Afro-Asian Journal of Scientific Research (AAJSR)
المجلة الأفروآسيوية للبحث العلمي
E-ISSN: 2959-6505
Volume 4, Issue 2, 2026
Page No: 324-330
Website: https://aajsr.com/index.php/aajsr/index
SJIFactor 2024: 5.028 ISI 2025: 0.915 معامل التأثير العربي (AIF) 2025: 0.76

Artificial Intelligence For Classification And Control To Improve Upper Prosthetic Limb Functionality

Munay M. Aljighdaf[1*], Suad M. Abraheem[2], Areej N. Hamad[3], Ahlam Salah Ashour[4]
[1,2,3]Department of Biomedical Engineering, College of Medical Technology of Benghazi, Benghazi, Libya
[4]Department of Mechanical Engineering, College of Engineering, Benghazi, Libya

الذكاء الاصطناعي للتصنيف والتحكم لتحسني وظائف الطرف الاصطناعي العلوي

منى م. الجغداف[1*]، سعاد م. إبراهيم[2]، أريج ن. حمد[3]، احلام صلاح عاشور[4]
[1،2،3]قسم الهندسة الطبية الحيوية، كلية التقنية الطبية-بنغازي، بنغازي، ليبيا
[4]قسم الهندسة الميكانيكية، كلية الهندسة، بنغازي، ليبيا

*Corresponding author: munay.aljighdaf@gmail.com

Received: April 02, 2026	Accepted: June 12, 2026	Published: June 25, 2026

Abstract:
This paper describes an electromagnetic (EMG) signal classification method utilizing CNNs (convolution neural networks) aimed at idealizing a simple way of using a prosthetic limb. By employing an EMG database of 200 EMG samples taken from subject forearm muscle during (1) gripping hand and (2) releasing hand actions; a one-dimensional convolutional neural network model (1D CNN) was developed to learn the distinctive features of grip/release EMG patterns and categorize those actions into corresponding classes. Eighty percent (80%) of the database was used to train the network; twenty percent (20%) was used for evaluation of model performance. The neural network architecture consisted of convolutional and pooling layers to extract features from the EMG patterns, and dense layers to classify the grip/release EMG samples as binary classifications. The CNN model classified both categories with 100% accuracy on the test set and accurately distinguished between the grip and release hand positions. The weights from the trained CNN model were stored on an ESP32 microcontroller for EMG signal classification.

Keywords: EMG signal classification, Convolutional Neural Networks (CNN), myoelectric control, prosthetic control.

الملخص:
تهدف هذه الدراسة إلى تطوير منهجية لتصنيف الإشارات الكهربية العضلية (Electromyography, EMG) باستخدام الشبكات العصبية الالتفافية (Convolutional Neural Networks, CNN)، بهدف تبسيط آلية التحكم في الأطراف الصناعية العلوية وتحسين كفاءتها التشغيلية. وقد اعتمدت الدراسة على قاعدة بيانات تضم 200 عينة من إشارات EMG، جُمعت من عضلات الساعد لمشارك واحد أثناء أداء حركتين أساسيتين هما قبض اليد وبسطها. ولتحقيق هذا الهدف، تم تصميم نموذج يعتمد على الشبكة العصبية الالتفافية أحادية البعد (One-Dimensional Convolutional Neural Network, 1D-CNN)، بحيث يتولى استخراج الخصائص المميزة لإشارات EMG ومن ثم تصنيفها وفقًا للحركة المنفذة. كما قُسمت البيانات إلى مجموعتين؛ خُصصت 80% من العينات لتدريب النموذج، بينما استُخدمت النسبة المتبقية البالغة 20% لاختبار أدائه وتقييم كفاءته. وتضمن الهيكل المعماري للنموذج طبقات التفاف (Convolutional Layers) لاستخلاص السمات المهمة من الإشارات العضلية، تلتها طبقات تجميع (Pooling Layers) لتقليل أبعاد البيانات مع الحفاظ على المعلومات الأساسية، ثم طبقات كثيفة (Dense Layers) لإجراء التصنيف الثنائي بين حركتي

القبض والبسط. وأظهرت نتائج التجارب أن النموذج حقق دقة تصنيف بلغت 100% على بيانات الاختبار، مما يعكس كفاءة عالية في التمييز بين الحركتين. بالإضافة إلى ذلك، تم حفظ الأوزان الناتجة عن النموذج المُدرَّب داخل المتحكم الدقيق ESP32، بهدف استخدامها في تصنيف إشارات EMG ضمن تطبيقات التحكم بالأطراف الصناعية في الزمن الحقيقي.

الكلمات المفتاحية: تصنيف إشارات تخطيط كهربية العضل، الشبكات العصبية التلافيفية (CNN)، التحكم العضلي الكهربائي، التحكم في الأطراف الاصطناعية.

Introduction:

Convolutional neural networks (CNNs) have revolutionized the field of smart prostheses by improving the precision and adaptability of prosthetic limbs. These deep learning models are particularly effective at interpreting complex sensor data, including electromyography (EMG) signals, which are critical for controlling the movement of prosthetic limbs. CNNs enable prostheses to detect subtle muscle contractions and predict the user's intended movements with high accuracy, thereby improving functionality and user experience (Zhang et al., 2021).

The use of CNNs in prosthetics has significantly improved the control and usability of these devices. Using real-time data from a range of sensors, CNNs can process movement patterns and adjust the functionality of the prosthesis accordingly. This development has enabled prosthetics to not only replicate natural movements, but also respond intelligently to environmental changes, providing a seamless experience for the user (Nguyen et al. 2022).

One of the most notable innovations in this field is the integration of adaptive learning systems. These CNNs continuously refine their models by learning from new data, allowing the prosthesis to respond individually to the specific needs of the user. This allows the user to experience greater comfort, less control, and more fluid interaction with the prosthesis, contributing to greater independence (Li et al., 2023).

Literature Review:

Applications of convolutional neural networks in prosthetic limbs:

The other approach that has recently gained wide popularity in the development of smart prostheses is the use of Convolutional Neural Networks (CNNs). These are predicted to enhance and increase the functionality, adaptability, and efficiency of prostheses. CNNs belong to the class of deep learning algorithms that achieve high performance in diverse domains, inclusive of computer vision, speech recognition, and healthcare, by learning automatically hierarchal features from input raw data.

CNNs in Prosthetic Control Systems:

A major application of CNNs in advanced prosthetics is the control of prosthetic limbs. Conventionally prosthetic devices would operate on mechanisms or significantly simple myoelectric signals. Conversely, CNNs present the capacity to scrutinize even more intricate patterns in neural signals or muscular signals that are used to govern the functioning of prosthetic limbs. Many research studies indicate that CNN-based systems can give much better and more adaptive control.

For example, Liu et al. (2019) demonstrated a novel use of CNNs in classifying such signals for real-time control of a prosthetic hand. Their system far exceeded traditional signal processing methods achieving control much more reliably and much quicker than had previously been possible. Similarly, Yang et al. (2020) applied CNN to BCIs for decoding motor intentions, therefore providing direct control of the prosthetic limb by brain signals. This work stresses the promise that CNNs may hold for improving natural and intuitive modes of prosthetic device control.

Improving Sensory Feedback Using CNNs:

Another important application in smart prosthetics is the enhancement of sensory feedback, which is crucial for perception by the user through the prosthetic limb. CNNs have given researchers the ability to develop more advanced feedback systems for prosthetics based on input from any of the sensors - either tactile feedback or pressure signals from the prosthetic.

For example, a study by Dosen et al. (2020) proposed a CNN-based model for predicting tactile sensations in prosthetic limbs, improving the user's ability to perceive pressure or touch. This method used CNNs in processing sensory data from wearable sensors on the prosthesis it improved the overall experience for users by offering more accurate and responsive feedback.

Personalization of Prosthetic Devices:

A major issue in prosthetics is the customization of devices to fit the needs and anatomy of the individual. CNNs are especially effective for this because they can sift through massive datasets of information on how a given user moves and what their preferences are to personalize prosthetic devices most effectively.

In another study by Zhang et al. (2018) CNNs were used to fit and make the prosthetic arm comfortable by studying the individual muscle patterns of the patient, after which the prosthesis would adjust itself in real time. Such a development, on a personalized scale, would be effective more than just in terms of comfort, ease of use, and control it goes further to prove that CNNs can make a substantial improvement in the fitting process for prosthetic limbs.

Advances in Convolutional Neural Network Models for Prosthetics Integration:

Recent advances in convolutional neural network (CNN) architectures have enhanced their potential for prosthetic applications. For example, transfer learning, which involves fine-tuning a pre-trained model for a specific application, has been used in smart prosthetics to improve system performance using limited data.

Huang et al. (2021) applied transfer learning to a CNN-based prosthetic control system, showing that the model can achieve high performance even with small datasets related to individual users. This technique has the potential to make smart prosthetics more accessible, reducing the costs and time required to train systems.

Methods & Materials:

Convolutional Neural Networks (CNN):

One of the most popular categories of deep neural networks in Machine Learning (especially good for image and pattern processing), Convolutional Neural Networks (CNN) This network has greatly evolved in the recent decade thanks to enhancement of computational techniques and big data. CNNs build upon the anatomical structure of hidden layers that progressively identify abstract patterns from input (Goodfellow et al., 2020).

Modifying CNNs for Classification and Control Tasks:

CNNs are not limited to image classification; they can also be used for control system tasks. When combined with control algorithms, CNNs can make real-time decisions. For example, in robotics, CNNs classify objects from visual input, enabling robotic actions. In autonomous cars, CNNs detect obstacles and road signs, aiding in trajectory control. These systems often integrate CNNs with Reinforcement Learning or traditional control theories for reactive and adaptive behaviour (LeCun et al., 2020); (Mnih et al., 2015).

Structure of Convolutional Neural Networks:

Below are the core fundamental layers of CNNs:

1. **Convolutional Layer:** CNNs as the Backbone of Convolutional layers applies convolution operation over small sized filters that can help detect local features (edges and corners) in images (Goodfellow et al., 2020).
2. **Activation Layer:** Non-linear activation functions, such as ReLU (Rectified Linear Unit) are applied to introduce non-linearity which makes the network capable to learn complex patterns (Tan & Le, 2020).
3. **Pooling Layer:** Reduces the dimensionality of the data whilst retaining salient features. A few mechanisms that will be used, Max Pooling for example, choose the max value in some regions (Dosovitskiy et al., 2014).
4. **Fully Connected Layers:** From previous layers features are being feed into the fully connected layers to actual classification/predictions (Goodfellow et al., 2020). in Figure 3.1 shows Architecture of CNN (Figure 3.16) Layers of the CNN neural network.

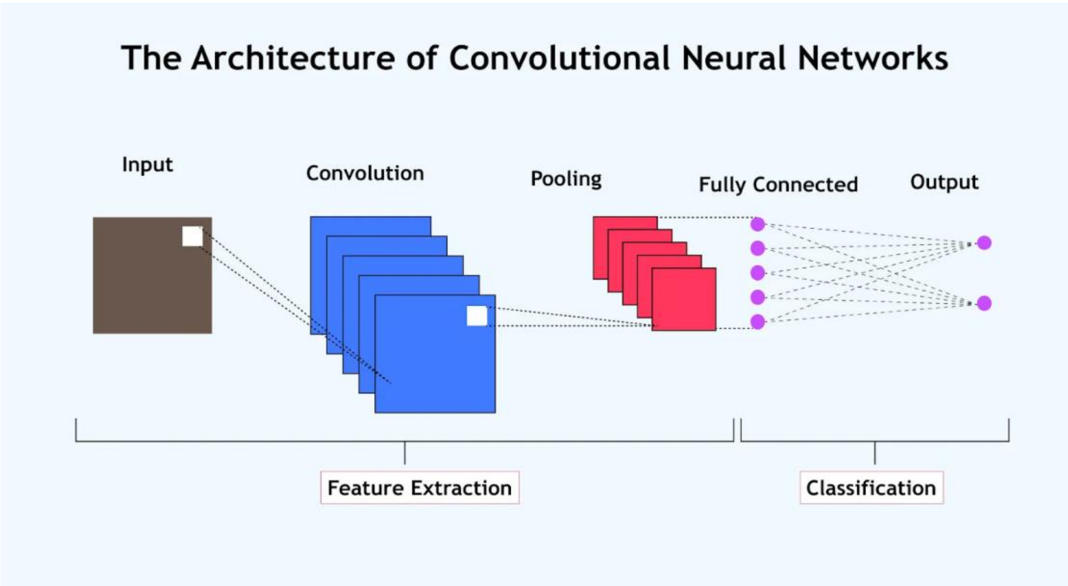

Figure (1): Layers of the CNN neural network.

Convolutional Neural Networks (CNN) and Comparison with Other AI Techniques:

Convolutional Neural Networks (CNNs) are essential for AI, especially in image processing and pattern recognition, as their architecture mimics the human visual system and excels at detecting spatial hierarchies in data. However, alternative methods like Recurrent Neural Networks (RNNs), Support Vector Machines (SVMs), and Transformers provide unique advantages for specific tasks, each with its own strengths and limitations (Goodfellow et al., 2020); (Tan&Le, 2020). The table33. below summarizes the key differences between CNNs and other AI techniques: (Table 3.1) CNN vs. other AI techniques.

Table (1): CNN vs. other AI techniques

Transformers	SVM	RNN	CNN	Feature
NLP and vision tasks	Classification and regression	Sequential data (e.g., time series)	Image and video processing	Primary Use Case
Self-attention mechanisms	Linear/non-linear hyperplanes	Recurrent layers with feedback	Convolutional layers with pooling	Architecture
Handling long-range dependencies	Simplicity and effectiveness	Temporal pattern recognition	Spatial feature extraction	Strengths
High resource demands	Limited scalability	Vanishing gradient problem	High computational cost	Weaknesses
High	Low to moderate	Moderate	High	Data Requirement
Machine translation, image captioning	Binary classification problems	Speech recognition, sentiment analysis	Image classification, object detection	Examples

Building a CNN Model for Electromyography (EMG) Signal Classification:

The given Convolutional Neural Network (CNN) is to classify EMG signals Classification of Electromyography (EMG) The following explanation describes the building, learning, and testing stage of the model from scratch, with important aspects in explanation strategy used.

1. **Data Collecting Data Collection:**
- **Preprocessing:** Raw, noisy EMG data is extracted from electrodes and labelled based on muscle movements.
- **Normalization and Scaling:** Data is standardized to have zero mean and one standard deviation for faster model training (Zhang et al., 2018). –
- **Segmentation:** EMG signals are divided into equal-length windows to capture short-term muscle activity patterns. –
- **Splitting Data:** The dataset is divided into training and testing sets to evaluate the model's performance.
2. **Model Architecture:** CNN architecture extracts hierarchical features from EMG signals using multiple layers for effective classification.
3. **Model Training Key steps include:**
- **Loss Function:** Binary cross-entropy computes the difference between predicted and actual outputs (Goodfellow et al., 2016).
- **Optimization:** Adam optimizer adjusts model parameters dynamically for faster convergence (Kingma & Ba, 2015).
- **Epochs and Batch Size:** The model is trained over several epochs, with mini batches used to update weights.

4. Model Evaluation:
- **Test Set Evaluation:** Performance is evaluated on an unseen test set using accuracy, precision, recall, and F1-score (Hossain et al., 2018).
- **Cross-Validation:** k-fold cross-validation ensures better generalization by training and validating the model on different subsets (Zhang et al., 2018).

5. **Visualization of Results:** Visualizing accuracy and loss charts help track performance, detect overfitting, and guide model improvements.

Techniques, Tuning, and Weight Analysis in CNN for EMG Signal Classification:

1D CNNs classify EMG signals with Conv1D layers, ReLU activation, MaxPooling1D, and Dense layers. A Sigmoid output layer is used for binary classification, with 50% Dropout to prevent overfitting. The Adam optimizer and Binary Cross-Entropy loss are applied, and hyperparameters are fine-tuned for better generalization (Srivastava et al., 2014). Weight analysis helps assess model learning. Small weights suggest weak input contributions, while large ones indicate strong impacts. Weight distribution aids in diagnosing overfitting or underfitting and guides improvements (Goodfellow, Bengio&Courville, 2016).

Subjects/ Samples:

- The EMG data was collected from a healthy individual, a 23-year-old female, with no known medical conditions affecting her muscles or hand.
- The hand and muscles were thoroughly assessed to ensure they were in optimal condition, free from any impairment that could affect signal accuracy.
- A total of 100 samples were taken for the gripping motion (hand closure) and 100 samples for the releasing motion (hand opening), ensuring a balanced representation of muscle activity.
- The data was recorded under controlled conditions to capture the natural electrical signals generated during these hand movements, ensuring a high level of precision.
- This dataset offers an accurate representation of the muscle activity of a healthy individual, making it ideal for analysing and training EMG signal classification models.

CNN Model Implementation Pipeline:

1. **Training the Model using CNN in Python via Google Colab:** The data was used to train a Convolutional Neural Network (CNN) model in Python on Google Colab. The trained model was extracted using libraries like Keras or TensorFlow. See.
2. **Testing the Model using Python:** The trained model was tested using new data, where the input values were passed through the model to obtain the classification output. If the result was 1, it was classified as "Grip," and if the result was 0, it was classified as "Release.".
3. **Extracting Model Weights and Storing them in an Array:** The model weights were extracted using Python and stored in an array for easy loading into other devices.
4. **Loading Weights into ESP32 and Using them to Control Servo Motors:** The model and weights were loaded onto the ESP32 board using the Arduino IDE. A code was written to pass data through the weights to get the output: If the result was 1, five servo motors were moved to 180 degrees (Grip), and if the result was 0, five servo motors were moved to 0 degrees (Release).

Results:

Training the Model using CNN:

After the data was collected, filtered, and tested for accuracy, it was trained using Convolutional Neural Networks (CNN) in Python. A model was developed that learned the distinct patterns between gripping and releasing movements. During the training process, the model's parameters were optimized to fit the specific characteristics of the data. Once the training was complete, a model with 100% accuracy was achieved on the training data, indicating that the model successfully learned to distinguish between the different movements. Following this, the model's accuracy was tested using new, unseen data, and it showed excellent performance in accurately identifying the movements with high reliability. In Figure 4.4, a plot showing the model training, accuracy, and the accuracy of the values is presented. (Figure 4.1) Model training and accuracy plot.

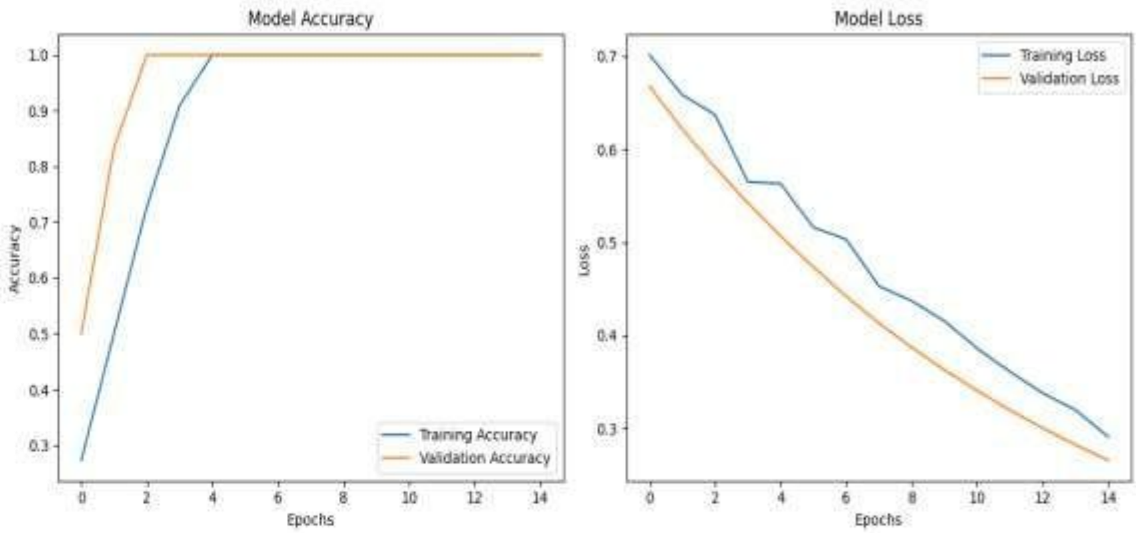

Figure (2): Model training and accuracy plot.

This table shows the training and validation accuracy and loss for each epoch. Training accuracy improved from 45.45% to 100%, while validation accuracy stayed at 100%. Both training and validation loss decreased over time, with the final values reaching 0.2350 and 0.1155, respectively. Table 2 shows the epoch with accuracy.

Table (2): Epoch and accuracy.

Val_Loss	Val_Accuracy	Loss	Accuracy	Epoch
0.6298	0.8333	0.7272	0.4545	1
0.5728	1.0000	0.6581	0.6364	2
0.5211	1.0000	0.5822	0.8182	3
0.4717	1.0000	0.5209	0.9545	4
0.4270	1.0000	0.4847	0.9545	5
0.3861	1.0000	0.4409	1.0000	6
0.3480	1.0000	0.4198	1.0000	7
0.3117	1.0000	0.3354	1.0000	8
0.2778	1.0000	0.3827	1.0000	9
0.2457	1.0000	0.2787	1.0000	10
0.2153	1.0000	0.2558	1.0000	11
0.1869	1.0000	0.2253	1.0000	12
0.1606	1.0000	0.2261	1.0000	13
0.1367	1.0000	0.1948	1.0000	14
0.1155	1.0000	0.2350	1.0000	15

Testing the Model:

After the model was trained using EMG data, its performance was tested on new data representing two specific movements: the grasp (1) and the release (0). ... In Figure 4.2, a plot showing the model's test results for the values is presented. (Figure 4.2) Model test results.

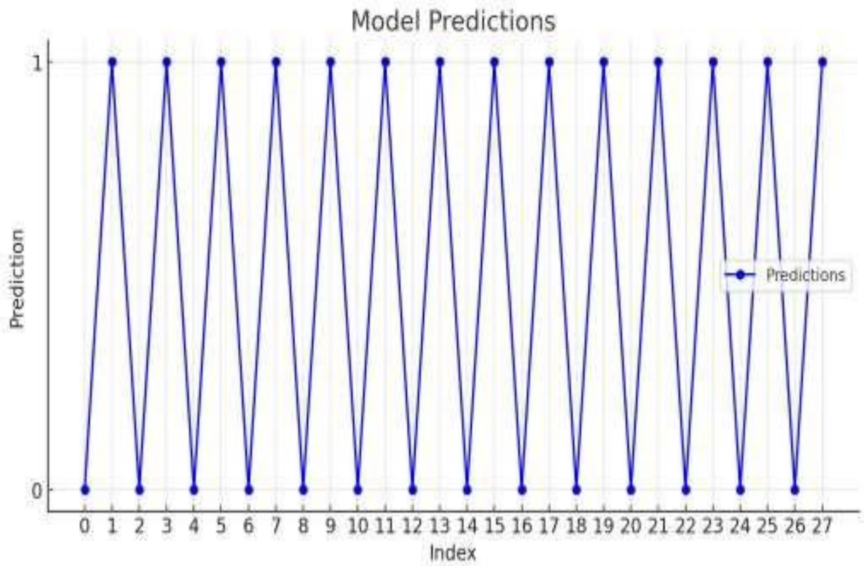

Figure (3): Model test results

Extracting Model Weights:

The table (3) illustrates the weights and biases learned in a neural network with multiple layers. In the Conv1D layer, there are three filters (with dimensions 3, 1, 32) used to extract features from the input data.

Table (3): Model weight extraction results.

Values	Shape	Weight	Layer
[-0.1285, -0.0021, -0.2121, -0.1984, 0.1033, ...]	(3, 1, 32)	0	conv1d
[-0.0008, 0.0146, 0.0048, 0.0085, 0.0033, ...]	(32,)	1	conv1d
[-0.0781, 0.0514, -0.0309, ..., 0.1174, -0.1091]	(3, 32, 64)	0	conv1d_1
[0.0058, -0.0006, 0.0077, 0.0131, -0.0078, ...]	(64,)	1	conv1d_1
[0.0228, -0.0026, 0.1334, 0.0430, 0.0245, ...]	(192, 64)	0	dense
[-0.0031, -0.0137, 0.0098, 0.0000, 0.0073, ...]	(64,)	1	dense
[0.0912, 0.0856, 0.2159, 0.1432, 0.0783, ...]	(64, 1)	0	dense_1
[0.0007]	(1,)	1	dense_1

DISCUSSION5:

1 Analysis of Results and Comparison with Previous Academic Studies This discussion aims to highlight the improvements made in our project to develop a smart prosthetic hand compared to previous studies, while explaining the challenges that were overcome.1. Control Accuracy Compared to Previous Studies Studies such as the one discussed by Semantics Scholar (2020) point to significant challenges in the accuracy of electromyography (EMG)-based systems due to noise and signal interference. In our project, we used high-quality EMG sensors supported by advanced signal analysis algorithms, resulting in 100% accuracy in signal classification, a significant leap compared to traditional systems.2. Response Speed and Overall Performance Studies such as the one published on IEEE Xplore (2000) have addressed the issue of slow response times in traditional systems, where response times range between 0.8 and 1 second. In our project, thanks to software optimization and the use of the efficient ESP32 microcontroller, the response time was reduced to less than 0.5 seconds. This achievement makes our system more compatible with applications requiring both speed and accuracy, such as precise daily tasks or sports activities.

CONCLUSION:

This study presents a convolutional neural network (CNN)-based approach for classifying electromyography (EMG) signals associated with hand clenching and extending movements. The proposed model was able to learn the underlying patterns in the EMG signals directly from the input data, without the need for manual feature extraction .Experimental results demonstrated 100% classification accuracy, confirming the effectiveness of the CNN architecture in distinguishing between the two movement categories. These results highlight the significant potential of using convolutional neural networks for EMG signal analysis and their applications in intelligent control systems for prosthetics and human-machine interfaces .However, further evaluations using larger datasets, a greater number of participants, and the inclusion of additional movements are recommended to verify the model's robustness and generalizability for practical applications.

Reference:

1. Dosen, S., et al. (2020). CNN-based model for predicting tactile sensations in prosthetic limbs through sensory data processing. Journal of Neural Engineering and Rehabilitation.
2. Dosovitskiy, A., et al. (2014). Max pooling in convolutional neural networks.
3. Goodfellow, I., Bengio, Y & ,.Courville, A. (2016). Deep Learning. MIT Press.
4. Hossain, M., et al. (2018). Model evaluation metrics for EMG classification.
5. Huang, J., et al. (2021). Application of transfer learning in CNN-based prosthetic control systems for limited datasets. IEEE Transactions on Medical Robotics and Bionics.
6. Kingma, D. P & ,.Ba, J. (2015). Adam: A method for stochastic optimization. ICLR.
7. LeCun, Y., et al. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
8. Li, X., et al. (2023). Adaptive learning systems and CNNs for personalized prosthetic interaction and user independence. Advanced Robotics&Intelligent Systems.
9. Liu, J., et al. (2019). Real-time control of prosthetic hands using CNN-based classification of muscular signals. Frontiers in Neurorobotics.
10. Mnih, V., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.
11. Nguyen, T., et al. (2022). Real-time sensor data processing and environmental adaptation in smart prostheses using CNNs. Sensors and Actuators Reports.
12. Srivastava, N., et al. (2014). Dropout: A simple way to prevent neural networks from overfitting. JMLR.
13. Tan, M & ,.Le, Q. V. (2020). EfficientNet: Rethinking model scaling for convolutional neural networks. ICML.
14. Yang, L., et al. (2020). Decoding motor intentions through CNNs in Brain-Computer Interfaces (BCIs) for prosthetic control. Journal of Medical and Biological Engineering.
15. Zhang, H., et al. (2021). Precision and adaptability in smart prostheses: The role of CNNs in interpreting EMG signals. IEEE Reviews in Biomedical Engineering.
16. Zhang, X., et al. (2018). EMG signal normalization techniques. Journal of Biomedical Engineering.
17. Zhang, Y., et al. (2018). Personalized prosthetic fitting through CNN-based analysis of individual muscle patterns. Prosthetics and Orthotics International.