On Automated Object Grasping for Intelligent Prosthetic Hands Using Machine Learning

Jethro Odeyemi , Akinola Ogbeyemi , Wenjun Zhang * , Kelvin Wong

Posted Date: 15 December 2023

doi: 10.20944/preprints202312.1134.v1

Keywords: Computer vision; electromyography; hand gestures; machine learning; prosthetics

Article

On Automated Object Grasping for Intelligent Prosthetic Hands Using Machine Learning

Jethro Odeyemi, Akinola Ogbeyemi, Kelvin Wong and Wenjun Zhang *

Division of Biomedical Engineering, Advanced Engineering Design Laboratory, University of Saskatchewan, Saskatoon, Canada; rye164@usask.ca (J.O.); akinola.ogbeyemi@usask.ca (A.O.); kelvin.wong@usask.ca (K.W.)
* Correspondence: chris.zhang@usask.ca

Abstract: This paper explores the application of machine learning techniques for automated object grasping. Current electronic prosthetics often require extensive training for users to gain fine motor control over the prosthetic fingers, hindering their usability and acceptance. To address this challenge, this paper proposes an automated method that leverages computer vision-based techniques and machine learning algorithms. In this study, three reinforcement learning algorithms, namely Soft Actor-Critic (SAC), Deep Q-Network (DQN), and Proximal Policy Optimization (PPO), are employed to train agents for automated grasping tasks. The results indicate that the SAC algorithm achieves the highest success rate of 99% among the three algorithms at just under 200,000 timesteps. This research also shows that object's physical characteristics can affect the agent's ability to learn an optimal policy. Moreover, the findings highlight the potential of the SAC algorithm in developing intelligent prosthetic hands with automatic object gripping capabilities.

Keywords: computer vision; electromyography; hand gestures; machine learning; prosthetics

1. Introduction

Prosthetic organs, e.g., hands [1,2], are a robotic system, which has the power generator, actuator, body, sensor, and controller [3,4]. One of the important challenges with prosthetic organs, e.g., hand, is to control the prosthetic hands from the patient brain. Specifically, given a task to be performed with the prosthetic hand, how the human intention or desire to accomplish the task is mapped to the actuator on the prosthetic hand is a challenge. A popular approach is to make use of Electromyography (EMG) signals to represent the patient intent to perform a task. Therefore, EMG-controlled prostheses are popular today, which incorporate embedded motors that move the fingers based on recorded electrical activities during muscle contraction that comes from the patient brain [5]. These prostheses allow for multiple grip patterns and user-friendly interaction [6]. However, a major challenge with EMG prosthetics is that it takes users many months of continuous training to achieve full mastery over EMG-controlled prosthetics [7]. The training process takes place through repetitive exercises where the user learns to consciously control their muscle contraction. In practice, because of this strenuous training process, many users abandon the prosthesis before mastery is achieved. Another challenge with EMG-controlled prosthetic hand is an inherent difficulty to represent high resolution brain signals for task manipulation; often in practice the failure in grasping an object often happens, either the object being slipped out of the prosthetic hand or finger or the object being damaged due to too high force from the prosthetic hand [8]. Several methods to enhance the EMG representation have been proposed, including Electroencephalography (EEG) [9], eye tracking, transfer learning [10,11], and facial recognition. EMG along with its enhancing methods is limited because the desired resolution with the patient Brain signal (e.g., to grasp an object) increases far faster than the resolution of the representation of patient intent in brain, which can be achieved with the current technology such as EMG, EEG, and so on.

The other idea to overcome the limit is to make the prosthetic hand of highly autonomy, meaning that they can automatically perform the task of gripping an object. Hao et al. (2021) designed a low-cost soft prosthetic hand with embedded actuation and sensors, allowing for initial contact detection during gripping to prevent damage to objects [12]. However, the inclusion of extra sensors increased

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 December 2023 doi:10.20944/preprints202312.1134.v1

the weight of the prosthetic hand and posed the risk of sensor degradation. Castro et al. (2022) presented a prototype incorporating EMG-based control and computer vision for seamless control of a prosthetic hand with different grip patterns [13]. However, the paper did not address the control of grip pressure, which is crucial when handling delicate objects. Czimmermann et al. (2020) conducted a comprehensive review of defect detection technologies, emphasizing the resource requirements for effective training of neural networks [14]. They also highlighted the challenges of parallelization when dealing with large datasets. Abbasi et al. (2019) used unsupervised learning techniques to analyze patterns in different grasp types, but their approach was deemed expensive and impractical [15].

The motivation of the study presented in this paper was to advance the technology along the idea -- i.e., improving the autonomy of the prosthetic hand. The objective of this paper is to a train an agent (i.e., prosthetic hand) using three different algorithms—Soft Actor-Critic (SAC) [16], Proximal Policy Optimization (PPO) [17], and Deep Q Networks (DQN)—to enable the prosthetic hand to autonomously grasp objects. To achieve the objective, the impact of each algorithm on the effectiveness of the gripping action will be evaluated [18,19] and an optimal model for a variety of objects will be tested, specifically investigating how the physical properties of the objects may significantly influence the capability of the prosthetic hand to grip them effectively.

This paper presents significant contributions to the field of Myo prosthetic hand development by leveraging computer vision and machine learning technologies. Also, this paper will provide an understanding of the impact of object properties on grasping success. These contributions are important in improving the development of more efficient and effective prosthetic grasping systems with low cost and high usability.

2. Experiment

This section covers the experimental setup in using the reinforcement learning [20] to train the prosthetic arm gripper to be able to grip an object in minimum time and with optimal force.

2.1. Sensor Setup

The sensor in the task represents a camera mounted at the midpoint of the gripper's base which records observations captured in discrete steps. The observations captured by the sensor represent the state space in the Markov decision process. In an approach like (Baris, 2020), we implemented a perception pipeline using RGBD observations captured by the camera (Figure 1).

Figure 1. Example of raw RGBD data captured by the camera at a specific frame.

RGBD stands for Red, Green, Blue, and Depth and refers to a type of imaging technology that combines traditional RGB color data with depth information. The method to preprocess the RGBD data before feeding it into the convolutional neural network is like the method implemented in [21]. Since our task involves a gripper with a certain width, we need to include this information in our processing of the sensor output. To achieve this, we pad the gripper width information into a three-dimensional array that has the same shape as the sensor data (i.e., 64x64x1 or 64x64x4). This extra information is added as an extra "layer" to the sensor data, which we call a "channel".

By doing this, we can make sure that the width information of the actuator is included in the processing of the sensor data without changing the way we process it. During training, we remove

 doi:10.20944/preprints202312.1134.v1

this extra channel so that the robot can learn how to use the sensor data to control its movements, considering the width of the gripper. By including the gripper width information in this way, we can ensure that the robot can use this information to perform the task as we want it to do.

The RGBD data is then fed into a convolutional neural network to learn meaningful features and representations of the scene, which is used as input to the agent's decision-making process. The output is passed through one or more fully connected layers, which are also sometimes called dense layers. The output of the last fully connected layer can then be flattened into a one-dimensional array. The difference in the CNN architecture from the work of [22] and the one used in this study, is the introduction of dropout layers [23] which determines the fraction of the input units to drop out during the training of a neural network. The importance of this is to avoid overfitting which is a reoccurring problem in machine learning. We also employed an Exponential Linear Unit because of its success in speeding up learning compared to other LUs and in dealing with sparse data. After the tensor output has been flattened into a one-dimensional array, the unpadded gripper width information is then concatenated to the RGBD tensor array, to form the observation vector. This process is represented with the schematic shown in Figure 2.

2.2. Reward Function

The reward function implemented for this task is a custom-shaped reward function designed to incentivize the gripper's behaviour in the environment. The aim of the reward function is to incentivize the agent to grasp the object efficiently and to lift it to the desired height quickly, while penalizing the agent for taking too long to complete the task. The reward function consists of several parameters that influence the reward value and can be summarized with Table 1. Whether an object has been grasped is detected by checking the gripper width after the robot runs a close gripper event.

2.3. Training Process

The agent was trained using three existing policies, SAC, DQN and PPO. The choice of these algorithms was based on their proven effectiveness and wide usage in the field [22]. By leveraging the strengths of SAC, DQN, and PPO, it was anticipated that the agent could benefit from their advantages in terms of stability, sample efficiency, and performance.

Training begins at the reset state where the objects have the spawned on the table and the gripper also spawned midair. For each of the algorithms, we defined the reward function that provides positive feedback when the agent takes actions to move it closer to the goal, and negative feedback when it takes actions to move it further away from the goal. Next, the agent is run through a series of episodes in timesteps of 1000 in the environment. Each episode consists of the agent taking actions and receiving rewards. The agent's goal is to maximize the cumulative reward over all episodes. During each episode, the agent observes the current state of the environment, chooses an action based on its current policy, and receives a reward based on the outcome of the action. The agent then updates its policy based on the observed rewards and the current state of the environment.

The process of updating the policy varies depending on which algorithm was being tried but follows a similar update rule (1). During training, the agent explores the environment to learn the optimal policy. The training process continues until the agent has the maximum number of episodes which is predefined.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \underbrace{max}_{a} Q(S_{t+1}, a) - Q(S_t, A_t))] \quad (1)$$

$Q(S_t, A_t)$ is the estimate of the action-value function at time t, R_{t+1} is the reward received at time t, $\propto$ is the learning rate, γ is the discount factor, and $\underbrace{max}_{a} Q(S_{t+1}, a)$ is the maximum value over all actions that can be taken in the next state.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 December 2023 doi:10.20944/preprints202312.1134.v1

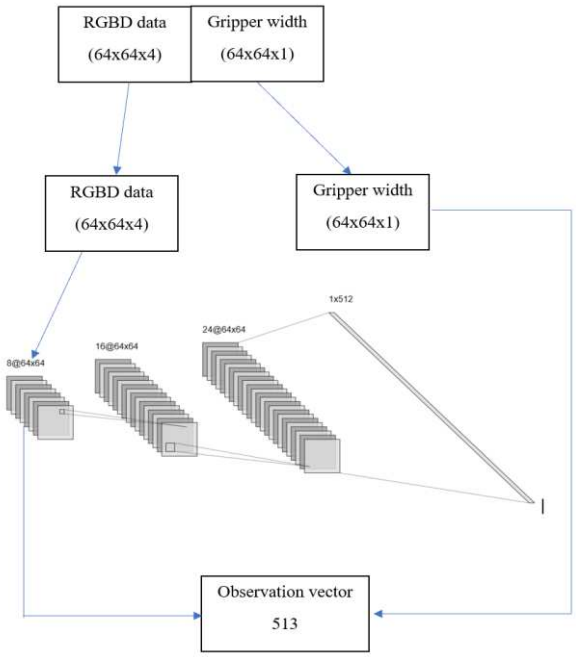

Figure 2. RGBD data into a convolutional neural network to form the observation vector for the Markov Decision Process.

Table 1. Detailed Parameters of the Custom-Shaped Reward Function Utilized to Influence and Optimize the Gripper's Behavior in the Environment.

	Non-terminal state	Terminal state
Object grasped?	$r_g - r_{tp}$	$r_t - r_{tp}$
Not grasped?	$-r_{tp}$	@timeout $-r_{tp}$

[1] r_t = terminal reward, r_g= grasping reward, r_{tp}=time penalty.

3. Results and Discussions

3.1. Algorithm-Specific Success Rate

Three different algorithms were tested during training: DQN, SAC and PPO. The training went on for 1,000,000 timesteps and the performance of each algorithm by success rate is shown in Figure 3.

From the results above, we can see that SAC outperformed the other two algorithms in terms of success rate by converging the quickest in just under 200000 timesteps. On the DQN algorithm, the agent never fully learns the optimal policy required for the grasping task. We can see it reaches a peak success rate of 80% in about 450,000 timesteps then stays at equilibrium at that point. There are several possible reasons why SAC performed better than the other two algorithms. First, SAC is an off-policy actor-critic algorithm that uses a soft value function to estimate the Q-value of actions. This soft value function provides a smoother estimate of the Q-value than the hard value function used in DQN and PPO, which can lead to better performance in the action spaces. SAC also uses a stochastic policy that is updated using a combination of maximum entropy reinforcement learning and entropy regularization. This encourages the policy to explore the state space more thoroughly, which can lead to better performance in environments with sparse rewards, such as the object picking task in our study. DQN might have performed badly because it is an on-policy algorithm that suffers from instability when learning from action spaces.

From the results, the SAC algorithm performed the best with a mean success rate of 0.990. This is significantly higher than the mean success rate of the DQN algorithm (0.602) and the PPO algorithm

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 December 2023 doi:10.20944/preprints202312.1134.v1

(0.821). The high success rate achieved by the SAC algorithm can be attributed to its inherent advantages in handling high-dimensional action spaces.

On the other hand, the DQN algorithm performed the worst, with a mean success rate of 0.602. The lower success rate achieved by the DQN algorithm can be attributed to the fact that the Q-learning architecture employed by the DQN algorithm is known to have stability issues when applied to continuous control tasks, as it tends to overestimate the Q-values and leads to suboptimal policies. This result is summarised in Table 2.

MANOVA test was further conducted on the success rates of the three algorithms: DQN, SAC, and PPO, to investigate whether there were significant differences in their performance. The results are shown in Table 3. Based on the results of the MANOVA analysis, there is strong evidence to suggest that there are significant differences between the three different algorithms in terms of their performance. The Pillai's trace statistic of 0.57982 and the highly significant p-value (< 2.2e-16) indicate that the different algorithms have a substantial impact on the success rate of the gripper. The approximate F-value of 2040.7 further supports the presence of a significant effect. These findings indicate that the choice of algorithm significantly influences the observed differences in performance of the prosthetic gripper.

Table 2. Mean Success Rate Achieved by Each Algorithm (DQN, SAC, PPO) in the Experiment.

Summary	Result
Mean success rate for DQN	0.6021689086910577
Mean success rate for SAC	0.9903811107807406
Mean success rate for PPO	0.821488216618748

Table 3. MANOVA Analysis Results: Assessing Significant Performance Differences Among DQN, SAC, and PPO Algorithms.

	DF	Pillai	F-Value	Den DF	Pr(>F)
Algorithms	2	0.57982	2040.7	19994	< 2.2e-16 ***
Residuals	9997				

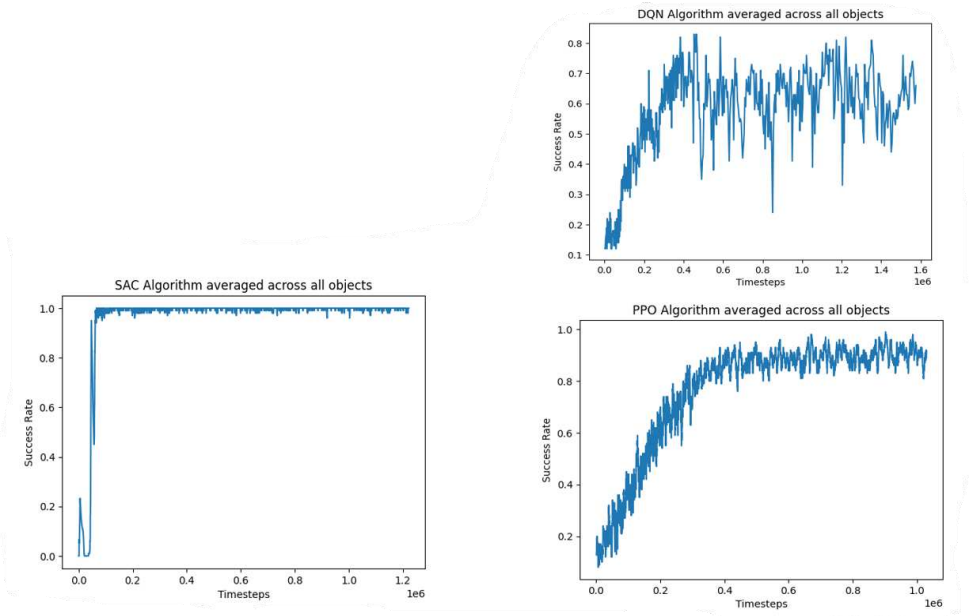

Figure 3. Comparative performance of SAC, DQN, and PPO algorithms: success rate by timesteps averaged across all objects. (a) SAC algorithm. (b) DQN algorithm. (c) PPO algorithm.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 December 2023 doi:10.20944/preprints202312.1134.v1

3.2. Comparative Analysis of SAC, PPO and DQN for Object Grasping – Hyperparameter Exploration

Increasing the batch size to 128 had a positive effect on the convergence speed of the SAC algorithm, enabling it to converge quickly after just 100,000 time-steps (Figure 4). This performs much better than the SAC with RGBD presented in (Baris, 2020).

With a larger batch size, the SAC algorithm benefits from an increased sample efficiency. More data is available for each update step, allowing for better estimation of the policy gradient and reducing the impact of noise or outlier experiences. This increased data diversity enhances the stability of the learning process and facilitates faster convergence towards an optimal policy. Increasing the batch size increases the amount of information available for each update, and this significantly increases the agent's training time.

The performance of the DQN algorithm drops at both the batch size of 32 and 128, compared to the initial analysis at batch size 64 (Figure 3). We can infer that at batch size 64, there is an optimal balance between sample efficiency and computational overhead. Deviations from this batch size, either lower or higher, lead to a diminished performance due to limitations in data availability or computational inefficiencies.

We observe a similar pattern, as shown in Figure 5, when varying the number of hidden layers for the SAC and PPO algorithms. The SAC algorithms benefit from an increase in the number of hidden layers while the PPO algorithm performs poorly in both cases at 500000 timesteps and would possibly take a longer time to converge. Based on these results, we can make an inferred comparison of these three algorithms, as shown in Table 4.

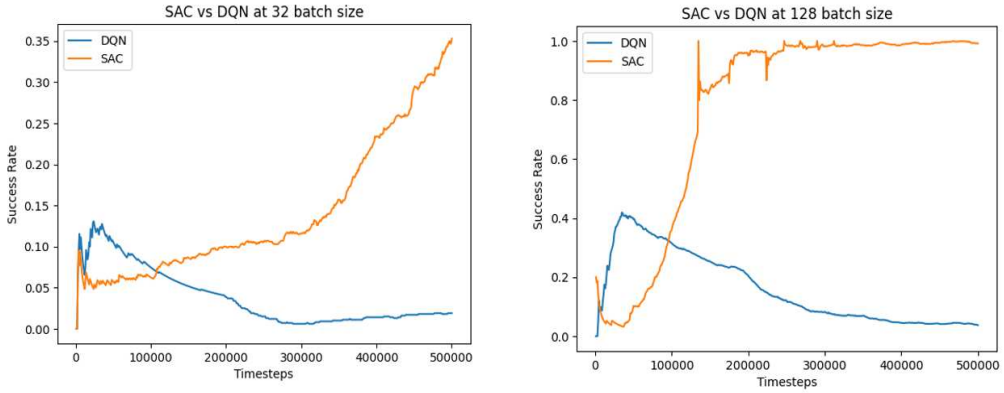

Figure 4. Comparative performance of SAC and PPO algorithms at batch sizes 128 and 32. (a) 32 batch size. (b) 128 batch size.

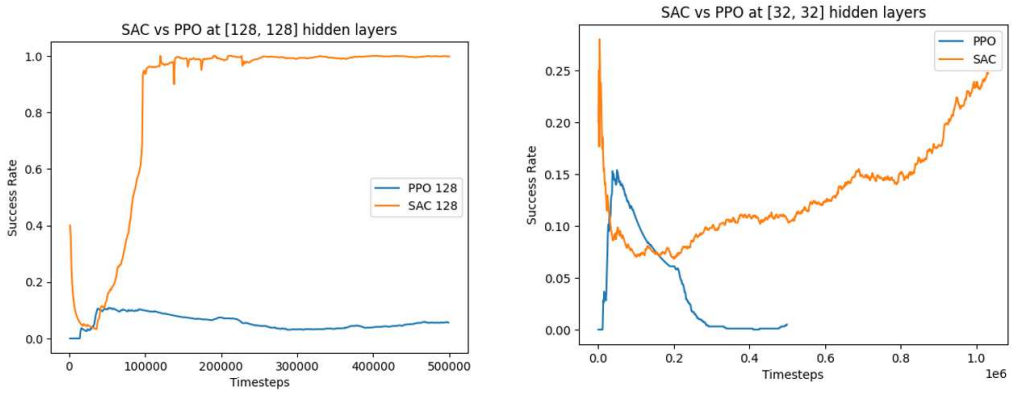

Figure 5. Comparative performance of SAC and PPO algorithms with varied number of hidden layers. (a) [128, 128] hidden layers. (b) [32, 32] hidden layers.

 doi:10.20944/preprints202312.1134.v1

3.3. Object-Specific Success Rate

Using SAC as the preferred algorithm, the agent was once again trained to grasp three different types of objects – remote controller, soap bar and mug (Figures 6 and 7). This is to determine how the object shape and size affect the agent convergence at an optimal policy.

The results show that the mug was the easiest object for the agent to grasp, as it converged the fastest, achieving a success rate of over 0.9 after 200,000 timesteps. The remote controller was the second easiest object to grasp, with a success rate of over 0.8 after 250,000 timesteps. However, the soap bar was the most challenging object for the agent, as it failed to converge, with a success rate that remained between 0.0 and 0.008 throughout the training process.

Table 4. Comparative Analysis of SAC, PPO, and DQN Algorithms with Varied Number of Hidden Layers in the Model Architecture.

Algorithm	SAC	PPO	DQN
Convergence Speed	Quickest to convergence	Slow convergence	Slow convergence
Hyperparameter Sensitivity	Robust to hyperparameters, relatively easier to tune	Performance is heavily affected when deviating slightly from an optimal hyperparameter	Performance is also heavily affected
Training time	Takes the longest time to train	Lower training time than SAC	Requires the least amount of training time

These results are consistent with our understanding of the properties of these objects. The remote controller and mug are both easy to grip with well-defined shapes and surfaces that the agent can easily detect and grasp. On the other hand, the soap bar has a smooth surface and is difficult to grip, making it challenging for the agent to learn a successful grasping policy.

These results can also be explained in terms of the exploration-exploitation trade-off. The agent's task is to maximize its reward, which in this case is the success rate of grasping an object. To do this, the agent must explore different grasping strategies to find the one that maximizes its reward. However, if the exploration process is too extensive, the agent may fail to converge to an optimal policy. On the other hand, if the agent exploits the same grasping strategy without exploring others, it may miss out on better strategies.

In the case of the soap bar, the slippery surface and difficult-to-grip nature of the object may have made it more challenging for the agent to explore different grasping strategies. As a result, the agent may have become stuck in a suboptimal policy that did not lead to successful grasping.

In contrast, the well-defined edges and surface of the remote controller and mug may have made it easier for the agent to explore different grasping strategies, leading to faster convergence to an optimal policy. Also, the fact that the success rate of these objects continued to improve over time suggests that the agent was able to find better grasping strategies through exploration without getting stuck in a suboptimal policy.

MANOVA test was further conducted on the success rates of the three objects: mug, remote controller, and bar of soap, to investigate whether the type of object significantly affects the performance of the SAC algorithm. The results are shown in Table 5.

Based on the results of the MANOVA analysis, there is strong evidence to suggest that the type of object (remote controller, soap, mug) has a significant effect on the success rates of the prosthetic gripper. The Pillai's trace statistic of 0.6888 and the highly significant p-value (< 2.2e-16) indicate that the different object types contribute significantly to the variation in the success rate of the agent. The approximate F-value of 552.9 further supports the presence of a substantial effect. These findings

 doi:10.20944/preprints202312.1134.v1

suggest that different objects cause the RL agent to exhibit distinct performance levels in task completion.

Table 5. Results of the MANOVA Test Investigating the Impact of Different Object Types (Mug, Remote Controller, Soap Bar) on the Success Rates of the SAC Algorithm.

	DF	Pillai	F-Value	Den DF	Pr(>F)
Algorithms	2	0.6888	552.9	4210	< 2.2e-16 ***
Residuals	2105				

5. Conclusions

This study demonstrates the superiority of the SAC algorithm over DQN and PPO for training prosthetic hands to grasp objects effectively in terms of force and contact point. Importantly, the physical properties of the object, including shape and texture, significantly influence the success of prosthetic grasping. Recognizing these elements offers insights into creating robust prosthetic systems that prevent slippage or damage.

This research contributes significantly to the domain of Myo prosthetic hand development and emphasizes the advancing role of computer vision and machine learning technologies in this field. Notably, it highlights the influence of object physical properties, such as shape and texture, on prosthetic grasping. The study also refines traditional autoencoder models like those presented by Breyer, et al., (2019), introducing an innovative approach to sparse data handling in EMG-based applications by optimizing the learning rate for each parameter.

Future work, first, the Myo armband's second channel has hinted at significant muscle activities, warranting a deeper investigation. Understanding the distinct role of these muscles, detected by this specific channel, can unveil crucial insights. By identifying the prominence of this channel's captured activities, there's potential to refine prosthetic algorithms, enhancing grip accuracy and responsiveness based on the underlying muscular dynamics. Second, a Post Hoc analysis, potentially employing methods like pairwise comparisons or interaction effect reviews, can be performed to shed light on how different algorithms perform across various objects. Considering that objects can differ in attributes such as shape, texture, and weight, it is essential to pinpoint which algorithms work best with each attribute. Understanding this can help us choose the right algorithm for specific tasks, making prosthetics work better in different real-life situations. Third, the robustness and resilience of the method will be studied. While robustness may be well known, resilience refers to whether a method still works if underlying conditions are changed, see the definition or resilience in literature [24,25].

Author Contributions: Writing – original draft preparation, J.O.; writing – review and editing, J.O., A.O. and W.Z. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu S., Zhang H., Yin R., Chen A., Zhang W.J., 2017. Finite Element Analysis and Application of a Flexure Hinge Based Fully Compliant Prosthetic Finger. In: Fei M., Ma S., Li X., Sun X., Jia L., Su Z. (eds) Advanced Computational Methods in Life System Modeling and Simulation. ICSEE 2017, LSMS 2017. Communications in Computer and Information Science, vol. 761. Springer, Singapore.
2. S.Q. Liu, H. B. Zhang, R. X. Yin, A. Chen, and W. J. Zhang, "Flexure hinge based fully compliant prosthetic finger," in SAI Intelligent Systems Conference 2016, London, UK, September 2016.

3. A. Chen, R. Yin, L. Cao, C. Yuan, H. K. Ding and W. J. Zhang, "Soft robotics: Definition and research issues," 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Auckland, New Zealand, 2017, pp. 366-370, doi: 10.1109/M2VIP.2017.8267170.
4. A. Tony et al., "Toward a Soft Microfluidic System: Concept and Preliminary Developments,"2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), 2021, pp. 755-759, doi: 10.1109/M2VIP49856.2021.9665022.
5. A. Waris, I. K. Niazi, M. Jamil, K. Englehart, W. Jensen and E. N. Kamavuako, "Multiday Evaluation of Techniques for EMG-Based Classification of Hand Motions," in IEEE Journal of Biomedical and Health Informatics, vol. 23, no. 4, pp. 1526-1534, July 2019, doi: 10.1109/JBHI.2018.2864335.
6. Y. Zheng, L. Cao, Z. Q. Qian, A. Chen, and W. J. Zhang, "Topology optimization of a fully compliant prosthetic finger: design and testing," in proceedings of the 6th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2016), Singapore, Jun 2016, pp. 1037-1042.
7. Van der Riet, D., Stopforth, R., Bright, G., and Diegel, O. (2013). "An overview and comparison of upper limb prosthetics," in AFRICON (Mauritius: IEEE), 1–8.
8. Kyberd, P., Wartenberg, C., Sandsjö, L., Jönsson, S., Gow, D., Frid, J., Sperling, L. (2007). Survey of Upper-Extremity Prosthesis Users in Sweden and the United Kingdom. JPO: Journal of Prosthetics and Orthotics, 19(2). Retrieved from https://journals.lww.com/jpojournal/Fulltext/2007/04000/Survey_of_Upper_Extremity_Prosthesis_Users_in.6.aspx.
9. V. Mihajlović, B. Grundlehner, R. Vullers and J. Penders, "Wearable, Wireless EEG Solutions in Daily Life Applications: What are we Missing?," in IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 1, pp. 6-21, Jan. 2015, doi: 10.1109/JBHI.2014.2328317.
10. Y. Lin, W.J. Zhang, C. Wu, J. Dy, and G.S. Yang, 2009. A fuzzy logics clustering approach to computing human attention allocation using eyegaze movement cue. International Journal of Human-Computer Studies. Volume 67, Issue 5 (May 2009): Pages 455-463.
11. Y. Lin, W.J. Zhang, and G. Watson, 2003. Using Eye Movement Parameters for Evaluating Human-Machine Interface Frameworks under Normal Control Operation and Fault Detection Situations, International Journal of Human Computer Studies, Vol. 59/6 pp 837-873.
12. Hao, Z., Charbel, T., & Gursel, A. (2021). A 3D Printed Soft Prosthetic Hand with Embedded Actuation and Soft Sensing Capabilities for Directly and Seamlessly Switching Between Various Hand Gestures. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 75-80.
13. Castro, M., Pinheiro, W., & Rigolin, G. (2022, January 24). A Hybrid 3D Printed Hand Prosthesis Prototype Based on sEMG and a Fully Embedded Computer Vision System. Frontiers in Neurorobotics, 15. doi:10.3389/fnbot.2021.751282.
14. Czimmermann, T., Ciuti, G., Milazzo, M., Chiurazzi, M., Roccella, S., Oddo, C., & Dario, P. (2020). Visual-Based Defect Detection and Classification Approaches for Industrial Applications—A SURVEY. Sensors, 20(5). Retrieved from https://www.mdpi.com/1424-8220/20/5/1459.
15. Abbasi, B., Sharifzadeh, M., Noohi, E., Parastegari, S., & Žefran, M. (2019). Grasp Taxonomy for Robot Assistants Inferred from Finger Pressure and Flexion. 2019 International Symposium on Medical Robotics (ISMR), 1-7.E. P. Wigner, "Theory of traveling-wave optical laser," *Phys. Rev*., vol. 134, pp. A635–A646, Dec. 1965.
16. Y. Wang, H. Wu, R. H. Jhaveri and Y. Djenouri, "DRL-Based URLLC-Constraint and Energy-Efficient Task Offloading for Internet of Health Things," in IEEE Journal of Biomedical and Health Informatics, doi: 10.1109/JBHI.2023.3297525.
17. S. Lee, J. Kim, S. W. Park, S. -M. Jin and S. -M. Park, "Toward a Fully Automated Artificial Pancreas System Using a Bioinspired Reinforcement Learning Design: In Silico Validation," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 2, pp. 536-546, Feb. 2021, doi: 10.1109/JBHI.2020.3002022.
18. Zhiqin Qian, Yi Lv, Dongyuan Lv, Huijun Gu, W.J. Zhang, M.M. Gupta, 2020. A New Approach to New Polyp Detection by Pre-Processing of Images and Enhanced Faster R-CNN. IEEE Sensor Journal. doi: 10.1109/JSEN.2020.3036005.
19. Zhiqin Qian, Weiji Jing, Yi Lv, and Wenjun Zhang, 2022. Automatic Polyp Detection by Combining Conditional Generative Adversarial Network and Modified You-Only-Look-Once. IEEE Sensor Journal. DOI: 10.1109/JSEN.2022.3170034.

20. Tan Zhang, Zihe Wang, Fengwei Li, Haoyang Zhong, Xuejuan Hu, Wenjun Zhang, Dan Zhang and Xiaoxu Liu, 2023. Automatic Detection of Surface Defects Based on Deep Random Chains, Expert Systems with Applications, Volume 229, Part A,2023, 120472, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2023.120472.
21. Baris, Y. (2020). Branch Dueling Deep Q-Networks for Robotics Applications. Master's Thesis.
22. M. Breyer, F. Furrer, T. Novkovic, R. Siegwart and J. Nieto, "Comparing Task Simplifications to Learn Closed-Loop Object Picking Using Deep Reinforcement Learning," in IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1549-1556, April 2019, doi: 10.1109/LRA.2019.2896467.Britannica, T. E. (2018, June 20). Metacarpal. Retrieved from Encyclopedia Britannica: https://www.britannica.com/science/metacarpal.
23. Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 15. 1929-1958.
24. B. Han, C.L. Liu, W.J. Zhang, 2016. A Method to Measure the Resilience of Algorithm for Operation Management, IFAC MIM 2016. France, Troyes on June 27-30.
25. Ratan Raj, Ashutosh Nayak, M.K. Tiwari, J.W. Wang, B. Han, C.L. Liu, W.J. ZHANG, 2014. Measuring Resilience of Supply Chain Systems using a Survival Model. IEEE Systems Journal. 9(2):377-381. 10.1109/JSYST.2014.2339552.