Intelligent Systems with Applications 22 (2024) 200352

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

CNN-LSTM and post-processing for EMG-based hand gesture recognition

Lorena Isabel Barona López, Francis M. Ferri, Jonathan Zea, Ángel Leonardo Valdivieso Caraguay, Marco E. Benalcázar *

Artificial Intelligence and Computer Vision Research Laboratory "Alan Turing", Departamento de Informática y Ciencias de la Computación (DICC), Escuela Politécnica Nacional, Quito, 170525, Ecuador

ARTICLE INFO

Keywords:
Hand gesture recognition
EMG-EPN-612
Post-processing
Convolutional neural networks
Long short-term memory
EMG
Spectrogram

ABSTRACT

Hand Gesture Recognition (HGR) using electromyography (EMG) signals is a challenging problem due to the variability and noise in the signals across individuals. This study addresses this challenge by examining the effect of incorporating a post-processing algorithm, which filters the sequence of predictions and removes spurious labels, on the performance of a HGR model based on spectrograms and Convolutional Neural Networks (CNN). The study also compares CNN vs CNN-LSTM to assess the influence of the memory cells on the model. The EMG-EPN-612 dataset, which contains measurements of EMG signals for 5 hand gestures from 612 subjects, was used for training and testing. The results showed that the post-processing algorithm increased the recognition accuracy by 41.86% for the CNN model and 24.77% for the CNN-LSTM model. The inclusion of the memory cells increased accuracy by 3.29%, but at the cost of 53 times more learnables. The CNN-LSTM model with post-processing achieved a mean recognition accuracy of 90.55% (SD = 9.45%). These findings suggest new paths for research in HGR architectures beyond the traditional focus on the classification and feature extraction stages. For reproducibility purposes, we made publicly available the source code in Github.

1. Introduction

Hand Gesture Recognition (HGR) involves recognizing the specific class of hand gesture from a predefined set of movements and identifying the exact moment when the gesture is performed (Benalcázar et al., 2017). HGR has a wide range of applications, including bionics, video games, sign language recognition, and medicine (Luh et al., 2015, Rafiee et al., 2011, Riillo et al., 2014, Saggio et al., 2020, Saponas et al., 2009, Sathiyanarayanan & Rajan, 2016, Shi et al., 2018, Zhu & Yuan, 2014). One approach for HGR is the use of surface electromyography (EMG) signals. EMG signals are electrical signals that are generated by the contraction of skeletal muscles (Farina et al., 2014, Reaz et al., 2006, Rodriguez-Falces et al., 2012). These signals are recorded by placing small electrodes on the skin overlying the muscles being studied (non-invasive) or directly inserting a needle electrode into the muscle tissue (invasive). Invasive methods, such as intramuscular EMG or implants (Hahne et al., 2016), provide more accurate measurement of the EMG signal but are impractical for general use. Non-invasive methods, such as surface electromyography (referred to as EMG in this work), are less precise but more practical for general use. Surface electromyograms are bioelectric signals that provide information on the intensity and duration of muscle activation. These signals are obtained by measuring the aggregate activity of motor units from electrodes placed on the skin over the muscle of interest (Gerdle et al., 1999).

1.1. Structure of a HGR model

The structure of a Hand Gesture Recognition (HGR) model typically consists of five stages: data acquisition, pre-processing, feature extraction, classification, and post-processing (Barona López et al., 2020, Benalcázar et al., 2020). In the data acquisition stage, various technologies such as vision sensors (Zhu & Yuan, 2014), Inertial Measurement Units (IMU) (Moschetti et al., 2016), sensor gloves (Jiménez et al., 2017, Pallotti et al., 2021), and EMG (Benalcázar et al., 2020, Benatti et al., 2017, Pallotti et al., 2021, Shi et al., 2018) are utilized. The pre-processing stage involves techniques such as rectification and filtering (Benalcázar et al., 2018, 2017, Neto & Christou, 2010). In the feature extraction stage, common methods include Mean Absolute Value (MAV), Root Mean Square (RMS), standard deviation (SD), variance (VAR), as well as automatic feature extraction methods like CNNs (Chen et al., 2020, Shi

* Corresponding author.
E-mail addresses: lorena.barona@epn.edu.ec (L.I.B. Barona López), jonathan.a.zea@ieee.org (J. Zea), angel.valdivieso@epn.edu.ec (Á.L. Valdivieso Caraguay), marco.benalcazar@epn.edu.ec (M.E. Benalcázar).

https://doi.org/10.1016/j.iswa.2024.200352
Received 6 December 2023; Received in revised form 14 February 2024; Accepted 1 March 2024

Available online 7 March 2024

et al., 2018, Ullah et al., 2020, Wang et al., 2017). For the classification stage, various Machine Learning techniques are employed, including k-Nearest Neighbors (kNN) (Jiménez et al., 2017), Support Vector Machines (SVM) (Saha et al., 2015), Random Forests (Joshi et al., 2017, Sohn et al., 2016), and feed-forward neural networks (Tsironi et al., 2017). Finally, in the post-processing stage, common methods involve filtering consecutive repetitions, removing different labels from the mode gesture, applying a threshold, or utilizing velocity ramps (Barona López et al., 2020). It's important to note that HGR systems must work in real-time. In that respect, according to (Benatti et al., 2017) the computing time must be less than 300 ms, or below 100 ms according to (Farrell & Weir, 2007). Thus, when analyzing HGR systems, it is important to consider the trade-off between accuracy and processing time.

1.2. Types of HGR models

There are two types of Hand Gesture Recognition (HGR) models: general models and user-specific (or individual) models. A general model is trained with data from multiple individuals and is designed to recognize samples from any new person. On the other hand, user-specific models are trained with data from a specific individual and can only process samples from that individual. These models tend to be more precise, but require training a model for each new subject. Among the best accuracy results for user-specific models are 98.33% (Seok et al., 2018), 95.32% (Benalcázar et al., 2020), and 94.20% (Barona López et al., 2020). Some general models have achieved results such as 87.53% (Jaramillo-Yanez et al., 2019), 85.08% (Chung & Benalcázar, 2019), 83.5% (Song et al., 2018), and 80.31% (Barona López et al., 2020). However, it is important to note that comparing these results should be done with caution. As Atzori et al. pointed out, it is fundamental to compare accuracy only when the number of classes is comparable (Atzori et al., 2016). Even more, works in the literature use different EMG sensors, electrode placements, sets of gestures, and number of individuals, among other factors. For example, in (Seok et al., 2018) 3 gestures are classified and tested over 1 subject, in (Song et al., 2018) 6 gestures are classified over 6 subjects, in (Jaramillo-Yanez et al., 2019) 5 gestures are classified over 60 subjects, and in (Vásconez et al., 2022) 11 gestures are classified over 85 subjects. As can be observed, the common factor in the literature is the relatively small sample size on which HGR models are evaluated. Results obtained with a small sample size can not be generalized to a variety of people, which is especially important in the case of EMG signals, where the effect of inter-personal variability is well known (Guidetti et al., 1996, Hug et al., 2010, Wojtara et al., 2014). To obtain statistically significant results, this work uses a large dataset of 612 subjects, with 50% of the subjects being used for training and 50% for testing (hold-out). Using a hold-out method allows us to reduce biases from other types of model validation techniques such as cross-validation (Varma & Simon, 2006).

The development of general models for Hand Gesture Recognition (HGR) has not been extensively studied in the literature. Many researchers have primarily focused on the development of user-specific models. However, the inherent variations in the EMG signal, along with the challenges posed by inter and intra-personal variability in the distribution of EMG data (Rodriguez-Falces et al., 2012) make it challenging to extrapolate user-specific architectures to general models. User-specific models are commonly trained with traditional (or shallow) Machine Learning algorithms, and cannot be solved with Deep Learning as it would be infeasible to obtain such a large dataset from a single person. Additionally, shallow ML methods often require manual feature selection which can be tedious, time-consuming, and inflexible (Janiesch et al., 2021). Deep Learning, on the other hand, can be a viable alternative for general models as it can be trained with more data and can adapt to different subjects. Convolutional Neural Networks (CNNs) can be used for automatic feature extraction, which has the advantage of automated feature learning to extract discriminative feature representations with minimal human effort (Janiesch et al., 2021). Long Short-Term Memory Networks (LSTMs) can also be used to learn from sequences by modeling time dependencies and forming a memory (Janiesch et al., 2021).

In this study, we explore the potential of spectrograms combined with CNNs for Hand Gesture Recognition. This innovative approach, which has already demonstrated promising outcomes in the realm of speech recognition (Kingsbury et al., 1998, Satt et al., 2017), is now being applied to HGR. Our research addresses a significant gap in the current literature by examining the impact of post-processing algorithms on the performance of HGR models. Through this investigation, we aim to advance the field of HGR and pave the way for future research directions.

The remainder of this paper is organized as follows. Section 2 presents a comprehensive study of the related works in the field. Section 3 provides a detailed description of the proposed HGR model. In Section 4, the experimental process is described including model tuning, validation, and evaluation in terms of classification and recognition accuracy. Finally in Section 5, the conclusions of this research are presented, along with suggestions for future work in the field.

2. Related works

Several studies in the literature have applied CNNs to the HGR problem (Wang, Fu, et al., 2023, Wang, Zhang, et al., 2023, Zhang & Zhang, 2022). For instance, Atzori et al. (2016) used a simple CNN architecture to classify approximately 50 gestures of one of the Ninapro databases, and found that the more influential factors in the results were the shape of the first layer, the initial weights, the data augmentation procedures and the learning rate. For his part, Yang et al. (2019) explored an approach of using the raw information of the EMG signals as input of the CNNs, comparing a time domain and a frequency domain representation using 2 public dataset, with good results only in the dataset CapgMyo-Dba. In the study conducted by Asif et al. (2020), a comprehensive analysis was performed to understand the effects of various hyperparameters, such as learning rate and number of epochs, on the accuracy of the model. The study concluded that the selection of these parameters is crucial to the overall efficiency of the network (Asif et al., 2020). Chamberland et al. (2023) developed EMaGer, a flexible, and extensible 64-channel HD-EMG sensor that can adapt to various forearm sizes. In addition, they implemented a CNN-based model that is robust to shifts. However, it is not robust against longitudinal shifts. This work additionally underscores the challenges associated with finer motions beyond gesture classification. Also, Shanmuganathan et al. (2020) proposes a method that employs R-CNN and wavelet feature extraction for hand gesture recognition using EMG signals. The model achieved high results, with an accuracy rate of 96.48%. The dataset used for this study comprises ten subjects performing four distinct gestures. However, the requirement for electrodes to be placed in specific positions poses a practicality challenge for real-world applications. In Sunil et al. (2023) was utilized a CNN to extract spatial features in parallel with an LSTM that extracts temporal features. This approach achieved a remarkable accuracy of 98.01% on NinaPro DB2. However, the complexity of the model may not allow real-time performance. Similarly, Yang et al. (2021) proposed a two-stage framework for HGR using MVMD and separable CNN, that may not be suitable for real-time applications due to the computational cost of the neural network. Other works have opted or included LSTM networks. For instance, in (Ghislieri et al., 2021) was demonstrated the feasibility of using LSTM networks for the accurate detection of muscle activity in EMG signals without the need for background-noise estimations. Another work (Cai & Zhu, 2021) used CNNs with LSTM for extracting spatial and temporal features, however, it still requires professional domain knowledge as it used additionally handcrafted EMG features. A similar study by Li et al. (Li & Langari, 2022) proposed a CNN-LSTM network to classify 5 dynamic hand gestures in 5 different limb positions, and found out that the gesture classification accuracy of the CNN-LSTM is influenced by the dynamic gesture

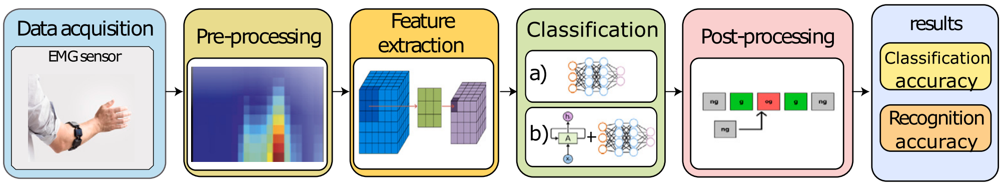


Fig. 1. The stages of a Hand Gesture Recognition model. a) CNN, b) CNN-LSTM.

selection, the inter-personal variability and the limb position. In (Zhao et al., 2022) used Fuzzy logic with the inclusion of LSTM networks to take into account the temporal correlation of the EMG signal in the entire window, and were able to classify up to 4 hand gestures. For his part, Antonius et al. (Antonius & Tjahyadi, 2021) combined CNNs with a recurrent neural network, similar to LSTM, with good results, but only on a very limited number of simple gestures. Similarly, another study (Toro-Ossaba et al., 2022) developed a LSTM network for HGR of 5 gestures only with 4 EMG channels, but obtained an accuracy reduction of 12% during real-time testing allegedly because of small changes in the armband position and noise. In (Karnam et al., 2022), Karman et al. combined CNN with a bidirectional LSTM to learn both inter-channel and bidirectional temporal information in an end-to-end manner. Other techniques such as transfer learning, cross-domain or sensor fusion have been explored. For instance, Bird et al. (Bird et al., 2020) analysed a cross-domain transfer learning approach with CNN between electroencephalographic EEG and EMG signals representing the biological waves as images, obtaining good preliminary results. Also, Tryon et al. (Tryon & Trejos, 2021) fused EEG and EMG to classify one gesture with CNN, with the objective to find relationship not yet found when using manual or automatic feature extraction from a unique source, but their results showed a small improvement gain when using EEG/EMG with respect of only EMG. For his part, Chen et al. (Chen et al., 2020) applied transfer learning from a base CNN architecture, but required 2 new samples per gesture for fine tuning.

2.1. Evaluation metrics

When evaluating the performance of Hand Gesture Recognition (HGR) models, it is important to consider the metrics used. While traditional classification metrics such as precision (Tam et al., 2021), F1-score (Jaramillo-Yánez et al., 2020), or classification accuracy (Cai & Zhu, 2021) are commonly used in the literature, they may not fully capture the performance of a model in real-time scenarios. In this work, we propose the use of recognition accuracy as an additional evaluation metric. Recognition accuracy measures the alignment of the predicted class vector with the ground truth (Zea & Benalcázar, 2019) and has been shown to be a more appropriate metric for evaluating HGR models. The metric that has traditionally been used is classification accuracy (Weir, 2018). However, this metric can be problematic for HGR systems. When evaluating the complete EMG signal, it does not reflect real-time operation and is impractical for most applications. On the other hand, segmenting the EMG signal and evaluating each window separately may simulate real-time operation, but it does not account for the fact that any incorrect classification in a middle window could be disastrous in a real-time application. For example, imagine a drone that receives a sudden command to land or a robotic arm that suddenly drops an object. To address this issue, post-processing algorithms have been developed to have a smooth transition between the *noGesture* state and the predicted hand gesture without spurious predictions. Former research has shown that while classification accuracy may seem high, when analyzing the whole sequence of predictions, the recognition accuracy is extremely low. Using recognition accuracy allows us to measure this and the importance of post-processing algorithms.

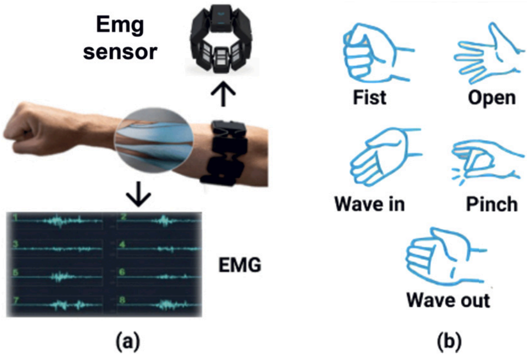


Fig. 2. a) EMG sensor and EMG signals example, b) the 5 hand gestures of the EMG-EPN-612 dataset.

3. Material and methods

In this section, we describe the proposed architecture for tackling the HGR problem based on EMG signals, as illustrated in Fig. 1. We implemented and compared two Deep Learning architectures: a CNN and a CNN-LSTM. These architectures are composed of five stages: data acquisition, pre-processing, feature extraction, classification, and post-processing. The only distinction between the two models is the addition of an LSTM layer between the feature extraction and classification stages in the CNN-LSTM model. The pseudocode of the proposed method is presented in Algorithm 1. In the following subsections, we will provide a detailed explanation of each stage.

Algorithm 1 Real-time HGR model.

```
Input: $i, E \in \mathbb{R}^{[t \times 8]}$                          ▷ $E$: EMG signal
Output: $i, S_i$                                    ▷ $S$: vector of predictions
$W \leftarrow WINDOWING(E, 300)$                 ▷ $W$: portion of the EMG signal
$c \leftarrow 1$
while $c <= 8$ do                                        ▷ loop by channel
    $W_c \leftarrow$ channel c of W
    $\psi \leftarrow \Psi(|W_c|)$             ▷ $\Psi$: rectification and filtering function
    $M_c \leftarrow STFT(\psi)$          ▷ $M_c$: spectrogram of channel $c$ using STFT
    $c \leftarrow c + 1$
end while
$\Lambda \leftarrow (M_1, M_2, ..., M_8)$                     ▷ $\Lambda \in \mathbb{R}^{[13 \times 24 \times 8]}$
$Y_i \leftarrow INFERENCE(\Lambda, model)$       ▷ model is either CNN or CNN-LSTM
$S_i \leftarrow POSTPROCESSING(Y_i)$                       ▷ View Algorithm 2
$i \leftarrow i + 1$
```

3.1. Data acquisition

In this study, we utilized the EMG-EPN-612 dataset (Benalcazar et al., 2020) for our experimentation. This dataset is intended for use in the development and benchmarking of hand gesture recognition models based on EMG signals. This dataset contains measurements of EMG and IMU signals for 5 hand gestures. Fig. 2 depicts the EMG sensor along with the 5 hand gestures used in this work. The device used for the acquisition is equipped with eight EMG dry surface electrodes at 200Hz. In addition to this, it incorporates a 9-axes inertial measurement unit (IMU) that measures linear, angular motion and orientation, at 50Hz.

Table 1
Characteristics of the EMG-EPN-612 dataset.

Parameter	Value
Num. subjects	612
Male | female percentage	66% | 34%
Num. training subjects	306
Num. testing subjects	306
NUm. channels	8
EMG sample frequency	200 Hz
IMU sample frequency	50 Hz
ADC resolution	8 bits
Num. gestures	5
Num. samples per class	50
EMG samples per subject	300
Hold-out	50%
Time recording per sample	5 seconds
Classes	6 (*wave in, wave out, fist, open, pinch* and *noGesture*)

The device utilizes Bluetooth Low Energy (BLE) for wireless connectivity. The characteristics of this dataset are summarized in Table 1. The data acquisition process was executed indoors, with an average room temperature of 24 °C. The dataset is divided into two subsets, with one subset containing 306 subjects for training and validation, and the other containing 306 subjects for testing. The EMG samples of the training and validation subset have the corresponding class and ground truth information freely available. In contrast, the ground truth of half of the samples in the testing subset is not publicly accessible. The utilization of the online testing platform is encouraged to ensure the prevention of overfitting or manipulation of the testing accuracy results (Barona López et al., 2020). The online evaluation platform associated with the dataset allows for the comparison of different models' performance in terms of classification and recognition accuracy. This can facilitate advancements in the field of gesture recognition, contributing to applications such as human-computer interaction, prosthetics, and more. The technical validation of the quality and reliability of the dataset consisted of sample size calculation, a channel cross-correlation analysis, and an outlier score computation.

3.2. Data feeding and pre-processing

An EMG sample is denoted as a matrix $\mathbf{E} = (\mathbf{e}(1), ..., \mathbf{e}(t))^T \in \mathbb{R}^{[L\times 8]}$. Here, $t = \{1, 2, ..., L\}$ represents a discrete time instant and $L \in \mathbb{Z}^+$ is the number of sample points in the EMG signal. The vector $\mathbf{e}(t) = (e_1, ...e_8) \in \mathbb{R}^8$ at time instant t is composed of the measurements from the 8 channels of the device. Each measurement $e_c \in \mathbb{R}$ corresponds to the cth channel and falls within the range [-1, 1]. During real-time execution, L denotes the size of the data buffer streamed by the device. However, during training, $\mathbf{E}$ represents an EMG sample from the EMG-EPN-612 dataset. In this dataset, the average length of the samples is approximately $L \approx 1000$.

We employed a sliding window approach to process segments of the EMG signal consecutively. A window width of $|W| = 300$ sample points was selected experimentally (equivalent to a portion of 1.5 seconds of the signal). The portion of the EMG signal viewed through a window i is denoted with the matrix $\mathbf{W}(i) = (\mathbf{e}(t_i - 299), ..., \mathbf{e}(t_i))^T \in \mathbb{R}^{300\times 8}$, where t_i is the last time instant of the corresponding ith window. In Algorithm 1, $\mathbf{W}(i)$ is obtained in line 1. To simulate online scenarios, the models are fed with consecutive windows one at the time, (i.e., $t_1 = |W|, t_2 = |W| + s, t_3 = |W| + 2\times s, ..., t_i = |W| + (i-1)\times s$) where s is the distance between consecutive windows (a.k.a. stride), view Fig. 3.a. In the validation experiments, we compared the performance of two different values of stride, $s = 15$ and $s = 30$, and selected $s = 30$.

We performed initial processing on the EMG signals by rectifying the matrix $\mathbf{W}(i)$ using the absolute value function $|\mathbf{W}(i)|$. Subsequently, we applied a digital low-pass Butterworth filter, denoted as Ψ. The purpose of this filtering step is to attenuate the signal and minimize noise interference. The filter Ψ is designed with an order of 5 and a cutoff frequency of 10 Hz. These parameter were selected to achieve a balance between the number of spectrograms and their temporal resolution. This pre-processing methodology aims to work with the envelope of the EMG signals. According to the mathematical model by Shwedyk et al. (1977), an EMG signal can be modeled as Gaussian noise modulated by a second low-frequency Gaussian process (envelope) that encodes the movement activity.

At a given time instant t, by applying the filter channel-wise, we obtained the matrix $\psi = \Psi(|\mathbf{W}(i)|)$. In Algorithm 1, ψ is obtained in line 5. Fig. 3.b illustrates this rectification and filtering procedure. Next, we calculate the spectrogram of the filtered and rectified matrix ψ. The spectrogram generates a time-frequency representation of the input window by dividing it into segments and applying the Discrete Fourier Transform (DFT) on each segment. We set the segment size to 24 sample points and the number of overlapping samples to 12 points, view Fig. 3.c. The frequency range is set to $[0 - 12]$ Hz, which determines the temporal and frequency resolution of the feature maps generated by the spectrogram calculation. The DFT is applied to each segment of the window, creating the matrix $\mathbf{X} \in \mathbb{C}^{13\times 24}$, Fig. 3.d. The spectrogram of the cth channel is then calculated as $\mathbf{M}_c = |\mathbf{X}|^2$, as shown in Fig. 3.e. In Algorithm 1, $\mathbf{M}_c$ is obtained in line 6. For each channel of the EMG, we perform the aforementioned calculation, leading to the formation of a tensor $\mathbf{\Lambda} = (\mathbf{M}_1, \mathbf{M}_2, ..., \mathbf{M}_8) \in \mathbb{R}^{[13\times 24\times 8]}$. In Algorithm 1, $\mathbf{\Lambda}$ is obtained in line 9. This tensor serves as the input for the feature extraction stage and is depicted in Fig. 3.g.

3.3. Feature extraction

We utilize CNNs as a feature extraction method (LeCun et al., 2015), that extracts time-frequency domain features from the spectrograms. The proposed CNN approach incorporates multiple blocks of parallel convolutions and max-pooling, drawing inspiration from the "Inception modules" employed in GoogLeNet (Szegedy et al., 2015). The parallel convolution block is depicted in Fig. 4. This block includes different convolution filters sizes ($1\times 1, 3\times 3$ and 5×5) that allows the network to extract a wide range and diverse types of features. The output size of the parallel convolution block is $[13\times 24\times 72]$.

The structure of the feature extraction stage comprises of 6 blocks of parallel convolutions, along with 2 residual connections. The use of residual connections is essential to avoid the problem of vanishing/exploding gradients, and also enables faster training (He et al., 2016). In the proposed feature extraction module, a residual connection is established between the input of block 2 and the output of block 3. Additionally, a second residual connection connects the input of block 4 with the output of block 5. The cross-channel normalization after block 6 has a channel window size of 5. This configuration is demonstrated in Fig. 5.

3.4. Classification

In this study, we conducted a comparison between two different approaches for classification. The first approach is the CNN model, illustrated in Fig. 6.a. In this model, the feature maps $\mathbf{\Lambda}$ obtained from the feature extraction stage are flattened and passed through a fully connected layer consisting of 6 neurons, corresponding to the number of classes. The output of the fully connected layer is then processed by the softmax activation function to calculate the probabilities of belonging to each class. The resulting class output is determined by selecting the class with the highest probability, provided it exceeds a threshold of $T = 50\%$. Otherwise, the class *noGesture* is predicted. The second model is a CNN-LSTM model, depicted in Fig. 6.b. This model includes the sequence folding and sequence unfolding layers required by the recurrent network. The sequence folding layer is used to process the elements that are in a sequence one by one in order to apply the different convolution and pooling operations to each element of the sequence indepen-

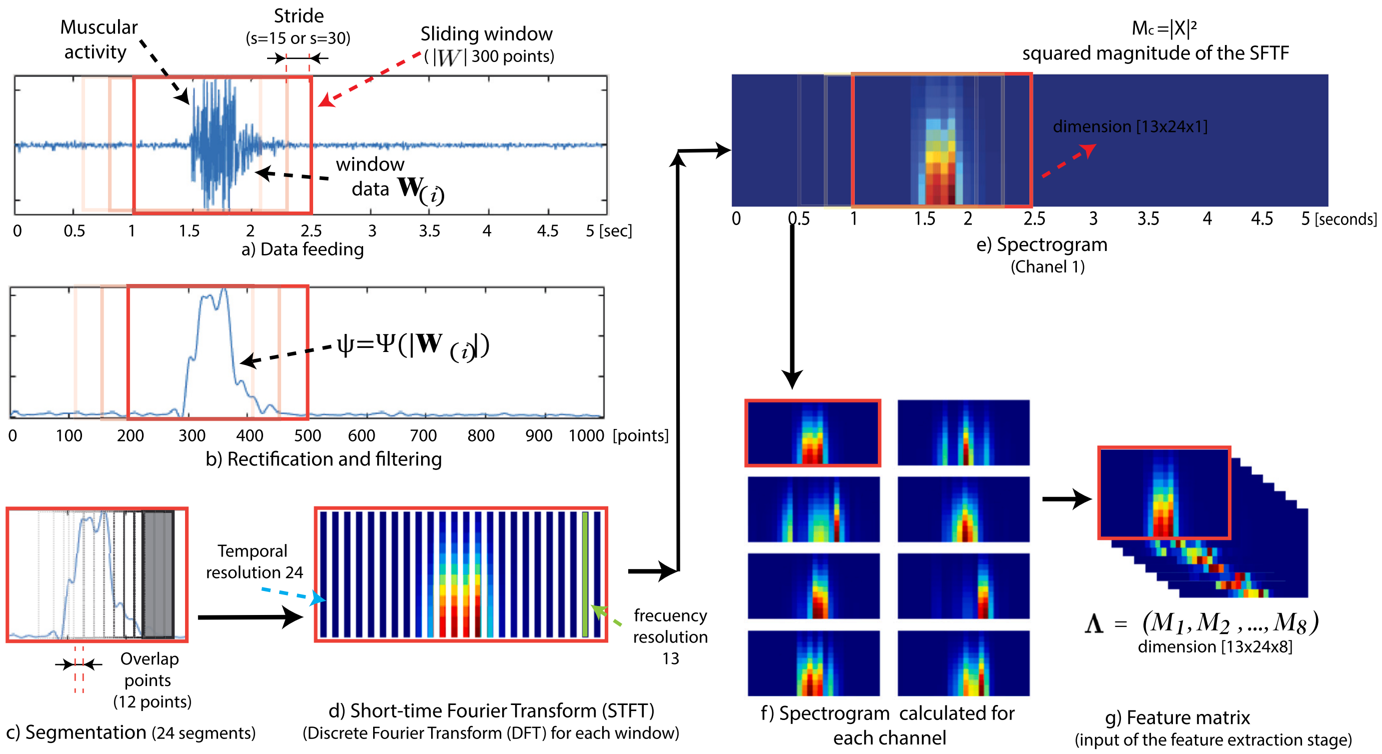

Muscular activity
Stride (s=15 or s=30)
Sliding window (|W| 300 points)
window data $\mathbf{W}_{(i)}$
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 [sec]
a) Data feeding
$\psi=\Psi(|\mathbf{W}_{(i)}|)$
0 100 200 300 400 500 600 700 800 900 1000 [points]
b) Rectification and filtering
Temporal resolution 24
Overlap points (12 points)
c) Segmentation (24 segments)
frecuency resolution 13
d) Short-time Fourier Transform (STFT)
(Discrete Fourier Transform (DFT) for each window)
Mc=|X|²
squared magnitude of the SFTF
dimension [13x24x1]
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 [seconds]
e) Spectrogram
(Chanel 1)
f) Spectrogram calculated for each channel
$\Lambda = (M_1, M_2, ..., M_8)$
dimension [13x24x8]
g) Feature matrix
(input of the feature extraction stage)

Fig. 3. Pre-processing and spectrogram generation.

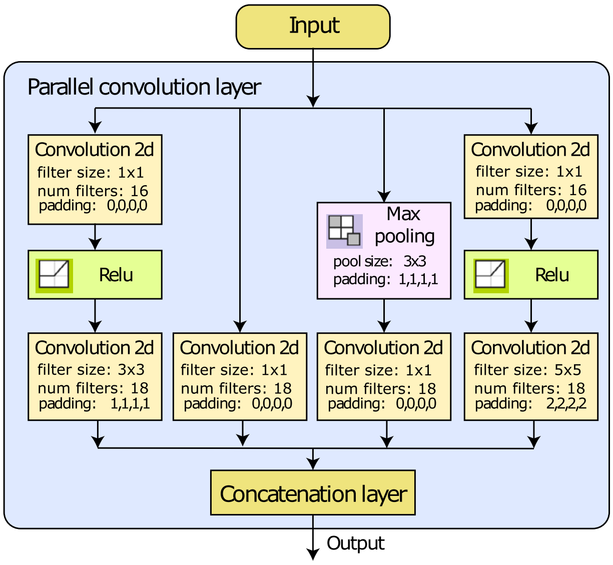


Fig. 4. The structure of the parallel convolution block.

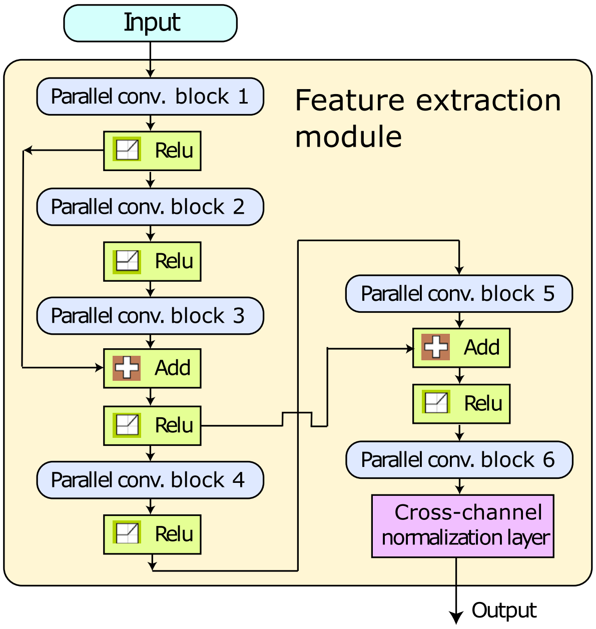


Fig. 5. Structure of the feature extraction module formed with 6 parallel convolution blocks and 2 residual connections.

dently (Hochreiter & Schmidhuber, 1997). We utilized an LSTM layer, known for its ability to capture long-term dependencies in sequence data (Hochreiter & Schmidhuber, 1997). For the LSTM layer, we selected 128 hidden units. The output of the LSTM layer is then passed to a fully connected layer with 6 neurons, followed by softmax activation and a classification layer. This process yields the resulting class output with the highest probability. To determine the final class Y_i, we apply a threshold of $T = 50\%$. Feature extraction and classification are carried on by the inference step of the neural networks, represented in line 10 of Algorithm 1. For ease of replication, the code for this work is publicly available at https://github.com/laboratorioAI/2023-HGR5-CNN_LSTM

3.5. Post-processing

The models described are applied on the adjoining sliding windows of each EMG, producing a sequence of class predictions, where the output of the neural network Y_i is the ith label on the sequence. The post-processing stage is intended to improve the recognition accuracy by removing and filtering spurious predictions in the sequence. The sequence obtained after post-processing is denoted as S, line 11 of Algorithm 1. For further clarity, the pseudocode of the proposed post-processing is described in Algorithm 2. The input of the algorithm is

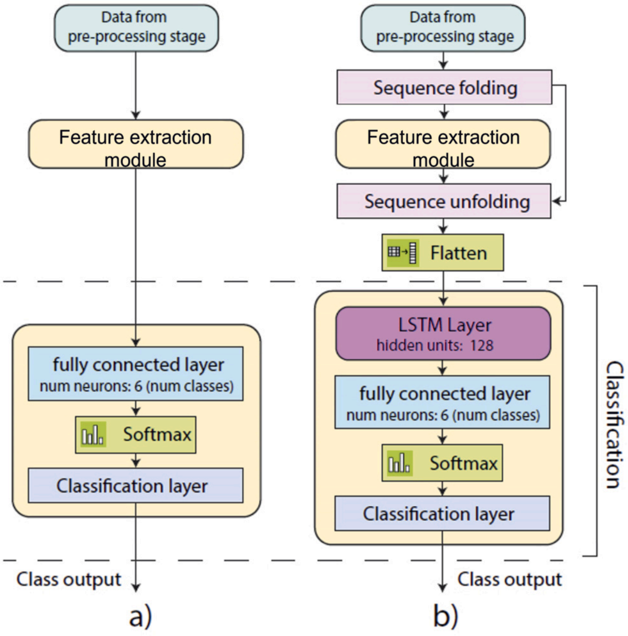


Fig. 6. Structure of the proposed a) CNN model and b) CNN-LSTM model.

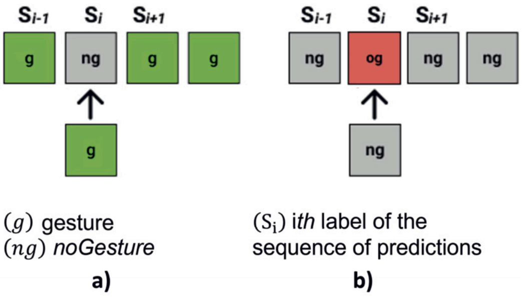


Fig. 7. Post-processing stage, a) addition of skipped labels, b) replacement of spurious labels.

the vector of predictions $Y_{[1:i+1]}$ up to window $i+1$. From $Y_{[1:i+1]}$ is obtained the mode gesture g, ignoring the class *noGesture* ng. In Algorithm 2, g is obtained in line 3. In the case of the first window ($i=1$), S_i is set as "*noGesture*", line 1 in Algorithm 2. When Y_i is different from g, the algorithm further checks if the previous prediction Y_{i-1}, g and the next prediction Y_{i+1} are equal. If it is the case, it means that the current label Y_i is an outlier and should be replaced with the mode g, lines 5-7 in Algorithm 2. If it is not the case, it means that the current label Y_i is not an outlier and should be preserved. Therefore, the algorithm assigns Y_i to S_i, lines 8-9 in Algorithm 2. The described algorithm is designed to address two scenarios: the first scenario is during the classification of early windows, when the HGR model tends to oscillate in its predictions. For instance, as depicted in Fig. 7.a, the incorrect label ng is replaced with the mode class g. The second scenario occurs when there is no significant muscle activity, but due to noise or other artifacts, the HGR model predicts a class. This scenario is illustrated in Fig. 7.b where the algorithm replaces the incorrect label og with ng. Note that Algorithm 2 introduces a delay in the HGR model's response because it requires Y_{i+1} to modify S_i. This delay spans one window and is equivalent to the value of the stride (75 - 150 ms).

Algorithm 2 Post-processing algorithm.

```
Input: $i, Y_{[1:i+1]}, S_{[1:i-1]}$                ▷ $Y$: vector of predictions
Output: $S_i$                  ▷ $S$: vector of predictions after post-processing
 1: $g \leftarrow MODE(Y_{[1:i]})$         ▷ $g$: mode of the labels ignoring noGesture
 2: if $i = 1$ then
 3:     $S_1 \leftarrow \}\}noGesture\varepsilon$
 4: else
 5:     if $Y_i \neq g$ then
 6:         if $Y_{i-1} = g \;\varsigma\; g = Y_{i+1}$ then
 7:             $S_i \leftarrow g$
 8:         else
 9:             $S_i \leftarrow \}\}noGesture\varepsilon$
10:         end if
11:     else
12:         $S_i \leftarrow Y_i$
13:     end if
14: end if
```

Table 2
Characteristics and Hyper-parameters of the CNN and CNN-LSTM models.

Hyper-parameter	Value
Window size	300
stride	30
Network input size	13x24x8
solver	Adam
Num. of learnables	219 190 (CNN model) 11'652 790 (CNN-LSTM model)
Num. training epochs	10
Initial learning rate	0.001
Learning rate decay	0.2
Drop period	8
Training time [approx]	6 h.
Prediction time	25.48 ± 16.80 ms (CNN model) 34.41 ± 39.32 ms (CNN-LSTM model)
Training hardware	Intel Xeon Gold 6252, 24/48 cores @2.1Ghz, 256GB RAM GPU Nvidia Quadro P4000 8GB
Mini-batch size	1024 (CNN model) 64 (CNN-LSTM model)
Sequence length (CNN-LSTM model)	sequences truncated to the same length as shortest sequence)
Layer output mode (CNN-LSTM model)	sequence

4. Results and discussion

This section presents the results obtained from the evaluation of the proposed HGR models. The analysis is divided into two parts: Validation and Testing. The Validation subsection focused mainly on the tuning of hyperparameters, with the majority being selected heuristically. Table 2 displays the selected hyper-parameters used for training the models. For the parameters that had the most impact on performance, such as stride and post-processing, rigorous experiments were conducted. The Testing subsection consisted of evaluating the selected models of each type (CNN and CNN-LSTM) using the public platform of the EMG-EPN-612 dataset. The results of this evaluation are extensively analyzed, and a comparison with other state-of-the-art models is also presented.

4.1. Validation: stride selection

The experiments conducted with the validation subset (306 subjects) of the EMG-EPN-612 dataset compared two stride configurations ($s = 15$ and $s = 30$). For these experiments only the CNN model was used. Fig. 8 shows classification and recognition accuracy using different strides and post-processing. On the left half, the model does not use post-processing, and achieves an accuracy of 97.58% with stride 15, and 97.39% with stride 30. On the right half, the model includes post-processing, and achieves the same classification for both strides. Recognition accuracy is the metric most benefited by post-processing, increasing from 28.73 to 88.40% with stride 15, and from 38.8 to 93.31% with stride 30. Fig. 8 indicates that post-processing improves recognition accuracy for both strides, from 28.73% to 88.40% for stride 15, and from 38.8% to 93.31% for stride 30. Fig. 8 also suggests that

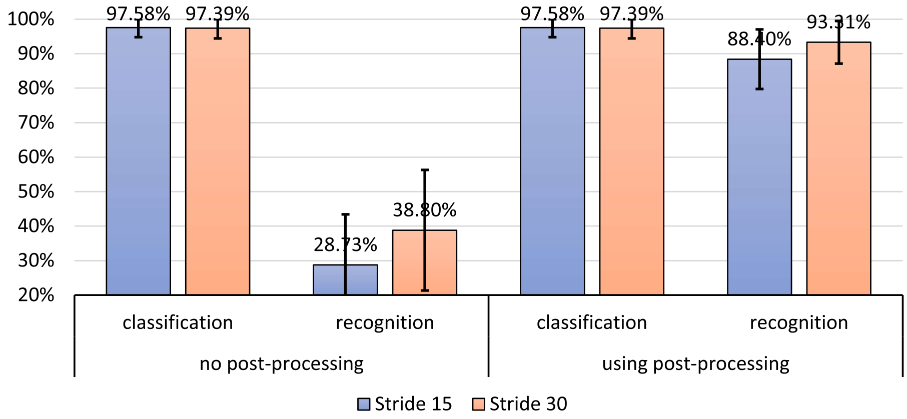


Fig. 8. Classification and recognition accuracy of the validation experiments using the CNN model: stride 15 vs stride 30.

Table 3
Results on the testing subset of the EMG-EPN-612 dataset (306 subjects).

	Classification accuracy	Recognition accuracy
CNN No PP.	90.62% ± 9.58%	45.40% ± 15.51%
CNN-LSTM No PP.	92.00% ± 8.82%	65.78% ± 15.51%
CNN post-processing	90.62% ± 9.58%	87.26% ± 11.14%
CNN-LSTM post-processing	92.00% ± 8.82%	**90.55% ± 9.45%**

Table 4
Results by sex on the testing subset (202 male M, 104 female F).

	Sex	Classification accuracy	Recognition accuracy
CNN	M	90.56% ± 10.36%	87.06% ± 12.07%
	F	90.74% ± 7.89%	87.65% ± 9.11%
CNN-LSTM	M	92.02% ± 9.55%	90.53% ± 10.21%
	F	91.96% ± 7.25%	90.58% ± 7.83%

stride 30 leads to a higher accuracy, specially the recognition accuracy with post-processing. Based on these results, a stride of $s = 30$ was selected.

4.2. Testing

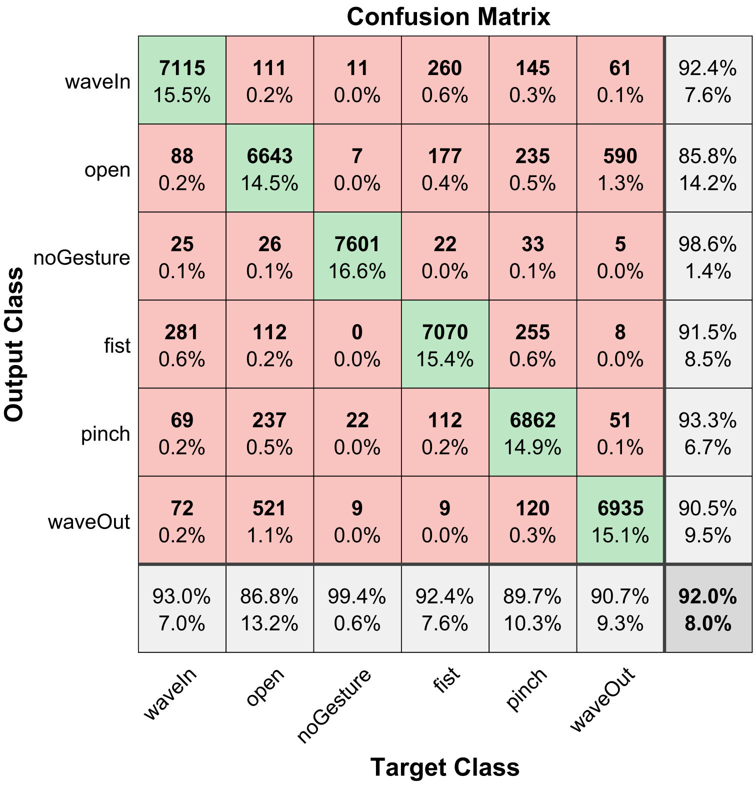
We evaluated the proposed models on the testing subset of the EMG-EPN-612 dataset using its online public evaluator system (Laboratorio de Investigación en Inteligencia y Visión Artificial “Alan Turing”, 2019). This evaluator is a closed system with intended samples for testing with no public available labels. The results obtained from the evaluation system are utilized to compare the performance of the CNN and CNN-LSTM models, both with and without the post-processing algorithm. Based on this comparison, a model will be chosen for further analysis.

4.2.1. Comparison of CNN, CNN-LSTM and post-processing

In Table 3, we present the classification and recognition results obtained by evaluating the proposed general models on the testing subset. Four different configurations were compared: CNN or CNN-LSTM, in conjunction with or without post-processing. It can be observed that the classification accuracy is independent of post-processing, as shown in the CNN model (90.62% ± 9.58%) and the CNN-LSTM model (92.00% ± 8.82%). The inclusion of the LSTM increased the classification accuracy only by 1.38%, and did not significantly change the standard deviation, for it had a reduction improvement of less than 1%. In terms of recognition accuracy, without post-processing the LSTM increased recognition accuracy by 20.38% (from 45.40% ± 15.51% to 65.78% ± 15.51%), while with post-processing, it increased only by 3.29% (from 87.26% ± 11.14% to 90.55% ± 9.45%). It is clear that the post-processing had a larger impact in performance. In fact, due to the post-processing algorithm, the recognition accuracy increased 41.86% (from 45.40% ± 15.51% to 87.26% ± 11.14%) for the CNN model, and 24.77% for the CNN-LSTM model (from 65.78% ± 15.51% to 90.55% ± 9.45%). Including post-processing increased the recognition accuracy in all cases, whereas it had no impact on classification accuracy. This finding evidences the limitations of relying solely on classification accuracy for assessing HGR, and instead recognition accuracy should be preferred. Although the CNN-LSTM models demonstrated superior performance compared to the CNN models, it resulted from a significant increase in network size (53-fold more learnable parameters, view Table 2). Instead, a relatively simple post-processing algorithm can lead to a more substantial improvement in overall performance. This highlights the significance of post-processing algorithms in HGR models and the potential for further progress in this area.

The standard deviation obtained is around 10% in all configurations tested. This behavior has been observed consistently in previous studies (Barona López et al., 2020) and is likely due to the inherent inter-personal variability of EMG signals. In terms of processing time per window, the CNN and CNN-LSTM models achieved times of 25.48 ± 16.80 ms and 34.41 ± 39.32 ms, respectively, demonstrating their real-time capabilities.

In Table 4, we present the results segmented by sex. The average classification and recognition accuracy were similar for males and females in both models. For instance, the classification accuracy in the CNN model was 90.56% for males vs 90.74% for females. A comparable scenario was witnessed regarding the recognition accuracy of the CNN-LSTM model. male obtained 90.53% for male vs 90.58% for female. On the contrary, the standard deviation for females was consistently lower in both models for both metrics (around 2.5%). For the CNN model, it was observed that the standard deviation was 2.47% lower for classification and 2.96% lower for recognition accuracy in females compared to males. In relation to the CNN-LSTM model, the standard deviation for classification and recognition accuracy was 2.3% and 2.38% lower, respectively, in females compared to males. The lower standard deviation observed in females compared to males could be due to several factors. One possible factor is the anatomical and physiological differences between males and females, which may affect the EMG signal characteristics. Additionally, it is possible that the motor skills and muscle activation patterns of females are more consistent than those of males, resulting in less variability in the EMG signals. However, further research is needed to determine the underlying reasons for this difference.


Confusion Matrix
Output Class
Target Class

	waveIn	open	noGesture	fist	pinch	waveOut	
waveIn	**7115** 15.5%	**111** 0.2%	**11** 0.0%	**260** 0.6%	**145** 0.3%	**61** 0.1%	92.4% 7.6%
open	**88** 0.2%	**6643** 14.5%	**7** 0.0%	**177** 0.4%	**235** 0.5%	**590** 1.3%	85.8% 14.2%
noGesture	**25** 0.1%	**26** 0.1%	**7601** 16.6%	**22** 0.0%	**33** 0.1%	**5** 0.0%	98.6% 1.4%
fist	**281** 0.6%	**112** 0.2%	**0** 0.0%	**7070** 15.4%	**255** 0.6%	**8** 0.0%	91.5% 8.5%
pinch	**69** 0.2%	**237** 0.5%	**22** 0.0%	**112** 0.2%	**6862** 14.9%	**51** 0.1%	93.3% 6.7%
waveOut	**72** 0.2%	**521** 1.1%	**9** 0.0%	**9** 0.0%	**120** 0.3%	**6935** 15.1%	90.5% 9.5%
	93.0% 7.0%	86.8% 13.2%	99.4% 0.6%	92.4% 7.6%	89.7% 10.3%	90.7% 9.3%	**92.0%** **8.0%**

Fig. 9. Confusion matrix of the CNN-LSTM model.

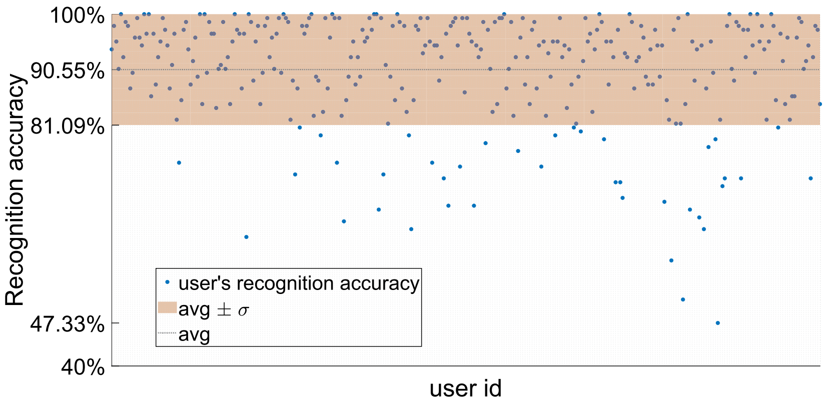


Fig. 10. Recognition accuracy by specific users for the CNN-LSTM model with post-processing.

4.2.2. Analysis of the CNN-LSTM model

The confusion matrix presented in Fig. 9 provides a detailed breakdown of the correct and incorrect predictions made by the CNN-LSTM model for each class. The model demonstrates an overall classification accuracy of 92.0% on the test data. The model exhibits the highest precision, 98.6%, for the *noGesture* class. This implies that the model's prediction for *noGesture* is accurate 98.6% of the time. On the other hand, the *open* class has the lowest precision at 85.8%, indicating that when the model predicts *open*, it is correct 85.8% of the time. In terms of recall, the *noGesture* class again stands out with the highest value of 99.4%. This means that the model correctly identifies *noGesture* 99.4% of the time when it is indeed the true class. Conversely, the *open* class has the lowest recall at 86.8%, suggesting that when the true class is *open*, the model correctly predicts it 86.8% of the time.

Fig. 10 is a visual representation of the accuracy of correct recognition among different users. The horizontal axis denotes each the user id. On the other hand, the vertical axis signifies the percentage of correct recognition achieved by each user. The blue dots scattered across the chart represent the accuracy of each individual user. These dots provide a clear picture of the variation in recognition accuracy among different users. Overlaying these blue dots is an orange area, which illustrates the range of one standard deviation from the mean accuracy. This area provides a sense of the dispersion or variability in the data. It shows where the majority of individual accuracies lie relative to the average. Speaking of the average, the mean accuracy across all users is 90.55%. This is a high average, suggesting that the recognition system generally performs well. However, the standard deviation is 9.45%, indicating a substantial spread in the accuracy scores around this mean. The range of accuracies is quite wide, with the lowest obtained accuracy being as low as 47.33%. This wide range suggests that the task's success may heavily depend on individual factors. These could include the user's familiarity with the system, their method of interaction, or even inherent personal characteristics. In conclusion, while the system shows a high average accuracy, there is a significant variation in individual user performance. This variation highlights the importance of considering individual user factors when evaluating the system's overall

Table 5
Comparison with other studies found in the literature.

Model	Recognition acc.	N. subjects
SVM (Jaramillo-Yanez et al., 2019)	87.5% ± 4.13%	60
Autoencoder (Chung & Benalcázar, 2019)	85.08% ± 15.21%	60
Orientation Correction (Barona López et al., 2020)	80.3%	306
Prop. CNN model	**87.26% ± 11.14%**	**306**
Prop. CNN-LSTM model	**90.55% ± 9.45%**	**306**

acc. = accuracy, Prop. = proposed, N. = number

effectiveness. It also suggests areas for potential improvement, particularly for those users experiencing lower recognition accuracies.

4.2.3. Comparison with other works in the literature

Lastly, we compare the outcomes of our general models with those presented in other studies found in the literature, as depicted in Table 5. It is evident that the CNN-LSTM model achieves the highest level of recognition accuracy among all the results analyzed. The key to reach these results, we attribute it to the combination of post-processing with Deep Learning, specifically the automatic feature extraction thanks to the parallel convolutions from the CNNs, and the sequential information from the LSTM. By using Deep Learning, we do not limit the model to a specific set of characteristics, but rather, we allow the model to learn the most relevant characteristics by itself. Another characteristic of our models is that these were trained with more subjects than other works from the literature which allows to a better generalization. As a side note, we consider that comparing the raw published results with other works in the literature is oversimplification and prone to confusion. So we compared with works that have the same set of gestures, evaluate recognition accuracy and have a significant number of subjects (> 10).

4.3. Discussion

Our results underscore the feasibility and effectiveness of employing spectrograms and CNN-LSTM networks for real-time Hand Gesture Recognition using EMG signals. The study also emphasizes the significance of integrating post-processing algorithms to enhance the accuracy and robustness of HGR models, particularly in the face of inter-personal variability and noise within EMG signals. Although our proposed post-processing algorithms introduce a delay in the recognition process, this is a fair trade-off to ensure the model's robustness and accuracy. Our proposed HGR model has the potential to improve user experience and satisfaction in HGR applications, as well as broaden the scope of possible applications and scenarios where it can be implemented. We believe our model is readily available and robust enough for applications such as human-computer interaction, entertainment applications, video games, and sign language recognition, and to a certain extent, in prosthetics.

However, critical applications, such as tele-operation of robotics in demanding environments, may still necessitate further research. Indeed, we identify the following points as potential directions for future investigation.

- Gesture Set Expansion: The current work recognizes only five hand gestures, limiting the applications of the HGR model. Future research could delve into more complex and diverse gestures, including dynamic, continuous, or sign language gestures.
- User Variance Minimization: We have observed that the HGR model's performance varies based on user characteristics such as age, sex, and muscle activation patterns. Future studies could explore methods to decrease this variability, enhancing the model's robustness and generalizability. This could involve data augmentation techniques, transfer learning, or personalized calibration for each user. Transfer learning, in particular, offers a promising approach to adapt the model to new users by leveraging knowledge learned from previous users, requiring fewer samples.
- Computational Cost Optimization: Our best model employs a CNN-LSTM architecture, which has a high number of learnable parameters and necessitates a large amount of training data. Future work could investigate ways to optimize the model's computational cost, such as employing smaller or simpler architectures. Specifically, a smaller model could be more suitable for embedded systems with limited computational resources or energy constraints.
- Real-Time Performance Enhancement: The current HGR model exhibits a latency of around 200 ms, which may not be suitable for some real-time applications. Future research could explore methods to reduce this latency, such as utilizing more efficient architectures, parallel processing, or hardware acceleration.
- Sensor Fusion: The current HGR model solely uses superficial EMG signals for hand gesture recognition. Future work could investigate sensor fusion techniques to combine EMG with other modalities, such as IMU, computer vision, or force sensors, to enhance the accuracy and robustness of gesture recognition.
- Adaptation to Intra-Personal Variability: EMG signals of the same user may change over time due to factors such as fatigue, electrode displacement, or environmental noise. Future work could employ online learning techniques to enable the model to learn from new data and adapt to changing conditions without requiring off-line retraining.

The integration of the HGR model with prosthetics represents a challenging yet promising research area, aiming to provide more natural and intuitive control for amputees using prosthetic hands. Some of the research topics and challenges in this field include:

- Haptic Feedback Integration: Incorporating haptic feedback mechanisms could enhance the user's sense of touch and proprioception, reducing the mental effort required to operate the prosthesis.
- Control Strategy Exploration: Different control strategies, such as hierarchical, biomimetic, or adaptive methods, could be explored to manage the complexity and variability of hand gestures.
- Prosthetic Hand Optimization: Optimizing the design and performance of the prosthetic hand, potentially through the use of shape memory alloy actuators, could achieve more realistic and dexterous movements.
- User Satisfaction Evaluation: Evaluating user satisfaction and acceptance of the prosthetic hand, as well as its impact on their quality of life and social interaction, could reduce the abandonment rate of prosthesis devices.

4.4. Ethical aspects

As EMG-based Hand Gesture Recognition systems gain traction in fields such as medicine and bionics, it becomes imperative to address the ethical implications associated with their use. EMG data, being personal health information, necessitates adherence to relevant privacy laws and regulations during its collection, storage, and utilization. The design of applications using EMG-based hand gesture recognition systems should prioritize privacy and security. Users must be thoroughly informed about the usage of their EMG data and should provide explicit consent prior to data collection. This includes a clear explanation of the purpose of data collection, the duration of data storage, and the entities that will have access to it. To ensure fairness and accountability, it is crucial that the models are trained with a diverse dataset that accurately represents the population that will use the system. This approach helps to avoid bias and discrimination. In the case of retraining or fine-tuning a model, users should be informed and their consent should be sought. When it comes to prosthetics, the model should be tested with amputees to evaluate its performance and robustness in real-world scenarios. In such cases, the predictability and explainability of the model are vital to building trust. It is important for users to understand how the system works, which includes transparency about the system's capabilities, limitations, and its decision-making process. Moreover, continuous monitoring and evaluation of the system's performance should be conducted to ensure its reliability and safety. Feedback from users should be actively sought and incorporated to improve the system. Ethical guidelines should be regularly updated to reflect technological advancements and societal changes. Lastly, collaboration with regulatory bodies, healthcare professionals, and ethicists can help in addressing ethical challenges and promoting responsible use of technology.

5. Conclusions

In this study, we proposed a novel Hand Gesture Recognition (HGR) model that uses EMG signals and combines spectrograms with CNN-LSTM networks for real-time recognition of 5 hand gestures. The experiments compared the impact of incorporating an LSTM network and a post-processing algorithm on the performance of the HGR architecture. The results showed that the memory cells improved the recognition accuracy of the CNN-LSTM model (90.55% ± 9.45%) by 3.29% compared to the CNN model (87.26% ± 11.14%). Despite the better performance of the CNN-LSTM model, it had 53 times more learnables than the CNN model. Instead, the post-processing originated an increment of 41.86% between CNN models, and 24.66% between CNN-LSTM models. Thus, the simple post-processing algorithm had a more significant impact on recognition accuracy than the inclusion of memory cells via LSTM networks. This finding indicates that incorporating post-processing algorithms can have a greater impact on recognition accuracy than traditional approaches such as hyper-parameter tuning and signal representation. By focusing on post-processing algorithms, researchers can potentially enhance the accuracy of HGR models and make them more robust against variability in EMG signals. Overall, our results show the advantages of using Deep Learning techniques combined with post-processing algorithms in the field of HGR, and are promising for improving the performance of HGR applications in the future. To ease the replication of our findings and potentially enable further advancements in this research, we made the code of our models publicly available, along with documentation and instructions for its execution. Future work includes expanding the set of gestures the system can recognize, so more versatile and diverse applications can be developed. Also, further efforts should be made to minimize the variance among users in order to achieve the level of robustness necessary for this kind of systems.

CRediT authorship contribution statement

Lorena Isabel Barona López: Project administration, Writing – review & editing. **Francis M. Ferri:** Investigation, Methodology, Software, Validation, Writing – original draft. **Jonathan Zea:** Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – review & editing. **Ángel Leonardo Valdivieso Caraguay:** Resources, Writing – review & editing. **Marco E. Benalcázar:** Conceptualization, Funding acquisition, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data freely available in Zenodo.

Acknowledgement

This work received support from Escuela Politécnica Nacional through the project PIGR-22-09 "Advancements in the Development of a Myoelectric Hand Prosthesis Prototype and Advanced Control of its Operation using Artificial Intelligence".

References

Antonius, R., & Tjahyadi, H. (2021). Electromyography gesture identification using cnn-rnn neural network for controlling quadcopters. *Journal of Physics. Conference Series*, *1858*, Article 012075. https://doi.org/10.1088/1742-6596/1858/1/012075. https://iopscience.iop.org/article/10.1088/1742-6596/1858/1/012075.

Asif, A. R., Waris, A., Gilani, S. O., Jamil, M., Ashraf, H., Shafique, M., & Niazi, I. K. (2020). Performance evaluation of convolutional neural network for hand gesture recognition using emg. *Sensors*, *20*, 1642. https://doi.org/10.3390/s20061642.

Atzori, M., Cognolato, M., & Müller, H. (2016). Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in Neurorobotics*, *10*. https://doi.org/10.3389/FNBOT.2016.00009.

Barona López, L. I., Valdivieso Caraguay, Á. L., Vimos, V. H., Zea, J. A., Vásconez, J. P., Álvarez, M., & Benalcázar, M. E. (2020). An energy-based method for orientation correction of emg bracelet sensors in hand gesture recognition systems. *Sensors*, *20*, 6327.

Benalcázar, M. E., Anchundia, C. E., Zea, J. A., Zambrano, P., Jaramillo, A. G., & Segura, M. (2018). Real-time hand gesture recognition based on artificial feed-forward neural networks and emg. In *2018 26th European signal processing conference (EUSIPCO)* (pp. 1492–1496). IEEE.

Benalcazar, M. E., Barona, L., Valdivieso, L., Aguas, X., & Zea, J. (2020). EMG-EPN-612 dataset. https://doi.org/10.5281/zenodo.4421500. retrieved on 2022-12-6.

Benalcázar, M. E., Motoche, C., Zea, J. A., Jaramillo, A. G., Anchundia, C. E., Zambrano, P., Segura, M., Palacios, F. B., & Pérez, M. (2017). Real-time hand gesture recognition using the myo armband and muscle activity detection. In *2017 IEEE second Ecuador technical chapters meeting (ETCM)* (pp. 1–6). IEEE.

Benalcázar, M. E., Valdivieso Caraguay, Á. L., & Barona López, L. I. (2020). A user-specific hand gesture recognition model based on feed-forward neural networks, emgs, and correction of sensor orientation. *Applied Sciences*, *10*, 8604.

Benatti, S., Rovere, G., Bösser, J., Montagna, F., Farella, E., Glaser, H., Schönle, P., Burger, T., Fateh, S., Huang, Q., et al. (2017). A sub-10mw real-time implementation for emg hand gesture recognition based on a multi-core biomedical soc. In *2017 7th IEEE international workshop on advances in sensors and interfaces (IWASI)* (pp. 139–144). IEEE.

Bird, J. J., Kobylarz, J., Faria, D. R., Ekart, A., & Ribeiro, E. P. (2020). Cross-domain mlp and cnn transfer learning for biological signal processing: Eeg and emg. *IEEE Access*, *8*, 54789–54801. https://doi.org/10.1109/ACCESS.2020.2979074.

Cai, Z., & Zhu, Y. (2021). A hybrid cnn-lstm network for hand gesture recognition with surface emg signals. In *Thirteenth international conference on digital image processing (ICDIP 2021) 11878, 74*. https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11878/2601074/A-hybrid-CNN-LSTM-network-for-hand-gesture-recognition-with/10.1117/12.2601074.full.

Chamberland, F., Buteau, E., Tam, S., Campbell, E., Mortazavi, A., Scheme, E., Fortier, P., Boukadoum, M., Campeau-Lecours, A., & Gosselin, B. (2023). Novel wearable hd-emg sensor with shift-robust gesture recognition using deep learning. *IEEE Transactions on Biomedical Circuits and Systems*, *17*, 968–984. https://doi.org/10.1109/TBCAS.2023.3314053.

Chen, X., Li, Y., Hu, R., Zhang, X., & Chen, X. (2020). Hand gesture recognition based on surface electromyography using convolutional neural network with transfer learning method. *IEEE Journal of Biomedical and Health Informatics*, *25*, 1292–1304.

Chung, E. A., & Benalcázar, M. E. (2019). Real-time hand gesture recognition model using deep learning techniques and emg signals. In *2019 27th European signal processing conference (EUSIPCO)* (pp. 1–5). IEEE.

Farina, D., Jiang, N., Rehbaum, H., Holobar, A., Graimann, B., Dietl, H., & Aszmann, O. C. (2014). The extraction of neural information from the surface emg for the control of upper-limb prostheses: Emerging avenues and challenges. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *22*, 797–809.

Farrell, T. R., & Weir, R. F. (2007). The optimal controller delay for myoelectric prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *15*, 111–118. https://doi.org/10.1109/TNSRE.2007.891391. https://ieeexplore.ieee.org/document/4126535/.

Gerdle, B., Karlsson, S., Day, S., & Djupsjöbacka, M. (1999). *Acquisition, processing and analysis of the surface electromyogram*. Berlin, Heidelberg: Springer Berlin Heidelberg (pp. 705–755). chapter 1.

Ghislieri, M., Cerone, G. L., Knaflitz, M., & Agostini, V. (2021). Long short-term memory (lstm) recurrent neural network for muscle activity detection. *Journal of NeuroEngineering and Rehabilitation*, *18*, 1–15. https://doi.org/10.1186/S12984-021-00945-W/FIGURES/6. https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-021-00945-w.

Guidetti, L., Rivellini, G., & Figura, F. (1996). Emg patterns during running: Intra- and inter-individual variability. *Journal of Electromyography and Kinesiology*, *6*, 37–48. https://doi.org/10.1016/1050-6411(95)00015-1.

Hahne, J. M., Farina, D., Jiang, N., & Liebetanz, D. (2016). A novel percutaneous electrode implant for improving robustness in advanced myoelectric control. *Frontiers in Neuroscience*, *10*. https://doi.org/10.3389/FNINS.2016.00114.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Hug, F., Turpin, N. A., Guével, A., & Dorel, S. (2010). Is interindividual variability of emg patterns in trained cyclists related to different muscle synergies?. http://www.jap.org. *Journal of Applied Physiology*, *108*, 1727–1736. https://doi.org/10.1152/JAPPLPHYSIOL.01305.2009/ASSET/IMAGES/LARGE/ZDG0061090770008.JPEG. https://journals.physiology.org/doi/10.1152/japplphysiol.01305.2009.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*, 685–695. https://doi.org/10.1007/S12525-021-00475-2/TABLES/2. https://link.springer.com/article/10.1007/s12525-021-00475-2.

Jaramillo-Yanez, A., Unapanta, L., & Benalcázar, M. E. (2019). Short-term hand gesture recognition using electromyography in the transient state, support vector machines, and discrete wavelet transform. In *2019 IEEE Latin American conference on computational intelligence (LA-CCI)* (pp. 1–6). IEEE.

Jaramillo-Yánez, A., Benalcázar, M. E., & Mena-Maldonado, E. (2020). Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review. *Sensors*, *20*, 2467. https://doi.org/10.3390/s20092467. https://www.mdpi.com/1424-8220/20/9/2467.

Jiménez, L. A. E., Benalcázar, M. E., & Sotomayor, N. (2017). Gesture recognition and machine learning applied to sign language translation. In *VII Latin American congress on biomedical engineering, CLAIB 2016, Bucaramanga, Santander, Colombia, October 26th-28th, 2016* (pp. 233–236). Springer.

Joshi, A., Monnier, C., Betke, M., & Sclaroff, S. (2017). Comparing random forest approaches to segmenting and classifying gestures. *Image and Vision Computing*, *58*, 86–95.

Karnam, N. K., Dubey, S. R., Turlapaty, A. C., & Gokaraju, B. (2022). Emghandnet: A hybrid cnn and bi-lstm architecture for hand activity classification using surface emg signals. *Biocybernetics and Biomedical Engineering*, *42*, 325–340. https://doi.org/10.1016/j.bbe.2022.02.005. https://linkinghub.elsevier.com/retrieve/pii/S0208521622000080.

Kingsbury, B. E., Morgan, N., & Greenberg, S. (1998). Robust speech recognition using the modulation spectrogram. *Speech Communication*, *25*, 117–132.

Laboratorio de Investigación en Inteligencia y Visión Artificial “Alan Turing” (2019). EMG Gesture Recognition Evaluator. https://aplicaciones-ia.epn.edu.ec/webapps/home/session.html?app=EMGGestureRecognitionEvaluator. retrieved on 2022-12-6.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

Li, Q., & Langari, R. (2022). Emg-based hci using cnn-lstm neural network for dynamic hand gestures recognition. *IFAC-PapersOnLine*, *55*, 426–431. https://doi.org/10.1016/j.ifacol.2022.11.220. https://linkinghub.elsevier.com/retrieve/pii/S2405896322028634.

Luh, G. C., Lin, H. A., Ma, Y. H., & Yen, C. J. (2015). Intuitive muscle-gesture based robot navigation control using wearable gesture armband. In *2015 International Conference on Machine Learning and Cybernetics (ICMLC)* (pp. 389–395). IEEE.

Moschetti, A., Fiorini, L., Esposito, D., Dario, P., & Cavallo, F. (2016). Recognition of daily gestures with wearable inertial rings and bracelets. *Sensors*, *16*, 1341.

Neto, O. P., & Christou, E. A. (2010). Rectification of the emg signal impairs the identification of oscillatory input to the muscle. *Journal of Neurophysiology*, *103*, 1093–1103.

Pallotti, A., Orengo, G., & Saggio, G. (2021). Measurements comparison of finger joint angles in hand postures between an semg armband and a sensory glove. *Biocybernetics and Biomedical Engineering*, *41*, 605–616.

Rafiee, J., Rafiee, M., Yavari, F., & Schoen, M. (2011). Feature extraction of forearm emg signals for prosthetics. *Expert Systems with Applications*, *38*, 4058–4067.

Reaz, M. B. I., Hussain, M. S., & Mohd-Yasin, F. (2006). Techniques of emg signal analysis: detection, processing, classification and applications. *Biological Procedures Online*, *8*, 11–35.

Riillo, F., Quitadamo, L. R., Cavrini, F., Saggio, G., Pinto, C. A., Pastò, N. C., Sbernini, L., & Gruppioni, E. (2014). Evaluating the influence of subject-related variables on emg-based hand gesture classification. In *2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA)* (pp. 1–5). IEEE.

Rodriguez-Falces, J., Navallas, J., & Malanda, A. (2012). Emg modeling. In G. R. Naik (Ed.), *Computational Intelligence in Electromyography Analysis*. Rijeka: IntechOpen. chapter 1, p. 1.

Saggio, G., Cavallo, P., Ricci, M., Errico, V., Zea, J., & Benalcázar, M. E. (2020). Sign language recognition using wearable electronics: implementing k-nearest neighbors with dynamic time warping and convolutional neural network algorithms. *Sensors*, *20*, 3879.

Saha, S., Konar, A., & Roy, J. (2015). Single person hand gesture recognition using support vector machine. In *Computational advancement in communication circuits and systems* (pp. 161–167). Springer.

Saponas, T. S., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., & Landay, J. A. (2009). Enabling always-available input with muscle-computer interfaces. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (pp. 167–176).

Sathiyanarayanan, M., & Rajan, S. (2016). Myo armband for physiotherapy healthcare: A case study using gesture recognition application. In *2016 8th International Conference on Communication Systems and Networks (COMSNETS)* (pp. 1–6). IEEE.

Satt, A., Rozenberg, S., & Hoory, R. (2017). Efficient emotion recognition from speech using deep learning on spectrograms. In *Interspeech* (pp. 1089–1093).

Seok, W., Kim, Y., & Park, C. (2018). Pattern recognition of human arm movement using deep reinforcement learning. In *2018 International Conference on Information Networking (ICOIN)* (pp. 917–919). IEEE. http://ieeexplore.ieee.org/document/8343257/.

Shanmuganathan, V., Yesudhas, H., Khan, M., et al. (2020). R-cnn and wavelet feature extraction for hand gesture recognition with emg signals. *Neural Computing & Applications*, *32*, 16723–16736. https://doi.org/10.1007/s00521-020-05349-w.

Shi, W. T., Lyu, Z. J., Tang, S. T., Chia, T. L., & Yang, C. Y. (2018). A bionic hand controlled by hand gesture recognition based on surface emg signals: A preliminary study. *Biocybernetics and Biomedical Engineering*, *38*, 126–135.

Shwedyk, E., Balasubramanian, R., & Scott, R. N. (1977). A nonstationary model for the electromyogram. *IEEE Transactions on Biomedical Engineering*, *BME-24*, 417–424. https://doi.org/10.1109/TBME.1977.326175. http://ieeexplore.ieee.org/document/4122723/.

Sohn, M. K., Lee, S. H., Kim, H., & Park, H. (2016). Enhanced hand part classification from a single depth image using random decision forests. *IET Computer Vision*, *10*, 861–867.

Song, C., Chen, C., Li, Y., & Wu, X. (2018). Deep Reinforcement Learning Apply in Electromyography Data Classification. In *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)* (pp. 505–510). IEEE. https://ieeexplore.ieee.org/document/8612213/.

Sunil, G., Saranya, A., Duraimurugan, N., Dineshkumar, R., & Ramachandra, A. C. (2023). An effective hand gesture recognition using convolutional neural network with long short-term memory. In *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)* (pp. 1–7).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

Tam, S., Boukadoum, M., Campeau-Lecours, A., & Gosselin, B. (2021). Intuitive real-time control strategy for high-density myoelectric hand prosthesis using deep and transfer learning. *Scientific Reports*, *11*, Article 11275. https://doi.org/10.1038/s41598-021-90688-4. http://www.nature.com/articles/s41598-021-90688-4.

Toro-Ossaba, A., Jaramillo-Tigreros, J., Tejada, J. C., Peña, A., López-González, A., & Castanho, R. A. (2022). Lstm recurrent neural network for hand gesture recognition using emg signals. *Applied Sciences*, *12*, 9700. https://doi.org/10.3390/APP12199700. https://www.mdpi.com/2076-3417/12/19/9700/htm.

Tryon, J., & Trejos, A. L. (2021). Evaluating convolutional neural networks as a method of eeg–emg fusion. *Frontiers in Neurorobotics*, *15*, 157. https://doi.org/10.3389/FNBOT.2021.692183/BIBTEX.

Tsironi, E., Barros, P., Weber, C., & Wermter, S. (2017). An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. *Neurocomputing*, *268*, 76–86.

Ullah, A., Ali, S., Khan, I., Khan, M. A., & Faizullah, S. (2020). Effect of analysis window and feature selection on classification of hand movements using emg signal. In *Proceedings of SAI Intelligent Systems Conference* (pp. 400–415). Springer.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*, 1–8. https://doi.org/10.1186/1471-2105-7-91/FIGURES/4. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-7-91.

Vásconez, J. P., López, L. I. B., Leonardo Valdivieso Caraguay, Á., & Benalcázar, M. E. (2022). Hand gesture recognition using emg-imu signals and deep q-networks. *Sensors*, *22*, 9613. https://doi.org/10.3390/S22249613. https://www.mdpi.com/1424-8220/22/24/9613/htm.

Wang, L., Fu, J., Chen, H., & Zheng, B. (2023). Hand gesture recognition using smooth wavelet packet transformation and hybrid cnn based on surface emg and accelerometer signal. *Biomedical Signal Processing and Control*, *86*, Article 105141. https://doi.org/10.1016/j.bspc.2023.105141. https://www.sciencedirect.com/science/article/pii/S1746809423005748.

Wang, N., Lao, K., & Zhang, X. (2017). Design and myoelectric control of an anthropomorphic prosthetic hand. *Journal of Bionics Engineering*, *14*, 47–59.

Wang, T., Zhang, W., Zhou, E., Zheng, S., Wang, Y., Pang, G., & Li, Y. (2023). An emg gesture recognition model based on batchnorm2d and incremental broad learning system. In *2023 3rd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT)* (pp. 330–337).

Weir, R. F. (2018). The emg properties limit ultimate classification accuracy in machine learning for prosthesis control. *Constructivist Foundations*, *13*, 265–266. http://constructivist.info/13/2/265.

Wojtara, T., Alnajjar, F., Shimoda, S., & Kimura, H. (2014). Muscle synergy stability and human balance maintenance. *Journal of NeuroEngineering and Rehabilitation*, *11*, 1–9. https://doi.org/10.1186/1743-0003-11-129/FIGURES/6. https://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-11-129.

Yang, K., Xu, M., Yang, X., & Chen, Y. (2021). An emg gesture recognition method based on multivariate variational mode decomposition. In *2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)* (pp. 518–521).

Yang, W., Yang, D., Liu, Y., & Liu, H. (2019). Emg pattern recognition using convolutional neural network with different scale signal/spectra input, https://doi.org/10.1142/S0219843619500130. 16.

Zea, J. A., & Benalcázar, M. E. (2019). Real-time hand gesture recognition: A long short-term memory approach with electromyography. In *(CSEI) International Conference on Computer Science, Electronics and Industrial Engineering* (pp. 155–167).

Zhang, W., & Zhang, J. (2022). Emg gesture recognition algorithm based on parallel multi-scale cnn. In *2022 2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT)* (pp. 562–568).

Zhao, N., Zhao, B., Shen, G., Suppiah, R., Kim, N., Sharma, A., & Abidi, K. (2022). Fuzzy inference system (fis) - long short-term memory (lstm) network for electromyography (emg) signal analysis. *Biomedical Physics & Engineering Express*, *8*, Article 065032. https://doi.org/10.1088/2057-1976/AC9E04. https://iopscience.iop.org/article/10.1088/2057-1976/ac9e04/meta.

Zhu, Y., & Yuan, B. (2014). Real-time hand gesture recognition with kinect for playing racing video games. In *2014 International Joint Conference on Neural Networks (IJCNN)* (pp. 3240–3246). IEEE.