which has N data samples.

II Waveform Length: It is an increasing change in magnitude over the entire time span from sample to sample which implies the caliber of diversity over the sEMG signal. It is done by [9],

$$WL = \sum_{n=1}^N |x_n - x_{n-1}| \quad (7)$$

III Integral of EMG : The function is an approximation of the addition of sEMG signal infinite values. It is done by [9],

$$I_{EMG} = \sum_{n=1}^N |x_n| \quad (8)$$

IV Zero Crossings : The function calculates the time the signal passes through zero. This framework is sensitive to sound, so to mitigate noise-induced zero crossing a threshold approach is implemented. It's done by [9],

$$ZC = \sum_{n=1}^N sgn(-x_n * x_{n+1}) > 0 |x_n - x_{n+1}| \geq 0.06 \quad (9)$$

V Slope Sign Changes : The function records the times sign shifts in the slope of the signal. Using this threshold means that the only important variations are calculated to decrease the noise caused by changes in the slope symbol. It's done by [9],

$$SSC = \sum_{n=1}^N [(x_n - x_{n-1}) * (x_n - x_{n+1})] \geq 0.06 \quad (10)$$

VI Root mean square: It represents the mean energy of the signal and is correlated with the constant intensity and non-tidy constriction. It's done by [17]:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (11)$$

VII Mean absolute value: It provides data about muscle constriction levels [17] . It indicates as:

$$MAV = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (12)$$

VIII Autoregressive Model: EMG signal has two type of nature which are non-linearity and non-stationary nature [12]. Because of these natures it is difficult to successfully examine the signals [12]. But EMG signal can be represented as a stationary Gaussian random process if the time interval is short. Each sample of the EMG signal time series is modeled as a linear combination of previous samples which is indicated by [12] ,

$$y_k = -\sum_{i=1}^n b_i y_{k-i} + g_k \quad (13)$$

Here AR is co-efficient, n is AR order and white noise [?]. Every orders of AR model are employed occasionally but the fourth order of AR is widely used [12]. EMG signal can be modeled as a list of parameters and this is the main benefit of AR model [12].Instead of using original EMG data model, parameters can be fed to the classification algorithm for recognition purposes which reduces computational inefficiency [12].

IX Integrated Average Value: The summation of the absolute signal value amplitude is known as integrated absolute value (IAV) of that signal [12]. It is written as,

$$IAV = \sum_{k=1}^N |x_k| \quad (14)$$

$$= N \cdot \frac{1}{N} \sum_{k=1}^N |x_k| \quad (15)$$

$$= N \cdot MAV \quad (16)$$

Here, N is used as the length of the signal and x represents the EMG signal in a segment [12] .

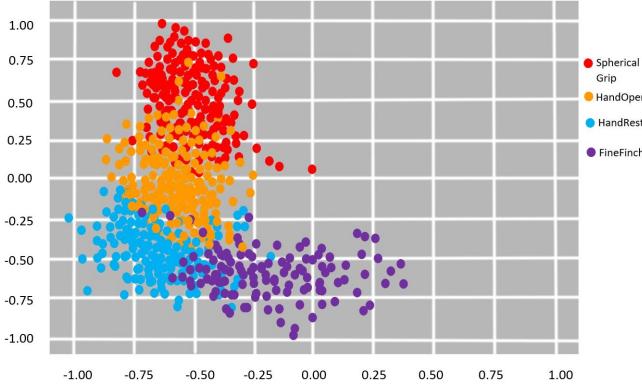


FIGURE 13: Distribution of four hand gesture classes after performing feature extraction methods

IX. DATASET PROPERTIES

Each dataset represents uninterrupted readings from 8 sensors. By this we get EMG data of altogether 64. And the final column stands for the result gesture which is collected by the recording of the given data of class 0 to class 3 [27]. So each line has the following structure of the given below: Data was recorded at 200 Hz, meaning that every line is

[8 sensors] [Gesture_Classes]

FIGURE 14: Interface of of Data set

$8*(1/200)$ seconds = 40ms record time [27]. A classifier can predict a class of gesture (0-3) given 64 numbers. Hand-rest-0, Hand-open-1, Spherical grip-2, Fine-pinch-3. Fine pinch is the index finger that hits the thumb and extends the rest of the fingers. Spherical grip is like the posture of grabbing a tennis ball. Gestures have been picked very randomly and Subjects enlisted were naturally limbed and without physical or neurological problems. They were implanted via the surface electrodes and tested with a sensor EMG (MayoWare). For 20 seconds, each movement was registered six times [27]. Each recording started with the movement already being planned and kept. Recording stopped while still holding the gesture. Through movement is held in a fixed position for a total of 120 seconds [27]. All of them reported in a short time from the same right forearm. That record of a certain type of gesture was concatenated into a .csv file [18] with a name (0-3) at Table 1.

These are just a sample of our data set which classifies gestures by reading muscle activity. A recording [27] of a human hand muscle activity producing four different hand gestures.The sample of our data set shown in Fig 15, Fig 16, Fig 17, Fig 18 sequentially :

TABLE 1: Column Structure of Dataset

Data Source	About This File	Columns
□0.csv □1.csv □2.csv □3.csv	65 columns 65 columns 65 columns 65 columns	muscle activity while gesture 0 ⊗+26.0 muscle reading 1 sensor 1 ⊗+04.0 muscle reading 1 sensor 2 ⊗+05.0 muscle reading 1 sensor 3 ⊗+08.0 muscle reading 1 sensor 4 ⊗-01.0 muscle reading 1 sensor 5 ⊗-13.0 muscle reading 1 sensor 6 ⊗-109.0 muscle reading 1 sensor 7 ⊗-66.0 muscle reading 1 sensor 8

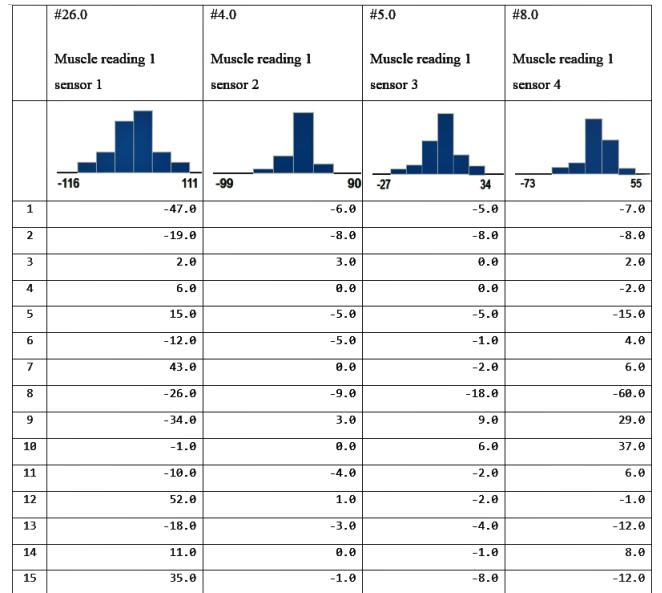


FIGURE 15: □ 0.CSV

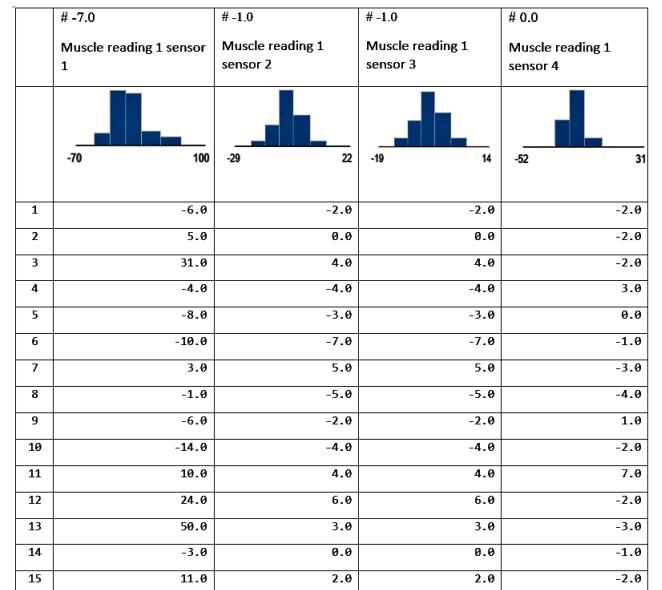


FIGURE 16: □1.CSV

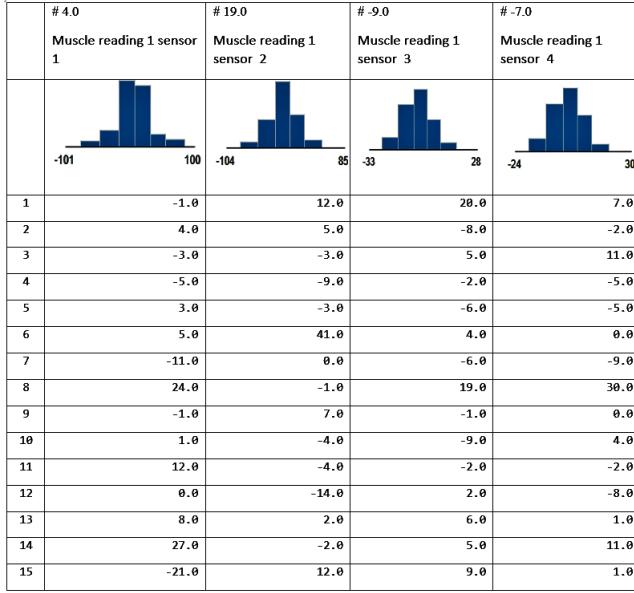


FIGURE 17: □ 2.CSV

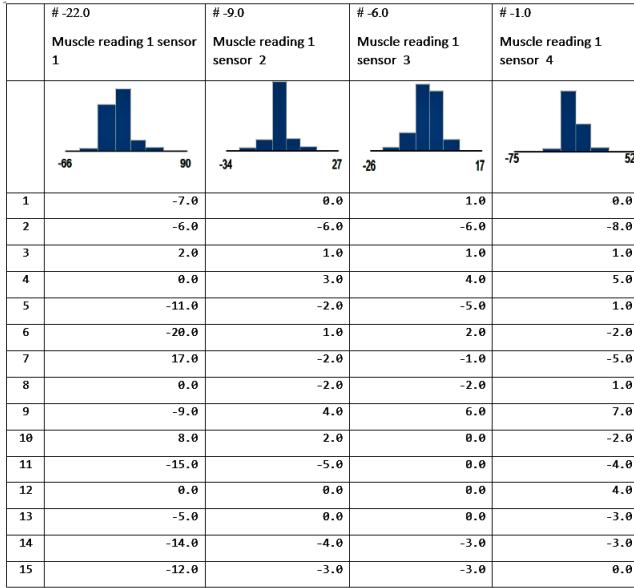


FIGURE 18: □ 3.CSV

X. SYSTEM IMPLEMENTATION

A. APERTURE SIZE SELECTION

The data gathered from the EMG is aperture -sized, and each window extracts features. The functions derived are often connected to build a broader collection of features. choosing the best window size multiple sliding apertures are selected as 128 ms, 256 ms, 384 ms, 512 ms, 640 ms, 768 ms, 896 ms, 1024 ms aperture and their execution was tested precisely for exactness of EMG signal ramification [12]. The enhancement of the aperture has been considered as 128 ms for all the cases. The ramification rate of hand movements with each of the aperture sizes has been measured 5 times for each subject. It has then determined the overall ramification rate

TABLE 2: Attribute Sets

Attribute Set	Attribute Parameters
Set 1	ARM (Order 4), WL
Set 2	WL, ARM, MAV
Set 3	MAV, ARM (Order 4), RMS,
Set 4	RMS, ARM (Order 4), WL
Set 5	RMS, WL, IAV, MAV
Set 6	WL, RMS, ARM (Order 4), IAV
Set 7	MAV, ARM ,WL, RMS,
Set 8	ARM, WL, MAV, IAV, RMS

[12]. Then, it determined the mean of all subject details for every aperture [12].

B. FEATURE PARAMETERS SELECTION

Classification accuracy is agitated by the possibilities of feature parameters. Table 2 determines the time domain traits which are molded into eight feature position for the representation of evaluation. To determine each feature set combined finger movements of individual and the mean sorting of distinct is calculated. The process is repeated for each subject. After that for the certain feature set over all the subjects the total average classification rate is determined. For the aspect of eight sets the process is iterated and this eight feature sets inaccuracy or the entire average classification rate is assessed.

C. FEATURE VECTOR CALCULATION

EMG signals sorting mainly lean on the detection of right attribute of vectors which are exclusively for them. Incompatible attribute parameters like ZC, MAV, RMS, WL and etc model are assessed, within each aperture attributes are extracted from EMG data. By using these feature parameters from each EMG data channel features are being calculated. First of all, by choosing the apertures of data then a feature parameter assessed are estimated for that particular aperture. Afterwards sliding the aperture away to review a new aperture of data and then the computation of feature values for each factor parameters is replicated. The computation features from one class of EMG data for example Hand Close of 1st aperture, 50Th aperture, 100Th aperture and 150Th aperture as an example is shown on Table 3 and Table 4 [12]. Concatenation of the calculated feature values are then used to form a substantial attribute vector [12].

TABLE 3: 1st and 50th aperture Attribute parameter values calculation

Attribute Parameter	Aperture 1		Aperture 50	
	1 st channel	2 nd channel	1 st channel	2 nd channel
MAV	0.000258	0.000451	0.000303	0.000497
RMS	0.000373	0.000562	0.000296	0.000501
WL	0.012994	0.017981	0.014003	0.013027
IAV	0.139863	0.219958	0.141967	0.201129
ARM (order 4)	-1.39889	-1.9117	2.02136	3.10024
	0.299758	0.714381	0.809527	3.001324
	0.286628	0.399899	0.340131	-1.20913
	-0.03101	-0.24217	-0.20142	0.121514

TABLE 4: 100th and 150th aperture Attribute parameter values calculation

Attribute Parameter	Aperture 100		Aperture 150	
	1 st channel	2 nd channel	1 st channel	2 nd channel
MAV	0.000312	0.000231	0.000251	0.000221
RMS	0.000398	0.000292	0.000301	0.000247
WL	0.009352	0.006997	0.008125	0.006507
IAV	0.156218	0.12597	0.114012	0.110201
ARM (order 4)	-2.52587	-1.9543	1.80252	-2.924
	2.49896	0.918212	0.656075	3.012529
	-0.91074	0.321091	0.298457	-1.58325
	0.059786	-0.2318	-0.16286	0.271091

TABLE 5: Interconnected Attribute values

Attribute parameters	1st Aperture		
MAV	0.000258		-1.39889
	0.000451		0.299758
RMS	0.000373		0.286628
	0.000562	ARM (order 4)	-0.03101
WL	0.012994		-1.9117
	0.017981		0.714381
IAV	0.139863		0.399899
	0.219958		-0.24217

Channels computed feature values are interconnected solely underneath one another. Interconnection of feature values of channels for 1st aperture is shown on Table 5. Then the feature vector is formed by all the integrated feature values of all aspect frameworks which are solely underneath one another. All 4 aperture feature vectors are shown on Table 6, for example:

TABLE 6: 1st-150th,4 aperture aspect vectors

Aperture 1	Aperture 50	Aperture 100	Aperture 150
0.000258	0.000303	0.000312	0.000251
0.000451	0.000497	0.000231	0.000221
0.000373	0.000296	0.000398	0.000301
0.000562	0.000501	0.000292	0.000247
0.012994	0.014003	0.009352	0.008125
0.017981	0.013027	0.006997	0.006507
0.139863	0.141967	0.156218	0.114012
0.219958	0.201129	0.12597	0.110201
-1.39889	-2.02136	-2.52587	1.80252
0.299758	0.809527	2.49896	0.656075
0.286628	0.340131	-0.91074	0.298457
-0.03101	-0.20142	0.059786	-0.16286
-1.9117	-3.10024	-1.9543	-2.924
0.714381	3.001324	0.918212	3.012529
0.399899	-1.20913	0.321091	-1.58325
-0.24217	0.121514	-0.2318	0.271091

D. KNN-SVM

A new approach is introduced the new category that blends the Support Vector Machine (SVM) and the K nearest neighbor (KNN). The classifier based on the classification of SVM and KNN where only one specific point for each class is picked. The method calculates the distance between the sample group and the appropriate SVM super-plane in feature space throughout the class step the test procedure will be listed on SVM if the distance is larger than the threshold given [16]; the KNN algorithm would instead be utilized. In the KNN algorithm, each support vector is chosen as a specific point and the distance between the test samples and all support vectors is compared [16]. The test sample can be classified by identifying the k-nearest reference neighbor. The computational analysis shows that the combined algorithm not only increases precision in comparison with SVM alone, and can also resolve the issue of choosing the kernel parameter for SVM [16].

E. KNN

KNN is admired now days because of its processing speed. Not only this simplicity in the process of recognition is really amazing but it increases the classification accuracy [14]. If the distance from test sample to optimal super plane in SVM is not greater than the threshold value computed by SVM, the responsibility falls on KNN to do the classification. In terms of time it can perform really well and has better accuracy

TABLE 7: KNN Standard classification accuracy rate

Data Sub	1	2	3	4	5	6	7	8	9	10	Avg. in %
1 st Sub	79	90	87	79	82	93	74	80	92	90	84.6
2 nd Sub	93	87	87	87	87	77	73	73	84	84	83.2
3 rd Sub	91	95	90	82	78	72	72	72	82	92	82.6
4 th Sub	95	75	90	100	90	90	90	90	90	95	90.5
5 th Sub	88	93	87	86	85	82	92	90	90	90	88.3
											Av. 85.84

in classification. EMG signals recorded from muscles of the user's and activation from these muscles were used in order to control a human arm. From the point of accuracy, simplicity KNN is very good to be implemented on many types of medical related appliances such as artificial limbs and high accurate robotic appliances where human interacts with robot. But KNN has to maintain a simple rule and that is its algorithm is fully based on the k data point in the data which it will be trained. There it can predict because of its nearest neighbor. For this reason its k must be selected very cautiously since it has a great influence on the classification performance [7]. The k value is fully dependent on the sets of data and the model which's specification is going to be used. The ranging of this k value is 1 to 10 [7].

Though the accuracy rate is lesser than SVM, it efficient in certain field where the classification rate of SVM is lower and overall it increases the accuracy of classification by supporting SVM. When the efficiency or feature distance in the plane gets lower for SVM, The KNN code gets the opportunity to run. Here is a glimpse of KNN classification code of our system at Fig 19. Without any due it can be

```

64 #KNN
65
66 class KNearestNeighbors:
67
68     def __init__(self, k=3, distance_metric=euclidean):
69         """Initialize k value and distance metric used for model."""
70         self.k = k
71         self.distance = distance_metric
72         self.data = None
73
74     def train(self, X, y):
75         """Zip labels and input data together for classification."""
76         # raise value error if inputs are wrong length or different types
77         if len(X) != len(y) or type(X) != type(y):
78             raise ValueError("X and y are incompatible.")
79         # convert ndarrays to lists
80         if type(X) == np.ndarray:
81             X, y = X.tolist(), y.tolist()
82         # set data attribute containing instances and labels
83         self.data = [X[i]+[y[i]] for i in range(len(X))]
84
85     def predict(self, a):
86         """Predict class based on k-nearest neighbors."""
87         neighbors = []
88         # create mapping from distance to instance
89         distances = {self.distance(x[:-1], a): x for x in self.data}
90         # collect classes of k instances with shortest distance
91         for key in sorted(distances.keys())[:self.k]:
92             neighbors.append(distances[key][-1])
93         # return most common vote
94         return max(set(neighbors), key=neighbors.count)

```

FIGURE 19

said that KNN algorithm is efficient and fast regarding of its simplicity. However, the performance of KNN is mostly depending on the value of k [14]. The k-value differs for the variety of data sets which is why k-value of several subjects are done. There the best k-value is found to be 1 which leads to optimal classification accuracy, it is also offering a consistent result due to smaller standard deviation value. KNN with k-value of 1 gives the optimal performance using the features.

F. SVM

The EMG categorization accuracy rate invariably exceeds 90 percent by using an SVM. The only lingering and challenging task is the selection of configuration of the SVM and the selection of feature extraction from the EMG signal of the functionality. Although SVMs are binary classifiers, only two categories of gestures might be distinguished by a unified SVM. A multi-class SVM is expected to interpret multiple gestures, which can be conducted in one of two ways. Very first methodology could be to solve the interoperability problem that includes details from all multiple classes. The second alternative certainly postulates the multi-class SVM from a binary SVM confluence. The juxtaposition of binary SVMs is much simpler and does not impede the classification performance [21]. There are two plausible schemes using binary SVMs for enacting multi-class SVM which are: "one against all" (OAA) or "one against one" (OAO) [21]. OAA implements n binary classifiers to train each classifier to distinguish from the residual classes. OAO entails $(n(n - 1))/2$ binary categories in which each category deviate from one pair of categories [21]. The penultimate classification in OAO is contingent on a voting mechanism, which comprises all the outputs of all classifiers. The ultimate outcome is the incident with the most votes. There is a more or less equivalent efficiency across both schemes as indicated in [21]. Nonetheless, the OAO scheme derives a stronger probability approximation across each class through evaluating all group pairs. This research incorporated with the EMG classification system with the OAO regime. We will first collect real data through EMG signal. Then we will do pre-processing and feature extraction of those signals using different methods. After that, we will classify our data set for getting the most accurate value for motion recognizing. Finally, we will feed those values to our simulated model and will test the accuracy of our model which is represented in Table 8.

Here are some glimpse of the python code we applied in predicting the states at Fig 20. For different hand gestures, we are predicting the possibilities of crossing the threshold which is determined by the algorithm itself. After the prediction, later on the value gets passed in the variables which is used for determining the state of hand simulation and giving command to the simulation after getting the online values from EMG sensor. After determining, the code sets the value of the variables for decided state by if else condition which sets commands the hand to be in different gestures, a glimpse of that code is shown below in Fig 21.

TABLE 8: Using support vector machine Standard classification accuracy rate

Data Sub	1	2	3	4	5	6	7	8	9	10	Avg. in %
1 st Sub	91	86	84	89	86	92	86	88	89	96	88.70
2 nd Sub	89	94	85	96	79	97	89	87	88	97	90.10
3 rd Sub	87	90	100	89	96	89	95	85	88	90	90.90
4 th Sub	96	92	100	97	97	86	97	93	97	92	94.70
5 th Sub	91	96	96	96	97	96	89	100	96	92	94.90
									Av.	91.86	

```

1 import matplotlib.pyplot as plt
2 from sklearn import datasets
3 from sklearn import svm
4 import numpy as np
5
6 from sklearn.externals import joblib
7
8 train_data = np.load('train_data.npy',encoding="l1")
9
10 X = np.array([i[0] for i in train_data])
11 Y = [i[1] for i in train_data]
12 print(len(X))
13 print(len(Y))
14 print((X[0]))
15 print(Y[0])
16 print(Y)
17
18 #SVM
19 clf = svm.SVC(kernel='linear')
20 clf.fit(X,Y)
21 filename = 'dataset.sav'
22 joblib.dump(clf, filename)
23
24 clf1 = joblib.load(filename)
25 print("output")
26 print(clf1.predict([X[0]]))
27 print(clf1.predict([X[1]]))
28 print(clf1.predict([X[2]]))
29 print(clf1.predict([X[3]]))
30 #print(clf.predict([X[4]]))
31 # print(clf.predict([X[5]]))

```

FIGURE 20

```

246 def emg_data():
247     global ax
248     if (state.get()):
249         ax.cla()
250
251     filename = filedialog.askopenfilename(initialdir = "/",
252                                         title = "Open File")
253     file_str=str(filename)
254     file_list=filename.split('/')
255     print(file_list[-1])
256     f_name=file_list[-1].split('.')
257     ff_name=f_name[0]
258     print(ff_name)
259     file_name_read.set(ff_name)
260
261     with open(filename, 'r') as in_file:
262         stripped = (line.strip() for line in in_file)
263         lines = (line.split(',') for line in stripped if line)
264
265         with open('0/' + ff_name+'.csv', 'w') as out_file:
266             writer = csv.writer(out_file)
267             writer.writerow(['title', 'intro'])
268             writer.writerows(lines)
269
270         return ff_name
271
272     clf1 = svm.SVC(kernel='linear')
273     filename = 'EMG-model.sav'
274     clf1 = joblib.load(filename)
275     print("output")
276
277     pred=clf1.predict([x])
278
279     if(pred==0):
280         v.set("state : HandRest")
281         #print("normal")
282     elif(pred==1):
283         v.set("state : HandOpen")
284         #print("normal")

```

FIGURE 21

```

int inByte = 0;
void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);
    establishContact(); // send a byte to establish contact until receiver responds
}

void loop() {
    // read the input on analog pin 0:
    int sensorValue = analogRead(A1);
    // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
    float voltage = sensorValue * (5.0 / 1023.0);
    // print out the value you read:
    Serial.print(voltage);
    float a = voltage ;
    if (a > 0)
    {
        inByte = a; // wait for a byte to be sent from the receiver
        int z_byte = a; // read the value on analog input A0
        Serial.write(z_byte * 0.25); // multiply the value by .25 and send it
    }
}
void establishContact()
{
    while (a< 0)
    {
        Serial.print("No Sensor Value"); // send a capital A
        delay(300);
    }
}

```

FIGURE 22

XI. SIMULATION AND ANALYSIS

A. ARDUINO INTERFACING

To attain the value from the EMG sensor it was attached with an embedded microprocessor. The EMG sensor had 8 probes which were put on to different part of the arm. The probes got raw data from the arm and the sensor optimized it and sends it to the microprocessor. And the microprocessor sends that data for processing to the processing device. The embedded microprocessor which is used with EMG sensor is mainly bound by the analog pin 1. Before using the EMG sensor it was calibrated so that sensor can give accurate results.

B. SIMULATION

The EMG sensor has 8 electrode and the raw EMG data stand in for an 8-column matrix. Trial starting time and the values representing the label of each trial are declared by variables

in the Autodesk file. After that the animation and python script was applied instead of any additional system for the simulation. To make sure our design could work well in real life, a simulation software named Autodesk 3D max is used in this work. 3d max is software for 3D modeling and rendering software for visualization, games, and animation. We did our simulated tests of 4 gestures accuracy in this software. We have shown our 3D models accuracy easily using 3D max's toolkits and motion techniques. However firstly, the 3D model of our prosthetic hand were made and then we have modified the model according to our work to meet the need for calculation and kept the most dimensions. We can get the test result by connecting the MyoWareMuscle Sensor with the embedded micro-processor by analogue pin. When

the electrodes will collect muscle value from the subject the values pass through the processing unit by embedded microprocessor. The simulator checks if the variable from python script in order to decide states. If the collected value is greater than threshold value, the variable sets its value to the right state. We have tested our gestures accuracy in the model and we have got accurate 4 gesture movement from our prosthetic model.

In Fig 23, when the online value from EMG sensor gets



FIGURE 23: Hand-Rest Simulation

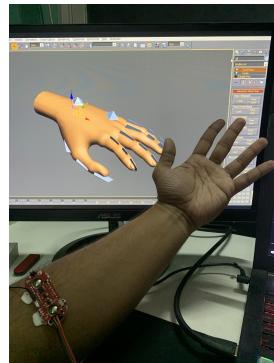


FIGURE 24: Hand-Open Simulation



FIGURE 25: Spherical Grip Simulation



FIGURE 26: Fine Pinch Simulation

higher than the threshold value of “Hand-Rest” state, the variable determines the state by prediction from algorithm and sets the state in “Hand-Rest” posture of hand. As a result from the python script, Autodesk gets the command of the posture hand rest and represents it virtually through animation. For Fig 24 ,Fig 25 and Fig 26 the prediction, decision and presentation is previewed in the simulation acquired in the same way.

XII. RESULT AND DISCUSSION

A. APERTURE SIZE VS. CLASSIFICATION RATE

Changing the aperture area, the divergence of classification rate is shown on Fig 27. Observing the aperture area change it is found that the expand of classification rate happens due to the increase of aperture with fixed aperture enhancement.

512 ms aperture area is the highest rate of classification rate. Substantial aperture takes substantial processing time and 512 ms aperture renders better classification percentage. In this study 128 ms as aperture size with fixed aperture enhancement has been chosen.

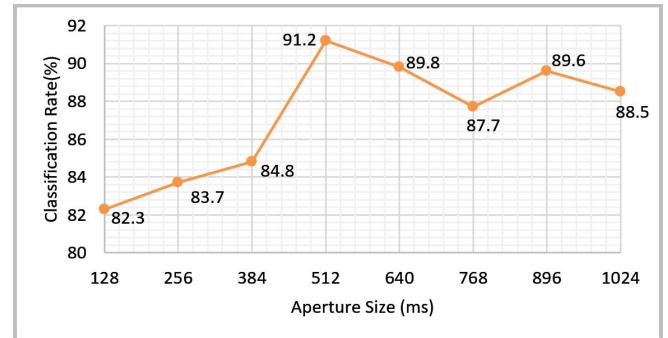


FIGURE 27: Relation between classification rate and aperture sizes

B. CHARACTERISTIC FRAMEWORK VS. CLASSIFICATION PERCENTAGE

Selection of characteristic frameworks assigning error rate or classification rate is shown on Fig 28. Standard error rate decreases due to the expanding characteristic parameters. Characteristic frameworks of characteristic set seven and eight from Table 2 manifests more fascinating outcome.

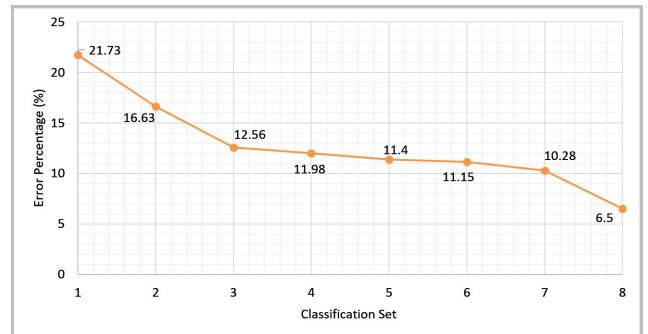


FIGURE 28: Error percentage and classification set relevance

According to Categorization set seven and eight the deduction of average error rate is small. Error percentage doesn't lessen if classification set eight is used which also includes AR model. It has been found that classification set eight's classification parameter has been recon to systematize EMG patterns.

C. AVERAGE CATEGORIZATION RATE

Comprehensive to All subject's average Categorization rate is 96.33 percent. Fig 29 shows mean categorization rate for each subject where x-axis stand for specific subject and y-axis specifies each value according to the categorizing of

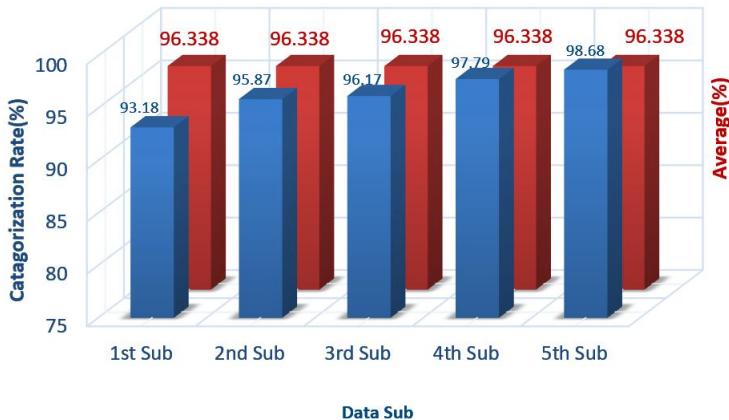


FIGURE 29: SVM-KNN classifier based mean categorization accuracy rate

each subject along with a horizontal line which represent overall mean categorizing rate.

XIII. LIMITATIONS AND FUTURE WORKS

A. LIMITATIONS

Though this system has an accuracy level close to 100 % and this is one of the precise predicting system, it has some limitations too. These are the limitations which were not possible to remove within the time binding we got and within the system budget we got. Some of the system limitations are stated bellow:

- Accuracy: The system has 96.33% of accuracy which concludes that the system is not 100% accurate and the inaccuracy rate is 3.67% which is certainly very low comparing other comparable system but it is not null.
- Feeling objects and temperature: The system has not described any solution about feeling objects, temperature and feeling texture of a surface just like the real organic hands.
- Gesture Variance: Unfortunately only four gestures were described throughout the whole project. Though classifying other gestures is very similar like classifying the four gestures.
- Rigidness about placements of electrodes: The points for placing the electrodes on the surface of the muscles has to be precise for every subject. It may work inaccurately in cas of not placing the electrodes precisely on the designated point of muscle which is already described in the system documentation.
- Two algorithms: More algorithms could be used to experiment the accuracy variance by merging them together, though it is still a conceptual concept among the researchers.
- Degree of movement: The system has not accounted the fact of degree of movement. Degree of movement allows every move of hand or any parts of body with a variance of precise angle.

- Dataset: As we tried to keep the system budget friendly, we couldn't afford to buy more costly datasets for the system
- No hardware implementation: The implementation is only shown through the simulation software, not through hardware implementation.
- Required hand portion: The system is valid for those people who have at least up to the elbow part of their hand. Unfortunately this is not valid for those amputees who lost their whole hand portion

B. FUTURE WORKS

The implemented prosthetic hand with 4 gesture movement control system is specific to biomedical applications which can be further extended, As future work and major development recommendations as it is suggested:

- Exploring other machine learning algorithms applied to the problem of hand gesture classification and comparing obtained results for experimenting the accuracy variance and merging them together if needed.
- Building a 3D printed hand and implementing our classifier as for now our implementation is shown only by software simulation system.
- Working on the points for placing the electrodes to be more precise for every subject.
- Increasing the accuracy to 100% for our classifier as now it has accuracy of 96.33%, so Focusing more on the algorithms to get rid of inaccuracy of 3.67%.
- Replicating the results presented in this thesis in trials with actual amputees.
- Introducing gesture recognition with both hands, enabling the creation of more natural interaction environments.
- Extending our results to other types of movements as in our thesis we have worked on 4 specific gestures only.
- Considering the Degree of Freedom in our classifier as this allows every move with a variance of angles.
- Introducing feedback system in our prosthetic hand so that amputees feel this hand as part of their body by placing pressure sensors on the tip of the fingers.

In conclusion, we can say that although there is still much to do for the betterment of this system but the implemented classifier and the system is solid foundation for the development of prosthesis hand systems.

XIV. CONCLUSION

To conclude, we have used a unique approach named electromyography for our prosthetic hand which is controlled by the electrical signals instinctively produced by a person's muscle system. Using the Machine learning and deep learning the EMG based prosthetic hand movement control is designed, developed and represented in Autodesk 3D max software. At first, the system is implemented with different methods of extraction and classification of features. Three Algorithms are implemented using the methods of machine learning as well as classification accuracy is achieved in this

work. For feature extraction, Furthermore, Mean absolute value, Root mean square, Slope Sign Changes, Zero Crossings and other techniques are used. In addition, for feature classification SVM, KNN, and combination of KNN and SVM algorithms are used. But the combination of SVM and KNN has the highest accuracy rate among other two algorithm as we get the accuracy rate of this classifier 96.33 percent while classifying the four different hand gestures which are hand-open, hand-close, spherical grip, and fine pinch. So, it can be easily concluded that the mix combination of KNN and SVM algorithm is more suitable for classifying different hand gestures with highest classification rate. We have also used the 3D modeling and rendering software name Autodesk 3D max for implementing our system the software simulation system. Output can be seen through the motion of our designed prosthetic hand model which performs accurately the intended action of the subject based on EMG signal (test data). This model also takes muscles time for processing the motions according the data which is collected by MyoWare Muscle Sensor. In the near future, our 3D cost effective Printable prosthetic hand with the accurate hand movement control system can be used in the by amputees to do the necessary activities in their daily regular lives to make their life more easier.

REFERENCES

- [1] Amputee statistics you ought to know, 2016. Limb Loss in the USA.
- [2] Sara Abbaspour. Electromyogram Signal Enhancement and Upper-Limb Myoelectric Pattern Recognition. PhD thesis, Mälardalen University, 2019.
- [3] TopHatRunner Airsoft. Electromyography (emg) sensors and signal processing. <https://www.youtube.com/watch?v=W5ZOg1x49Rc>, 2017. Accessed 1/8/2019.
- [4] M. Z. Al-Faiz, A. A. Ali, and A. H. Miry. A k-nearest neighbor based algorithm for human arm movements recognition using emg signals. In 2010 1st International Conference on Energy, Power and Control (EPC-IQ), pages 159–167, 2010.
- [5] J. Carpaneto, Karin Somerlik, T.B. Krueger, Thomas Stieglitz, and Silvestro Micera. Natural muscular recruitment during reaching tasks to control hand prostheses. pages 165–168, 06 2012.
- [6] Chih-Chung Chang, Chih-Jen Lin, and Chih-Wei Hsu. A practical guide to support vector classification. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2009. A Library for Support Vector Machines.
- [7] A. B. M. S. U. Doulah, S. A. Fattah, W. . Zhu, and M. O. Ahmad. Wavelet domain feature extraction scheme based on dominant motor unit action potential of emg signal for neuromuscular disease classification. IEEE Transactions on Biomedical Circuits and Systems, 8(2):155–164, 2014.
- [8] J. Fajardo, V. Ferman, A. Lemus, and E. Rohmer. An affordable open-source multifunctional upper-limb prosthesis with intrinsic actuation. In 2017 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pages 1–6, 2017.
- [9] J. Fajardo, A. Lemus, and E. Rohmer. Galileo bionic hand: semg activated approaches for a multifunction upper-limb prosthetic. In 2015 IEEE Thirty Fifth Central American and Panama Convention (CONCAPAN XXXV), pages 1–6, 2015.
- [10] R. Brent Gillespie, Jose Luis Contreras-Vidal, Patricia A. Shewokis, Marcia K. OMalley, Jeremy D. Brown, HarshavardhanAgashe, Rodolphe-Gentili, and Alicia Davis. Toward improved sensorimotor integration and learning using upper-limb prosthetic devices. August 31 -September 4 2010.
- [11] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.
- [12] Ariful Islam and M Alam. Classification of electromyography signals using support vector machine. 01 2017.
- [13] Subash Chandra Bose Jaganathan, Chandra Bose, Ahmed Badawy, Jameel Baili, Babu Suresh, Suresh Changalasetty, Wade Ghribi, and Harun Bangali. A hybrid ga/knn/svm algorithm for classification of data. 02 2019.
- [14] Kim KS, Choi HH, Moon CS, and Mun CW. Comparison of k-nearest neighbor, quadratic discriminant and linear discriminant analysis in classification of electromyogram signals based on the wrist-motion directions. 11:740–745, May 2011.
- [15] K. Kurabayashi, S. Shimizu, A. Kawachi, and T. Taniguchi. A discrimination system using neural network for sma prosthesis. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94), volume 3, pages 1832–1839 vol.3, 1994.
- [16] R. Li, S.-W Ye, and Z.-Z Shi. Svm-knn classifier - a new method of improving the accuracy of svm classifier. 30:745–748, 05 2002.
- [17] Hassen Maamri. Classifying Hand Gestures for Prosthetic Arms by Identifying EMG Signals with Machine Learning and Deep Learning. PhD thesis, 06 2019.
- [18] Dr. Roberto Merletti and Politecnico di Torino. Standards for reporting emg data. Journal of Electromyography and Kinesiology JEK, 1999.
- [19] NettaGurari, Kathryn Smith, Manu Madhav, and Allison M. Okamura. Environment discrimination with vibration feedback to the foot, arm, and fingertip. IEEE, January 2009.
- [20] Johnny L. G. Nielsen, Steffen Holmgård, Ning Jiang, Kevin B. Englehart, Dario Farina, and Philip A. Parker. Simultaneous and proportional force estimation for multifunction myoelectric prostheses using mirrored bilateral training. IEEE Trans. Biomed. Engineering, 58(3):681–688, 2011.
- [21] Mohammadreza Oskoei and Huosheng Hu. Oskoei, m.a.: Support vector machine-based classification scheme for myoelectric control applied to upper limb. iee transactions on biomedical engineering 55(8), 1956–1965. IEEE transactions on bio-medical engineering, 55:1956–65, 08 2008.
- [22] Saki Oyama. Emg processing. <https://www.youtube.com/watch?v=6y6QDPBRaCc>. Accessed 9/5/2019.
- [23] Access Prosthetics. 15 limb loss statistics that may surprise you. <https://accessprosthetics.com/category/living-with-limb-loss/>, Oct 18, 2017. Access Prosthetics Blog, Living with Limb Loss.
- [24] Lemon R. Precise Finger Movements in Monkeys, volume 62 of Neuromethods. Humana Press, London UK, 2011.
- [25] N. S. Rekhi, H. Singh, A. S. Arora, and A. K. Rekhi. Analysis of emg signal using wavelet coefficients for upper limb function. In 2009 2nd IEEE International Conference on Computer Science and Information Technology, pages 357–361, 2009.
- [26] Hardeep S. Ryait, A. S. Arora, and Ravinder Agarwal. Study of issues in the development of surface emg controlled human hand. Journal of Materials Science: Materials in Medicine, 25 June 2008.
- [27] Kirill Yashuk, Igor Perikov, and Miroslav Bulgakov. Classify gestures by reading muscle activity. <https://www.kaggle.com/kyr7plus/emg-4/discussion>.

•••