# EMGCaps: Electromyographic Hand Gesture Recognition using Capsule Network

Oussema Dhaouadi
*School of Electrical & Electronic Engineering*
*Nanyang Technological University, Singapore*
oussema.dhaouadi@gmail.com

***Abstract*—In recent years, Deep Learning (DL) techniques have played an important role in computer vision and speech recognition fields due to its high performance. The classification of physiological data has significant usage in the human-machine interfaces for building prosthetic system, such as robotic hands. Yet, physiological signals did not prove high potential in the medical area as 2D medical imaging analysis does. Given the recent progress in DL-based computer vision, we present EMGCaps, a framework for classifying hand gestures using Capsule Networks (CapsNets). We conduct a detailed study to understand the basic functioning of capsules within the network compares its performance with the current machine and deep learning techniques by trying different input features. In our analysis, we have used the NinaPro DB5 dataset. It contains sEMG signals acquired from 10 subjects at 53 different hand movements. Our experimental results show that CapsNet in almost all settings surpasses the state-of-the-art results and outperforms the other conventional classifiers such as Random Forest (RF), k-Nearest Neighbour (kNN), Fully Connected Network (FCN) and Convolutional Neural Network (CNN) with an accuracy of 93.27%.**

***Index Terms*—surface electromyography (sEMG), classification, capsule network (CapsNet), gesture recognition, pattern recognition**

## I. Introduction

In the past few decades, bio-medial signals have been used for medical purposes. For instance, the human body muscles generate the myoelectric signals (MES), which could be used for pattern recognition purposes, after converting them into electromyographic (EMG) signal. Electromyography is a biomedical procedure, which measures electrical currents in muscles in reaction to neuromuscular stimulation by a nerve in form of EMG signals [1]. The surface EMG (sEMG) signals are collected using electrodes mounted on the skin surface, which captures muscular information during contractions like the flexion or expansion of an articulation.

Several computational frameworks have been designed to comprehend human intention using sEMG signals and to develop sophisticated systems such as hand prosthetic, wheelchair control, teleoperating robots, and virtual interface [2]. Recent work has emphasized the significance of hand motion recognition for collaborative human and machine. Even though the novel classification strategies have a tremendous success rate, the identification of sEMG signals is not stable due to signal dependency on hand position, weather condition, human physiology and psychology, the number of electrodes available for data acquisition, and the number of gestures to be recognized.

In the medical field, deep learning has been competitive over conventional machine learning approaches, owing to its impressive performance of classification. In comparison to conventional machine learning, deep learning frameworks have a particular potential to extract features from the raw data. Nevertheless, the electromyographic gesture recognition efficiency based on deep learning approaches is far from the ideal. For instance, the end-to-end Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) based model [3] only achieved an accuracy of 61.4%. Traditional sEMG-based gesture recognition approaches are able to achieve adequate accuracy by means of hand-crafted features. Such classical feature sets include significant heuristic knowledge. Therefore, incorporating deep learning approaches when using classical features may enhance the performance of gesture recognition. Several works [4]–[7] replace traditional machine learning techniques by deep learning networks to classify the complex-structured deterministic features. From the EMG signals, features are derived to perform an accurate pattern recognition for myoelectric applications. The characterization of features may be conducted on the various domains such as the time domain [6]–[8], the frequency or spectral domain [9], and time-frequency [4], [5], [10], among others.

One of the key techniques used in EMG-based pattern recognition is the Support Vector Machines (SVM) method [11]. Its main idea is to determine an n-D hyperplane to divide the input feature points into distinct sets. This method can achieve accuracy above 90% [11] using the kernel trick and the soft margin technique but toward a small number of classes. There are many works on a large number of sEMG-based approaches for hand gestures recognition.

Kuzborskij et al. [12] focused on classifying sEMG signal of 52 hand gestures from the NinaPro dataset [13]. Different features such as Histogram (HIST), Mean Absolute Value (MAV), Short-Time Fourier Transform (STFT), and Multidimensional Discrete Wavelet Transform (mDWT) were derived from different EMG signals and several models for testing and evaluation such as k Nearest Neighbor (k-NN), Multilayer Perceptron (MLP), LDA models, Nonlinear Support Vector Machines (SVM-BRF), and Linear Support Vector Machine (SVM-linear) were used. It was observed that the best model, which is SVM-BRF model combined with MAV features, reaches a recognition accuracy of around 80%.

Approximately 75% accuracy was reached by He et al. [14] in the differentiating between 52 variations of hand gestures of 27 subjects. They integrated LSTMs and MLPs to identify sEMG signals from the NinaPro DB1 dataset. Hu et al. [15] suggested an attention-based hybrid CNN and Recurrent Neural Network (RNN) model that was evaluated on databases including NinaPro DB1 and NinaPro DB2 datasets, among others and achieved an accuracy of 87.0%.

It is clear that the deep learning approach will bypass the restriction of feature engineering involving a higher quality of pattern recognition. Many studies have shown that the deep learning methods perform better than traditional machine learning methods. Nonetheless, the deep learning-based recognition methods for sEMG signals are to boost in terms of feature engineering and accuracy.

In 2017, Sabour et al. proposed Capsule Networks (CapsNet) [16] to resolve weaknesses of CNNs such as the lack of spatial knowledge within the pooling layers and the ambivalence of spatial connections between learned features, especially in the field of image classification. CapsNets learn not only the existence of feature but encode properties of the input features. The spatial relation between features is also learned to help the network to learn the general structure of the data and the connection between features. The capsules, which replaces neurons in a conventional neural network, are dynamically routed in one layer to the capsules in the next layer. The route is based on an agreement protocol, i.e. the information is routed to the next layer only if meaningful part-whole relationships are observed.

In this paper, we present EMGCaps, a CapsNets-based hand gesture recognition model using EMG signals. In our model, we use the structure of 2D CapsNet to build the 1D-CapsNet by designing capsules that work along the feature map axis. EMGCaps is, therefore, able to captures the relationship between the features. Various feature maps are used for classification and comparison.

We divide this paper into 4 sections. Section I introduces this work by presenting the related work on the recognition of sEMG and the state-of-the-art computer vision classifier namely CapsNet. The EMGCaps model that we developed is presented in Section II. Section III evaluates the model on the NinaPro DB5 dataset. The work is outlined in Section IV.

## II. Methodology

### A. *Framework*

Fig. 1 shows the framework of the proposed EMGCaps for hand gesture recognition. The framework is divided into a DATA CONSTRUCTION module, FEATURE EXTRACTION module und CLASSIFICATION module. In the DATA CONSTRUCTION module, segmentation is performed, and the data is divided into train and validation sets. The FEATURE EXTRACTION module builds feature spaces. The CLASSIFICATION module is fed by the features and trained. We implemented different types of feature extractors and classifiers for comparison purposes.The modules are described in the next subsections.

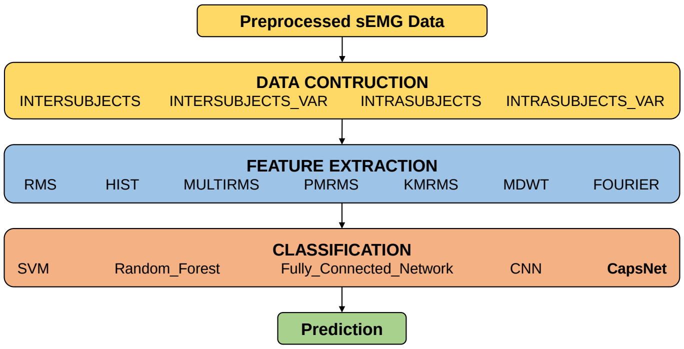


Fig. 1. Framework of EMGCaps for sEMG gesture recognition

### B. *Data Processing*

Since sEMG is a stochastic process, the signal differs gradually over time. This variation results in a signal showing non-stationarity, which violates the standard assumption of stationary signals for the feature extraction module. As a consequence, data segmentation is widely used to arrange the EMG signal into data frames from which the properties may be called wide-sense stationary. In this procedure, sEMG is processed by sliding windows. This window is shifted with the increment window. Fig. 2 illustrate the segmentation procedure.

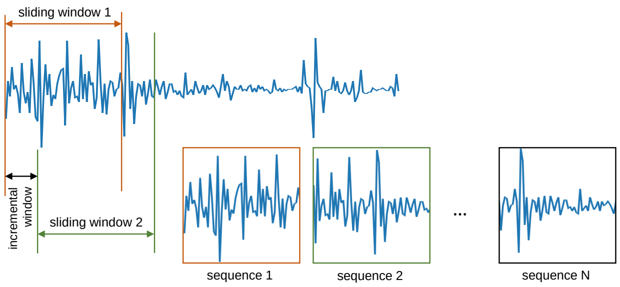


Fig. 2. Ground Truth

The input delay is a significant factor to consider for prosthetic control operating in real-time. In this work, we choose an incremental-window of 100 ms and a sliding-window of 200 ms to separate the sEMG. The maximum delay proposed by Hudgins et al. [17] is 300 ms. Our settings meet the requirements of a real-time recognition in term of data acquisition delay, however, the delay needed for the classification may be a critical factor for a real-time application.

Four types of data structures are available for the feature extractor. We distinguish between 2 possibilities of splits into a train and test sets. The first method, called INTERSUBJECT, splits the data according to the repetitions. In this work, we use the second and the fifth repetitions for the validation. The second method, called INTRASUBJECT, splitting is to divide the data according to the subject. The sEMG recorded for patients with numbers 8 and 10 are stored in the validation set. The VAR option refers to the length of a variable window of the segment, where we keep track of the sEMG signal until the rest class is detected. This method provides the classifier with the entire data of the single gesture, which may improve the accuracy but could lead to a violation of real-time requirements. The fact that the gesture lengths are not equal is due to the following. Movements conducted by subjects may not completely match those seen on the computer used for recording the gestures due to individual response times. Movement detection algorithms such as the Lidierth threshold dependent algorithm [18] and the generalized likelihood ratio algorithm [12] have been used to correct incorrect labeling.
Through the above construction, sets of segments, forming the train and validation sets, are available for the FEATURE EXTRACTION module.

### C. Feature Extraction

Each retrieved segment is processed by this module in order to extract hand-crafted features. Feature extraction is usually done in three domain representations of the EMG signal: the Time-Domain (TD), the Frequency-Domain (FD), and the Time-Frequency-Domain (TFD). Extraction of the TD feature refers to procedures which extract characteristics from the time sequence. These are widely used features in myoelectric pattern recognition systems owing to their high precision in low-noise conditions [4], [19]. TFD applies to any transformation applied to the EMG signal that integrates knowledge about time and frequency. FD features are generally derived from the Fourier transform [20]. The most widely employed TFD is the wavelet transition, where several scale/shift combinations take the cross-correlation between the EMG signal and a wavelet function [4]. We define the data matrix $\mathbf{X}$, where its rows represent the sEMG signal value over the time steps $T$, and its columns represent the $C$ channels (i.e. the number of electrodes). One typical TD feature is the Root Mean Square feature vector $\mathbf{x}^{RMS} \in \mathbb{R}^C$ of size equal to the channels and is defined as follows:

$$x_j^{RMS} = \sqrt{\sum_{i=0}^{T-1} X_{i,j}^2}. \tag{1}$$

Another feature vector is the distribution of the signals could be represented by the bins of histograms. The feature vector $\mathbf{x}^{HIST} = vec(\mathbf{X}^{HIST})$ of size $C \cdot B$ is the flattening of the matrix $\mathbf{X}HIST$ with elements defined as follows:

$$X_j^{HIST} = hist_B(\mathbf{x}_j), \tag{2}$$

where $\mathbf{x}_j$ denotes the j-th channel signal vector (the j-th column of $\mathbf{X}$). The feature vector $\mathbf{x}^{MULTIRMS} = vec(\mathbf{X}_{d,j}^{MULTIRMS})$ is computed by dividing the segment into $D$ subsegments and computing their RMS value.

$$\mathbf{X}_{d,j}^{MULTIRMS} = \sqrt{\sum_{i=d\cdot\frac{T}{D}}^{(d\cdot\frac{T}{D}+W-1) \mod T} X_{i,j}^2}. \tag{3}$$

The feature vector $\mathbf{x}^{PMRMS}$ is the padded version of MULTIRMS but uses the Principle Component Analysis (PCA) to reduce the dimensionality of the vector by preserving the components with the highest energy. The KMRMS determines the K centrists of the sEMG energy points using the k-Means clustering method in the channel space. The feature vector $\mathbf{x}^{KMRMS}$ represents the euclidean distances (defined in the channel space) of the centroids from energy points, which are RMS of sub-segments of the input data. The Fourier feature vector $\mathbf{x}^{FOURIER}$ consists of the norms of the Fourier coefficients.

### D. Conventional sEMG Classifiers

*1) Support Vector Machines:* Support Vector Machines (SVMs) are vector binary classifiers that aim to optimize the two classes' margin [21]. Their wide usage is attributed to the potential of utilizing the kernel functions. These functions are used for non-linear problems and allow SVMs to map the data into a large dimensional space. Since the regular SVM is specified in binary separation, the multiple binary classification problems could be solved by multiple binary classification. A single trade-off hyperparameter $C$ may be used to control the equilibrium between overfitting and underfitting, thereby restricting the classifier to the certain capacity.

*2) Random Forests:* The Random Forest (RF) [22] is one of the most popular and powerful ensemble methods used in Machine Learning. This supervised learning algorithm randomly generates many decision trees and merges them into one forest. The aim is not to focus on a single model of learning, but instead on the combination of decision models to increase precision. The concept behind a tree is to search for the pair of variable-value within the training set so that two best child subsets are created. The goal is to perform divisions that optimize a splitting criteria. The RF method applies the bagging method to the feature space, which increases the randomness and diversity.

*3) Fully Connected Networks:* The most common form of Artificial Neural Network (ANN) is undoubtedly the Fully Connected Network (FCN) (aka. Multi-Layer Perceptron (MLP)) [23]. The network consists of more than three entirely inter-connected layers. An input and output layers and at least one hidden layer are connected in sequence. Each layer consists of many neurons, which feed non-linear activation of weighted inputs. Using the back-propagation algorithm, the weights updated.

*4) Convolution Neural Networks:* A Convolutionary Neural Network (CNN) [24] has an input layer, an output layer, and multiple hidden layers. Any of these layers are convolutionary, utilizing a process of mathematics to feed information on to successive layers, which simulates some of the behavior in the visual cortex of humans. It shares convolution weights called kernel or filter in the convolutional layer. These kernels are learned by the means of back-propagation algorithm. Specific learned features are extracted from the input data and a class is predicted depending on their existence. It has been extended effectively to the identification of sEMG signals over the last years [10], [25]. In this paper, experiments related to the CNN model are conducted on one hidden convolutional layer followed by a max-pooling layer to extract features and two fully connected hidden layers for classification purposes.

### *E. Capsule-based sEMG Classifier*

Although CNN achieved high performance in recognition tasks, it drops out some important information through the pooling layers. This operation looses the global pose information of a feature, i.e. the relationship and correlation between features themselves. Sabour et al. [16] presented Capsule Network, which solved this problem and achieved an accurate classification in the computer vision field without using the pooling layer.

The structure of EMGCaps is inspired by the architecture of Capsule Networks, where the capsules are adjusted to 1D input tensors. A capsule is a group of neurons that represent properties of a single feature [16]. Capsules encapsulate in a vector all relevant information about features they are recognizing. In other words, they learn not only the existence (as CNNs do by to the activations of the last layer) but also the "instantiation parameters" such as the generalised pose. The activation of a vector capsule is represented by its length and encodes the probability of detection of a feature. A property that makes Capsules successfull is the spatial equivariance, which is the combination of invariance (existence probability) fulfilled in CNNs and variance (all values forming the capsule and encoding relationships and feature characteristics) missed in the CNN. Since we are operating on a time sequence data, features extracted at time $t$ are related to features extracted at a previous time $t - N$. That makes capsules suitable for encoding these correlations.

*1) Capsule Structure:* Unlike a simple neuron that sums the weighted scalar inputs, applies a non-linear function and outputs a scalar, a capsule sums inputs multiplied by matrix weight, applies a non-linearity, and outputs a vector. A layer $L$ of capsules is the set of all capsules $i$ with $i \in \Omega_L$. The computation inside a capsule is as follows [16]:

- **Matrix multiplication** Input vectors of the a capsule $j \in \Omega_{L+1}$ in layer $L + 1$ are the outputs of capsules $i \in \Omega_L$ of the previous layer $L$. They encode probabilities of detection a specific features in form of norm $\|\mathbf{u}_i\|$ and some internal state $\mathbf{u}_i$ of them (frequency, number of zero crossings, etc). By multiplying each input vector $\mathbf{u}_i$ with the corresponding learnable weight matrix $\mathbf{W}_{i,j}$, the capsule encodes the relationship between a lower level feature (e.g. a single pulse) and a higher level feature (e.g. multiple pulses). This relationship is represented in the so-called coordinate frame, i.e. the reference by which the pose (temporal causality) of a feature is described. That outputs a prediction $\hat{\mathbf{u}}_i$ of the pose of an entity in the next layer referred to the entity described by the capsule $\mathbf{u}_j$ in the current layer. In other words, $\hat{\mathbf{u}}_i$ represents where the higher level feature should be compared to the detected pose of the lower level feature.
- **Scalar weighting** By scaling the predictions $\hat{\mathbf{u}}_i$ with $c_i$, the capsule $cap_i$ can agree to which higher level capsule $cap_j \in \Omega_{L+1}$ it will forward its output. If the predicted position $\hat{\mathbf{u}}_{j|i}$ for the feature represented by the capsule $\mathbf{u}_i$ is in some space close the vector of the higher level capsule $\mathbf{v}_j$ the capsule $cap_j$ will be activated and the length of the corresponding vector $\mathbf{v}_j$ is high. Otherwise, if the projection poses (cosine similarity) lands far from the cluster, the higher level feature exists but with very low probability and can be assumed as unexisting. In this case, the capsule remains inactivated. That process is called routing by agreement, i.e. the preference of lower-level capsules to forward its output to higher-level capsules on whose predicted output they agree on. The goal is to compute the weights $c_i$ in an appropriate way.
- **Predictions summation** Similarly to a regular artificial neuron, it is represented by a combination of inputs.
- **Capsule activation** The paper [16] introduces a novel non-linear activation function called squash function, that ensures that the length of a vector is not greater than one without changing its direction and is defined as:

$$squash(\mathbf{s}_j) = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|} . \tag{4}$$

Figure 3 shows an example of how the routing by agreement works in capsule network with 2 Caps layers and 2 classes.

*2) Dynamic routing:* The routing algorithm is executed after computing the predicted positions of the higher level feature described in the previous section. It calculates the forward by assigning the correspondent value to the coupling weights $c_i$:
The inputs of the routing function are the predictions $\hat{\mathbf{u}}_{j|i}$, the number of routing $r$ and the capsules set $\Omega_L$ of layer $L$.
The coefficients $b_{i,j}$ are temporary values that are iteratively ($r$ times) updated to calculate the agreement of each lower level capsule with the higher level capsule and will be finally stored in the weights $c_{i,j}$ by applying the softmax function, which converts the values to non negative scalars to map into a probabilistic space.

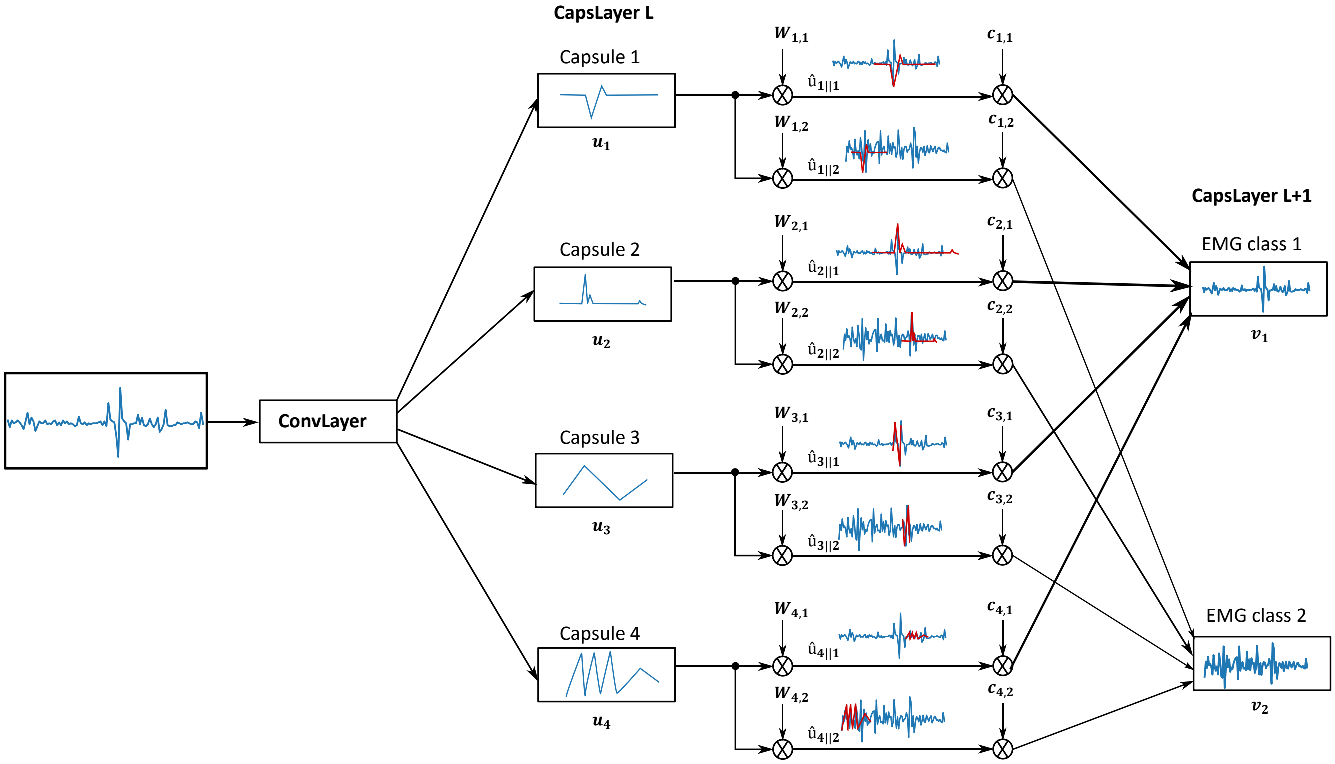


Fig. 3. Internal workings of the capsule and routing by agreement. The Intensity of the output arrows are proportional to the corresponding weights $c_{i,j}$ that are adjusted by the dynamic routing algorithm.

$s_j$ describes the average of the predicted positions $\hat{\mathbf{u}}_{j|i}$ and constructs the centroid of the cluster formed by these positions. By applying the squash function to $s_j$, the length of that vector will be smaller than 1 in order to model the probability of activating the capsule $j$.

The temporal values of $b_{i,j}$ are updated by projecting the predicted positions $\hat{\mathbf{u}}_{j|i}$ (position of a higher-level feature described by the capsule $j \in \Omega_{L+1}$ relative to all lower-level features described by the $i \in \Omega_L$) on the squashed cluster vector $\mathbf{v}_j$ (vector of the higher level capsule). If those two vectors are similar, the projection value is high. In other words, a higher-order capsule receives multidimensional prediction vector from lower-level capsules and outputs a high presence probability of an entity if a cluster of predictions that are in strong agreement is found.

After $r$ iterations the value of $b_{i,j}$ increase if most capsules agree for higher-level capsules. The number of iteration $r$ is a hyperparameter which must be determined carefully because otherwise, this can lead to overfitting. The pseudo code of the routing algorithm is described in Algorithm 1.

**Algorithm 1** Dynamic routing algorithm [16]

```
procedure DYNAMIC ROUTING($\hat{\mathbf{u}}_{j|i}$, $r$, $\Omega_L$)
$\forall\, i \in \Omega_L$ and $\forall\, j \in \Omega_{L+1} : b_{i,j} \leftarrow 0$.
    for r iterations do
        $\forall\, i \in \Omega_L : c_{i,j} \leftarrow softmax(\mathbf{b}_i)$
        $\forall\, j \in \Omega_{L+1} : \mathbf{s}_j \leftarrow \sum_i c_{i,j} \hat{\mathbf{u}}_{j|i}$
        $\forall\, j \in \Omega_{L+1} : \mathbf{v}_j \leftarrow squash(\mathbf{s}_i)$
        $\forall\, i \in \Omega_L$ and $\forall\, j \in \Omega_{L+1} : b_{i,j} \leftarrow b_{i,j} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j$
    return $\mathbf{v}_j$
```

*3) EMGCaps Architecture:* While a simple classifier outputs the belonging probability of the input sample, EMGCaps uses additionally the values of the output vectors of the capsules as a low-dimensional representation of the input feature, as it contains the most relevant features (relationship between features) for the recognition task. The length of the vector formed after this encoding being element of the low dimensional space is the belonging probability. To encourage learning some specific features, EMGCaps introduce after encoding a decoder using FCN that has a very low impact on the total loss function. To summarize, the architecture is formed as follows:

The first layer of the encoder is a 1D convolutional layer used to detect primary features from the input feature vector.

The second layer has $A$ primary capsules that produce combi-

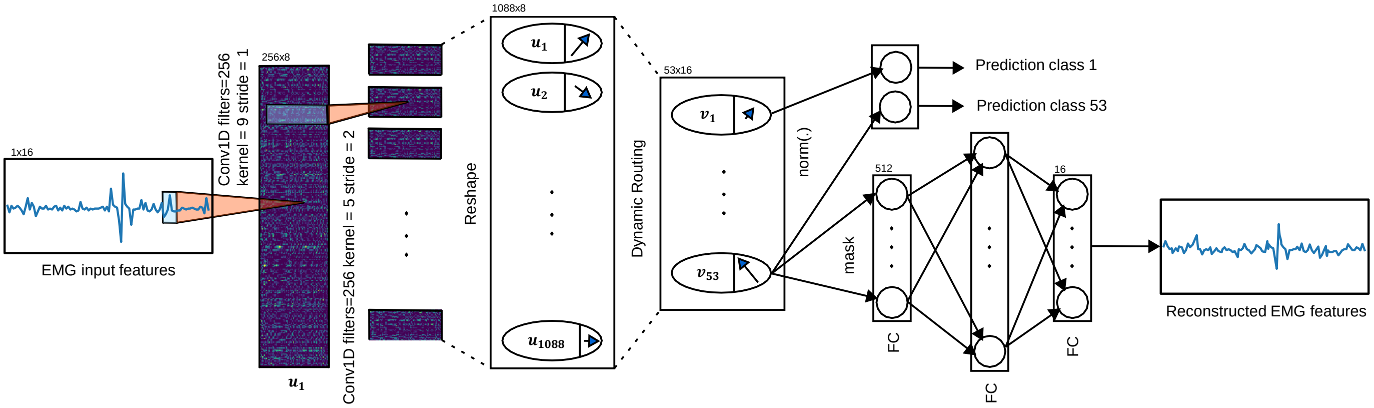


Fig. 4. Architecture for EMGCaps for $1 \times 80$ input feature vector: Encoder and Decoder.

nations from the extracted features in the previous layer. They are similar to a convolutional layer.

The third layer has $B$ parent (class) capsules. They use the dynamic routing algorithm to maps the combined features from the second layer to $B$ capsules. Each capsule has a $D$-dimensional vector.

At this point the network can classify the inputs by optimizing the following margin loss function, defined by [16]:

$$\begin{aligned}\mathcal{L}_{margin} = \sum_{i\in\Omega_{last}} & T_i \max(0, m^+ - \|\mathbf{v}_i\|)^2 + \\ & \lambda(1-T_i)\max(0, \|\mathbf{v}_i\| - m^-)^2\end{aligned} \tag{5}$$

The first part of the loss function corresponding to $(1 - T_c)$ is the loss calculated for correct capsules and the last for incorrect capsules, i.e. if the NN classifying the gesture 1, the corresponding class capsule is composed of only the first part of the loss function. All other class capsule (from 2 to 53) are using only its second part. Therefore the variable $T_c$ is Boolean that selects of which loss part the corresponding loss function is composed. $\lambda$ is a constant used for numerical stability and generally equal to 0,5. $\|\mathbf{v}_i\|$ is the existence probability of an object. In case the capsule is correct and the probability $\|\mathbf{v}_i\|$ is greater than $m^+$, the corresponding loss is zero. I.e., if the correct DigitCap recognize the correct class with a probability greater than $1/n_{classes}$ then the loss is zero and penalizing this class capsule is not necessary any more. If the class capsules is incorrect (2 to 53) and its length (probability is less than 0.1) the corresponding loss is zero.

Using the L2 norm in the loss function (squares) is an empirical choice and works better than other norms after preliminary tests [16].

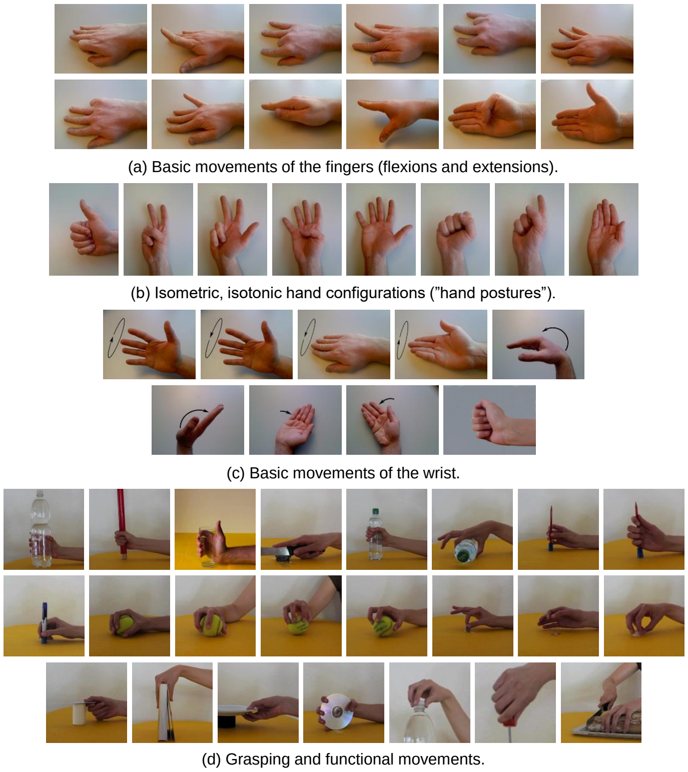
The next layer is the first layer of the decoder. It inputs only the values of the vector corresponding to the correct class capsule. All other class capsules are masked. The capsule values propagated through two dense layers and each neuron of the last layer is a pixel of the reconstructed input vector. The decoder is used as a regularizer and forces capsules to learn only useful features that contribute in the reconstruction of the input. $\mathcal{L}_{reconst}$ denotes the loss function of this part, which uses the Euclidean distance between the original and the reconstructed input. Therefore, the total loss of the EMG Caps is:

$$\begin{aligned}\mathcal{L} &= \gamma \cdot \mathcal{L}_{margin} + (1-\gamma)\cdot\mathcal{L}_{reconst} \\ \mathcal{L}_{reconst} &= \sum_i (\mathbf{x}_i^{FEATURE} - \mathbf{x}_{i,reconst}^{FEATURE})^2,\end{aligned} \tag{6}$$

where $\gamma = 0.9$. Figure 4 presents the overall structure of EMGCaps using dynamic routing.

## III. Results

### A. NinaPro DB5 Dataset



(a) Basic movements of the fingers (flexions and extensions).

(b) Isometric, isotonic hand configurations ("hand postures").

(c) Basic movements of the wrist.

(d) Grasping and functional movements.

Fig. 5. The 52 movements of interest (rest pose excluded). [26]

We train and evaluate the networks using the NinaPro DB5 dataset. NinaPro DB5 includes data from 10 intact subjects (8 males, 2 females; 10 right handed; age 28 ± 3.97 years) performing repetitively (6 repetitions) 53 different movements (including rest position) [27]. Figure 5 shows the different gestures. Compared to other datasets, the number of subjects is high, and some movements are very similar, which makes the classification task more challenging. Within this data acquisition system, two Thalmic Myo armbands measure the electromyographic activity. The Myo armband operates at 200 Hz frequency, reads 8 EMG sensors data and streams the data out to the recording device. Each repetition of the exercise lasts 5s, and it alternates with a 3s rest pose to prevent muscle exhaustion [28]. According to human response times the gestures executed by the participants can not exactly match those seen on the video. The generalized likelihood ratio algorithm and the algorithm based on the Lidierth threshold are used to detect the movements and correct incorrect labeling. A notch filter is used to drop frequencies at 50Hz in order to exclude power grid interference [27]. The public dataset includes filtered and synchronized data for each subject and exercises in MATLAB format [1]. We visualize the data on its 3 principle components using Principle Component Analysis (PCA) to provide insight into the relation between movements and sEMG signals. Figure 6 shows that is very challenging to differentiate between the movements due to the huge overlap of regions occupied by different gestures.

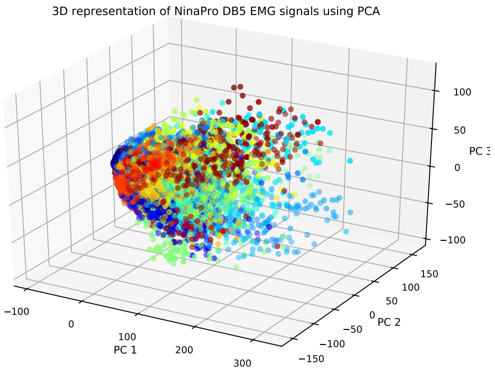


Fig. 6. 3D Representation of NinaPro DB5 EMG signals using PCA

### B. Model Optimization

EMGCaps classifier is trained and evaluated by optimizing loss w.r.t. the weights using the ADAM optimizer. The evaluation is based on the 2nd and 5th EMG repetitions. Additionally, we train 5 other classifiers, namely the SVM, RF, FCN, and CNN. We train our models using the aforementioned datasets: INTERSUBJECT, INTERSUBJECT_VAR, INTRASUBJECT, and INTRASUBJECT_VAR. We also try different feature vectors in order to conclude the best setting for EMG recognition. We discuss the training and the evaluation of EMGCaps using selected features for each dataset type. As shown in Fig. 7, the model is overfitted when using the INTRASUBJECT dataset and FOURIER features. Reducing the number of capsules, i.e. reducing the complexity of the network did not improve the learning. Hence, FOURIER features do not provide enough knowledge for a generalization. The best achievable accuracy on the validation set of INTRASUBJECT dataset is when using MULTIRMS features. Its training behavior is shown in Fig. 8. The best achievable accuracy of EMGCaps is on the INTRASUBJECT_VAR using MULTIRMS features. The loss function seems to have multiple local minima, which makes the training loss oscillating. Nevertheless, the accuracy is stable. When using the INTERSUBJECT dataset, the validation accuracy remains very low. This problem does not occur because of overfitting but because of the structure of the data. In fact, EMG structure differs from a subject to another. This makes the generalization over subjects very hard. Figure 10 shows the behavior of the training process. The same problem occurs when using the INTERSUBJECT_VAR dataset. The

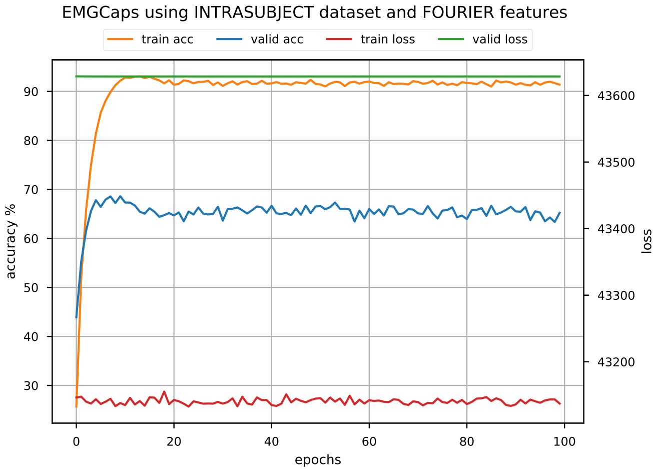


Fig. 7. Training and validation of EMGCaps on INTRASUBJECT dataset using FOURIER features.

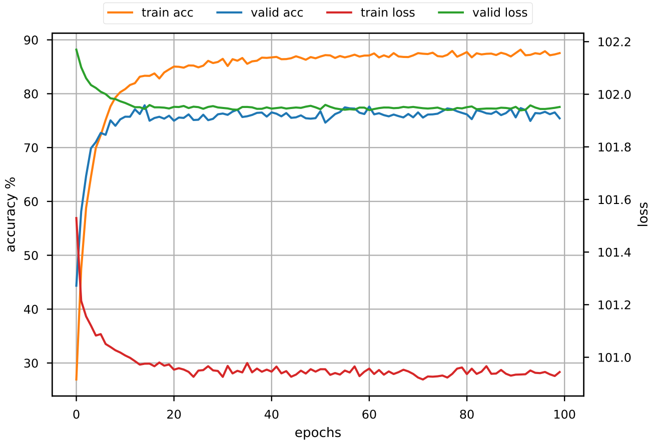


Fig. 8. Training and validation of EMGCaps on INTRASUBJECT dataset using MULTIRMS features.

[1] http://ninapro.hevs.ch/

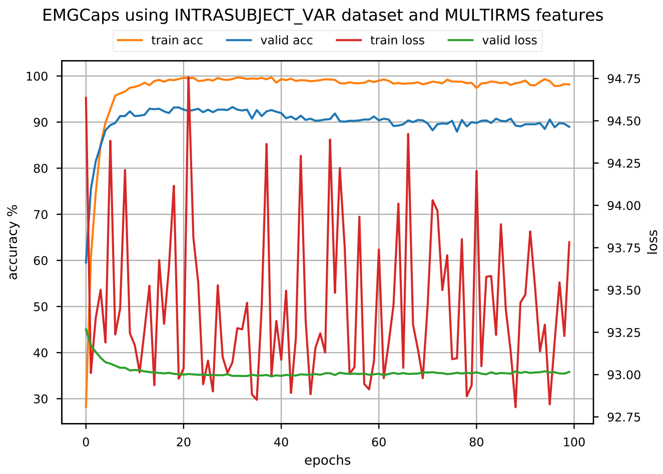


Fig. 9. Training and validation of EMGCaps on INTRASUBJECT_VAR dataset using MULTIRMS features.

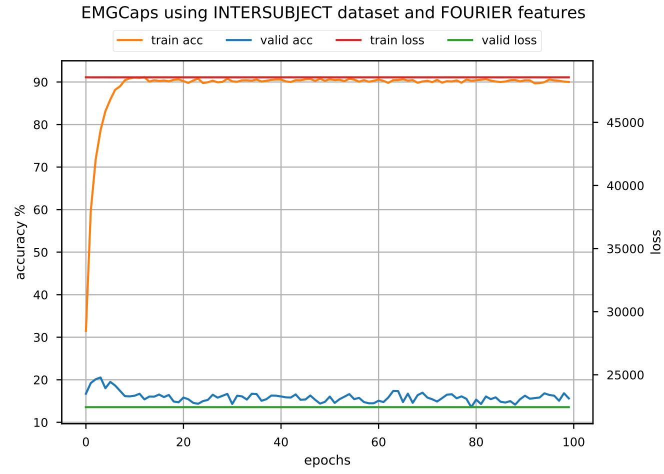


Fig. 10. Training and validation of EMGCaps on INTERSUBJECT dataset using FOURIER features.

validation accuracy is, however, slightly better, due to the larger input vector feature. Figure 11 shows the behavior of the corresponding training and validation processes.
As illustrated on the training curves, the learning using capsules is fast and the optimal solution could be achieved in less than 10 epochs.

We used Tensorflow modules for the development. Using the Adam optimizer with an initial learning rate of 0.001 and weight decay of 0.00001, we trained the network for 100 epochs. The models were trained on Nvidia GTX1060.

The source code are available on our Github repository [2].

### C. Comparison of different models

Different traditional and deep-learning-based models are trained on sEMG signals. Tables I, II, III and IV present the results of classifiers training and validation using the datasets INTRASUBJECT, INTRASUBJECT_VAR, INTERSUBJECT and INTERSUBJECT_VAR, respectively. Regardless of the type of the features, the validation accuracy on the INTRASUBJECT and INTRASUBJECT_VAR datasets are significantly higher than the other datasets. This is due to the high variation of EMG signal within different subjects [26].

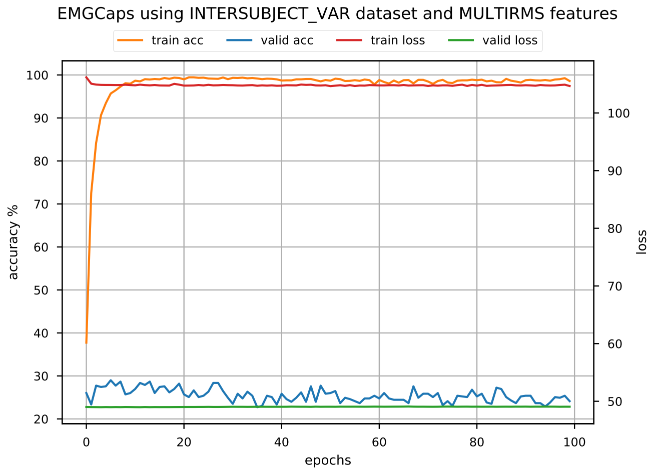


Fig. 11. Training and validation of EMGCaps on INTERSUBJECT_VAR dataset using MULTIRMS features.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON NINAPRO DB5 INTRASUBJECT DATASET USING DIFFERENT FEATURES

| Features | RF | SVM | FCN | CNN | EMGCaps |
|---|---|---|---|---|---|
| RMS | 66.56% | 52.06% | 67.67% | 55.54% | **76.04%** |
| HIST | 56.34% | 38.58% | 49.28% | 53.88% | **65.67%** |
| MULTIRMS | 65.29% | 50.93% | 57.26% | 61.47% | **77.85%** |
| PMRMS | 66.44% | 63.15% | 63.29% | 61.72% | **76.41%** |
| KMRMS | 54.4% | 25.57% | 38.73% | 39.65% | **67.47%** |
| FOURIER | 62.32% | 48.34% | 57.14% | 56.77% | **69.18%** |

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON NINAPRO DB5 INTRASUBJECT_VAR DATASET USING DIFFERENT FEATURES

| Features | RF | SVM | FCN | CNN | EMGCaps |
|---|---|---|---|---|---|
| RMS | 79.53% | 53.27% | 74.67% | 66.26% | **90.56%** |
| MULTIRMS | 50.93% | 66.82% | 48.69% | 81.21% | **93.27%** |
| PMRMS | 81.96% | 82.42% | 52.05% | 90.09% | **92.52%** |
| KMRMS | 64.67% | 23.92% | 15.88% | 39.65% | **85.04%** |

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON NINAPRO DB5 INTERSUBJECT DATASET USING DIFFERENT FEATURES

| Features | RF | SVM | FCN | CNN | EMGCaps |
|---|---|---|---|---|---|
| RMS | 15.00% | 14.8% | **17.42%** | 16.20% | 15.56% |
| HIST | 14.82% | 14.95% | 14.67% | 14.88% | **15.21%** |
| MULTIRMS | 15.97% | 15.03% | 15.51% | 15.49% | **17.57%** |
| PMRMS | 9.8% | **13.40%** | 11.93% | 10.97% | 11.78% |
| KMRMS | 11.85% | 11.07% | 14.09% | 13.40% | **15.00%** |
| FOURIER | 16.02% | 14.95% | 16.98% | 17.42% | **20.54%** |

[2] https://github.com/ussaema/EMGCaps

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON NINAPRO DB5 INTERSUBJECT_VAR DATASET USING DIFFERENT FEATURES

| Features | RF | SVM | FCN | CNN | EMGCaps |
|---|---|---|---|---|---|
| RMS | 15.20% | 15.67% | 19.27% | 19.59% | **20.84%** |
| MULTIRMS | 22.88% | 20.53% | 27.27% | 27.11% | **28.99%** |
| PMRMS | 18.18% | 23.19% | 20.68% | 20.06% | **23.51%** |
| KMRMS | 14.89% | 10.81% | 12.06% | 18.02% | **19.74%** |

Our network outperforms all other current models (Table V) when using the INTRASUBJECT_VAR and MULTIRMS features. As shown on Table V, our model improved 3.27% over the state-of-the-art model on NinaPro DB5 with 53 movements.

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT METHODS ON NINAPRO DB5 DATASET

| Model | Accuracy |
|---|---|
| Atzori et al.(2012) [26] - LS-SVM | 79.7% |
| Wei et al.(2019) [25] - Multi-View CNN | 90.0% |
| Chen et al. (2020) [5] - EMGNet | 69.62% |
| Shen et al. (2019) [29] - Ensemble CNN | 72.09% |
| EMGCaps (our model) | **93.27%** |

## IV. CONCLUSION

This paper proposed a novel architecture focused on Capsule Network, EMGCaps that had been designed to identify EMG signals. Our results showed that EMGCaps outperforms other state-of-the-art approaches. One significant benefit of EMGCaps is its ability to represent a high dimensional feature of EMG signals by a low dimensional vector (capsule). We further evaluated EMGCaps performance in an EMG classification and compared it with other EMG recognition approaches. Experiments showed that, with capsule network, we outperformed the state-of-the-art performance and achieved an accuracy of 93.27%. The potential EMGCaps network can be further extended to an end-to-end system, where feature engineering is not required anymore. This high precision is related to an offline analysis, but the latency induced by the neural network is still a problem for a real-time application. Hence, a reduction of the size of the segmentation and optimization regarding the number of learnable parameters is to consider. A Hybrid system based on ensemble learning could also improve the accuracy of the system.

## REFERENCES

[1] R. Alam, S. R. Rhivu, and M. A. Haque, "Improved gesture recognition using deep neural networks on semg," in *2018 International Conference on Engineering, Applied Sciences, and Technology (ICEAST)*, 2018, pp. 1–4.

[2] S. Zhou, K. Yin, Z. Liu, F. Fei, and J. Guo, "semg-based hand motion recognition by means of multi-class adaboost algorithm," 12 2017, pp. 1056–1061.

[3] Y. Wu, B. Zheng, and Y. Zhao, "Dynamic gesture recognition based on lstm-cnn," in *2018 Chinese Automation Congress (CAC)*, 2018, pp. 2446–2450.

[4] T. Ruan, K. Yin, and S. Zhou, "Convolutional neural network based human movement recognition using surface electromyography," in *2018 IEEE 1st International Conference on Micro/Nano Sensors for AI, Healthcare, and Robotics (NSENS)*, 2018, pp. 68–72.

[5] C.-Y. Chen, Fu, D. Yu, Li, and Y. Zheng, "Hand gesture recognition using compact cnn via surface electromyography signals," *Sensors*, vol. 20, p. 672, 01 2020.

[6] A. Chaiyaroj, P. Sri-Iesaranusorn, C. Buekban, S. Dumnin, C. Thanawattano, and D. Surangsrirat, "Deep neural network approach for hand, wrist, grasping and functional movements classification using low-cost semg sensors," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019, pp. 1443–1448.

[7] M. Simão, P. Neto, and O. Gibaru, "Emg-based online classification of gestures with recurrent neural networks," *Pattern Recognition Letters*, vol. 128, 08 2019.

[8] M. Atzori, A. Gijsberts, B. Caputo, and H. Müller, "Natural control capabilities of robotic hands by hand amputated subjects," vol. 2014, 08 2014.

[9] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for emg signal classification," *Expert Systems with Applications*, vol. 39, p. 7420–7431, 06 2012.

[10] D. Huang and B. Chen, "Surface emg decoding for hand gestures based on spectrogram and cnn-lstm," in *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, 2019, pp. 123–126.

[11] D. Toledo Pérez, J. Rodriguez, R. Gómez Loenzo, and J. Jauregui, "Support vector machine-based emg signal classification techniques: A review," *Applied Sciences*, vol. 9, p. 4402, 10 2019.

[12] I. Kuzborskij, A. Gijsberts, and B. Caputo, "On the challenge of classifying 52 hand movements from surface electromyography," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2012, pp. 4931–7, 08 2012.

[13] Ninapro repository.

[14] Y. He, O. Fukuda, N. bu, H. Okumura, and N. Yamaguchi, "Surface emg pattern recognition using long short-term memory combined with multilayer perceptron," vol. 2018, 07 2018, pp. 5636–5639.

[15] Y. Hu, Y. Wong, W. Wei, Y. Du, M. Kankanhalli, and W. Geng, "A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition," *PLOS ONE*, vol. 13, p. e0206049, 10 2018.

[16] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *CoRR*, vol. abs/1710.09829, 2017. [Online]. Available: http://arxiv.org/abs/1710.09829

[17] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.

[18] M. Lidierth, "A computer based method for automated measurement of the periods of muscular activity from an emg and its application to locomotor emgs," *Electroencephalography and Clinical Neurophysiology*, vol. 64, no. 4, pp. 378 – 380, 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/001346948690163X

[19] S. Samui, "An experimental study on upper limb position invariant emg signal classification based on deep neural network," *Biomedical Signal Processing and Control*, vol. 55, 01 2020.

[20] M. Bigliassi, P. Scalassara, T. Kanthack, T. Abrao, A. Moraes, and L. Altimari, "Fourier and wavelet spectral analysis of emg signals in 1-km cycling time-trial," *Applied Mathematics*, vol. 5, pp. 1878–1886, 07 2014.

[21] Cristianini, Nello, and J. Shawe-Taylor, "An introduction to support vector machines and other kernel-based learning methods. repr," *Introduction to Support Vector Machines and other Kernel-Based Learning Methods*, vol. 22, 01 2001.

[22] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. USA: Wiley-Interscience, 2000.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[25] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, and W. Geng, “Surface-electromyography-based gesture recognition by multi-view deep learning,” *IEEE Transactions on Biomedical Engineering*, vol. PP, pp. 1–1, 02 2019.

[26] M. Atzori, A. Gijsberts, S. Elsig, A.-G. Mittaz Hager, O. Deriaz, P. van der Smagt, C. Castellini, B. Caputo, and H. Müller, “Building the ninapro database: A resource for the biorobotics community,” 12 2012.

[27] S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Müller, and M. Atzori, “Comparison of six electromyography acquisition setups on hand movement classification tasks,” *PLoS ONE*, vol. 12, 2017.

[28] M. Atzori and H. Müller, “The ninapro database: a resource for semg naturally controlled robotic hand prosthetics,” 08 2015.

[29] S. Shen, K. Gu, X. Chen, M. Yang, and R. Wang, “Movements classification of multi-channel semg based on cnn and stacking ensemble learning,” *IEEE Access*, vol. 7, pp. 137 489–137 500, 2019.