ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DISEÑO Y COMPARACIÓN DE MODELOS DE RECONOCIMIENTO DE GESTOS DE LA MANO COMBINANDO APRENDIZAJE SUPERVISADO Y APRENDIZAJE POR REFUERZO

DISEÑO DE UN MODELO DE RECONOCIMIENTO DE 5 GESTOS DE LA MANO QUE FUNCIONE EN TIEMPO REAL USANDO APRENDIZAJE SUPERVISADO DQN (DEEP Q-LEARNING)

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERA/O EN CIENCIAS DE LA COMPUTACIÓN

KEVIN ALEXANDER PÉREZ CRUZ

kevin.perez.03@epn.edu.ec

DIRECTOR: ANGEL LEONARDO VALDIVIESO CARAGUAY

angel.valdivieso@epn.edu.ec

DMQ, Febrero 2024

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESIGN AND COMPARISON OF HAND GESTURE RECOGNITION MODELS COMBINING SUPERVISED AND REINFORCEMENT LEARNING

DESIGN OF A 5 HAND GESTURE RECOGNITION MODEL OPERATING IN REAL-TIME USING SUPERVISED DEEP Q-LEARNING (DQN)

WORK OF CURRICULAR INTEGRATION PRESENTED AS A REQUIREMENT FOR THE DEGREE OF ENGINEER IN COMPUTER SCIENCES

KEVIN ALEXANDER PÉREZ CRUZ

kevin.perez.03@epn.edu.ec

DIRECTOR: ANGEL LEONARDO VALDIVIESO CARAGUAY

angel.valdivieso@epn.edu.ec

DMQ, February 2024

CERTIFICATIONS

I, KEVIN ALEXANDER PÉREZ CRUZ declare that the curricular integration work described herein is of my authorship; that it has not been previously submitted for any degree or professional qualification; and, that I have consulted the bibliographical references included in this document.

_____

KEVIN ALEXANDER PÉREZ CRUZ

I certify that this work of curricular integration was developed by KEVIN ALEXANDER PÉREZ CRUZ, under my supervision.

_____

ANGEL LEONARDO VALDIVIESO CARAGUAY

DIRECTOR

DECLARATION OF AUTHORSHIP

Through the present declaration, we affirm that the curricular integration work described herein, as well as the product(s) resulting from it, are public and will be available to the community through the institutional repository of the National Polytechnic School; however, the ownership of the patrimonial rights corresponds to the authors who have contributed to the development of this work; observing for this purpose the provisions established by the competent body on intellectual property, internal regulations and other rules.

KEVIN ALEXANDER PÉREZ CRUZ

ANGEL LEONARDO VALDIVIESO CARAGUAY

DEDICATION

To my dear parents Mario Perez and Norma Cruz who supported me unconditionally throughout my university career with their advice and encouragement.

To my brothers Luis and Mario Pérez who always looked after my well-being and helped me in difficult times.

To my classmates for always lending me a hand and giving me good encouragement when I needed it.

To all of them and those who always trusted me, I thank them for forging and helping me to build this path full of sacrifice and effort.

ACKNOWLEDGEMENTS

I would like to thank my thesis tutor, Dr. Leonardo Valdivieso for his unconditional help, understanding and guidance during the development of this project.

In addition, I want to thank Dr. Marco E. Benalcazar and Dr. Lorena Barona for their support in terms of their knowledge and experience in this area of research.

I acknowledge Dr. Juan Pablo Vásconez for taking the time to explain the basics and guide us through the plans for this research.

I offer my deepest gratitude to the distinguished members of the Alan Turing Research Laboratory at the *Escuela Politécnica Nacional* for their invaluable support, guidance and technical support through this work process.

Finally, to my thesis partner, Cristian Bastidas for his continuous collaboration and invaluable support during the uncertain and most difficult moments.

TABLE OF CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

RESUMEN

El reconocimiento de gestos de la mano (HGR) es una aplicación importante de la interacción humano-computadora (HCI) que utiliza señales electromiográficas (EMG) y técnicas de aprendizaje profundo. En el presente trabajo se desarrollan dos modelos de HGR para usuarios generales usando el algoritmo de aprendizaje por refuerzo Deep Q-Network (DQN) y el algoritmo Double Deep Q-Network (DDQN), ambos en base al conjunto de datos EMG-EPN-612. Los modelos desarrollados son un ajuste fino de un modelo pre-entrenado con aprendizaje supervisado y basado en una red convolucional (CNN). Para dicho refinamiento, se diseñó una función de recompensa capaz de premiar la constancia en las predicciones del modelo mediante recompensas graduales. También se realizó un post-procesamiento del modelo en base a la medida de tendencia de moda para lidiar con etiquetas intermedias erróneas. Para las pruebas se realizaron experimentos en un conjunto de datos con acceso público el cual contiene 612 usuarios. Posteriormente se realizó una medición y comparación de la precisión de reconocimiento y clasificación entre 6 gestos, incluyendo el “no gesto”. Los resultados demostraron que los modelos ajustados de manera fina con DQN y DDQN presentan valores similares en cuanto a las métricas planteadas. El modelo DQN presenta una mejora del 3.61% y el DDQN una mejora del 3.45% en precisión de reconocimiento con respecto al modelo base. Por otro lado, la precisión de clasificación no ilustra alguna mejora significativa. Finalmente, concluimos que DQN presenta un desempeño en general mayor que DDQN en tareas de clasificación y reconocimiento de gestos de la mano.

PALABRAS CLAVE: DQN, DDQN, ajuste fino, EMG, Reinforcement Learning, función de recompensa, Reconocimiento de Gestos de la Mano.

ABSTRACT

Hand gesture recognition (HGR) is an important application of human-computer interaction (HCI) using electromyographic (EMG) signals and deep learning techniques. In the present work, two HGR models for general users are developed using the Deep Q-Network (DQN) reinforcement learning algorithm and the Double Deep Q-Network (DDQN) algorithm, both based on the EMG-EPN-612 dataset. The developed models are a fine tuning of a pre-trained model with supervised learning and based on a convolutional network (CNN). For such refinement, a reward function capable of rewarding constancy in model predictions through gradual rewards was designed. A post-processing of the model was also performed based on the fashion trend measure to deal with erroneous intermediate labels. For testing, experiments were performed on a publicly accessible dataset containing 612 users. Subsequently, a measurement and comparison of the recognition and classification accuracy between 6 gestures, including "no gesture", was performed. The results showed that the fine-tuned models with DQN and DDQN present similar values in terms of the stated metrics. The DQN model presents a 3.61% improvement and the DDQN a 3.45% improvement in recognition accuracy with respect to the base model. On the other hand, classification accuracy does not show any significant improvement. Finally, we conclude that DQN presents an overall better performance than DDQN in classification and hand gesture recognition tasks.

KEYWORDS: DQN, DDQN, fine-tuning, EMG, Reinforcement Learning, reward function, Hand Gesture Recognition.

1. INTRODUCTION

Nowadays, human beings seek to interact effectively with the computer, and this is becoming the top priority in people's daily life in terms of interactive and intelligent computing [1]. Hand Gesture Recognition (HGR) is widely used in Human-Robot Interaction (HRI) to create user interfaces (UI), games or easy-to-learn applications [2]. The main problem in this type of recognition is to determine the class from a predefined set of classes. In the present project we seek to identify 5-hand-gestures: left hand (wave in), right hand (wave out), fist, open hand (open) and double finger tap (pinch) [3]. Figure 1 shows these gestures, which are considered static as they do not involve forearm movement to perform them.

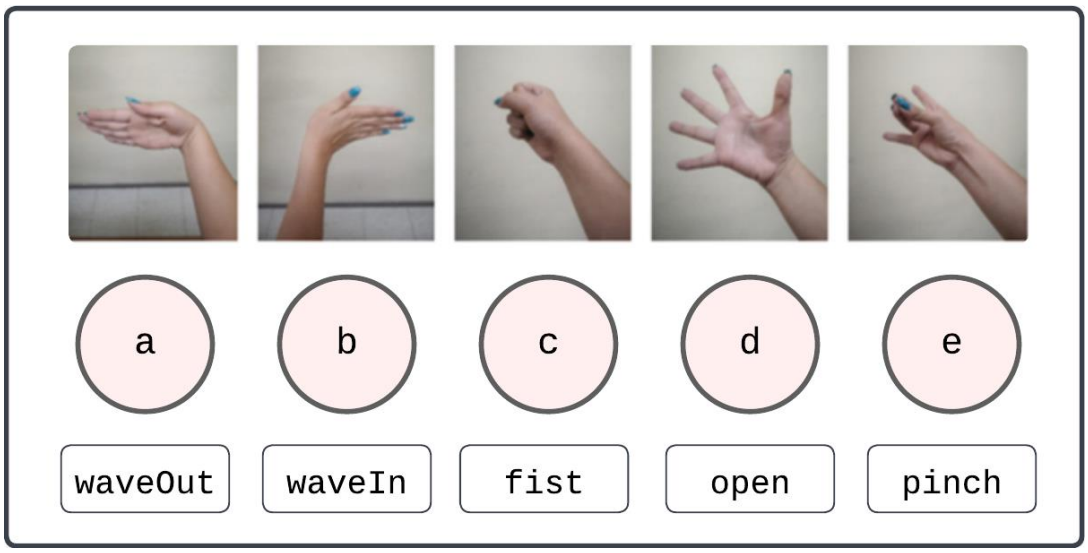

Figure 1. Gestures to be recognized.

The present project will design two models for the recognition of 5 static hand gestures which will use EMG signals and Deep Learning [4], as described in Figure 2.

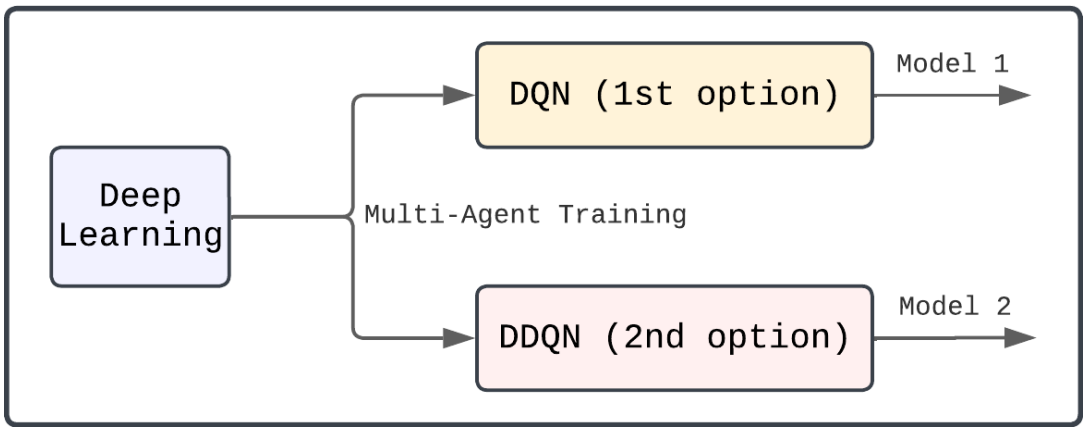

Figure 2. Description of the components of the proposed system.

The components will be organized as follows:

- **Component 1:** implementation of stage 1 and stage 2 (1st option - DQN).
- **Component 2:** implementation of stage 1 and stage 2 (2nd option - DDQN).

The present work corresponds to component 1. DQN (Deep Q-Network) is a reinforcement learning algorithm that seeks to learn optimal policies for sequential decision-making problems by optimizing a cumulative future reward signal. Unlike the popular Q-learning algorithm, which sometimes learns unrealistically high action values due to a maximization step over estimated action values, DQN addresses this problem [5].

First, we will start from a hand gesture recognition model previously trained with supervised learning, using techniques such as artificial neural networks (ANN) and fully connected layers (FCL). The refinement of the model will aim to obtain a predictive model that works in real-time (with a response time of less than 300ms from the input EMG signal until the system returns a label) to predict compact sequences of labels, using reinforcement learning. To this end, a deep learning model will be combined with the Q-learning reinforcement learning algorithm. The benefit of Q-learning lies in the fact that, being an "off-policy" algorithm, it allows us to interact with the environment in one way, while learning a completely different strategy. The interaction of the environment with the agent through rewards and actions will allow predicting compact sequences of labels composed of 3 blocks of labels: a) labels that identify the hand in the resting state, b) labels that identify the movement performed and finally c) labels that indicate again the hand in the resting state.

Once the DQN model has been designed and evaluated, a comparison of results with the model fitted with DDQN will be performed. The main metrics to compare are classification accuracy, recognition, and response time.

1.1 General Objective

To develop a real-time recognition model of five hand gestures using the DQN (Deep Q-learning) reinforcement learning technique based on the EMG-EPN-612 dataset [6].

1.2 Specific Objectives

- To review the state of the art of the DQN learning algorithm, its applications in the field of hand gesture recognition (HGR) and in the prediction of compact tag sequences.
- Design and evaluate a 5-hand-gesture recognition model operating in real-time, using DQN and receiving EMG signals.
- Evaluate and compare the proposed model with the results of component 2 model with DDQN (Double Deep Q-learning) in terms of percentage of classification accuracy, recognition, and response time.

1.3 Scope

This component will focus on improving and optimizing a 5-hand-gesture recognition model using DQN (Deep Q-learning) supervised learning.

For this purpose, Figure 3 summarizes the components of the DQN model:

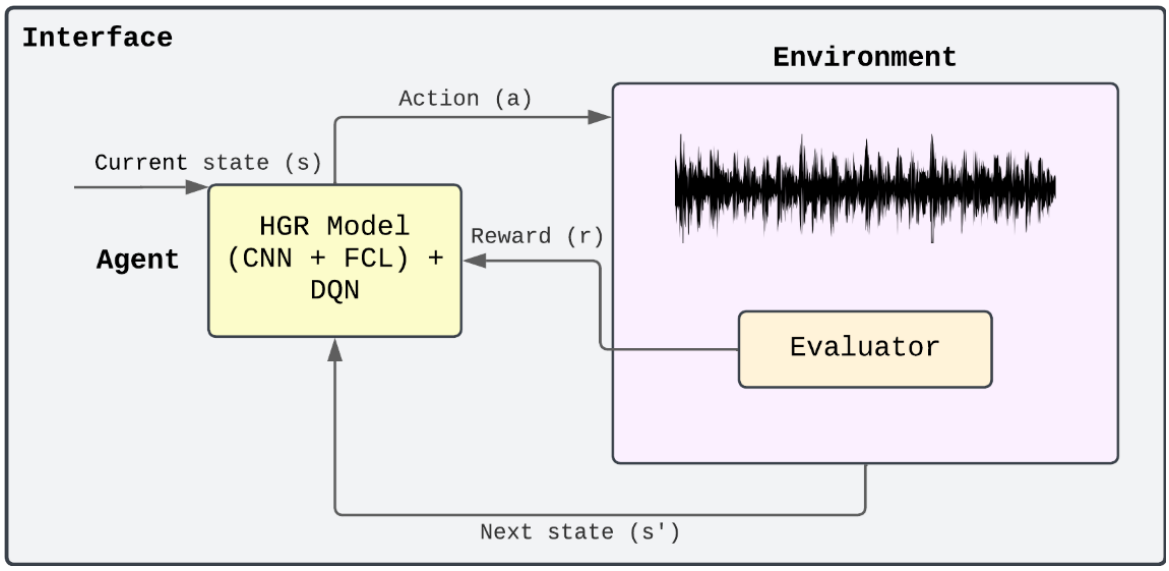

Figure 3. Components of the DQN Model.

The figure above consists of three main components: an agent, an environment, and an evaluator. The agent is the HGR model, which receives as input the current state of the window (s) of the EMG signal and returns as output (a) a label that is coupled to a sequence of predictions. The environment is composed of the window that runs through the EMG signal and an evaluator, which will have the ground truth to compare it with the sequence of predicted labels and return a reward (r). During the development of this component, the reward function has been defined. Finally, the environment returns the next portion of the EMG signal, and the cycle of the diagram is repeated.

1.4 Theoretical Framework

This section contains the concepts and theoretical foundations on which the information of this project will be based.

1.4.1 Hand Gesture Recognition

The field of Hand Gesture Recognition (HGR) has gained significant traction, especially for applications in human-computer interaction (HCI) [7]. The Myo Armband is a pivotal device in this domain, featuring electromyography (EMG) sensors and inertial measurement units (IMUs) to track and interpret muscle movements and orientations in the forearm [8]. These sensors are key to distinguishing various hand gestures.

Utilizing the Myo Armband's EMG sensors, researchers can capture the muscle activity corresponding to different hand movements. The additional data provided by IMUs, which include accelerometers and gyroscopes, enhances the accuracy of gesture detection [9].

Several research works have shown the practicality of the Myo Armband in HGR. For example, there were demonstrated the integration of deep learning techniques with EMG data from the Myo Armband for effective real-time gesture recognition. This highlights the synergy of machine learning and EMG data for improved gesture detection [10].

Furthermore, the Myo Armband can be used for recognizing brief hand gestures using wavelets and support vector machines. Some studies showcase the adaptability of the Myo Armband in various computational models [11].

The above underlines the effectiveness of the Myo bracelet in the development of advanced HGR systems, essential for a more fluid and natural HCI.

1.4.2 Reinforcement Learning

Reinforcement Learning (RL) is a branch of Artificial Intelligence that studies how agents can learn from their own experience and improve their behavior through interaction with the environment. Unlike other learning paradigms, such as supervised learning or unsupervised learning, RL does not require labeled data or a predefined objective function but relies on the reward or punishment that the agent receives because of its actions. The goal of RL is to find an optimal policy that maximizes the accumulated reward over time. RL has proven its effectiveness in a wide variety of complex and challenging problems, such as robotic control, games, optimization, planning, and decision making [12].

The following Table 1 briefly summarizes the elements that make up a typical system based on reinforcement learning and some other that are related to this study. In addition, Figure 4 illustrates how these elements interact with each other.

Table 1. Elements of Reinforcement Learning.

Element	*Description*
Agent and Environment	The agent, as the learner and decision-maker, interacts with the environment, which encompasses everything outside the agent. The agent's actions influence the environment, which in turn responds with new situations and rewards [13].
Reward Signal	The agent receives a numerical reward signal from the environment at each time step. This signal serves as an indicator of the agent's performance and its goal. The agent's objective is to maximize the total reward it receives over time [13].
Action	This refers to the limited possibilities of the agent in the environment, in this case, to the 5 gestures and the "non-gesture".
State	It denotes the immediate observation from the environment. In this research, the state is determined by the spectrograms of EMG signals.
Interface	A system that enables the interaction between two entities, namely the agent and the environment.

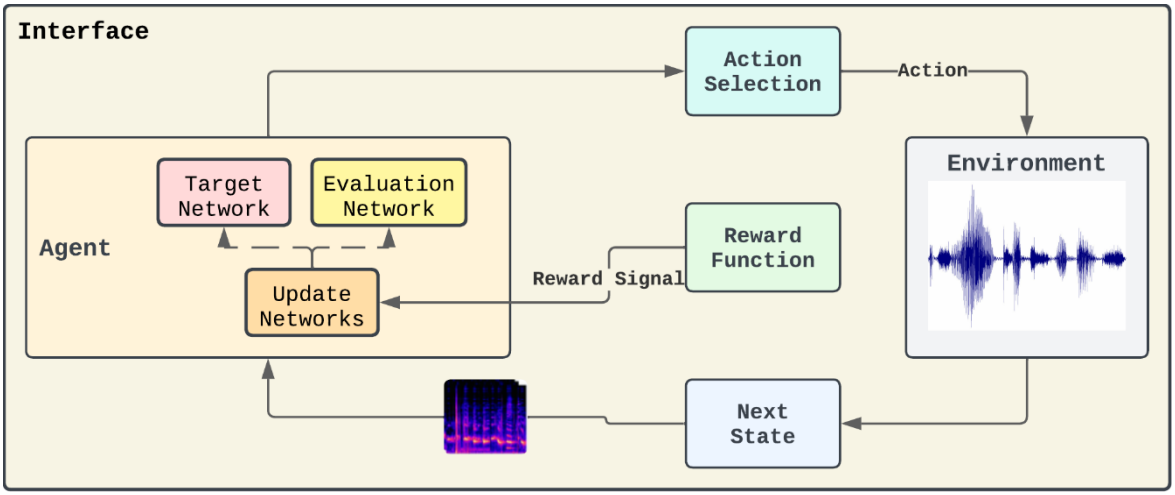

Figure 4. Reinforcement Learning elements related to DQN algorithm.

1.4.3 Q-learning

Q-learning is a reinforcement learning algorithm that aims to learn an action-value function, Q(s, a), which represents the expected value of taking an action *a* in a state *s* and following an optimal policy thereafter [14].

The fundamental mechanism of Q-learning is the iterative update of the Q function using the Bellman equation, which links the value of Q(s, a) with the value of Q(s', a') for the next state and action. The update is done using observed rewards and estimated Q values [14].

Q-learning is an off-policy and model-free algorithm, meaning it doesn't require knowledge of the environment's transition model and can learn from transitions generated by an exploration policy different from the optimal one [15].

Q-learning is used to make decisions in reinforcement learning by choosing the action that maximizes the Q value for the current state. This involves a balance between exploitation and exploration, which can be regulated using an action selection strategy like the ε-greedy rule [15].

1.4.4 Deep Q-Network (DQN)

Deep Q-Network (DQN) is a reinforcement learning algorithm that merges Q-learning with deep neural networks. This combination allows reinforcement learning to be applied in complex, high-dimensional environments such as video games or robotics. Q-learning is a value-based reinforcement learning method that estimates a state-action value function, which represents the expected return when taking an action in a state and following an optimal policy [16].

DQN is a variant of Q-learning that approximates the value function with a neural network and uses techniques like experience replay memory and fixed targets to enhance the stability and performance of learning. DQN has proven its ability to outperform humans in several Atari games using only pixel images as input [17].

1.4.5 Fine-Tuning

Fine-tuning is a technique to adapt a pre-trained model to a specific task, by adjusting some or all the model's parameters with new data. It is useful when we have a large and general pre-trained model for a similar problem, but not enough data for the task at hand. Depending on the level of similarity between the tasks and the features, we can fine-tune the entire model or only a subset of its layers. Fine-tuning can improve the performance and the efficiency of the model, but it can also reduce its robustness to distribution shifts [18].

Fine-tuning is a transfer learning technique that consists of modifying the weights and biases of a deep learning model that has been pre-trained on a source dataset [19]. It has several benefits for improving the performance of deep learning models [20] which are listed in the Table 2.

Table 2. Fine-Tuning benefits in Deep Learning Models performance

Benefict	Description
Preventing overfitting	By using the weights of a pre-trained model on a large and diverse dataset, the risk of memorizing the training data is avoided and the model's generalization is improved.
Speeding up training	By starting from an initialization close to the optimal solution, the number of iterations needed to reach convergence is reduced and time and computational resources are saved.
Overcoming data scarcity	By leveraging the knowledge extracted from a source dataset, an effective model can be trained with a small or limited target dataset, which is common in many problems of computer vision, gesture recognition, etc.

1.4.6 Related Works

To contextualize the work done, some research on hand gesture recognition (HGR) research with reinforcement learning (RL) is summarized in the Table 3.

Table 3. Summary of related works.

Related Work	Applied techniques	Classification Results (%)	Recognition Results (%)	Conclusions
Hand Gesture Recognition Using EMG-IMU Signals and Deep Q-Networks	Agent-based in DQN on an ANN	97.50%±1.13%	88.15%±2.84%	This research concludes that DQN can learn policies from online

				experiences [21].
A Hand Gesture Recognition System Using EMG and Reinforcement Learning: A Q-learning Approach	Agent-based in Q-learning	90.78%	87.51%	This work shows that Q-learning can learn from online experiences [22].
A comparison of EMG-based hand gesture recognition systems based on supervised and reinforcement learning	CNN-based model (supervised learning)	90.49%±9.7%	86.83%±11.30%	The simplest model based on supervised learning has better results than the reinforcement learning approach [23].
	Deep Q-Network model (reinforcement learning)	76.27%±11.9%	67.89%±13.3%	
A Deep Q-Network-based hand gesture recognition system for control of robotic platforms	DQN agent-based	97.45%±1.02%	88.05%±3.10%	It is possible to design a human-machine interface based on an EMG-IMU-based HGR system and IMU signals [24].

In summary, the studies on hand gesture recognition using reinforcement learning techniques showcase promising results. The DQN-based models, particularly in the contexts of EMG-IMU signals, consistently demonstrated high accuracy in classifying and

recognizing hand gestures. Additionally, the application of Q-learning also yielded competitive results, indicating the adaptability of RL approaches to this domain.

1.4.7 Contribution

Our research seeks to build upon existing studies by directly comparing the DQN and DDQN RL techniques in Hand Gesture Recognition (HGR) using the EMG-EPN-612 dataset. The key contributions of our project are summarized as follows:

- We created two user-general reinforcement learning models for HGR, employing DQN and DDQN algorithms. These models take spectrograms of EMG signals as their input.
- To identify the algorithm best suited for our context, we assessed the DQN and DDQN models based on classification and recognition accuracies.
- We formulated a comprehensive reward function tailored to the specifics of our scenario, considering recommendations from [25]. This adaptable reward function can be easily tailored for future enhancements and research.

2. METHODOLOGY

In this segment, we introduce a fundamental process aimed at tackling the problem of EMG-based HGR, as depicted in the Figure 5. This structure includes an RL multi-agent environment, reward function, hyperparameter optimization, fine-tuning, and post-processing. As noted earlier, DQN employs a deep neural network as its target network. In our suggested implementation, we will use a CNN-based model from prior research by [4] in two phases. The first phase involves initializing the CNN with randomly allocated weights. The second phase involves fine-tuning the CNN using a pre-trained model. Both models are based on the same GoogLeNet CNN architecture.

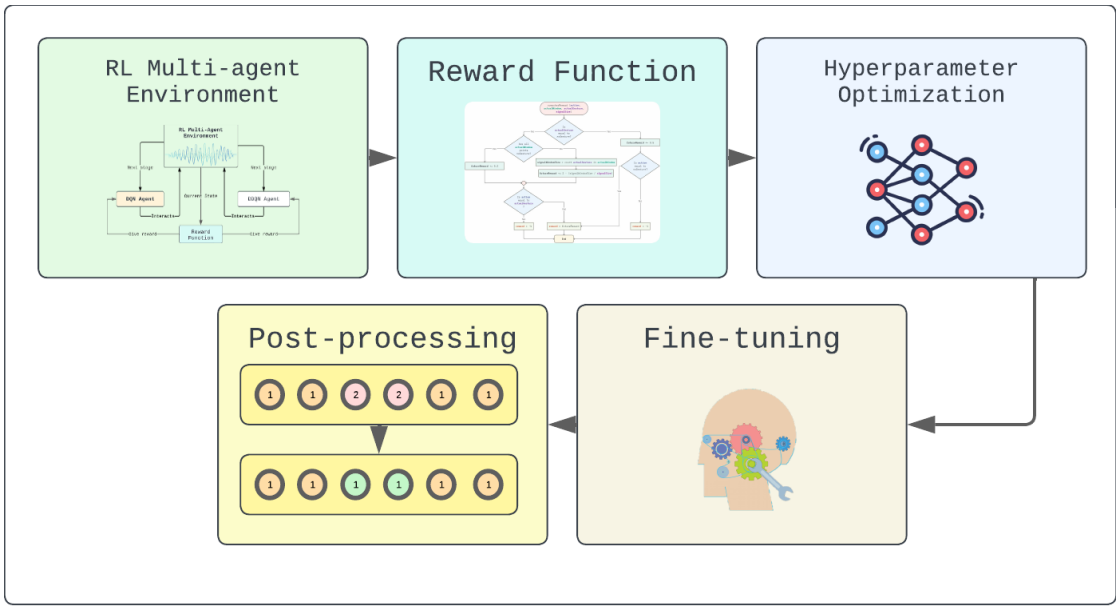

Figure 5. Applied Methodology

As mentioned in previous studies, we used the same EMG-EPN-612 dataset [6] for our research, as we employed the same CNN architecture. This extensive dataset includes EMG signals from 612 users, recorded using the Myo armband over a five-second duration, resulting in 1000 data points per user for HGR model development and benchmarking. The signals represent five unique hand gestures: wave-in, wave-out, fist, open, and pinch. The dataset also includes EMG signals from users' relaxed hands, referred to here as "*noGesture*". These gestures are depicted in Figure 1. The dataset is divided into two groups: one group of 306 users (with 150 samples per user) for training, which includes ground truth data (the part of the signal where the gesture was performed), and the remaining 306 users for testing.

2.1 RL multi-agent environment

To guarantee an equitable comparison between DQN and DDQN, it's essential to equip both agents with a similar learning environment during their respective training stages. Monitoring their behavior from the outset lays the groundwork for additional analysis during the fine-tuning and validation phases. It's noteworthy that, in the context of this research scenario, the environment functions under two distinct actions, states, and rewards, as illustrated in Figure 6.

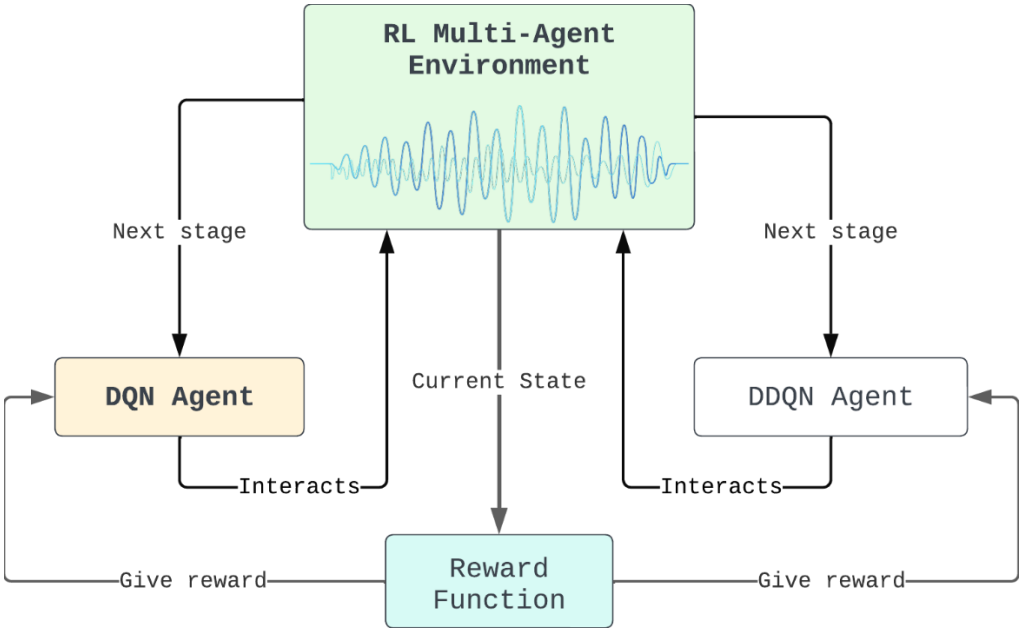

Figure 6. RL Multi-Agent Environment

2.2 Reward function

Creating a functional reward system for a Reinforcement Learning agent poses a significant challenge [25]. Our reward system considers the proportion of the effective EMG signal present in the current window, the complexity of the current gesture, and the agent's performance consistency.

The reward system is designed to gradually assign rewards based on the degree of similarity between the predicted and expected actions. If the model's prediction is correct, the reward is the percentage of the effective signal. However, if the prediction is incorrect, the reward is -1. This percentage is reduced by 2 to account for the increased complexity of predictions at the start and end of the effective signal, where full information may not be available. As a result, if the current window encompasses 100% of the effective signal, the maximum reward is 1.

We've noticed that the default gesture (*noGesture*) is relatively easy for the model to predict, so the reward for these instances is reduced. Instead of a +1 reward for a correct prediction, we award +0.5 when processing a complete default gesture signal or +0.25 when processing signal parts that don't correspond to a gesture. This aspect of the reward function, combined with the gradual reward, encourages the model to make correct predictions, providing a dense learning signal [25]. Figure 7 summarizes how the reward function works.

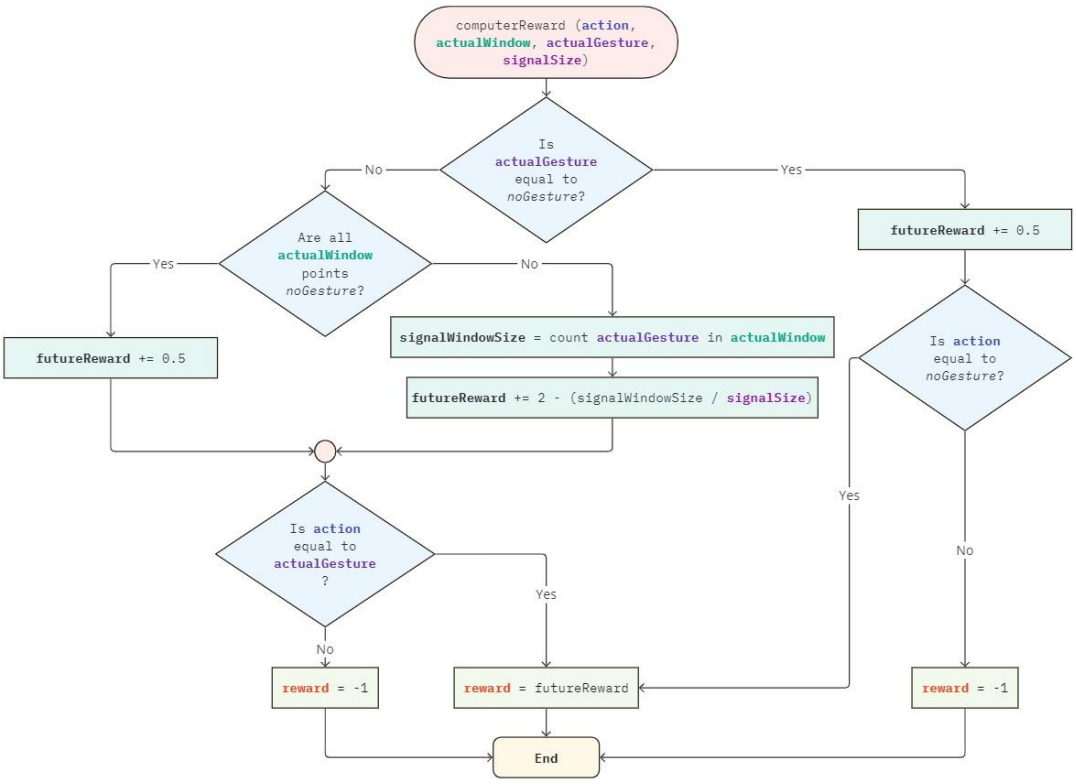

Figure 7. Designed Reward Function

In conclusion, if the model consistently makes correct decisions over several windows, it receives a reward. This strategy encourages the model to maintain steady performance, as it will lead to additional rewards over the EMG signal. By assigning rewards in this manner, we can enhance the model's recognition phase. This method is referred to as an intrinsic motivation signal, which is inspired by curiosity or a desire for novelty and biases the model towards consistency [25]. To implement this, we consider the following factors:

- A counter is kept, which is incremented each time the model makes a correct prediction within a window.
- At the end of the current window, the long-term reward is calculated and given to the agent.

- If a prediction is incorrect, the counter is reset to zero, and the process begins anew.

As we aim to provide larger rewards for more consecutive correct predictions, we use an exponential function (Equation 1) to calculate the additional reward. Here, α and β determine the reward's behavior, and W represents the number of consecutive correct predictions.

$$R(W) = \alpha e^{\beta W} - \alpha$$

Equation 1. Long-term reward function

2.3 Hyperparameters

After setting up the environment, which included the target network and reward function, several experiments to pinpoint the best hyperparameters for the training process are carried out. Given that the Convolutional Neural Network (CNN) was initialized from scratch, we ran tests with datasets that included 1 user, 75 users, and the full training dataset of 306 users.

Table 4 provides a summary of our experiments. The learning rate dictates the speed at which the algorithm modifies the network based on new learnings. The epsilon decay manages the balance between exploration and exploitation by gradually reducing random actions over time. The target smooth factor influences the stability of the network updates during training.

Table 4. Hyperparameter experiments for the DQN model starting from zero.

Users Count	Episodes	Learn Rate	Epsilon Decay	Target Smooth Factor
1	15000	1.00E-02	1.00E-05	1.00E-04
1	15000	1.00E-03	1.00E-05	1.00E-04
1	15000	1.00E-05	1.00E-04	1.00E-03
1	5835	1.00E-03	1.00E-05	1.00E-03
1	30000	1.00E-06	1.00E-05	1.00E-03
75	44693	1.00E-02	1.00E-04	1.00E-03
75	45000	1.00E-06	1.00E-05	1.00E-03
75	44998	1.00E-02	1.00E-05	1.00E-04
306	91800	1.00E-05	1.00E-04	1.00E-03
306	91800	1.00E-06	1.00E-05	1.00E-04

306	91800	9.00E-06	1.00E-04	1.00E-03
306	45432	1.00E-06	1.00E-05	1.00E-03

2.4 Fine-tuning

To accelerate our research, we utilized an existing CNN user general model from [4], which is grounded on GoogLeNet and has been trained on an identical dataset. We enhanced this model by fine-tuning it using the DDQN algorithm. The fine-tuning process necessitated significant work in optimizing hyperparameters. As a result, we conducted iterative experiments to achieve a satisfactory level of convergence in the training progress, following the same methodology outlined in the hyperparameters section. The fine-tuning experiments are summarized in Table 5.

Table 5. Hyperparameter fine-tuning experiments for the DQN model.

Users Count	Episodes	Learn Rate	Epsilon Decay	Target Smooth Factor
306	45900	1.00E-07	1.00E-06	1.00E-04
306	45900	1.00E-06	1.00E-05	1.00E-04
75	22500	1.00E-07	1.00E-05	1.00E-04
75	22500	1.00E-06	1.00E-05	1.00E-04
1	15000	1.00E-08	1.00E-03	1.00E-04

2.5 Post-processing

Post-processing is performed to refine the predicted EMG signals. The main goal is to detect and process ongoing sequences of valid gestures while discarding isolated occurrences of disregarded or insignificant actions. Disregarded gestures (*noGesture*) are easy to identify, hence they are given less focus and don't need extra modifications during processing.

Nonetheless, it's crucial to consider that a sequence of valid gestures might be succeeded by a disregarded gesture. To handle this scenario, our post-processing method looks for ongoing sequences of valid gestures in the predicted data. Once these sequences are found, all following classes within that sequence are substituted with the most frequent class. This strategy guarantees that each intended gesture is precisely depicted while

preserving the overall coherence and continuity of the processed EMG signal. Figure 8 illustrates the scenarios that this method addresses.

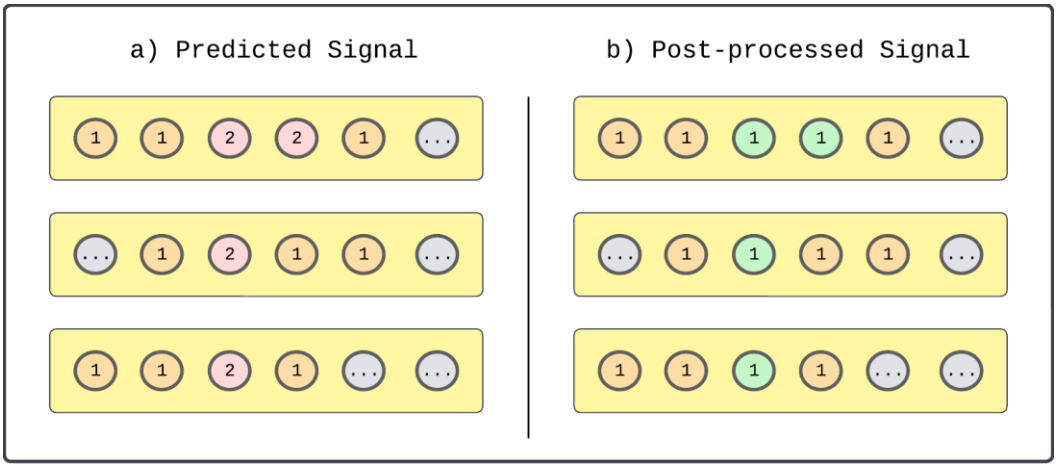

Figure 8. Post-processing stage. a) Raw predicted data from the RL model. b) Post-processed data ready for an application.

Using this post-processing technique, the model is capable of efficiently identifying and categorizing intricate patterns from initial predictions, thereby enhancing the precision of classification and recognition.

3. RESULTS

This section will present the results obtained from the evaluation of our proposed general user HGR models. This evaluation involved testing a variety of hyperparameters and assessing the performance of the model both with and without the implementation of the post-processing stage. The hyperparameters that remained unchanged for the Deep Q-Network (DQN) and Double DQN (DQN) models, which were developed from scratch, it means the baseline model hyperparameters, are detailed in the following Table 6.

Table 6. Baseline model hyperparameters

Hyperparameter	Fixed Value
Learn Rate	1.00E-06
Gradient Threshold	1
Optimizer	“adam”
Gradient Threshold Method	“l2nrom”
Used Device	“gpu”
Target Smooth Factor	1.00E-04
Minibatch Size	32
Number of steps to look ahead	1
Discount Factor	0.98
Experience Buffer Length	50
Epsilon Decay	1.00-E05
Use Double DQN?	TRUE
Save Experience Buffer with Agent?	TRUE
Max Episodes	918000

Considering the intricate process of pinpointing the best hyperparameters during the fine-tuning stage, Table 7 lists the most efficient hyperparameters that we found for the defined DQN and DDQN models.

Table 7. Best-founded hyperparameters for the DQN and DDQN models

Hyperparameter	Fixed Value
Learn Rate	1.00E-07
Gradient Threshold	1
Optimizer	“adam”
Gradient Threshold Method	“l2nrom”

Used Device	“gpu”
Target Smooth Factor	1.00E-04
Minibatch Size	32
Number of steps to look ahead	1
Discount Factor	0.98
Experience Buffer Length	50
Epsilon Decay	1.00-E06
Use Double DQN?	TRUE
Save Experience Buffer with Agent?	TRUE
Max Episodes	45900

The recognition accuracy results of our proposed general-user HGR model are shown in Figure 9. Although the models we developed do not exceed the accuracy of the existing post-processed model (which achieves accuracies of 83.26%, 82.87%, 91.91%, and 91.98% respectively). It's important to note that using existing models results in an accuracy improvement of up to 9.11% for the post-processed DQN model and up to 8.69% for the post-processed DDQN model compared to a model trained from scratch. However, when we focus on models without post-processing, our proposed models show higher recognition accuracy. Specifically, the use of fine-tuning results in a 3.45% increase over the DDQN and a 3.61% increase with the DQN model. In addition, DDQN models trained from scratch improve recognition by 1.59%, while DQN models result in a 2.46% increase in recognition accuracy compared to the existing models. Moreover, fine-tuned models without processing also perform better than both the models trained from scratch and the post-processed models.

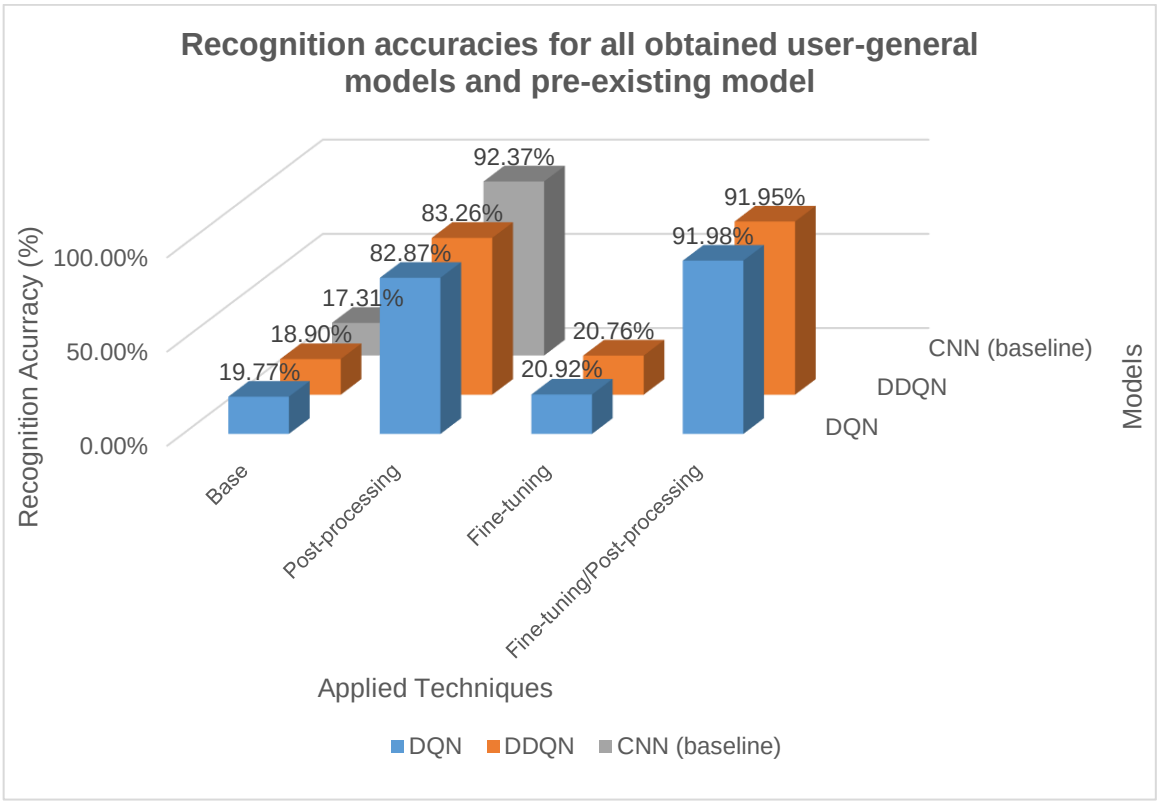

Figure 9. Comparison of the recognition accuracy between the obtained *user*-general models and the pre-existing (baseline) model by applying different improvement techniques.

The results of the classification accuracy for our suggested general-user HGR models are illustrated in Figure 10. The data shows that there isn't a significant difference in classification accuracy between the models that are post-processed and those that aren't. However, it's a model trained from scratch doesn't exceed the performance of the pre-existing model in this research, whether it's DDQN or DQN. Interestingly, the pre-existing post-processed DDQN model outperforms our suggested post-processed DDQN model by 8.19% and 8.18%, respectively. When it comes to fine-tuned models, these models enhance the classification accuracy of DDQN by up to 0.89%, and of DQN by 0.92%. Moreover, when looking at the fine-tuned models that aren't post-processed, DDQN fine-tuned improves classification accuracy by 0.75% and DQN fine-tuned by 0.81%. Furthermore, DQN shows superior performance than DDQN, regardless of whether they are fine-tuned or not. Specifically, the post-processed DQN model that's fine-tuned outperforms its equivalent by 0.03%, while the fine-tuned DQN model without post-processing advances it by 0.06%. Additionally, the post-processed DQN model trained from scratch surpasses its non-post-processed counterpart by 0.11%, and the DQN model trained from scratch without post-processing sees a slight improvement of 0.01%.

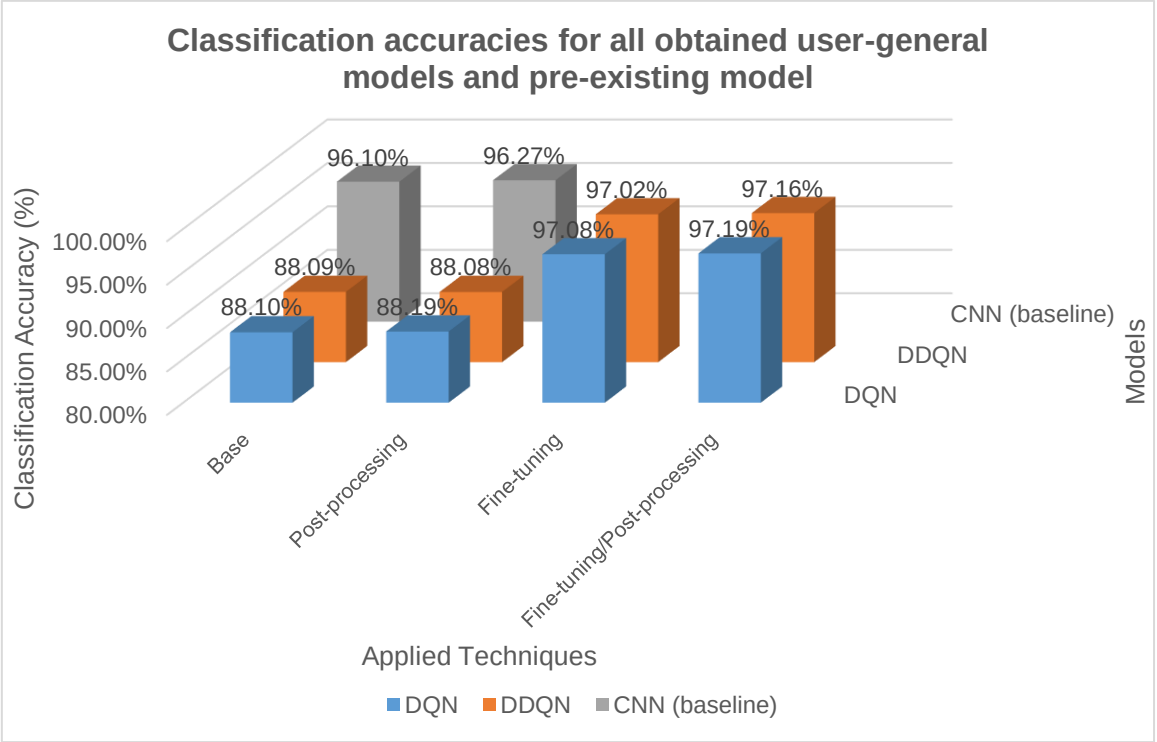

Figure 10. Comparison of the classification accuracy between the obtained user-general models and the pre-existing (baseline) model by applying different improvement techniques.

Figure 11 showcases the confusion matrices for the best-performing general-user DQN and DDQN model's classification. Figure 11a shows a confusion matrix for the DQN model without post-processing, where high values on the main diagonal indicate good performance. In contrast, Figure 11b presents the confusion matrix for the DDQN model without post-processing, with values on the main diagonal like those of the DQN, suggesting comparable efficiency. Figure 11c reveals an improvement in the performance of the DQN model with post-processing, as evidenced by increased values on the main diagonal compared to Figure 11a. Finally, Figure 11d illustrates a confusion matrix for the DDQN model with post-processing, where an increase in correct predictions is observed compared to its version without post-processing, indicating that post-processing can boost the accuracy of both DQN and DDQN models.

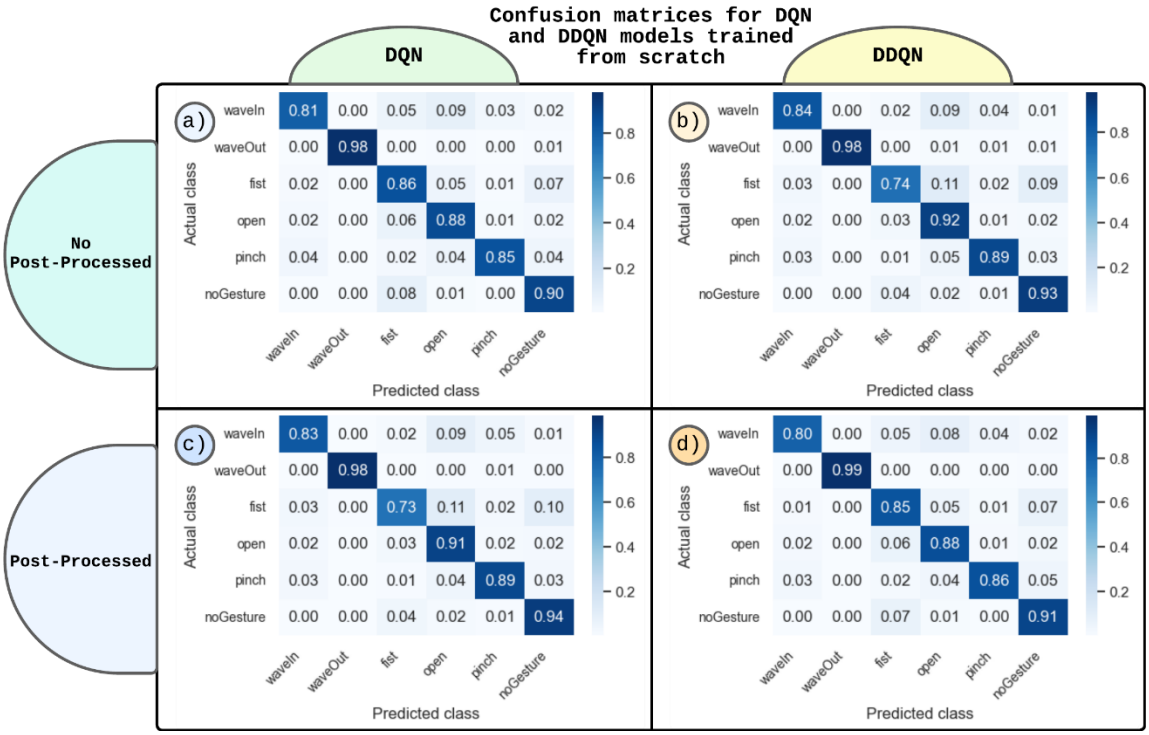

Figure 11. Confusion matrices for DQN and DDQN models trained from scratch. a) DQN no post-processed model, b) DDQN no post-processed model, c) DQN post-processed model, d) DDQN post-processed model.

On the other hand, Figure 12 presents the confusion matrices for the top performing fine-tuned DQN and DDQN models. Figure 12a, representing the DQN without post-processing, shows a good performance evidenced by the high values on the main diagonal. Figure 12b, corresponding to the DDQN without post-processing, presents a similar prediction efficiency to that of the DQN, as can be seen from the values on the main diagonal. On the other hand, Figure 12c, corresponding to the DQN with post-processing, shows an improvement in performance, evidenced by the increased values on the main diagonal compared to Figure 12a. Finally, Figure 12d, representing the DDQN with post-processing, shows an increase in correct predictions compared to its version without post-processing, suggesting that post-processing can improve the accuracy of both DQN and DDQN fine-tunned models.

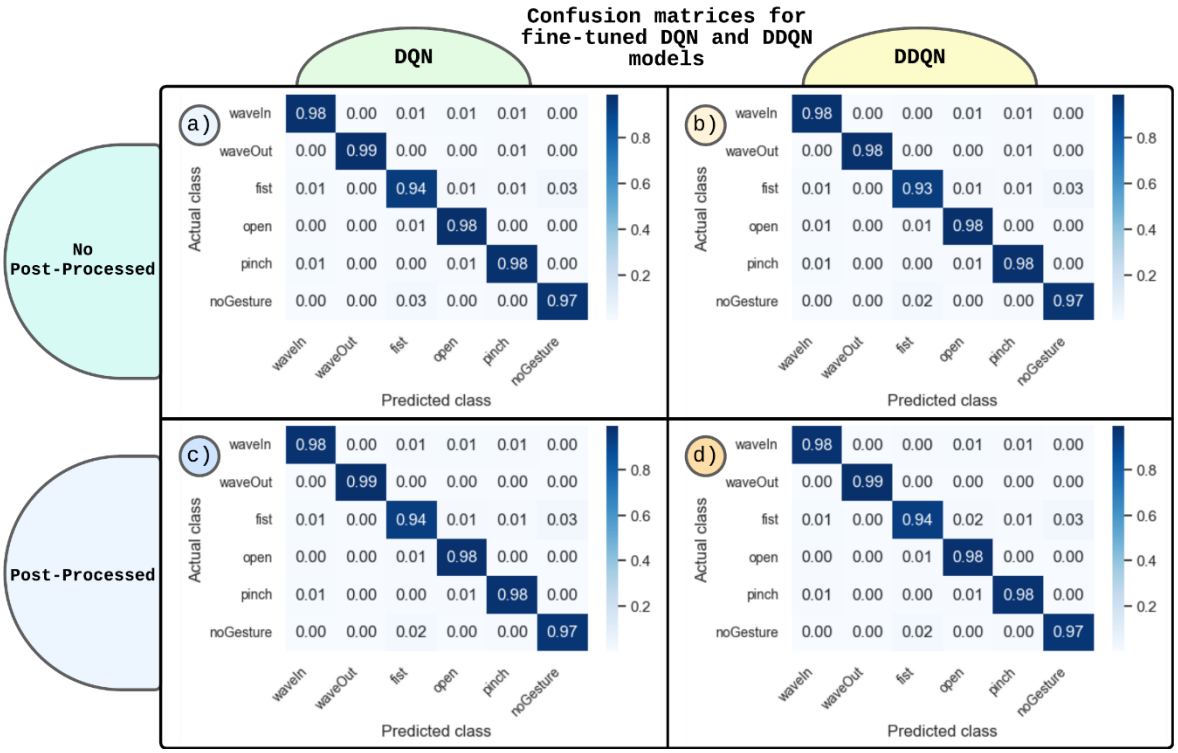

Figure 12. Confusion matrices for fine-tuned DQN and DDQN models. a) DQN no post-processed fin-tuned model, b) DDQN no post-processed fine-tuned model, c) DQN post-processed fine-tuned model, d) DDQN post-processed fin-tuned model.

It's clear to conclude the models depicted in Figure 11 are less effective than those in Figure 12. Specifically, the least accurately classified gesture in the fine-tuned models achieves up to 93%, whereas the least accurately classified gesture in the models trained from scratch only reaches a performance level of 73%.

Finally, the detailed examination of the comparison between DQN and DDQN reveals minor but important variations in their effectiveness for recognition and classification tasks. When it comes to recognition, DQN typically surpasses DDQN, especially in scenarios that do not involve fine-tuning. Table 8 summarizes the experiments performed to obtain the general user model with the best metrics. For greater understanding, experiments that start with "*ft*" imply that they are a fine-tuning of the base model, otherwise it is a model experiment from scratch. Furthermore, all these experiments were trained with 306 user's information, this is referenced in the "*u306*" nomenclature. For the reward, 2 function variations were experimented: linear (*lin*) and exponential (*exp*). The "*gtr*" nomenclature implies that ground truth is used to give the reward. The hyperparameters learning rate, epsilon decay and target smooth factor are linked to "*lr*", "*ed*" and "*tsm*" nomenclature, respectively. Numerical values close to the hyperparameter nomenclature imply a negative exponential factor, e.g., lr6 means a value of 1e-06 for the learning rate.

Table 8. Best model experiments obtained.

Experiment	Name	Description
E1	ft-u306-gtr-exp-lr6-ed5-tsm4	Fine-tuned model with an exponential reward function based on the ground truth; learning rate = 1e-06; epsilon decay = 1e-05; target smooth factor = 1e-04
E2	ft-u306-gtr-exp-lr7-ed6-tsm4	Fine-tuned model with an exponential reward function based on the ground truth; learning rate = 1e-07; epsilon decay = 1e-05; target smooth factor = 1e-04
E3	u306-exp-gtr-lr6-ed5-tsm3	Model trained from scratch with an exponential reward function based on the ground truth; learning rate = 1e-06; epsilon decay = 1e-05; target smooth factor = 1e-03
E4	u306-gtr-exp-lr5-ed4-tsm3	Model trained from scratch with an exponential reward function based on the ground truth; learning rate = 1e-05; epsilon decay = 1e-04; target smooth factor = 1e-03
E5	u306-gtr-exp-lr6-ed5-tsm4	Model trained from scratch with an exponential reward function based on the ground truth; learning rate = 1e-06; epsilon decay = 1e-05; target smooth factor = 1e-04

E6	u306-gtr-lin-lr5-ed4-tsm3	Model trained from scratch with a lineal reward function based on the ground truth; learning rate = 1e-05; epsilon decay = 1e-04; target smooth factor = 1e-03

Figure 13 illustrates the recognition accuracy of six different models (E1 to E6) using two methods: Recognition DQN and Recognition DDQN. Each model corresponds to a different configuration of parameters such as learning rate, epsilon decay, and target smooth factor, as detailed in the Table 8 provided. The graph shows that the model E2, a fine-tuned model with an exponential reward function based on the ground truth and a learning rate of 1e-07, achieved the highest recognition score for both DQN and DDQN methods at approximately 20.76% and 20.92%, respectively. On the other hand, model E6, which was trained from scratch with a linear reward function based on the ground truth and a learning rate of 1e-05, had the lowest performance for the DDQN method at about 15.99%. Similarly, model E3, trained from scratch with an exponential reward function based on the ground truth and a learning rate of 1e-06, had the lowest performance for the DQN method at around 16.10%.

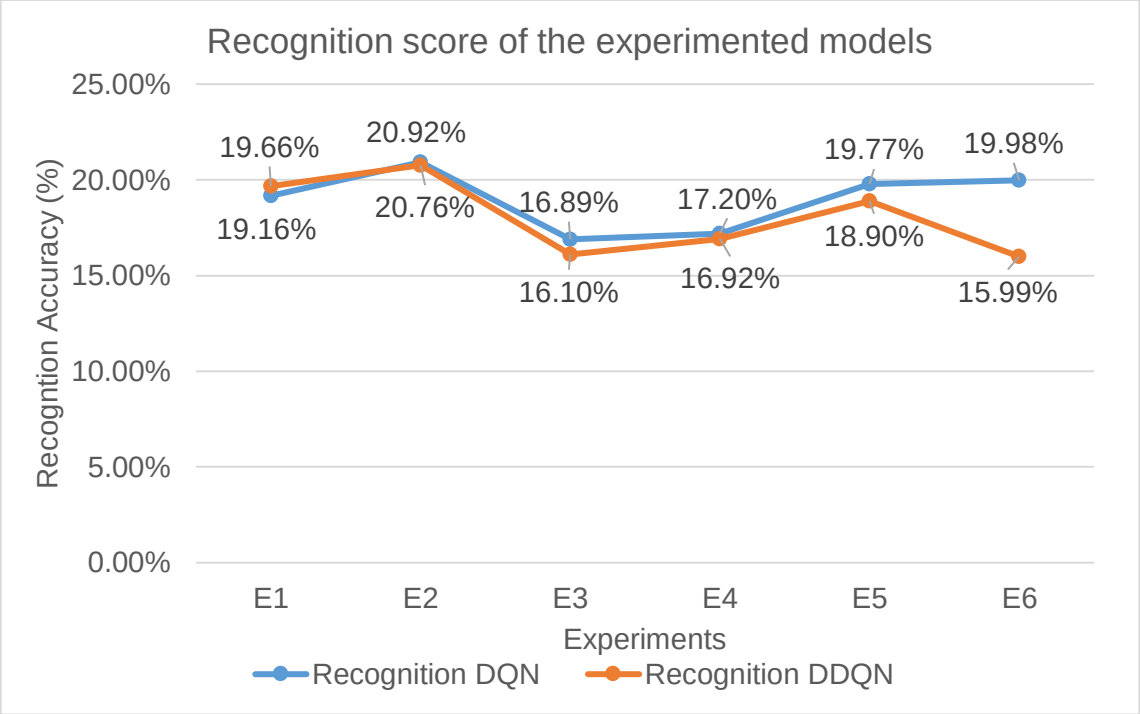

Figure 13. Comparison of the recognition accuracy between the experimented models.

Figure 14 presents the classification success rates of six distinct models (E1 to E6), each employing one of two methods: Classification DQN and Classification DDQN. Each model is defined by a unique set of parameters, as outlined in the provided Table 8.

The graph reveals that the Classification DQN method begins with a 95.45% accuracy at E1, reaches its peak at 97.02% at E3, and concludes at 87.12% at E6. Conversely, the Classification DDQN method starts at a lower 84.64% at E1, but it overtakes Classification DQN from E4 and finishes slightly higher at 87.43% at E6.

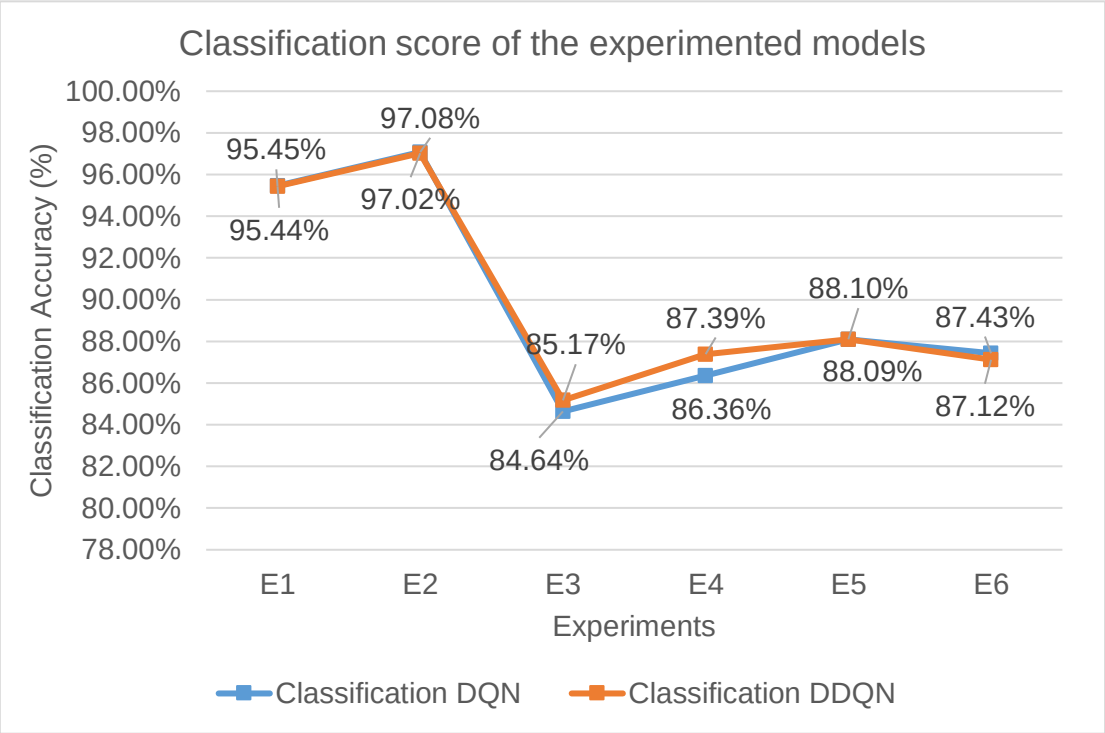

Figure 14. Comparison of the classification accuracy between the experimented models.

The code used for this work is hosted in the following GitHub repository:

https://github.com/laboratorioAI/2024_EMG_DQN_DDQN

Belonging to the Alan Turing Artificial Intelligence Laboratory of the *Escuela Politécnica Nacional*.

4. CONCLUSIONS AND RECOMMENDATIONS

4.1 Conclusions

The objective of this research was to create real-time hand gesture recognition models using the Deep Q-Network (DDQN) reinforcement learning method, utilizing the EMG-EPN-612 dataset. During the fine-tuning phase, optimal hyperparameters for both the fine-tuned DQN and DDQN were discovered. The findings indicated that while our suggested models did not exceed the recognition accuracy of the pre-existing post-processed model, they demonstrated superior recognition accuracy when focusing on non-post-processed models due to the design of the reward function. Fine-tuning resulted in a 3.61% improvement in DQN and a 3.45% increase in DDQN model's recognition accuracy compared to the pre-existing models. Moreover, fine-tuned models without processing also outperformed both the models trained from scratch and the post-processed models.

In terms of classification accuracy, there was no significant difference between post-processed and non-post-processed models. However, a model trained from scratch did not exceed the pre-existing model in this study. Fine-tuned models increased the classification accuracy of DQN by up to 0.92%, and of DDQN by 0.89%. Furthermore, fine-tuned models that are not post-processed improved DQN’s performance by 0.81% and DDQN’s by 0.75%. Finally, DQN demonstrated superior overall performance than DDQN in both recognition and classification tasks.

Based on the previous results, we can conclude that the development of hand gesture recognition models has significant potential for future applications. These include the development of more advanced models, control of hand prosthetics, creation of immersive gaming experiences, and intuitive control of consumer products. Continued research and experimentation are crucial to enhance the accuracy and effectiveness of these systems.

4.2 Recommendations

Considering this study’s conclusions, the following suggestions are proposed for future investigations in the realm of real-time hand gesture recognition using reinforcement learning techniques:

- The outcomes demonstrated that fine-tuning resulted in enhanced recognition and classification precision. Nevertheless, there is potential for further exploration of other hyperparameters and their influence on model performance. Undertaking a

more extensive hyperparameter search could yield superior models and deeper insights into the behavior of reinforcement learning-based methods.

- Additionally, an unexplored area worth investigating is the refinement of the reward function. Given its pivotal role in determining the model's final performance through reinforcement learning methods, researching innovative reward functions and their implications within this specific application could significantly improve our comprehension and yield better results.

5. REFERENCES

[1] S. M. y. T. Acharya, «Gesture Recognition: A Survey,» *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews),* vol. 37, nº 3, pp. 311-324, 2007.

[2] J. S. y. R. R. Murph, «Hand gesture recognition with depth images: A review,» *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication,* pp. 411-417, 2012.

[3] M. E. B. e. al., «Real-Time Hand Gesture Recognition Using the Myo Armband and Muscle Activity Detection,» *IEEE Xplore,* pp. 1-6, 2017.

[4] F. Ferri, *Desarrollo de un modelo de reconocimiento de gestos de la mano utilizando señales EMG y Deep Learning,* Quito: Escuela Politénica Nacional, 2021.

[5] A. G. y. D. S. H. van Hasselt, «Deep Reinforcement Learning with Double Q-Learning,» *AAAI,* vol. 30, nº 1, 2016.

[6] M. E. Benalcazar, L. Barona, L. Valdivieso, X. Aguas y J. Zea, «EMG-EPN-612 Dataset,» Zenodo, 6 November 2020. [En línea]. Available: https://zenodo.org/records/4421500. [Último acceso: 28 January 2024].

[7] A. Matos, T. Adão, L. Magalhães y E. Peres, «Procedia Computer Science,» *A Myographic-based HCI Solution Proposal for Upper Limb Amputees,* vol. 100, pp. 2-13, 2016.

[8] P. Visconti, F. Gaetani, G. A. Zappatore y P. Primiceri, «Technical Features and Functionalities of Myo Armband: An Overview on Related Literature and Advanced Applications of Myoelectric Armbands Mainly Focused on Arm Prostheses,» *International Journal on Smart Sensing and Intelligent,* vol. 11, nº 1, pp. 1-25, 2018.

[9] M. R. Andagama Lasso, «Implementación de un algoritmo para el reconocimiento de gestos del brazo humano en tiempo real utilizando la Imu del dispositivo Myo Armband,» Escuela Politécnica Nacional, Quito, 2022.

[10] E. A. Chung Liu, «Modelo de reconocimiento en tiempo real de gestos de la mano utilizando técnicas de deep learning y señales electromiográficas,» Escuela Politécnica Nacional, Quito, 2018.

[11] L. D. Unapanta Benavides, «Reconocimiento de gestos de la mano en tiempo real basado en señales electromiográficas utilizando Myo Armband con wavelets y máquinas de vectores de soporte,» Escuela Politécnica Nacional, Quito, 2019.

[12] L. P. Kaelbling, M. L. Littman y A. W. Moore, «Reinforcement Learning: A Survey,» *Journal of Artificial Intelligence,* vol. 4, pp. 237-285, 1996.

[13] R. S. Sutton y A. G. Barto, «The Reinforcement Learning Problem,» de *Reinforcement Learning: An Introduction*, London, MIT, 1988, pp. 7-9.

[14] B. Jang, M. Kim, G. Harerimana y J. Wook, «Q-Learning Algorithms: A Comprehensive Classification and Applications,» *IEEE Access,* vol. 7, pp. 133653-133667, 2019.

[15] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver y D. Wierstra, «Continuous control with deep reinforcement learning,» 2019.

[16] J. Xiong, Q. Wang, Z. Yang, P. Sun, L. Han, Y. Zheng, H. Fu, T. Zhang, J. Liu y H. Liu, «Parametrized Deep Q-Networks Learning: Reinforcement Learning with Discrete-Continuous Hybrid Action Space,» 2018.

[17] J. Zhai, Q. Liu, Z. Zhang, S. Zhong, H. Zhu, P. Zhang y C. Sun, «Deep Q-Learning with Prioritized Sampling,» *ICONIP 2016: Neural Information Processing,* vol. 9947, pp. 13-22, 2016.

[18] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, J. Yi, W. Zhao, X. Wang, Z. Liu, H.-T. Zheng, J. Chen, Y. Liu, J. Tang y J. Li, «Parameter-efficient fine-tuning of large-scale pre-trained language models,» *Nature Machine Intelligence,* vol. 5, pp. 220-235, 2023.

[19] J. Howard y S. Ruder, «Universal Language Model Fine-tuning for Text Classification,» 2018.

[20] J. P. Sahoo, A. J. Prakash, P. Pławiak y Saunak, «Real-Time Hand Gesture Recognition Using Fine-Tuned Convolutional Neural Network,» *Sensors,* vol. 22, nº 3, p. 706, 2022.

[21] J. P. Vásconez, L. I. Barona López, Á. L. Valdivieso Caraguay y M. E. Benalcázar, «Hand Gesture Recognition Using EMG-IMU Signals and Deep Q-Networks,» *Sensors,* vol. 22, nº 24, p. 9613, 2022.

[22] J. P. Vásconez, L. I. B. López, Á. L. V. Caraguay, P. J. Cruz, R. Álvarez y M. E. Benalcázar, «A Hand Gesture Recognition System Using EMG and Reinforcement Learning: A Q-Learning Approach,» *Artificial Neural Networks and Machine Learning – ICANN 2021,* vol. 12894, nº ICANN 2021, 2021.

[23] J. P. Vásconez, L. I. B. López, Á. L. V. Caraguay, P. J. Cruz y M. E. Benalcázar, «A comparison of EMG-based hand gesture recognition systems based on supervised and reinforcement learning,» *Engineering Applications of Artificial Intelligence,* vol. 123, p. 106327, 2023.

[24] P. J. Cruz, J. P. Vásconez, R. Romero, A. Chico, M. E. Benalcázar, R. Álvarez, L. I. Barona López y Á. L. Valdivieso Caraguay, «A Deep Q-Network based hand gesture recognition system for control of robotic platforms,» *Scientific,* vol. 13, nº 1, 2023.

[25] J. Eschmann y B. A. H. K. P. P. S. P. J. Belousov, «Reward Function Design in Reinforcement Learning,» *Studies in Computational Intelligence ,* vol. 883, pp. 25-33, 2021.