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 CON FUNCIÓN EN TIEMPO REAL MEDIANTE EL USO DE DOUBLE DEEP Q-NETWORK (DDQN)

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN CIENCIAS DE LA COMPUTACIÓN

CRISTIAN GABRIEL BASTIDAS VERDUGA

cristian.bastidas@epn.edu.ec

DIRECTOR: ÁNGEL LEONARDO VALDIVIESO CARAGUAY

angel.valdivieso@epn.edu.ec

DMQ, febrero de 2024

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

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

DESIGN OF A REAL-TIME HAND GESTURE RECOGNITION MODEL FOR FIVE GESTURES USING DOUBLE DEEP Q-NETWORK (DDQN)

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

CRISTIAN GABRIEL BASTIDAS VERDUGA

cristian.bastidas@epn.edu.ec

DIRECTOR: ÁNGEL LEONARDO VALDIVIESO CARAGUAY

angel.valdivieso@epn.edu.ec

DMQ, February 2024

CERTIFICATIONS

I, Cristian Gabriel Bastidas Verduga hereby declare that the curricular integration work herein described as my original work; that it has not been previously presented for any degree or professional certification; and that I have consulted all the bibliographic references included in the document.

____

CRISTIAN GABRIEL BASTIDAS VERDUGA

I certify that the curricular integration work herein was developed by Cristian Gabriel Bastidas Verduga, under my supervision.

____

ÁNGEL LEONARDO VALDIVIESO CARAGUAY

DIRECTOR

COPYRIGHT STATEMENT

CRISTIAN GABRIEL BASTIDAS VERDUGA

ÁNGEL LEONARDO VALDIVIESO CARAGUAY

DEDICATION

A sincere feeling of gratitude to my devoted parents for their unconditional support and sacrifices throughout my academic journey. Their commitment has served as a solid foundation for my aspirations and objectives.

To my brother Esteban Sánchez, his presence has offered a source of strength, and his motivating words have served as my foundation when the journey seemed too arduous.

I also dedicate this work to my friend and brother Ernesto Yaselga for his assistance and uplifting words. His wisdom and genuine friendship have been an invaluable guide.

To every individual who has contributed to my journey, I wish to express my appreciation for your constant and unselfish encouragement.

ACKNOWLEDGMENT

I express my deepest gratitude to the esteemed members of the Alan Turing Research Laboratory at the *Escuela Politécnica Nacional* for their indispensable technical assistance and guidance throughout the research process. Their expertise and dedication have been pivotal in ensuring the success of this work.

Additionally, I acknowledge Juan Pablo Vásconez for his generous provision of resources that proved essential for conducting this research. His support has been instrumental in enabling this study to come to an outcome.

I offer special thanks to Jonathan Zea, whose insights and ideas have significantly influenced the development of this research. His collaboration and knowledge-sharing have deepened my understanding of the subject matter, contributing to the overall of this work.

Lastly, I express my most sincere gratitude to my esteemed director, Ángel Valdivieso, for his expert guidance, boundless patience, and commitment throughout this process. His dedication has been crucial in providing me with the necessary support at every stage.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

RESUMEN

El reconocimiento de gestos de la mano (HGR) mediante señales de electromiografía (EMG) es fundamental en la interacción humano-computador, permitiendo una comunicación natural entre humanos y tecnología en diversas aplicaciones como prótesis, robótica y dispositivos de rehabilitación. Aunque las técnicas de aprendizaje supervisado, como las Redes Neuronales Convolucionales (CNN), han sido ampliamente exploradas en sistemas de HGR y han alcanzado altos niveles de precisión, el aprendizaje por refuerzo (RL) se destaca por su capacidad para aprender de la interacción y resolver problemas de decisión secuenciales. Algoritmos como Q-Learning y Deep Q-Network (DQN) han mostrado resultados prometedores en HGR y EMG. En esta investigación, se propone comparar el rendimiento de las técnicas de aprendizaje por refuerzo DQN y DDQN en HGR utilizando el conjunto de datos EMG-EPN-612, evaluando la precisión de los modelos tanto en reconocimiento como en clasificación. Las contribuciones principales incluyen el diseño de una función de recompensa para este contexto y la utilización de una red neuronal convolucional preexistente para un ajuste fino. Los resultados muestran que los modelos con ajuste fino, junto con DQN y DDQN, presentaron valores similares en las métricas propuestas. Se observó una mejora del 3.61% en la precisión de reconocimiento para el modelo DQN y una mejora del 3.45% para el modelo DDQN en comparación con el modelo base. Sin embargo, no se encontraron mejoras significativas en la precisión de clasificación. En conclusión, se encontró que DQN generalmente tuvo un mejor rendimiento que DDQN en este escenario.

PALABRAS CLAVE: Aprendizaje por Refuerzo, Deep Q-Network, Double Deep Q-Network, Reconocimiento de Gestos de la Mano, Electromiografía, Función de Recompensa.

ABSTRACT

Hand Gesture Recognition (HGR) using Electromyography (EMG) signals is a crucial component of Human-Computer Interaction, enabling natural communication between humans and technology in various applications such as prosthetics, robotics, and rehabilitation devices. While supervised learning techniques like Convolutional Neural Networks (CNNs) have been extensively explored in HGR systems and have achieved high levels of accuracy, reinforcement learning (RL) stands out for its ability to learn from interaction and solve sequential decision problems. Algorithms such as Q-Learning and Deep Q-Network (DQN) have shown promising results in HGR and EMG. This research aims to compare the performance of reinforcement learning techniques DQN and DDQN in HGR using the EMG-EPN-612 dataset, evaluating the accuracy of the models in both recognition and classification. Key contributions include the design of a reward function for this context and the use of a pre-existing convolutional neural network for fine-tuning. The results show that models with fine-tuning, alongside DQN and DDQN, exhibited similar values in the proposed metrics. A 3.61% improvement in recognition accuracy was observed for the DQN model and a 3.45% improvement for the DDQN model compared to the baseline model. However, no significant improvements were found in classification accuracy. In conclusion, it was found that DQN outperformed DDQN in this scenario.

KEYWORDS: Reinforcement Learning, Deep Q-Network, Double Deep Q-Network, Hand Gesture Recognition, Electromyography, Reward Function.

1 INTRODUCTION

Hand Gesture Recognition (HGR) enables natural interaction between humans and technology [1]. The process entails recognizing both the temporal aspect of a hand movement and its associated gesture class [2]. Myoelectric control, which relies on interpreting electrical signals produced by muscles using Electromyography (EMG), has gained significant traction in recent years. EMG recognition systems have demonstrated remarkable applicability across human-computer interaction. This preference arises from their ability to accurately capture subtle movements, which results in reliable data [3].

In this study, we plan to conduct a comparison of two widely adopted reinforcement learning techniques: the Deep Q-Network (DQN) and Double Deep Q-Network (DDQN). Through an evaluation, we will assess their performance in both recognition accuracy and classification accuracy. Furthermore, these methods will be employed using two distinct approaches, with models being trained from scratch and fine-tuned from a pre-existing model. Additionally, post-processing will be implemented on these models to observe its impact on the overall performance. Consequently, this work aims to assess a total of eight different models:

- DQN model trained from scratch.
- DDQN model trained from scratch.
- DQN model trained from scratch with post-processing.
- DDQN model trained from scratch with post-processing.
- DQN fine-tuned model.
- DDQN fine-tuned model.
- DQN fine-tuned model with post-processing.
- DDQN fine-tuned model with post-processing.

Given that, the complete research encompasses two components, with the first focusing on Deep Q-Network (DQN) and the second on Double Deep Q-Network (DDQN). The work herein will take the second component. Consequently, all the models mentioned in relation to DDQN (including those trained from scratch and fine-tuned with or without post-processing) will form an integral part of the development process within this work.

1.1 General objective

Develop a real-time hand gesture recognition model using the Double Deep Q-Network (DDQN) reinforcement learning technique based on the EMG-EPN-612 dataset.

1.2 Specific objectives

1. Review the state of the art of hand gesture recognition models using reinforcement learning, specifically DDQN-based models.
2. Design and assess a hand gesture recognition model for five different gestures using EMG signals from the EMG-EPN-612 dataset through the DDQN learning technique.
3. Compare the results of the DDQN model with the results of the Deep Q-Network (DQN) model in terms of gesture classification and recognition accuracies.

1.3 Scope

This work consists of four phases detailed below:

1. Agent-Environment implementation phase: In this phase an interface is created for interaction between the agent and its environment, which is represented by the EMG signal spectrogram. This interface allows the agent to perform actions and receive information about the state of the environment.
2. DDQN configuration phase: The reinforcement learning (RL) reward function will be created based on the aspects of the problem and the characteristics of the involved environment. Then, an optimal set of hyperparameters will be determined for satisfactory results in solving the issue.
3. Training with DDQN: The RL model training will be conducted using a convolutional neural network as a target network, focusing especially on optimizing its hyperparameters.
4. Results evaluation phase: In the final phase, the model's performance will be exhaustively evaluated by analyzing its performance, accuracy, and response time. This evaluation will be conducted using a validation dataset, allowing us to gain a solid understanding of the efficiency of the model.

1.4 General background

Hand Gesture Recognition (HGR) is a key component of Human-Computer Interaction (HCI), enabling natural interaction between humans and technology. HGR systems identify specific hand gestures from a predefined set and the instant of their occurrence, serving various applications such as controlling upper-limbs prosthetics and robotics, human-computer interfaces including mouse control and gaming, and medical applications like data visualization and image manipulation during medical procedures. It is also suitable for rehabilitation devices, device control, and sign language recognition [1], [4], [5]. HGR can be achieved through different methods, one of which is non-invasive surface Electromyography (EMG) using devices such as the Myo Armband [6]. This method is less intrusive and more comfortable compared to invasive EMG techniques, making it a good option for everyday use in diverse applications.

Supervised learning HGR recognition systems have been broadly researched through supervised learning techniques such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN) [1] , or even Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) approaches. Those models have demonstrated that HGR systems can take advantage of the supervised learning approach reaching an accuracy of 92.93% ± 8.23% and a recognition accuracy of 91.60% ± 8.81% [2].

Reinforcement learning (RL) stands out as a technique that differs significantly from supervised learning. While supervised learning involves being explicitly told which actions lead to an outcome, RL discovers which actions maximize a reward signal by a trial-and-error approach. Although supervised learning could generalize responses in situations, alone it is not adequate for learning from interaction [7] which is particularly relevant for HGR systems as it enables learning from new data. Additionally, unlike supervised learning, RL is appropriate for solving sequential decision problems [8].

Q-learning is one of the popular reinforcement learning algorithms due to its simplicity. On the other hand, DQN is an algorithm that introduces deep neural networks to approximate a policy [9]. However, DQN evaluates and selects actions over the same target network that can result in overestimated values. To prevent this behavior, DDQN introduces a second network, one for action selection and the other for evaluation [8]. Elements of DDQN are illustrated in Figure 1 include:

- Action: This refers to the actions taken by the agent within an environment.
- Reward function: One of the crucial aspects of both DQN and DDQN algorithms. This function is responsible for determining the agent's behavior [10].
- State: Represents the current observation given by the environment. In this work, the state is defined as spectrograms of EMG signals.
- Agent: An entity that runs actions inside the environment and its goal is to optimize its performance through learning.
- Environment: The context where the agent makes decisions. In this case, EMG signals represent the environment.
- Interface: A system that facilitates the interaction between two entities (agent and environment).

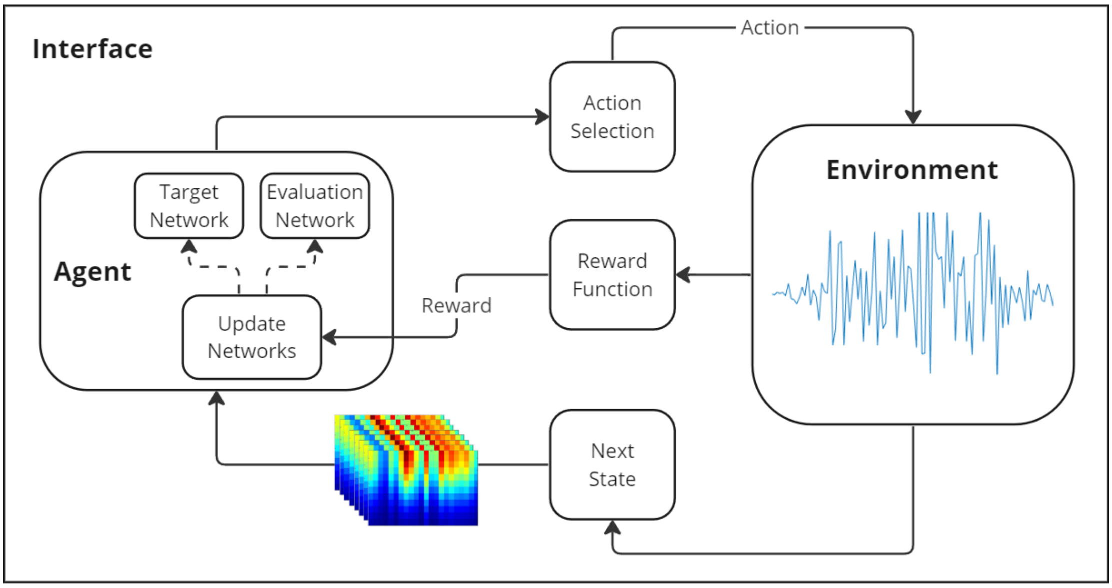

Figure 1. DDQN algorithm elements.

Previous research on EMG HGR using RL has shown promising results. For instance, [11], who employed Q-learning for EMG classification, achieving classification accuracy of up to 90.78% and recognition accuracy of 87.51%. This demonstrates the potential of Q-learning for addressing both recognition and accuracy challenges within the EMG domain. However, it is essential to note that this method requires training multiple models. They trained 306 distinct models using 100 samples for model development, 13 for validation, and 12 for testing. Additionally, they divided the training phase into two stages: the first stage involves optimizing hyperparameters with 306 user data, and the second stage utilized the remaining data [11].

In line with previous research, [12] applied DDQN for adapting HGR models to changing gesture requirements. Through enhancements in experience replay and algorithmic post-processing, the DDQN model achieved classification accuracy of 97.36% and recognition accuracy of 94.83% on the same dataset. These results surpass those obtained by other methods such as SVM, Artificial Neural Networks (ANN), and Q-learning [12]. This study highlights the potential of DDQN for enhancing EMG HGR performance in dynamic gesture recognition scenarios.

A separate study compared the performance of supervised learning and reinforcement learning (RL) EMG HGR systems using the EMG-EPN-612 dataset [13]. Supervised learning produced superior results, with classification accuracy of 90.49% and recognition accuracy of 86.83%. These findings underscore the supervised learning technique [13]. This research utilized a CNN approach incorporating parallel convolutions and max-pooling layers inspired by GoogLeNet [14].

Finally, a study proposes a user-specific HGR system using DQN and DDQN algorithms [5]. This research incorporated LSTM as an additional layer to an ANN, resulting in the DQN model without LSTM as the optimal one. This model achieved classification accuracy of up to 90.37% and recognition accuracy of 82.52% [5].

1.5 Contribution

Our work aims to expand upon this body of work by directly comparing DQN and DDQN RL techniques in HGR using the EMG-EPN-612 dataset. The primary contributions of this project are outlined below.

- We developed two HGR user-general reinforcement learning models based on DQN and DDQN algorithms. These models accept spectrograms of EMG signals as input.
- To determine which algorithm is best suited for our environment, we evaluated the DQN and DDQN models in terms of classification and recognition accuracies.
- We designed a comprehensive reward function to cover this work scenario considering suggestions from [10]. This versatile reward function can be easily customized for further enhancements and research in the future.

2 METHODOLOGY

In this section, we present a base flow designed to address the EMG-based HGR problem, as illustrated in Figure 2. This architecture encompasses an RL multi-agent environment, reward function, hyperparameter optimization, fine-tuning, and post-processing. As previously mentioned, DQN utilizes a deep neural network as its target network. In our proposed implementation, we will employ a CNN-based model from the previous research by [2] in two stages. The first stage involves initializing the CNN with randomly assigned weights. The second stage entails fine-tuning the CNN using an already trained model. Both models are founded on the same GoogLeNet CNN architecture [2], [14].

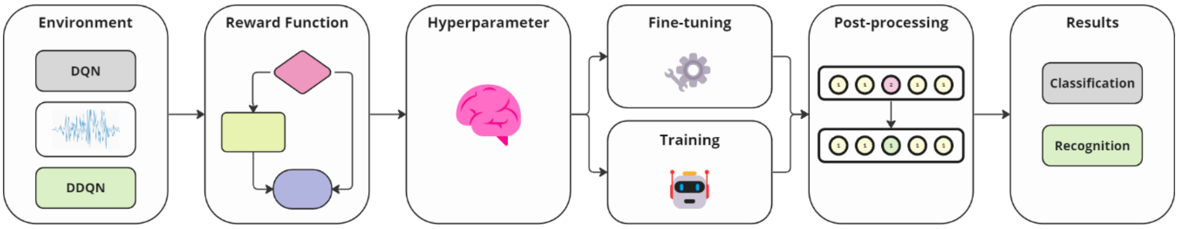

Figure 2. Hand gesture recognition proposed workflow.

Like the studies mentioned previously, we adopted the use of the same EMG-EPN-612 dataset [15] for our research, given that we utilized the identical CNN architecture. This comprehensive dataset comprises 612 users' EMG signals recorded via the Myo armband for a duration of five seconds, which translates to 1000 data points per user intended for HGR model development and benchmarking. These signals represent five distinct hand gestures: wave-in, wave-out, fist, open, and pinch. Additionally, the dataset encompasses EMG signals of users' hands remaining relaxed, denoted herein as "noGesture". The mentioned gestures are shown in Figure 3. The dataset is partitioned into two separate groups: one consisting of 306 users (150 samples per user) for training purposes due to its inclusion of ground truth data (the part of the signal where the gesture was performed), and the other remaining 306 users for testing purposes [15].

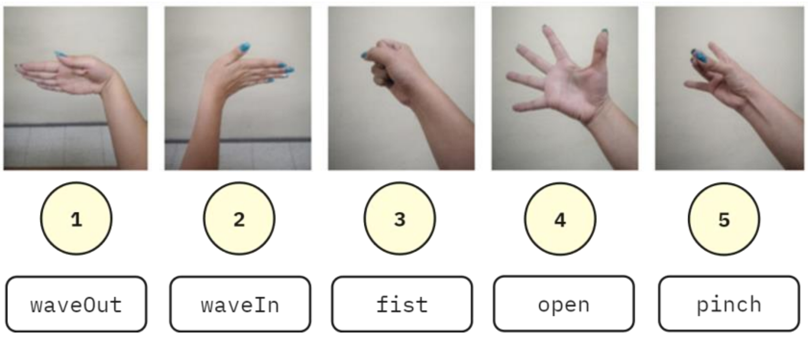

Figure 3. The five gestures to be classified and recognized.

2.1 RL multi-agent environment

To ensure a fair comparison between DQN and DDQN, it is crucial to provide both agents with an equivalent learning environment through their respective training phases. Observing their behavior from the beginning establishes a foundation for further analysis during fine-tuning and validation stages. It is important to note that, within the context of this research scenario, the environment operates under two distinct actions, states, and rewards, as described in Figure 4.

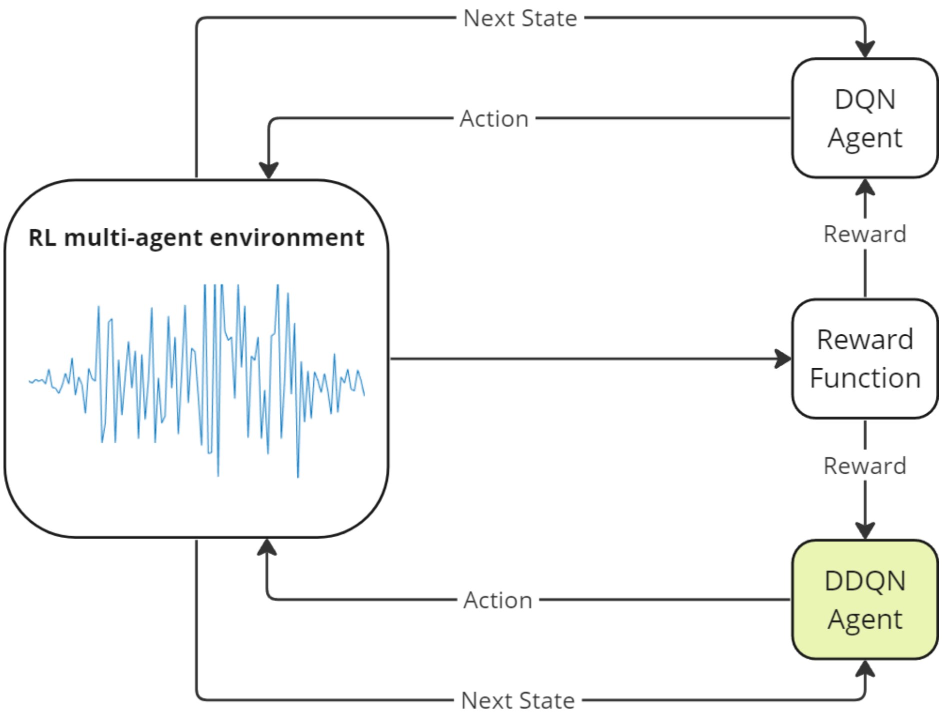

Figure 4. Proposed reinforcement learning multi-agent environment.

2.2 Reward function

Designing a practical reward function for an RL agent is a challenging problem [10]. Thus, our reward function considers the percentage of the effective EMG signal on the current window as the difficulty of the current gesture and consistent performance.

A gradual reward involves assigning proportionate rewards based on the similarity between the predicted action and the expected action. In this case, the reward function uses the percentage of the effective signal as a reward if the model makes a correct prediction; otherwise, the reward is -1. This percentage will be subtracted by 2 because complex predictions are located at the beginning and end of the effective signal where it does not have all the information. Hence, if the current window covers 100% of the effective signal, the maximum reward will be 1.

Our observations indicate that the default gesture (noGesture) is an easy prediction for the model, so the reward is lower in these cases. Instead of awarding +1 for a correct prediction, we award +0.5 when processing an entire default gesture signal or +0.25 when processing parts of the signal that do not correspond to a gesture. Thus, this part of the reward function and the gradual reward attracts the model towards correct predictions, the reward function accomplishes a dense learning signal [10]. A high-level explanation can be observed in Figure 5 flowchart.

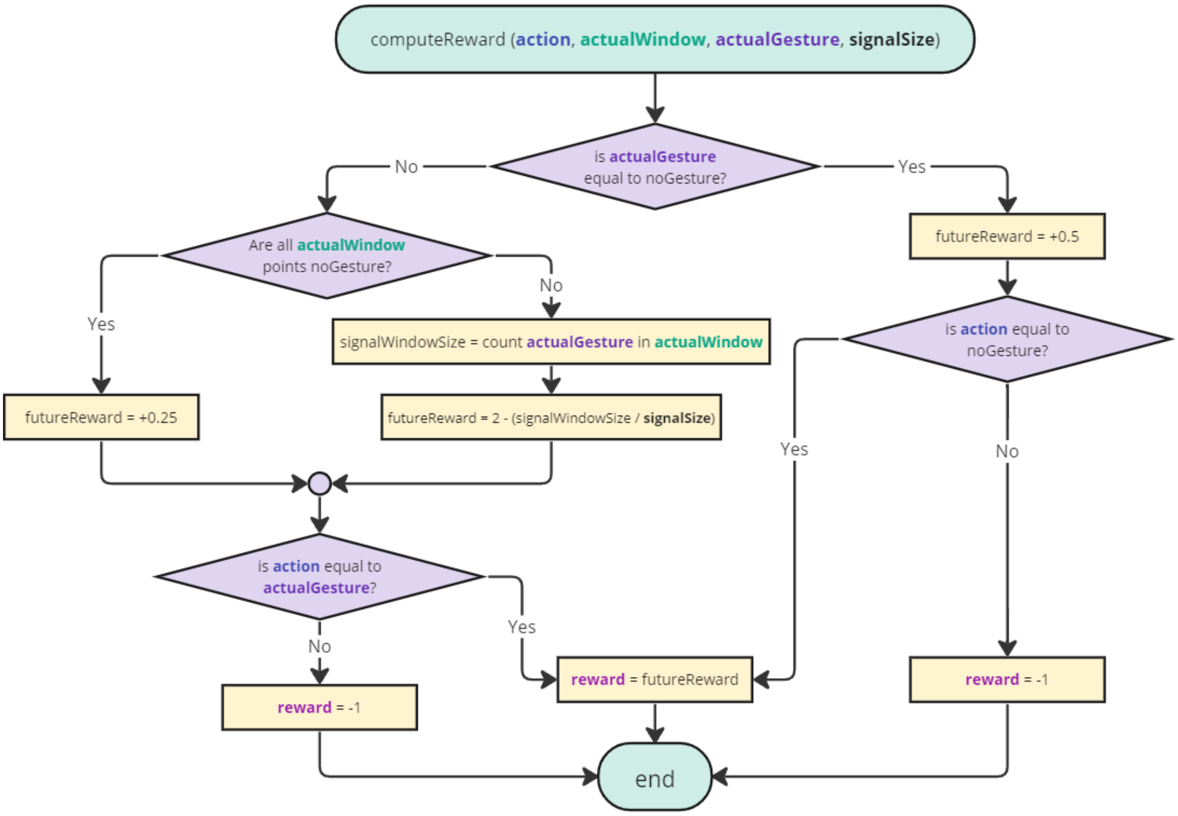

Figure 5. Proposed reward function flowchart.

Finally, if the model has been making correct decisions for consecutive windows, then it is awarded a reward. The idea is that this mechanism incentivizes the model to maintain consistent performance, as this will result in additional rewards over the EMG signal. With this method of assigning rewards, we could strengthen the recognition phase of the model. This approach is known as an intrinsic motivation signal inspired by curiosity or desire of novelty, biasing the model towards being consistent [10]. For that, consider the following aspects:

- A counter is maintained 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 computed and given to the agent.
- The counter is reset to zero upon a bad prediction and the process restarts.

Since we want to give greater rewards for more consecutive correct predictions, we consider an exponential function (Equation 1) for computing the additional reward. Where α and β defines the behavior of the reward and W is the number of consecutive correct predictions.

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

Equation 1. Proposed long-term reward function.

2.3 Hyperparameters

Following the establishment of the environment, which included the target network and reward function, we executed multiple experiments to identify the optimal hyperparameters for the training process. Since the convolutional neural network was initialized from scratch, we conducted trials with datasets containing 1 user, 75 users, and the complete training dataset consisting of 306 users.

Table 1. DDQN hyperparameters experiments from a scratch model.

Users Count	Episodes	Learn Rate	Epsilon Decay	Target Smooth Factor
1	15000	1E-02	1E-05	1E-04
1	15000	1E-03	1E-05	1E-04
1	15000	1E-05	1E-04	1E-03
1	5835	1E-03	1E-05	1E-03
1	30000	1E-06	1E-05	1E-03
75	44693	1E-02	1E-04	1E-03
75	45000	1E-06	1E-05	1E-03
75	44998	1E-02	1E-05	1E-04
306	91800	1E-05	1E-04	1E-03
306	91800	1E-06	1E-05	1E-04
306	91800	9E-06	1E-04	1E-03
306	45432	1E-06	1E-05	1E-03

Table 1 summarizes our experiments, with learning rate determining the rate at which the algorithm adjusts the network based on new learnings. Meanwhile, epsilon decay regulates the exploration vs exploitation balance by gradually decreasing random actions over time. Lastly, the target smooth factor impacts the stability of the network updates during training.

2.4 Training

During our training phase, we utilized a graphical process unit to optimize performance, specifically an NVIDIA Quadro P4000. Our training script incorporates a range of parameters designed to manage and evaluate the training process effectively. Particularly, the "*StopTrainingCriteria*" parameter is set to "*EpisodeCount*", implying that training will cease upon achieving a predetermined number of episodes, as defined by "*maxEpisodes*". The "*SaveAgentCriteria*" parameter is set to "*EpisodeFrequency*", ensuring regular saves of the agent's parameters throughout training at the specified interval, determined by "*SaveAgentValue*". Lastly, the "*Verbose*" parameter is set to "*true*", yielding detailed logs and outputs during training, thereby granting valuable insights into the dynamics of the training process.

2.5 Fine-tuning

To expedite our research, we also took advantage of the pre-existing CNN user general model from [2], which is based on GoogLeNet and has been trained over the same dataset [2]. We optimized this model by fine-tuning it through the DDQN algorithm. Fine-tuning required extensive efforts in hyperparameters optimization. Consequently, we experimented iteratively to bring the training progress to a satisfactory convergence level using the same approach of the hyperparameters section. The hyperparameters experiments for fine-tuning are resumed in Table 2.

Table 2. DDQN fine-tuning hyperparameters experiments.

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

2.6 Post-processing

A post-processing is conducted to clean EMG predicted signals. The primary objective is to identify and process continuous sequences of valid gestures while eliminating isolated instances of ignored or inconsequential actions. Ignored gestures (noGesture) are simple to recognize; thus, they receive minimal attention and require no additional adjustments during processing. However, it is important to consider the possibility that a valid gesture sequence might be followed by an ignored gesture. To account for this situation, our post-processing technique searches for continuous sequences of valid gestures within the predicted data. Once identified, all subsequent classes within that sequence are replaced with the most common class. This approach ensures that every intended gesture is accurately represented while maintaining the overall integrity and continuity of the processed EMG signal. Figure 6 illustrates instances where our proposed method encounters discrepancies between the predicted and actual EMG signals. Figure 6a depicts such scenarios when mismatches occur at the beginning (Case 1), middle (Case 2), and end (Case 3) of the signal, respectively. In contrast, Figure 6b presents the post-processed signals for each of these instances (Case 1, Case 2, and Case 3).

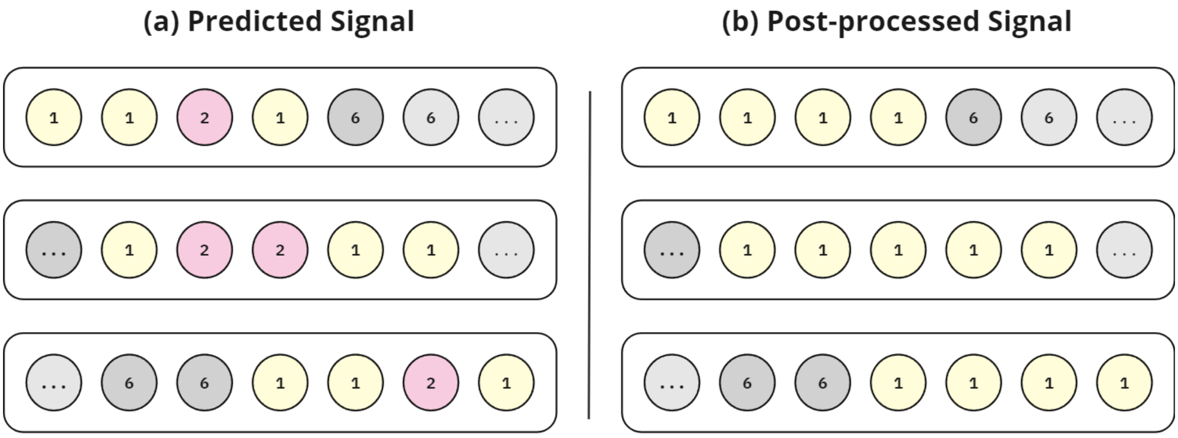

Figure 6. Post-processing stage.

a) Raw predicted data from the RL model, **b)** Post-processed data ready for an application.

By employing this post-processing method, our model can effectively recognize and classify complex patterns from raw predictions, leading to improved classification and recognition accuracy.

3 RESULTS

In this section, we present the outcomes of evaluating our proposed user-general HGR models by testing various hyperparameters and evaluating the model with and without the post-processing stage. The unaltered hyperparameters for scratch DQN and DDQN models are outlined in Table 3.

Table 3. DDQN hyperparameters experiments.

Hyperparameter	Value
LearnRate	1.00E-06
Optimizer	"adam"
GradientThresholdMethod	"l2norm"
UseDevice	"gpu"
TargetSmoothFactor	1.00E-04
MiniBatchSize	32
NumStepsToLookAhead	1
DiscountFactor	0.98
ExperienceBufferLength	50
EpsilonDecay	1.00E-05
UseDoubleDQN	TRUE
SaveExperienceBufferWithAgent	TRUE
MaxEpisodes	91800

Given the complexity of identifying optimal hyperparameters during the fine-tuning phase, Table 4 provides the most effective hyperparameters for both the fine-tuned DQN and DDQN models.

Table 4. DDQN fine-tuning hyperparameters experiments.

Hyperparameter	Value
LearnRate	1.00E-07
Optimizer	"adam"
GradientThresholdMethod	"l2norm"
UseDevice	"gpu"
TargetSmoothFactor	1.00E-04
MiniBatchSize	32
NumStepsToLookAhead	1
DiscountFactor	0.98
ExperienceBufferLength	50
EpsilonDecay	1.00E-06
UseDoubleDQN	TRUE
SaveExperienceBufferWithAgent	TRUE
MaxEpisodes	45900

The recognition accuracy results for our proposed user-general HGR model are presented in Figure 7. The developed models do not surpass the pre-existing post-processed model's recognition accuracy (which reaches accuracies of 83.26%, 82.87%, 91.91%, and 91.98% respectively). It is noteworthy that using the pre-existing model results in a recognition 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 the model trained from scratch. However, when focusing on non-post-processed models, our proposed models exhibit superior recognition accuracy. Specifically, employing fine-tuning leads to a 3.45% enhancement over DDQN and a 3.61% improvement with the DQN model. Additionally, DDQN models trained from scratch increase recognition by 1.59%, while DQN models yield a 2.46% improvement in recognition accuracy compared to the pre-existing models. Furthermore, fine-tuned models without processing also outperform both the models trained from scratch and the post-processed models.

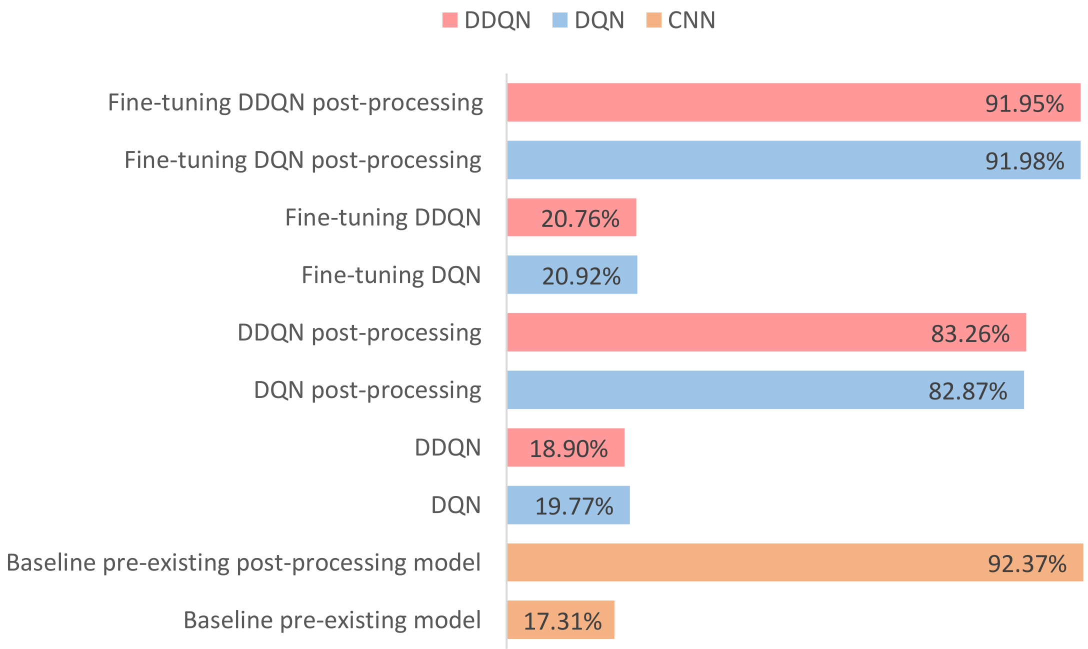

Figure 7. Recognition accuracies for all models.

The classification accuracy results for our proposed user-general HGR models are depicted in Figure 8. The findings reveal that there is no significant difference in classification accuracy between the post-processed and non-post-processed models. However, it is evident that training a model from scratch does not surpass the pre-existing model in this study, be it DDQN or DQN. Notably, the pre-existing DDQN post-processed model outperforms our proposed DDQN post-processed model by 8.19% and 8.18%, respectively. In terms of fine-tuned models, these models augment the classification accuracy of DDQN by up to 0.89%, and of DQN by 0.92%. Furthermore, when examining the fine-tuned models that are not post-processed, DDQN fine-tuned increases classification accuracy by 0.75% and DQN fine-tuned by 0.81%. Moreover, DQN demonstrates a better performance than DDQN regardless of whether they are fine-tuned or not. Specifically, the fine-tuned post-processed DQN model outperforms its counterpart by 0.03%, while the fine-tuned DQN model without post-processing improves upon it by 0.06%. Additionally, our experiments revealed that the DQN post-processed model outperforms the DDQN post-processed model by a margin of 0.11%. Furthermore, the DQN model trained from scratch also displays a slight improvement of 0.01% compared to the DDQN model.

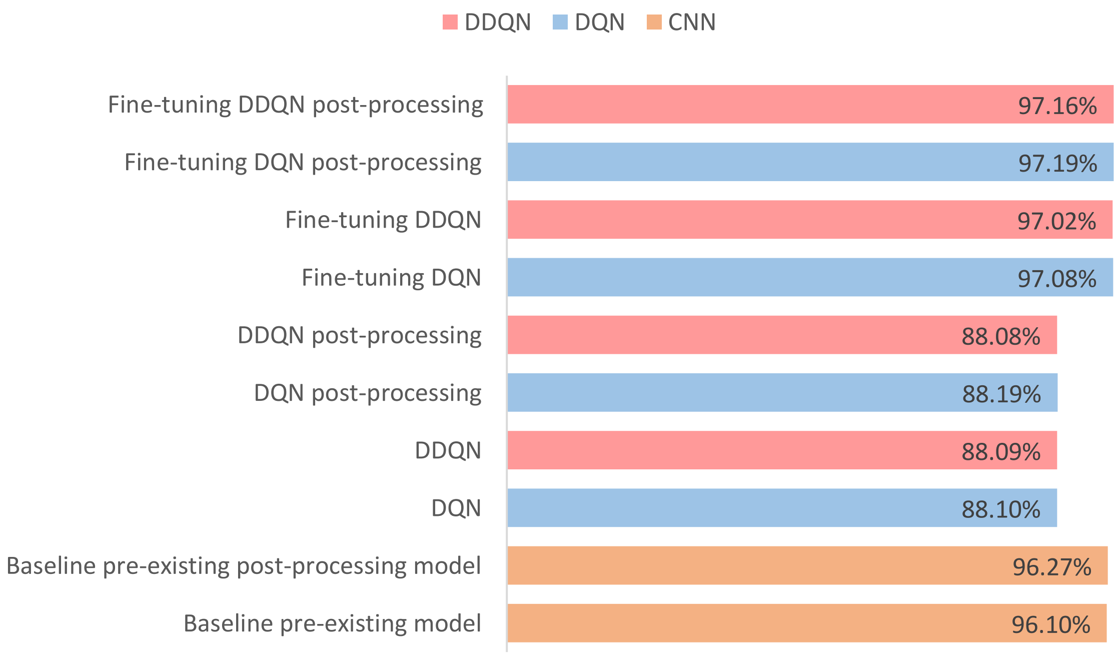

Figure 8. Classification accuracies for all models.

In Figure 9, we provide the confusion matrices for our best-performing user-general DQN and DDQN models' classification results. Figure 9a represents the confusion matrix for DQN, while Figure 9b displays the confusion matrix for DDQN. Additionally, Figure 9c illustrates the confusion matrix for DQN with post-processing, and Figure 9d displays the confusion matrix for DDQN with post-processing. Importantly, all these matrices represent models trained from scratch to provide a more accurate comparison between the methods.

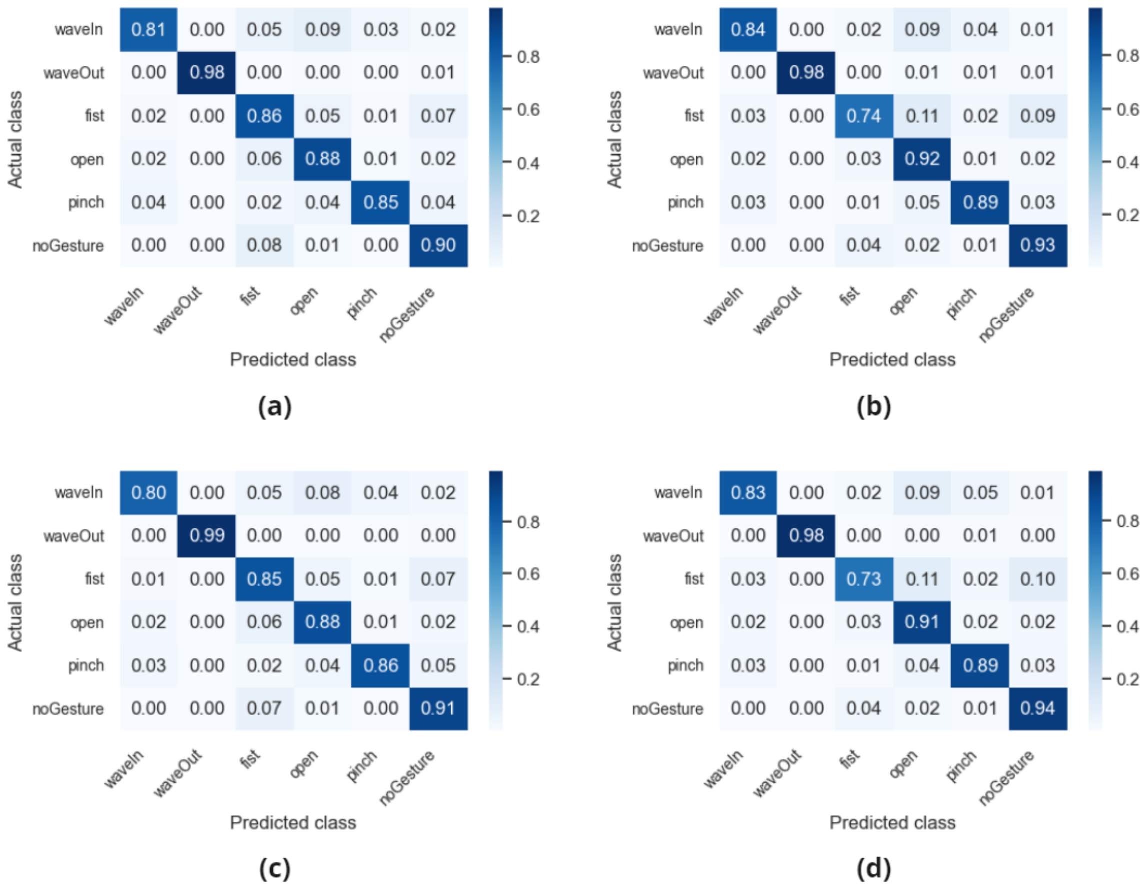

Figure 9. Confusion matrices for DQN and DDQN models trained from scratch.

a) DQN, **b)** DDQN, **c)** DQN post-processed, **d)** DDQN post-processed.

In Figure 10, we exhibit the confusion matrices for our best achieved user-general DQN and DDQN fine-tuned models' classification results. Figure 10a illustrates the confusion matrix for DQN fine-tuned, while Figure 10b demonstrates the confusion matrix for DDQN fine-tuned. Furthermore, Figure 10c displays the confusion matrix for DQN post-processed fine-tuned, and Figure 10d provides the confusion matrix for DDQN post-processed fine-tuned.

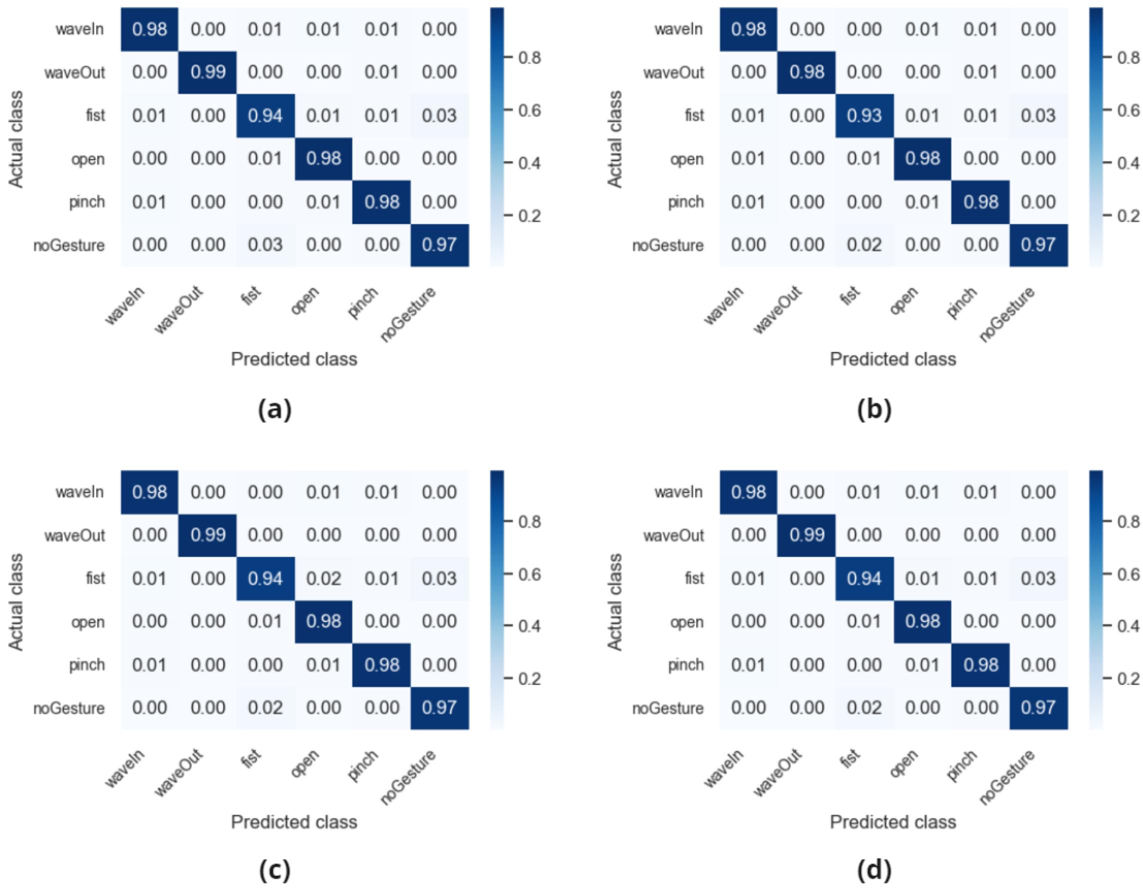

Figure 10. Confusion matrices for fine-tuned DQN and DDQN models.

a) DQN fine-tuned, **b)** DDQN fine-tuned, **c)** DQN post-processed fine-tuned, **d)** DDQN post-processed fine-tuned.

The figures in Figure 10 reveal that the fine-tuned models exhibit better performance compared to those illustrated in Figure 9. Specifically, the lowest correctly classified gesture in the fine-tuned models attains a performance of 93%, whereas the lowest correctly classified gesture in the scratch-trained models only reaches 0.73%.

The analysis of our study between DQN and DDQN reveals subtle yet significant differences in their performance regarding recognition and classification tasks. DQN outperforms DDQN, particularly in situations without fine-tuning as it is shown in Figure 11. The chart in Figure 11a represents the recognition accuracy for all user-general experiments labeled from 1 to 7. In contrast, Figure 11b illustrates the classification accuracy for these same user-general experiments numbered from 1 through 7.

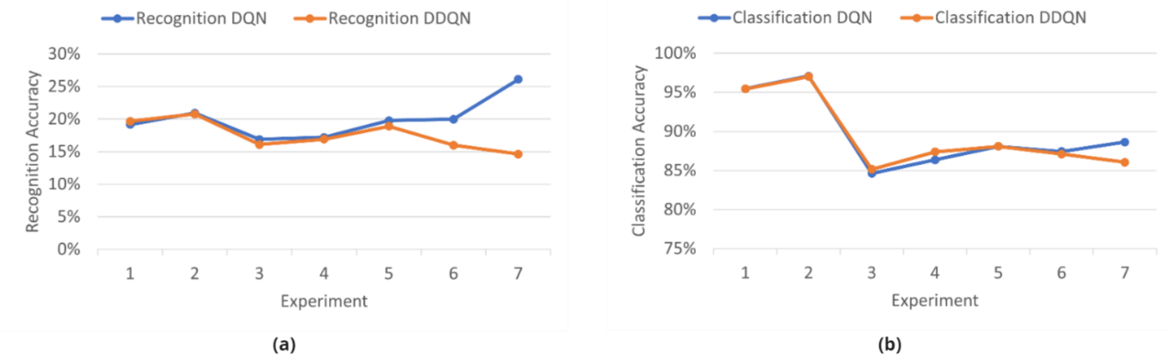

Figure 11. All user-general experiments.

a) All user-general experiments recognition accuracies, **b)** All user-general experiments classification accuracies.

As a final note, all relevant implementation details and source code for our research are available to the public through the following GitHub repository: https://github.com/laboratorioAI/2024_EMG_DQN_DDQN.

4 CONCLUSIONS

This study aimed to develop real-time hand gesture recognition models using the Double Deep Q-Network (DDQN) reinforcement learning technique based on the EMG-EPN-612 dataset. Through the fine-tuning phase, effective hyperparameters for both the fine-tuned DQN and DDQN were identified. The results showed that while our proposed models did not surpass the pre-existing post-processed model's recognition accuracy, the exhibited recognition accuracy is superior when focusing on non-post-processed models due to the reward function design. Fine-tuning led to a 3.45% enhancement in DDQN and a 3.61% improvement with DQN model's recognition accuracy compared to the pre-existing models. Additionally, 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, training a model from scratch did not surpass the pre-existing model in this study. Fine-tuned models augmented the classification accuracy of DDQN by up to 0.89%, and of DQN by 0.92%. Furthermore, fine-tuned models that are not post-processed improved DDQN's performance by 0.75% and DQN's by 0.81%. Lastly, DQN demonstrated a better overall performance than DDQN in both recognition and classification tasks.

In summary, this research aimed to develop a real-time hand gesture recognition model using the Double Deep Q-Network (DDQN) reinforcement learning technique based on the EMG-EPN-612 dataset. Through the evaluation of various hyperparameters and the assessment of model performance with and without post-processing, it was concluded that fine-tuned models exhibited superior recognition and classification accuracy compared to models trained from scratch and pre-existing models without post-processing. However, in terms of post-processed models, the pre-existing DDQN and DQN models outperformed the proposed ones. The accuracy outcomes suggest that fine-tuning can be an effective way to enhance the performance of both DQN and DDQN models. Overall, this study contributes to the advancement of real-time hand gesture recognition using reinforcement learning techniques and highlights the significance of fine-tuning for improving model performance.

4.1 Recommendations

Based on the findings of this study, the following recommendations can be made for future research in the field of real-time hand gesture recognition using reinforcement learning techniques:

- The results showed that fine-tuning led to improved recognition and classification accuracy. However, there is still room for exploring other hyperparameters and their impact on model performance. Conducting a more hyperparameter search can lead to even better models and insights into the behavior of reinforcement learning-based methods.
- Furthermore, an area ripe for exploration is refining the reward function. Given its crucial role in shaping the model's ultimate performance through reinforcement learning methods, investigating novel reward functions and their repercussions within this specific application could significantly enhance our understanding and achieve superior results.

5 BIBLIOGRAPHIC REFERENCES

[1] M. E. Benalcázar *et al.*, "Real-time hand gesture recognition using the Myo armband and muscle activity detection," in *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Oct. 2017, pp. 1–6. doi: 10.1109/ETCM.2017.8247458.

[2] F. M. Ferri Ripalda, "Desarrollo de un modelo de reconocimiento de gestos de la mano utilizando señales EMG y Deep Learning.," bachelorThesis, Quito, 2021, 2021. Accessed: Jan. 31, 2024. [Online]. Available: http://bibdigital.epn.edu.ec/handle/15000/21762

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

[4] A. Jaramillo-Yánez, M. E. Benalcázar, and E. Mena-Maldonado, "Real-Time Hand Gesture Recognition Using Surface Electromyography and Machine Learning: A Systematic Literature Review," *Sensors*, vol. 20, no. 9, Art. no. 9, Jan. 2020, doi: 10.3390/s20092467.

[5] Á. L. Valdivieso Caraguay, J. P. Vásconez, L. I. Barona López, and M. E. Benalcázar, "Recognition of Hand Gestures Based on EMG Signals with Deep and Double-Deep Q-Networks," *Sensors*, vol. 23, no. 8, Art. no. 8, Jan. 2023, doi: 10.3390/s23083905.

[6] E. A. Chung and M. E. Benalcázar, "Real-Time Hand Gesture Recognition Model Using Deep Learning Techniques and EMG Signals," in *2019 27th European Signal Processing Conference (EUSIPCO)*, Sep. 2019, pp. 1–5. doi: 10.23919/EUSIPCO.2019.8903136.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[8] H. van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-learning." arXiv, Dec. 08, 2015. doi: 10.48550/arXiv.1509.06461.

[9] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning." arXiv, Dec. 19, 2013. doi: 10.48550/arXiv.1312.5602.

[10] J. Eschmann, "Reward Function Design in Reinforcement Learning," in *Reinforcement Learning Algorithms: Analysis and Applications*, B. Belousov, H. Abdulsamad, P. Klink, S. Parisi, and J. Peters, Eds., in Studies in Computational Intelligence. , Cham: Springer International Publishing, 2021, pp. 25–33. doi: 10.1007/978-3-030-41188-6_3.

[11] J. P. Vásconez, L. I. B. López, Á. L. V. Caraguay, P. J. Cruz, R. Álvarez, and M. E. Benalcázar, "A Hand Gesture Recognition System Using EMG and Reinforcement Learning: A Q-Learning Approach," in *Artificial Neural Networks and Machine Learning – ICANN 2021*, I. Farkaš, P. Masulli, S. Otte, and S. Wermter, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 580–591. doi: 10.1007/978-3-030-86380-7_47.

[12] D. Díaz, M. E. Benalcázar, L. Barona, and Á. L. Valdivieso, "Development of a Hand Gesture Recognition Model Capable of Online Readjustment Using EMGs and Double

Deep-Q Networks," in *Information Technology and Systems*, Á. Rocha, C. Ferrás, and W. Ibarra, Eds., in Lecture Notes in Networks and Systems. Cham: Springer International Publishing, 2023, pp. 361–371. doi: 10.1007/978-3-031-33258-6_34.

[13] J. P. Vásconez, L. I. Barona López, Á. L. Valdivieso Caraguay, and M. E. Benalcázar, "A comparison of EMG-based hand gesture recognition systems based on supervised and reinforcement learning," *Eng. Appl. Artif. Intell.*, vol. 123, p. 106327, Aug. 2023, doi: 10.1016/j.engappai.2023.106327.

[14] C. Szegedy *et al.*, "Going Deeper with Convolutions." arXiv, Sep. 16, 2014. doi: 10.48550/arXiv.1409.4842.

[15] M. E. Benalcazar, L. Barona, L. Valdivieso, X. Aguas, and J. Zea, "EMG-EPN-612 Dataset." Zenodo, Nov. 06, 2020. doi: 10.5281/zenodo.4421500.