

# Development of Gesture-Controlled Robotic Assistant for the Disabled Using Deep Learning

Nurul Nadhirah Azli<sup>1</sup>, Allan Melvin Andrew<sup>1,2\*</sup> and Charis Samuel Solomon Koilpillai<sup>3</sup>

<sup>1</sup>Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia

<sup>2</sup>Centre of Excellence for Intelligent Robotics & Autonomous Systems (CIRAS), Universiti Malaysia Perlis, Pauh Putra Campus, 02600 Arau, Perlis, Malaysia

<sup>3</sup>Faculty of Industrial Management, Universiti Malaysia Pahang Al- Sultan Abdullah, 26300 Gambang, Pahang, Malaysia

Received 28 November 2025, Revised 19 December 2025, Accepted 28 December 2025

## ABSTRACT

*The rising rate of motor disabilities, particularly in stroke patients, has underscored the need for innovative assistive technologies to enhance their quality of life. This paper focuses on the development of a gesture-controlled robotic assistant using deep learning techniques that aimed for individuals with partial hand mobility. The primary objective was to design a system that recognizes predefined hand gestures in real-time to control household appliances and notify caregivers of patient needs. The research leverages Convolutional Neural Network (CNN), You Only Look Once (YOLO) and Single-Shot Multibox Detector (SSD) for gesture recognition. The system integrates Internet of Things (IoT) devices, enabling real-time feedback and automation through platforms like Telegram. A dataset consists of five specific hand gestures was used to train models, with augmented data to improve robustness across varying environmental conditions. The system achieved 97% real-time gesture recognition accuracy in controlled settings, demonstrating its reliability in improving stroke patients' interaction with their environment. This research highlights the potential of combining deep learning with IoT for the development of accessible, cost-effective assistive technologies.*

**Keywords:** Assistive Technology, CNN, Hand Gesture Recognition, Real-Time Detection, SSD, Stroke Patient, YOLO

## 1. INTRODUCTION

The World Stroke Organization (WSO) reported that each year, there are 12.2 million new stroke cases globally, with 101 million individuals living with long-term stroke effects [1]. The recovery process can take weeks, months, or even years, influenced by factors such as the stroke severity, physical and psychological condition, and social support [2]. Thus, stroke patients often require assistant or caregiver and experience limited motor capabilities to perform daily activities [3][4].

Recent advances in deep learning and Internet of Things (IoT) technologies have led to the development of intelligent assistive systems capable of enhancing rehabilitation and independence. These technologies have been widely applied in applications such as navigation assistance for visually impaired users and gesture-based interfaces in healthcare systems. Deep learning enables efficient processing of large volumes of data from IoT-based sensors, allowing patterns to be identified and meaningful insights to be generated for decision-making support [5]. Gesture-controlled systems have shown strong capability in translating human hand movements into executable device commands, making them suitable for assistive and

---

\*allanmelvin@unimap.edu.my

rehabilitation applications [4][6]. Models such as Convolutional Neural Networks (CNN), YOLO, and SSD have improved gesture-recognition performance under controlled conditions [7][8][9].

However, several limitations hinder real-world deployment of gesture-based assistive system. Existing models typically require high computational power, making implementation on low-cost embedded devices challenging [10]. Furthermore, environmental variables such as illumination and background complexity, along with the lack of personalized model tuning for diverse user needs, often reduce accuracy and reliability in practical environments [9][10][11]. Additionally, deep learning models rely on large, labelled datasets, and performance may degrade significantly if the model is not trained on sufficiently diverse samples [2][12].

To address these limitations, this paper develops an optimized gesture-controlled assistant using YOLO integrated with IoT integration to support real-time control for disabled users.

## 2. METHODOLOGY

The proposed system aims to assist stroke patients by recognizing specific hand gestures, which are used to control home appliances and alert caregivers. The system consists of three main components include a hardware setup, a software platform and an IoT communication for data transmission between the system and devices.

The core of the system is a gesture recognition model that is trained to detect and classify various hand gestures made by the patient. The system receives these gestures via a camera system and translates them into specific commands, such as activating or deactivating appliances or notifying a caregiver.

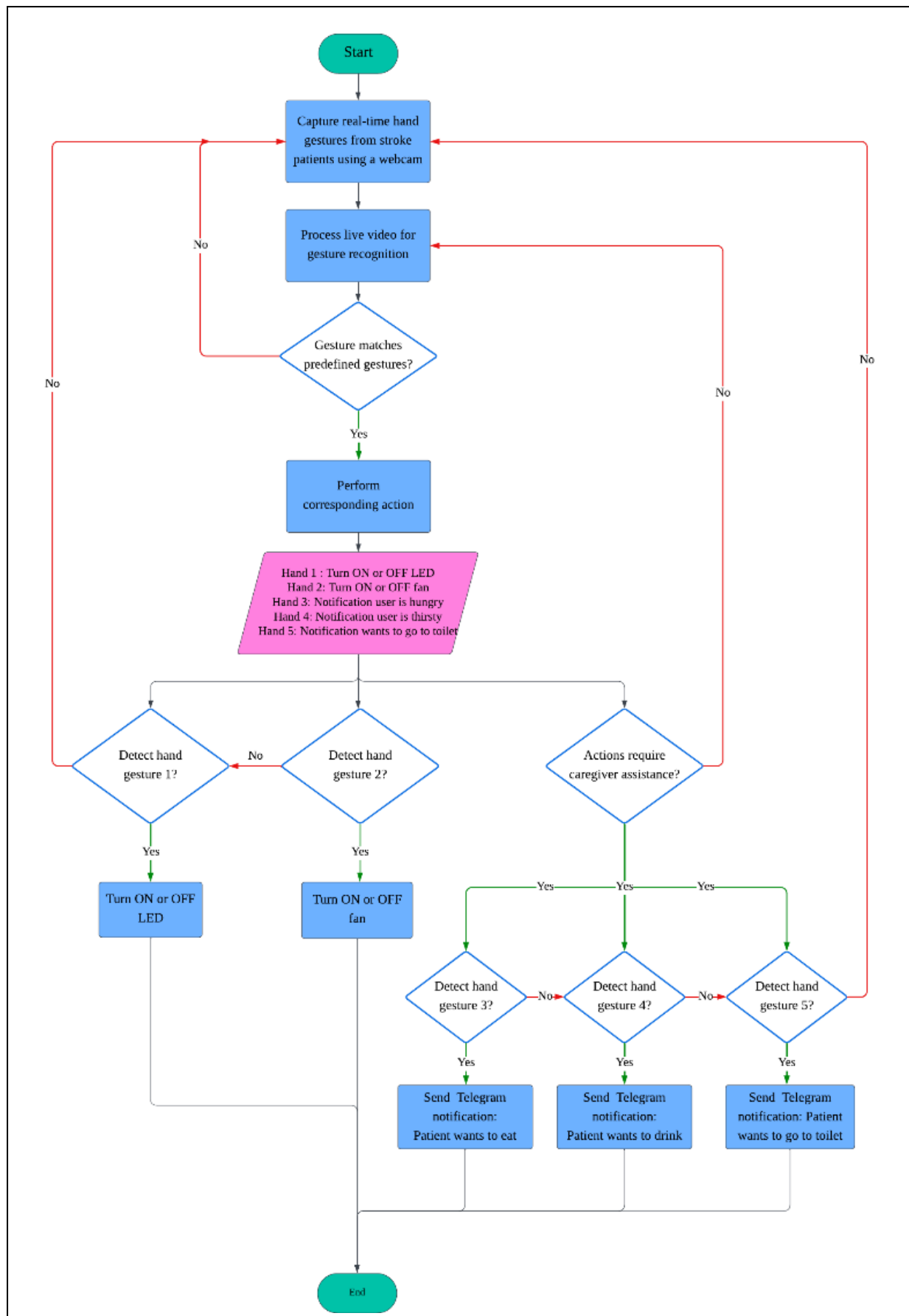
The flowchart in Figure 1 illustrates the integration of real-time gesture detection, IoT device control and caregiver notifications, ensuring that the stroke patient can easily interact with their environment and communicate their needs without needing physical assistance.

### 2.1 Software Integration

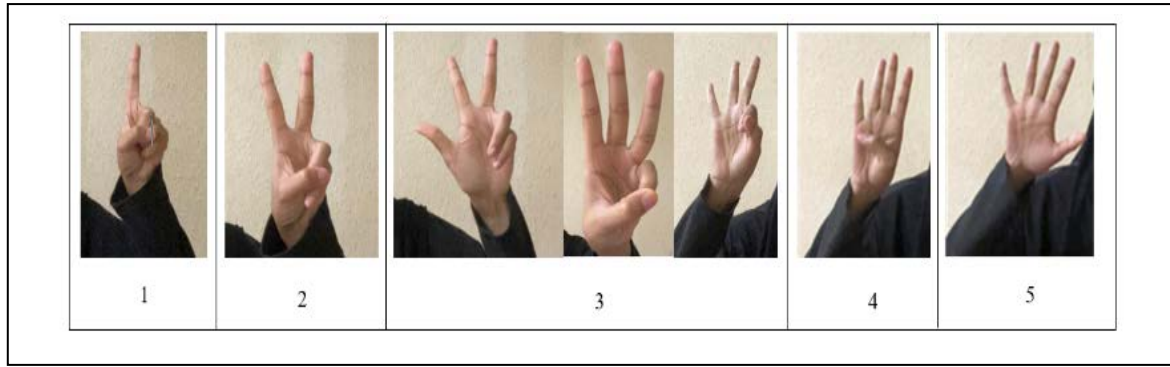
The software part of the system focuses on the integration of gesture recognition models and communication that bridges the devices and the IoT network.

Data collection is the first critical step in software integration. Hand gesture images, including five predefined gestures as in Figure 2 for tasks such as controlling lights or fans and notifying caregivers, are collected from Roboflow and Kaggle and captured under diverse conditions, such as varying lighting, background complexity and camera orientations. These images from the dataset that is used to train the system.

Model training involves using deep learning models, including Convolutional Neural Networks (CNN), YOLO and SSD. These models are trained using Google Colab and are optimized for accuracy in real-time gesture detection [13][14]. The dataset consists of five predefined gesture classes with an equal distribution of 90 images per class, with a total of 450 images. The collected dataset of 450 images was augmented to increase variability in lighting, background and orientation, and was split into training (80%), validation (10%) and testing (10%) sets. The training process ensures that the model can detect gestures accurately under real-world conditions. The performance of the trained models is evaluated using metrics such as accuracy, precision, recall and F1-score to measure the reliability and efficiency of gesture classification.



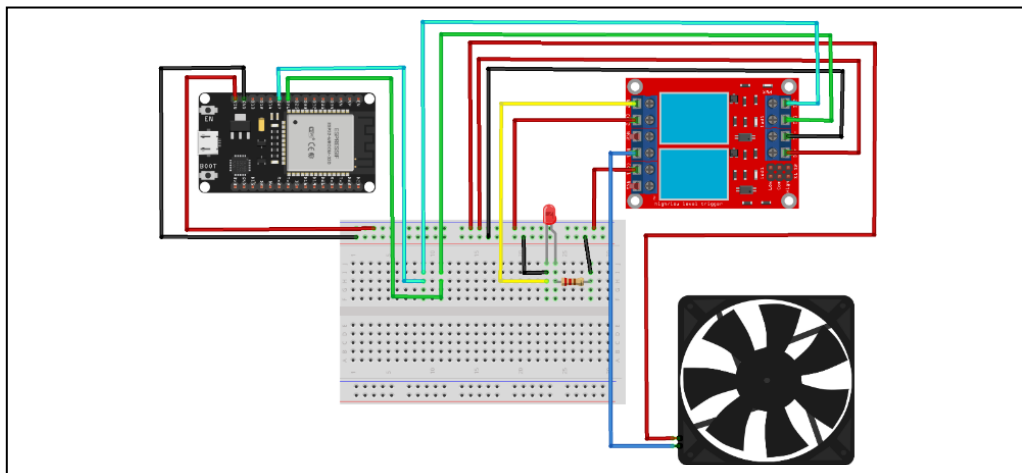
**Figure 1.** Overall system operation flowchart



**Figure 2.** Predefined hand gestures

## 2.2 Hardware Integration

The hardware integration focuses on IoT devices, which are responsible for carrying out the actions corresponding to recognized gestures. The camera system captures the patient's hand gestures and processes these images using the trained models. The system employs ESP32, which interfaces with the IoT devices and controls their actions based on the gestures detected. The circuit diagram in Figure 3 illustrates the connection between the ESP32, 2-channel relay, the fan and the LED. This integration allows the patient to interact with their environment seamlessly using hand gestures, and it ensures reliable real-time performance for device control and caregiver notifications.



**Figure 3.** Schematic diagram of hardware system

## 2.3 Overall System Operation

The overall system operates in real-time to assist stroke patients by recognizing their hand gestures, which are captured through a webcam. The system initiates with the capture of hand gestures from the patient using the webcam. Once the live video feed is obtained, it is processed for gesture recognition, where the system evaluates if the gesture matches one of the predefined hand gestures.

If the gesture matches one of the predefined gestures, the system proceeds to perform the corresponding action. The predefined gestures include:

- Gesture 1: Turn ON or OFF LED.
- Gesture 2: Turn ON or OFF fan.
- Gesture 3: Notify caregiver that the patient is hungry.
- Gesture 4: Notify caregiver that the patient is thirsty.
- Gesture 5: Notify caregiver that the patient wants to go to the toilet.

If the correct gesture is detected as Gesture 1 or Gesture 2, the system will perform the corresponding action, such as turning on / off the LED or fan. In cases where actions are detected that may require caregiver assistance such as Gestures 3, 4 and 5, the system will send a Telegram notification to the caregiver alerting them of the patient's needs. If no assistance is needed, the system proceeds to wait for the next gesture.

### 3. RESULTS AND DISCUSSION

The performance of the developed gesture-controlled robotic assistant system was evaluated using several metrics to determine its efficiency in recognizing hand gestures in real-time. Using deep learning models, including YOLO, SSD and CNN, the system demonstrated the ability to detect five predefined hand gestures that correspond to controlling a light, fan, and notifying caregivers of the patient's needs.

#### 3.1 Model Evaluation Metrics

The model evaluation metrics in Table 1 provides a comparative analysis of the performance of YOLO, CNN, and SSD used for the gesture recognition system. The metrics used to evaluate these models include accuracy, precision, recall, and F1-score, all of which offer insights into how well each model detects and classifies hand gestures.

**Table 1** Model evaluation metrics

Model	Accuracy (%)	Precision	Recall	F1-score
YOLO	97	0.96	0.95	0.95
CNN	31.6	0.32	0.31	0.31
SSD	72	0.72	0.71	0.72

Accuracy measures the proportion of correct predictions out of all predictions made. YOLO outperforms both CNN and SSD with an impressive accuracy of 97%, demonstrating its ability to correctly classify the majority of gestures. In contrast, CNN only achieved a low accuracy of 31.6%, indicating that it struggled significantly with gesture recognition, while SSD performed better with 72% accuracy, though still far behind YOLO.

Precision assesses the number of correct positive predictions out of all positive predictions made by the model YOLO again leads with precision of 0.96, meaning that when it identified a gesture as positive, it was correct 96% of the time. On the other hand, CNN's precision of 0.32 indicates that many of the gestures it identified as positive were false positives, thus it had a high rate of misclassifications. SSD achieved a precision of 0.72, suggesting it performed better than CNN in terms of minimizing false positives, but still underperformed compared to YOLO.

Recall, which measures the model's ability to detect all true positive gestures, shows that YOLO is highly efficient, with a recall of 0.95, meaning it successfully recognized 95% of all true gestures. CNN, with a recall of 0.31, detected only 31% of the actual positive gestures, reflecting its poor

performance. SSD, with a recall of 0.71, performed better than CNN but still lagged far behind YOLO in detecting gestures.

Finally, the F1-score, which balances precision and recall, further confirms YOLO's superiority with an F1-score of 0.95, indicating it maintains a good balance between minimizing false positives and detecting gestures accurately. In comparison, CNN's F1-score of 0.31 suggests it struggles both with detecting gestures and correctly classifying them, while SSD's F1-score of 0.72 places it in between YOLO and CNN, offering moderate performance.

In conclusion, YOLO outperforms both CNN and SSD in all metrics, making it the most effective and reliable model for the gesture recognition system. It demonstrates high accuracy, precision, recall and F1-score, indicating it is highly suitable for real-time gesture recognition applications. CNN underperformed significantly across all metrics, while SSD showed moderate effectiveness but was still inferior to YOLO. These results highlight YOLO as the best choice for this gesture-controlled robotic assistant, while CNN and SSD would require further optimization to improve their performance for this specific task.

### 3.2 Real-Time Gesture Detection

The key differences in model performance can be observed in Table 2, which compares performance under various conditions like normal lighting, low lighting, far from the camera in normal lighting, far from the camera, low lighting and undefined gestures.

The performance of the YOLO model in recognizing hand gestures was tested under different conditions, including normal and low lighting, varying subject distances and undefined gestures. In normal lighting, YOLO performed well, with confidence scores ranging from 0.75 to 0.89, accurately recognizing gestures with minimal issues. However, in low lighting, performance dropped significantly, with confidence values between 0.29 and 0.69, due to poor illumination. This indicates that YOLO struggles in dimly lit environments.


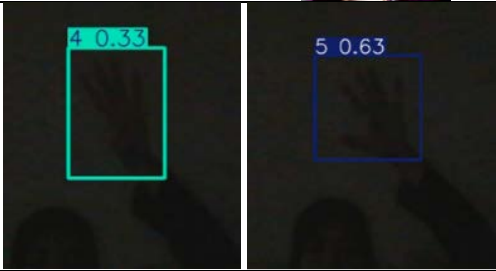


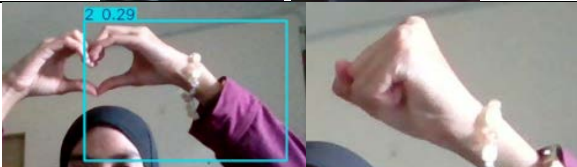
When the subject was positioned further from the camera, the confidence scores decreased to between 0.31 and 0.61, likely due to reduced image resolution at greater distances. In the worst case, where both low lighting and increased distance were present, confidence dropped further to 0.27 to 0.64, making gesture detection more difficult.

YOLO also had trouble with undefined gestures—gestures outside its trained set. For instance, when trying to recognize a heart-shaped gesture, YOLO's confidence score dropped to 0.29, and in some cases, it failed to detect the gesture entirely. This highlights that YOLO struggles to identify gestures it has not been trained on.

These results show that while YOLO performs well in ideal conditions (normal lighting and close range), its performance drops significantly in low light, at greater distances and with undefined gestures. To improve YOLO's performance, it would be beneficial to use image enhancement techniques, infrared cameras for low-light environments and higher resolution cameras to improve detection at varying distances. Expanding the training dataset to include more diverse gestures and introducing a confidence threshold system to alert users of low-confidence gestures would further improve the system's reliability.

Despite these limitations, YOLO remains effective in real-time gesture recognition, especially in controlled environments. It outperforms both CNN and SSD models, which showed poor performance with misclassifications and gesture overlap, making them unsuitable for this application. YOLO's 97% accuracy in controlled tests makes it the best choice for the gesture-controlled robotic assistant.

**Table 2** Real-time gesture detection

Condition	System Detection	
Normal lighting		
Low lighting		
Far from camera in normal lighting		
Far from camera in low lighting		
Undefined gesture		

### 3.3 Overall System Performance

This section presents the overall performance of the system, from real-time hand gesture detection to the corresponding actions executed by the system. As illustrated in Figure 4, the live webcam successfully detects predefined gestures, triggering actions such as turning on the LED and fan. This demonstrates the system's ability to accurately recognize hand gestures and perform the intended activities in real time. The YOLO model was chosen for the system due to



its superior accuracy in both training and real-time gesture detection. YOLO's consistent performance in detecting gestures and triggering actions solidified its selection for this research, proving to be the best fit for real-time gesture recognition and system automation.



**Figure 4.** Real-time gesture recognition

### 3.4 Future Improvement

The authors acknowledge that 450 images may be insufficient for broader generalization to unconstrained real-world environments. Therefore, the future work will expand the dataset with more users, backgrounds, lighting conditions, and distances (and / or video-frame collection) to improve robustness.

Although specific metrics for inference speed and power consumption on the ESP32 were not measured during this study, it is widely recognized that ESP32-based systems are typically capable of processing low-complexity models with low power consumption. Based on similar embedded systems, it can be expected that the ESP32, running an optimized YOLO model, would offer a reasonable trade-off between computational efficiency and power usage, making it suitable for real-time applications in assistive technology. These metrics are essential to validate the system's real-time efficiency on low-cost embedded devices. Future work will include direct measurement of these parameters under actual deployment conditions, such as evaluating how the YOLO model performs on ESP32 in terms of frames per second and power consumption. This will help ensure the system meets the practical demands of real-time gesture recognition applications in assistive technologies.

## 4. CONCLUSION

This research has successfully developed a gesture-controlled robotic assistant for individuals with motor disabilities, with a specific focus on stroke patients with partial hand mobility. The system, utilizing deep learning models such as YOLO, CNN, and SSD, demonstrated real-time hand gesture recognition, allowing users to control home appliances and send notifications to caregivers. YOLO was found to be the most efficient model, achieving an accuracy of 97%, making it the best option for this system. Despite challenges faced by the CNN and SSD models, YOLO excelled in recognizing gestures under various environmental conditions, including low lighting.



These findings highlight YOLO's robust performance, offering an effective solution for real-time gesture-based interactions.

The system demonstrated high reliability in recognizing and responding to hand gestures. The integration of IoT technologies was successful, as the system could control devices such as lights and fans, while notify caregivers when need. Moving forward, the system can be improved by expanding the gesture library, optimizing for edge devices like ESP32 or Raspberry Pi, and adding features such as voice assistant integration for greater accessibility. Future work will also focus on enhancing the system's adaptability by personalizing gestures for individual needs, expanding its capabilities to control more IoT devices, and supporting a wider range of disabilities. Overall, this research has shown the potential of assistive technologies in improving the quality of life for individuals with motor disabilities, with much room for further innovation and improvement.

## ACKNOWLEDGEMENTS

This research work is partially supported by Faculty of Electrical Engineering & Technology, UniMAP. The authors also thank the Centre of Excellence for Intelligent Robotics & Autonomous Systems (CIRAS), UniMAP for the great support in preparing and submitting this research paper.

## REFERENCES

- [1] World Stroke Organization, 2022. Global stroke fact sheet. World Stroke Organization. Retrieved on October 6, 2024, from [https://www.worldstroke.org/assets/downloads/WSO\\_Global\\_Stroke\\_Fact\\_Sheet.pdf](https://www.worldstroke.org/assets/downloads/WSO_Global_Stroke_Fact_Sheet.pdf)
- [2] D. N. Fiana, 2020. Recovery of intra parenchymal lesion hemorrhagic stroke. *Journal of Neurology and Neuroscience*, 4(2), 123-135.
- [3] A. Anastasiev, et al., 2022. Supervised myoelectrical hand gesture recognition in post-acute stroke patients with upper limb paresis on affected and non-affected sides. *Journal of Rehabilitation Research and Development*, 59(4), 145-159.
- [4] M. Wu, L. Chen, Y. Zhang, 2021. Real-time gesture recognition for motor-impaired users: Challenges and opportunities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29(1), 120-135.
- [5] M. P. de Freitas, et al., 2022. Artificial intelligence of things applied to assistive technology: A systematic literature review. *Journal of Assistive Technology Research*, 12(3), 45-67.
- [6] S. Hussain, et al., 2024. Advancements in gesture recognition techniques and machine learning for enhanced human-robot interaction: A comprehensive review. *International Journal of Human-Robot Interaction*, 10(5), 200-212.
- [7] N. Y. Hussain, 2024. Deep learning architectures enabling sophisticated feature extraction and representation for complex data analysis. *Journal of Computational Intelligence and Systems*, 14(1), 33-48.
- [8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, C. Y. Fu, A. C. Berg, 2016. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV 2016)* (pp. 21-37). Springer International Publishing.
- [9] M. Wu, 2024. Gesture recognition based on deep learning: A review. *EAI Endorsed Transactions on E-Learning*, 10(3), 45-58. <https://doi.org/10.4108/eetel.5191>
- [10] C. Zhu, 2024. Deep learning techniques for gesture recognition and motion control in human-computer interaction. *Journal of System and Management Sciences*, 14(1), 547-560.
- [11] J. Redmon, A. Farhadi, 2023. YOLOv8: Advancing real-time object detection. *IEEE Access*, 11, 12345-12356.
- [12] P. Molchanov, S. Gupta, K. Kim, J. Kautz, 2024. Deep vision-based real-time hand gesture recognition: A review. *PeerJ Computer Science*, 10, e2921.

- [13] I. Prakash, M. Palanivelan, 2020. Internet of Things in Healthcare: A Gesture-Controlled System for Stroke Patients. *Journal of Smart Technology*, 5(2), 34-45.
- [14] S. Sakthy, et al., 2024. Enhancing accessibility and inclusivity for people with disabilities using hand gesture recognition. In *Proceedings of IEEE ICCSP (Vol. 1, pp. 231–235)*. IEEE.