MATEC Web of Conferences **417**, 04005 (2025) https://doi.org/10.1051/matecconf/202541704005
2025 RAPDASA-RobMech-PRASA-AMI Conference

Towards robust gesture-based control for mobile robots: implementing and validating a limited recognition system on Voyager

Natasha Botha[1*], *Beatrice* van Eden[1] and *Thabisa* Maweni[1]

[1]Centre for Robotics and Future Production, Council for Scientific and Industrial Engineering, Pretoria, South Africa

Abstract. Gesture recognition plays a crucial role in human-robot interaction, allowing mobile robots to interpret and respond to non-verbal commands. This study presents the implementation of a limited gesture recognition system for the Voyager mobile ground robotic platform using a limited set of hand gestures. The system is validated in both simulation (Gazebo) and real-world environments and compared using metric such as recognition accuracy, latency, and user variability. This research highlights the challenges and potential solutions for deploying a limited gesture-based control in real-world robotic applications.

1 Introduction

In human-robot interaction (HRI), the push for more intuitive, natural communication methods have grown beyond traditional interfaces to incorporate modalities such as speech, gaze, and gestures [1,2]. Among these, hand gestures are a powerful, flexible form of non-verbal communication capable of conveying a wide range of commands and intentions [1-5]. Gesture recognition has become a prominent research area within HRI due to its inherent intuitiveness and potential for direct reflection of human intent [3].

By interpreting hand movements and configurations, robots can understand and respond to non-verbal commands, facilitating seamless collaboration and control [1,3]. This capability is especially useful for controlling mobile robotic systems in real-time, which operate in dynamic and often unstructured environments where traditional interfaces such as keyboards, joysticks or touchscreens are impractical or limit the user's mobility and focus [4,5,7]. Implementing effective gesture recognition can enhance a mobile robot's versatility and usability in various applications, ranging from service robotics to industrial automation [4].

Gesture recognition systems can be broadly categorised into device-based and vision-based approaches [8]. While device-based methods, such as data gloves equipped with sensors, offer high accuracy, they often present constraints for workers due to their bulky or intrusive nature [3-5,9]. In contrast, vision-based methods, using monocular or RGB-D cameras, provide a more natural, contactless and user-friendly alternative [1,3-6,8,9]. These

* Corresponding author: nbotha1@csir.co.za

systems extract spatial information directly from images, detecting hands and recognising gestures [1,3-6,8,9]. The increasing affordability and capabilities of camera technology have further propelled the adoption of vision-based solutions [1,3,4,8,9].

Monocular camera setups for hand gesture recognition often employ techniques such as image pre-processing, segmentation, feature extraction (e.g., shape-based features, texture, colour), and classification using methods like cross-correlation, Support Vector Machines (SVMs), or deep learning models [4,8]. Tamiru et al. [4] proposed a real-time vision-based hand gesture recognition approach for controlling a mobile service robot using five hand gesture commands, achieving a high recognition rate with a simple algorithm.

RGB-D cameras, such as the Microsoft Kinect, provide additional depth information, leading to more robust hand segmentation and gesture recognition, particularly in challenging lighting conditions [1,9]. For example, Robinson et al. [10] explored the use of skeleton tracking, provided by RGB-D sensors, for gesture-based control of robot manipulators and mobile robots.

Deep learning techniques, such as Convolutional Neural Networks (CNNs), have revolutionised vision-based gesture recognition showing state-of-the-art performance in recognising gestures from images and videos [1,4,6,11]. Deep learning models have notably improved issues such as edge blurring caused by complex backgrounds, rotation inaccuracy induced by fast movement, and delays due to computational costs [6]. Pre-trained CNN architectures like ResNet50 and MobileNetV2 have been fine-tuned using transfer learning for gesture recognition tasks, demonstrating promising results even with limited datasets [7]. More advanced techniques, such as 3D CNNs and Convolutional Long Short-Term Memory (LSTM), combine spatial and temporal data for multimodal gesture recognition [11]. CNNs excel at feature extraction for accuracy, LSTMs are effective for capturing temporal features and improving rotation accuracy in continuous gestures, while attention mechanisms facilitate lightweight and parallel processing, enhancing response time [6]. Often, these models are combined to leverage their complementary strengths, aiming for both high accuracy and efficiency [6].

Vision-based hand gesture recognition is being explored in a growing array of applications for human-robot interaction. In advanced manufacturing, it facilitates human-robot collaboration on cobotic platforms for tasks like polishing and deburring, enabling operators to provide commands for proper parameter settings and tool trajectories [5]. This allows human operators to take on more elevated roles of inspection and supervision [5]. In healthcare, autonomous service robots assist medical staff by transporting supplies and guiding patients, reducing physical strain and allowing professionals to focus on patient care [2]. They also offer crucial support to the elderly and physically impaired, from managing daily routines to providing mobility assistance through walker robots, cane-robots, and autonomous wheelchairs [2,12]. Beyond these, gesture recognition is vital for robot control in agriculture, enabling drones to be regulated via gestures for crop monitoring [6]. It also supports real-time silent communication and command in military operations and allows surgeons to control surgical equipment in clinical operations [6].

However, achieving robust real-time performance in vision-based gesture recognition for mobile robots presents several challenges. These include the sensitivity to visual occlusions and varying camera viewpoints, which are frequent in dynamic cobotic platforms where both the robot and operator move [5,6].

Environmental factors such as complex backgrounds, presence of obstacles, variations in lighting and the distance between the user and robot, can interfere with data acquisition and hand segmentation [1,4,6-8].

Computational constraints and the need for real-time performance introduce difficulties in resource efficiency and reaction time [6,9,11]. Many existing research efforts have primarily focused on maximising offline classification accuracy, sometimes at the expense of real-time feasibility and resource constraints [8].

Gesture variations between users (e.g. hand size, style, or skin tone) and unanticipated conditions add complexity to the development of robust, user-independent systems that can generalise across different individuals [6,8].

A persistent challenge remains in bridging the performance gap between controlled simulations and real-world deployment, where unseen conditions and environmental variability often degrade performance [6].

Addressing these challenges is critical to creating effective and efficient human-robot interactions. Validation in both simulation and real-world environments remains essential to refine performance and identify potential limitations [8].

This study contributes to the ongoing research by implementing and validating a gesture recognition system using the Voyager mobile robotic platform [13]. Recognising the constraints of real-world deployment, particularly the challenge of acquiring extensive training data, this study explores the feasibility of utilising a limited set of hand gestures to achieve basic control. To assess the system's performance and robustness, we conduct a comparative evaluation in both a simulated environment (Gazebo) and real-world scenarios. Performance is quantified using metrics such as recognition accuracy, latency, and how well the system adapts to different users.

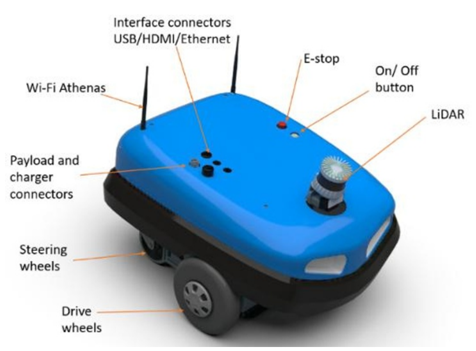
2 Methodology

In this section the methodology to develop and implement the limited gesture recognition system on the Voyager mobile ground robotics platform [13] is discussed.

2.1 Voyager mobile ground robotic platform

Voyager is a mobile ground robotic platform, Figure 1, designed for research and education in robotics and autonomous systems. Its architecture allows for the integration of various sensors and computational resources, making it a suitable platform for exploring advanced human-robot interaction techniques like gesture-based control.

Voyager is equipped with a Logitec C920 HD Pro Web camera for visual data collection. The computational specifications for Voyager are an Intel Core i9-10900TE CPU with 16GB DDR4 RAM, operating on Ubuntu 22.04 LTS with ROS2 Humble.



(a) CAD rendering of Voyager

(b) Voyager Platform

Fig. 1. Voyager mobile ground robotic platform.

2.2 Gesture recognition system

For the limited gesture recognition system, a vision-based approach is first considered for a more natural contactless method of communication with the robotic platform.

2.2.1 Limited data set

Implementing gesture recognition for mobile ground robotic platforms poses unique challenges. One major issue is the limitation of a gesture set—that is, choosing a small number of gestures that are both intuitive for users and reliably recognised by the system. The need to balance simplicity (for real-time recognition) with expressiveness (for a diverse set of commands) is a key consideration.

A limited set of five gestures is therefore considered for the gesture recognition system which includes forward, backward, turning left or right and the stop commands as illustrated in Figure 2. These gestures were selected based on their simplicity and relevance to the typical robotic navigation tasks. A dataset comprising 25 images per gesture (left and right hand) captured from a single user was used to train the gesture recognition model.

2.2.2 Algorithm

The gesture recognition system used in this study is MediaPipe Hands [14], an open-source Python package from Google, which allows for the detection of hand landmarks in an image. It is a high-fidelity hand and finger tracking solution which uses machine learning to identify 21 3D landmarks within an image of a hand [14], illustrated in Figure 3. MediaPipe Hands outputs the hand landmarks in both image and world coordinates and indicates whether it is the left or right hand when multiple hands are detected [14].

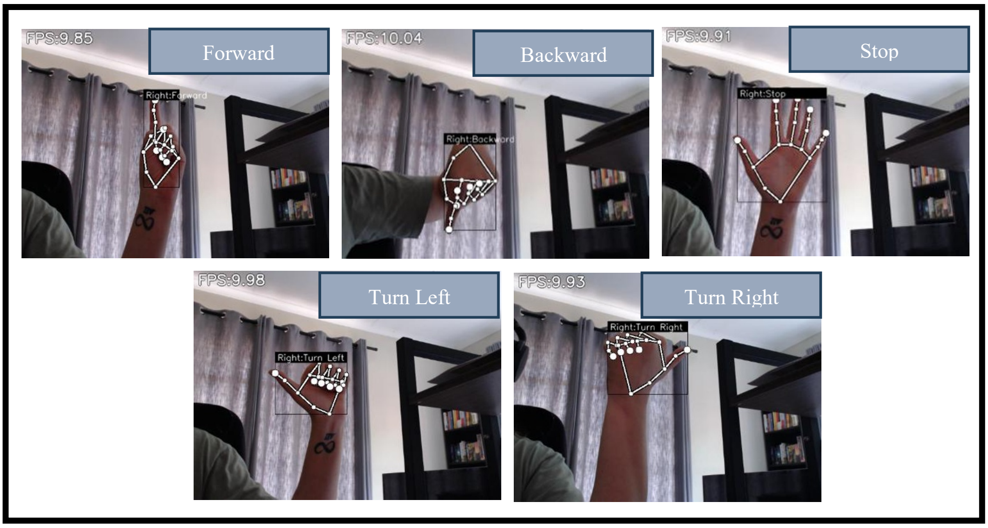


Fig. 2. Limited set of simple gesture recognition commands for basic robotic navigation.

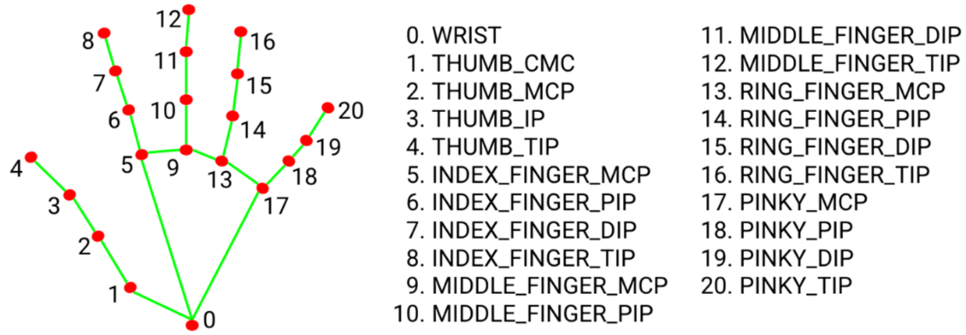


Fig. 3. MediaPipe Hands 21 3D landmarks of a hand.

MediaPipe Hands was integrated with Python using a machine learning model, via TensorFlow, to develop a gesture recognition system by Kiniv [15]. This repository uses MediaPipe Hands to detect the key points from an image and classifies the gesture, in real-time, into a predefined action (e.g. stop, forward, etc.) using a sequential machine learning model. A sequential machine learning model is a plain stack of layers where each layer has one input and one output tensor.

Different users may perform the same gesture with varying speed, angle, or style, which can affect recognition accuracy. Gesture recognition systems therefore must be robust enough to generalise across different users or provide adaptive learning mechanisms to customise the system to individual users over time [8, 16]. For this study, the gesture recognition system will be trained using an individual user's data but tested with different users to determine the system's recognition accuracy.

Implementing gesture recognition for mobile ground robotic platforms poses unique challenges. Additionally, the integration of gesture recognition with the robot's control and navigation systems introduces further complexities. For instance, the robot must interpret gestures while simultaneously processing environmental data for tasks like obstacle avoidance and path planning. Achieving seamless integration between the gesture recognition system and the robot's operating software is crucial for smooth, responsive operation [10,11]. For this study, the output of the gesture recognition system is integrated with the robotic platform's navigation control system only. Once a gesture is recognised, the robot performs the corresponding action (e.g., move forward, move backward, stop, etc.). The system operates in a closed-loop manner, where the robot continuously monitors its environment and responds to gestures in real-time. The gesture recognition system is illustrated in Figure 4.

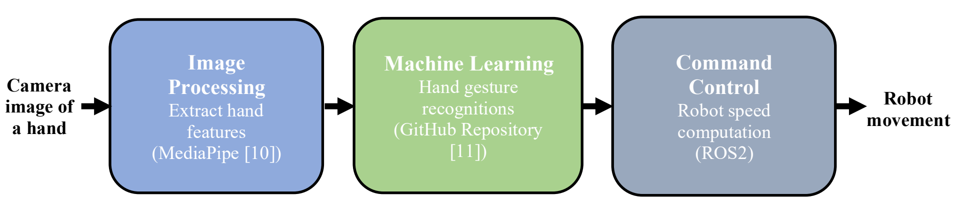


Fig. 4. Software implementation overview for the gesture recognition system.

The gesture recognition system is integrated into the Voyager platform's base code using ROS2 as illustrated in Figure 5. The green parts are existing within the Voyager code base, and those in blue indicate the nodes and topics relevant to the gesture recognition system.

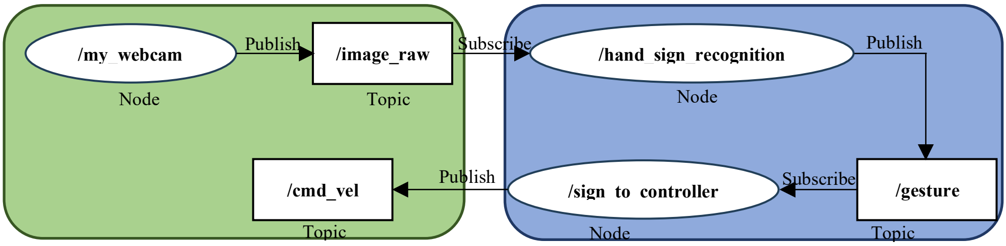


Fig. 5. Gesture recognition ROS2 integration on Voyager.

The web camera images from Voyager are published to the */image_raw* topic. The */hand_sign_recognition* node extracts the landmarks from the hand image by subscribing to */image_raw* and classifies the gesture. The gesture results are then published to the /gesture topic. The */sign_to_controller* node subscribes to the */gesture* topic to determine the gesture and applies the corresponding control commands to initiate the desired movement of Voyager. The control commands are then published to the */cmd_vel* topic resulting in Voyager moving according to the desired gesture.

2.2.3 Validation process

Simulation environments, such as Gazebo and Rviz2, have become essential tools for developing and testing gesture recognition systems in robotics. These environments allow researchers to model different scenarios, test system performance under controlled conditions, and refine algorithms before deploying them in the real world. In these controlled conditions variables such as lighting and background clutter can be manipulated.

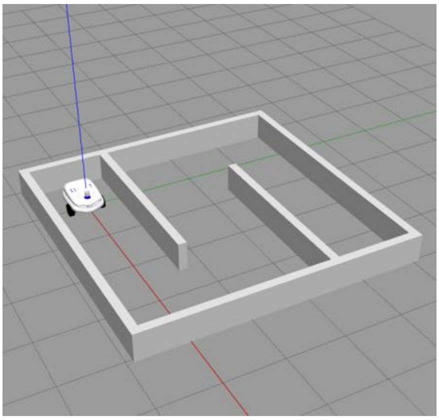
A pre-defined testing area is designed, illustrated in Figure 6, which will utilise all five gestures for testing. The simulation environment in Gazebo is developed on a Dell G16 7630 laptop with an Intel Core i7-13700HX within a virtual machine to launch Ubuntu 22.04 LTS with ROS2 Humble. In the simulation environment the webcam is mounted on the top of the computer screen, and the user sits in front of the screen and webcam to perform the gestures. In the real-world environment the webcam is mounted on the shell of the Voyager robot at an angle. The user then stands in front of the Voyager and webcam to perform the gestures and moves backwards as the Voyager moves forward to ensure they stay within the view of the camera. The action also occurs at a distance further away from the webcam and at an angle compared to the distance and angle from the webcam in the simulation.

However, a persistent challenge remains in bridging the gap between simulation and real-world performance.

To bridge this gap, researchers often validate systems by comparing key metrics—such as accuracy, latency, and error rates—between simulation and real-world tests. One common approach is to measure false positive and false negative rates across both environments to identify where the system's performance deviates [7]. This comparison is crucial for refining the algorithms and ensuring robustness before real-world deployment.

The gesture recognition system's performance in simulation and real-world environments is compared using the following metrics:

- **Recognition Accuracy:** The percentage of correctly recognised gestures.
- **Error Rates:** The percentage of false positives and false negatives.
- **User Variability:** The system's ability to generalise across different users with varying gesture styles.



(a) Simulation environment (Gazebo)

(b) Real-world environment

Fig. 6. Pre-defined testing area for the simulation and real-world testing.

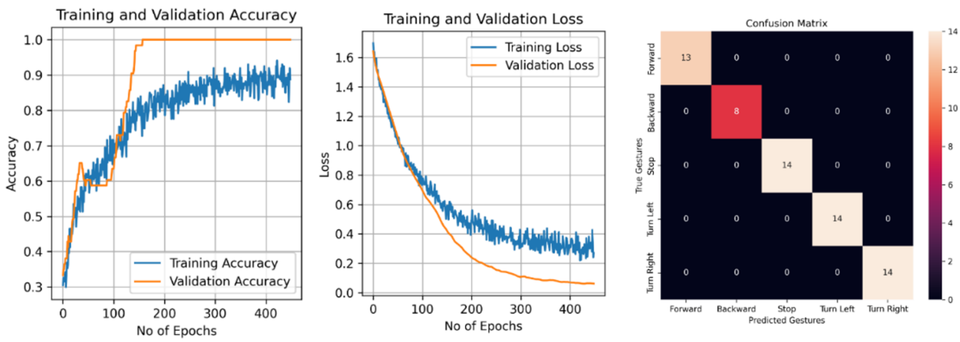
3 Results and discussion

This section presents the results from training the gesture recognition model and its subsequent implementation and validation in both simulated (Gazebo) and real-world environments. The primary objective was to use a limited set of five hand gestures (forward, backward, turn left, turn right and stop) to navigate the Voyager from a starting point to an end point. User variability was assessed by involving participants from different ethnic groups and genders including the single user from whom the training dataset was captured, performing tasks with both right and left hands.

3.1 Gesture recognition model

The gesture recognition model was trained using a limited dataset of 25 images per gesture for each hand, captured from a single user. The dataset was split 75% for training and 25% for validation. The results from training and validating the gesture recognition model are shown in Figure 7. The number of epochs on the x-axis refers to each time a dataset passes through the algorithm. Figure 7(a) illustrates the accuracy of the model which reached approximately 90 % for the training dataset, while the validation dataset achieved 100 % accuracy from around 150 epochs. In Figure 7(b) the loss function is illustrated where the aim is to minimise the loss, which provides an indication of the prediction error with regards to the ground truth. The loss function was minimised to approximately 0.3 for the training dataset and 0.06 for the validation dataset.

The confusion matrix, Figure 7(c), provides an indication on the model's performance by comparing its predictive and true results. The model demonstrated high recognition precision for the training data, with no false positives or negatives, although it's important to note this was for a single user.



(a) Model Accuracy (b) Model Loss (c) Confusion Matrix

Fig. 7. Gesture recognition model training and validation accuracy and loss results.

3.2 Simulation results

In the Gazebo simulation environment, the gesture recognition system exhibited a very low latency, with an almost instantaneous response between gesture input and robot execution.

The gesture recognition accuracy for each user is provided in Table 1, and an overall recognition accuracy of 95,64 % was achieved across various users. User 2 generally showed the lowest recognition accuracy, primarily attributed to her smaller hand size and distinct gesture style compared to the other users. Users with darker skin tones (users 3 and 4) also experienced slightly lower accuracy, as the system occasionally failed to recognise gestures for several frames, indicating "NONE".

The error rates for the gesture recognition system are summarised in Figure 8. False positive and false negative error rates were very low, below 2%. True positive rates were higher for "NONE" and "Stop" categories, which were frequently encountered gestures. True negative rates were lowest for "NONE" and "Stop," suggesting the system was better at identifying negative instances, possibly due to an imbalanced dataset favouring "Forward" and "Stop" gestures in the simulation.

Table 1. Simulation gesture recognition accuracy results (WF = White Female, BF = Black Female, IM = Indian Male, BM = Black Male, WM = White Male).

User	Left Hand Accuracy (%)			Right Hand Accuracy (%)		
	Test 1	Test 2	Average	Test 1	Test 2	Average
User 1 (WF, dataset)	98,11	97,38	97,75	97,18	94,09	95,63
User 2 (BF)	93,80	90,46	92,13	95,15	96,87	96,01
User 3 (IM)	95,14	96,66	95,90	94,06	94,06	94,06
User 4 (BM)	93,02	96,48	94,75	97,39	94,09	95,74
User 5 (WM)	96,50	97,43	96,97	97,80	97,65	97,73
User 6 (WF)	96,75	97,66	97,20	96,58	98,21	97,39
Average	**95,55**	**96,01**	**95,78**	**96,36**	**95,82**	**96,09**

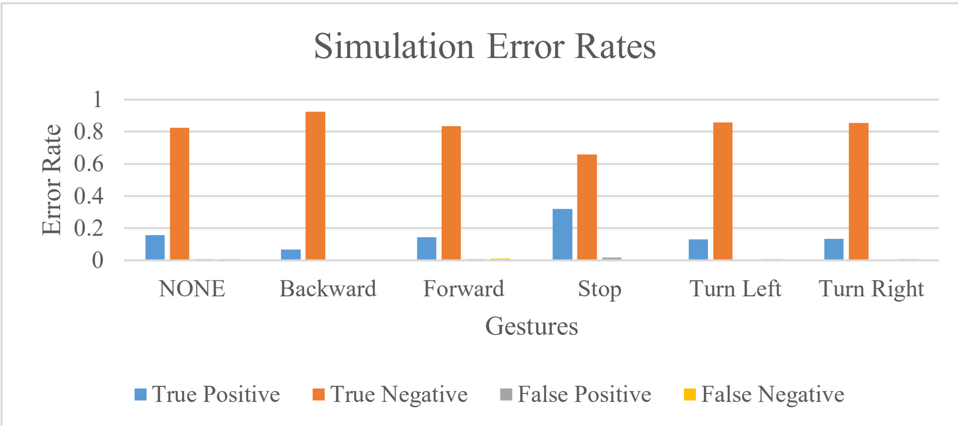


Fig. 8. Error rates for the simulation results.

A confusion matrix containing all the simulation results for all user tests is presented in Figure 9. The gesture recognition system can correctly identify gestures from a limited dataset, with only slight deviations between the intended (or actual) gesture provided and that predicted by the system.

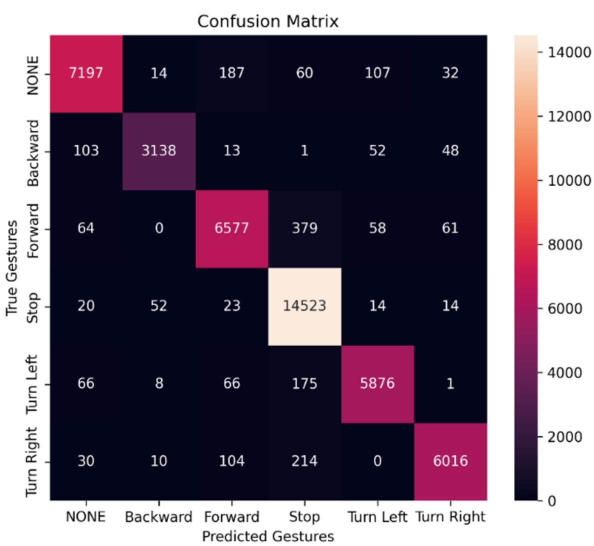
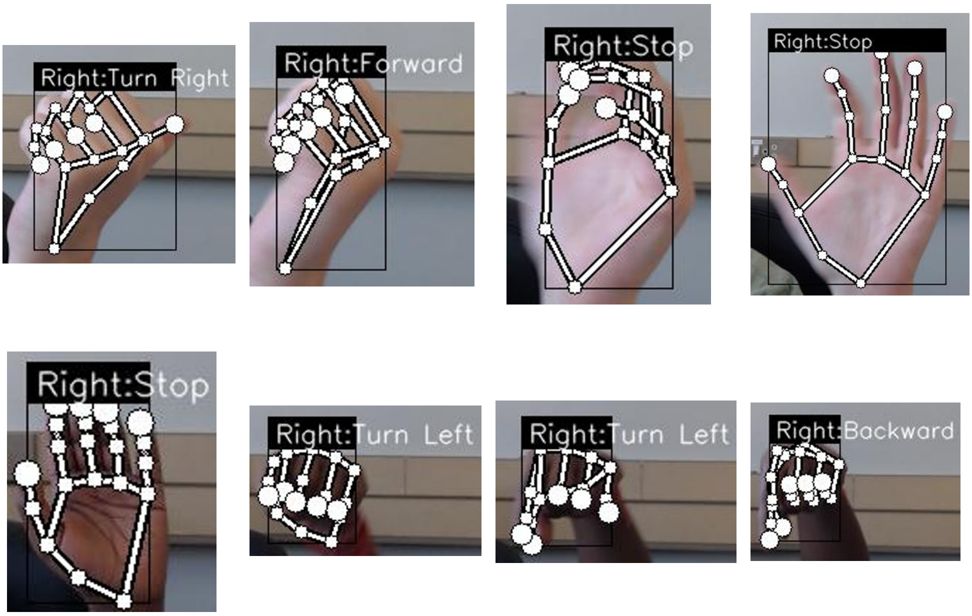


Fig. 9. Confusion matrix for all simulation results (accuracy = 95,64 %).

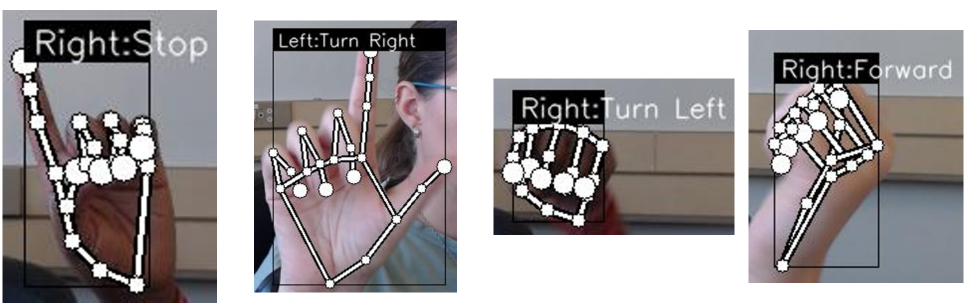
The lower accuracy and error rates are largely due to:

- **Transitions between different gestures:** Intermediary hand poses during transitions (e.g., from "Turn Right" to "Stop") were sometimes incorrectly categorised (e.g., as "Forward"), see Figure 10(a), because the system was not trained on these transitional poses.
- **Incorrect recognition due to hand positions:** For instance, "Forward" gestures were often misidentified as "Stop" if knuckles were not fully down, or as "Turn" gestures if a thumb was extended, see Figure 10(b).

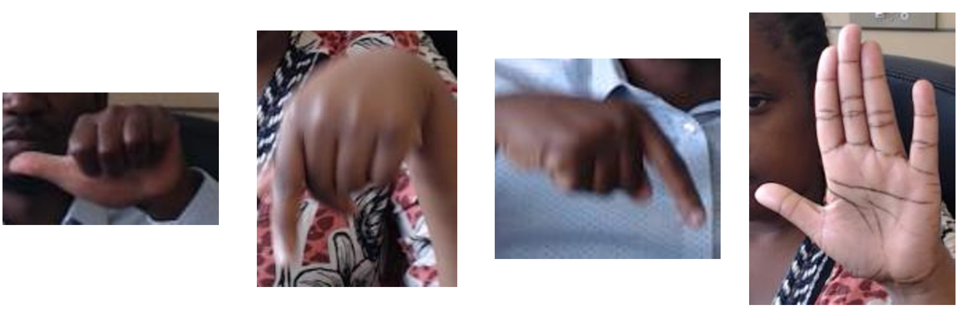
- **Inability to identify gestures:** This was particularly observed for users with darker skin tones, even when gestures were clearly shown as seen in Figure 10(c).



(a) Transitional error



(b) Incorrect recognition



(c) Not identifying a gesture

Fig. 10. Examples of identified causes for the lower gesture recognition accuracy.

3.3 Real-world results

Real-world experimental testing on the Voyager also showed very low latency.

The gesture recognition accuracy for each user is provided in Table 2, and an overall recognition accuracy was reduced to 88,29 % in the real-world environment. User 2's accuracy was significantly lower, with her smaller hand size and gesture style being further exacerbated by the increased distance of the camera from the user, leading to more false positives and negatives. While darker skin tones had more "NONE" gestures in post-processing, their overall recognition accuracy was similar to other users in real-world tests, suggesting a lesser influence compared to simulation.

Table 2. Experimental gesture recognition accuracy results (WF = White Female, BF = Black Female, IM = Indian Male, BM = Black Male, WM = White Male).

User	Left Hand Accuracy (%)			Right Hand Accuracy (%)		
	Test 1	Test 2	Average	Test 1	Test 2	Average
User 1 (WF, dataset)	99,42	97,83	98,63	97,16	98,31	97,73
User 2 (BF)	88,97	73,43	81,20	64,59	72,86	68,73
User 3 (IM)	95,29	96,52	95,91	91,81	98,02	94,91
User 4 (BM)	92,93	93,79	93,36	85,41	94,46	89,93
User 5 (WM)	92,82	95,09	93,96	95,44	95,56	95,50
User 6 (WF)	95,95	94,07	95,02	92,99	94,87	93,93
Average	94.24	91,79	93,01	87,90	92,35	90,12

The error rates for the gesture recognition system are summarised in Figure 11. False positive and false negative error rates remained low, below 8%. The "NONE" category had a very high true negative rate, indicating instances where gestures were outside the camera's view or not identified.

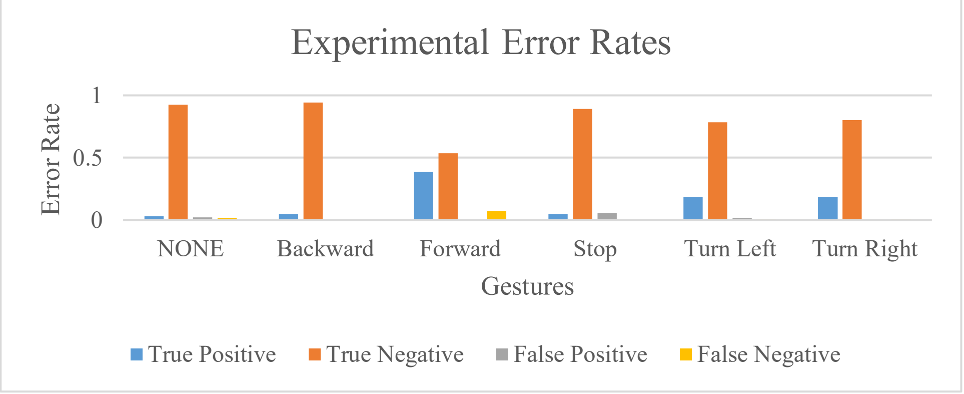


Fig. 11. Error rates for the experimental results.

A confusion matrix containing all the simulation results for all user tests is presented in Figure 12.

The lower accuracy for the real-world test is attributed to the following additional causes:

- **Unintended gestures:** Gestures made while waiting for the test to start/finish, or while lowering hands, were sometimes incorrectly identified.
- **Gestures outside camera view:** Gestures outside the camera's view were logged as "NONE" or incorrectly identified if only a partial hand was visible.
- **Distance from camera and hand size:** Increased distance particularly impacted users with smaller hands, leading to difficulties in landmark detection and frequent misclassifications (e.g., "Forward" being recognized as "Stop" or "Turn").
- **Environmental conditions:** Unlike simulation, inability to identify gestures was observed for all skin tones, possibly due to natural light variations in the laboratory.

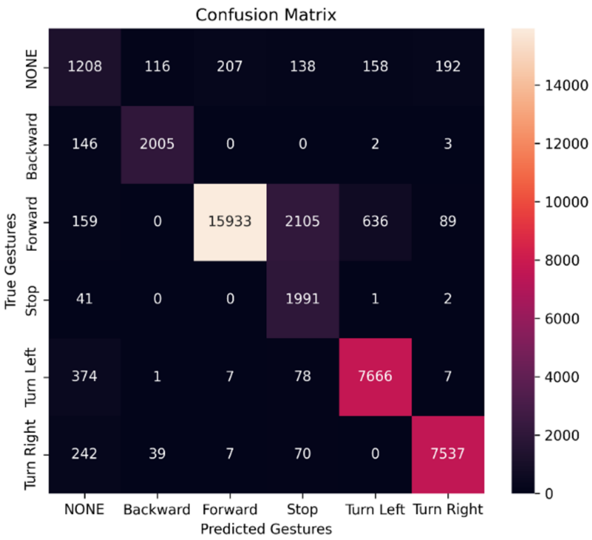


Fig. 12. Confusion matrix for all experimental results (Accuracy = 88,29 %).

3.4 User variability

The system exhibited slight deviations in accuracy based on skin tone, with generally lower accuracy for darker skin tones in simulation, but this effect was less pronounced in real-world tests, possibly due to improved natural lighting.

Variations in how users formed gestures, especially user 2's distinct "Forward" sign and users 4 and 5's extended thumbs, led to misclassification.

Despite these differences and being trained on a limited single-user dataset, the gesture recognition system was considered robust, adaptable and user-independent.

3.5 Performance comparison

The overall gesture recognition accuracy decreased from 95,64 % in simulation to 88,29 % in the real-world environment. This drop is primarily attributed to the differences in camera setup (angle, distance, and mounting position) between simulation (camera mounted on screen, user close) and real-world (camera mounted on robot, angled upwards, user further away). The camera distance particularly impacted users with smaller hands, making landmark prediction challenging. Despite these challenges, the low false positive and false negative error rates in both environments indicate the system's robustness in correctly identifying gestures.

The system generally adapted well to different users and camera positions, emphasising its robustness given the single-user training dataset.

4 Conclusion

This study demonstrated the development and validation of a hand gesture recognition system for mobile ground robotic platforms using a limited dataset, highlighting both its potential and challenges encountered.

Results from the simulations illustrated an overall gesture recognition accuracy of 95,64 %, with relatively low false positive and negative error rates. The lower accuracy can be attributed to recognition errors during the transitions between different gestures, inaccurate labelling of a gesture and not being able to identify a gesture even though it is clearly shown.

Results from the real-world showed a reduced overall gesture recognition accuracy of 88,29 %, with low false positive and negative error rates. In addition to the same recognition errors observed in the simulation environment, the lower accuracy of the real-world environment is attributed to unintended gestures while waiting for the test or officially start or finish, gestures not within the view of the camera being predicted incorrectly based on a partial view of the hand, and the distance between the user and the camera.

While the system showed promising results in controlled simulations, real-world deployment exposed performance limitations due to environmental variability, camera position and user differences. The differences in the camera setup between the simulation and real-world tests is considered to contribute most to the decrease in recognition accuracy as the gesture recognition system was only trained on the camera setup used for the simulation environment. These findings emphasise the need to bridge the performance gap between simulation and real-world application.

Despite these challenges, the limited dataset gesture recognition system still performed extremely well and is robust, adaptable and user independent.

Gesture recognition remains a powerful tool for enhancing human-robot interaction (HRI), enabling intuitive and natural control. However, achieving robust, real-time performance across diverse environments and users requires further innovation. Future work should prioritise several key directions:

- **Adaptive Learning Algorithms** that can generalise across different users and environments, which allow robots to personalise their gesture recognition models based on individual user behaviour. This approach reduces the likelihood of false positives or negatives, particularly in environments where users may perform gestures differently.
- **Multimodal Input Systems** by integrating inputs from other sensors and recognition modalities can improve robustness by compensating for the weaknesses of any single modality. This fusion of data sources can enhance recognition accuracy and reliability in dynamic, unstructured environments.
- **Contextual Awareness** by incorporating environmental context and user behaviour patterns can improve gesture interpretation. Context-aware systems could differentiate between intentional commands and incidental movements, reducing false positives and enhancing system intelligence.

By undertaking these future directions, gesture recognition technology can move beyond its current limitations, enabling more intuitive, accessible, and reliable HRI. With continued advancements in machine learning, sensor fusion, and context-aware systems, gesture recognition holds significant promise for shaping the future of mobile robotics, transforming how humans and robots collaborate in diverse and dynamic environments.

References

[1] J. Qi, L. Ma, Z. Cul and Y. Yu, Computer vision-based hand gesture recognition for human-robot interaction: a review. Complex & Intelligent Systems, (2023). DOI: https://doi.org/10.1007/s40747-023-01173-6.
[2] A. Eirale, M. Martini, M. Chiaberge, Human Following and Guidance by Autonomous Mobile Robots: A Comprehensive Review. IEEE Access, vol. 13, pp. 42214 – 42253 (2025). DOI: https://doi.org/10.1109/ACCESS.2025.3548134.
[3] L. Guo, Z. Lu, L. Yao, Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review. IEEE Transactions on Human-Machine Systems, vol. 51(4), pp. 300-309 (2021). DOI: https://doi.org/10.1109/THMS.2021.3086003.
[4] H.G. Tamiru, R.S. Yan, D.H. Long, D.H., Vision-based Hand Gesture Recognition for Mobile Service Robot Control. 8th International Conference on Manufacturing Science and Engineering, pp. 48-55 (2018). DOI: https://doi.org/10.2991/icmse-18.2018.11.
[5] C. Hubert, N. Odic, M. Noel, S. Gharib, S.H.H. Zargarbashi, L. Séoud, MuViH: Multi-View Hand gesture dataset and recognition pipeline for human-robot interaction in a collaborative robotic finishing platform. Robotics and Computer-Integrated Manufacturing, vol. 94, pp. 102957 (2025). DOI: https://doi.org/10.1016/j.rcim.2025.102957.
[6] C. Cui, M.S. Sunar, G.E. Su, Deep vision-based real-time hand gesture recognition: a review. PeerJ Computer Science, vol. 11, pp. 2921 (2025). DOI: https://doi.org/10.7717/peerj-cs.2921.
[7] S. Budzan, R. Wyzgolik, M. Kciuk, K. Kulik, R. Masłowski, W. Ptasinski, O. Szkurłat, M. Szwedka, Ł. Wozniak, Using Gesture Recognition for AGV Control: Preliminary Research. Sensors, 23, pp. 3109 (2023). DOI: https://doi.org/10.3390/s23063109.
[8] N. Mohamed, M.B. Mustafa, N. Jomhari, A Review of the Hand Gesture Recognition System: Current Progress and Future Directions. IEEE Access, vol. 9, pp. 157422 – 157436 (2021). DOI: https://doi.org/10.1109/ACCESS.2021.3129650.
[9] M. Oudah, A. Al-Naji, J. Chahl, Hand Gesture Recognition Based on Computer Vision: A Review of Techniques, Journal of Imaging, vol. 6 (2020). DOI: https://doi.org/10.3390/jimaging6080073.
[10] N. Robinson, B. Tidd, D. Campbell, D. Kulic, P. Corke, Robotic Vision for Human-Robot Interaction and Collaboration: A Survey and Systematic Review. Journal of the Association for Computing Machinery, vol. 37(4), pp. 111 (2021). DOI: https://doi.org/10.1145/1122445.1122456.
[11] O. Köpüklü, A. Gunduz, N. Kose, G. Rigoll, Real-time Hand Gesture Detection and Classification using Convolutional Neural Networks. 14th IEEE International Conference on Automatic Face and Gesture Recognition, 14-18 May (2019). DOI: https://doi.org/10.1109/FG.2019.8756576.
[12] P. Gawli, T. Desale, T. Deshmukh, S. Dongare, V. Gaikwad, S. Hajare, Gesture and Voice Controlled Wheelchair. 4th International Conference on Sentiment Analysis and Deep Learning, Bhimdatta, Nepal, 8-20 February 2025 (2025). DOI: https://doi.org/10.1109/ICSADL65848.2025.10933372.
[13] K. Purdon, J.S. Dickens, W. De Ronde, K. Ramruthan, G. Crafford, Voyager, a ground mobile robotic platform for research development, RAPDASA-RobMech-PRASA-AMI Conference, CSIR International Convention Centre, Pretoria, South Africa, 30 October – 2 November 2023 (2023). DOI: https://doi.org/10.1051/matecconf/202338804016.

[14]Google AI for Developer, Hand landmarks detection guide. Available at: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker. Accessed on: 19 December 2024. (2024)
[15]Kiniv, Github: hand-gesture-recognition-mediapipe. Available at: https://github.com/kinivi/hand-gesture-recognition-mediapipe. Accessed on: 19 December 2024. (2020)
[16]S. Mitra, T. Acharya, Gesture recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37(3), 311-324 (2007). DOI: https://doi.org/10.1109/TSMCC.2007.893280.