ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN MODELO DE RECONOCIMIENTO DE GESTOS DE LA MANO UTILIZANDO SEÑALES EMG Y DEEP LEARNING

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

FRANCIS MICAEL FERRI RIPALDA
francis.ferri@epn.edu.ec

DIRECTOR: MARCO E. BENALCÁZAR, PHD.
marco.benalcazar@epn.edu.ec

CO-DIRECTOR: LORENA I. BARONA, PHD.
lorena.barona@epn.edu.ec

QUITO, JULIO 2021

CERTIFICATION

I certify that the present work was made by Francis Micael Ferri Ripalda, under my supervision.

Firmado electrónicamente por:
MARCO ENRIQUE
BENALCAZAR
PALACIOS

MARCO ENRIQUE BENALCÁZAR, PHD.

ADVISOR

LORENA ISABEL
BARONA LOPEZ

Firmado digitalmente por LORENA ISABEL BARONA LOPEZ
Nombre de reconocimiento (DN): c=EC, o=SECURITY DATA S.A. 1, ou=ENTIDAD DE CERTIFICACION DE INFORMACION, serialNumber=120820121611, cn=LORENA ISABEL BARONA LOPEZ
Fecha: 2021.07.28 13:53:11 -05'00'

LORENA ISABEL BARONA, PHD.

CO-ADVISOR

COPYRIGHT STATEMENT

I, Francis Micael Ferri Ripalda, declare under oath that the work described here is my own; that this work has not previously been submitted for any degree or professional qualification; and that I have consulted all the bibliographic references included in this document.

By means of this declaration, I assign my intellectual property rights over this work to the *Escuela Politécnica Nacional*, as established by the Intellectual Property Law, and by the current institutional rules and regulations at the time the work was made.

FRANCIS MICAEL FERRI RIPALDA

ACKNOWLEDGMENT

To my family for always giving me their support and understanding.

To all members of the *Laboratorio de Investigación en Inteligencia y Visión Artificial "Alan Turing"* for knowing how to advise and guide me during the development of this project, and for having allowed me to be part of the team.

To Marco Benalcázar, the advisor of my degree work, for have given me his support and patience during the realization of this project. To Juan Pablo Vásconez for his help during this project and writing advice.

Francis Micael Ferri Ripalda

TABLE OF CONTENTS

RESUMEN

El Reconocimiento de gestos de la mano (HGR, por sus siglas en inglés) es uno de los campos de investigación que ha desarrollado con éxito aplicaciones de interacción hombre-máquina en los últimos años. Los sistemas HGR consisten en identificar el momento en el que se realizó un determinado gesto con la mano, así como el tipo de gesto realizado. En este trabajo, proponemos la creación de un modelo HGR basado en una *Convolutional Neural Network* (CNN). Luego, adaptamos una capa de memoria *Long Short-Term Memory* (LSTM) a la arquitectura del modelo para observar su efecto en la precisión de clasificación, precisión de reconocimiento y el tiempo de procesamiento. La entrada del modelo son espectrogramas creados mediante señales electromiográficas (EMG) del antebrazo obtenidas a través del sensor comercial Myo Armband. Para las pruebas, realizamos experimentos utilizando un conjunto público de datos de 612 usuarios, y luego medimos y comparamos la precisión de clasificación y la precisión de reconocimiento entre 5 gestos diferentes y el no gesto. Los resultados fueron evaluados para el modelo propuesto (modelo basado en CNN) y su adaptación para usar la capa LSTM (modelo basado en CNN-LSTM). Los resultados mostraron una precisión de clasificación de 90,49% ± 9,70% y de reconocimiento del 86,83% ± 11,30% para el modelo basado en CNN; y una precisión de clasificación del 92,93% ± 8,23% y de reconocimiento del 91,60% ± 8,81% para el modelo basado en CNN-LSTM. Finalmente, concluimos que el uso de una capa LSTM ayuda al modelo a incrementar la precisión de clasificación y de reconocimiento, por lo cual definimos el modelo basado en CNN-LSTM como el modelo final del presente trabajo.

PALABRAS CLAVE: Reconocimiento de Gestos de la Mano, Deep Learning, Convolutional Neural Networks, Long Short-Term Memory, EMG, Espectrograma.

ABSTRACT

Abstract—Hand gesture recognition (HGR) is one of the fields of research that has successfully developed human-machine interaction applications in the last years. HGR systems consist of identifying the moment in which a certain hand gesture was made, as well as the type of gesture performed. In this work, we propose the creation of a HGR model based on a Convolutional Neural Network (CNN). Then, we adapt a Long Short-Term Memory (LSTM) layer to the architecture of the model to observe its effect on the classification accuracy, recognition accuracy and processing time. The input of the model are the spectrograms created using Surface electromyography (sEMG) on the forearm through the commercial sensor Myo Armband. For testing, we performed experiments using a public EMGs dataset of 612 users, and we measured and compared the classification and recognition accuracy between 5 different gestures and the no gesture. The results were evaluated for the proposed model (CNN-based model) and its adaptation to use the LSTM layer (CNN-LSTM-based model). The results showed that the classification accuracy reaches up to 90.49% ± 9.70% and the recognition up to 86.83% ± 11.30% for the CNN-based model, and classification accuracy up to 92.93% ± 8.23% and recognition up to 91.60% ± 8.81% for the CNN-LSTM-based model. Finally, we conclude that the use of an LSTM layer helps the model to increase the classification and recognition accuracy, for which we define CNN-LSTM-based model as the final model of the present work.

KEYWORDS: Hand Gesture Recognition, Deep Learning, Convolutional Neural Networks, Long Short-Term Memory, EMG, Spectrogram.

1. INTRODUCTION

Hand Gesture Recognition (HGR) consists of identifying to which class a hand gesture belongs from a set of defined movements, and determining the instant of time in which the gesture occurs [1]. The HGR has different human-machine interaction application domains that include bionics, video games, sign language recognition, medicine, among others [2]–[6]. One of the techniques applied for HGR is the processing of electromyography signals (EMGs), which are biomedical signals that measure electrical currents generated in muscles during their contraction that represent neuromuscular activities [7]–[9].

1.1. Research question

Using Deep learning techniques and EMG signals, it will be possible to develop an HGR model capable of identifying a gesture corresponding to one of the five classes considered: fist, wave in, wave out, open, pinch, which works with a recognition accuracy of at least 85% and running in real time (300 ms) [10].

1.2. General objective

Develop an HGR model based on EMG signals and Deep Learning techniques capable of recognizing five different hand gestures, with an accuracy of at least 85% and that works in real time (300 ms)

1.3. Specific objectives

- Review the state of the art to define Deep Learning architectures that can be used to build HGR models using EMG.
- Design an HGR model, that uses a CNN-based feature extractor and a memoryless classifier, that is capable of recognizing 5 different hand gestures.
- Build an HGR model that uses the same feature extractor as the non-memory model and uses an LSTM-based memory classifier that is capable of recognizing 5 different hand gestures.
- Evaluate the results of the generated models in terms of classification accuracy, recognition accuracy and response time.

1.4. General background

There are two types of methods to obtain muscle information using EMGs, which are invasive and non-invasive. Invasive methods allow a better measurement of the EMG signal, but they are impractical to be used commercially, since they require controlled environments, specialized staff, and cause discomfort to the user. The most common invasive method involves inserting needle-shaped electrodes into the muscles (intramuscular EMG) to obtain the signals. On the other hand, non-invasive methods are less precise when obtaining EMGs and more susceptible to noise. However, non-invasive methods are more practical, and can obtain high performance results for several HGR applications that could be used for commercial purposes such as bionic active prostheses, tele-operation systems, video games, among others [2]–[4], [11], [12]. The most common non-invasive method is called Surface electromyography (sEMG), which obtains movement information through the measure of the electric potential field produced by active muscle fibers by using electrodes placed on the skin [13].

To date, several HGR works have been developed, based on sEMG obtaining high performances. There are two types of HGR model: general models (trained with data from multiple users) and user-specific models (trained for each user with only its own data). Among the best results of specific user recognition are: reaching 95.32% [14], and 94.20% [15]. On the other hand, some examples of general models are: reaching 87.53% [16], 85.08% [17], and 80.31% [15]. There are also works that use Convolutional neural networks (CNNs) such as [18]. Additionally, the use of spectrograms combined with CNNs is an approach widely used in Speech recognition [19], [20], but thanks to the nature of EMGs it can also be applied to HGR.

A HGR system can be divided in 5 stages: data acquisition, pre-processing, feature extraction, classification, and post-processing [14], [15]. For each stage, several methods have been used obtaining different results. For data acquisition, different types of mechanisms have been used such as vision sensors [4], Inertial measurement units (IMUs) [21], gloves [22] and EMG [3], [10], [14]. In the pre-processing stage, rectification and filtering are widely used [1], [23], [24]. For feature extraction, several methods have been used to extract relevant features such as mean absolute value (MAV), root mean square (RMS), standard deviation (SD), variance (VAR), or automatic feature extraction methods such as CNNs, among others [3], [18], [25], [26]. For the classification stage, different Machine Learning techniques have been used such as k-nearest neighbors (kNN) [22], Support Vector Machines [27], Random Forests [28], [29], and Feed-forward and Recurrent Neural Networks [30]. Finally, for the post-processing stage, among the most used methods we can mention the elimination of consecutive repetitions, the gesture mode, threshold method, and velocity ramps [15]. It is worth mentioning that one of the most important characteristics of HGR systems is that they must work online, which means a computing time of less than 300 ms [10]. Thus, a trade-off between accuracy and processing time should be considered to analyze the HGR systems.

1.5. Contribution

Although several works from the current literature have obtained high accuracy performances for HGR systems, there are still several problems that must be solved to reach a robust model. For example, several authors are focused only on the development of user-specific models that get high accuracy results. However, the problems of inter-personal and intra-personal variability in the distribution of the EMG data [8] make it difficult to obtain high performances when using a user-general model. For this, the motivation of this work is focused on the search for user-general HGR models that obtain state-of-the-art performances. We summarize the main contributions of this project below.

- We tested our models in a public large dataset (EMG-EPN-612) composed of 612 users. We used 306 users for training and validation, and the other 306 users for testing two different user-general models. For the training and validation users we used 50% of each user's data for training, and the other 50% for validation. For the testing users, we only used the portion of the data without ground truth.
- We successfully developed two HGR user-general models. The first is a CNN-based model and the second is a CNN-LSTM-based model. The input of the models is the spectrogram information that is extracted from the EMGs. We also compare such models in terms of classification and recognition accuracy to define the model with best performance as the final model.
- We compare the accuracy results of the proposed models with other HGR systems found in the literature that worked with the same dataset. This is key since the dataset distribution should be the same to make an unbiased comparison of the models.

2. METHODOLOGY

In this section, we propose a base architecture to tackle the HGR problem based on EMG recorded signals, which is illustrated in Figure 1. As can be observed, such architecture is conformed by data acquisition, pre-processing, feature extraction, classification, and post-processing. Based on such architecture, we proposed two different user-general models: a CNN-based model and a CNN-LSTM-based model. The main difference between them relies on the classification stage since the first model uses an Artificial feed-forward neural network (ANN), whilst the other adds an LSTM layer before the ANN. Aditionally, the feature extraction for both models is based on the same CNN, and the pre-processing is based on spectrograms. We explain each stage in detail in the following sections.

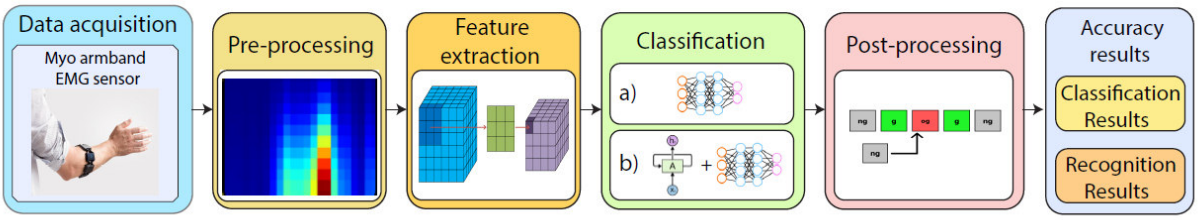


Figure 1: Hand gesture recognition proposed base architecture.

2.1. Data acquisition

In this work, we used the EMG-EPN-612 dataset [31]. This dataset consists of measurements of the EMG signals (EMGs) of 612 users. The measurements were performed by using the commercial Myo Armband bracelet sensor. The dataset is divided into 306 users for training and validation, as well as 306 for testing. For each user, 25 repetitions of each hand gesture were recorded. The dataset is composed by the following gestures: fist, wave in, wave out, open, pinch, and there are also measures for the no gesture or relax gesture. For each EMG sample, the ground truth is composed of information about when the gesture started and ended, as well as information about the class of each gesture. On the other hand, the ground truth of the test data is not accessible to the public since the authors encourage the use of their online testing platform to prevent the accuracy results from being manipulated [15]. The EMGs were recorded through 8 dry sensors, at a sample rate of 200Hz with 8 bits of resolution for each sensor. The sensor was placed on the forearm of the individual at the time of taking the samples. Each of the samples has a duration of 5 seconds. An illustration of the Myo armband sensor and the five hand gestures is presented in Figure 2.

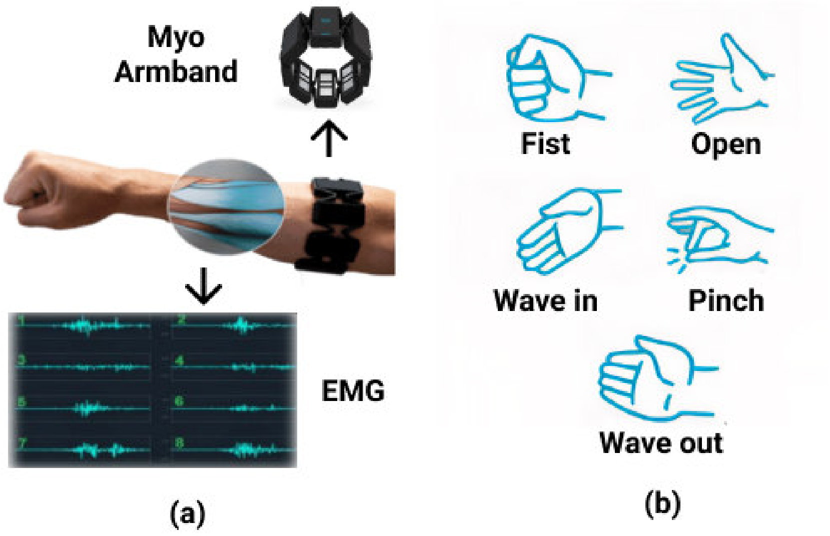


Figure 2: Myo Armband bracelet sensor and hand gestures illustration. a) Myo Armband bracelet sensor and sEMG measuring sample, b) the five gestures to be recognized.

2.2. Pre-processing

The Myo Armband sensor returns a normalized and discrete vector $\boldsymbol{E}(n, \omega) = (E_1(n, \omega), \dots,$ $E_8(n, \omega))^T \in [-1,1]^8$ for a given discrete instant of time $n \in Z^+$ and an EMG sample number $\omega \in Z^+$. Each component $E_i(n, \omega) \in \boldsymbol{E}(n, \omega)$ contains the measurement obtained by the channel i of the Myo Armband sensor. To simplify the notation a fixed EMG sample ω is assumed, so we can write $\boldsymbol{E}(n) = (E_1(n), \dots, E_8(n))^T$ where each component $E_i(n)$ is equal to the sum of the discretized and normalized values of the muscle signal $S_i(n)$ and the noise $N_i(n)$ respectively for $i = 1,2, \dots ,8$ [32].

In this work, we realized the segmentation procedure –splitting an EMG into multiple windows– by using a sliding window W over the EMGs. We selected experimentally a window width of $|W| = 300$ sample points as a design criterion, since it allows us to obtain high performance results. For an instant n the signal obtained from the window W is represented as follows: $\underline{\boldsymbol{S}}_n = \left(\boldsymbol{E}(n-299), \dots, \boldsymbol{E}(n)\right) \in [-1,1]^{8\times300}$ since we have 8 channels and 300 points for each window W. For the distance between each window observation (stride), we use two different values –stride of 15 and 30– to compare the accuracy results. An illustration that represents the segmentation procedure is shown in Figure 3.a.

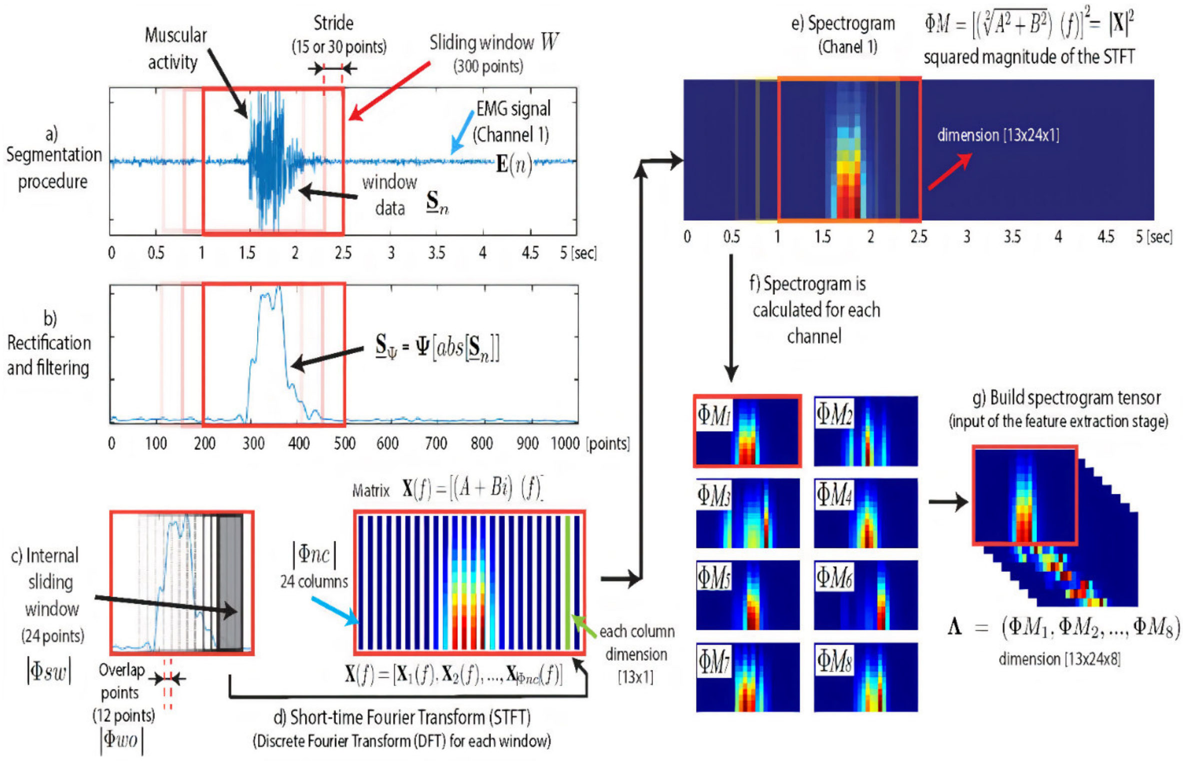

Muscular activity
Stride (15 or 30 points)
Sliding window W (300 points)
a) Segmentation procedure
EMG signal (Channel 1) $\mathbf{E}(n)$
window data $\underline{\mathbf{S}}_n$
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 [sec]
b) Rectification and filtering
$\underline{\mathbf{S}}_\Psi = \Psi[abs[\underline{\mathbf{S}}_n]]$
0 100 200 300 400 500 600 700 800 900 1000 [points]
c) Internal sliding window (24 points) $|\Phi sw|$
Overlap points (12 points) $|\Phi wo|$
Matrix $\mathbf{X}(f) = [(A+Bi)\ (f)]$
$|\Phi nc|$ 24 columns
each column dimension [13x1]
$\mathbf{X}(f) = [\mathbf{X}_1(f), \mathbf{X}_2(f), \ldots, \mathbf{X}_{\Phi nc}(f)]$
d) Short-time Fourier Transform (STFT)
(Discrete Fourier Transform (DFT) for each window)
e) Spectrogram (Chanel 1)
$\Phi M = [(\sqrt[2]{A^2+B^2})\ (f)]^2 = |\mathbf{X}|^2$
squared magnitude of the STFT
dimension [13x24x1]
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 [sec]
f) Spectrogram is calculated for each channel
ΦM_1
ΦM_2
ΦM_3
ΦM_4
ΦM_5
ΦM_6
ΦM_7
ΦM_8
g) Build spectrogram tensor (input of the feature extraction stage)
$\Lambda = (\Phi M_1, \Phi M_2, \ldots, \Phi M_8)$
dimension [13x24x8]

Figure 3: Pre-processing stage. a) Raw EMG signal information $\underline{\boldsymbol{S}}_n$ of a single channel, and window W with stride representation. b) Rectification and filtering process, c) Internal sliding window and overlap specification. d) Short Time Fourier transform (STFT). e) Spectrogram calculation. f) Spectrogram calculation for the 8 channels of the Myo armband sensor. g) Spectrogram concatenation to create the tensor that is the input of the feature extraction stage.

Once we performed the segmentation process, we rectify the $\underline{\boldsymbol{S}}_n$ signal by applying the absolute value $abs[\underline{\boldsymbol{S}}_n]$. By using rectification, we avoid that the mean in each channel of $\underline{\boldsymbol{S}}_n$ becomes zero [32]. Then, a digital low-pass Butterworth filter Ψ is applied to the signal $abs[\underline{\boldsymbol{S}}_n]$ to soften the signal and reduce noise. The filter Ψ has an order of 5 and a cutoff frequency of 10Hz selected as a design criterion. By applying the filter, we obtain the signal $\Psi[abs[\underline{\boldsymbol{S}}_n]]$ denoted as $\underline{\boldsymbol{S}}_\Psi$. An illustration that represents the rectification and filtering procedure is shown in Figure 3.b.

Once obtained $\underline{\boldsymbol{S}}_\Psi$, we use an internal sliding window Φsw inside the window W with width $|\Phi sw| = 24$ sample points and an overlapping $|\Phi wo|$ = 12 sample points as shown in Figure 3.c. Then, we use the Short-time Fourier transform (STFT) which consists in applying the Discrete Fourier Transform (DFT) by using the internal sliding window Φsw over the signal $\underline{\boldsymbol{S}}_\Psi$ as illustrated in Figure 3.d. The STFT creates a matrix expressed as $\boldsymbol{X}(f) =$

$[X_1(f), X_2(f), \dots, X_{|\Phi nc|}(f)]$ where f is the frequency analyzed (0 to 12 Hz) and $|\Phi nc|$ is the number of columns of $\boldsymbol{X}(f)$. It is to be noticed that the expression $\boldsymbol{X}(f)$ can also be represented as $\boldsymbol{X}(f) = [(A + Bi)(f)]$ which includes the real and imaginary part of the STFT. This procedure can be visualized in Figure 3.d.

Then the spectrogram is calculated as $\Phi M = [(\sqrt{A^2 + B^2})(f)]^2$ as can be visualized in Figure 3.e. Since the EMG signal has 8 different channels, the spectrogram ΦM is calculated for each of them as illustrated in Figure 3.f. Finally, we concatenate the ΦM of each channel in order to create the tensor $\Lambda = (\Phi M_1, \Phi M_2, \dots, \Phi M_8)$ which is the input of the feature extraction stage and is illustrated in Figure 3.g.

2.3. Feature extraction

Feature extraction methods are useful to extract relevant features from the EMGs, which can be defined in time, frequency, or time-frequency domains [32]. In this work, we extract time-frequency domain features from the spectrograms with a CNN feature extraction method [33]. The proposed CNN approach consists of several blocks of parallel convolutions and max-pooling inspired by the “Inception modules” used by GoogLeNet [34]. Thus, our proposed parallel convolution layer is presented in Figure 4. It can be observed that the internal blocks of the parallel layer allow the network to extract features with different convolution filters sizes (1x1,3x3, and 5x5), which allows extracting a large number of features that allow reaching high classification and recognition performances.

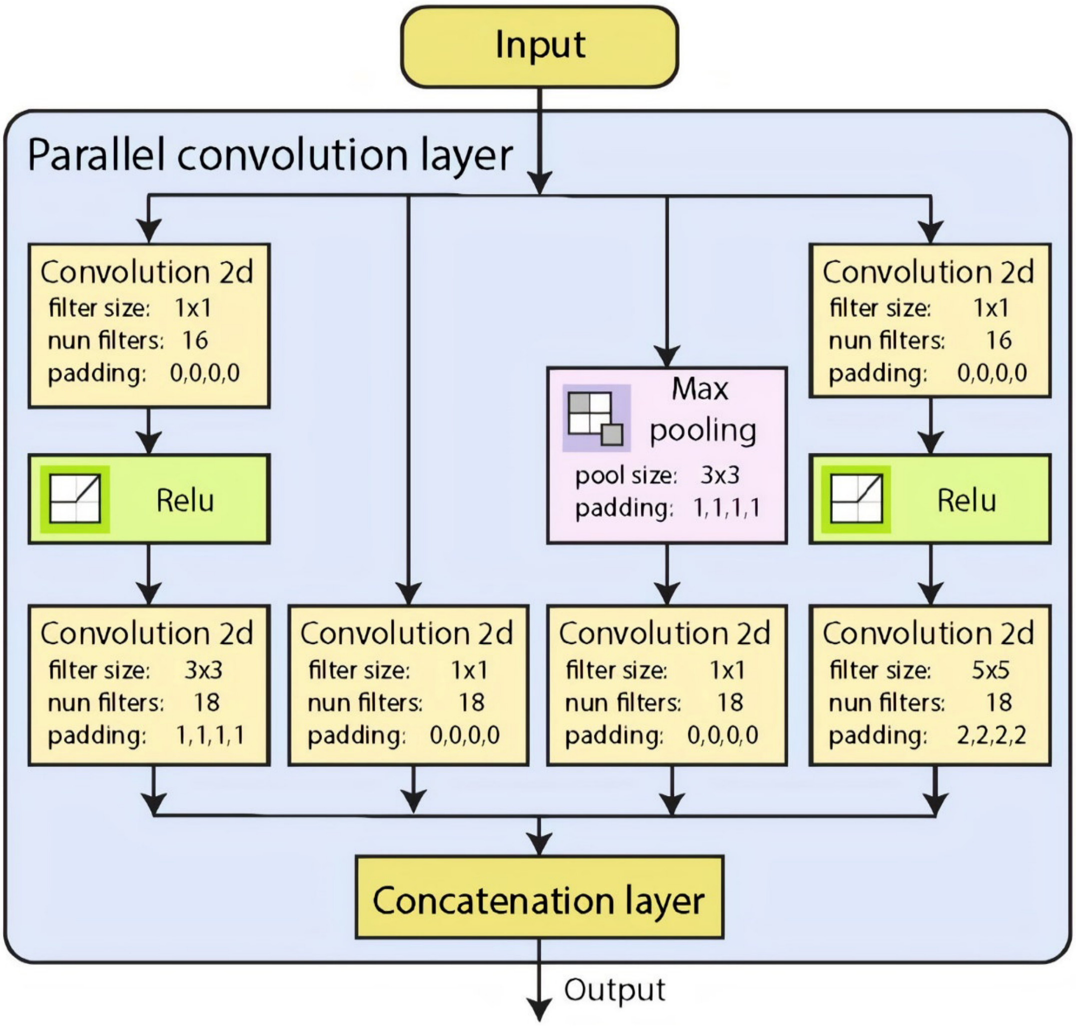


Figure 4: The base structure of parallel convolution layer and max-pooling.

We use several blocks of the base structure of parallel convolution layer combined with residual blocks to build the proposed feature extraction layer illustrated in Figure 5. The proposed feature extraction method adds residual blocks, which make it possible to avoid the problem of vanishing/exploding gradients and also allow faster training [35]. In this layer, we used 6 parallel convolution layers and 2 residual blocks. The first residual block takes the output of block 1 and adds it to the output of block 3, the second residual block takes the input of block 4 and adds it to the output of block 5. We add in the parallel convolution layer 6 a relu layer for each internal block before the concatenation layer. It is to be noticed that the input of the feature extraction layer has dimension of [13,24, 8], corresponding respectively to the output Λ of the pre-processing stage.

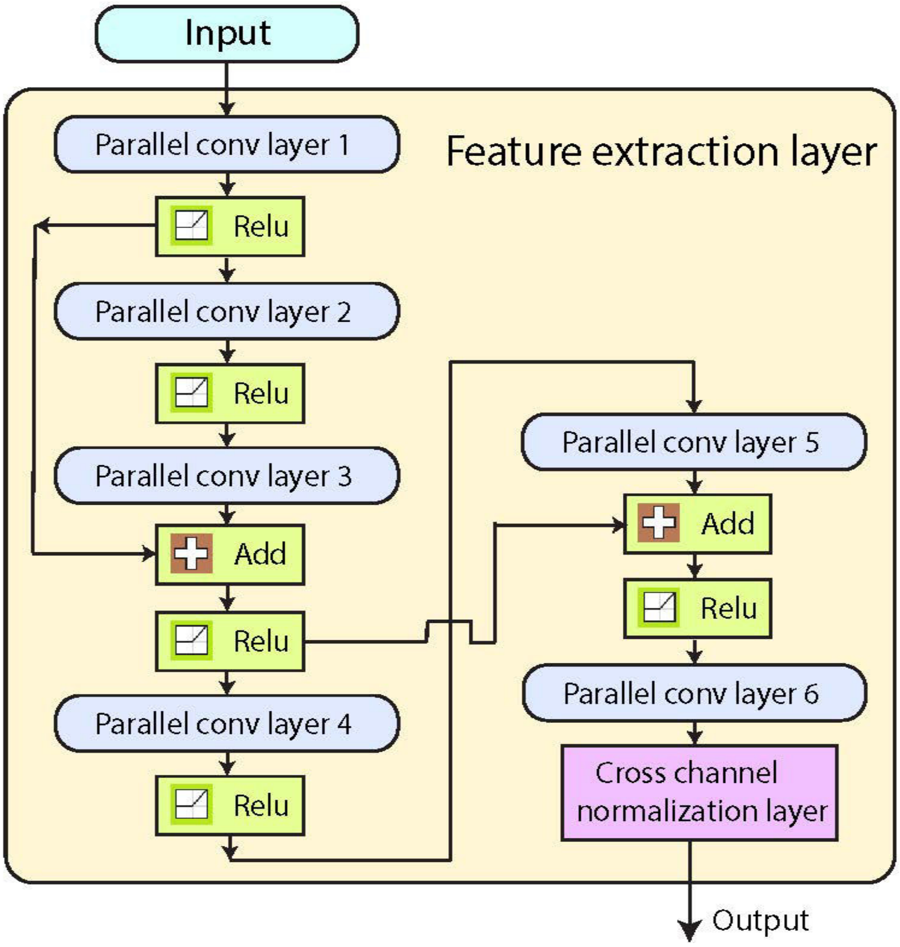


Figure 5: The structure of the feature extraction layer.

2.4. Classification

In this work, we propose two different approaches for the classification stage. The first approach is the CNN-based model that can be observed in Figure 6.a. This model takes the feature maps $\widehat{\Lambda}$ resulting from the feature extraction stage, and then such features are converted in a column vector to be the input of a fully connected layer of 6 neurons. The output of the fully connected layer is processed by the softmax activation function to calculate the probability of belonging to each of the classes. Finally, in the classification layer, we obtain the resulting class output with the highest probability if a threshold criterion $T = 50\%$ is accomplished, otherwise, the no gesture is considered as the resulting class.

The second model is the CNN-LSTM-based model, that can be observed in Figure 6.b. This model takes the spectrogram tensor Λ resulting from the pre-processing layer, and then such tensor becomes the input of a sequence folding layer. The sequence folding layer converts a batch of spectrogram tensor sequences to a batch of spectrogram tensors, which is useful to perform convolution operations on time steps of the spectrogram tensor sequences independently [36]. The output of the feature extraction layer $\widehat{\Lambda}$ then becomes the input of the sequence unfolding layer, which restores the sequence structure of the feature map input data. Then, a flatten layer is used to transforms the spatial dimensions of the feature maps into the LSTM input dimensions (128 hidden units). An LSTM layer is then used to learn long-term dependencies between sequence data [37]. The output of the LSTM layer is sent to a fully connected layer with 6 neurons, then processed by softmax activation function and a classification layer to obtain the resulting class output with the highest probability. Finally, the threshold criterion $T = 50\%$ is applied to define the resulting class. To facilitate the replication of the results of this work, it is important to mention that the architectures presented were implemented entirely in MATLAB.

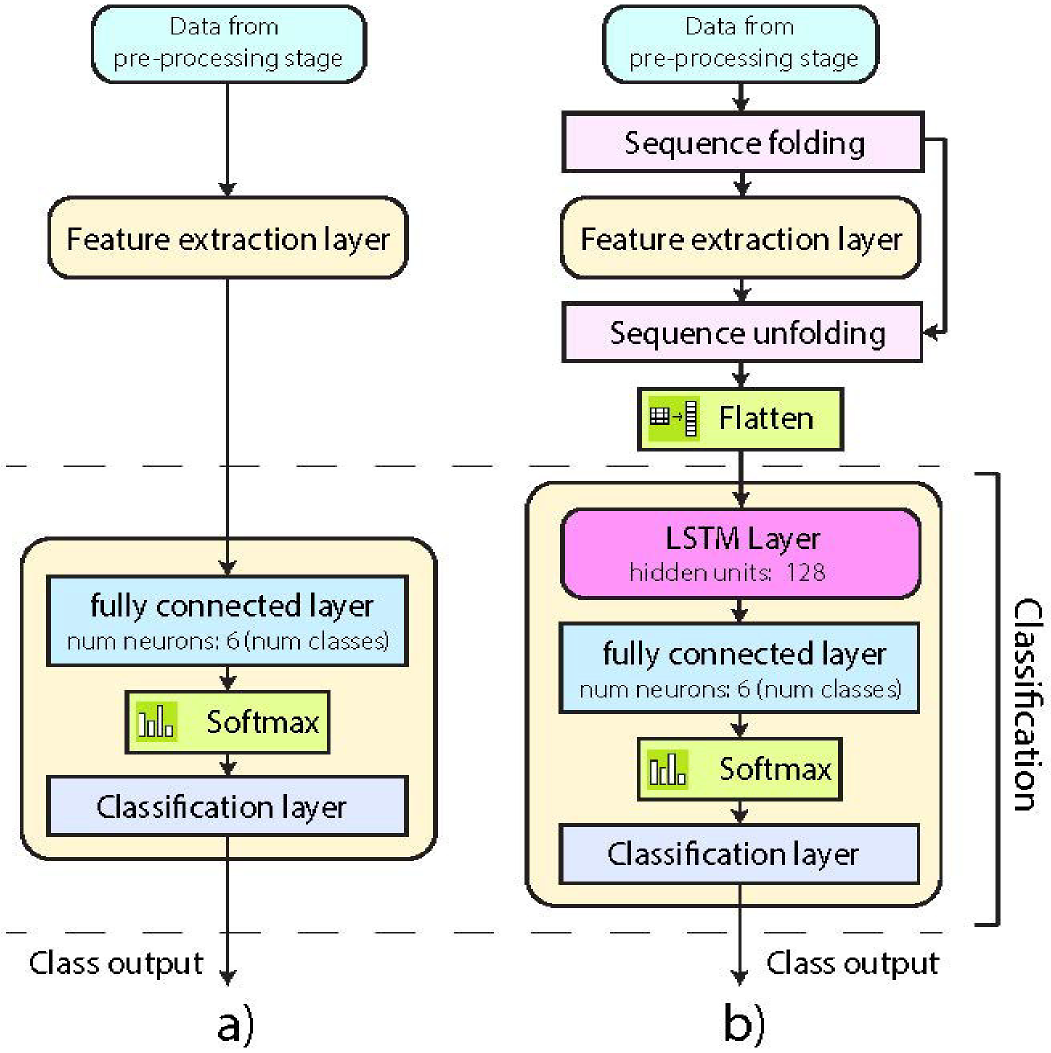


Figure 6: Proposed HGR models composed of the structures presented in Figure 4 and Figure 5. a) CNN-based model. b) CNN-LSTM-based model.

2.5. Post-processing

The post-processing filters the output of the classification stage to improve the accuracy of the proposed HGR system. In this work, we calculate the class mode of the whole sequence that is different from the no gesture. Then, all the labels in such sequence that are different from the mode gesture are replaced with no gesture. Next, we analyze each label in the sequence, and if the previous label equals the next label, the middle label is replaced with that value. However, for replacement, only the next label and the previous label are considered for the start and the end of the sequence, respectively.

3. RESULTS AND DISCUSSION

In this section, we present the results of evaluating the HGR user-general proposed models on the validation set. For this, we test different stride configurations and evaluate the models with and without the post-processing stage. The fixed hyper-parameters for the proposed models are presented in Table 1.

Table 1: Hyper-parameters for CNN-based and CNN-LSTM-based models.

Hyper-parameter name	**Hyper-parameter configuration**
Epochs	10
Learning rate	0.001
Learning rate decay	0.2
Learning rate drop period	8
Mini-batch size	1024 (lambdas) (CNN-based-model) 64 sequences of (lambdas) (CNN-LSTM-based-model)
Sequence length (Only CNN-LSTM-based model)	shortest (Truncate the sequences in each mini-batch to have the same length as the shortest sequence)
LSTM layer output mode (Only CNN-LSTM-based model)	sequence

3.1. Results

In Figure 7, we present the classification and recognition results for the proposed user-general models evaluated in the validation set. In Figure 7.a, we can observe the results of the CNN-based model, the stride changes from 15 to 30 and we add or remove the post-processing stage. It can be seen that the best results were obtained for stride of 30 with post-processing, with which values of 97.39% ± 2.96% for classification and 93.31% ± 6.19% for recognition were reached. The effect of using post-processing increases recognition results by up to 59.67%. However, the classification results were the same with and without post-processing. In Figure 7.b, we can observe the results of the CNN-LSTM-based model while the stride change from 15 to 30 and we add or remove the post-processing stage. The obtained results were similar to the results of Figure 7.a, where we can observe that post-processing is key to reach high-performance recognition results. It can be seen that the best result was obtained for stride of 30 with post-processing, with which an accuracy of 97.75% ± 2.49% for classification and 96.33% ± 3.65% for recognition were reached. Finally, we compare the best accuracy results obtained from the CNN-based model and CNN-LSTM-based model in Figure 7.c. It is to be seen that the CNN-LSTM-based model increased by 0.36% and 3.02% for classification and recognition accuracy, respectively. In addition, the standard deviation decreased by 0.47% and 2.54% for classification and recognition accuracy, respectively.

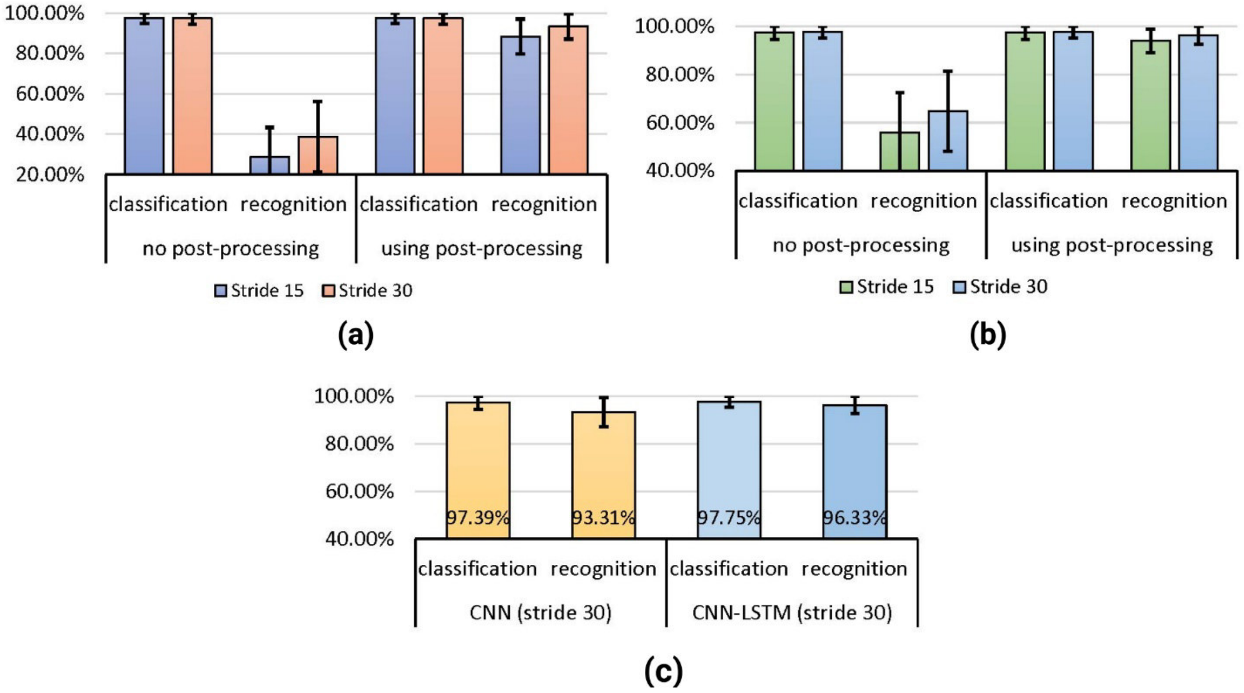


Figure 7: Validation results for 306 users, a) CNN-based model results, b) CNN-LSTM-based model results, c) Comparison between CNN-based and CNN-LSTM-based models.

In Figure 8, we present the classification and recognition results for the proposed user-general models evaluated in the testing set. In Figure 8.a, we can observe the results for the best-obtained models from the validation experiments for both, the CNN-based model and the CNN-LSTM-based model. A stride of 30 and the post-processing stage were used during this experiment. It can be observed that an accuracy of 90.49% ± 9.70% for classification and 86.83% ± 11.30% of recognition were obtained for the CNN-based model. On the other hand, an accuracy of 92.93% ± 8.23% for classification and 91.60% ± 8.81% for recognition were obtained for the CNN-LSTM-based model. It is to be noticed that the performance of the CNN-LSTM-based model outperforms the results obtained with the CNN-based model. Regarding the processing time per window, a time of 25.48±16.80 ms and 34.41±39.32 ms were obtained, for the CNN-based and CNN-LSTM-based models, respectively. This means that both models can work in real time (i.e., ≤300 ms).

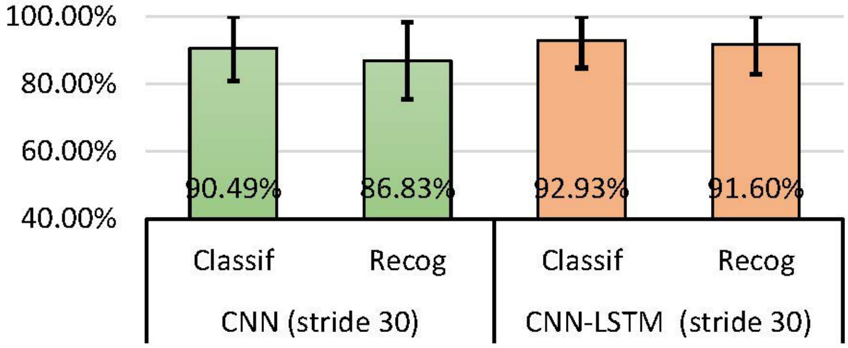


Figure 8: Testing results for 306 users.

In Figure 9, we present the comparison of the results of the CNN-based model and the CNN-LSTM-based model for men and women evaluated with the testing set. There was no significant difference between both models for the average of the classification and recognition accuracies, but the standard deviation for women is less in both models. For the CNN-based model, the standard deviation for women was lower in 2.66% for classification and in 2.58% for recognition compared to men. For the CNN-LSTM-based model, the standard deviation for women was lower in 1.57% for classification and in 1.77% for recognition compared to men.

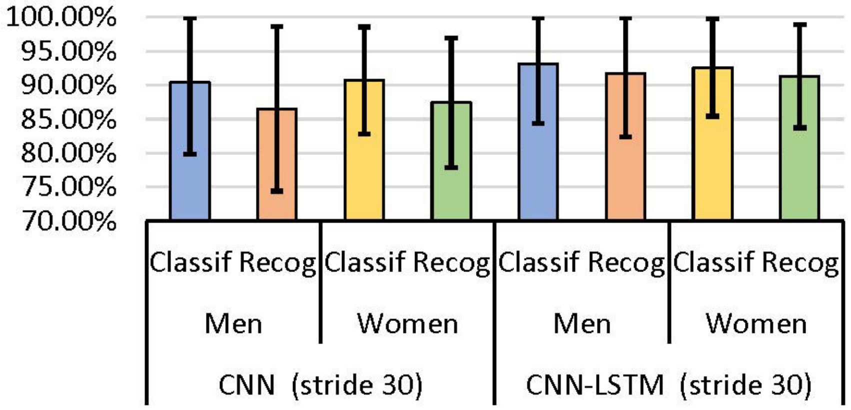
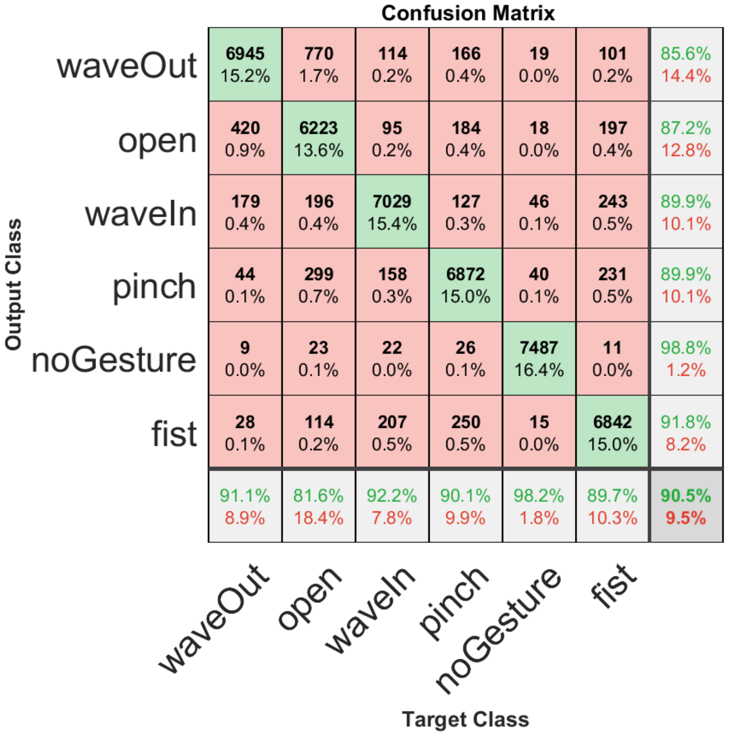


Figure 9: Testing results for 306 users. Comparison between the CNN-based model with the CNN-LSTM-based model for men and women.

Finally, we present the confusion matrix with the classification accuracy results for the best obtained user-general CNN-based model in Table 2, and for the best CNN-LSTM user-general model in Table 3. It is worth to mention that we used an online public evaluator to test the proposed models [38]. The online evaluator can be accessed through the following link: https://aplicacionesia.epn.edu.ec/webapps/home/session.html?app=EMG\%20Gesture\%20 Recognition\%20Evaluator.

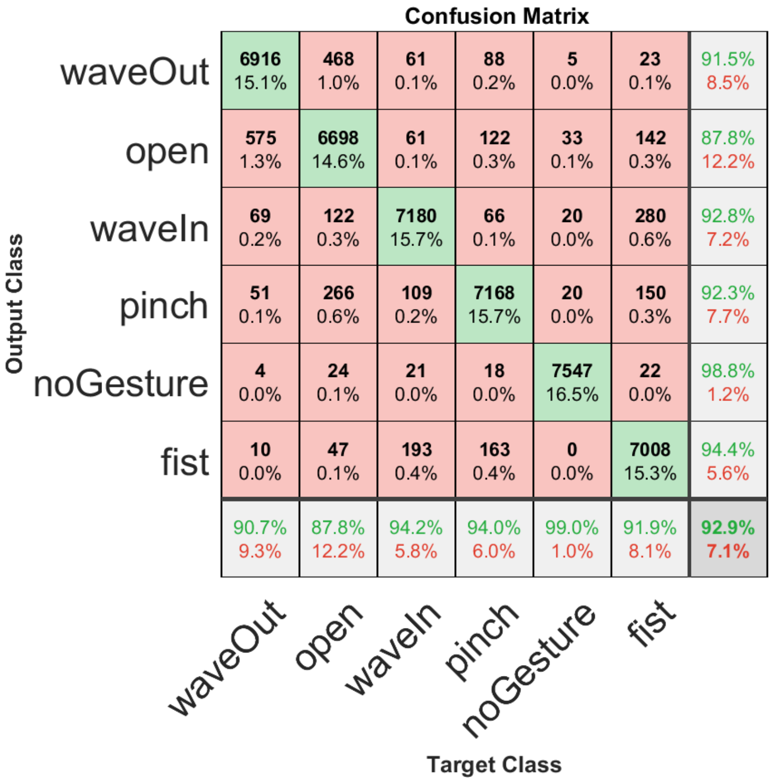
Table 2: Confusion matrix of classification for CNN-LSTM-based model.



Confusion Matrix

Output Class \ Target Class	waveOut	open	waveIn	pinch	noGesture	fist	
waveOut	6945 15.2%	770 1.7%	114 0.2%	166 0.4%	19 0.0%	101 0.2%	85.6% 14.4%
open	420 0.9%	6223 13.6%	95 0.2%	184 0.4%	18 0.0%	197 0.4%	87.2% 12.8%
waveIn	179 0.4%	196 0.4%	7029 15.4%	127 0.3%	46 0.1%	243 0.5%	89.9% 10.1%
pinch	44 0.1%	299 0.7%	158 0.3%	6872 15.0%	40 0.1%	231 0.5%	89.9% 10.1%
noGesture	9 0.0%	23 0.1%	22 0.0%	26 0.1%	7487 16.4%	11 0.0%	98.8% 1.2%
fist	28 0.1%	114 0.2%	207 0.5%	250 0.5%	15 0.0%	6842 15.0%	91.8% 8.2%
	91.1% 8.9%	81.6% 18.4%	92.2% 7.8%	90.1% 9.9%	98.2% 1.8%	89.7% 10.3%	**90.5%** **9.5%**

Table 3: Confusion matrix of classification for CNN-LSTM-based model.



Confusion Matrix

Output Class \ Target Class	waveOut	open	waveIn	pinch	noGesture	fist	
waveOut	**6916** 15.1%	**468** 1.0%	**61** 0.1%	**88** 0.2%	**5** 0.0%	**23** 0.1%	91.5% 8.5%
open	**575** 1.3%	**6698** 14.6%	**61** 0.1%	**122** 0.3%	**33** 0.1%	**142** 0.3%	87.8% 12.2%
waveIn	**69** 0.2%	**122** 0.3%	**7180** 15.7%	**66** 0.1%	**20** 0.0%	**280** 0.6%	92.8% 7.2%
pinch	**51** 0.1%	**266** 0.6%	**109** 0.2%	**7168** 15.7%	**20** 0.0%	**150** 0.3%	92.3% 7.7%
noGesture	**4** 0.0%	**24** 0.1%	**21** 0.0%	**18** 0.0%	**7547** 16.5%	**22** 0.0%	98.8% 1.2%
fist	**10** 0.0%	**47** 0.1%	**193** 0.4%	**163** 0.4%	**0** 0.0%	**7008** 15.3%	94.4% 5.6%
	90.7% 9.3%	87.8% 12.2%	94.2% 5.8%	94.0% 6.0%	99.0% 1.0%	91.9% 8.1%	**92.9%** **7.1%**

3.2. Discussion

In addition to the results shown, we compare our CNN-LSTM-based model results for the user-general model tested in the public dataset EPN-EMG-612 with other works in the literature as illustrated in Table 4. As can be seen, our model obtains the best recognition results compared to other works in the literature.

Table 4: Recognition accuracy comparison with other works from the literature.

Model	Recog. accuracy	Num. users
SVM [16]	87.5% ± 4.13%	60
Autoencoder [17]	85.08% ± 15.21%	60
SVM - Orientation Correction [15]	80.3%	306
Prop. CNN-based model	**86.83% ± 11.3%**	**306**
Prop. CNN-LSTM-based model	**91.6% ± 8.81%**	**306**
Recog=recognition, Num=number of, Prop=proposed		

Although the CNN-LSTM-based model obtained higher classification and recognition results than CNN-based model for the dataset EPN-EMG-612, there was an increase in the average processing time per window of 8.93 ms for the CNN-LSTM-based model. Considering that the CNN-based model also obtained high classification and recognition performances, the use of one model or another could depend on the kind of application. Using the CNN-based model would mean a decrease in the accuracy of classification and recognition respect to the CNN-LSTM-based model, it would have a shorter response time. On the other hand, using the CNN-LSTM-based model would mean greater precision of classification and recognition at the cost of a longer response time. Additionally, future works can include testing more classification configurations, as well as testing this approach on more data-sets.

4. CONCLUSIONS

In this work, we proposed and compared two HGR systems that works with CNN-based and CNN-LSTM-based models to classify and recognize five hand gestures using EMGs from a public dataset (306 for training and 306 for testing). We created spectrograms from the EMGs that were processed by the CNN and CNN-LSTM models. The results were evaluated for user-general HGR models (models trained with data from 306 users). The results obtained were encouraging, and they show that the CNN-based model and the CNN-LSTM-based model can classify and recognize successfully gestures based on EMGs. We demonstrated that the recognition accuracy of the CNN-LSTM-based model (91.60%±8.81%) is higher than the CNN-based model (86.83%±11.30%), which might be because the LSTM learns sequential information from the EMGs. It is important to highlight the importance of the post processing stage, which was able to increase the recognition results of the proposed models by up to 59.67%. Additionally, we compared the proposed approach with other works that we found in the literature that worked with the same public dataset, and we demonstrated that the CNN-LSTM-based model exceeds the recognition accuracy of the other models. Finally, after analyzing all the results we define the CNN-LSTM-based model using stride of 30 and post-processing as our final model.

5. REFERENCES

[1] M. E. Benalcázar *et al.*, “Real-time hand gesture recognition using the Myo armband and muscle activity detection,” in *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, 2017, pp. 1–6.

[2] G. C. Luh, H. A. Lin, Y. H. Ma, and C. J. Yen, “Intuitive muscle-gesture based robot navigation control using wearable gesture armband,” in *Proceedings - International Conference on Machine Learning and Cybernetics*, Nov. 2015, vol. 1, pp. 389–395, doi: 10.1109/ICMLC.2015.7340953.

[3] W. T. Shi, Z. J. Lyu, S. T. Tang, T. L. Chia, and C. Y. Yang, “A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study,” *Biocybern. Biomed. Eng.*, vol. 38, no. 1, pp. 126–135, 2018, doi: 10.1016/j.bbe.2017.11.001.

[4] Y. Zhu and B. Yuan, “Real-time hand gesture recognition with Kinect for playing racing video games,” in *Proceedings of the International Joint Conference on Neural Networks*, Sep. 2014, pp. 3240–3246, doi: 10.1109/IJCNN.2014.6889481.

[5] G. Saggio, P. Cavallo, M. Ricci, V. Errico, J. Zea, and M. E. Benalcázar, “Sign language recognition using wearable electronics: implementing k-nearest neighbors with dynamic time warping and convolutional neural network algorithms,” *Sensors*, vol. 20, no. 14, p. 3879, 2020.

[6] M. Sathiyanarayanan and S. Rajan, “MYO Armband for physiotherapy healthcare: A case study using gesture recognition application,” Mar. 2016, doi: 10.1109/COMSNETS.2016.7439933.

[7] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, “Techniques of EMG signal analysis: detection, processing, classification and applications,” *Biol. Proced. Online*, vol. 8, no. 1, pp. 11–35, 2006.

[8] J. Rodriguez-Falces, J. Navallas, and A. Malanda, “EMG modeling,” *Comput. Intell. Electromyogr. Anal. Perspect. Curr. Appl. Futur. Challenges*, pp. 3–36, 2012.

[9] D. Farina *et al.*, “The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 797–809, 2014, doi: 10.1109/TNSRE.2014.2305111.

[10] S. Benatti *et al.*, “A sub-10mW real-Time implementation for EMG hand gesture recognition based on a multi-core biomedical SoC,” in *Proceedings - 2017 7th International Workshop on Advances in Sensors and Interfaces, IWASI 2017*, Jul. 2017, pp. 139–144, doi: 10.1109/IWASI.2017.7974234.

[11] J. Rafiee, M. A. Rafiee, F. Yavari, and M. P. Schoen, “Feature extraction of forearm

EMG signals for prosthetics," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 4058–4067, Apr. 2011, doi: 10.1016/j.eswa.2010.09.068.

[12] T. S. Saponas, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay, "Enabling always-available input with muscle-computer interfaces," in *UIST 2009 - Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, 2009, pp. 167–176, doi: 10.1145/1622176.1622208.

[13] M. J. Zwarts and D. F. Stegeman, "Multichannel surface EMG: Basic aspects and clinical utility," *Muscle and Nerve*, vol. 28, no. 1. John Wiley & Sons, Ltd, pp. 1–17, Jul. 01, 2003, doi: 10.1002/mus.10358.

[14] M. E. Benalcázar, Á. L. V. Caraguay, and L. I. B. López, "A user-specific hand gesture recognition model based on feed-forward neural networks, emgs, and correction of sensor orientation," *Appl. Sci.*, vol. 10, no. 23, pp. 1–21, Dec. 2020, doi: 10.3390/app10238604.

[15] L. I. Barona López *et al.*, "An energy-based method for orientation correction of EMG bracelet sensors in hand gesture recognition systems," *Sensors*, vol. 20, no. 21, p. 6327, 2020.

[16] A. Jaramillo-Yanez, L. Unapanta, and M. E. Benalcazar, "Short-Term Hand Gesture Recognition using Electromyography in the Transient State, Support Vector Machines, and Discrete Wavelet Transform," Nov. 2019, doi: 10.1109/LA-CCI47412.2019.9036757.

[17] E. A. Chung and M. E. Benalcázar, "Real-time hand gesture recognition model using deep learning techniques and EMG signals," in *European Signal Processing Conference*, Sep. 2019, vol. 2019-September, doi: 10.23919/EUSIPCO.2019.8903136.

[18] X. Chen, Y. Li, R. Hu, X. Zhang, and X. Chen, "Hand Gesture Recognition based on Surface Electromyography using Convolutional Neural Network with Transfer Learning Method," *IEEE J. Biomed. Heal. Informatics*, vol. 25, no. 4, pp. 1292–1304, 2020.

[19] B. E. D. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech Commun.*, vol. 25, no. 1–3, pp. 117–132, Aug. 1998, doi: 10.1016/S0167-6393(98)00032-6.

[20] A. Satt, S. Rozenberg, and R. Hoory, "Efficient emotion recognition from speech using deep learning on spectrograms," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2017, vol. 2017-August, pp. 1089–1093, doi: 10.21437/Interspeech.2017-200.

[21] A. Moschetti, L. Fiorini, D. Esposito, P. Dario, and F. Cavallo, "Recognition of daily gestures with wearable inertial rings and bracelets," *Sensors*, vol. 16, no. 8, p. 1341, 2016.

[22] L. A. E. Jiménez, M. E. Benalcázar, and N. Sotomayor, "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*, 2017, pp. 233–236.

[23] M. E. Benalcázar, C. E. Anchundia, J. A. Zea, P. Zambrano, A. G. Jaramillo, and M. Segura, "Real-time hand gesture recognition based on artificial feed-forward neural networks and emg," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 1492–1496.

[24] O. P. Neto and E. A. Christou, "Rectification of the EMG signal impairs the identification of oscillatory input to the muscle," *J. Neurophysiol.*, vol. 103, no. 2, pp. 1093–1103, 2010.

[25] N. Wang, K. Lao, and X. Zhang, "Design and myoelectric control of an anthropomorphic prosthetic hand," *J. Bionic Eng.*, vol. 14, no. 1, pp. 47–59, 2017.

[26] A. Ullah, S. Ali, I. Khan, M. A. Khan, and S. Faizullah, "Effect of analysis window and feature selection on classification of hand movements using EMG signal," in *Proceedings of SAI Intelligent Systems Conference*, 2020, pp. 400–415.

[27] S. Saha, A. Konar, and J. Roy, "Single person hand gesture recognition using support vector machine," in *Computational advancement in communication circuits and systems*, Springer, 2015, pp. 161–167.

[28] A. Joshi, C. Monnier, M. Betke, and S. Sclaroff, "Comparing random forest approaches to segmenting and classifying gestures," *Image Vis. Comput.*, vol. 58, pp. 86–95, 2017.

[29] M.-K. Sohn, S.-H. Lee, H. Kim, and H. Park, "Enhanced hand part classification from a single depth image using random decision forests," *IET Comput. Vis.*, vol. 10, no. 8, pp. 861–867, 2016.

[30] E. Tsironi, P. Barros, C. Weber, and S. Wermter, "An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition," *Neurocomputing*, vol. 268, pp. 76–86, 2017.

[31] M. E. Benalcazar, L. Barona, L. Valdivieso, X. Aguas, and J. Zea, "EMG-EPN-612 Dataset." Zenodo, Nov. 2020, doi: 10.5281/zenodo.4421500.

[32] M. E. Benalcázar, A. G. Jaramillo, J. A. Zea, A. Paéz, and V. H. Andaluz, "Hand gesture recognition using machine learning and the myo armband," in *25th European Signal Processing Conference, EUSIPCO 2017*, Oct. 2017, vol. 2017-January, pp. 1040–1044, doi: 10.23919/EUSIPCO.2017.8081366.

[33] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015, doi: 10.1038/nature14539.

[34] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Oct. 2015, vol. 07-12-June-2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] MathWorks, “Long Short-Term Memory Networks.” p. 1, 2021, Accessed: Jul. 15, 2021. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/.

[37] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] Laboratorio de Investigación en Inteligencia y Visión Artificial “Alan Turing,” “EMG Gesture Recognition Evaluator.” https://aplicaciones-ia.epn.edu.ec/webapps/home/session.html?app=EMG Gesture Recognition Evaluator (accessed Jul. 21, 2021).