



City Research Online

City St George's, University of London

Citation: Jadeja, Y., Shafik, M., Wood, P. & Makkar, A. (2025). Enhancing Healthcare Assistance with a Self-Learning Robotics System: A Deep Imitation Learning-Based Solution. *Electronics*, 14(14), 2823. doi: 10.3390/electronics14142823

This is the published version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/35597/>

Link to published version: <https://doi.org/10.3390/electronics14142823>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

Article

Enhancing Healthcare Assistance with a Self-Learning Robotics System: A Deep Imitation Learning-Based Solution

Yagna Jadeja , Mahmoud Shafik , Paul Wood  and Aisha Makkar

College of Science and Engineering, University of Derby, Markeaton Street, Derby DE22 3AW, UK; m.shafik@derby.ac.uk (M.S.); p.wood7@derby.ac.uk (P.W.); a.makkar@derby.ac.uk (A.M.)

* Correspondence: yagnatechnologies@outlook.com; Tel.: +44-742-455-9512

Abstract

This paper presents a Self-Learning Robotic System (SLRS) for healthcare assistance using Deep Imitation Learning (DIL). The proposed SLRS solution can observe and replicate human demonstrations, thereby acquiring complex skills without the need for explicit task-specific programming. It incorporates modular components for perception (i.e., advanced computer vision methodologies), actuation (i.e., dynamic interaction with patients and healthcare professionals in real time), and learning. The innovative approach of implementing a hybrid model approach (i.e., deep imitation learning and pose estimation algorithms) facilitates autonomous learning and adaptive task execution. The environmental awareness and responsiveness were also enhanced using both a Convolutional Neural Network (CNN)-based object detection mechanism using YOLOv8 (i.e., with 94.3% accuracy and 18.7 rms latency) and pose estimation algorithms, alongside a MediaPipe and Long Short-Term Memory (LSTM) framework for human action recognition. The developed solution was tested and validated in healthcare, with the aim to overcome some of the current challenges, such as workforce shortages, ageing populations, and the rising prevalence of chronic diseases. The CAD simulation, validation, and verification tested functions (i.e., assistive functions, interactive scenarios, and object manipulation) of the system demonstrated the robot's adaptability and operational efficiency, achieving an 87.3% task completion success rate and over 85% grasp success rate. This approach highlights the potential use of an SLRS for healthcare assistance. Further work will be undertaken in hospitals, care homes, and rehabilitation centre environments to generate complete holistic datasets to confirm the system's reliability and efficiency.

Keywords: self-learning robotics; healthcare assistance; imitation learning; YOLOv8; pose estimation algorithms; MediaPipe; human–robot interaction



Received: 13 May 2025
Revised: 30 June 2025
Accepted: 2 July 2025
Published: 14 July 2025

Citation: Jadeja, Y.; Shafik, M.; Wood, P.; Makkar, A. Enhancing Healthcare Assistance with a Self-Learning Robotics System: A Deep Imitation Learning-Based Solution. *Electronics* **2025**, *14*, 2823. <https://doi.org/10.3390/electronics14142823>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial Intelligence (AI) and robotics have profoundly influenced the healthcare sector, as it drives innovation in patient support and service delivery [1,2]. One of the most promising developments within clinical environments is the emergence of assistant robots, which are designed to aid healthcare professionals, enhance patient care, and streamline operations [3,4]. Conventional robotic systems are constrained by their dependence on traditional programming methods and rigid task structures. This limits their effectiveness in dynamic and unpredictable healthcare settings, where adaptability and operational efficiency are essential [5]. To address these challenges, recent research has focused on Self-Learning Robotics Systems (SLRSs) that can autonomously acquire new

skills through demonstration and observation [6,7]. The current implementation is typically limited to predefined tasks such as object handovers with restricted object diversity and no personalisation based on patient-specific needs [8].

This research presents an SLRS for healthcare assistance that implements a hybrid model approach (i.e., deep imitation learning and pose estimation algorithms) that facilitates autonomous learning and adaptive task execution. Leveraging Deep Imitation Learning (DIL) and computer vision frameworks has also enabled context-aware robotic behaviour. In this context, deep imitation learning is an advanced learning process, wherein a self-learning robotic system observes actions demonstrated by multiple human experts, analyses these demonstrations through policy learning, and autonomously optimises, selects, and replicates the most optimal action. This approach allows for the robot to not only imitate human behaviour but also adapt it contextually, thereby enabling the efficient execution of complex tasks within dynamic and unpredictable environments. This means that, by incorporating technologies such as CNN-based object detection (YOLOv8), pose estimation, and LSTM-based gesture recognition, the system can autonomously perform assistive tasks typical to the ones in healthcare environments. This research also presents the initial evaluation of the applicability and effectiveness of an SLRS using both a CAD simulation model and real-world scenarios. During this assessment stage, special attention is given to the system architecture, training procedures, and key challenges encountered during deployment [9,10].

2. Related Work

2.1. Healthcare Robotics: Current Trends and Limitations

AI and robotics have rapidly evolved, offering promising solutions to meet the growing demands for enhanced services and patient care, especially in healthcare systems, as this evolution could be a good platform solution to some of the current healthcare challenges, i.e., workforce shortages, ageing populations, and the increasing prevalence of chronic diseases. Robotic systems have been employed in various healthcare applications, including surgical assistance, rehabilitation, elder care, and logistics [3]. For instance, robots such as the Da Vinci Surgical System and Robear exemplify robotics solutions used to support clinical tasks. However, these systems often rely heavily on pre-programmed routines and lack the adaptability required for dynamic environments and personalised patient needs [4,6]. This inadaptability limits their effectiveness, especially in real-world clinical settings where patient behaviour and care contexts frequently vary [10]. These limitations include rigid and task-specific programming, which cannot generalise to handle diverse scenarios or patient behaviours [11]. This shows the needs for a new generation of robotic systems such as an SLRS that are capable of learning and adapting independently.

2.2. Self-Learning Robotics in Healthcare

The foundational theories of an SLRS are built upon Reinforcement Learning (RL) and Imitation Learning (IL) paradigms [12,13]. An SLRS integrates sensory inputs with real-time feedback and learning algorithms to iteratively refine its operational capabilities [14]. For example, Jadeja et al. (2022) presented a self-learning robotic system that effectively replicated human movements using imitation learning frameworks, illustrating its potential for healthcare applications [13,15]. Thus, an SLRS offers abilities to enhance task performance through observation, environmental interaction, and iterative learning. The core advantages lie in their capacity to adapt to individual patient needs and evolving medical protocols. Therefore, recent research shows the success of using these systems in practical applications as a healthcare assistant, i.e., patient handling, medication dispensing, and surgical assistance.

2.3. Deep Imitation Learning: The Backbone of Adaptability

Imitation learning enables an SLRS to acquire skills from expert demonstrations instead of relying on explicit programming. This is particularly valuable in healthcare settings, where tasks involve subtle human interactions, such as therapeutic assistance or complex procedural support [16]. Deep Imitation Learning (DIL) advances traditional IL by incorporating deep neural architectures, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models, which process high-dimensional visual and temporal data [17]. CNNs extract visual features, such as object boundaries and textures, while LSTMs model temporal dependencies in human motion. These capabilities allow for robust mapping from video and sensor data to meaningful robotic actions. Techniques like behavioural cloning, Inverse Reinforcement Learning (IRL), and Generative Adversarial Imitation Learning (GAIL) have shown effectiveness in healthcare robotics—even in scenarios with limited training datasets—by improving behavioural fidelity and adaptability [18].

2.4. Applications and Efficacy of Self-Learning Robots in Healthcare

DIL-trained robotic assistants have been successfully applied in caregiving tasks such as object handovers, medication delivery, and mobility support for patients with restricted function [19]. These systems demonstrate adaptability to diverse patient requirements through continuous observation and learning, supporting their use in hospital wards and care facilities [15]. The integration of computer vision with deep learning further enhances robotic perception and decision-making. Controlled experimental settings have shown that SLRSs achieve strong performance in real-time decision-making tasks such as object tracking, trajectory planning, and corrective action in response to dynamic changes [14].

2.5. Research Gaps and Challenges

Even though progress has been made in the SLRS areas of development and deployment, there are still several challenges that remain. These are the following:

- **Lack of a Dataset:** Collecting and annotating healthcare demonstrations are difficult due to privacy concerns, data ownership, and variation across institutions [20].
- **General Limitations:** Systems trained on narrow demonstration sets often perform poorly in unfamiliar environments or with unexpected task variations [17].
- **Human–Robot Interaction (HRI):** Natural, intuitive communication between robots and patients or clinicians is still underdeveloped, despite its critical importance [1].
- **Safety and Ethics:** Ensuring fail-safe operation in sensitive contexts remains a major concern, particularly when an SLRS interacts directly with patients [6].

The fact is, addressing these challenges demands the integration of advanced policy-learning techniques and closer interdisciplinary collaboration among engineers, clinicians, and ethicists. This is to ensure that, as SLRS systems are deployed in real-world care environments, both technical robustness and ethical compliance are fully considered. These gaps highlight the needs for modular and learning-driven assistive robots that are capable of real-time perception, adaptive behaviour, and ethical operation. This is where this potential research programme with its aim and objectives comes in.

3. SLRS Architecture, Functioning Principles, Control Blocks, and Implementation

3.1. SLRS System Architecture

Figure 1 illustrates the system architecture of the Self-Learning Robotic System (SLRS). It is a modular system framework that integrates industrial-grade robotics with Deep Imitation Learning (DIL). This is to enable autonomous replication of complex tasks demon-

strated by humans in healthcare settings [15]. The system is built upon three core modules: perception, learning, and action.

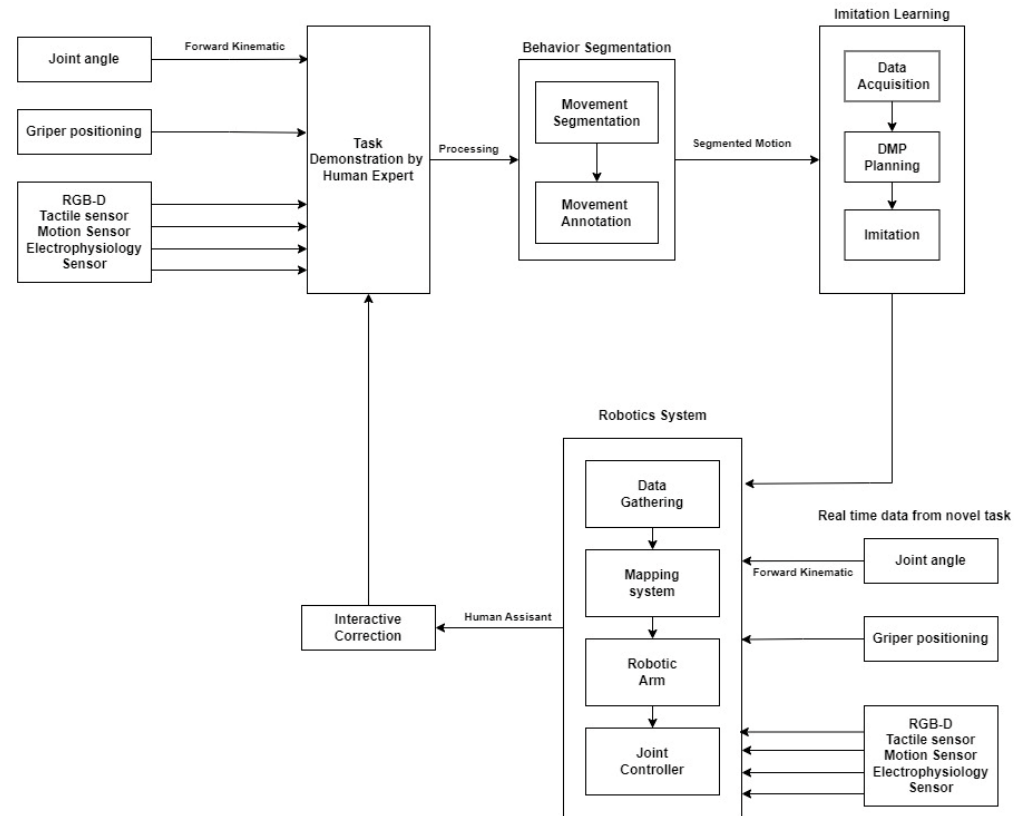


Figure 1. Architecture of a self-learning collaborative robot system using imitation learning [15].

The perception layer combines RGB-D sensors, stereo vision modules, and standard web cameras. This is to acquire high-dimensional visual data, including object locations and human skeletal poses [2,5,10]. These feed into a policy-learning module, which translates the sensory data into actionable robotic movements [7,9]. The action process is at the hardware level, as the SLRS employs a six Degrees-Of-Freedom (DOF) robotic arm, the MyCobot 280 Jetson Nano. This is controlled by an onboard Jetson Nano processor. This compact, power-efficient platform enables real-time inference for deep learning models, making it suitable for use in clinical settings (i.e., hospitals, care homes, and rehabilitation centres) [1,4,19]. The SLRS integrates YOLOv8 for object detection (operating at 91 frames per second) and a MediaPipe + LSTM pipeline for human action recognition to ensure robust perception and responsive interaction. This dual-modality setup allows for the robot to interpret spatiotemporal human behaviour by analysing key point trajectories over time, thus facilitating smooth and context-aware interaction [1,6]. These three core modules (i.e., perception, learning, and action) are key features of the SLRS that show how it is Programmed via Demonstration (PvD). The robot records these task sequences and learns to generalise them, which eliminates the need for technical programming expertise [16]. This means that it allows for healthcare professionals to teach the robot tasks simply by performing them. To promote task flexibility and detail, complex activities are decomposed into motion primitives—basic, reusable actions that can be recombined for different tasks. This modularity architecture enhances the SLRS’s adaptability across various scenarios, from physical assistance to medication delivery [10,20–22], as it offers three interdependent functional modules:

- **Perception Module:** Utilises multimodal sensors to interpret environmental and spatial cues [2,5].

- Learning Module: Applies imitation learning algorithms, including Dynamic Movement Primitives (DMPs), to convert observational data into robotic actions [9,12,16].
- Action Module: Executes learned behaviours via low-level control and feedback mechanisms for real-time refinement [13,17].

The training data were split 70:30 between training and testing, using a sampling rate of 10 Hz to synchronise 3D skeletal key points with RGB frames. This structure ensures effective alignment between observed behaviours and the system's decision-making logic. The control logic of the SLRS is structured into two primary phases, learning and reproducing, as illustrated in Figure 2. In the learning phase, expert demonstrations are recorded and split into three data streams: Follow Path: Captures spatial trajectory and motion paths, Actuation Control: Records force and joint control signals required to execute the task, and De-actuation Control: Identifies motion completion and transition points. These elements are integrated into a learning-based controller that generalises the demonstration data and generates a regulation model for downstream execution. In the Reproducing Phase, the robot uses the learned policy to determine motion parameters such as joint angles, velocities, and forces in response to its current sensory state. The robot executes these actions, transitions to the next task phase, and continuously monitors feedback to refine its performance. This two-phase control system ensures the SLRS can learn and execute complex sequences in real-time, adapt to new inputs, and provide responsive support in diverse healthcare tasks. By combining perception, learning, and actuation into a cohesive framework, the SLRS operates not just as a manipulator but also as a healthcare assistant capable of safe and intelligent support across a range of healthcare contexts [3,8,14,19].

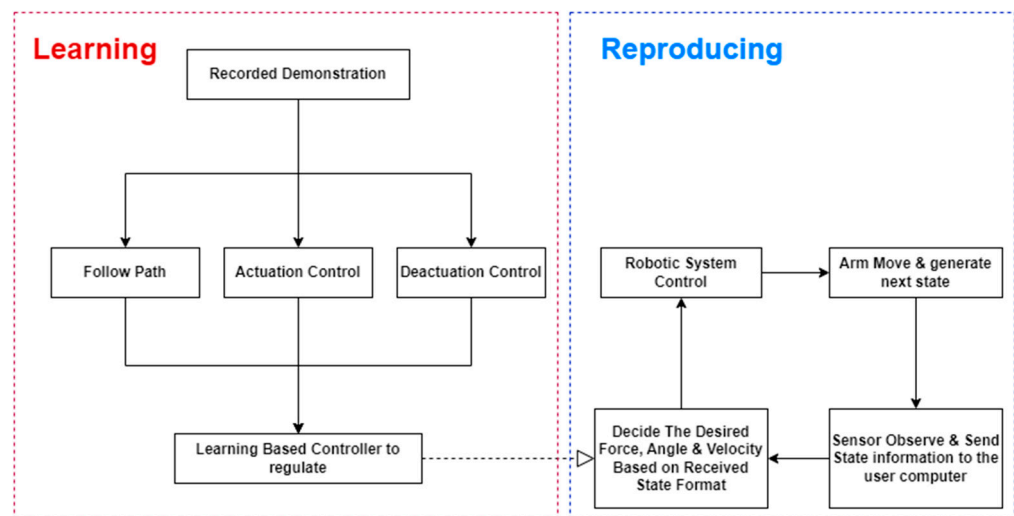


Figure 2. Control blocks of the self-learning collaborative robot system using imitation learning.

3.2. Learning Framework and Policy Acquisition

The SLRS is built upon a multi-stage learning pipeline that integrates deep learning, computer vision, behavioural segmentation, and real-time control [7,8,10,23]. At the core of this framework is imitation learning, which treats the learning process as a Markov Decision Process (MDP) but without relying on explicitly defined reward functions. Instead, the system learns optimal policies directly from expert demonstrations [9,16,24]. An overview of this learning framework is shown in Figure 3. The figure shows how the learning framework links data collection and perception to policy acquisition and execution. The robot begins by processing raw sensory inputs, extracting relevant spatial and behavioural features. These features inform the policy learning module, which is trained to replicate expert behaviour through learned control actions.

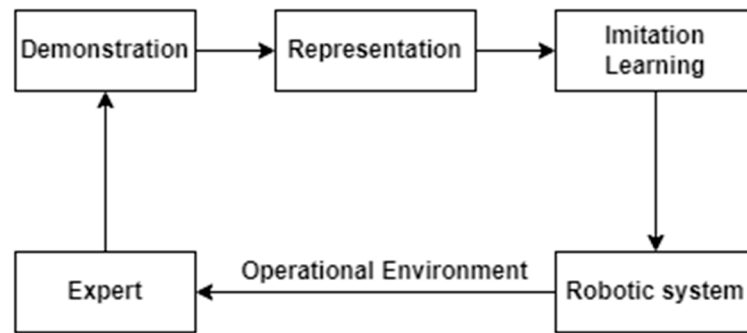


Figure 3. Learning pipeline of SLRS using imitation learning [7].

3.2.1. Data Acquisition and Preprocessing

During demonstration sessions, the SLRS collects synchronised RGB video and depth data using stereo vision and RGB-D cameras [2,5,10]. These inputs allow for the following:

- Object detection via YOLOv8 [25];
- Human skeletal pose extraction using MediaPipe [22,26];
- Environmental context awareness, such as object affordances and workspace boundaries [23].

As illustrated in Figure 4, human experts perform tasks (e.g., handovers), which are captured and parsed into structured input suitable for robot imitation. To ensure time-aligned learning, visual and motion data are synchronised at 10 Hz, associating environmental changes with observed human actions [27–30].

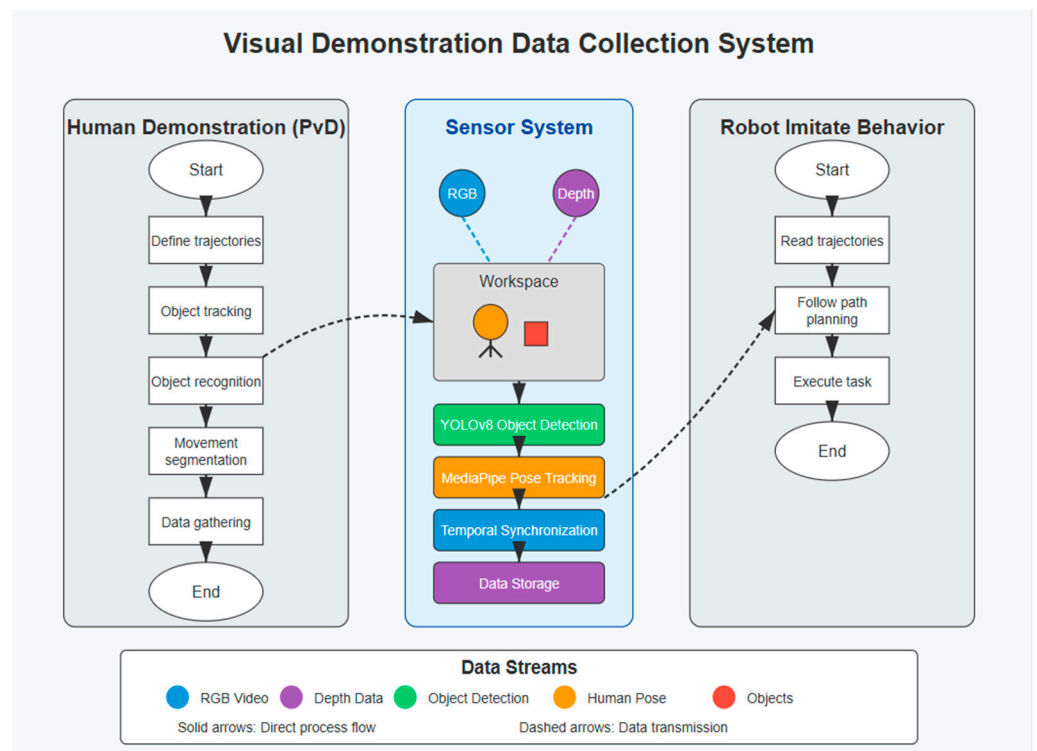


Figure 4. Visual demonstration data collection system: demonstration, data collection, processing, and imitation.

The collected demonstrations are then segmented into behaviour units using a three-stage preprocessing approach:

- Temporal segmentation isolates atomic actions such as reach, grasp, lift, and release [16,30].
- Pose tracking extracts 3D joint coordinates and movement patterns of the demonstrator [22,26].
- Trajectory encoding represents the demonstrator's hand or tool movement as parameterised curves, enabling precise robotic reproduction [7,23,27,31].

This behavioural abstraction enables the creation of a library of motion primitives, which can be recombined for novel tasks or simplified across different environments [12,30,32].

3.2.2. The Policy Learning Framework

The SLRS employs a hybrid policy learning strategy that combines structured behavioural segmentation with two complementary imitation learning methods: Behavioural Cloning (BC) and Inverse Reinforcement Learning (IRL). These are supported by perception algorithms for real-time object and action recognition. To enable effective training, two custom-designed preprocessing algorithms are applied.

Algorithm 1, behavioural segmentation algorithm, processes a set of expert demonstrations $D = \{D_1, D_2, \dots, D_n\}$, where each demonstration contains spatiotemporal movement data. For each demonstration,

- Feature trajectories are extracted.
- An affinity matrix is computed based on motion similarity.
- Spectral clustering is applied to group motion patterns into clusters.
- Segment boundaries are determined based on changes in motion labels.
- Valid segments are grouped using a similarity threshold θ to form higher-level motion categories.

These segmented actions, such as reaching, grasping, and handing over, serve as discrete learning units for policy training.

This process ensures that demonstrations are converted into semantically meaningful and reusable sub-tasks.

Algorithm 2, object detection and colour recognition, forms the core of the SLRS visual perception module. It begins with object detection using deep feature extraction and applies secondary validation based on colour data.

- Deep features are extracted using a ResNet-50 backbone and CNN layers [33–35].
- Classification is performed via SoftMax.
- If confidence is low, feature vectors are compared using Euclidean distance.
- RGB values are also used to confirm object identity.

This dual pathway ensures robust detection in real-world environments with visual noise or occlusion.

The segmented demonstrations and recognised objects are used as input to two learning frameworks as shown in Table 1:

- Behavioural Cloning (BC) maps sensory input directly into actions through supervised learning. It is fast but sensitive to noise and lacks exploration [8,16].
- Inverse Reinforcement Learning (IRL) infers the underlying reward structure and then optimises a policy accordingly. It is more robust but computationally heavier [9,13,24,30].

Algorithm 1: Behavioural Segmentation Algorithm

```

start
Read the set of Demonstrations  $D = \{D_1, D_2, \dots, D_n\}$ ;
for each Demonstration  $D_i$  in  $D$  do
Extract Feature Trajectories  $T_i$  from  $D_i$ ;
Compute Affinity Matrix  $A_i$  using motion similarity between  $T_i$ ;
Apply Spectral Clustering on  $A_i$  to obtain Cluster Labels:  $C_i$ ;
Initialize Segment Boundaries =  $\emptyset$ ;
Set  $PrevLabel = C_i[0]$ ,  $StartIndex = 0$ ;
for  $j = 1$  to  $length(C_i) - 1$  do
if
 $C_i[j] \neq PrevLabel$  and  $(j - StartIndex) \geq minLen$  then
Append  $j$  to Segment Boundaries;
Set  $StartIndex = j$ ;
Set  $PrevLabel = C_i[j]$ ;
else if
Slice Demonstration  $D_i$  using Segment Boundaries;
for each Segment  $s$  in Sliced  $D_i$  do
else if
Segment  $s$  is Valid (based on  $\theta$  and motion consistency) then
Add  $s$  to Segment List;
for each pair of Segments  $(s_i, s_j)$  in Segment List do
else
Motion Similarity( $s_i, s_j$ )  $\geq \theta$  then
Merge  $s_i$  and  $s_j$  into same Motion Group;
end
end
end
end
Output all Behavioural Segments;
end

```

These learning methods form a robust hybrid framework capable of both speedy deployment and long-term behavioural adaptation.

Figure 5 illustrates the three policy representations underpinning these approaches.

- Symbolic representations encode abstract logical task steps, useful for high-level planning [11,12];
- Trajectory-based representations retain motion fidelity through continuous path encoding [7,30];
- Action–state mappings directly link sensor input to motor responses, enabling real-time reactivity in dynamic environments [23].

These representation modes enable the robot to operate both reactively and strategically across use cases.

The training of the policy models uses Stochastic Gradient Descent (SGD). SGD is particularly suitable for imitation learning due to its scalability, efficiency with large datasets, and suitability for online or incremental learning [24,36]. It iteratively minimises the error between predicted and demonstrated actions. It allows for the SLRS to continually refine its learned policies as additional demonstrations are collected over time, thereby

supporting personalisation and domain adaptation [8,24]. Figure 6 shows the process where policy accuracy improves with training progression.

Algorithm 2: Object Detection and Colour Recognition Algorithm

start

Read the Object Image O_i from input Device;

Apply the pre-processing Algorithm;

Apply Resnet-50 on the Object Image: $F_{resnet}(O_i)$;

Apply the ReLU Activation Function: $F_{ReLU}(x) = \max(0, x)$;

Apply the CNN Algorithm: $F_{CNN}(F_{ReLU}(F_{resnet}(O_i)))$;

Apply SoftMax Function: $F_{softmax}(x) = \frac{e^x}{\sum_j e^x}$;

Extract the RGB features of Object Image: $RGB_features(O_i)$;

if Object Identified then

Print "Pick Object"

else if

for $I = 1$ to N do // N is the number of DS images;

Read the Datasets (DS) image features: $DS_features$;

Calculate the distance between the two feature vectors;

$$Distance(O_i, DS_features) = \sqrt{\sum_{k=1}^n (O_i[k] - DS_features[k])^2}$$

else if (Classified Threshold Satisfied) then

Print "Place Object to Destination;"

else

Print "Object not Identified with DS Images;"

end

end

end

end

end

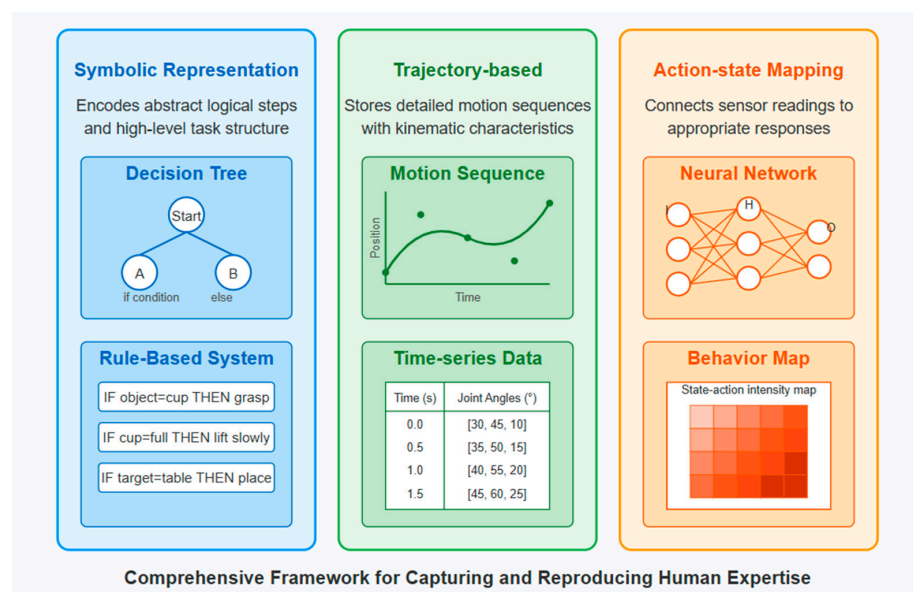


Figure 5. Imitation learning and policy representation types (symbolic (logic-based), trajectory (motion-based), and action–state (sensor-based) policies offer layered learning strategies aligned with robotic control architecture).

Table 1. Comparison of Behavioural Cloning (BC) and Inverse Reinforcement Learning (IRL).

Aspect	Behaviour Cloning (BC)	Inverse Reinforcement Learning (IRL)
Definition	Learns a policy by directly mimicking expert demonstrations	Infers the reward function underlying expert behaviour
Learning Target	Policy function	Reward function (then derive policy via reinforcement learning)
Supervision Type	Supervised learning	Combination of supervised + reinforcement learning
Data Requirement	Requires a large and diverse set of expert trajectories	Requires fewer demonstrations but needs exploration capability
Generalisation	Poor generalisation outside seen states	Better generalisation by learning the underlying intent (reward)
Robustness to Noise	Sensitive to imperfect demonstrations	More robust to suboptimal or noisy expert actions
Exploration	No exploration; purely imitation-based	Includes exploration as part of reward function optimisation
Implementation	Simpler and faster to implement	Computationally expensive and complex
Advantages	Easy to implement, Fast training	Learns the intent of the expert, Better long-term behaviour
Limitations	Prone to compounding errors, Does not learn intent	Complex optimisation, Needs RL to derive policy from reward
Typical Use Cases	Autonomous driving, robotics with ample data	Strategic planning, robotics with sparse demonstrations

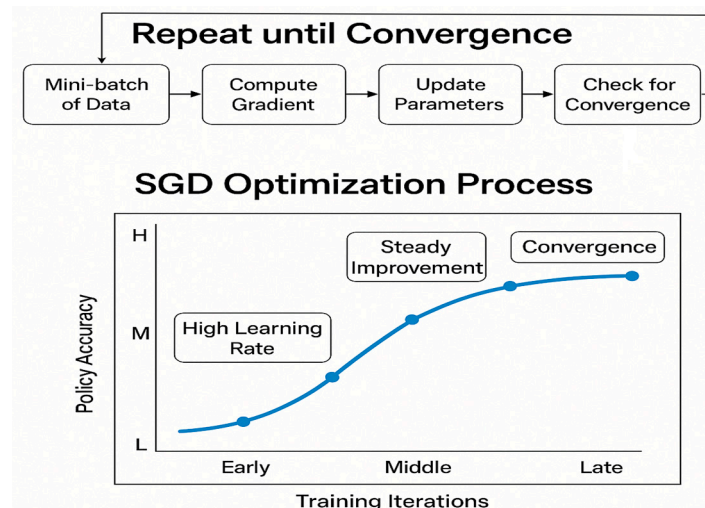


Figure 6. SGD optimization process improves policy prediction accuracy with each epoch, supporting continuous model refinement in large datasets.

3.2.3. Context-Aware Perception

The combination of structured behavioural segmentation, robust perception, and deep imitation learning strategies equips the SLRS with task reliability, adaptability, and the ability to operate effectively and safely in the unstructured conditions [1,4,37]. To support robust policy learning, the SLRS incorporates a tightly integrated perception pipeline, which captures real-time visual and behavioural information. It also combines object recognition, human pose estimation, and action detection. This is to ensure temporal alignment of visual features with the robot’s learning process.

- **Object Detection:** The system uses the YOLOv8 algorithm, fine-tuned on a domain-specific healthcare dataset prepared via Roboflow [21,25]. YOLOv8 was selected for its speed (91 FPS) and high detection accuracy, particularly under dynamic conditions common in clinical environments. Detected objects include assistive items such as bananas, toothbrushes, and medicine bottles, which are classified and localised in the robot's workspace in real time.
- **Pose Estimation:** To detect and classify human hand pose, a MediaPipe-LSTM framework is used. MediaPipe extracts 3D skeletal key points, while LSTM networks model the temporal dynamics of these key points to classify gesture sequences. This enables the robot to interpret motion over time, for example, distinguishing between reaching and giving gestures. The system processes 1662 pose features per second (55 key points \times 30 FPS), enabling fine-grained motion tracking and real-time interpretation of human intent.
- **Temporal Synchronisation and Feature Alignment:** The outputs of object and action recognition pipelines are temporally synchronised to ensure consistent alignment between observed behaviours and environmental states. This is essential for policy learning, where the robot must understand both what action is required and how it is executed within a specific context.
- **System Performance Evaluation:** Figure 7 presents a performance evaluation of the perception system under various lighting conditions and object occlusions. Detection accuracy and frame processing time were benchmarked for three representative object classes:

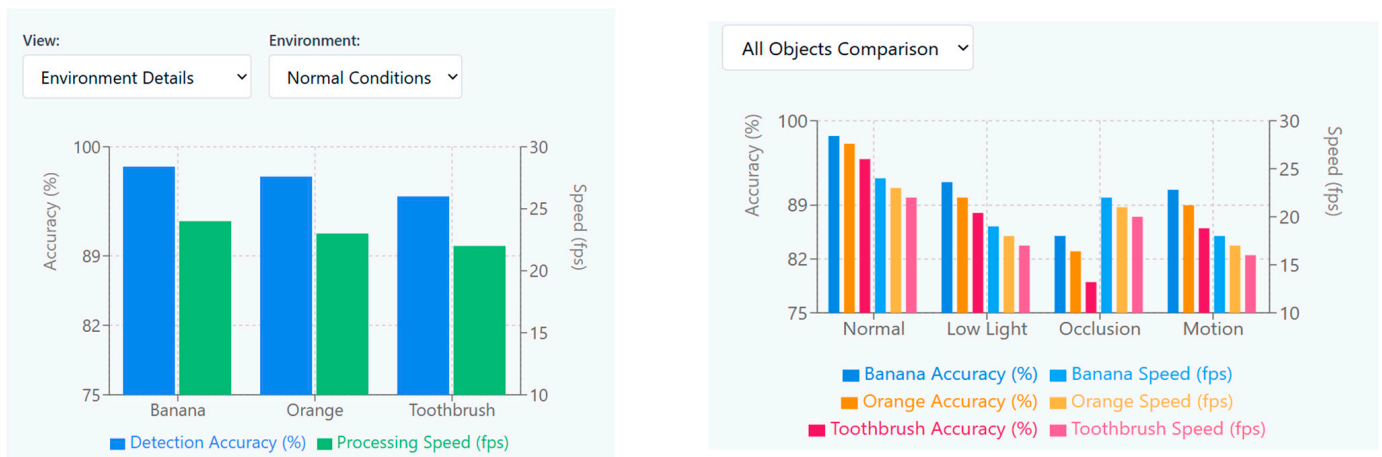


Figure 7. Perception system performance under different environmental conditions. Accuracy and speed were recorded for objects, including banana, orange, and toothbrush, with obstruction and lighting variations. Evaluation of all objects maintained >75% accuracy.

- **Banana:** Highest accuracy (up to 9%), even under occlusion.
- **Orange:** Moderate accuracy across conditions.
- **Toothbrush:** Lower accuracy in cluttered scenes, dropping to 79%.

A strong correlation was observed between detection accuracy and environmental complexity, confirming the robustness of the system's multimodal perception pipeline. This layered perception learning architecture allows for the SLRS to interpret the following:

- What actions to perform (object and goal recognition);
- How they should be performed (human gestures and motion recognition).

By integrating perception and behaviour analysis, the system is capable of replicating human-like behaviour with high temporal precision and contextual awareness. It indicates

the practicality of the system deployment in complex environments, such as hospitals, care homes, and rehabilitation centres. This is where object identity and human intent must be accurately recognised despite environmental variability [21,22,27,38,39].

3.3. Implementation and System Integration

The SLRS was implemented using a cost-effective hardware and software configuration. This is designed to meet performance, safety, and usability requirements in typical healthcare environments. The system comprises a physical robot platform, an integrated software stack, and a hierarchical control architecture. Together, these support real-time task execution, training, and adaptation [28].

3.3.1. Hardware Platform

The SLRS uses the MyCobot 280 Jetson Nano robotic arm and a compact, six degrees-of-freedom (DOF) manipulator. This is designed for close-proximity human–robot interaction. This lightweight platform is particularly suited for fine motor tasks, such as tool handovers or delicate object retrieval, commonly required in clinical and home-care settings [7,40,41].

Key Hardware Components:

- Manipulator: MyCobot 280, with six DOF for flexible articulation.
- Computing Unit: Jetson Nano—selected for its real-time inference capability and low power consumption [21,25,41].
- Sensing Suite: Stereo vision cameras, RGB-D sensors for 3D depth mapping, and webcams for high-resolution RGB input.

The stereo vision system comprises dual cameras and depth sensors, utilising stereo-matching algorithms and K-means clustering for spatial analysis [10,42]. This setup enhances environmental awareness by enabling 3D object localisation, thereby reducing the likelihood of collision or misplacement during object interaction [10,43]. The system computing backbone is powered by the Jetson Nano [44], which provides sufficient real-time processing capability for deep learning inference tasks while maintaining energy efficiency and a compact form factor suitable for integration into mobile assistive platforms [21,25,45].

3.3.2. Software Platform

The SLRS framework is built for modularity, scalability, and real-time performance. The core framework is based on the Robot Operating System (ROS2), which facilitates sensor integration, motion planning (via MoveIt!) [46], real-time task execution, and system synchronisation [11,47,48]. Additional components include the following:

- Simulation and Control Logic: Developed in MATLAB/Simulink 2022b, enabling multi-body dynamic simulation, PID tuning, and safety validation before real-world deployment [6,13,20].
- Deep Learning Pipelines:
 - YOLOv8 for object detection;
 - MediaPipe + LSTM for human gesture/action recognition [49];
 - Implemented in TensorFlow, optimised for edge inference on Jetson Nano [21,22,26,36].

Figure 8 shows an overview of the SLRS software stack, showing the flow from sensor input, through middleware, to policy execution.

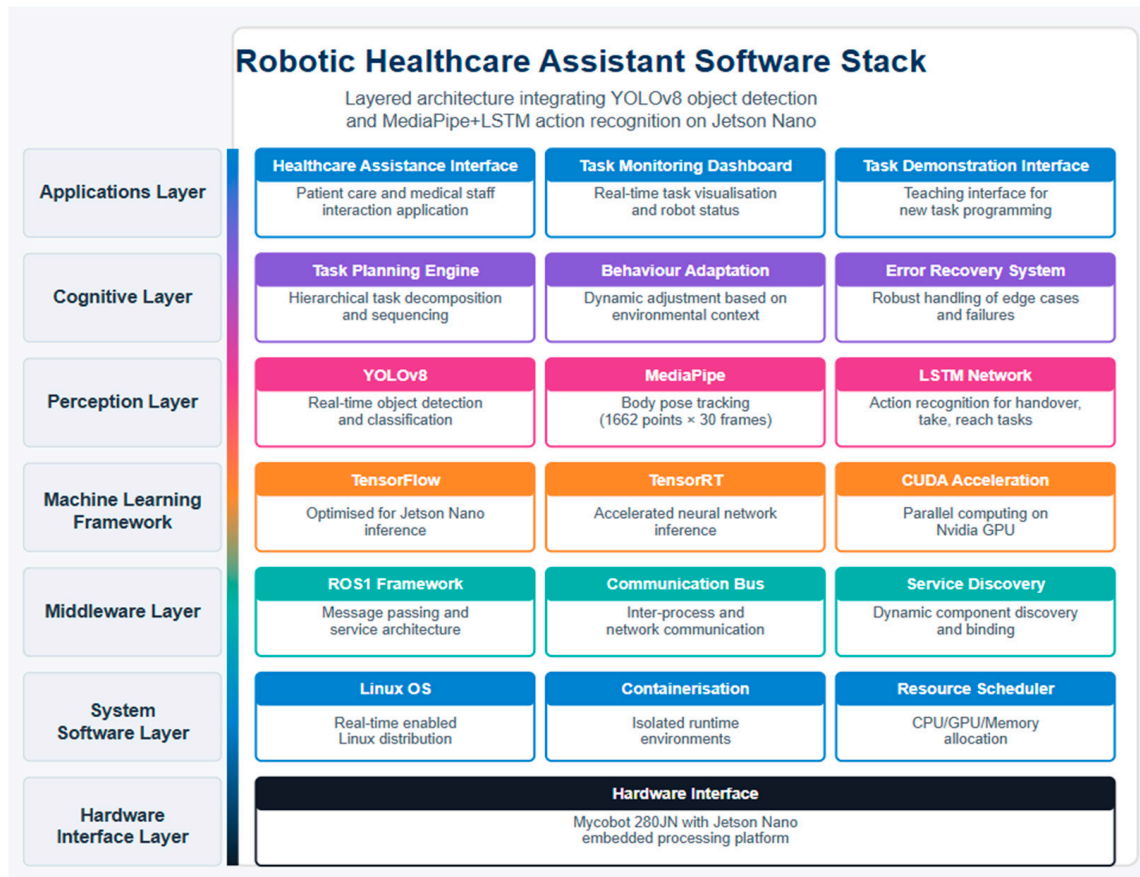


Figure 8. Diagram of software stack.

3.3.3. Control System Implementation

- High level controller: The SLRS control system adopts a hierarchical structure comprising three levels. At the high level, task planning is informed by detected objects and recognised actions, allowing for the robot to determine the optimal sequence of operation to achieve the desired task outcome [3,12,38].
- Mid-level controller: This handles the trajectory generation, combining Stochastic Gradient Descent (SGD) for initial optimisation with Sequential Quadratic Programming (SQP) for fine-tuned trajectory refinement [24,36,43]. This hybrid approach produces smooth and biologically plausible motion patterns while respecting the robotic system's kinematic constraints [13].
- Low level controller: This is a PID controller that regulates the robotic arm's joint positions and velocities, ensuring accurate execution of planned trajectories [6,11,20]. This classical control foundation, while complemented by learning-based approaches, provides deterministic safety and robustness under variable environmental conditions. A multimodal feedback loop is implemented, enabling auto-correction based on real-time visual feedback to recover from minor failures without requiring human intervention [22,27,38].

Figure 9 illustrates the closed-loop interaction between the perception module, learning policies, and the control hierarchy. This integrated control framework enables efficient and reliable translation of learned policies into physical actions while satisfying healthcare-specific safety requirements.

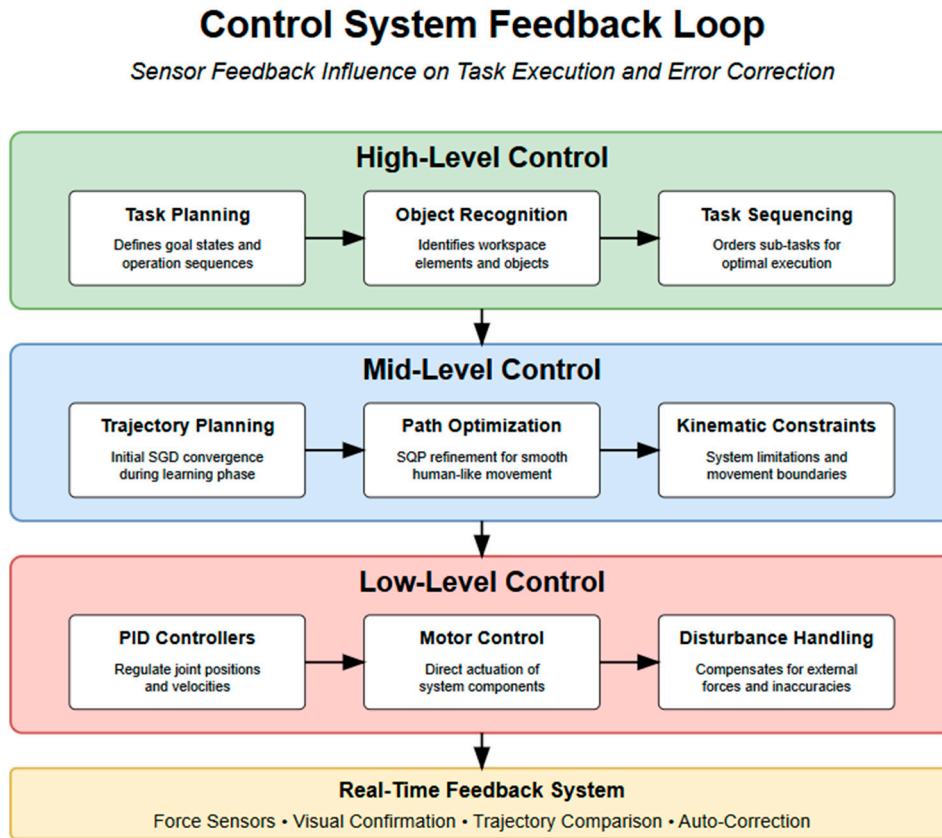


Figure 9. Closed-loop interaction between the perception module, learning policies, and the control hierarchy.

3.4. Evaluation Methodology and Experimental Design

3.4.1. Training and Testing Protocol

Training data was collected using the multimodal perception setup described in the system architecture in Section 3.1. Each demonstration sequence included the following:

- RGB-D video streams capturing human motion and object states;
- 3D skeletal pose data from MediaPipe;
- Annotated task phases, including reach, grasp, transfer, and release.

The data was synchronised at 10 Hz, with segmentation based on behavioural units generated by Algorithm 1. Table 2 summarises the dataset structure.

Table 2. Summary of data collection statistics for training and testing.

Category	Value
Expert Demonstration Videos	20
Total Images	1500
Task Types	3 (Fruits Handover, Vegetables Handover, Daily Use Items Handover)
Example Items	Apples, Oranges, Bananas, Carrots, Cucumbers, Toothbrushes, etc.
Total Recorded Data	8.5 h
Number of Expert Subjects	5
Testing Environments	2 (Laboratory Setting, Home Environment)
Robot Platform Used	1 (Assistive Robotic Arm)

The dataset was split into 70% for training and 30% for testing, ensuring each sub-task type appeared in both partitions. Demonstrations were performed by different human subjects to introduce variability in style, speed, and hand orientation supporting the system's requirements.

3.4.2. Model Training Procedure

Two parallel models were trained:

- Behavioural Cloning (BC) policy using supervised learning;
- Inverse Reinforcement Learning (IRL) policy using learned reward functions.

Both policies were trained using Stochastic Gradient Descent (SGD) with the following hyperparameters:

- Learning rate: 0.001
- Batch size: 32
- Epochs: 100
- Optimiser: Adam

The models were trained on NVIDIA Jetson Nano using TensorFlow, with reduced precision to meet edge device constraints. Loss convergence was monitored to assess overfitting and underfitting capability. As shown in Figure 6, SGD enabled stable convergence in both Behavioural Cloning (BC) and Inverse Reinforcement Learning (IRL) models, with IRL exhibiting greater robustness to noise and temporal inconsistencies.

To further assess learning efficiency, both models were trained for 300 epochs. The training and validation accuracy curves are presented in Figure 10. BC achieved approximately 78% validation accuracy, while IRL continued to improve, ultimately reaching 93%. This illustrates IRL's ability and learning efficiency, despite its increased computational cost. The performance trends in Figure 10 also demonstrate IRL's narrower gap and training stability. To evaluate the deployed policies under real-world constraints, a set of key performance indicators was established. This included trajectory accuracy, task completion success rate, grasp reliability, execution time, adaptability, emergency stop response time, and user satisfaction. Table 3 outlines the quantitative thresholds associated with each metric, reflecting the operational standards required for safe and effective assistive robot deployment in healthcare environments.

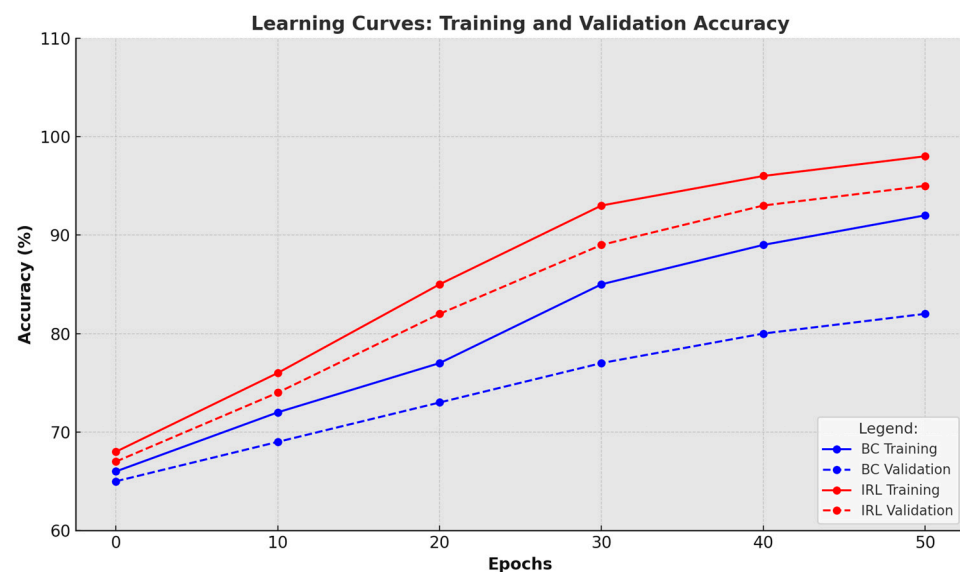


Figure 10. Learning curves for BC and IRL training and validation.

Table 3. Validation metrics thresholds.

Metric	Threshold	Justification
Trajectory Accuracy	<10 mm deviation	Ensures sufficient precision for household object handover tasks
Task Completion Success Rate	>90%	Required reliability for daily assistance operations
Execution Time	<2.0× human time	Balance between efficiency and safety for household tasks
Grasp Success Rate	>85%	Reliable grasping of varied objects (fruits, vegetables, daily items)
Adaptability Score	>80%	Ensures robustness in varying home lighting and object placement
Safety Metric (Emergency Stop Response)	<300 ms	Rapid response to unexpected situations
User Satisfaction Rating	>3.8/5.0	Acceptance criteria for home assistance users
Object Recognition Accuracy	>92%	Ability to correctly identify various household items
Handover Position Accuracy	<15 cm	Comfortable handover zone for human users

3.5. SLRS CAD Simulation and Real Applications Case Studies Evaluation

Three case study scenarios were conducted to assess SLRS validity and generalisability. These included representative scenarios of a hospital (i.e., surgical assistance), a domestic care setting (daily living support), and a rehabilitation environment. While these trials were conducted in university labs, they were modelled closely on actual task demands, workflows, and user interactions observed in clinical and assisted-living contexts [1,4,32].

3.5.1. Surgical Health Assistance

In this scenario, the SLRS operated as a robotic nurse analogue, retrieving surgical instruments and responding to gesture-based or verbal cues from a test operator. The simulation included the following:

- A surgical tray setup with identifiable tools;
- Timed request prompts from the operator;
- Lighting and spatial constraints reflecting operating theatre conditions.

Key metrics included tool selection accuracy and response latency. The system achieved over 92% accuracy in identifying and transferring the correct tool, with an average response time under 800 milliseconds. These results indicate that the SLRS could support surgical workflows by reducing manual search time and enhancing procedural efficiency [3,32,50].

3.5.2. Daily Living Health Assistance

In a mock home-care setup, the SLRS assisted participants in retrieving or delivering common household items (e.g., toothbrushes, medication containers, and packaged food items). The system operated on gesture cues and voice commands, and its usability was evaluated through task execution and participant feedback. Performance highlights included a grasp success rate of over 85% and task completion rate above 90%. These findings suggest the potential for SLRS deployment in eldercare and independent living scenarios, particularly for users with mobility impairments [19,37,51].

3.5.3. Rehabilitation and Care Support (Adaptive Interaction Trials)

This scenario simulated a rehabilitation setting with dynamic task environments. The robot was evaluated on its ability to give away items (e.g., toothbrush, banana, pen), adapt to unfamiliar users and lighting variations, and respond safely to occlusion or sudden human motion. The SLRS maintained a handover position error of less than 12 cm and responded to contact interruptions within 276 milliseconds, satisfying safety and responsiveness criteria. Importantly, it retained performance consistency across multiple user profiles and object types, indicating strong policy generalisation [22,51,52].

4. SLRS Testing, Validation, Results, and Discussion

The SLRS was tested, validated, and evaluated based on laboratory simulations that mimic typical healthcare use cases. The evaluation focused on key metrics including object detection accuracy, task execution success rate, interaction fluency, and system responsiveness under varying conditions, such as lighting changes and user motion. A formal risk assessment was conducted to ensure compliance with UK healthcare safety standards [18]. Additionally, data protection protocols were followed, with all sensor data encrypted and stored locally to support secure learning [38,52]. Hardware-level safety features, including emergency stop and redundant sensors, were implemented to ensure safe operation during human–robot interaction.

SLRS Performance Results

Object detection was a key component of the SLRS’s performance. Three computer vision models—YOLOv5, YOLOv8, and Single Shot Multi-Box Detector (SSD)—were evaluated as shown in Table 4. This was conducted under varying conditions, including different lighting, cluttered backgrounds, and object occlusion. YOLOv8 outperformed the others, achieving 94.3% accuracy and an average processing latency of just 18.7 milliseconds. These results demonstrate YOLOv8’s suitability for real-time object recognition in healthcare environments, enabling the robot to reliably identify assistive items (e.g., bananas, toothbrushes), even under challenging visual conditions.

Table 4. Computer vision algorithms comparison.

Algorithm	Detection Accuracy (%)	Processing Time (ms)	Reliability in Dynamic Settings
YOLOv8	94.3	18.7	High
YOLOv5	89.7	22.4	Moderate
SSD	86.2	27.9	Moderate

By combining YOLOv8 with a pose estimation pipeline using MediaPipe and LSTM, the system improved its ability to interpret human gestures and hand positions in real time. This multimodal perception enabled more accurate alignment between detected objects and user actions, which is critical for smooth handover tasks in care scenarios. For example, as illustrated in Figure 11, the robot was able to observe a user gesture and respond by picking up the correct item with precision and timing.

To assess general performance, the SLRS was tested in 150 trials across three healthcare-related scenarios: surgical assistance, daily living support, and rehabilitation. The system completed 131 out of 150 tasks successfully, resulting in a task success rate of 87.3%. A task was considered successful if the robot correctly interpreted the user’s intent, retrieved the appropriate object, and delivered it within acceptable time and spatial accuracy thresholds. Compared to a baseline rule-based system (62.1% success rate), the SLRS showed significantly higher adaptability and reliability. Statistical validation using the Wilcoxon

signed-rank test ($p < 0.05$) confirmed the effectiveness of the imitation learning-based approach in improving task performance under variable conditions. Figures 12–14 illustrate examples from the daily living use case, including object handovers of items such as bananas and toothbrushes. These tasks represent common assistive functions in eldercare or home environments, where the robot must respond accurately to gesture cues and operate safely near users. The SLRS demonstrated consistent behaviour across object types and participant styles, confirming its ability to generalise handover tasks with minimal variation in task timing or success rates.



Figure 11. Human demonstration of ‘Pick Up and Give Away’ action. This figure illustrates the robot learning from a human demonstration to perform handover tasks within a simulated care environment.

Even with these encouraging results, the current research and studies still have some notable limitations.

- First, the training dataset was relatively small, consisting of 20 demonstration videos totalling 8.5 h.

- Second, the evaluated tasks were confined to object handovers, excluding more complex healthcare interactions such as patient mobility support or natural language communication.
- Third, all trials were conducted in simulated environments and did not include clinical staff or real patients, limiting external validity.
- Finally, the system's performance was not benchmarked against other published SLRS implementations, making it difficult to assess relative advancement beyond the rule-based baseline.

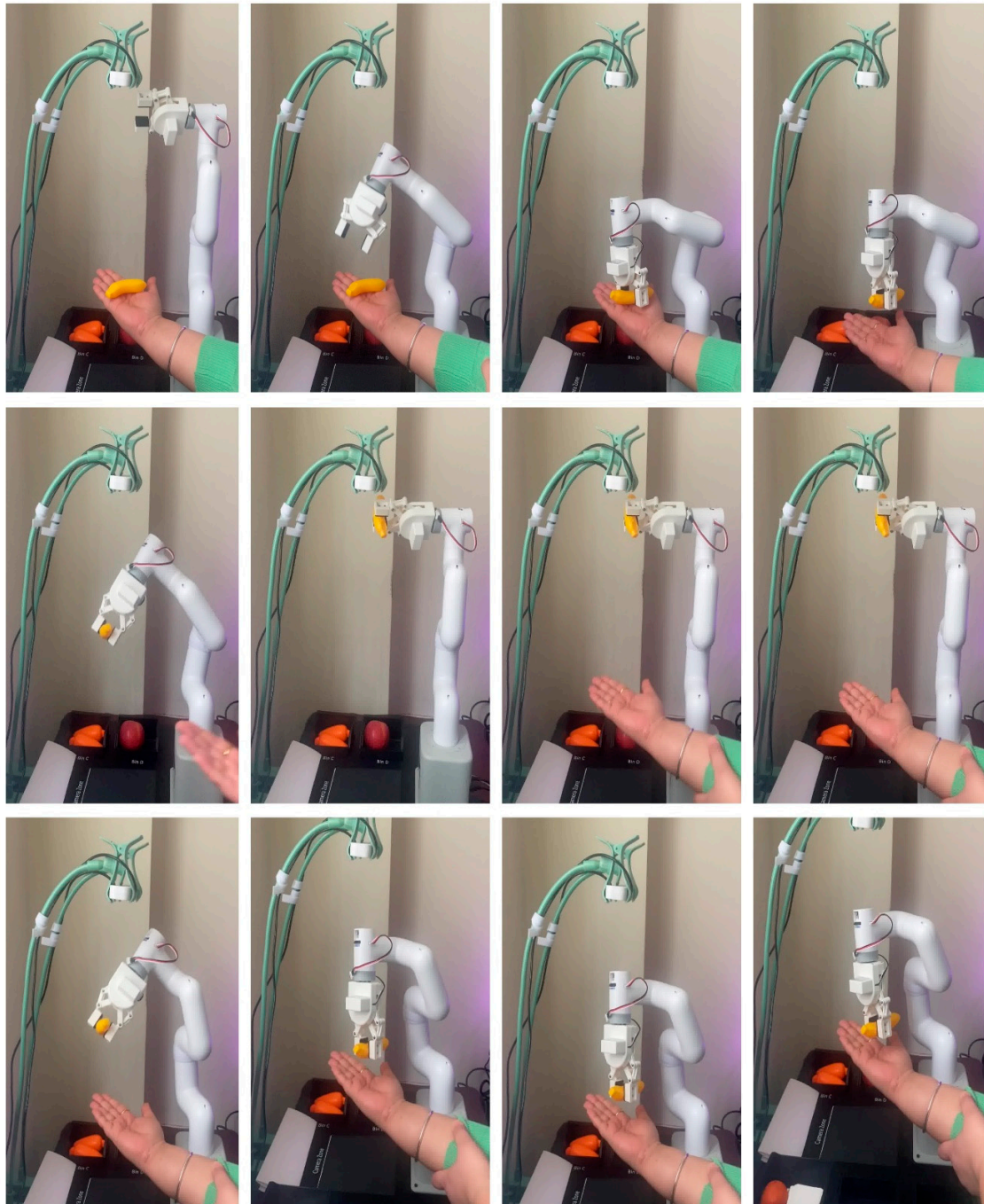


Figure 12. Take banana from carer and deliver to patient. The robot identifies and transfers a personal item as part of a handover task.

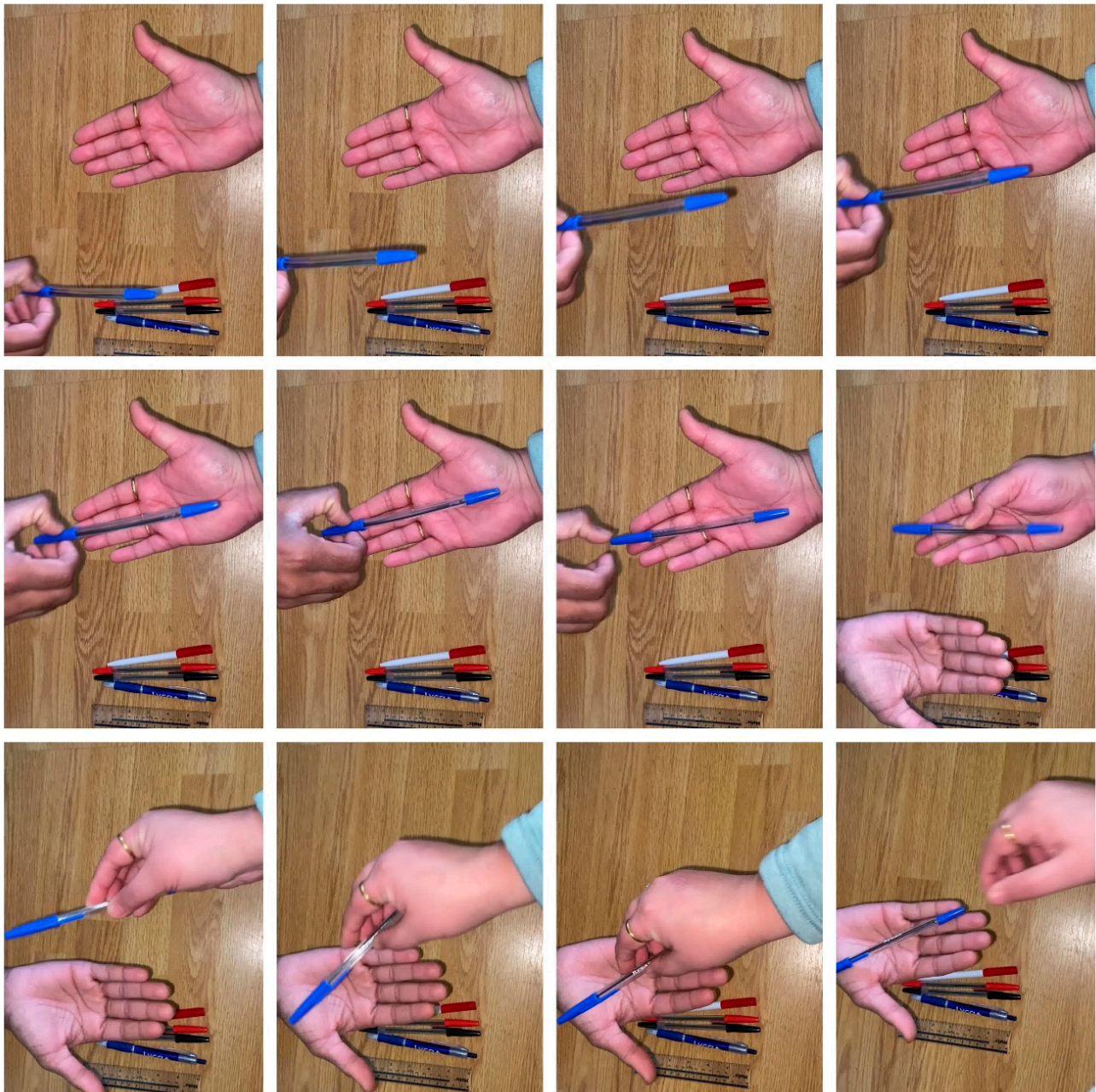


Figure 13. Human demonstration of the “Pick up and Give away” method demonstrating the structured routine the robot imitates for effective object handling.

Overall, the results indicated that an imitation learning approach combined with real-time visual and gesture recognition can enable robots to perform responsive, semi-autonomous assistive tasks in structured healthcare-like settings. The system’s ability to function on a cost-effective platform such as the Jetson Nano further suggests the feasibility of deploying similar solutions in resource-constrained environments, like care homes or community clinics. However, broader validation in uncontrolled and real-world healthcare settings remains necessary before full deployment.

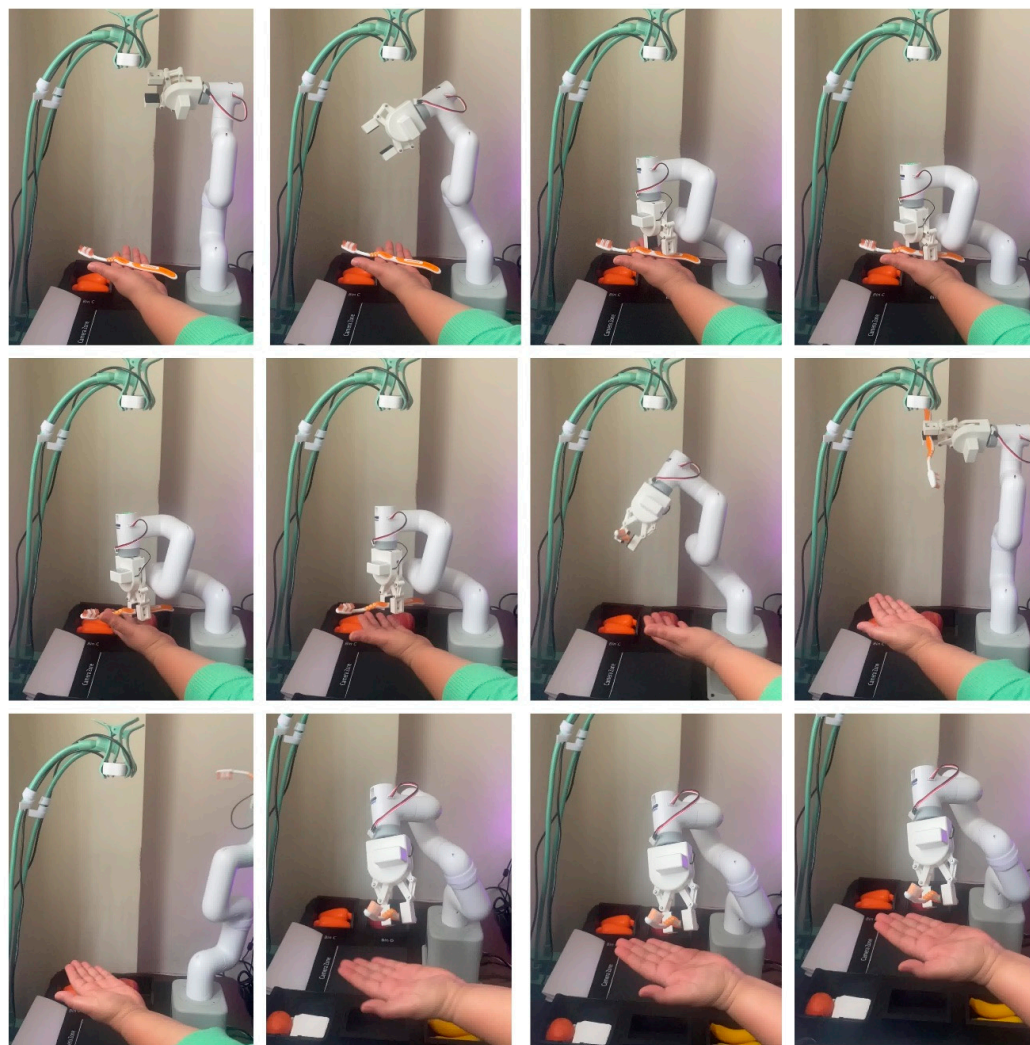


Figure 14. Pick up toothbrush and deliver to patient. Depicting task flexibility with different objects in a case scenario.

5. Conclusions and Future Work

A Self-Learning Robotic System (SLRS) for healthcare assistance applications was developed and presented in this paper. The SLRS utilised a hybrid model approach. This was to enable the system to acquire complex skills through observation of expert demonstrations without requiring explicit task-specific programming. It implemented deep imitation learning, where the optimised real-time learning and training methodology using more than one expert is implemented. The system contextual awareness and responsiveness were also enhanced using both a Convolutional Neural Network (CNN)-based object detection mechanism using YOLOv8 (i.e., with 94.3% accuracy and 18.7 ms latency) and pose estimation algorithms, alongside a MediaPipe and Long Short-Term Memory (LSTM) framework for human action recognition. The SLRS system was designed to address critical challenges in healthcare, including workforce shortages, ageing populations, and the increasing prevalence of chronic conditions. As discussed in Section 3.4.2, Figure 10, the comparative training results between BC and IRL highlight IRL's superior learning efficiency and training stability. These findings underscore the SLRS's potential for scalable and adaptive real-time deployment, even with the computational overhead associated with IRL. The 150 trials conducted to assess and evaluate the system were limited to a defined set of tasks and object types and do not address long-term evaluation or patient-specific personalisation. The CAD simulation, validation, and verification tested functions (i.e., assistive

functions, interactive scenarios, and object manipulation) of the system demonstrated the robot's adaptability and operational efficiency, achieving an 87.3% task completion success rate and over 85% grasp success rate. The key performance indicators parameters, such as object detection accuracy, task completion time, handover reliability, and interaction fluency, showed the system's operational effectiveness and possible useability in healthcare environments. Future work is essential to test and report the system's capabilities and include additional tasks such as patient mobility assistance and medication delivery, incorporate reinforcement learning for enhanced adaptability, and develop personalised interaction models. The integration of multimodal feedback systems and large-scale trials in real clinical settings will also be crucial for validating the system's reliability, safety, and user acceptance. It is believed that such an ongoing research programme will indeed enhance care quality and delivery across diverse clinical contexts in healthcare services.

Author Contributions: Conceptualization, Y.J. and M.S.; Methodology, Y.J.; Software, Y.J. and A.M.; Validation, Y.J. and M.S.; Formal analysis, Y.J., M.S. and P.W.; Investigation, Y.J.; Resources, Y.J., M.S. and A.M.; Data curation, Y.J. and A.M.; Writing—original draft, Y.J.; Writing—review & editing, Y.J., M.S., P.W. and A.M.; Visualization, Y.J. and A.M.; Supervision, M.S., P.W. and A.M.; Project administration, Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CNN	Convolutional Neural Network
DIL	Deep Imitation Learning
DOF	Degrees of Freedom
GAIL	Generative Adversarial Imitation Learning
HRI	Human–Robot Interaction
IL	Imitation Learning
IRL	Inverse Reinforcement Learning
LSTM	Long Short-Term Memory
MATLAB	Matrix Laboratory (MathWorks software)
MDP	Markov Decision Process
PID	Proportional–Integral–Derivative (controller)
PvD	Program via Demonstration
RGB-D	Red–Green–Blue with Depth (sensor)
RL	Reinforcement Learning
ROS	Robot Operating System
SGD	Stochastic Gradient Descent
SLRS	Self-Learning Robotic System
SSD	Single Shot Multi-mid-Box Detector
YOLOv8	You Only Look Once version 8

References

1. Yang, G.; Pang, Z.; Deen, M.J.; Dong, M. Homecare robotic systems for healthcare 4.0: Visions and enabling technologies. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2759–2771. [[CrossRef](#)] [[PubMed](#)]
2. Raj, R.; Kos, A. An Extensive Study of Convolutional Neural Networks: Applications in Computer Vision for Improved Robotics Perceptions. *Sensors* **2025**, *25*, 1033. [[CrossRef](#)]

3. Shah, S.I.H.; Coronato, A.; Naeem, M.; De Pietro, G. Learning and assessing optimal dynamic treatment regimes through cooperative imitation learning. *IEEE Access* **2022**, *10*, 84045–84060. [CrossRef]
4. Al-Hamadani, M.N.A.; Fadhel, M.A.; Alzubaidi, L.; Harangi, B. Reinforcement learning algorithms and applications in healthcare and robotics: A comprehensive and systematic review. *Sensors* **2024**, *24*, 2461. [CrossRef] [PubMed]
5. Alshammari, R.F.N.; Arshad, H.; Rahman, A.H.A.; Albahri, O.S. Robotics utilization in automatic vision-based assessment systems from artificial intelligence perspective: A systematic review. *IEEE Access* **2022**, *10*, 145621–145635. [CrossRef]
6. Alatabani, L.E.; Ali, E.S.; Saeed, R.A. Machine Learning and Deep Learning Approaches for Robotics Applications. In *Lecture Notes in Networks and Systems*; Springer: Berlin/Heidelberg, Germany, 2023; Volume 519.
7. Jadeja, Y.; Shafik, M.; Wood, P. An industrial self-learning robotic platform solution for smart factories using machine and deep imitation learning. In *Advances in Manufacturing Technology XXXIV*; IOS Press: Amsterdam, The Netherlands, 2021; pp. 63–68.
8. Zare, M.; Kebria, P.M.; Khosravi, A.; Nahavandi, S. A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *54*, 7173–7186. [CrossRef]
9. Pan, Y.; Cheng, C.-A.; Saigol, K.; Lee, K.; Yan, X.; A Theodorou, E.; Boots, B. Imitation learning for agile autonomous driving. *Int. J. Robot. Res.* **2020**, *39*, 515–533. [CrossRef]
10. Mahajan, H.B.; Uke, N.; Pise, P.; Shahade, M.; Dixit, V.G.; Bhavsar, S.; Deshpande, S.D. Automatic robot manoeuvres detection using computer vision and deep learning techniques. *Multimed. Tools Appl.* **2023**, *82*, 21229–21255. [CrossRef]
11. Siciliano, B.; Khatib, O. *Springer Handbook of Robotics*, 2nd ed.; Springer: Cham, Switzerland, 2016.
12. Jadeja, Y.; Shafik, M.; Wood, P. A comprehensive review of robotics advancements through imitation learning for self-learning systems. In Proceedings of the 2025 9th International Conference on Mechanical Engineering and Robotics Research (ICMERR), Barcelona, Spain, 15–17 January 2025.
13. Jadeja, Y.; Shafik, M.; Wood, P. Computer-aided design of self-learning robotic system using imitation learning. In *Advances in Manufacturing Technology*; IOS Press: Amsterdam, The Netherlands, 2022; pp. 47–53.
14. Argall, B.D.; Chernova, S.; Veloso, M.; Browning, B. A survey of robot learning from demonstration. *Robot. Auton. Syst.* **2009**, *57*, 469–483. [CrossRef]
15. Tonga, P.A.; Ameen, Z.S.; Mubarak, A.S. A review on on-device privacy and machine learning training. *IEEE Access* **2022**, *10*, 115790–115805.
16. Calp, M.H.; Butuner, R.; Kose, U.; Alamri, A. IoHT-based deep learning controlled robot vehicle for paralyzed patients of smart cities. *J. Supercomput.* **2022**, *78*, 17635–17657. [CrossRef]
17. Butuner, R.; Kose, U. Application of artificial intelligence algorithms for robot development. In *AI in Robotics and Automation*; CRC Press: Boca Raton, FL, USA, 2023.
18. Otamajakusi. YOLOv8 on Jetson Nano. i7y.org. 2023. Available online: <https://i7y.org/en/yolov8-on-jetson-nano/> (accessed on 14 July 2023).
19. Fang, Y.; Xie, J.; Li, M.; Maroto-Gómez, M.; Zhang, X. Recent advancements in multimodal human–robot interaction: A systematic review. *Front. Neurobotics* **2023**, *17*, 1084000.
20. Smarr, C.A.; Prakash, A.; Beer, J.M.; Mitzner, T.L.; Kemp, C.C.; Rogers, W.A. Older Adults’ Preferences for and Acceptance of Robot Assistance for Everyday Living Tasks. *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* **2014**, *56*, 153–157. [CrossRef]
21. Kruse, T.; Pandey, A.K.; Alami, R.; Kirsch, A. Human-aware Robot Navigation: A Survey. *Robot. Auton. Syst.* **2013**, *61*, 1726–1743. [CrossRef]
22. Chen, X.; Wang, Z.; Zhang, Z.; Wang, X.; Yang, R. A Deep Learning Approach to Grasp Type Recognition for Robotic Manipulation in Healthcare. *IEEE Robot. Autom. Lett.* **2023**, *8*, 2358–2365.
23. Kyrarini, M.; Lygerakis, F.; Rajavenkatanarayanan, A.; Sevastopoulos, C.; Nambiappan, H.R.; Chaitanya, K.K.; Babu, A.R.; Mathew, J.; Makedon, F. A Survey of Robots in Healthcare. *Technologies* **2021**, *9*, 8. [CrossRef]
24. Banisetty, S.B.; Feil-Seifer, D. Towards a Unified Framework for Social Perception and Action for Social Robotics. *ACM Trans. Hum. Robot Interact.* **2022**, *11*, 1–26.
25. Gao, Y.; Chang, H.J.; Demiris, Y. User Modelling for Personalised Dressing Assistance by Humanoid Robots. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1840–1847.
26. Muelling, K.; Venkatraman, A.; Valois, J.-S.; Downey, J.E.; Weiss, J.; Javdani, S.; Hebert, M.; Schwartz, A.B.; Collinger, J.L.; Bagnell, J.A. Autonomy Infused Teleoperation with Application to Brain Computer Interface Controlled Manipulation. *Auton. Robot.* **2017**, *41*, 1401–1422. [CrossRef]
27. Johansson, D.; Malmgren, K.; Murphy, M.A. Wearable Sensors for Clinical Applications in Epilepsy, Parkinson’s Disease, and Stroke: A Mixed-Methods Systematic Review. *J. Neurol.* **2022**, *265*, 1740–1752. [CrossRef]
28. Redmon, J.; Farhadi, A. YOLOv8: An Incremental Improvement. *arXiv* **2023**, arXiv:2304.00501.
29. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

30. Maddox, T.; Fitzpatrick, J.M.; Guitton, A.; Maier, A.; Yang, G.Z. The Role of Human-Robot Interaction in Intelligent and Ethical Healthcare Delivery. *Nat. Med.* **2022**, *28*, 1098–1102.
31. Fang, B.; Jia, S.; Guo, D.; Xu, M.; Wen, S.; Sun, F. Survey of Imitation Learning for Robotic Manipulation. *Int. J. Intell. Robot. Appl.* **2021**, *5*, 3–15. [[CrossRef](#)]
32. Manzi, F.; Sorgente, A.; Massaro, D. Acceptance of Robots in Healthcare: State of the Art and Suggestions for Future Research. *Societies* **2022**, *12*, 143.
33. Draper, H.; Sorell, T. Ethical Values and Social Care Robots for Older People: An International Qualitative Study. *Ethics Inf. Technol.* **2022**, *19*, 49–68. [[CrossRef](#)]
34. Jokinen, K.; Wilcock, G. Multimodal Open-Domain Conversations with Robotic Platforms. In *Computer Vision and Pattern Recognition, Multimodal Behavior Analysis in the Wild*; Academic Press: Cambridge, MA, USA, 2019; pp. 9–16.
35. Shin, H.V.; Tokmouline, M.; Shah, J.A. Design Guidelines for Human-AI Co-Learning: A Mixed Methods Participatory Approach. *Proc. ACM Hum. Comput. Interact.* **2023**, *7*, 1–26.
36. Azeta, J.; Bolu, C.; Abioye, A.A.; Oyawale, F.A. A Review on Humanoid Robotics in Healthcare. *MATEC Web Conf.* **2018**, *153*, 02004.
37. Breazeal, C.; DePalma, N.; Orkin, J.; Chernova, S.; Jung, M. Crowdsourcing Human-Robot Interaction: New Methods and System Evaluation in a Public Environment. *J. Hum. Robot Interact.* **2022**, *2*, 82–111. [[CrossRef](#)]
38. Yang, G.-Z.; Cambias, J.; Cleary, K.; Daimler, E.; Drake, J.; Dupont, P.E.; Hata, N.; Kazanzides, P.; Martel, S.; Patel, R.V.; et al. Medical Robotics—Regulatory, Ethical, and Legal Considerations for Increasing Levels of Autonomy. *Sci. Robot.* **2017**, *2*, eaam8638. [[CrossRef](#)]
39. Tosun, O.; Aghakhani, M.; Chauhan, A.; Sefidgar, Y.S.; Hoffman, G. Trust-Aware Human-Robot Interaction in Healthcare Settings. *Int. J. Soc. Robot.* **2023**, *15*, 479–496.
40. Ho, J.; Ermon, S. Generative Adversarial Imitation Learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 4565–4573.
41. Celemin, C.; Ruiz-del-Solar, J. Interactive imitation learning: A survey of human-in-the-loop learning methods. *arXiv* **2022**, arXiv:2203.03101.
42. Alotaibi, B.; Manimurugan, S. Humanoid robot-assisted social interaction learning using deep imitation in smart environments. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 2511–2525.
43. Rösmann, M.; Hoffmann, F.; Bertram, T. Trajectory modification considering dynamic constraints of autonomous ROBOTIK 2012. In Proceedings of the 7th German Conference on Robotics, Munich, Germany, 21–22 May 2012; p. 16.
44. NVIDIA. Jetson Nano Developer Kit: Datasheet and Power Performance Guide. 2022. Available online: <https://developer.nvidia.com/embedded/jetson-nano> (accessed on 25 January 2023).
45. Tsuji, T.; Lee, Y.; Zhu, Y. Imitation learning for contact-rich manipulation in assistive robotics. *Robot. Auton. Syst.* **2025**, *165*, 104390.
46. Developers, R.O. MoveIt: Motion Planning Framework for ROS2. 2022. Available online: <https://moveit.ros.org> (accessed on 9 January 2023).
47. Lugaresi, C.; Tang, J.; Nash, H.; McClanahan, C.; Uboweja, E.; Hays, M.; Zhang, F.; Chang, C.L.; Yong, M.G.; Lee, J.; et al. MediaPipe: A Framework for Building Perception Pipelines. *arXiv* **2019**, arXiv:1906.08172.
48. Zhang, H.; Thomaz, A.L. Learning social behaviour for assistive robots using human demonstrations. *ACM Trans. Hum.-Robot. Interact.* **2020**, *9*, 1–25. [[CrossRef](#)]
49. Sharma, S.; Sing, R.K.; Fryazinov, O.; Adzhiev, V. K-means based stereo vision clustering for 3D object localization in robotics. *IEEE Access* **2022**, *10*, 112233–112245.
50. Caldwell, M.; Andrews, J.T.A.; Tanay, T.; Griffin, L.D. AI-Enabled Future Surgical Healthcare: Ethics Principles and Epistemic Gaps. *AI Soc.* **2023**, *38*, 1–16.
51. Sharma, S.; Sing, R.K.; Fryazinov, O.; Adzhiev, V. Computer Vision Techniques for Robotic Systems: A Systematic Review. *IEEE Access* **2022**, *10*, 29486–29510.
52. Robotics, E. MyCobot 280 Jetson Nano: Robotic Arm Specification Sheet. 2023. Available online: <https://www.elephantrobotics.com/product/mycobot-280-jetson-nano/> (accessed on 14 July 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.