HAND GESTURE CLASSIFICATION FROM ELECTROMYOGRAPHIC SIGNALS USING DEEP LEARNING

DIEGO ANDRES ESPINEL FERNANDEZ
RAÚL ANDRES HINCAPIE HERNANDEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
INGENIERÍA ELECTRÓNICA
BUCARAMANGA
2026

HAND GESTURE CLASSIFICATION FROM ELECTROMYOGRAPHIC SIGNALS USING DEEP LEARNING

DIEGO ANDRES ESPINEL FERNANDEZ
RAÚL ANDRES HINCAPIE HERNANDEZ

Thesis submitted in partial fulfillment of the requirements for the degree of Electronic Engineer

Advisor
MSc. (c) Harold H. Rodriguez

Co-advisors
MSc. Camilo Santos
Carlos A. Fajardo, Ph.D.

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2026

CONTENT

LIST OF FIGURES

LIST OF TABLES

ANNEXES

RESUMEN

TÍTULO: CLASIFICACIÓN DE GESTOS DE LA MANO A PARTIR DE SEÑALES ELECTROMIOGRÁFICAS USANDO DEEP LEARNING

AUTOR: DIEGO ANDRES ESPINEL FERNANDEZ, RAUL ANDRES HERNANDEZ HINCAPIE

PALABRAS CLAVE: EMG**,** Multimodal, Aprendizaje profundo, Latencia

DESCRIPCIÓN:

El procesamiento de señales electromiográficas (EMG) constituye un pilar fundamental en el desarrollo de interfaces hombre-máquina, específicamente para el reconocimiento de gestos y el control de prótesis mioeléctricas. No obstante, este campo enfrenta desafíos críticos como la alta variabilidad de las señales entre diferentes sujetos, la influencia de la calidad del sensor en la captura de datos y las estrictas limitaciones computacionales que surgen al intentar implementar algoritmos complejos en dispositivos embebidos que requieren una respuesta inmediata en tiempo real. Para abordar esta problemática, el presente estudio se centró en el desarrollo de modelos de aprendizaje profundo capaces de clasificar gestos de la mano con alta precisión y eficiencia. La metodología propuesta abarcó desde el preprocesamiento y la estandarización de las señales hasta el diseño de arquitecturas basadas en redes neuronales convolucionales (CNN). Se evaluaron dos enfoques: una CNN 2D que procesa la señal directa y un modelo multimodal que integra la señal estandarizada con su envolvente RMS. Estas arquitecturas fueron puestas a prueba utilizando datos de sensores de grado médico (BTS) y sensores de bajo costo (MyoWare 2.0), aplicados tanto en personas sanas como en sujetos con amputación. Los resultados obtenidos fueron sobresalientes, alcanzando métricas de precisión, recall y F1-score superiores al 91% de forma general, y llegando hasta un 99% en el caso de sujetos sanos. Un hallazgo relevante es que los sensores de bajo costo demostraron una competitividad notable frente a los sistemas médicos. Finalmente, tras aplicar técnicas de cuantización, el segundo modelo logró reducir su latencia de inferencia a menos de 0.5 ms. Este hito garantiza la viabilidad de la propuesta para ser integrada en sistemas de asistencia y rehabilitación de bajo costo con un rendimiento excepcional.

---

[*] Tesis

[**] Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunucaciones. Advisor: MSc. (c) Harold H. Rodriguez. Co-advisors: MSc. Camilo Santos; Carlos A. Fajardo, Ph.D. Electronic Engineer.

ABSTRACT

TITLE: Hand Gesture Classification from Electromyographic Signals Using Deep Learning

AUTOR: DIEGO ANDRES ESPINEL FERNANDEZ, RAUL ANDRES HERNANDEZ HINCAPIE

Keywords: EMG, Multimodal, Deep Learning, Latency

DESCRIPTION:

Electromyographic (EMG) signal processing constitutes a fundamental pillar in the development of human-machine interfaces, specifically for gesture recognition and the control of myoelectric prostheses. However, this field faces critical challenges such as high signal variability between different subjects, the influence of sensor quality on data capture, and strict computational limitations that arise when attempting to implement complex algorithms on embedded devices requiring immediate real-time response. To address these issues, this study focused on developing deep learning models capable of classifying hand gestures with high precision and efficiency. The proposed methodology spanned from signal preprocessing and standardization to the design of architectures based on convolutional neural networks (CNN). Two approaches were evaluated: a 2D CNN that processes the raw signal and a multimodal model that integrates the standardized signal with its RMS envelope. These architectures were tested using data from medical-grade sensors (BTS) and low-cost sensors (MyoWare 2.0), applied to both healthy individuals and subjects with amputations. The results obtained were outstanding, reaching precision, recall, and F1-score metrics exceeding 91% overall, and up to 99% in the case of healthy subjects. A relevant finding is that the low-cost sensors demonstrated remarkable competitiveness compared to medical systems. Finally, after applying quantization techniques, the second model managed to reduce its inference latency to less than 0.5 ms. This milestone guarantees the feasibility of the proposal for integration into low-cost assistance and rehabilitation systems with exceptional performance.

---

* Thesis

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunucaciones. Advisor: MSc. (c) Harold H. Rodriguez. Co-advisors: MSc. Camilo Santos; Carlos A. Fajardo, Ph.D. Electronic Engineer.

INTRODUCTION

The absence of a hand directly reduces functional autonomy and social participation, restricting the ability to perform everyday activities such as writing, eating, or dressing[1]. Several studies have shown that these physical restrictions can also impair psychological health, leading to problems such as anxiety, depression, and diminished quality of life[2], [3]. Limb loss may arise from diverse causes, including traumatic amputations (road or industrial accidents, armed conflict), vascular or metabolic complications (e.g., diabetes-related ischemia), severe infections, tumor resection, or congenital malformations; the origin and context of the impairment influence both clinical presentation and rehabilitation needs.

The connection between practical limitation and psychological deterioration operates through several pathways: inability to carry out daily tasks undermines independence and self-efficacy, creates dependence on caregivers for basic activities, and often entails the forfeiture of work or interpersonal roles, all of which increase stress, hopelessness and social withdrawal. Economic barriers and constrained access to rehabilitation services and appropriate prostheses further hinder participation and recovery. Taken together, decreased independence in daily life, deprivation of roles, and impeded access to rehabilitative resources explain why hand absence frequently results in sustained negative effects on mental health and quality of life. In this context, the development of assistive technologies aimed at partial restoration of motor function is an active area of research with high societal impact.

---

[1] FONT-JIMÉNEZ, M. et al. Quality of life and functionality in upper limb amputees. En: Disability and Rehabilitation. 2016. vol. 38, no. 20, p. 1954–1961.

[2] GARCÍA, J. y PÉREZ, M. Psychological impact of limb amputation. En: Revista de Psicología de la Salud. 2015. vol. 27, no. 2, p. 85–92.

[3] JOHNSON, V. et al. Mental health outcomes following limb loss. En: Journal of Rehabilitation Research. 2016. vol. 53, no. 4, p. 473–482.

One of the most widely used approaches in active prostheses and motor assistance systems is the use of electromyographic (EMG) signals, which reflect the electrical activity generated by muscle contraction and allow the user's motor intention to be inferred[4]. EMG-based systems have been extensively studied for hand gesture recognition and myoelectric prosthesis control because they offer a non-invasive alternative that is compatible with real-time applications[5].

Nevertheless, these signals present inherent challenges, such as high inter- and intra-subject variability, sensitivity to noise, and dependence on the sensor used, which makes it difficult to obtain robust and generalizable classification models[6]. Traditionally, gesture recognition based on EMG has been approached through manual feature extraction in the temporal and frequency domains, combined with conventional classifiers [7]. However, in recent years, deep learning techniques have demonstrated a superior ability to learn discriminative representations directly from raw or minimally processed data, reducing the dependence on manual feature design stages[8]. In particular, convolutional neural networks (CNNs) have shown outstanding performance in gesture classification from multichannel EMG signals by automatically capturing temporal patterns and relationships between channels [9], [10].

---

[4] MERLETTI, R. y PARKER, P. Electromyography: Physiology, Engineering, and Non-Invasive Applications. Wiley-IEEE Press, 2004.

[5] ENGLEHART, K. y HUDGINS, B. A robust, real-time control scheme for multifunction myoelectric control. En: IEEE Transactions on Biomedical Engineering. 2003. vol. 50, no. 7, p. 848–854.

[6] FARINA, D. et al. The extraction of neural strategies from the surface EMG. En: Journal of Applied Physiology. 2004. vol. 96, no. 4, p. 1486–1495.

[7] HUDGINS, B., PARKER, P. y SCOTT, R. A new strategy for multifunction myoelectric control. En: IEEE Transactions on Biomedical Engineering. 1993. vol. 40, no. 1, p. 82–94.

[8] LECUN, Y., BENGIO, Y. y HINTON, G. Deep learning. En: Nature. 2015. vol. 521, p. 436–444.

[9] ATZORI, M. et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. En: Scientific Data. 2014. vol. 1.

[10] DING, Z. et al. Gesture recognition based on sEMG using convolutional neural networks. En: Sensors. 2018. vol. 18, no. 5.

Despite these advances, the practical application of CNN-based models in myoelectric control systems poses additional constraints. For use in real-world scenarios, such as prosthetics or assistive interfaces, models must meet strict requirements for latency, memory consumption, and stability in continuous operation[11]. This is particularly relevant when considering different types of acquisition devices, ranging from high-fidelity commercial systems to low-cost sensors, which differ in resolution, noise, and number of channels[12]. Consequently, it is not only necessary to achieve high performance in terms of accuracy, but also to evaluate the computational feasibility of the model for deployment in resource-constrained environments.

In this context, this thesis focuses on the design, implementation, and evaluation of a multi-class hand gesture classification system based on sEMG signals, using CNN-1D and CNN-2D architectures. The study considers different datasets acquired with BTS and MyoWare sensors and adopts a methodological flow aimed at preserving experimental validity through controlled partitioning schemes and cross-validation. The performance of the models is analyzed using standard multi-class classification metrics—accuracy, recall, and F1-score—along with confusion matrices, and is complemented by a study of latency, inference times, and computational resource consumption, including quantized versions of the model for efficient execution.

---

[11] SCHEME, T. y ENGLEHART, K. Electromyogram pattern recognition for control of powered upper-limb prostheses. En: Journal of Rehabilitation Research & Development. 2011. vol. 48, no. 6, p. 643–660.

[12] PHINYOMARK, A. et al. Feature extraction and selection for myoelectric control. En: Expert Systems with Applications. 2012. vol. 39, no. 8, p. 7420–7431.

1. OBJECTIVES

1.1. GENERAL OBJECTIVE

Develop a deep learning model to classify hand gestures using electromyographic (EMG) signals.

1.2. SPECIFIC OBJECTIVES

Preprocess a database that allows training a deep learning model for the classification of hand gestures.

Select a state-of-the-art deep learning architecture for the classification of hand gestures based on EMG signals..

Evaluate model performance using classification metrics such as F1 score, recall, cross-validation, and precision.

2. PRELIMINARY STUDIES

The evolution of hand gesture recognition has transitioned from classical machine learning to deep learning, primarily driven by the ability of Convolutional Neural Networks (CNN) to extract hierarchical features directly from raw surface electromyography (sEMG) signals. (Tian et Luo)[13] established a baseline for this approach by demonstrating that a 1D-CNN, optimized with specific sliding window parameters, can achieve accuracies exceeding 98% in multi-class classification, significantly outperforming traditional models like SVM and LSTM.

The architectural choice between 1D and 2D representations remains a pivotal discussion in the literature. While 1D-CNNs focus on the temporal dynamics of individual muscle fibers[14],[15], 2D-CNNs allow for the integration of spatial information by treating sEMG signals as heatmaps or spectrograms. Geng et al.[16] and Wei et al.[17] have shown that multi-view 2D-CNNs can capture correlations between adjacent electrodes that 1D models might overlook, particularly in high-density sEMG setups.However, for systems with fewer sensors, such as those using MyoWare technology, 1D-CNNs often maintain superior computational efficiency without sacrificing precision[18] .

---

[13] TIAN, J. y LUO, X. EMG-Based Gesture Classification Using Convolutional Neural Networks. En: CONFERENCIA INTERNACIONAL SOBRE TECNOLOGÍA DE SENSORES E INFORMACIÓN (5: 2025: Ciudad de la conferencia). Memorias. IEEE, 2025. p. 1288–1293.

[14] GENG, H. et al. Gesture recognition using high-density surface EMG and deep convolutional neural networks. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2016. vol. 24, no. 11, p. 1156–1164.

[15] ZHONG, W. y JIANG, X. Deep Feature Learning from Electromyographic Signals for Gesture Recognition Systems. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2024. vol. 32, p. 435–446.

[16] WEI, W. et al. A Multi-View CNN Framework for Gesture Recognition from Surface EMG Signals. En: IEEE Transactions on Cybernetics. 2019. vol. 49, no. 4, p. 1150–1162.

[17] DU, Y. et al. Surface EMG-based inter-session gesture recognition using a deep convolutional neural network. En: Sensors. 2017. vol. 17, no. 9, p. 2046.

[18] ZHAI, T. et al. Self-Recalibrating Surface EMG Pattern Recognition for Neuroprosthesis Control Based on Convolutional Neural Network. En: Frontiers in Neuroscience. 2017. vol. 11, p. 379.

The use of time-domain descriptors integrated into fully connected neural networks (FCNNs) remains a highly efficient alternative for real-time gesture recognition. Neacsu et al.[19] propose the use of features such as Root Mean Square (RMS) and Mean Absolute Value (MAV) because they are simple and computationally inexpensive to obtain. Unlike time-frequency analysis methods (such as Wavelet Transform), which require longer computation times without guaranteeing significant performance improvements, these descriptors allow muscle activation to be captured effectively with minimal computational cost. Under this approach, their system achieved 99.78% accuracy with a total processing time of just 4.4 ms, allowing for near-instantaneous inference for the user.

A significant portion of recent research focuses on the clinical applicability of these models for prosthetic control in subjects with limb loss. The signal characteristics of amputees differ fundamentally from healthy subjects due to muscle transposition and atrophy[20],[21]. Atzori et al.[22], through the Ninapro database, have provided a benchmark showing that while deep learning improves performance, the gap in classification accuracy between healthy subjects and amputees remains a challenge requiring adaptive architectures.

---

[19] NEACSU, A. A. et al. Automatic EMG-based Hand Gesture Recognition System using Time-Domain Descriptors and Fully-Connected Neural Networks. En: CONFERENCIA INTERNACIONAL SOBRE TELECOMUNICACIONES Y PROCESAMIENTO DE SEÑALES (42: 2019: Budapest). Memorias. IEEE, 2019. p. 232–235.

[20] CASTELLINI, C. et al. Proceedings of the first workshop on Peripheral Machine Interfaces: Going beyond traditional surface electromyography. En: Frontiers in Neurorobotics. 2014. vol. 8, p. 22.

[21] NILWONG, S. et al. Deep Learning-Based EMG Gesture Recognition for Amputee Subjects. En: CONFERENCIA INTERNACIONAL IEEE SOBRE ROBÓTICA Y BIOMIMÉTICA (ROBIO) (2022: Jinghong). Memorias. IEEE, 2022. p. 145–150.

[22] ATZORI, M. et al. Building the Ninapro database: A resource for the biorobotics community. En: CONFERENCIA INTERNACIONAL IEEE RAS & EMBS SOBRE ROBÓTICA BIOMÉDICA Y BIOMECATRÓNICA (4: 2012: Roma). Memorias. Roma: IEEE, 2012. p. 1258–1265.

To mitigate this, Zia ur Rehman[23] and Xie et al.[24] proposed transfer learning and multiday training strategies, proving that CNNs can compensate for signal degradation in transradial amputees. The hardware environment also plays a crucial role in signal quality. Research comparing low-cost sensors, like the MyoWare 2.0, against clinical systems like BTS or Delsys, indicates that while professional systems offer higher signal-to-noise ratios, deep learning models can act as equalizers. Mendes Jr. et al.[25] and Nasser-Cortés[26] evaluated these platforms, concluding that MyoWare is sufficient for gesture recognition if robust preprocessing and deep feature extraction are applied. This is further supported by Pan et al.[27], who utilized BTS systems to validate that the physiological patterns identified by clinical devices are replicable in affordable sensing environments when using 1D-CNN frameworks. Likewise, progress has been made toward model explainability and immediate response. Hu et al.[28] explored attention mechanisms within CNNs to identify which muscle groups contribute most to a specific gesture.

---

[23] ZIA UR REHMAN, M. et al. Multiday EMG-based gesture recognition using deep learning for amputees. En: IEEE Transactions on Instrumentation and Measurement. 2018. vol. 67, no. 8, p. 1914–1924.

[24] XIE, Z. et al. High-Performance EMG Gesture Recognition for Amputees via Residual CNN. En: IEEE Transactions on Cognitive and Developmental Systems. 2022. vol. 14, no. 3, p. 880–892.

[25] MENDES JR., J. et al. Evaluation of MyoWare and Delsys sensors for sEMG-based control. En: Biomedical Signal Processing and Control. 2020. vol. 57, p. 101765.

[26] NASSER-CORTÉS, J. et al. Comparative Analysis of Low-Cost and Clinical EMG Sensors for Gesture Recognition. En: IEEE Sensors Journal. 2025. vol. 25, no. 2, p. 45–55.

[27] PAN, L. et al. Clinical Validation of sEMG Pattern Recognition Using BTS Bioengineering Systems. En: Journal of NeuroEngineering and Rehabilitation. 2023. vol. 20, no. 1, p. 112.

[28] CÔTÉ-ALLARD, U. et al. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2019. vol. 27, no. 4, p. 760–771.

These developments, along with real-time implementation studies[29],[30], suggest that integrating diverse hardware and multi-dimensional architectures is viable for myoelectric prostheses. However, the increasing sophistication of these architectures raises the demand for computational resources, which presents difficulties for direct implementation in portable systems. The use of multimodal models in myoelectric gesture recognition is useful because it enables the integration of complementary sources of information that describe different aspects of human movement, reducing the ambiguity present when only a single modality is used. Several studies have shown that combining sEMG signals with derived representations such as the RMS envelope, as well as with kinematic or visual signals, improves classification stability under noise, inter-subject variability, and electrode placement shifts[31]. Recent reviews indicate that multimodal fusion facilitates the capture of both muscle activation patterns and temporal dynamics, increasing discrimination between similar gestures when deep learning architectures are applied[32]. Experimental works further demonstrate that multimodal approaches achieve more consistent performance in real-world and real-time scenarios compared to unimodal models, although at the cost of moderately increased computational requirements[33].

---

[29] HU, Y. et al. Attention-based CNN for Surface EMG Gesture Recognition. En: IEEE Journal of Biomedical and Health Informatics. 2018. vol. 22, no. 6, p. 1747–1756.

[30] PIZZOLATO, S. et al. Comparison of EMG-based classification and regression for prosthetic control in real-time. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2017. vol. 25, no. 9, p. 1629–1639.

[31] ZHANG, H., SIDEL MOCTAR, S. M., BOUDAOUD, S. e RIDA, I. A comprehensive review of sEMG-IMU sensor fusion for upper limb movements pattern recognition. En: Information Fusion. 2026. vol. 125, p. 103422.

[32] LI, W., SHI, P. y YU, H. Gesture Recognition Using Surface Electromyography and Deep Learning for Prostheses Hand: State-of-the-Art, Challenges, and Future. En: Frontiers in Neuroscience. 2021. vol. 15, p. 621885.

[33] ZANDIGOHAR, M. et al. Multimodal fusion of EMG and vision for human grasp intent inference in prosthetic hand control. En: Frontiers in Robotics and AI. 2024. vol. 11, art. 1312554.

As these hybrid and deep architectures increase in complexity, current research has identified a new critical challenge: the transition from high-performance software to resource-constrained embedded hardware. In this context, model quantization has emerged as a pivotal trend in the state of the art to ensure the commercial and portable viability of modern prostheses [34]. Authors such as Gholami et al.[35] argue that model reduction through techniques like TensorFlow Lite (TFLite) is essential for executing complex networks on microcontrollers without compromising accuracy. This approach, applied in recent developments such as the EffiE project, allows for the conversion of model weights into low-precision formats (INT8), achieving drastic reductions in both model size and inference latency. Consequently, this facilitates the deployment of prototypes on platforms like Sony Spresense, reaching response times compatible with real-time control, effectively closing the gap between clinical computing power and the end-user's need for mobility.

---

[34] LU, J. et al. EffiE: Efficient Convolutional Neural Network for Real-Time EMG Pattern Recognition System on Edge Devices. En: CONFERENCIA INTERNACIONAL IEEE/EMBS SOBRE INGENIERÍA NEURAL (11: 2023: Baltimore). Memorias. IEEE, 2023.

[35] AMERI, A. et al. Real-Time Simultaneous and Proportional Control of Multiple Degrees of Freedom from Surface EMG Using CNN. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. Diciembre, 2017. vol. 25, no. 12, p. 2445–2454.

3. METHODOLOGY

3.1 DATASET

The data used in this research was provided by Prof. Luis Bautista, a professor at the Industrial University of Santander (Barbosa Campus). These records are private and were originally obtained under institutional protocols for research purposes; therefore, this study is limited to the analysis and processing of the provided information, as the signal acquisition stage was conducted previously. In this process, Prof. Bautista was responsible for defining the specific electrode placement on the participants' forearms to ensure an appropriate capture of muscle activity for each system. Furthermore, the participant provided their informed consent for the use of their electromyographic data in research and academic publications.

For the datasets obtained with the MyoWare 2.0 system, four sensors were positioned equidistantly around the middle circumference of the forearm. In contrast, the BTS medical-grade system utilized six sensors distributed symmetrically: three electrodes located in the anterior section (flexor area) and three in the posterior section (extensor area). These configurations were designed to record electrical potential variations during the execution of static gestures, where the subject maintains a specific grip position while applying constant pressure. It is important to note that the signals provided are not raw data directly from the electrodes; instead, they underwent a hardware conditioning process performed by Prof. Bautista to optimize the quality of the recordings.

The data capture was based on a protocol where each gesture lasted approximately three seconds, corresponding to the duration of the pressure developed during the grip. Data acquisition was active only during this period of pressure and was suspended when the hand returned to a relaxed state, a procedure that was applied consistently across all datasets. For the purposes of this study, a sample is defined as a 200 ms analysis window. All available recordings were segmented into these fixed-length windows to serve as individual instances for the deep learning models, a design decision intended to facilitate real-time responsiveness.

Dataset 1 consists of EMG recordings from subject 2 using BTS medical sensors. This system provided six channels of EMG sampled at 1 kHz which, before segmentation, resulted in a total of 3517 instances of 3 seconds time. This set includes six gestures: G1 (index with thumb), G2 (middle finger with thumb), G3 (ring finger with thumb), G4 (little finger with thumb), G5 (closed fist), and G6 (rest). Dataset 2 comprises recordings from Subject 2 using the MyoWare 2.0 system with four channels at a sampling frequency of 1 kHz. These data were collected over two sessions and segmented into 200 ms windows, totaling 33,545 instances of 200ms seconds time. Finally, Dataset 3 contains recordings from Subject 1 using the same MyoWare 2.0 configuration, collected across four sessions for a total of 61,243 instances of 200ms seconds time.

During the inspection of the MyoWare 2.0 recordings, a baseline offset was identified in one of the acquisition sessions, which was addressed during the preprocessing stage. Table 1 presents the distribution of the number of instances time per gesture class for each dataset. These values correspond to the hardware-conditioned and segmented EMG data prior to the application of further digital signal processing techniques, providing an overview of the data availability and class balance used for model training and evaluation.

Table 1. Distribution of 3s instances for dataset 1 and 200 ms instances per gesture class in dataset 2 and 3, prior to preprocessing.

Class	Dataset 1	Dataset 2	Dataset 3
# instances G1	583	4764	8707
# instances G2	578	4961	8707
# instances G3	591	4764	8707
# instances G4	591	4764	8904
# instances G5	600	4764	8804
# instances G6	574	4764	8707
Total instances	3517	33545	61243

3.2 DATA PREPROCESSING

3.2.1 DC Offset Elimination (Signal Centering) Raw EMG signals often exhibit a constant voltage shift or DC offset, derived from the electrode-skin interface and sensor electronics. To correct this, a mean subtraction process was implemented. This method consists of calculating the average value of the signal across the entire time window and subtracting it from each individual sample:

$$y[n] = x[n] - \frac{1}{N}\sum_{i=1}^{N} x[i]$$

This procedure ensures that the signal is centered at zero millivolts, eliminating biases that could saturate the activation functions of the neural network and facilitating faster convergence during training[36].

.

[36] DE LUCA, C. J. The Use of Surface Electromyography in Biomechanics. En: Journal of Applied Biomechanics. 1997. vol. 13, no. 2, p. 135–163.

3.2.2 Elimination of overlap to prevent data leakage For datasets 2 and 3, the original windows have a duration of 200 ms with an overlap of 150 ms between consecutive windows, which means that the time stride between the start of one window and the next is 50 ms. Given this design, each window shares 150 ms of signal with the immediately following window; therefore, if overlapping windows are distributed between training and testing, part of the same temporal information would appear in both partitions, introducing data leakage and overestimating the model's performance.

To eliminate this risk, the set of overlapping windows was converted into a set of non-overlapping windows, maintaining the same duration of 200 ms. Specifically, the following procedure was followed: from the sequence of windows started every 50 ms, only those whose start is separated by at least 200 ms from the start of the previous retained window were retained. In practical terms, if the original windows start at 0 ms, 50 ms, 100 ms, 150 ms, 200 ms, 250 ms, ..., those starting at 0 ms, 200 ms, 400 ms, ... were retained (i.e., the distance between the starts of retained windows is equal to the window length). This ensures that no final window shares time samples with another final window, eliminating the possibility of identical signal instants appearing simultaneously in training and testing.However, it is important to note that this filtering process significantly reduces the total number of samples available in the datasets, as only one out of every four original windows is preserved.

3.2.3 Sample duration adjustment (Dataset BTS) The BTS sensor data originally had a duration of 3000 ms per sample. Each sample was segmented into 200 ms windows without overlap, resulting in 15 windows per execution. This segmentation generates multiple short-duration instances for each record, allowing for real-time processing and inference. Table 2 shows the final count of windows/segments per class, resulting from preprocessing (offset correction, overlap removal, and temporal segmentation). The values shown correspond to the samples actually used in the training, validation, and testing partitions of the model.

Table 2. Distribution of 200 ms instances per gesture class in each dataset, after preprocessing.

Class	Dataset 1	Dataset 2	Dataset 3
# instances G1	8745	1191	2178
# instances G2	8670	1241	2178
# instances G3	8865	1191	2178
# instances G4	8865	1191	2227
# instances G5	9000	1191	2202
# instances G6	8610	1191	2178
Total instances	**52755**	**7196**	**13141**

3.2.4 Standardization In Models 1 and 2, Z-score normalization was used under a global channel scheme (represented by the structure (1, 1, C)). A single mean and standard deviation were calculated for each channel by averaging all values over time and across all samples in the training set. This procedure ensures that each channel has a uniform scale, regardless of when muscle activation occurs within the time window. These global statistics were used to standardize the validation and test partitions, maintaining the integrity of the experiment.

the transformation applied is defined by the formula:

$$z = \frac{x-\mu}{\sigma}$$

where μ corresponds to the mean and σ to the standard deviation obtained from the training set.

3.3 FEATURE EXTRACTION

The RMS envelope allows for stable estimation of muscle activation level, as it quantifies the intensity of the contraction and highlights the relevant temporal variations of the sEMG signal. Therefore, it is an informative feature for classification models and myoelectric control systems, as it enhances the energy of the signal and attenuates high-frequency noise. It is also computationally simple and suitable for real-time processing[37],[38].

The implementation followed the same preprocessing order used for the raw inputs: first, the signal was standardized per channel (by subtracting the mean and dividing by the standard deviation calculated in the training set), then the RMS envelope was calculated. The calculation was performed by squaring the signal, applying a moving average of length W.

taking the square root of the local average, according to:

$$RMS_W[n] = \sqrt{\frac{1}{W} \sum_{k=n-[\frac{W}{2}]}^{n+[\frac{W}{2}]} x_k^2}$$

where x_k is the k-th discrete sample of the EMG signal, W is the length of the sliding window (number of samples), n is the central time index.

[37] DE LUCA, C. J. The use of surface electromyography in biomechanics. En: Journal of Applied Biomechanics. 1997. vol. 13, no. 2, p. 135–163.

[38] PHINYOMARK, A., KHUSHABA, R. N. y SCHEME, E. Feature extraction and selection for myoelectric control based on wearable EMG sensors. En: Sensors. 2018. vol. 18, no. 5, art. 1615

3.4 NEURAL NETWORK ARCHITECTURE

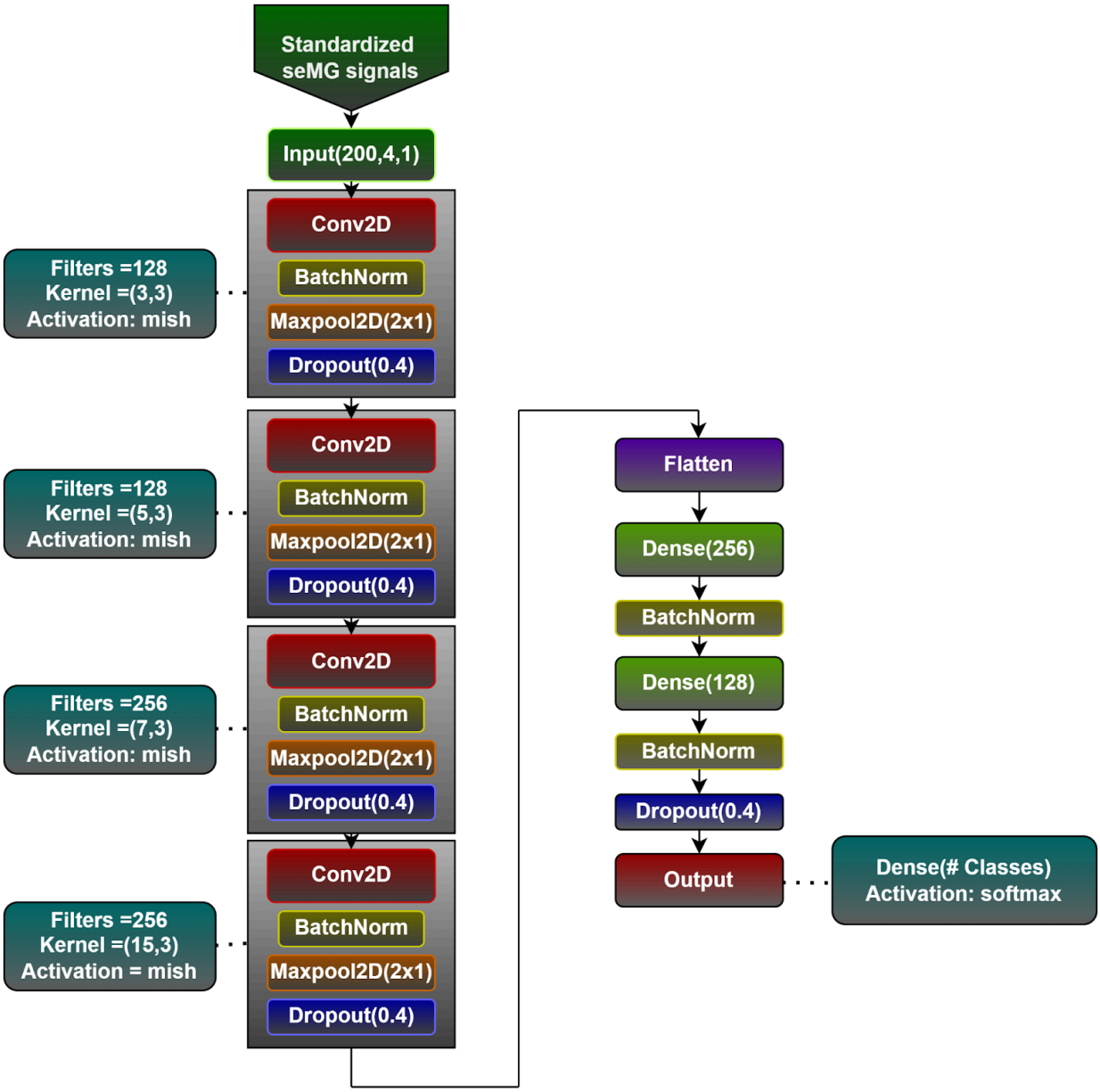
3.4.1 MODEL CNN2D The design of the 2D-CNN model implemented in this study is based on the architecture proposed by Fukano et al.[39], who utilized a two-dimensional convolutional neural network for gesture recognition using surface EMG data. This foundation was adapted and optimized for the specific characteristics of the datasets used in this research.

The convolutional neural network in Figure 1 processes standardized EMG signals with an input of (200, 4, 1). The architecture is organized into four feature extraction blocks, each of which integrates a 2D convolutional layer, batch normalization (BatchNorm), Maxpool2D of (2, 1), and a Dropout layer of 0.4. The blocks use filters that scale from 128 to 256 and kernels that vary from 3x3 to 15x3, always using the Mish activation function.

The 0.4 dropout applied to each block randomly deactivates 40% of the neurons during training. This technique is used to force the network to learn global patterns rather than memorizing specific details of the signal noise, acting as a regularization mechanism that prevents overfitting and ensures that the model can generalize to new data.
After extraction, a Flatten layer connects to the classification phase, which consists of two dense layers of 256 and 128 neurons. Both include batch normalization and Mish activation. In the last dense layer, a Dropout of 0.4 is applied again before the Output layer, reinforcing the robustness of the final decision before delivering the result.

[39] FUKANO, K. et al. Aprendizaje profundo para el reconocimiento de gestos basado en datos EMG de superficie. En: CONFERENCIA INTERNACIONAL SOBRE SISTEMAS MECATRÓNICOS AVANZADOS (ICAMechS) (2021: Tokio). Memorias. IEEE, 2021. p. 41–45.

Figure 1. Structure of the Conv2D block-based deep learning model, the model input can have a dimension of 200×4 (Dataset 2 and 3) or 200×6 (Dataset 1). The model output corresponds to C classes and uses the softmax activation function.



Model 1 was trained using stratified cross-validation with seven partitions on 80% of the available data, while the remaining 20% was reserved for final evaluation. It was configured for a maximum of 150 epochs with a batch size of 64 samples. To manage this process dynamically, three control mechanisms based on validation error monitoring were implemented. The first of these is an early stopping system, which automatically stops training if the validation error does not improve for 15 consecutive epochs. This

mechanism ensures that the model does not continue to process data unnecessarily once it has stopped learning, while also restoring the weights that allowed it to achieve the best performance recorded.

A learning rate reduction callback was used to adjust the optimization process. This mechanism monitors the validation error and, when it remains unchanged for five consecutive epochs, reduces the learning rate by half, allowing finer parameter updates and improving convergence.

Finally, the ModelCheckpoint callback was used as an automatic model saving protocol. This mechanism continuously monitors the validation error and updates the model file only when an improvement over previous results is detected. In this way, at the end of the training process, the availability of the model version that achieved the best performance is guaranteed, reflected in the highest accuracy and lowest validation error throughout all iterations.

3.4.2 MODEL MULTIMODAL The idea of processing the raw signal and its envelope in parallel was adopted following works that show fusion of complementary representations — the raw signal to capture fine temporal patterns and the envelope to represent amplitude dynamics — improves gesture discrimination and robustness to noise and inter-subject variability. Therefore, Figure 2 implements two parallel pathways that extract and fuse these two sources of information, inspired by the observations and recommendations in the deep-learning literature applied to sEMG.[40]

[40] LI, W., SHI, P. y YU, H. Gesture Recognition Using Surface Electromyography and Deep Learning for Prostheses Hand: State-of-the-Art, Challenges, and Future. En: Frontiers in Neuroscience. 2021. vol. 15, p. 621885.

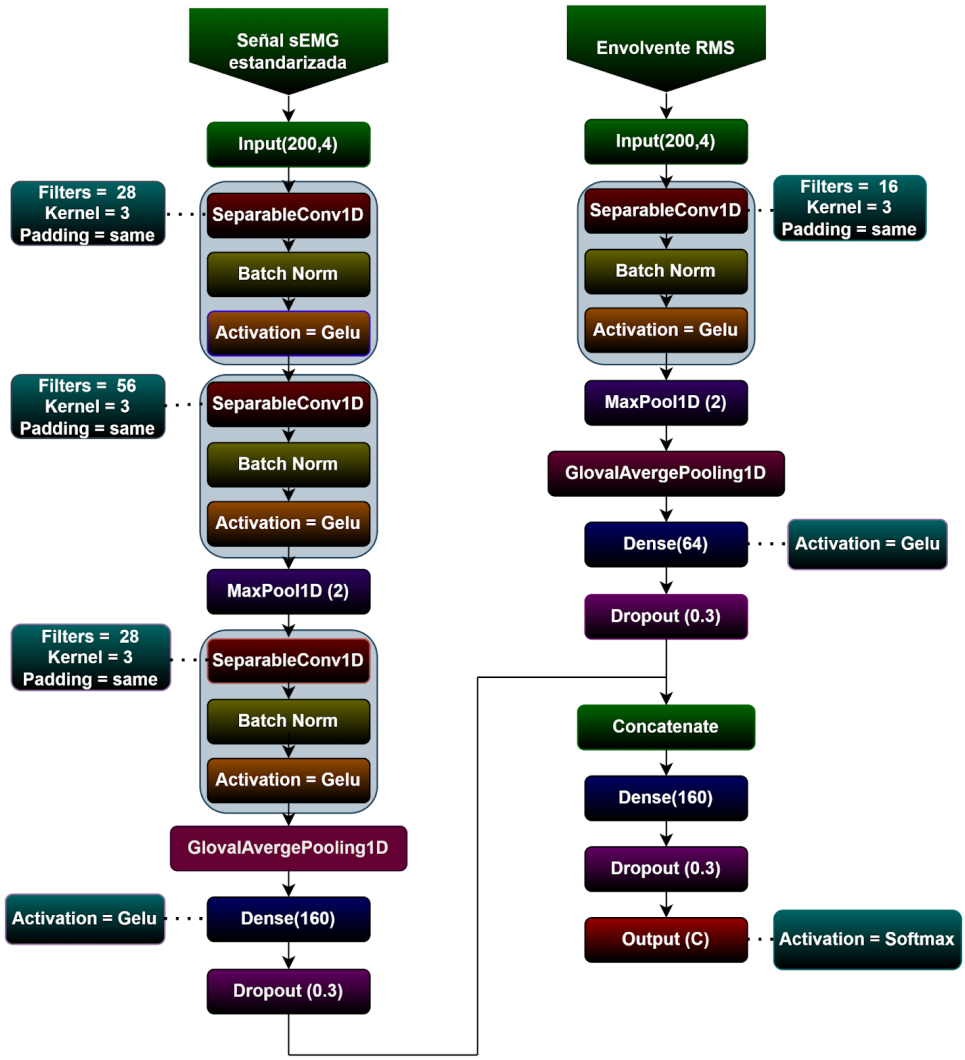
Figure 2 shows the proposed architecture. The two inputs correspond to the same acquisition window and share identical temporal length and number of channels: (1) the channel-standardized sEMG signal and (2) its RMS envelope computed over the same window.

The main branch processes the standardized signal using separable one-dimensional convolutions with small kernels; each block includes batch normalization and GELU activation. Temporal resolution is reduced via max pooling and an additional convolutional stage is added to capture higher-level temporal patterns. The branch output is summarized with global average pooling and projected to a feature vector by a dense layer; dropout (30%) is applied to mitigate overfitting.

The secondary branch operates on the RMS envelope. It contains a separable convolution followed by batch normalization, GELU activation and max pooling; after global average pooling, a dense layer with dropout produces a compact representation focused on the signal’s amplitude dynamics.

The representations from both branches are concatenated and passed through a fusion dense layer with GELU activation and dropout before the output layer. The final layer applies softmax to produce the probability distribution over classes, enabling multi-class classification of the seven gestures considered.

Figure 2. Model 2 employs a dual-input architecture, where one branch processes the standardized sEMG signal and the other the RMS envelope of the same signal. Depending on the dataset used, the model input can have a dimension of 200×4 (Datasets 2 and 3) or 200×6 (Dataset 1), the model output corresponds to class C and uses the softmax activation function.



The outputs are concatenated and pass through dense layers and dropout to the output layer.

Model 2 was trained using seven-partition stratified cross-validation on 80% of the available data, while the remaining 20% was reserved for final evaluation. Each partition was trained for a maximum of 150 epochs using the Adam optimizer and the categorical

cross-entropy loss function with label smoothing, with a batch size of 64 samples. To improve model generalization and training stability, early stopping strategies were applied based on validation accuracy, learning rate reduction upon validation loss plateauing, and storing the best model according to validation performance. Final predictions on the test and external data sets were obtained using an ensemble scheme, averaging the outputs of the models trained on each partition.

3.5 PERFORMANCE ASSESSMENT

The allocation of samples to the training and test sets was performed respecting the temporal structure of the data. Each gesture lasts three seconds and was segmented into 15 consecutive 200 ms windows; however, because the windows belonging to the same execution exhibit a high temporal correlation, random partitioning at the window level was not performed. Instead, the division process was carried out at the level of complete executions, alternating between four consecutive three-second blocks (equivalent to 60 windows) for the training set, followed by a complete three-second block (15 windows) for the test set. This pattern was repeated throughout the entire dataset, allowing for an 80% training and 20% test ratio without introducing information leakage between the two sets.

For model validation, a GroupKFold cross-validation strategy with seven partitions (7 k-folds) was used to avoid correlation between training and validation data. Since 200 ms windows belonging to the same 3-second window are highly correlated, these were grouped and treated as a single unit during the cross-validation process. In this way, in each GroupKFold iteration, all 200 ms windows associated with the same 3-second run were assigned exclusively to either the training or validation set, but never to both simultaneously, ensuring a more realistic evaluation of model performance and preventing information leakage between datasets that could artificially inflate its performance.

During cross-validation training, the weights corresponding to the model's best performance were automatically saved in each of the seven k-folds. At the end of the process, the performance of all k-folds was compared using the validation accuracy metric (val_accuracy), and the model associated with the best-performing k-fold was selected as the final model. Additionally, to analyze the learning behavior, the training and validation loss functions were plotted for each k-fold independently. Finally, the selected model was evaluated on the test set, where the accuracy, recall, and F1-score metrics were calculated for each gesture, and the confusion matrix was constructed, allowing for a detailed analysis of the system's performance in classifying the different gestures.

3.6 QUANTIZATION OF DEEP LEARNING MODELS

After training, all the developed models were quantized and converted to the TensorFlow Lite (TFLite) format. Quantization is a model optimization technique that reduces the numerical precision of the parameters and computations, typically from 32-bit floating point (float32) to lower-precision representations such as 16-bit floating point (float16) or 8-bit integers (int8).

The main purpose of quantization is to significantly reduce the model size and improve inference efficiency without causing a substantial degradation in classification performance. By using lower-precision arithmetic, the model requires less memory and fewer computational resources, which leads to faster inference times and lower power consumption. These characteristics are especially important for real-time applications and for deployment on embedded systems or low-power hardware, such as microcontrollers or wearable devices.

In this work, the conversion to TFLite enables the trained models to be executed in lightweight environments while preserving their learned behavior. The quantized models are therefore more suitable for practical implementation in assistive and wearable systems, where latency and computational efficiency are critical constraints.

4. RESULTS

System performance is presented using metrics per gesture and per dataset: Accuracy, Precision, Recall, and F1 score, calculated from the best-performing model selected for each fold (epoch weights with the highest accuracy in the validation). The distributions by class in the analyzed datasets are homogeneous, without significant imbalance between gestures; For this reason, the interpretation focuses primarily on the accuracy metric. Table 3 present the gesture performance metrics obtained from the selected models.

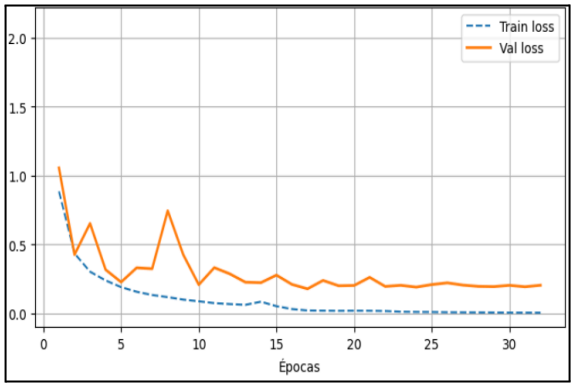
Table 3. Class-wise performance metrics for Model 1 and Model 2 across the three evaluated datasets. For each dataset (Dataset 1, Dataset 2, and Dataset 3), the table reports precision (Prec.), recall (Rec.), and F1-score (F1) for each gesture class (G1–G7), followed by the overall accuracy and the macro and weighted averages. The results for both models are presented side by side, allowing a direct comparison of their classification performance within each dataset and across different gesture classes.

Class	Model 1			Model 2		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Dataset 1						
G1	0.9778	0.9851	0.9814	0.9600	0.9700	0.9600
G2	0.9617	0.9606	0.9611	0.9200	0.9600	0.9400
G3	0.9672	0.9158	0.9408	0.9500	0.9400	0.9400
G4	0.8903	0.9305	0.9099	0.9400	0.8200	0.8700
G5	0.9389	0.9394	0.9392	0.8800	0.9500	0.9100
G6	0.9953	0.9965	0.9959	1.0000	0.9900	1.0000
Accuracy		0.9543			0.9400	
Macro avg	0.9552	0.9547	0.9547	0.9400	0.9400	0.9400
Weighted avg	0.9548	0.9543	0.9543	0.9400	0.9400	0.9400
Dataset 2						
G1	0.8360	0.9289	0.8800	0.9700	0.9500	0.9600
G2	0.8664	0.7833	0.8228	0.8800	0.9500	0.9100
G3	0.7822	0.7822	0.7822	0.8300	0.8900	0.8600
G4	0.7500	0.7600	0.7550	0.8300	0.7700	0.8000
G5	0.8227	0.8044	0.8135	0.9200	0.8600	0.8900
G6	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Accuracy		0.8425			0.9000	
Macro avg	0.8429	0.8431	0.8422	0.9100	0.9000	0.9000
Weighted avg	0.8431	0.8425	0.8420	0.9100	0.9000	0.9000
Dataset 3						
G1	1.0000	0.9778	0.9888	1.0000	0.9900	0.9900
G2	0.9759	1.0000	0.9878	0.9900	0.9900	0.9900
G3	1.0000	0.9926	0.9963	0.9999	1.0000	1.0000
G4	0.9976	1.0000	0.9988	0.9900	0.9900	0.9900
G5	1.0000	1.0000	1.0000	1.0000	0.9900	1.0000
G6	0.9975	1.0000	0.9988	1.0000	1.0000	1.0000
Accuracy		0.9951			0.9900	
Macro avg	0.9952	0.9951	0.9951	0.9900	0.9900	0.9900
Weighted avg	0.9952	0.9951	0.9951	0.9900	0.9900	0.9900

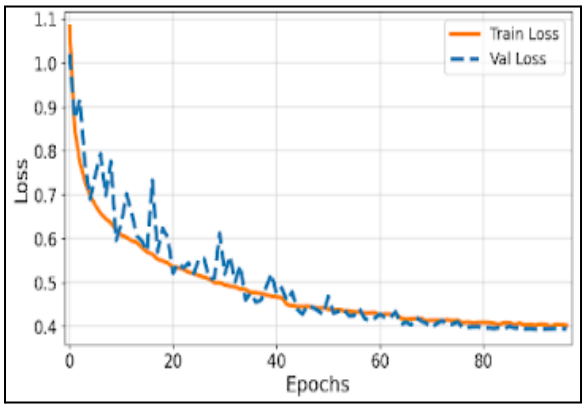
Figure 3 shows the evolution of training and validation loss for the selected folds. After completing the seven folds, the fold with the highest validation accuracy was selected.

Figure 3. Training and validation loss curves for selected folds across different datasets. Subplots (A), (C), and (E) show the performance of Model 1 for Fold 4 (Dataset 1), Fold 1 (Dataset 2), and Fold 5 (Dataset 3), respectively. Subplots (B), (D), and (F) illustrate Model 2 performance for Fold 5 (Dataset 1), Fold 6 (Dataset 2), and Fold 4 (Dataset 3). Dashed lines represent training loss, while solid lines indicate validation loss across epochs.

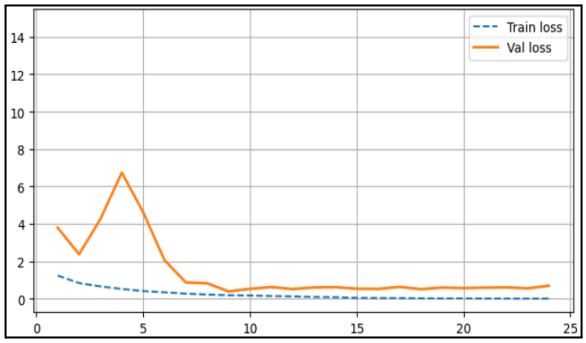
A) Dataset 1 Model 1 - fold 6



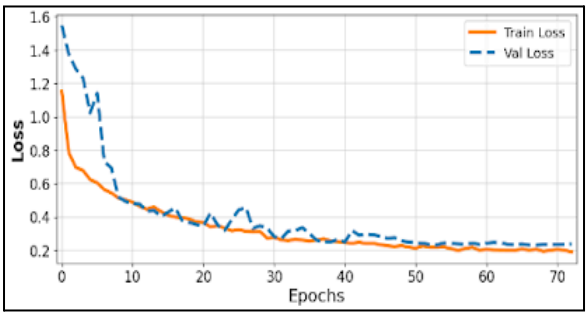
B) Dataset 1 Model 2 - fold 5



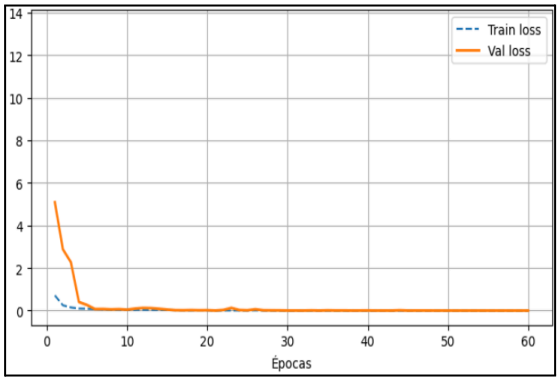
C) Dataset 2 Model 1 - fold 6



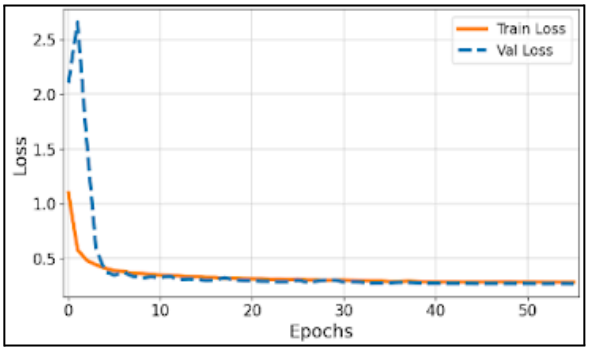
D) Dataset 2 Model 2 - fold 6



E) Dataset 3 Model 1 - fold 7



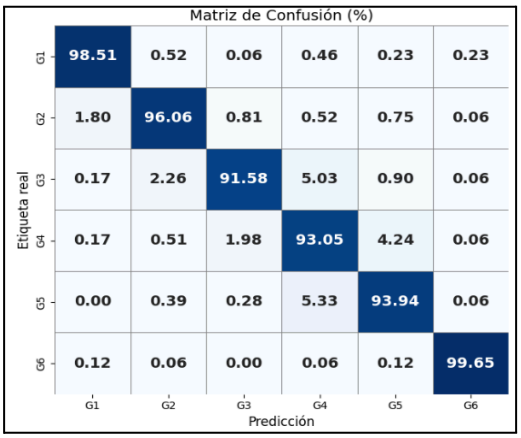
F) Dataset 3 Model 2 - fold 4



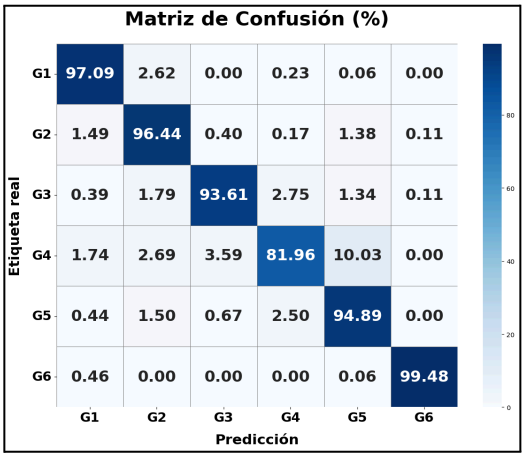
The confusion matrices corresponding to the best model selected for each dataset were calculated using the actual labels versus the predictions. Figure 4 shows the three confusion matrices (one per dataset), which allow us to visualize the most frequent confusions and compare the classification behavior.

Figure 4. Confusion matrices across the three datasets. Panels on the left (A, C, E) display the normalized performance (%) of Model 1 for Datasets 1, 2, and 3, respectively. Panels on the right show the performance for Model 2, specifically for Dataset 1 (F), Dataset 2 (D), and Dataset 3 (B). Each matrix compares real labels (y-axis) against model predictions (x-axis) for classes G1 through G7, where the color intensity on the diagonal indicates the success rate of correct classifications.

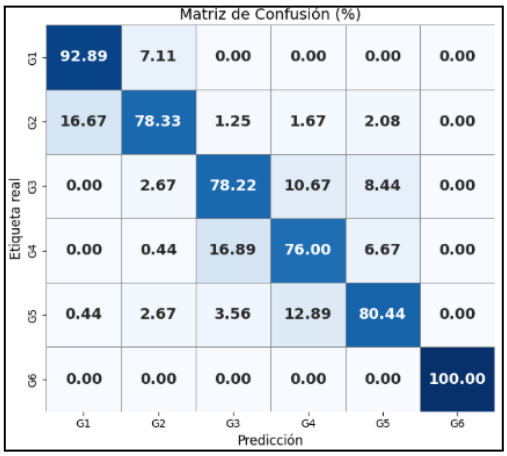
A) Dataset 1 Model 1



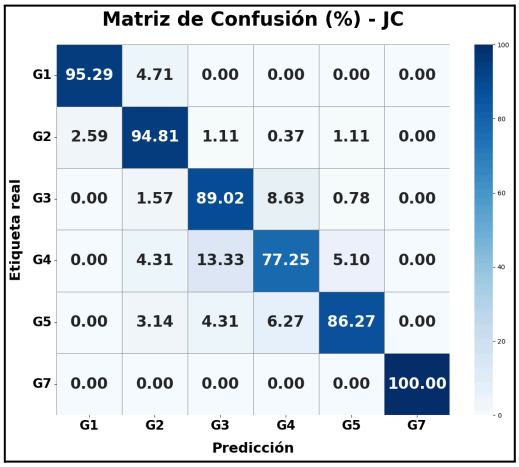
B) Dataset 1 Model 2



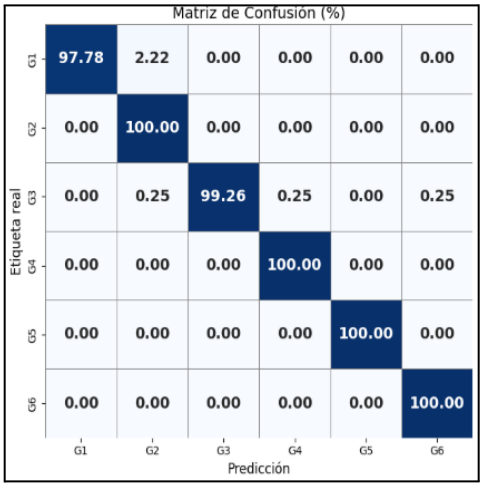
C) Dataset 2 Model 1



D) Dataset 2 Model 2



E) Dataset 3 Model 1



F) Dataset 3 Model 2

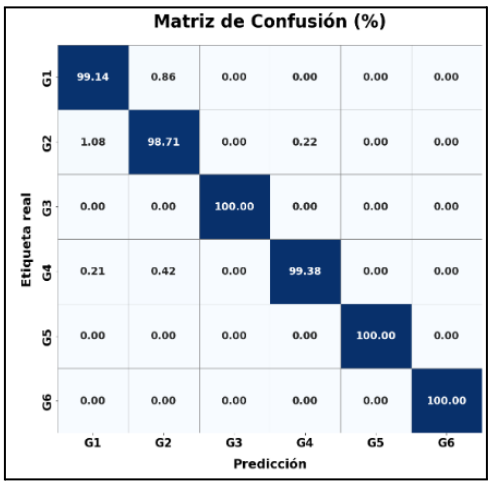


Table 4 shows a comparison of training times expressed in minutes for two different models under three different sensor configurations. For Model 1, times of 300 min are recorded with dataset 1, 35 min with dataset 2, and 90 min with dataset 3. Model 2, on the other hand, shows a reduction in execution times, reporting 94.4 min, 14.7 min, and 16.2 min for the same datasets, respectively.

Table 4. Total training time for models 1 and 2 per dataset.

Model 1	
Dataset	**Training time (min)**
Dataset 1	300
Dataset 2	35
Dataset 3	90
Model 2	
Dataset	**Training time (min)**
Dataset 1	94.4
Dataset 2	14.7
Dataset 3	16.2

Table 5 shows a comparison of average inference times between the non-quantized and quantized versions of the model. For each dataset, three indicators are reported per sample: Feature Extraction (ms) —average time spent on input preparation

(standardization and RMS envelope calculation)—, Prediction (ms) —average time of the forward pass of the classifier— and Total (ms), sum of the two stages.

Table 5. Inference time performance comparison between Model 1 and Model 2. The table contrasts execution times (in milliseconds) for non-quantized and quantized versions across three datasets. For Model 1, prediction times are reported, while for Model 2, the metrics are broken down into Feature Extraction, Prediction, and Total time to illustrate the computational impact of each stage.

	Model 1		Model 2					
	Prediction (ms)		Non-quantized (ms)			Quantized (ms)		
Dataset	Quant.	Non-Q.	Feat.	Pred.	Total	Feat.	Pred.	Total
Dataset 1	69.64	181.31	0.54	33.14	33.68	0.337	0.0606	0.3973
Dataset 2	49.633	156.26	0.44	36.45	36.89	0.335	0.0867	0.4221
Dataset 3	55.39	178.36	0.352	29.117	29.47	0.184	0.046	0.231

Table 6 shows the configuration of the computer used to evaluate the quantized model, detailing the Intel® Core™ i7-8550U @ 1.80 GHz processor with four physical cores and eight execution threads, a total RAM of approximately 7.89 GB, and an integrated Intel® HD Graphics 1500 GPU. ® Core™ i7-8550U @ 1.80 GHz processor with four physical cores and eight threads, a total RAM of approximately 7.89 GB, and an integrated Intel UHD Graphics 620 GPU, which defines the test environment in which the measurements were taken. Table 7 shows the resource consumption associated with the model's execution, where the initial impact on RAM is 3.03 MB, the average CPU usage during continuous inference reaches 1.28%, and the maximum CPU usage reaches 100% at specific moments.

Table 6. Technical specifications of the computer used for model evaluation.

Component	Specification
Processor	Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz
Physical cores	4
Logical cores	8
Total RAM	7.89 GB
GPU	Intel(R) UHD Graphics 620

Table 7. System resource consumption during quantized model execution.

Resource	Measurement (unit)
Initial RAM Impact (Load)	3.03 MB
Average CPU Usage	1.28 %
Maximum CPU Usage	100.00 %

5. DISCUSSION

When analyzing the results in table 3, it is clear that the performance of the models varies depending on the specific group of data used. In Dataset 1, Model 1 achieves 95.43% compared to 94.00% for Model 2, representing a small advantage of 1.43 percentage points for the larger model. However, in Dataset 2, the situation is reversed: Model 2 reaches 90.00%, surpassing Model 1, which stands at 84.25%. Finally, in Dataset 3, both models show nearly identical results, with accuracies nearing perfection at 99.51% and 99.00%.

These data allow us to identify key findings. Model 2 obtains very similar results to Model 1 in datasets 1 and 3; however, it achieves better results in dataset 2. This suggests that combining the standard signal with the RMS envelope helps the model better distinguish gestures under certain conditions. On the other hand, the slight superiority of Model 1 in dataset 1 (which is the largest) indicates that its greater number of parameters allows it to capture subtle details in data with high variability or changes. In the case of dataset 3, both models perform so well that any difference is practically negligible.

Regarding the training process, the graphs in Figure 3 show that both models learn steadily. The use of callbacks like early stopping and automatic learning rate reduction allowed the process to be efficient, stopping as soon as no real improvements were detected.

Finally, technical efficiency is demonstrated in Tables 4 and 5. While the training time reflects the effort of processing data multiple times to ensure valid results, the inference tests are what truly matter for the end-user. Thanks to quantization, Model 2 reduces its response time so significantly that, in practice, the system spends more time preparing the signal than actually performing the prediction.

6. CONCLUSIONS

The results obtained indicate that the choice of the most suitable model depends primarily on the needs and objectives of the system to be developed. When seeking a solution that can operate in real time, requires low memory consumption, and is easy to implement on devices with limited resources, Model 2 emerges as the most convenient alternative. Thanks to its more efficient architecture, this model achieved better performance when working with Dataset 2, which was acquired using low-cost sensors from EMG signals of a person with a hand amputation. At the same time, it maintained outstanding performance when evaluating data obtained under the same experimental conditions, but from a person without an amputation, demonstrating its capacity and potential for practical applications.

Conversely, Model 1 stands out as the best alternative when the primary factor is maximizing accuracy on very large and varied datasets, such as Dataset 1, provided the necessary computational power is available to handle their size. It is important to note that, regardless of the model, specific gestures, such as gestures 3 and 4, present a similar challenge for both, while others are correctly identified almost 100% of the time.

One of the most relevant conclusions of this work is the positive impact of quantization on system efficiency. As shown in Table 5, this technique is key to achieving real-time operation. In Model 1, quantization reduces the time required to make a prediction across all datasets by more than half. This effect is even more pronounced in Model 2, where the total inference time is reduced to less than 0.5 ms, compared to the 30–36 ms required without these optimizations. At these speeds, the system spends more time preprocessing the signals than performing the classification itself, ensuring a virtually instantaneous response. This behavior is fundamental for real-world gesture recognition applications, where speed is an essential requirement.

Finally, hardware testing confirmed that the optimized model is fully viable for everyday use. With a RAM consumption of only 3.03 MB and extremely low processor usage, the system can run on standard computers without impacting their overall performance. This ensures that the developed models are efficient for real-world gesture applications.

ETHICAL CONSIDERATIONS

This project is part of the larger project “A METHODOLOGY BASED ON ARTIFICIAL INTELLIGENCE FOR PATTERN RECOGNITION IN MYOELECTRIC SIGNALS WITH TRANSCUTANEOUS ELECTRICAL STIMULATION APPLICABLE IN TRANSRADIAL AMPUTATION PROSTHESIS,” which carried out the collection and management of the data used in this study. The research protocol was reviewed and approved by the Comité de Ética en Investigación Científica de la Universidad Industrial de Santander (CEINCI-UIS) under Act No. 29, dated November 19, 2024. In compliance with CEINCI-UIS requirements, the corresponding ethical documentation — including signed informed consent forms and the records of tests performed to date — has been submitted to the committee for monitoring and follow-up. Participant information was handled confidentially: data were anonymized and stored securely. The use of these data is restricted to academic and research purposes related to this project; any other use will require prior authorization and adherence to the committee’s directives and applicable ethical standards.

BIBLIOGRAPHY

AMERI, A. et al. Real-Time Simultaneous and Proportional Control of Multiple Degrees of Freedom from Surface EMG Using CNN. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. Diciembre, 2017. vol. 25, no. 12, p. 2445–2454.

ATZORI, M. et al. Building the Ninapro database: A resource for the biorobotics community. En: CONFERENCIA INTERNACIONAL IEEE RAS & EMBS SOBRE ROBÓTICA BIOMÉDICA Y BIOMECATRÓNICA (4: 2012: Roma). Memorias. Roma: IEEE, 2012. p. 1258–1265.

ATZORI, M. et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. En: Scientific Data. 2014. vol. 1.

CASTELLINI, C. et al. Proceedings of the first workshop on Peripheral Machine Interfaces: Going beyond traditional surface electromyography. En: Frontiers in Neurorobotics. 2014. vol. 8, p. 22.

CÔTÉ-ALLARD, U. et al. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2019. vol. 27, no. 4, p. 760–771.

DE LUCA, C. J. The Use of Surface Electromyography in Biomechanics. En: Journal of Applied Biomechanics. 1997. vol. 13, no. 2, p. 135–163.

DING, Z. et al. Gesture recognition based on sEMG using convolutional neural networks. En: Sensors. 2018. vol. 18, no. 5.

DU, Y. et al. Surface EMG-based inter-session gesture recognition using a deep convolutional neural network. En: Sensors. 2017. vol. 17, no. 9, p. 2046.

ENGLEHART, K. y HUDGINS, B. A robust, real-time control scheme for multifunction myoelectric control. En: IEEE Transactions on Biomedical Engineering. 2003. vol. 50, no. 7, p. 848–854.

FARINA, D. et al. The extraction of neural strategies from the surface EMG. En: Journal of Applied Physiology. 2004. vol. 96, no. 4, p. 1486–1495.

FONT-JIMÉNEZ, M. et al. Quality of life and functionality in upper limb amputees. En: Disability and Rehabilitation. 2016. vol. 38, no. 20, p. 1954–1961.

FUKANO, K. et al. Aprendizaje profundo para el reconocimiento de gestos basado en datos EMG de superficie. En: CONFERENCIA INTERNACIONAL SOBRE SISTEMAS MECATRÓNICOS AVANZADOS (ICAMechS) (2021: Tokio). Memorias. IEEE, 2021. p. 41–45.

GARCÍA, J. y PÉREZ, M. Psychological impact of limb amputation. En: Revista de Psicología de la Salud. 2015. vol. 27, no. 2, p. 85–92.

GENG, H. et al. Gesture recognition using high-density surface EMG and deep convolutional neural networks. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2016. vol. 24, no. 11, p. 1156–1164.

HU, Y. et al. Attention-based CNN for Surface EMG Gesture Recognition. En: IEEE Journal of Biomedical and Health Informatics. 2018. vol. 22, no. 6, p. 1747–1756.

HUDGINS, B., PARKER, P. y SCOTT, R. A new strategy for multifunction myoelectric control. En: IEEE Transactions on Biomedical Engineering. 1993. vol. 40, no. 1, p. 82–94.

JOHNSON, V. et al. Mental health outcomes following limb loss. En: Journal of Rehabilitation Research. 2016. vol. 53, no. 4, p. 473–482.

LECUN, Y., BENGIO, Y. y HINTON, G. Deep learning. En: Nature. 2015. vol. 521, p. 436–444.

LI, W., SHI, P. y YU, H. Gesture Recognition Using Surface Electromyography and Deep Learning for Prostheses Hand: State-of-the-Art, Challenges, and Future. En: Frontiers in Neuroscience. 2021. vol. 15, p. 621885.

LU, J. et al. EffiE: Efficient Convolutional Neural Network for Real-Time EMG Pattern Recognition System on Edge Devices. En: CONFERENCIA INTERNACIONAL IEEE/EMBS SOBRE INGENIERÍA NEURAL (11: 2023: Baltimore). Memorias. IEEE, 2023.

MENDES JR., J. et al. Evaluation of MyoWare and Delsys sensors for sEMG-based control. En: Biomedical Signal Processing and Control. 2020. vol. 57, p. 101765.

MERLETTI, R. y PARKER, P. Electromyography: Physiology, Engineering, and Non-Invasive Applications. Wiley-IEEE Press, 2004.

NASSER-CORTÉS, J. et al. Comparative Analysis of Low-Cost and Clinical EMG Sensors for Gesture Recognition. En: IEEE Sensors Journal. 2025. vol. 25, no. 2, p. 45–55.

NEACSU, A. A. et al. Automatic EMG-based Hand Gesture Recognition System using Time-Domain Descriptors and Fully-Connected Neural Networks. En: CONFERENCIA INTERNACIONAL SOBRE TELECOMUNICACIONES Y PROCESAMIENTO DE SEÑALES (42: 2019: Budapest). Memorias. IEEE, 2019. p. 232–235.

NILWONG, S. et al. Deep Learning-Based EMG Gesture Recognition for Amputee Subjects. En: CONFERENCIA INTERNACIONAL IEEE SOBRE ROBÓTICA Y BIOMIMÉTICA (ROBIO) (2022: Jinghong). Memorias. IEEE, 2022. p. 145–150.

PAN, L. et al. Clinical Validation of sEMG Pattern Recognition Using BTS Bioengineering Systems. En: Journal of NeuroEngineering and Rehabilitation. 2023. vol. 20, no. 1, p. 112.

PHINYOMARK, A. et al. Feature extraction and selection for myoelectric control. En: Expert Systems with Applications. 2012. vol. 39, no. 8, p. 7420–7431.

PHINYOMARK, A., KHUSHABA, R. N. y SCHEME, E. Feature extraction and selection for myoelectric control based on wearable EMG sensors. En: Sensors. 2018. vol. 18, no. 5, art. 1615.

PIZZOLATO, S. et al. Comparison of EMG-based classification and regression for prosthetic control in real-time. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2017. vol. 25, no. 9, p. 1629–1639.

SCHEME, T. y ENGLEHART, K. Electromyogram pattern recognition for control of powered upper-limb prostheses. En: Journal of Rehabilitation Research & Development. 2011. vol. 48, no. 6, p. 643–660.

TIAN, J. y LUO, X. EMG-Based Gesture Classification Using Convolutional Neural Networks. En: CONFERENCIA INTERNACIONAL SOBRE TECNOLOGÍA DE SENSORES E INFORMACIÓN (5: 2025: Ciudad de la conferencia). Memorias. IEEE, 2025. p. 1288–1293.

WEI, W. et al. A Multi-View CNN Framework for Gesture Recognition from Surface EMG Signals. En: IEEE Transactions on Cybernetics. 2019. vol. 49, no. 4, p. 1150–1162.

XIE, Z. et al. High-Performance EMG Gesture Recognition for Amputees via Residual CNN. En: IEEE Transactions on Cognitive and Developmental Systems. 2022. vol. 14, no. 3, p. 880–892.

ZANDIGOHAR, M. et al. Multimodal fusion of EMG and vision for human grasp intent inference in prosthetic hand control. En: Frontiers in Robotics and AI. 2024. vol. 11, art. 1312554.

ZHAI, T. et al. Self-Recalibrating Surface EMG Pattern Recognition for Neuroprosthesis Control Based on Convolutional Neural Network. En: Frontiers in Neuroscience. 2017. vol. 11, p. 379.

ZHANG, H., SID'EL MOCTAR, S. M., BOUDAOUD, S. e RIDA, I. A comprehensive review of sEMG-IMU sensor fusion for upper limb movements pattern recognition. En: Information Fusion. 2026. vol. 125, p. 103422.

ZHONG, W. y JIANG, X. Deep Feature Learning from Electromyographic Signals for Gesture Recognition Systems. En: IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2024. vol. 32, p. 435–446.

ZIA UR REHMAN, M. et al. Multiday EMG-based gesture recognition using deep learning for amputees. En: IEEE Transactions on Instrumentation and Measurement. 2018. vol. 67, no. 8, p. 1914–1924.

ANNEXES

Annex A. GitHub Repository

The implementation of the model in this study is documented and publicly available on GitHub through the repository:

https://github.com/Raul-coder07/Hand-Gesture-Clasification.git