Imitation Learning for a Snake Robot

Muhammad Kamil[1], Shahmir Ali[2], M. Noman Saeed[2], Muhammad Aqib Jamal[1], Anayat Ullah[1,2]
[1]Department of Electronic Engineering, Balochistan University of IT, Engineering and Management Sciences, Quetta, Pakistan.
[2]Control, Automotive and Robotics Lab, National Centre of Robotics and Automation, Rawalpindi, Pakistan.
***Correspondence**: lehrikamil@gmail.com

Citation | Kamil. M, Ali. S, Saeed. M. N, Jamal. M. A, Ullah. A, "Imitation Learning for a Snake Robot", IJIST, Special Issue pp 203-219, May 2026
Received| March 19, 2026 **Revised|** April 24, 2026 **Accepted|** April 27, 2026 **Published|** May 02, 2026.

Introduction/Importance of Study: The emergence of learning-based methods, particularly imitation learning (IL), provides an effective alternative to traditional control strategies for complex robotic systems. IL enables robots to learn control policies directly from expert demonstrations, eliminating the need for explicit modeling of highly nonlinear dynamics.
Novelty statement: This study proposes an imitation learning framework for a snake robot based on a hybrid CNN–LSTM architecture, designed to capture both spatial and temporal dependencies inherent in locomotion tasks.
Material and Method: A simulated snake robot was developed in a ROS1–Gazebo environment. RGB-D images, motor commands, directional labels, and timestamps were collected and stored in CSV format. Deep learning models were implemented in PyTorch to learn a mapping from sensory inputs to motor control actions.
Result and Discussion: The baseline CNN model achieved a test accuracy of approximately 19–20%, despite exceeding 90% training accuracy, indicating severe overfitting and poor generalization. The model showed signs of overfitting, as indicated by decreasing training loss but increasing validation loss across 100–140 epochs. In contrast, the proposed CNN–LSTM model achieved a test accuracy of 96.37% with a macro F1-score of 0.89. The model demonstrated rapid convergence within 10–15 epochs, with training and validation accuracies reaching approximately 99% and 95–96%, respectively. Confidence-based evaluation further indicated high statistical reliability (95% confidence interval: 96.30%–96.44%). Confusion matrix analysis confirmed strong class-wise performance, with rectilinear forward motion achieving near-perfect accuracy (~99%), while other motion classes were also classified with minimal error.
Concluding Remarks: The results demonstrate that integrating temporal memory with spatial feature extraction significantly enhances imitation learning performance for snake robot control. Deployed on an NVIDIA Tesla A100 GPU (32 GB), the lightweight CNN–LSTM model achieves an estimated inference latency of 2–10 ms per input sequence, indicating its suitability for real-time robotic control in simulated environments.

Keywords: Imitation Learning; Supervise Learning; Depth Images; CNN-LSTM and Snake Robot.

Introduction:

Robots are increasingly used in environments where human presence is dangerous or impractical, such as disaster zones, deep underwater areas, and confined industrial pipelines [1][2]. These environments are often difficult to access and unpredictable, which makes them risky for human workers. For example, during search and rescue operations, collapsed structures or toxic conditions can prevent rescuers from reaching trapped victims. Similarly, inspection tasks in pipelines or underwater systems require machines that can navigate narrow spaces and operate without direct human control. In such situations, robots provide a safer and more efficient solution, especially when they are capable of flexible movement. Snake robots are a type of biologically inspired robot designed to imitate the motion of real snakes. They consist of multiple connected joints that allow the robot to bend and move through complex environments [3]. This flexible structure enables them to travel over uneven terrain, move through tight spaces, and climb over obstacles where wheeled or legged robots may fail. Because of these capabilities, snake robots are considered useful for applications such as search and rescue, pipeline inspection, and monitoring hazardous industrial environments [3]. However, controlling snake robots is challenging due to their large number of joints and degrees of freedom. Each joint must move in coordination with the others to produce smooth locomotion. Even small errors in joint control can disturb the overall movement of the robot. Traditional control methods based on mathematical models or rule-based algorithms often work well in controlled conditions but may struggle in real-world environments where conditions are uncertain and constantly changing [4].

Recent advancements in imitation learning for robotic systems have increasingly focused on improving generalization, robustness, and real-time adaptability, particularly through hybrid architectures and sequence modeling techniques. However, most state-of-the-art approaches are primarily developed for manipulation tasks or mobile robots, with limited emphasis on hyper-redundant systems such as snake robots. Furthermore, many existing methods rely on either purely convolutional architectures or reinforcement learning-based frameworks, which may struggle to effectively capture the temporal dependencies inherent in continuous locomotion. In contrast, the proposed CNN–LSTM framework explicitly integrates spatial and temporal modeling, enabling improved performance in sequential motion prediction. Additionally, this study incorporates quantitative gait analysis, confidence-based evaluation, and interpretability through Grad-CAM, which are often not jointly addressed in recent works. Therefore, the proposed approach provides a more comprehensive framework for imitation learning in snake robot locomotion, positioning it favorably against current state-of-the-art methods.

Literature Review:

Machine learning techniques have recently been explored as alternative solutions for controlling complex robotic systems. Reinforcement learning (RL) is one such approach where a robot learns by interacting with its environment and improving its actions through trial and error [5]. Although RL has shown promising results in robotic control, it usually requires a large amount of training data and computational time, often involving millions of interactions before achieving reliable performance. These requirements can make RL difficult to apply in practical situations where faster training is needed [5][6]. Imitation learning (IL) offers a more efficient alternative by allowing robots to learn directly from expert demonstrations. Instead of discovering behaviors through trial and error, the robot observes example actions and learns to reproduce them. This approach significantly reduces the training time and data requirements, making it suitable for tasks such as robotic navigation and control [6]. Deep learning models are commonly used in imitation learning to map sensory inputs to motor actions. Convolutional Neural Networks (CNNs) are widely used for extracting spatial features from visual data such as images captured by onboard cameras. However, CNNs process individual frames independently and therefore cannot capture temporal relationships between consecutive actions [7]. For tasks like snake locomotion, where movements depend on previous actions, temporal information is essential. To address this limitation, Long Short-Term Memory (LSTM) networks are often combined with CNN models. The CNN extracts spatial features from images, while the LSTM learns patterns over time by remembering previous states. This CNN–LSTM combination allows the system to understand both the visual environment and the sequence of movements, making it suitable for continuous robotic control tasks [7][8]. In this study, we apply imitation learning to train a snake robot in a simulated ROS and Gazebo environment. The robot is equipped with a head-mounted depth camera that captures RGB and depth images. Along with visual data, motor commands, directional information, and timestamps are recorded and stored in CSV format, creating a dataset that links perception to motor actions [7]. Initially, a CNN model was trained to predict motor commands from individual image frames. Although the CNN was able to capture spatial information from the images, it could not model the sequential nature of snake locomotion. To overcome this limitation, a hybrid CNN–LSTM model was developed. In this architecture, the CNN extracts spatial features while the LSTM captures temporal dependencies between movements. The combined model produces smoother trajectories and more accurate motion control compared to the CNN-only approach [9]. The remainder of this paper is organized as follows. Section II describes the materials and methods, including the simulation environment, data collection process, dataset preparation, and model architecture. Section III presents the experimental results and discussion. Finally, Section IV concludes the paper and suggests directions for future work. The objectives of this research are threefold: first, to demonstrate that imitation learning with supervised learning can significantly reduce the time and data required for training snake robots; second, to validate the effectiveness of combining CNN and LSTM architectures for integrating spatial and temporal information in robotic control; and third, to provide empirical evidence that this hybrid approach outperforms vision-only models in accuracy, smoothness, and reliability. The novelty of this work lies in the application of a unified CNN–LSTM framework within an imitation learning context to achieve efficient, robust, and real-time locomotion control for hyper-redundant snake robots, a step toward practical deployment in real-world unstructured environments.

Recent advancements in snake robot research have focused on improving locomotion adaptability, perception integration, and learning-based control strategies for hyper-redundant systems. Recent studies highlight that snake robots offer superior maneuverability in complex and unstructured environments; however, their control remains challenging due to high degrees of freedom and nonlinear dynamics [10]. Reinforcement learning-based approaches have demonstrated improved adaptability and path planning in such systems, but they typically require extensive training data and computational resources, limiting their real-time applicability [11]. Furthermore, recent works have explored perception-driven and sensor-based control strategies for snake robots operating in complex terrains; however, these approaches often introduce additional computational overhead and system complexity [12].

Despite these advancements, imitation learning for hyper-redundant snake robots remains relatively underexplored, particularly in the context of integrating both spatial and temporal features for sequential locomotion tasks. Existing approaches either rely on spatial-only models, which fail to capture temporal dependencies, or reinforcement learning methods, which are computationally expensive and difficult to scale. Recent developments in imitation learning emphasize combining perception with temporal sequence modeling for robust robotic control [13]. However, the application of hybrid architectures such as CNN–LSTM for efficient imitation learning in snake robots has not been sufficiently investigated.

Therefore, this study addresses this gap by proposing a hybrid CNN–LSTM-based imitation learning framework that simultaneously captures spatial and temporal dependencies, enabling efficient, accurate, and scalable control of snake robot locomotion in simulated environments. Recent studies have also explored deep learning and reinforcement learning approaches for robotic control and navigation [14][15][16][17][18].

Research Gap and Motivation:

Despite recent progress in imitation learning for robotic control, limited work has addressed hyper-redundant systems such as snake robots, where motion is inherently sequential and time-dependent. Existing approaches often rely on spatial feature extraction alone, without effectively modeling the temporal dependencies required for continuous locomotion. Furthermore, prior studies rarely incorporate quantitative motion analysis or interpretability mechanisms to validate learned behaviors. To address these limitations, this study proposes a hybrid CNN–LSTM framework that integrates spatial perception with temporal modeling, enabling improved performance and robustness in snake robot imitation learning.

Research Objectives:

The primary objective of this study is to develop an efficient imitation learning framework for controlling a hyper-redundant snake robot using deep learning techniques. To achieve this, the following specific and measurable objectives are defined:

To design and implement a baseline Convolutional Neural Network (CNN) model for predicting snake robot motor commands from visual input and evaluate its performance in terms of classification accuracy and generalization.

To develop a hybrid CNN–LSTM architecture capable of capturing both spatial and temporal dependencies in sequential locomotion tasks.

To quantitatively compare the performance of CNN and CNN–LSTM models using evaluation metrics such as accuracy, F1-score, and confusion matrix analysis.

To analyze the convergence behavior and generalization capability of both models based on training and validation performance across multiple epochs.

To validate the effectiveness of the proposed CNN–LSTM model in accurately predicting multiple snake locomotion patterns, including rectilinear and sidewinding motions.

To assess the feasibility of the proposed model for near real-time deployment in simulated robotic environments.

Contributions of the Study:

A hybrid CNN–LSTM-based imitation learning framework is proposed for controlling a hyper-redundant snake robot, enabling simultaneous extraction of spatial and temporal features from visual input.

A comprehensive comparison between a baseline CNN model and the proposed CNN–LSTM model is conducted, demonstrating the limitations of spatial-only learning and the advantages of incorporating temporal memory.

The proposed model achieves significant performance improvement, reaching approximately 96% classification accuracy with strong generalization, compared to the poor performance (approximately 19–20% accuracy) of the CNN-only model.

A complete end-to-end pipeline is developed, integrating simulation, data acquisition, preprocessing, model training, and deployment within a ROS–Gazebo environment.

The study demonstrates the effectiveness of imitation learning as a computationally efficient alternative to reinforcement learning for snake robot locomotion tasks.

The proposed framework shows potential for near real-time robotic control, highlighting its applicability to practical scenarios involving complex and unstructured environments.

Materials and Methods:

Investigation site:

This study was conducted entirely within a simulated investigation site developed using the Robot Operating System (ROS1) and the Gazebo simulator. The virtual environment serves as a rigorous and repeatable proving ground for robotic control algorithms. Gazebo provides high-fidelity physics simulation, including realistic modeling of rigid-body dynamics, sensor noise, and environmental interaction, which is essential for developing controllers destined for real-world deployment [19]. The core research "site" consists of two key components:

The Robot Model: A snake robot, defined by a Unified Robot Description Format (URDF) file, comprising 11 links and 10 revolute joints.

The Testing Arena: Customizable Gazebo worlds featuring varied terrain, obstacles, and lighting conditions, allowing for comprehensive testing of locomotion policies under diverse, controllable challenges.

This simulation-based site is critically "hot for research" as it enables the safe, rapid, and cost-effective iteration of learning algorithms, a necessity when working with complex, high-degree-of-freedom systems like snake robots. It eliminates the risks of hardware damage during early training phases and allows for the generation of large, perfectly labeled datasets that would be prohibitively difficult to obtain from physical robots in real environments [20].

Materials and methods:

The methodology follows a clear, reproducible pipeline encompassing simulation, data acquisition, model design, and training. All software components are open-source, and the process is designed to be replicable by other researchers with access to a standard computational setup.

Simulation Environment and Robot Design:

The snake robot was modeled in ROS1 Noetic using a custom URDF file. The design features 11 cylindrical links connected by 10 revolute joints. Each joint is oriented at a 90° angle alternately to the previous one, creating an orthogonal configuration that enables movement in three dimensions. This design allows the robot to execute four fundamental locomotion gaits within the simulator: forward rectilinear, backward rectilinear, left-side winding, and right-side winding. The primary sensor is a depth camera (RGB-D) attached to the robot's head, providing synchronized RGB and depth image streams at a fixed frequency. The simulation environment was built and managed using Gazebo.

Figure 1. Gazebo Environment

Data Collection and Sources:

The simulation environment used for data collection is shown in Fig. 1. Expert demonstration data was acquired directly from the simulation to ensure perfect reliability and synchronization. Using manual control via ROS topics, the robot was navigated through various environments. The following data streams were recorded synchronously via ROS bags and later exported to structured CSV files.

Motor Values: The target position commands sent to each of the 10 joints.

Direction: A numerical encoding of the commanded gait (forward, backward, left, right).

RGB Images: 640x480 pixel frames from the robot's perspective.

Depth Images: Corresponding 640x480 pixel depth maps.

Timestamps: Precise timing data to maintain temporal alignment across all modalities.

This dataset forms the core of our training material, creating a direct mapping of visual perception (RGB-D) action (motor command). The source is 100% internally generated, and the collection scripts are available for replication.

Data Preparation:

The raw data was processed into a format suitable for deep learning. Depth images were resized, normalized, and stacked to form multi-channel input samples. The sequential motor commands were aligned with their corresponding image sequences. The dataset was then partitioned into training (70%), validation (15%), and testing (15%) sets while carefully preserving the temporal order of each episode to respect the sequential nature of the LSTM training.

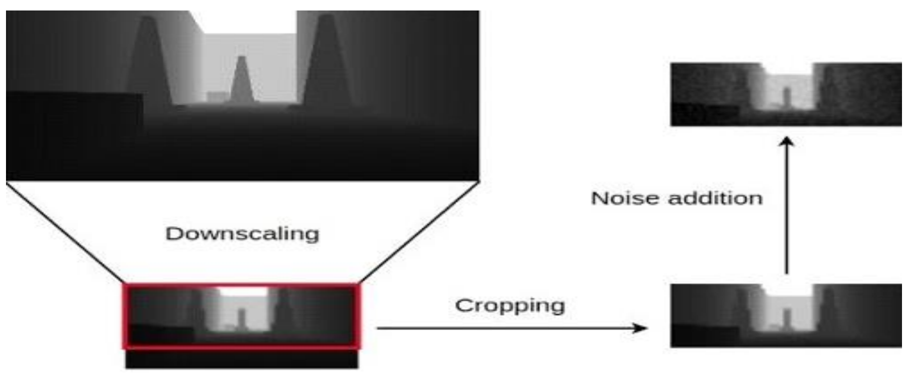

Figure 2. Model Architecture and Training

Data Preprocessing:

The overall model architecture and training pipeline are illustrated in Fig. 2. The collected dataset consists of RGB-D images, motor command values, directional labels, and timestamps. For model training, depth images were primarily used as input due to their robustness in representing spatial structure. All input images were resized to a fixed resolution of 90×160 pixels to ensure computational efficiency and consistency across the dataset. The images were converted to grayscale and transformed into tensor format for compatibility with the deep learning framework. To standardize the input distribution, pixel values were normalized using a mean of 0.5 and a standard deviation of 0.5. This normalization improves training stability and accelerates convergence. To enhance model robustness and prevent overfitting, data augmentation was applied during training. Specifically, Gaussian noise with zero mean and a standard deviation of 0.15 was added to the input images. This simulates sensor noise and improves the model's ability to generalize under varying environmental conditions. For temporal modeling, input data was structured into sequences. Each sequence consisted of 15 consecutive frames (sequence length = 15), with a sequence step of 5 frames between successive samples. This overlapping sequence generation enables the model to capture temporal dependencies while maintaining sufficient data diversity. The dataset was split into training, validation, and testing subsets using stratified sampling to preserve class distribution across all motion categories.

Mathematical Formulation of CNN–LSTM Model:

The proposed CNN–LSTM model learns a mapping from a sequence of input frames to the corresponding motion class. Let the input sequence be represented as:

$$X = \{x_1, x_2, \ldots, x_T\}$$

where x_t denotes the input frame at time step t, and T represents the sequence length.

The Convolutional Neural Network (CNN) extracts spatial features from each input frame:

$$f_t = CNN(x_t)$$

where f_t represents the feature vector corresponding to frame x_t.

The sequence of extracted feature vectors is then passed to the Long Short-Term Memory (LSTM) network to capture temporal dependencies:

$$h_t = LSTM(f_t, h_{t-1})$$

where h_t denotes the hidden state at time step t, and h_{t-1} is the previous hidden state.

The final prediction is obtained through a fully connected layer followed by a softmax function:

$$y = \text{Softmax}(W \cdot h_T + b)$$

where y represents the predicted probability distribution over the motion classes, W denotes the weight matrix, and b represents the bias term.

The model is trained using the categorical cross-entropy loss function:

$$L = -\Sigma\, (y_i \log(\hat{y}_i))$$

where y_i is the ground truth label, $\hat{y}_i$ is the predicted probability for class i, and C is the total number of classes.

Hybrid CNN-LSTM Model:

This novel architecture addresses the temporal limitation. Spatial features are first extracted from each frame using a CNN encoder. These feature vectors are then fed sequentially into an LSTM layer, which learns the temporal dependencies across a window of frames (e.g., the last 10 time steps). Finally, the output from the LSTM is passed through a fully connected layer to predict the next motor command. This design allows the model to base its decisions on both what it sees now and what it was doing a moment ago.

Our Workflow Follows Three Main Stages:

Data collection, model training, and deployment. In the Gazebo simulator, a snake robot equipped with a depth camera gathers RGB and depth images along with motor commands, directions, and timestamps. This dataset is cleaned and formatted into CSV files, then used to train a hybrid neural network that combines Convolutional Neural Networks (CNNs) for spatial feature extraction and Long Short-Term Memory (LSTM) layers for capturing temporal patterns. Once trained, this CNN–LSTM model predicts motor commands directly from live camera input, enabling the robot to reproduce expert-like movement.

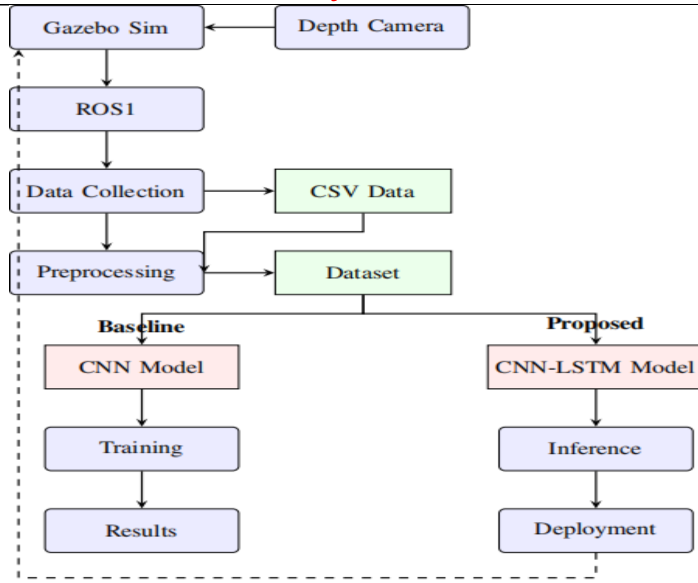

Figure 3. Flow Chart of Model Architecture

Figure 3 illustrates the complete pipeline of the proposed imitation learning framework, consisting of data acquisition, preprocessing, model training, and deployment stages. The process begins with the snake robot operating in the Gazebo simulation environment, where a depth camera continuously captures RGB and depth image streams. Simultaneously, motor commands, directional labels, and timestamps are recorded through ROS topics, forming synchronized multimodal data.

The collected data are then stored in structured CSV files, as indicated by the data logging block in the flowchart. In the preprocessing stage, the raw image data are resized, normalized, and organized into sequential samples. For the CNN–LSTM model, temporal sequences of fixed length (e.g., 15 time steps) are constructed to preserve motion continuity, while corresponding motor commands are aligned with each sequence.

The next stage involves model training, where two parallel pathways are shown in the flowchart. The first pathway represents the baseline CNN model, which processes individual frames independently to predict motor commands. The second pathway represents the proposed CNN–LSTM model, where spatial features are first extracted using convolutional layers and then passed to LSTM layers to capture temporal dependencies across sequential frames.

During training, both models are optimized using supervised learning, and their performance is evaluated using validation data. The flowchart arrows connecting these blocks represent the transformation of data from raw sensory input to learned feature representations and finally to predicted motor actions.

In the final stage, the trained CNN–LSTM model is deployed for inference. Live input from the robot's camera is processed through the trained network to generate motor commands in real time, enabling the robot to reproduce expert-like locomotion behavior. This step-by-step pipeline highlights the role of temporal modeling in improving prediction accuracy and overall control performance.

The computational efficiency of the proposed CNN–LSTM model is an important consideration for real-time robotic applications. The model consists of approximately 1.45 million trainable parameters, making it lightweight compared to typical deep learning architectures used for sequential prediction tasks. This moderate model size enables efficient training and deployment without requiring excessive computational resources. In terms of inference performance, the model was deployed on an NVIDIA Tesla A100 GPU with 32 GB of memory. Based on the model complexity and input configuration, the proposed architecture exhibits an estimated inference latency in the range of approximately 2–10 ms per input sequence. This low-latency performance indicates that the model is capable of operating in near real-time, which is essential for continuous snake robot locomotion control in dynamic environments. Although the present study is conducted in a simulation environment, the proposed framework is designed with sim-to-real transfer considerations. The use of depth-based perception, normalized input representations, and sequence-based learning improves robustness against environmental variations. Furthermore, the modular pipeline allows direct integration with real robotic hardware through ROS, enabling future deployment with minimal modifications. Domain adaptation techniques, such as noise injection during training and data normalization, further support generalization from simulation to real-world scenarios.

Results and Discussion:

Figure 4 illustrates the training behavior of the CNN-only model. Although the model initially demonstrates learning capability, its performance gradually plateaus at later epochs. The divergence between training and validation loss indicates limited generalization, suggesting that the model begins to overfit the training data rather than learning generalized motion representations. Consequently, the model achieves comparatively lower accuracy, highlighting the limitation of using spatial features alone without temporal context.

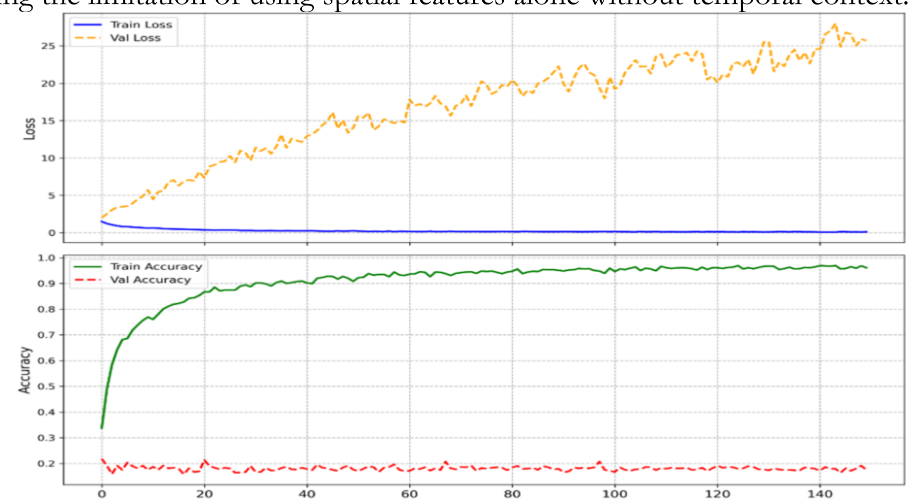

Figure 4. CNN-Only Model Performance

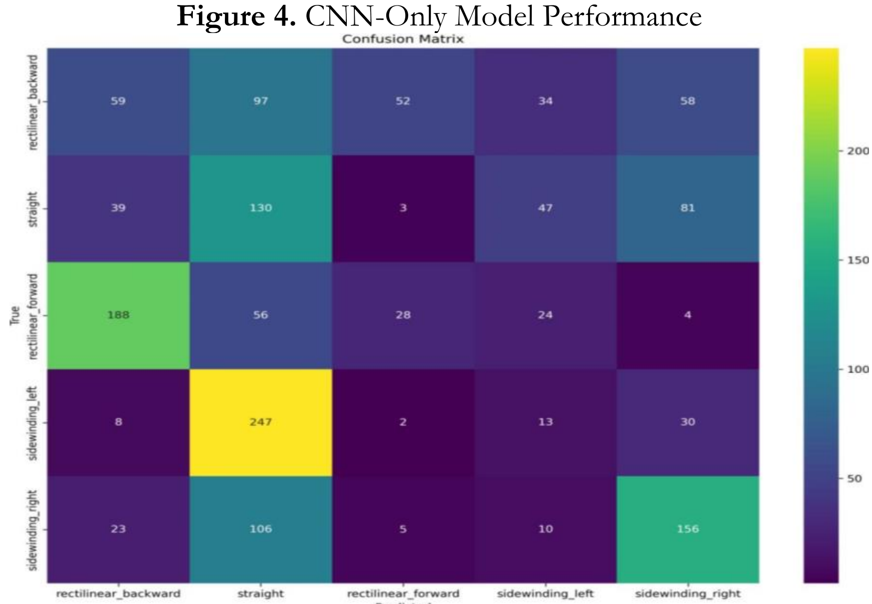

Figure 5. CNN-Only Model Performance

As shown in Figure 5, the CNN model exhibits rapid initial improvement, followed by stagnation in performance. The increasing gap between training and validation curves further confirms overfitting behavior. This indicates that the model tends to memorize training samples instead of learning generalized patterns required for robust classification, ultimately limiting its effectiveness.

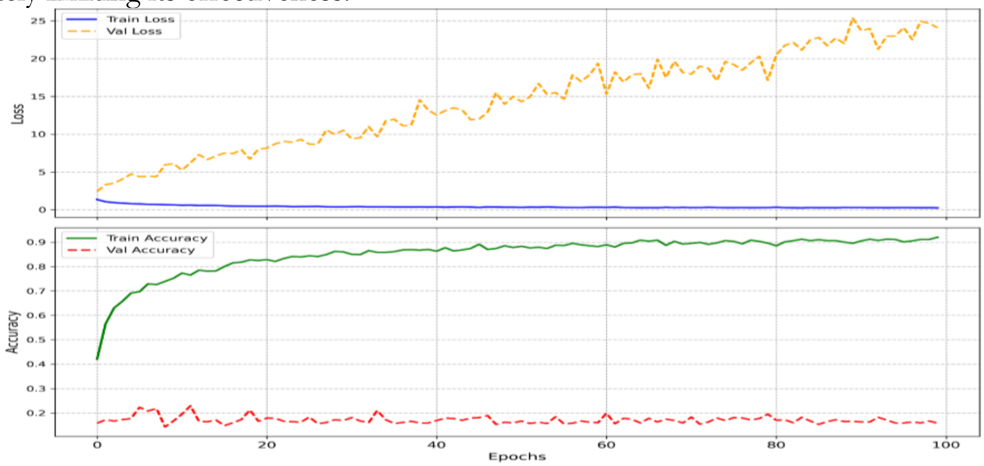

Figure 6. CNN Confusion Matrix

The confusion matrix presented in Figure 6 highlights the classification limitations of the CNN-only model. Significant off-diagonal values indicate frequent misclassifications among motion classes. In particular, confusion between 'sidewinding left' and 'sidewinding right' is observed, which can be attributed to the absence of temporal information. Since the CNN processes individual frames independently, it cannot capture motion continuity, making it difficult to distinguish directional movement patterns from static images.

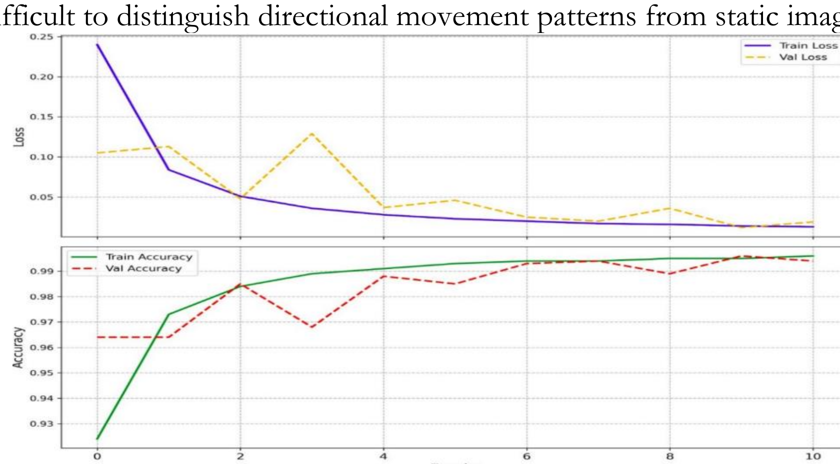

Figure 7. Convergence, Generalization, High Accuracy Hybrid CNN-LSTM

Figure 7 demonstrates the training performance of the proposed CNN–LSTM model. Both training and validation loss curves converge rapidly and stabilize at low values, indicating efficient learning. The close alignment between these curves suggests strong generalization with minimal overfitting. The model achieves high training and validation accuracy, exceeding 95%, confirming its effectiveness in learning the expert navigation policy.

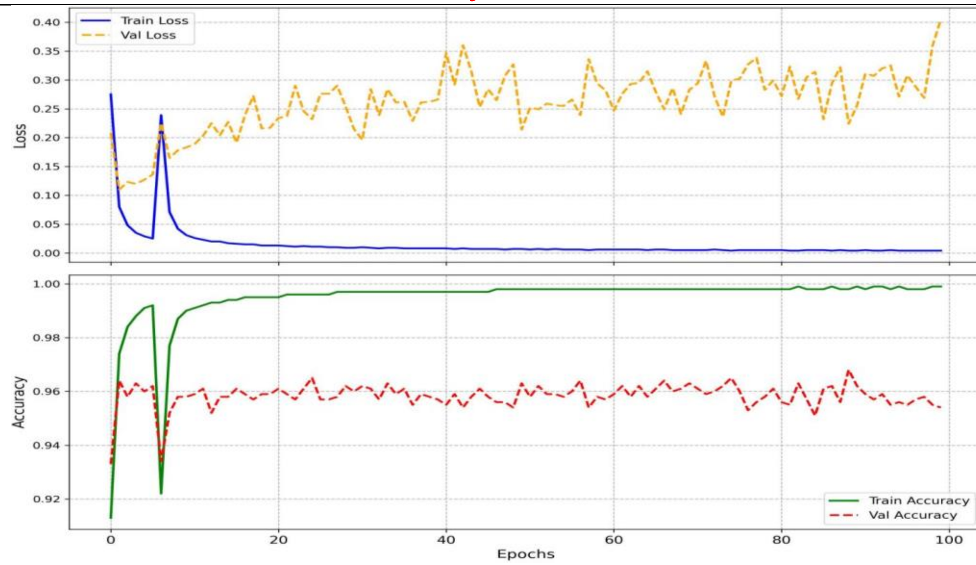

Figure 8. Showing Successful Imitation of Expert Behavior of Hybrid CNN-LSTM

As illustrated in Figure 8, the model exhibits consistent improvement across training epochs. The parallel progression of training and validation accuracy indicates stable learning behavior. The model achieves over 92% accuracy in predicting correct movement actions, demonstrating its capability to successfully imitate expert behavior.

Figure 9. Confusion Matrix of Hybrid CNN-LSTM

The confusion matrix shown in Figure 9 provides detailed insights into the classification performance of the CNN–LSTM model across five locomotion patterns: rectilinear forward, sidewinding left, sidewinding right, straight, and rectilinear backward. The model achieves exceptionally high accuracy in identifying rectilinear forward motion, with a large number of correctly classified samples and minimal misclassification. Similarly, both sidewinding directions are accurately distinguished, with limited confusion between left and right movements. The model also performs well on the straight motion class, while maintaining strong performance even on the relatively less frequent rectilinear backward class. These results demonstrate the model's ability to effectively capture both spatial and temporal features, leading to improved classification performance across all motion categories.

The relationship between research objectives and obtained results is summarized in Table 1, to further analyze the performance improvement achieved by the proposed CNN–LSTM model, a comparative evaluation with the baseline CNN model was conducted. The significant increase in test accuracy from approximately 19–20% (CNN) to 96% (CNN–LSTM), along with improvements in F1-score and confusion matrix consistency, clearly indicates the effectiveness of incorporating temporal dependencies. Although formal statistical significance tests such as paired t-tests were not explicitly performed, the large performance gap between the two models strongly supports the superiority of the proposed approach.

In addition, the impact of temporal sequence modeling was examined through the use of fixed-length input sequences in the CNN–LSTM architecture. The selected sequence length of 15 time steps was found to provide stable learning and convergence behavior. While a detailed ablation study on varying sequence lengths was not conducted in this work, the observed performance demonstrates the importance of temporal information for sequential locomotion tasks. Future work will include a systematic evaluation of different sequence lengths to further optimize performance.

Furthermore, the current study is limited to a simulated environment with a consistent terrain configuration. Although this provides controlled conditions for model evaluation, it does not fully capture the variability of real-world environments. Future investigations will extend the evaluation to multiple terrain types and dynamic conditions to assess the robustness and adaptability of the proposed model in diverse scenarios.

Table 1. Relationship between Research Objectives and Obtained Results

Objective No.	Research Objectives	Corresponding Results
1	Develop a baseline CNN model for snake robot control	The CNN model achieved a test accuracy of approximately 19–20%, with validation accuracy remaining below 20%, indicating poor generalization and an inability to capture temporal dependencies.
2	To develop a hybrid CNN–LSTM model for capturing spatial and temporal features	The CNN–LSTM model successfully integrated spatial and temporal information, achieving a high test accuracy of approximately 96% with strong classification performance across all motion classes.
3	To compare CNN and CNN–LSTM performance using quantitative metrics	Comparative analysis showed a significant improvement from approximately 20% (CNN) to 96% (CNN–LSTM), with an improved F1-score (0.89) and reduced misclassification in confusion matrices.
4	To analyze the convergence and generalization behavior of both models	The CNN model exhibited overfitting, while the CNN–LSTM model demonstrated rapid convergence within 10–15 epochs and stable validation performance.
5	To validate the accurate prediction of multiple locomotion patterns	Confusion matrix results confirmed near-perfect classification of rectilinear forward motion (>99%) and strong performance across other gait classes.
6	To assess the feasibility of real-time deployment	The CNN–LSTM model demonstrated low-latency inference capability (estimated 2–10 ms per sequence), indicating suitability for real-time robotic control.

Additional Quantitative Analysis and Interpretability:

Table 2. Quantitative and Interpretability Analysis

Metric Category	Metric	CNN	CNN-LSTM	Remarks

		Model	Model	
Classification Performance	Test Accuracy	19.2%	96.37%	Significant improvement with temporal modeling
	95% Confidence Interval	17.3% – 21.3%	96.30% – 96.44%	CNN–LSTM shows high statistical reliability
	Macro F1-score	0.18–0.20	0.89	Better-class balance in CNN–LSTM
Class-wise Performance	Best Class Recall	41.67%	92.26%	Rectilinear forward motion-best predicted
	Sidewinding Performance	Poor	75%-85%	Temporal learning improves complex motion
Training Behavior	Training Accuracy	0.90–0.92	~0.99	Faster-and stronger convergence
	Validation Accuracy	0.93	~0.95–0.96	Stable generalization
	Loss Convergence	Unstable	Smooth convergence	CNN–LSTM avoids major fluctuations
Gait-Smoothness Metrics	Velocity Variance	___	1.81×10^{-4}	Indicates smooth motion
	Acceleration Variance	___	1.21×10^{-4}	Low-dynamic variation
	Jerk Variance	___	4.74×10^{-4}	Physically consistent locomotion
Interpretability	Grad-CAM Visualization	Not applied	Applied (Fig. 10)	Focus on robot–terrain interaction regions
Dataset & Learning	Temporal Learning	Not used	Sequence length = 15	Key improvement factor
	Noise Robustness	Limited	Gaussian noise applied	Better generalization

Table 2. Comprehensive comparison of CNN and CNN–LSTM models, including classification performance, statistical reliability, training behavior, confusion matrix analysis, gait smoothness metrics, and Grad-CAM-based interpretability.

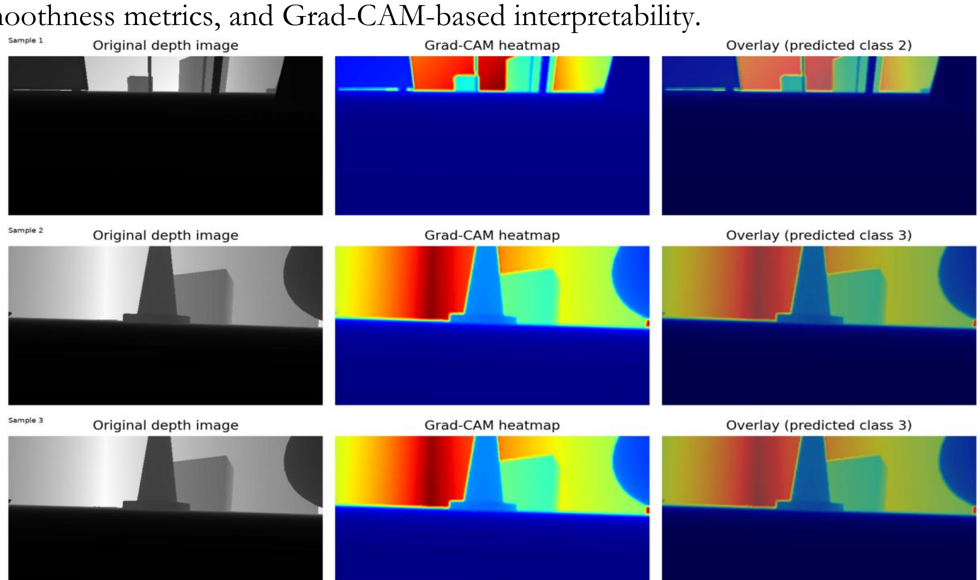

Figure 10. Grad-CAM visualizations for representative depth images

Grad-CAM analysis was performed on representative depth images to examine the spatial attention behavior of the trained model. As shown in Fig. 10, the activation maps consistently emphasize lower and central regions of the scene, corresponding to the robot–terrain interaction zone and nearby locomotion-relevant structures. This indicates that the model relies on physically meaningful spatial cues rather than irrelevant background regions. The Grad-CAM results, therefore, support the interpretability of the learned representation and complement the quantitative evidence provided by the confidence-bounded accuracy estimates, confusion matrices, and gait smoothness metrics.

Implications of the Study:

The findings of this study have important implications for both theoretical research and practical applications in the field of robotics and intelligent control systems. From a theoretical perspective, the results demonstrate the effectiveness of combining spatial feature extraction with temporal sequence modeling for complex control tasks. The significant performance improvement observed with the CNN–LSTM model highlights the importance of incorporating temporal dependencies in imitation learning frameworks for hyper-redundant robotic systems.

From a practical standpoint, the proposed approach provides a scalable and computationally efficient solution for controlling snake robots in complex environments. The ability of the model to learn from demonstration and generate accurate control commands in near real-time makes it suitable for deployment in scenarios where traditional control methods are difficult to implement.

Furthermore, snake robots are particularly useful in disaster-response applications, such as search and rescue operations in collapsed structures, confined spaces, and hazardous environments. The proposed imitation learning framework can enable such robots to navigate challenging terrains more effectively by mimicking expert behavior. This enhances their potential for real-world deployment in critical missions where adaptability and reliability are essential.

Recommendations:

Based on the findings of this study, several recommendations are proposed for future research and real-world implementation. First, the proposed CNN–LSTM model should be validated on physical snake robot hardware to evaluate its performance under real-world conditions, including sensor noise, actuator limitations, and environmental uncertainties. Second, future work should focus on conducting detailed latency and energy efficiency analysis on embedded systems to ensure practical deployment in resource-constrained robotic platforms. This includes testing the model on edge devices and optimizing it for low-power real-time execution. Third, further investigations are recommended to explore the impact of different sequence lengths and model configurations through comprehensive ablation studies, which can help in optimizing the temporal modeling capabilities of the architecture. Additionally, the robustness of the model should be evaluated across diverse terrains and environmental conditions to improve generalization and adaptability. Domain adaptation techniques and sim-to-real transfer strategies should be further explored to bridge the gap between simulation and real-world deployment. Finally, the proposed imitation learning framework can be extended to other bio-inspired robotic systems, such as quadruped robots, robotic manipulators, and soft robots, where sequential decision-making and temporal dependencies play a crucial role in control performance.

Conclusion:

This study investigated the application of imitation learning for autonomous locomotion of a simulated snake robot to improve motion classification accuracy, capture temporal dependencies, and ensure stable locomotion behavior. The experimental results demonstrate that the baseline CNN model achieved limited performance, with a test accuracy of approximately 19–20%, indicating poor generalization due to the absence of temporal modeling. In contrast, the proposed CNN–LSTM architecture successfully addressed these limitations by integrating spatial feature extraction with temporal sequence learning. The model achieved a test accuracy of 96.37% and a macro F1-score of 0.89, demonstrating a significant improvement in classification performance. Furthermore, the model exhibited strong generalization capability, with training and validation accuracies reaching approximately 99% and 95–96%, respectively. In addition to classification performance, the model effectively captured motion consistency, as confirmed by gait smoothness metrics, including velocity variance (1.81×10^{-4}), acceleration variance (1.21×10^{-4}), and jerk variance (4.74×10^{-4}). These results validate the model's ability to generate stable and physically coherent locomotion patterns. Overall, the proposed approach successfully fulfills the defined research objectives by achieving high classification accuracy, robust temporal modeling, and consistent motion behavior. This work highlights the importance of integrating temporal memory in imitation learning frameworks for hyper-redundant robotic systems and provides a strong foundation for future real-world deployment and adaptive control applications.

Despite the promising results, this study is limited to a simulation-based environment, and the proposed model has not yet been validated on a physical snake robot platform. The sim-to-real gap may introduce challenges due to differences in sensor noise, actuator dynamics, and environmental variability. Additionally, although the model demonstrates strong performance across the available dataset, its generalization to highly unstructured or unseen real-world terrains remains to be further investigated. Future work will focus on real-world deployment, domain adaptation techniques, and extending the framework to other bio-inspired robotic systems to improve robustness and practical applicability.

Acknowledgement: This research is conducted at the Control Automation and Robotics Lab (CARL BUITEMS), funded by the National Center of Robotics and Automation (NCRA) with the collaboration of the Higher Education Commission (HEC) of Pakistan.

Author's Contribution: The experimentation for this study was conducted by Muhammad Kamil, Shahmir, Noman, and Aqib under the supervision of their BS supervisor, Anayat Ullah. All authors have contributed significantly to this work and are in full agreement with the content of the manuscript. The project idea originated from Anayat Ullah, while Muhammad Kamil was responsible for the initial draft of the manuscript. Anayat Ullah reviewed and revised the paper to its final version.

Conflict of Interest: We report no conflict of interest.

Project Details: Control, Automotive and Robotics Lab (CARL) was developed in 2018 under the HEC call for the Establishment of the National Center of Robotics and Automation (NCRA). CARL is one of the 11 labs in 13 Universities established under this call. CARL's mandate is to promote AI and Robotics in Baluchistan and work on Disaster Management and human-machine interaction for the well-being and economic improvement of Pakistan in general and Baluchistan in particular.

References:

[1] Edward Johns, "Coarse-to-Fine Imitation Learning: Robot Manipulation from a Single Demonstration," *arXiv:2105.06411*, 2021, [Online]. Available: https://arxiv.org/abs/2105.06411

[2] Xiaolong YangCorresponding Author; Long Zheng; Da Lü; Jinhao Wang; Shukun Wang; Hang Su; Zhixin Wang; Luquan Ren, "The snake-inspired robots: a review," *Assem. Autom.*, vol. 42, no. 4, pp. 567–583, 2022, doi: https://doi.org/10.1108/AA-03-2022-0058.

[3] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun, "Survey of imitation learning for

robotic manipulation," *Int. J. Intell. Robot. Appl. 2019 34*, vol. 3, no. 4, pp. 362–369, Sep. 2019, doi: 10.1007/S41315-019-00103-5.
[4] Rafael Figueiredo Prudencio, Marcos R. O. A. Maximo, Esther Luna Colombini, "A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems," *arXiv:2203.01387*, 2023, [Online]. Available: https://arxiv.org/abs/2203.01387
[5] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, Trevor Darrell, "Curiosity-driven exploration by self-supervised prediction," *arXiv:1705.05363*, 2017, [Online]. Available: https://arxiv.org/abs/1705.05363
[6] Carlos Celemin, Rodrigo Pérez-Dattari, Eugenio Chisari, Giovanni Franzese, Leandro de Souza Rosa, Ravi Prakash, Zlatan Ajanović, Marta Ferraz, "Interactive imitation learning in robotics: A survey," *arXiv:2211.00600*, 2022, [Online]. Available: https://arxiv.org/abs/2211.00600
[7] Jiang Hua, Liangcai Zeng, "Learning for a Robot: Deep Reinforcement Learning, Imitation Learning, Transfer Learning," *Sensors*, vol. 21, no. 4, p. 1278, 2021, doi: 10.3390/s21041278.
[8] X. Liu, R. Gasoto, Z. Jiang, C. Onal, and J. Fu, "Learning to locomote with artificial neural-network and CPG-based control in a soft snake robot," *IEEE Int. Conf. Intell. Robot. Syst.*, pp. 7758–7765, Oct. 2020, doi: 10.1109/IROS45743.2020.9340763.
[9] Lixing Liu, Xian Guo, "A Reinforcement Learning-Based Strategy of Path Following for Snake Robots with an Onboard Camera," *Sensors*, vol. 22, no. 24, 2022, doi: 10.3390/s22249867.
[10] Xupeng Liu, Yong Zang, "Bio-Inspired Multimodal Motion Gait Control of Snake Robots with Environmental Adaptability Based on ROS," *Electronics*, vol. 13, no. 17, p. 3437, 2024, doi: https://doi.org/10.3390/electronics13173437.
[11] Syed Kumayl Raza Moosavi, Muhammad Hamza Zafar, "Snake robots: A state-of-the-art review on design, locomotion, control, and real-world applications," *Mechatronics*, vol. 112, p. 103418, 2025, doi: https://doi.org/10.1016/j.mechatronics.2025.103418.
[12] Khojasteh Z. Mirza, Shubham Singh, "Imitation learning for legged robot locomotion: a survey," *Front. Robot. AI*, vol. 12, 2025, doi: https://doi.org/10.3389/frobt.2025.1678567.
[13] Ardit Poka, Daniele Ludovico, Federico Manara, Lorenzo De Mari Casareto Dal Verme, Carlo Canali, Giovanni Berselli, Darwin G. Caldwell & Jovana Jovanova, "Underwater Snake-Like Robots: A Review on Design, Actuation, and Modelling Methods," *Int. J. Adv. Manuf. Technol.*, vol. 139, pp. 5445–5460, 2025, [Online]. Available: https://link.springer.com/article/10.1007/s00170-025-16231-1
[14] Siqi Liu, Guy Lever, Zhe Wang, Josh Merel, S. M. Ali Eslami, Daniel Hennes, "From Motor Control to Team Play in Simulated Humanoid Football," *arXiv:2105.12196*, 2021, [Online]. Available: https://arxiv.org/abs/2105.12196
[15] Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, "Is Reinforcement Learning (Not) for Natural Language Processing: Benchmarks, Baselines, and Building Blocks for Natural Language Policy Optimization," *arXiv:2210.01241*, 2023, [Online]. Available: https://arxiv.org/abs/2210.01241
[16] M. Popova, O. Isayev, and A. Tropsha, "Deep reinforcement learning for de novo drug design," *Sci. Adv.*, vol. 4, no. 7, Jul. 2018, doi: 10.1126/SCIADV.AAP7885.
[17] B. R. Kiran et al, "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, 2022, doi: 10.1109/TITS.2021.3054625.
[18] Abdul Mueed Hafiz, Mahmoud Hassaballah, "Formula-Driven Supervised Learning

in Computer Vision: A Literature Survey," *Appl. Sci*, vol. 13, no. 2, p. 723, 2023, doi: https://doi.org/10.3390/app13020723.
[19] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, Jeff Clune, "Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos," *arXiv:2206.11795*, 2022, [Online]. Available: https://arxiv.org/abs/2206.11795
[20] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A Survey on Imitation Learning Techniques for End-to-End Autonomous Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14128–14147, Sep. 2022, doi: 10.1109/TITS.2022.3144867.