Classification of Hand Movements Using EMG Spectrograms and CNN-Based Architectures

Flavio Alfonso Juárez-Castro
Departamento de ingeniería
Universidad Autónoma de Querétaro
Santiago de Querétaro, Querétaro

Sebastián Salazar-Colores
Departamento de ingeniería
Centro de investigaciones en Óptica (CIO)
León, Guanajuato

Marco Antonio Aceves-Fernández
Departamento de ingeniería
Universidad Autónoma de Querétaro
Santiago de Querétaro, Querétaro
Corresponding author:
marco.aceves@uaq.edu.mx

Jesús Carlos Pedraza-Ortega
Departamento de ingeniería
Universidad Autónoma de Querétaro
Santiago de Querétaro, Querétaro

***Abstract*— Electromyographic signals (EMG) provide important information corresponding to muscle contraction; thanks to technology, obtaining the signals is nowadays applicable in vast areas of medicine. Our research aims to make this technology even more practical and relevant.**

However, artificial intelligence applications using myoelectric signals are constantly developing, which may pose a problem, mainly in individuals who have undergone amputations. The ability to maintain an efficient connection between the brain and muscles is gradually limited over time, reducing the potential use of myoelectric signals.

By leveraging the principles of surface electromyography (sEMG) and a Myo Armband device, we propose to transform myoelectric signals into spectrograms and classify four movements using a convolutional neural network (CNN))--based models The results obtained are summarized with 91% for the LeNet-5 model, outperforming a simple CNN model, which attained 89%, and significantly surpassing the VGG16 model, which yielded an accuracy of 86%. These findings validate that CNN-based models can effectively classify muscle movements with high precision, particularly when combined with spectrogram representations of EMG signals. We compare the results with those of other models. The objective is to take advantage of non-stationary signals, convert them into spectrograms, and use their characteristics to classify them.

Keywords—Electromyographic signals, private database, EMG, spectrograms, classification

I. Introduction

EMG signals are generated through muscle contraction. They provide relevant information about the electrical activity of the muscles. These signals are generated when electrical impulses from the nervous system activate the muscles and can be captured by electrodes placed on the skin's surface (sEMG) or inside the muscle (intramuscular EMG). [1]

EMG signals belong to the bioelectrical signals, which provide information not only about the electrical activity of muscles but also about organs and the central nervous system. [2]

EMG signals are first generated in the brain, where voluntary commands concerning movement are initiated. The motor neurons are activated when a decision is made to move a muscle (such as lifting the arm).

These signals are generated when we think of the movement descending from the brain to the spinal cord through the motor pathways. These pathways carry the signals from the brain to the spinal cord levels corresponding to the muscles to be activated. In the spinal cord, the signals reach the motor neurons that transmit the electrical impulse to the muscles via the peripheral nerves. When the electrical impulse reaches the neuromuscular junction, a neurotransmitter called acetylcholine is released, thus activating muscle contraction. [3-5]

This whole process can be summarized as shown in Figure 1.

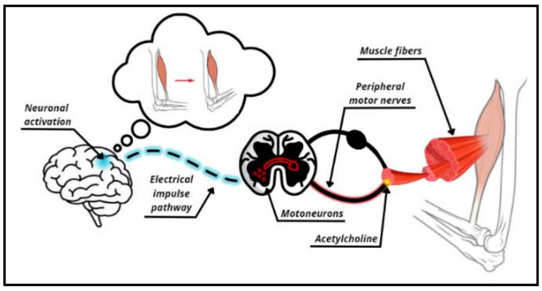


Figure 1. Process to generate a muscle contraction. Adapted from [5]

In Figure 2 shows the electrical activity of the muscle fibers that occurs during contraction is recorded as an EMG signal

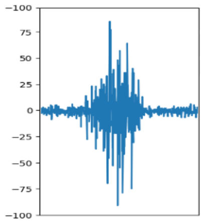


Figure 2. Representation of EMG signal. Adapted from [5].

Techniques exist to obtain electromyographic signals when movement is generated. However, it is essential to consider that the signal is not recorded under ideal conditions. These signals are often contaminated by noise or interference from various sources, such as the electrical activity of other muscles, environmental noise, or interference from electronic equipment. [6] In addition, body movements and poor electrode contact can introduce artifacts that complicate the correct interpretation of the signals.

The subject's anatomical, physiological, and nervous system characteristics affect the amplitude, time, and frequency properties of EMG signals. However, the measuring equipment used is also necessary. [7] Therefore, it is essential to consider actions when measuring an EMG signal, such as the intensity of the muscle contraction, the distance between the electrode (or sensor) and the muscle, and the amount of adipose tissue of the subject. As well as the level of contact between the skin and the electrode. [8]

Using spectrograms to analyze EMG signals can be an effective tool for mitigating noise problems from different sources. A spectrogram is a graphical representation that shows how the frequency of a signal varies over time.[9] By decomposing EMG signals into their frequency components, the spectrogram allows for identifying and isolating specific frequencies associated with noise, such as power line interference (50/60 Hz) or motion artifacts, typically in defined frequency ranges.

Works related to artificial intelligence follow the methodology of obtaining EMG signals and converting them into spectrograms.

Dawei H and Badong Ch. [10] explore sEMG classification to recognize hand movements, employing spectrograms, convolutional neural networks (CNNs), and long- and short-term memory (LSTM) neural networks. The main goal of the work is to decipher motion intention from sEMG signals. The study uses the Ninapro database, which provides a robust dataset with sEMG signals from 40 subjects performing 49 different gestures. This database is a standard reference for evaluating and comparing the effectiveness of sEMG classification methods. The proposed approach showed an accuracy of 79.329% for hand movement recognition.

Tryon J and Trejos AL. [11] investigate the use of CNNs to fuse electroencephalography (EEG) and EMG signals to improve the control of wearable robotic exoskeletons. EEG and EMG signals were obtained from 32 healthy subjects during elbow flexion and extension movements, varying the speed of the movement and the weight they were holding. CNN models were trained to classify tasks based on weight held during elbow movement, allowing the system to adjust to user conditions dynamically. The authors present that the spectrogram-based models outperformed the chance level (33.33%), achieving an average accuracy of 80.51% ± 8.07% with the stacked fusion model, which showed superior performance to models based solely on EEG or EMG. The stacked fusion model obtained an accuracy of 80.03% ± 7.02%, which also outperformed the individual EEG and EMG models, although in some cases, the differences were not statistically significant.

Jingwei T. et al. [12] present an approach for pattern recognition in EMG signals without resorting to manual feature extraction. They use spectrograms as direct input to a convolutional neural network (CNN) for hand and wrist motion classification. The study uses the NinaPro database data to evaluate two CNN architectures, including EMG signals from healthy subjects and amputees. The results indicate that the CNN-based approach can achieve an average accuracy of 88.04% in motion classification, demonstrating the effectiveness of this method in automatically classifying EMG signals without the need for manual preprocessing.

Thanks to the advances achieved in previous work, it is possible to address pattern recognition in EMG signals, obtaining better results in CNN-based architectures. However, it is also essential to consider that some previous works use public databases, which mainly obtain data in controlled conditions, with high-quality equipment, and, generally, with data ready for analysis. That is why, by using a private dataset containing myoelectric signals from 50 subjects obtained with a Myo Armband [13], a methodology for processing and classification of 4 movements using the signals converted into spectrograms is proposed. To validate the robustness of the proposed methodologies for EMG signal processing and classification. This will not only provide further generalization of the models but will also provide a framework for improving performance in real-world applications.

II. Methodology

A. Dataset

Data were acquired for Ramírez [13] to the Universidad Autonoma de Querétaro using a Myo Gesture Control Armband developed by Thalmic Labs.

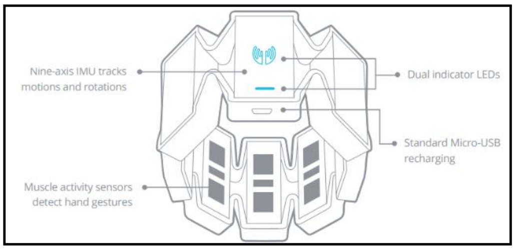

Nine-axis IMU tracks motions and rotations
Dual indicator LEDs
Standard Micro-USB recharging
Muscle activity sensors detect hand gestures

Figure 3. Representation of a Myo Armband.[14].

Myo Armband is a device that reads muscle activity in the subject's forearm using eight sensors, as shown in Figure 3, that function like surface electrodes. The bracelet was used to recognize hand position gestures and determine the muscles involved in those gestures. In addition, the bracelet has an accelerometer and a gyroscope to obtain the acceleration and orientation of the arm. [14, 15]

A total of 50 healthy people (i.e., they did not have motor problems or amputations) participated in the data collection. The age range was 18 to 60. There were 29 males and 21 females.

In our case study, we will only use four classes from the total dataset, including ulnar deviation, flexion, power grip, and initial position—all movements obtained from the right arm.

In Figure 4, the movements are described.

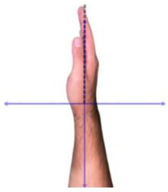
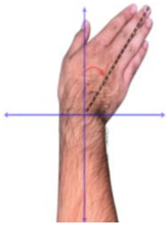
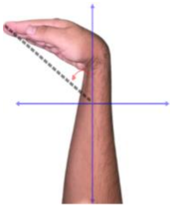
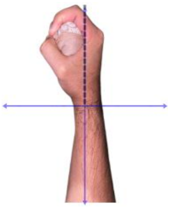
Movement name	Movement	Description
Initial position		*Hands in the initial position, palm extended with fingers closed, without exerting too much force on them. From this posture, the subsequent movements will begin.*
Ulnar deviation		*The arm is close to the trunk, the elbow is in 90° flexion, the hand is in the initial position, and the arm moves without moving the elbow, bends downwards, and returns to the initial position.*
Flexion		*The arm is close to the trunk, the elbow is in 90° flexion, and the hand is in the initial position. The palm is brought inwards without moving the elbow, facing the body, and then returned to the initial position.*
Power grip		*Arm close to the trunk, elbow in 90° flexion, hand in the initial position, proceed to move; make a fist, leaving the thumb outside over the other fingers, and return to the initial position.*

Figure 4. Movements made by 50 subjects.

Each participant executed the movement five times in three seconds, starting from second 1. This method allows us to determine when the movements (or events) start correctly.

It is worth mentioning that a notch filter was applied to attenuate the 60Hz frequencies generated by the power line when the data were acquired.

B. *Processing*

Once the data is acquired, it can be processed for subsequent use in training.

We previously commented that EMG signals present different distortions in a natural environment that directly affect the target we aim at. Therefore, applying a post-data acquisition filtering stage is necessary to reduce the problems when using the signals. [16]

As a first instance, we visually reviewed our signals to understand the nature of the signals (once the data were acquired). In this way, we can appreciate what type of logic needs to be applied.

Figure 5 compares the signals of the first subject obtained for the four movements using sensor 0.

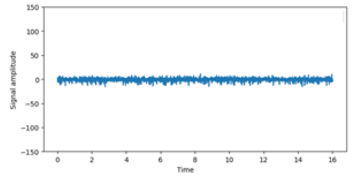
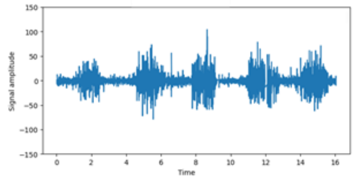
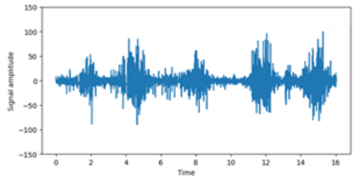
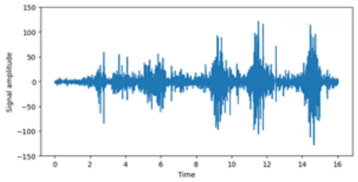
Movement	Signals from sensor 0 for the 1st subject
Initial position	
Ulnar deviation	
Flexion	
Power grip	

Figure 5. Comparison of the signals of the four movements obtained for the 1st subject using the 0 sensor.

As can be seen, unlike the initial position movement, the ulnar deviation, flexion, and power grip movements obtained very similar signals in sensor 0. The initial position class (or movement) will be considered a control class since, in this class, the signals obtained during the 16 seconds are empty (i.e., without any representative event).

Notably, the Myo Armband has eight sensors, so we have eight signals for each subject. In addition, despite applying a notch filter, the signals maintain a large amount of noise. For our processing logic, we use the wavelet transform to clean the noise from different sources present in our signal.

- *Wavelet transform (WT)*

The WT is a tool for simultaneously analyzing signals in the time and frequency domains. It allows the decomposition of a signal into components of different scales and resolutions, which is ideal for cases such as the present one, where non-stationary signals, i.e., signals that change over time, are handled. [17]

They are short-time mathematical functions used to represent a signal. They are scaled and shifted to fit different signal parts, allowing better resolution in both time and frequency.

Under the Wavelet Transform concept, the goal is to employ windows to analyze different portions of the signal of interest. This approach raises the question: How do we analyze a signal if we only segment it into windows of itself? In this context, the concept of the mother wavelet comes into play, which is used as a basis function for comparison with the segmented signal.

As shown in equation (1), Scaling and translation are two fundamental operations that adapt the mother wavelet. Scaling allows the wavelet to be adjusted, expanded, or compressed to capture details at different resolutions. On the other hand, translation moves the wavelet along the signal, allowing analysis at various time positions.

$$W_f(s,\tau) = \int f(t)\Psi_{s,\tau}(t)dt \qquad (1)$$

In this way, the wavelet transform offers a multiresolution analysis in the time-frequency domain.

Equation (2) describes that the WT allows the signal to be decomposed into low-frequency components (associated with overall signal trends) and high-frequency components (capturing details or fast fluctuations). This facilitates the identification and elimination of noise, improving the interpretation and processing of the original signal. [18-20]

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-\tau}{s}\right) \qquad (2)$$

By this logic, the mother wavelet must have a similar trend to our signal.

Different signals exist within the wavelet family; selecting the signal to be used depends on the problem to be addressed. Table 1 shows the 324 wavelet functions that can be used for different purposes, organized in 15 families. [1]

EMG signals, which are highly fluctuating, require a wavelet that can adapt to their rapid variations, enabling accurate time-frequency analysis. The Daubechies wavelet is an optimal choice because it can represent both local and global features of the signals and is suitable for denoising. In addition, the Daubechies wavelet has been widely used in previous studies on biomedical signal analysis, demonstrating its effectiveness in improving the interpretation and processing of electromyographic signals.

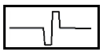
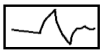
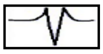
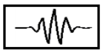
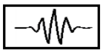
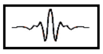
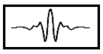
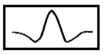
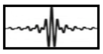
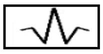
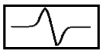
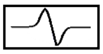
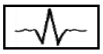
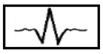
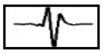
Wavelet family name	**known name**	**Wavelet number**	**Graphic representation**
Haar	Db1	1	
Daubechies	db2-db45	2-45	
Coiflet	coif1-coif5	46-50	
Morlet	morl	51	
Complex discrete morlet	cmor	52-147	
Discrete meyer	dmey	148	
Meyer	meyr	149	
Mexican hat	mexh	150	
Shannon	shan	151-200	
B-Spline (Basis splines)	fbsp	201-260	
Gaussian	gaus	261-267	
Complex gaussian	cgaus	268-275	
Biorthogonal	bior	276-290	
Inverse biorthogonal	rbio	291-305	
Symlet	sym	306-324	

Table 1. List of the 324 wavelets sorted into 15 families.

Once the mother wavelet is selected and applied, the mathematical logic described in equations (1) and (2) is used to our signals. Our signals will have eliminated the noise caused by different artifacts. Figure 6 Compares an unprocessed signal with a processed signal.

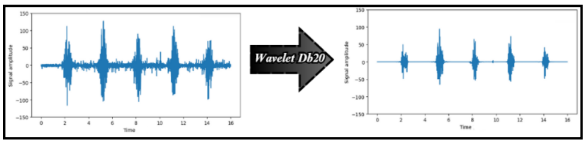


Figure 6. Application of WT to a noisy signal.

As previously mentioned, participants repeated each movement, so the total time of data collection was 16 seconds, and five movements were performed within that period.

The data were segmented as separate events as part of the processing, so each participant provided five events. Using equation (3) resulted in 250 events per movement.

$$(5\,events * 5\,subjects) = 250\,events \qquad (3)$$

The authors of the dataset specify that the data were taken every 3 seconds, so that, in the total space, windows of 3 seconds were taken starting at 0.5 seconds and increasing 3 seconds between windows, so that the segmentations were performed at times 0.5-3.5, 3.5-6.5, 6.5-9.5, 9.5-12.5, 12.5-15.5. In this way, we ensured that important information regarding an event was included in the segmentation space.

After applying the segmentation to the processed data, we obtain the events and store them in separate segments. Figure 7 shows an example of segmentation with the previously specified times.

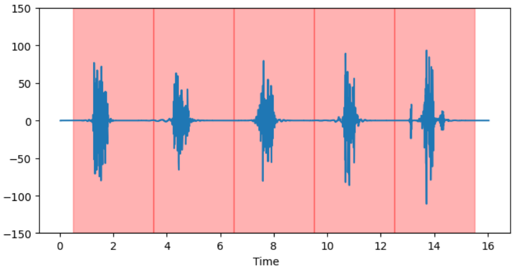


Figure 7. Segmentation of the signals.

Finally, by segmenting the data into events, separating them, and using equation (4), we have 1000 data (among the four classes), which are also divided into eight sensors, which can be used as input to the models.

$$(250\,events\, *\, 4\,classes) \;=\; 1000\,input\,data \qquad (4)$$

- *Spectrograms*

Spectrograms are visual representations that show how the energy of a signal is distributed over time and frequency. There are different methodologies for generating spectrograms, from the Wavelet Transform to the Fourier Transform. For our purpose, we propose using the Short-Time Fast Fourier Transform (STFT) to generate the spectrograms of the segmented signals to combine the logic applied for processing our data by WT with the logic of the STFT.

STFT allows for dividing the signal into smaller segments and applying the Fourier Transform to each of them, which provides a representation in the time-frequency domain, which is ideal for analyzing non-stationary signals such as EMG. In our case, having segmented windows, we apply the Fourier transform to our segment directly.

Using the STFT, it is possible to generate a two-dimensional representation. Thus, a spectrogram is a two-dimensional image where one axis represents time, the other axis represents frequency, and the intensity of the color or luminosity at each point of the image indicates the amplitude or energy of the signal at that frequency and at that specific time. [21]

The STFT of a discretized signal $x(n)$ is defined as equation (5):

$$x(n) = \; X(m,\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n-m) * e^{-j\omega m} \quad (5)$$

Where *x(n)* is the EMG signal in the time domain, *w(n)* is the window function that is applied to the signal to limit the analysis to a specific time segment, *m* is the index representing the displacement of the window in time, ω is the angular frequency and $e^{-j\omega m}$ is the Fourier transform term representing the complex oscillation. [22]

In this way and following a logic similar to WT, by applying STFT to the segmented EMG signal, complex values $X(m,\omega)$ are obtained that represent the amplitude and phase of the different frequencies in each time segment. These values are used to create spectrograms, which are visual representations of signal energy as a function of time and frequency.

Figure 8 shows the result of the conversion of a segment into a spectrogram.

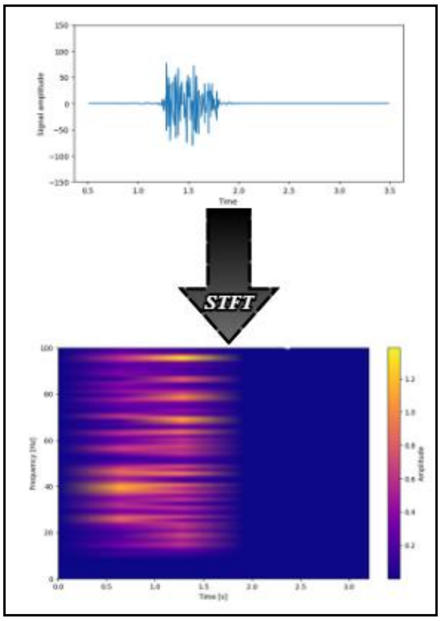


Figure 8. Conversion of the signal into a spectrogram.

Once the conversion of the segments into spectrograms has been applied, the data is ready to be used as input for a motion classifier.

- *Models used*

CNN-based models will be used to classify muscle movements in this work. Thanks to the state of the art, it has been confirmed that these models are particularly effective in image classification tasks and complex pattern analysis due to their ability to extract spatial and temporal features. Convolutional Neural Networks (CNNs) have demonstrated excellent performance by leveraging multiple convolutional and pooling layers to learn hierarchical representations of data, making them suitable for tasks such as gesture recognition, speech processing, and muscle movement classification from bioelectrical signals.

The first model used is a simple CNN, and its architecture is shown in Figure 9.

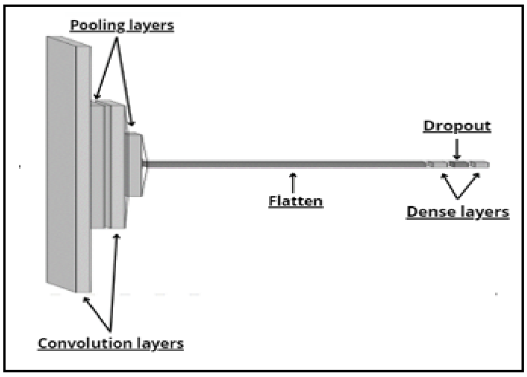


Figure 9. Simple CNN architecture

The architecture consists of *two convolution layers*, which detect patterns in the spectrograms when applying the filters (kernels). The first convolution has 32 filters, and the second has 64 filters, both of size (3x3) and two *pooling layers*. This layer reduces the dimensionality (the width and height of the features detected by the convolution), which helps to reduce the number of parameters and prevent overfitting. The pooling is performed with a size of (2x2), which divides by two the resolution of the images. The third layer, representing the *flatten layer*, takes the three-dimensional features (width, height, and depth) and converts them into a single dimension (a vector), for later use in the fully connected layers. The next layer represents the *dense layers*. These are fully connected layers in which each node is connected to all the nodes of the previous layer. These layers perform the final classification using 128 neurons for the first layer and four neurons for the output.

The last layer corresponds to the total number of classes we want to classify. In addition, an intermediate *dropout layer* between the dense layers allows us to regularize the model and avoid overfitting. During training, the Dropout layer randomly “turns off” a percentage of the neurons, which forces the model to be less dependent on specific features and more robust.

We also used LeNet-5, whose architecture is described in Figure 10, to compare the results.

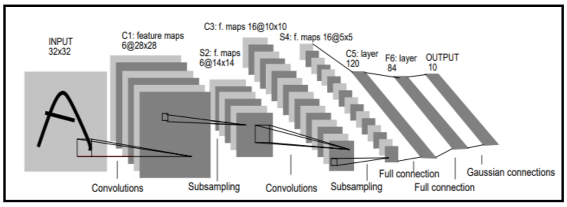


Figure 10. LeNet-5 architecture. Extracted from [24]

LeNet-5 is composed of convolutional layers, subsampling layers and fully connected layers. To adapt the LeNet-5 architecture to our data, which has an image input of size 32x256 with 3 channels, we have made some adjustments to the layers. The first layer is a convolution that applies 6 5x5 filters, followed by a pooling layer that reduces the dimensionality. Then, another convolution layer applies 16 5x5 filters, followed by another pooling layer. After these operations, the output is flattened by a Flatten layer, transforming the feature maps into a one-dimensional vector. Next, the fully connected layer has 120 neurons, and Dropout is applied to avoid overfitting. Another dense layer with 84 neurons also includes Dropout. Finally, the output layer has 4 neurons with softmax activation to perform classification. [23]

Finally, we use the VGG16 model; the architecture is described in Figure 11.

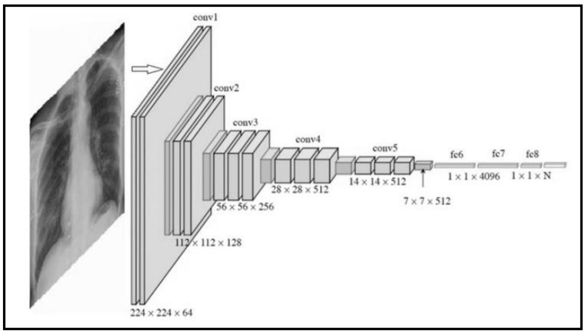


Figure 11. VGG16 architecture. Extracted from [24]

The original structure of VGG16 is characterized by having a series of convolutional layers followed by max-pooling operations, allowing the network to extract features at different levels of abstraction and culminating in fully connected layers to perform classification. [25] However, in this case, we have adapted the VGG16 architecture to our specific data (such as the input size that fits our 32x256 dimension images with three channels) by removing the top layers, keeping only the pre-trained convolutional layers, and adding new dense layers to perform classification on our dataset.

This comparison evaluates how each architecture performs when classifying our specific data.

The comparison between these three approaches allows us to determine whether a pre-trained model such as VGG16 offers significant advantages over a simpler model such as LeNet-5 or a proprietary model designed and trained directly on our data. In this way, we verify whether the spectrograms of the EMG signals effectively obtain a suitable classification for the four movements.

- *Training*

Before training the models, we prepare our data for input. Remembering that we have a total of 1000 data corresponding to the four movements, it is also essential to consider that in our case, we are using the information extracted from 8 sensors, i.e., we have a total of 8 channels. So, in the first instance, we concatenate the spectrograms corresponding to each movement.

Figure 12 shows an example of concatenated spectrograms; the image corresponds to a movement related to a subject.

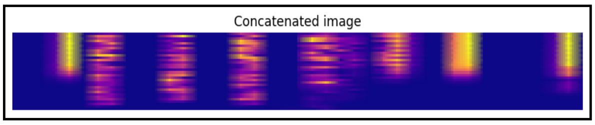


Figure 12. Example of spectrograms concatenation.

Once the concatenated spectrograms for each sensor for the different subjects and movements were obtained, models were trained.

It is important to note that when we concatenate the spectrograms, we obtain an input size of (32, 256, 3) due to the new image dimensions. 32 represents the height of the image, 256 represents the image's width, and 3 represents the color channels.

During training, we employed regularization techniques such as Dropout, optimization with the Adam algorithm, and callbacks such as Early Stopping to avoid overfitting and ensure that the model converged efficiently to an optimal solution. In addition, all models were trained with 30 epochs to compare the epoch at which they no longer achieved improved metrics. The training, validation, and test sets were distributed to evaluate the overall model performance and ensure that the results represented the different movements.

III. Results

Once the models have been trained, we compare the results obtained.

In the first instance, we compared accuracy and loss during training. Comparing these metrics is essential because accuracy is more intuitive when evaluating the quality of the predictions. At the same time, loss is more beneficial to understanding how the model behaves during optimization.

Figure 13 compares the accuracy of the three models.

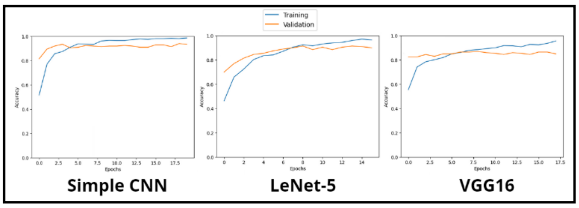


Figure 13. Comparison of the accuracy obtained by the 3 models

At the same time, in Figure 14, we compare the loss obtained during training for the three models.

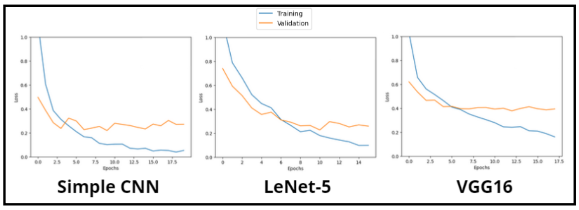


Figure 14. Comparison of the loss obtained by the 3 models

In the first comparison, LeNet-5 outperformed the other models. It obtained a total accuracy of 91%, compared to CNN Simple's 89% and VGG16's 86%. The latter had the worst results. On the other hand, for the case of loss during training, LeNet-5 also obtained better convergence between training and validation losses, which indicates that the model is generalizing better to the data without overfitting.

In addition, the Precision, F1-Score, and Recall metrics of the three models were also considered for a more accurate comparison. These metrics are fundamental to evaluating the performance of the three models, avoiding mostly false positives or negatives.

Table 2 shows the comparison of Precision, F1-Score, and Recall obtained by each of the models.

	Models		
Metric	*Simple CNN*	*LeNet-5*	*VGG16*
F1-Score	93.47%	90.42%	85.93%
Precision	93.63%	90.54%	85.95%
Recall	93.50%	90.50%	86%

Table 2. Comparison of F1-Score, Precision, and Recall obtained by each model.

Better results were obtained using the simple CNN for the F1-Score, precision, and recall. It should be noted that the difference in the accuracy of the Simple CNN and the LeNet-5 is only 2%. This means the CNN Simple will outperform the LeNet-5 model in accuracy with hyperparameter optimization. On the other hand, as in the previous case, the VGG16 model provided the worst metrics.

Finally, to finish the comparison, in Figure 15, we analyze the confusion matrices obtained in each model, thus verifying the number of True Positives in our data classification.

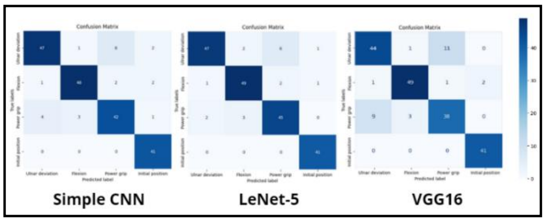


Figure 15. Confusion matrix obtained for the 3 models.

model was LeNet-5, with very little difference compared to the CNN Simple model. Conversely, the VGG16 model gave us the worst metrics, significantly different from CNN Simple and LeNet-5.

IV. Discussion

After evaluating the three models, LeNet-5, a simple CNN, and VGG16, the results showed that although all models could classify the movements accurately, their performance varied significantly depending on the metrics analyzed.

LeNet-5 obtained the best accuracy and confusion matrix results, showing that this model could correctly identify most classes in the dataset. This architecture has effectively captured the features of the EMG signals, and, given its simplicity compared to other more complex models, it may have avoided overfitting, resulting in more stable and generalizable performance. In addition, the confusion matrix suggests that LeNet-5 handled the classification between classes well, with few confounding errors between categories.

On the other hand, the simple CNN showed better metrics such as F1-score, precision, and recall. This may indicate that, while LeNet-5 was more accurate, the simple CNN was more efficient in correctly classifying the movements that belonged to each class (recall) and avoiding false positives (precision). So, as mentioned in the results section, the model may obtain better metrics by performing a hyperparameter optimization.

Although VGG16 is a more complex model and generally effective in image classification tasks, it showed the worst accuracy and results for additional metrics. This could be due to several reasons. One of the problems to emphasize is the amount of data. However, based on the logic proposed in this research, methodologies such as data augmentation can be applied to take advantage of a more robust model such as VGG16.

We concluded the work by proving that CNN-based models effectively classify muscle movements from spectrograms of EMG signals. CNN models are effective not only in classifying traditional images but also in classifying EMG signals represented as spectrograms. This reinforces the idea that spectrograms are a valuable tool for transforming temporal signals into a format that can be efficiently processed by convolutional neural networks, providing a perspective for the analysis of biomedical signals in classification applications.

References

[1] R. H. Chowdhury et al., "Surface electromyography signal processing and classification techniques," *Sensors*, vol. 13, pp. 12431–12466, 2013.

[2] D. Huang and B. Chen, "Surface EMG decoding for hand gestures based on spectrogram and CNN-LSTM," in *Proc. 2019 2nd China Symp. Cognitive Comput. Hybrid Intell. (CCHI)*, Xi'an, China, 2019, pp. 123–126.

[3] D. Stashuk, "EMG signal decomposition: How can it be accomplished and used?," *J. Electromyogr. Kinesiol.*, vol. 11, no. 3, pp. 151–173, Jun. 2001.

[4] J. V. Basmajian and C. J. De Luca, *Muscles Alive: Their Functions Revealed by Electromyography*, 5th ed. Baltimore, MD, USA: William & Wilkins, 1985.

[5] K. S. Saladin, *Anatomy & Physiology: The Unity of Form and Function*, 8th ed. New York, NY, USA: McGraw-Hill, 2020.

[6] C. J. De Luca, "Surface electromyography: Detection and recording," Delsys Inc., 2002.

[7] J. Wang et al., "Surface EMG signal amplification and filtering," *Int. J. Comput. Appl.*, pp. 15–22, 2013.

[8] S. Day, *Important Factors in Surface EMG Measurement*, Calgary, Canada: Bortech Biomedical Ltd., 2002.

[9] R. Merletti and P. A. Parker, *Electromyography: Physiology, Engineering, and Non-Invasive Applications*, 1st ed. Hoboken, NJ, USA: Wiley-IEEE Press, 2004, pp. 259–304.

[10] D. Huang and B. Chen, "Surface EMG decoding for hand gestures based on spectrogram and CNN-LSTM," in *Proc. 2019 2nd China Symp. Cognitive Comput. Hybrid Intell. (CCHI)*, Xi'an, China, 2019, pp. 123–126.

[11] J. Tryon and A. L. Trejos, "Evaluating convolutional neural networks as a method of EEG–EMG fusion," *Front. Neurorobot.*, vol. 15, no. 692183, 2021.

[12] T. Jingwei et al., "Featureless EMG pattern recognition based on convolutional neural network," *14th Int. Conf. Adv. Intell. Syst.*, 2019, pp. 1291–1297.

[13] I. Ramírez et al., "Metodología para la adquisición de señales electromiográficas en el brazo utilizando un lector de señales multicanal," *La Mecatrónica en México*, vol. 8, pp. 22–36, 2019.

[14] Thalmic Labs, *Lesson One: Getting Started with Myo*. N.d.

[15] I. Velasco et al., "Test of a Myo armband," *Rev. Cienc. Ambient. Rec. Nat.*, vol. 3, pp. 1–8, 2017.

[16] B. Rodríguez et al., "Myoelectric interfaces and related applications: Current state of EMG signal processing–A systematic review," *IEEE Access*, vol. 8, pp. 7792–7805, 2020.

[17] A. Graps, "An introduction to wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, Summer 1995.

[18] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. Burlington, MA, USA: Academic Press, 2008.

[19] W. Wang et al., "Revisiting signal processing with spectrogram analysis: EEG, ECG, and speech signals," *Future Gener. Comput. Syst.*, vol. 98, pp. 227–234, 2019.

[20] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[21] L. Cohen, *Time-Frequency Analysis*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[22] J. Huang et al., "ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network," *IEEE Access*, vol. 7, pp. 92871–92880, 2019.

[23] Y. Lecun et al., "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[24] Y. Dandi, "Detection and analysis of COVID-19 in medical images using deep learning techniques," *Sci. Rep.*, vol. 11, no. 19638, 2021.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.