

Article

A Vision–Locomotion Framework Toward Obstacle Avoidance for a Bio-Inspired Gecko Robot

Wenrui Xiang ^{1,2} , Barmak Honarvar Shakibaei Asli ^{2,*}  and Aihong Ji ^{1,*} ¹ College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; wenruixiang@nuaa.edu.cn² Centre for Life-Cycle Engineering and Management, Faculty of Engineering and Applied Sciences, Cranfield University, Cranfield, Bedfordshire MK43 0AL, UK

* Correspondence: barmak@cranfield.ac.uk (B.H.S.A.); meeahji@nuaa.edu.cn (A.J.)

Abstract

This paper presents the design and experimental evaluation of a bio-inspired gecko robot, focusing on mechanical design, vision-based obstacle perception, and rhythmic locomotion control as enabling technologies for future obstacle avoidance in complex environments. The robot features a 17-degrees-of-freedom mechanical structure with a flexible spine and multi-jointed limbs, providing a physical basis for adaptive locomotion. For perception, a custom obstacle detection dataset was constructed from the robot's onboard camera view and used to train a YOLOv5-based detection model. Experimental results show that the trained model achieves a mean average precision (mAP) of 0.979 and a maximum F1-score of 0.97 at an optimal confidence threshold, demonstrating reliable real-time obstacle perception under diverse indoor conditions. For motion control, a central pattern generator (CPG) based on Hopf oscillators is implemented to generate rhythmic locomotion. Experimental evaluations confirm stable diagonal gait generation, with coordinated joint trajectories oscillating at 1 Hz. The flexible spine exhibits periodic lateral deflection with peak amplitudes of $\pm 15^\circ$, $\pm 10^\circ$, and $\pm 8^\circ$ across spinal joints, enhancing locomotion continuity and turning capability. Physical robot experiments further demonstrate smooth straight-line crawling enabled by the coupled limb–spine motion. While visual perception and CPG-based locomotion are experimentally validated as independent subsystems, their real-time closed-loop integration is not implemented in this study. Instead, this work establishes a system-level framework and experimental baseline for future perception–motion coupling, providing a foundation for closed-loop obstacle avoidance and autonomous navigation in bio-inspired gecko robots.

Keywords: bio-inspired robotics; gecko robot; YOLOv5; CPG control; obstacle detection; flexible spine



Academic Editors: Ulises Orozco-Rosas and Oscar Montiel Ross

Received: 12 January 2026

Revised: 14 February 2026

Accepted: 16 February 2026

Published: 20 February 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Robots can replace humans to complete many highly repetitive and dangerous tasks, ensuring the safety of life and property while ensuring the smooth completion of tasks and reducing operating costs [1]. Since the 21st century, robotics has not only played a crucial role in the industrial sector but has also gradually expanded into various fields, including education, healthcare, smart homes, and other aspects of daily life. At the same time, robots have also shown their irreplaceable role in special scenarios such as disaster relief, spacecraft maintenance, and extraterrestrial exploration.

However, the ability to move in complex terrain still faces significant challenges, and traditional motion control systems are difficult to adapt to the needs of unstructured environments. In high-altitude operations such as bridge safety inspections, high-rise curtain wall cleaning, and large ship wall flaw detection, robots need to be able to attach to vertical surfaces or even ceilings for barrier-free climbing, which is almost impossible for ordinary ground robots [2].

In recent years, wall-climbing robots based on bionic adhesion and intelligent drive have emerged as a research hotspot for special mobile robots. They have three-dimensional motion capabilities and can autonomously attach and move on complex surfaces such as slopes, vertical walls, and inverted ceilings. Their core value lies in replacing manual labor with automated systems to perform high-risk tasks, such as nuclear power plant inspections and high-rise building maintenance, which not only improves efficiency but also significantly reduces the risk of falls and exposure to toxic environments.

As an important branch of the bionic field, the bio-inspired wall-climbing robot, with the giant gecko as a typical biological prototype, has attracted widespread attention due to its high adaptability and flexibility in complex environments. The van der Waals adhesion mechanism of its feet, the coupling of bone topology optimization and movement coordination mechanism, make it the largest known free-climbing organism. Research has shown that geckos use the periodic lateral bending of the spine to cause flexion and extension of the trunk in the sagittal plane, and coordinate the forward and backward swings of the limbs to achieve greater movement speed [3]. It improves motion stability and energy efficiency in complex wall environments, providing a highly adaptable solution for scenarios such as high-altitude detection and disaster rescue.

However, it is still challenging for bio-inspired gecko robots to achieve autonomous obstacle avoidance in complex environments due to limitations in real-time perception and dynamic motion control. Although significant progress has been made in the independent development of vision-based obstacle detection systems or bio-inspired motion controllers, their integration into a unified framework for gecko robots has not yet been fully achieved. This research gap provides an opportunity to combine state-of-the-art computer vision techniques with bionic control strategies to advance the field. Accordingly, the core research problem addressed in this study is not solely detection accuracy or locomotion stability, but the system-level formulation of a perception–motion integration framework suitable for gecko-inspired climbing robots.

2. Literature Review

This section provides a comprehensive survey of existing research relevant to bio-inspired climbing robots, with a focus on three key areas: bio-inspired gecko robots, biomimetic locomotion control, and vision-based obstacle avoidance.

2.1. Bio-Inspired Gecko Robots

The RiSE project conducted early systematic research on bio-inspired wall-climbing robots [4]. Boston Dynamics developed multiple RiSE robot generations featuring six-legged claw-based designs and progressively improved stability and climbing performance on rough vertical surfaces [5]. The Stickybot series developed by Stanford University demonstrated gecko-inspired vertical climbing with improved speed, but adhesion stability on inverted surfaces remains limited [6–8]. Ji Aihong's team developed quadruped and six-legged claw-based wall-climbing robots, demonstrating stable adhesion and omnidirectional locomotion on rough surfaces [9,10]. In 2020, Johanna et al. from the University of the Sunshine Coast in Australia developed a modular gecko-like robot X4 [11]. The robot foot end uses a claw lock to ensure gripping on the moving surface.

2.2. Flexible Spine in Legged Robots

Flexible spine mechanisms have been widely explored to enhance locomotion performance in bio-inspired robots. In 2012, Zhao Qian's team developed Kitty, a quadruped robot with a silicone–plastic flexible spine that improved motion agility [12]. In the same year, MIT introduced the Cheetah robot, which achieved enhanced dynamic running through spine–hind leg coupling [13]. In 2013, the modular Salamandra Robotica developed at EPFL combined an actively driven flexible spine with legged locomotion for amphibious movement [14], while the Pleurobot further optimized multi-chain spine–limb coordination with high-degree-of-freedom design [15]. More recently, NUAU proposed gecko-inspired robots with high-DOF legs and flexible spines, including the Slalom robot [16] and a shape-memory-alloy-driven design, further advancing spine-enabled gecko locomotion. In 2024, Paul et al. designed a bio-inspired robot LORIS, with a degree of longitudinal bending freedom that can crawl on uneven ground [17].

2.3. Motion Control Strategy of Bio-Inspired Gecko Robot

Throughout biological evolution, animals have developed specialized morphological structures and locomotion mechanisms to adapt to diverse environments [18–20]. Among them, climbing species such as geckos exhibit remarkable adhesion and locomotion capabilities, providing valuable inspiration for bio-inspired robotic design [21,22]. Bio-inspired robotic locomotion requires coordinated control of multiple joints and degrees of freedom, which has been addressed through hierarchical, hybrid, neural, and learning-based control strategies [23]. From early sensor-integrated architectures [24,25] to CNN-based parallel control [26,27] and recent deep reinforcement learning methods, modern systems increasingly rely on multimodal sensing for adaptive locomotion. CPG-based control has been shown to effectively generate stable gecko-inspired climbing gaits, as demonstrated by Kim et al. in the Stickybot robot [28]. Cai et al. proposed a motion control method for a gecko-like robot based on a central pattern generator (CPG). CPG uses the Matsuoka oscillator model to generate rhythmic signals through a mutually inhibiting neuronal network to simulate the periodicity of biological motion [29]. In 2019, Arthicha, Shao et al. proposed a multi-gait generation and adaptation algorithm for a gecko-like robot based on neural control, which enabled the robot to climb efficiently on different slopes through a central pattern generator (CPG) and dynamic feedback [30]. The control consists of three main components: a central pattern generator (CPG) for generating various rhythmic patterns, CPG post-processing for shaping the CPG signals, and a delay line for transmitting the shaped CPG signals to drive the legs of the gecko robot [30]. Pei et al. proposed a bio-inspired adaptive control strategy for gecko-like robots in space stations, combining rigid-flexible coupling and real-time force feedback. The system dynamically adjusts the impedance parameters to enable stable climbing, compliant detachment, and disturbance rejection in microgravity, demonstrating precise motion control and robust performance for space applications [31]. In 2023, Sun proposed a self-organizing gait control method based on distributed local CPG and adaptive feedback. This method achieves fast, robust and repeatable self-organizing gait generation through physical and neural communication mechanisms and has gait memory capabilities [32]. In 2025, Zhu et al. proposed a torso-leg coordinated motion control method based on parallel mechanism. This study adopted the model-driven control idea and regarded the active torso as the core coordination unit in the robot motion system [33].

2.4. YOLOv5 in Obstacle Avoidance Robots

YOLOv5 is designed based on the procedures. As a computer vision model for object detection that is built upon the YOLO series [34]. YOLOv5 includes several architectural improvements for simultaneously enhancing speed and accuracy [35,36].

Wang et al. proposed a more effective obstacle avoidance method for autonomous driving based on the YOLOv5 monocular vision system. The study used the YOLOv5 model to detect obstacles and landmarks in the environment in real time, including vehicles, pedestrians, traffic lights, etc., identify objects of different sizes and angles, and provide accurate obstacle avoidance decision input [37]. Yang et al. proposed a visual SLAM algorithm based on YOLOv5 to recognize environmental images, distinguish dynamic targets in the scene, eliminate their dynamic feature points, and then use the remaining static feature points for adversarial geometric constraints to improve the accuracy of the system [38]. In 2024, Liu et al. introduced a YOLOv5 algorithm based on the attention mechanism to enhance the expression ability of small target channel features and spatial features, thereby realizing dynamic obstacle avoidance in robot navigation control [39]. In 2024, Ndidiama Adiuku et al. proposed integrating YOLOv5 into the ROS navigation stack, converting detection results into laser scanning data through the OAK-D camera to optimize dynamic obstacle avoidance [40]. In 2025, Abood et al. proposed a lightweight target detection method based on YOLOv5, which is suitable for resource-constrained robot platforms [41].

2.5. Related Work

Significant progress has been achieved in the development of bio-inspired climbing robots over the past two decades. Early representative systems such as RiSE and Stickybot demonstrated the feasibility of gecko-inspired adhesion and wall-climbing mechanisms through bio-mimetic structural design and claw-based attachment strategies. Subsequent research further improved locomotion stability, structural flexibility, and actuation efficiency, enabling robots to achieve vertical and even inverted climbing on rough surfaces. More recent studies have incorporated flexible spines, multi-degree-of-freedom limbs, and modular architectures to enhance maneuverability and adaptability in unstructured environments. However, most of these works primarily focus on mechanical design and locomotion control, with limited integration of environmental perception for adaptive decision-making.

In parallel, vision-based obstacle detection has advanced rapidly with the development of deep learning-based object detection algorithms. One-stage detectors such as the YOLO series have achieved real-time performance with high detection accuracy, making them widely adopted in mobile robotic navigation systems. These models have been successfully deployed in ground vehicles and wheeled robots operating in structured environments. Nevertheless, existing applications are predominantly designed for horizontally moving platforms and do not specifically address the sensing characteristics and viewpoint constraints of gecko-inspired climbing robots, which operate under varying orientations, lighting conditions, and surface geometries.

Regarding locomotion control, Central Pattern Generator (CPG)-based approaches have been widely applied in bio-inspired robotics due to their ability to produce stable and coordinated rhythmic gaits. Hopf oscillator-based CPG models, in particular, offer smooth trajectory generation and parameter modulation capabilities. Although CPG-based controllers enable robust gait generation, they are typically designed as open-loop or predefined rhythmic systems. The coupling between high-level visual perception and low-level rhythmic motion generation remains insufficiently explored, especially in the context of climbing robots operating in complex environments.

Overall, while substantial advancements have been made in (1) bio-inspired climbing mechanisms, (2) deep learning-based visual detection, and (3) rhythmic locomotion control, these components are often investigated independently. A systematic framework that bridges robot-centric visual perception with CPG-based locomotion control for gecko-inspired robots in complex climbing environments is still lacking. Addressing this gap requires a system-level perspective that considers mechanical design, perception, and motion control as coordinated elements rather than isolated modules.

2.6. Research Gaps and Contribution Summary

Although substantial progress has been made in climbing robot mechanics, vision-based detection, and CPG-based locomotion control, these components are often investigated independently. A systematic framework that integrates robot-centric perception with rhythmic locomotion generation for gecko-inspired robots in complex climbing environments remains insufficiently explored.

Most existing studies on bionic gecko robots primarily focus on mechanical structure, adhesion mechanisms, and locomotion control, while obstacle perception and autonomous interaction with complex environments are relatively less explored. In parallel, vision-based obstacle avoidance methods have achieved remarkable success in ground mobile robot navigation systems; however, their application in bionic gecko robots is still almost blank.

Although more advanced deep learning models with higher benchmark accuracy have been proposed in recent years, their deployment on mobile robotic platforms remains challenging due to high computational cost, memory consumption, and inference latency.

In the context of gecko-inspired climbing robots, onboard computing resources are limited, and real-time perception is essential for future closed-loop control. Therefore, the objective of visual perception in this study is not to achieve state-of-the-art detection accuracy, but to provide reliable and low-latency obstacle information suitable for system integration.

YOLOv5 is selected due to its favorable trade-off between detection accuracy and real-time performance, its lightweight network variants, and its mature engineering ecosystem. Moreover, its simple output structure facilitates direct coupling with motion control modules, which is critical for perception–motion integration.

Furthermore, most existing studies focus on horizontal, structured, and relatively stable ground environments, whereas bionic gecko robots typically operate in complex and unstructured climbing scenes. These environments are often characterized by variable illumination conditions, diverse surface roughness, and irregular obstacle distributions.

Most YOLO-based obstacle avoidance research concentrates on detecting ground targets such as vehicles and pedestrians, and dedicated datasets or perception strategies for gecko robot working environments are still lacking. As a result, current perception models are not well adapted to the visual perspective and environmental characteristics of climbing robots.

In addition, there is a notable lack of effective integration between YOLO-based visual perception modules and bionic gecko robot motion control systems. Most existing works investigate visual perception or locomotion control independently, while few studies establish an explicit mapping between visual obstacle information and rhythmic motion generation mechanisms such as central pattern generators (CPGs). As a result, perception–motion closed-loop integration for gecko-like robots in complex climbing environments remains largely unexplored.

Although CPGs and other biologically inspired neural control methods have been widely used to generate rhythmic locomotion in gecko robots, they are rarely linked in real time with external visual feedback mechanisms. This limitation reduces the adaptability of gecko robots when operating in unknown or obstacle-intensive environments and restricts their practical deployment in scenarios such as high-altitude inspection and disaster rescue.

To address these gaps, this paper focuses on the system-level framework and enabling technologies required for perception–motion integration. Rather than treating mechanical design, vision perception, and motion control as isolated modules, this work emphasizes their coordinated design and interaction.

To clearly position this work within the existing literature, the main contributions are summarized as follows: The main scientific contributions of this paper are summarized as follows:

- A bio-inspired gecko robot system is designed, integrating mechanical structure, vision-based perception, and rhythmic locomotion control, providing a system framework toward obstacle avoidance in complex climbing environments.
- A vision-based obstacle detection framework is developed using a custom dataset collected from the robot's onboard camera, addressing the lack of perception models tailored to gecko-like climbing robots.
- A Central Pattern Generator (CPG) control strategy based on Hopf oscillators is implemented to generate stable diagonal gaits and coordinated flexible-spine motions, enabling smooth and adaptive locomotion at the subsystem level.
- A qualitative perception–locomotion coupling framework is formulated, in which visual obstacle information is conceptually mapped to the modulation of CPG parameters, providing a biologically inspired pathway toward future closed-loop autonomous navigation.
- Extensive experimental evaluations are conducted, including detection performance analysis and physical robot experiments, to validate the effectiveness of the proposed subsystems.

It should be noted that this study focuses on the development and experimental validation of the core components required for obstacle avoidance, namely visual perception and CPG-based locomotion generation. The full closed-loop integration between perception, decision-making, and motion execution is presented as a conceptual framework and will be realized in future work.

3. Methodology

This section presents the systematic approach adopted in this study to develop a bio-inspired gecko robot capable of obstacle avoidance framework in complex environments. The methodology integrates three key components: mechanical design, vision-based perception, and motion control.

3.1. Design of Robot System

This section describes the overall mechanical design of the gecko-inspired robot, which provides the physical basis for implementing rhythmic locomotion and obstacle avoidance. The design includes the structure of the limbs and body, a flexible spine mechanism, and a simplified kinematic model. The first part focuses on the layout and degrees of freedom of the robot's legs and torso. Then, a flexible spine is introduced to enhance body mobility, allowing the robot to perform bending motions that assist in turning and climbing. Lastly, a basic kinematic model of a single leg is presented to support the later development of control and motion planning.

3.1.1. Mechanical Structure Design of Limbs and Body

Based on the biomechanical analysis of geckos, a 3D model of the gecko robot will be designed in the SolidWorks 2024 software.

The front and rear legs of the gecko-like robot use the same mechanical structure. Each leg is designed with a series linkage mechanism and contains three motion joints with three degrees of rotational freedom.

The overall layout and geometric proportions of the gecko-like robot designed in this paper closely resemble the body features of a gecko (see Figure 1). The entire robot has 17 degrees of freedom (3 degrees of freedom for each leg and 5 degrees of freedom for the flexible spine). The figure shows the robot in a crawling posture. The total body length of the gecko-like robot is 650 mm, the width is 320 mm in crawling posture, and the height is 52 mm. The length of the robot spine is 200 mm. The front end of the flexible spine is connected to the shoulder belt, and the rear end is connected to the pelvic belt, which is responsible for connecting with the head and tail of the gecko robot.

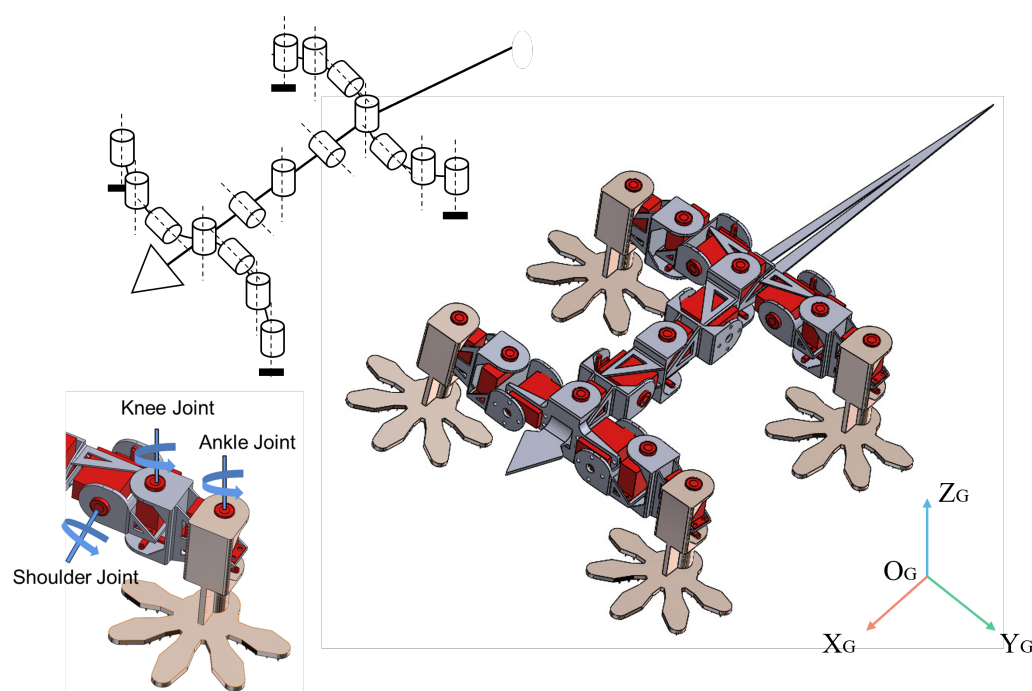


Figure 1. Robot legs and overall structural design.

3.1.2. Structural Design of Flexible Spine

The robot spine design includes three lateral swing degrees of freedom and two longitudinal swing degrees of freedom. As shown in Figure 2a, motors 1, 3, and 5 control the change of the horizontal lateral swing angle between the spine modules. The connecting parts between the motors are regarded as connecting rods, and the motor shafts are perpendicular to the horizontal plane where the flexible spine is located. Two motors for controlling longitudinal swing are added between the three motors for controlling lateral swing, as shown in Figure 2b, namely motor 2 and motor 4, and the shafts of these two motors are parallel to the horizontal plane where the flexible spine is located. The modular serial flexible spine can swing sideways in the horizontal plane and bend in the vertical plane. The length of the robot spine is 200 mm. The front end of the flexible spine is connected to the shoulder belt, and the rear end is connected to the pelvic belt, which is responsible for connecting with the head and tail of the gecko robot.

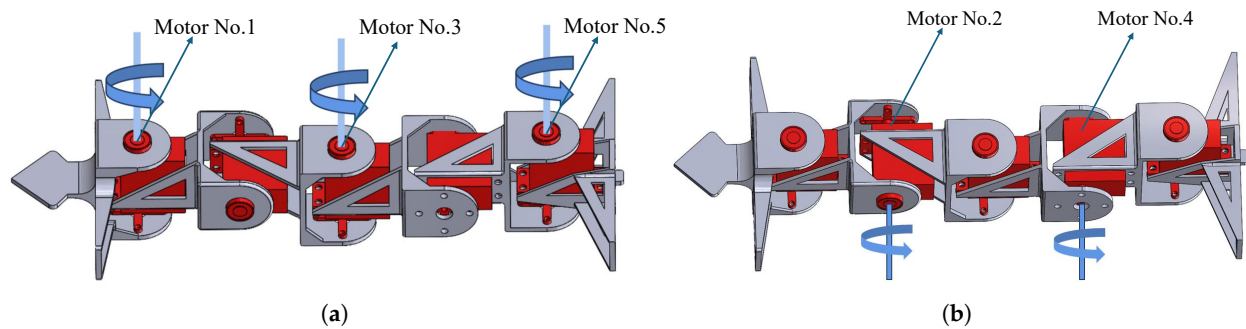


Figure 2. Structural Design of Flexible Spine. (a) the motor that controls the lateral swing; (b) the motor that controls the longitudinal swing.

3.1.3. Kinematic Model of a Single-Leg Robot

The kinematic analysis of this study is mainly based on forward kinematics, which verifies the motion coordination of the flexible spine and leg mechanism. The forward kinematic analysis of the robot's leg mechanism begins with establishing coordinate systems for each joint according to the D-H convention. As shown in Figure 3, the leg consists of three revolute joints with axes arranged to mimic biological gecko limbs. The robot's leg mechanism base coordinate system is located at the connection between the shoulder joint motor and the robot's body trunk. The foot tip mass p is located at the center of the sole of the foot. The three rotation joints of each leg of the robot are represented by j_1 , j_2 , and j_3 , respectively. The angle variables corresponding to each joint are represented by θ_1 , θ_2 , and θ_3 . The length of the connecting rod between the shoulder joint and the knee joint is l_1 , the length of the connecting rod between the knee joint and the ankle joint is l_2 , and the length of the connecting rod between the ankle joint and the foot end is l_3 ($l_1 = 22$ mm, $l_2 = 36$ mm, $l_3 = 28$ mm).

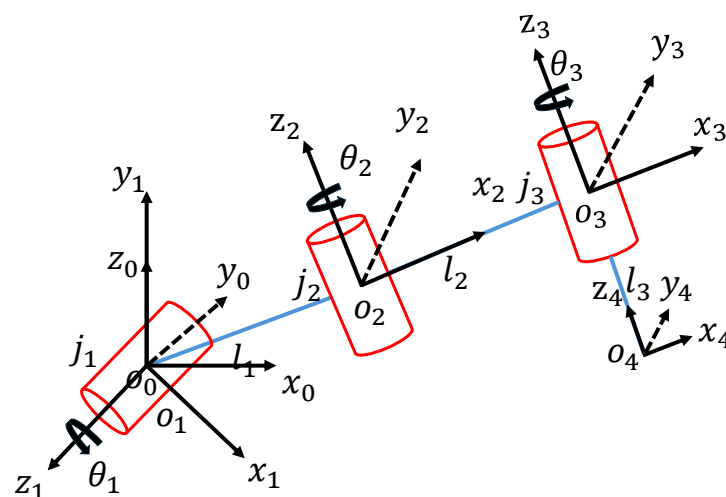


Figure 3. Coordinate system of robot's leg mechanism.

The D-H parameters are determined on the basis of the structural parameters of the single leg of the robot, as shown in Table 1.

The transformation matrix of the robot foot end coordinate system ${}^0_4T$ relative to the leg mechanism base coordinate system is:

$${}^0_4T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T, \quad (1)$$

where ${}^j_iT$ denotes the homogeneous transformation matrix from frame j to frame i .

Table 1. D-H parameter table of the robot leg mechanism.

j	θ_i (rad)	α_{i-1} (rad)	a_{i-1} (mm)	d_i (mm)
1	θ_1	$\pi/2$	0	0
2	θ_2	$\pi/2$	l_1	0
3	θ_3	0	l_2	0
4	θ_4	0	0	$-l_3$

The transformation matrix between every two adjacent coordinate systems is:

$$\begin{aligned}
 {}^0_1T &= \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \cos \theta_1 & \sin \theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1_2T = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ -\sin \theta_2 & -\cos \theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 {}^2_3T &= \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & l_2 \\ \sin \theta_3 & \cos \theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^3_4T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},
 \end{aligned} \quad (2)$$

According to the above formula, the transformation matrix of the robot foot end coordinate system ${}^0_4T$ relative to the leg mechanism base coordinate system is:

$${}^0_4T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

The position vector of the foot end mass point w_p relative to the leg mechanism base coordinate system is:

$$w_p = \begin{bmatrix} l_1 \cos \theta_1 + l_3 \sin \theta_1 + l_2 \cos \theta_1 \cos \theta_2 \\ l_2 \sin \theta_2 \\ l_1 \sin \theta_1 - l_3 \cos \theta_1 + l_2 \cos \theta_2 \sin \theta_1 \end{bmatrix}. \quad (4)$$

The above formula can be used to obtain the correct solution for the single-leg kinematics of the gecko-like robot. If the joint angle variables of the robot leg mechanism at a certain moment are known, the pose transformation matrix of the robot foot end coordinate system relative to the leg mechanism base coordinate system can be obtained through Equation (4) and the D-H parameter, and the position and posture of the foot end mass point in the global coordinate system can be derived. Through forward kinematics, the leg working space of the gecko-like robot can be obtained.

3.2. Vision-Based Obstacle Detection

This section describes the development and implementation of a real-time visual perception system based on the YOLOv5 object detection framework. A custom obstacle detection dataset is constructed from the robot's onboard camera view and manually labeled. The section details the training procedures, and deployment process. The perception module provides critical environmental awareness that enables the robot to dynamically adapt its locomotion strategy to avoid obstacles.

3.2.1. Dataset Collection

This study did not use any public datasets. Public benchmark datasets such as COCO or Pascal VOC are primarily designed for ground mobile robots and general object detection tasks. Due to the unique visual perspective, climbing posture, and close-range obstacle characteristics of gecko-inspired robots, these datasets are not directly applicable to the scenario investigated in this study. Therefore, a dedicated dataset was collected to reflect the robot's actual operating conditions, including tilted camera viewpoints, uneven illumination, and environment-specific obstacles. The obstacle detection dataset was constructed based on the actual motion perspective of the gecko robot. A wide-angle camera (Intel RealSense Depth Camera D405, sourced from Intel in Cupertino, CA, USA) is installed on the head of the robot to simulate the field of view of the gecko during movement. The robot moved in a lab environment with multiple obstacles, capturing video footage in real time. The video resolution is 1080×1080 , and the frame rate is 25 fps. The original video is about 3 min long and covers static obstacles (such as chairs, tables, etc.).

Then, Python 3.13.1 was used to extract 1000 frames evenly from the specified video and save them as high-quality JPG images. The specific extraction process is as follows: The video file is opened through OpenCV to obtain basic information such as the total number of frames, frame rate (FPS), and video duration. Then, the uniformly sampled interval frame number is calculated based on the target frame number and the total number of video frames. After that, the video is traversed frame by frame, and the qualified frames are saved as sequentially numbered JPG images according to the calculated interval. When the target frame number is reached or the video processing is completed, all images are saved in a specified folder in the same directory as the video file with a quality setting of 90%. Some dataset images are shown in Figure 4.

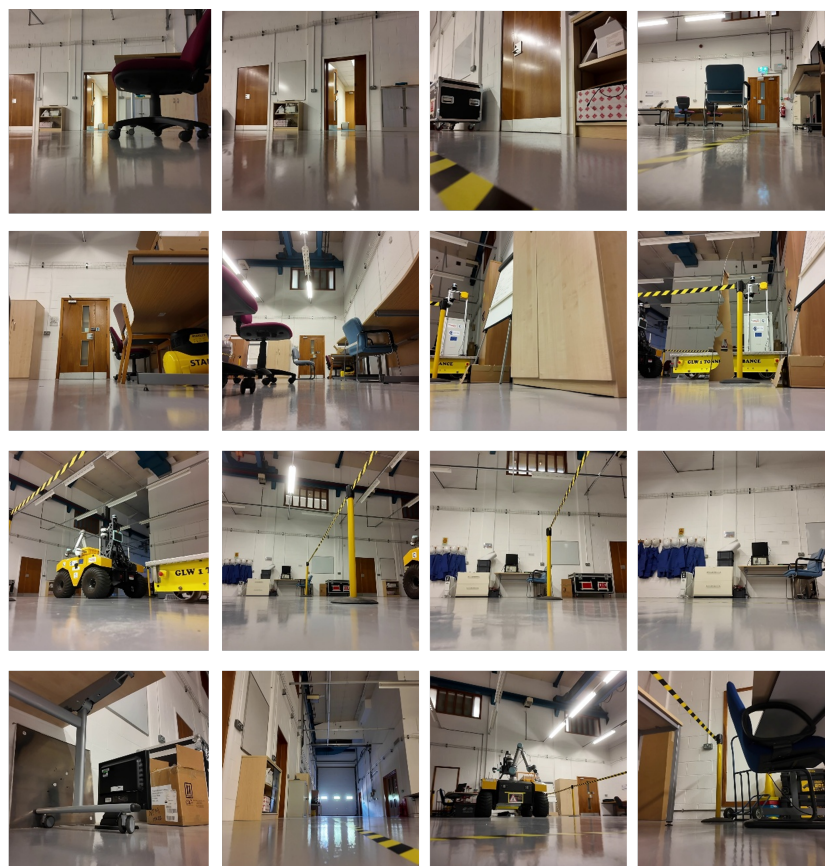


Figure 4. Some dataset images.

To improve clarity, additional details of the dataset construction and annotation process have been added. The dataset was collected under indoor laboratory conditions using the onboard camera mounted on the gecko robot. Video sequences were recorded under different lighting conditions and viewing angles to increase scene diversity.

Frame extraction was performed at uniform intervals to avoid temporal redundancy. All images were manually annotated using LabelImg in YOLO format, where each obstacle instance was labeled with a bounding box and a predefined category label. To ensure annotation consistency, all labeled samples were cross-checked, and ambiguous samples were corrected or removed.

3.2.2. Dataset Annotation

The images were manually annotated using LabelImg, which is an open-source image annotation tool that is mainly used to create labeled datasets for computer vision tasks such as object detection. Obstacles were classified into eight categories. The entire dataset has 2231 labels, and the number of labels for each category is shown in Table 2. Some of the labeled data images are shown in Figure 5. And after obtaining the labeled data, the dataset is divided into 80% for training, 10% for the validation set, and 10% for the test set, respectively.

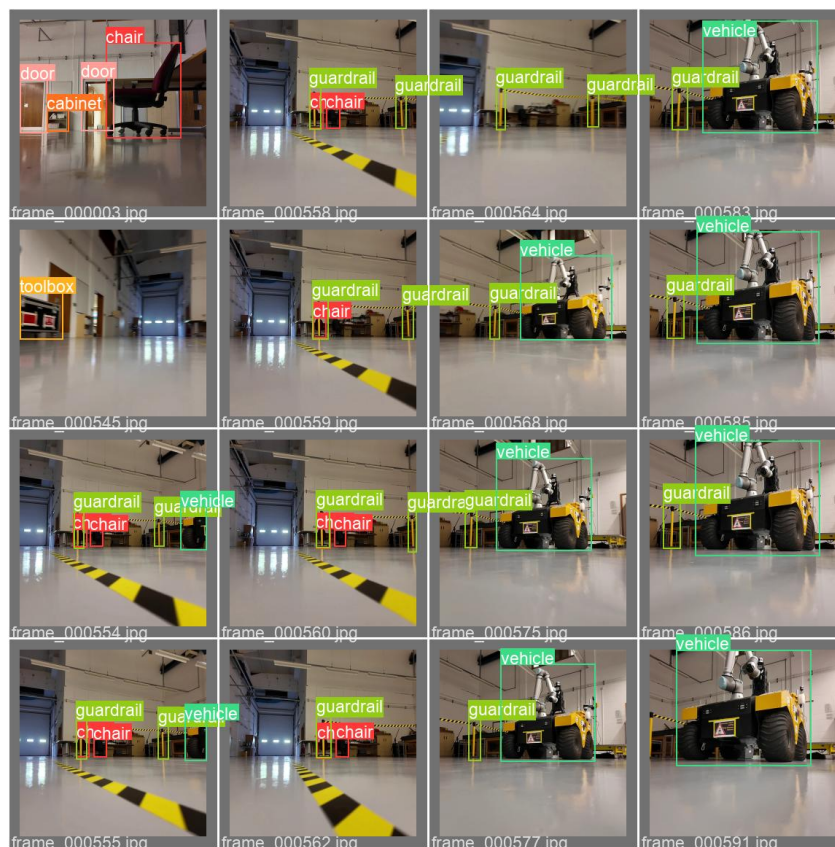


Figure 5. Some labeled images.

Table 2. Label Distribution Statistics.

Label	Chair	Door	Cabinet	Toolbox	Desk	Screen	Guardrail	Vehicle
Count	579	512	292	115	283	68	262	120
Percentage	25.9%	22.9%	13.1%	5.2%	12.7%	3.0%	11.7%	5.4%

3.2.3. Model Training and Validation

This section describes the implementation process of the YOLOv5 model for obstacle detection, including dataset preparation, model configuration, training, and deployment. The overall workflow is illustrated in Figure 6.

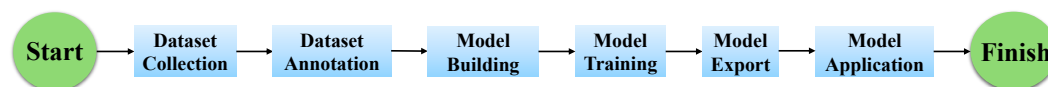


Figure 6. YOLOv5 model building process.

After dataset collection and annotation, the YOLOv5 model was configured according to the task requirements. Adjustments were made to the input image resolution and selected hyperparameters to better adapt to the characteristics of the robot-centric obstacle detection scenario. During training, appropriate loss functions, optimizers, and key hyperparameters (e.g., learning rate, batch size, and number of epochs) were selected and iteratively tuned to ensure stable convergence and effective feature learning.

After training, the model was exported to deployment-friendly formats such as ONNX and TorchScript to facilitate cross-platform inference. The trained model was then integrated into the gecko-inspired robotic system for real-time obstacle detection.

In this study, model suitability is evaluated from a system-level perspective rather than pure benchmark accuracy. The perception module must satisfy several practical requirements imposed by the gecko-inspired robotic platform, including real-time execution, limited onboard computational resources, stable deployment, and compatibility with future perception–motion coupling.

Although more recent state-of-the-art detection models may achieve higher benchmark accuracy, their increased architectural complexity often results in higher computational cost and inference latency. Under the hardware and integration constraints of the present robotic platform, YOLOv5 provides a balanced trade-off between detection reliability, computational efficiency, and deployment maturity. Therefore, it is selected as the perception backbone in this work. Table 3 summarizes the relationship between the system requirements and the capabilities of YOLOv5 under the current application scenario.

Table 3. System requirements and model selection rationale.

System Requirement	Relevance to Gecko Robot	YOLOv5 Capability
Real-time inference	Required for timely perception and future perception–motion coupling	Lightweight variants enable low-latency inference suitable for real-time operation
Limited onboard computation	Onboard processors impose strict constraints on model size and complexity	Moderate parameter count and memory footprint facilitate embedded deployment
Deployment stability	Reliable long-term operation is required for robotic experiments	Mature codebase and widely validated engineering ecosystem
Interface compatibility	Perception outputs must be easily integrated with locomotion control modules	Simple and interpretable output structure (bounding boxes, confidence scores)
Task-specific accuracy	Accurate obstacle detection under robot-centric viewpoints	High precision and recall achieved on a custom dataset

3.3. CPG Control Implementation

The Central Pattern Generator (CPG) is an efficient bionic neural mechanism control algorithm that controls the rhythmic movements of organisms such as walking, running, and flying by simulating the oscillation signals generated by the central nervous system of vertebrates [3]. The CPG control model is usually based on the following neural oscillators,

such as the Matsuoka neural oscillator [42], the VanderPol oscillator [43], and the Hopf oscillator [44], to build a network composed of multiple neural oscillatory units to achieve coordination and control of the robot's movement rhythm.

The Hopf oscillator was chosen as the central pattern generator (CPG) model because it has smooth limit cycle behavior, tunable frequency/amplitude, and robustness to perturbations, making it well suited for bio-inspired motion control. Its mathematical model is as follows:

$$\begin{aligned}\dot{x} &= \alpha(\mu - r^2) - \omega y \\ \dot{y} &= \alpha(\mu - r^2) + \omega x\end{aligned}\quad (5)$$

where x, y are the state variables of the oscillator, $r^2 = x^2 + y^2$, α, μ, ω are model parameters, which are related to the dynamic characteristics of the oscillator. Each parameter affects the dynamic performance separately.

In order to form an effective quadruped gait in a robot, multiple Hopf oscillators need to be connected to form a network through phase coupling terms, shown as Equation (6):

$$\dot{\phi}_i = \omega_i + \sum_j K_{ij} \sin(\phi_j - \phi_i - \phi_{ij}^{\text{offset}}), \quad (6)$$

where ϕ_i represents the current phase of the i th oscillator, K_{ij} represents the coupling strength with the adjacent oscillator j , and $\phi_{ij}^{\text{offset}}$ represents desired relative phase difference. By controlling $\phi_{ij}^{\text{offset}}$, different gait patterns can be achieved (such as diagonal gait, turning on the spot, etc.)

Then the oscillator state variables are mapped to the target angles of each joint of the robot through a linear transformation, shown in Equation (7):

$$\theta_i(t) = A_i \cdot x_i(t) + \theta_{\text{bias}}, \quad (7)$$

where $\theta_i(t)$ represents the target angle of the i th joint at time t , A_i represents the Oscillation amplitude coefficient (amplification factor) of the i th joint, $x_i(t)$ represents the state variable output of the i th Hopf oscillator at time t , θ_{bias} represents static offset angle, which is the neutral position of the joint.

The CPG neural network built based on the Hopf oscillator supports continuous and smooth periodic motion, and can easily control frequency, phase and amplitude to achieve dynamic gait switching. It also has strong system stability and is suitable for tasks such as bionic crawling, climbing, and turning.

3.4. System Architecture and Integration Framework

In the present study, the vision-based perception module and the CPG-based locomotion controller are experimentally validated as independent subsystems. A fully implemented real-time software interface enabling perception–motion closed-loop control has not yet been realized. Therefore, this section focuses on the formulation of a unified system architecture and an analysis of integration feasibility rather than reporting completed closed-loop experiments. The proposed framework establishes a structured data flow from perception to locomotion. Instead of directly transmitting high-dimensional visual features, the perception module abstracts obstacle information into a set of low-dimensional, robot-centric descriptors, such as obstacle direction and relative distance. This abstraction strategy reduces interface complexity and enhances interpretability at the control level.

These compact descriptors are designed to modulate a limited number of CPG parameters, including oscillation amplitude, frequency, and phase offsets. By adopting parameter-level coupling, the framework enables a biologically inspired and computationally efficient pathway for perception–locomotion interaction, while avoiding tight coupling between heterogeneous data representations.

From a dynamical systems perspective, the CPG controller operates as a continuous oscillator network capable of maintaining stable rhythmic motion in the absence of external updates. This inherent stability allows the perception module to function asynchronously at a lower update frequency without introducing motion discontinuities or timing conflicts. Such a modular and loosely coupled design enhances extensibility and reduces potential synchronization risks in future closed-loop implementations.

Although real-time closed-loop validation and system-level debugging are beyond the scope of this paper, the proposed architecture provides both a theoretical and practical foundation for future perception–decision–motion integration in bio-inspired gecko robots. The present contribution lies in defining the interface principles, data abstraction strategy, and parameter-level coupling mechanism necessary for scalable obstacle avoidance systems.

3.5. Obstacle-Aware CPG Parameter Modulation Strategy

To establish a structured connection between visual perception and rhythmic locomotion, this study formulates an obstacle-aware parameter modulation strategy within the Hopf-based CPG framework. The objective is to construct a transparent and interpretable coupling mechanism that links obstacle perception outputs to locomotion parameters, thereby validating the feasibility of perception–motion interaction at the parameter level.

In the proposed formulation, visual detection results are abstracted into two compact, robot-centric descriptors: obstacle direction $d \in \{left, front, right\}$ and relative proximity ρ . This abstraction reduces high-dimensional perception data into low-dimensional control-relevant variables, enabling simplified and stable interface design.

These descriptors modulate a limited set of CPG parameters, including oscillation amplitude, frequency, and phase offset. The modulation rules are defined as follows:

$$A_i = A_0 + k_A \cdot f(\rho), \quad (8)$$

$$\omega = \omega_0 \cdot (1 - k_\omega \cdot f(\rho)), \quad (9)$$

$$\phi_{ij}^{offset} = \begin{cases} \phi_0 + \Delta\phi, & d = left, \\ \phi_0 - \Delta\phi, & d = right, \\ \phi_0, & d = front, \end{cases} \quad (10)$$

where A_i denotes the oscillation amplitude of the i -th joint, ω represents the oscillation frequency, and ϕ_{ij}^{offset} is the inter-oscillator phase offset. The normalized proximity function is defined as:

$$f(\rho) = \min\left(\frac{\rho_{max} - \rho}{\rho_{max}}, 1\right). \quad (11)$$

Under this formulation, decreasing obstacle distance increases turning amplitude while reducing oscillation frequency, effectively lowering locomotion speed near obstacles. The direction variable determines the sign of phase offset adjustment, thereby inducing left or right turning behavior within the rhythmic gait pattern.

The proposed strategy adopts a deterministic rule-based mapping between perception descriptors and CPG parameters. This design prioritizes interpretability, parameter transparency, and compatibility with the Hopf oscillator dynamics. It provides a structurally consistent baseline for perception–motion coupling without introducing additional optimization layers or high-dimensional policy representations.

It should be noted that the present formulation does not incorporate adaptive learning, disturbance rejection control, or environment-dependent policy optimization. Real-time closed-loop adjustment and robustness enhancement under dynamic uncertainties remain important directions for future research. The current contribution lies in establishing a clear and implementable parameter-level coupling mechanism that bridges perception outputs with rhythmic locomotion control.

4. Experiments and Results

This section presents the experimental validation and performance evaluation of the bio-inspired gecko robot system developed in this study.

4.1. Prototype Assembly

The connecting parts of the robot spine and limbs are made of resin material through 3D printing. The flexible spine and rigid modules of the robot limbs are fabricated using resin-based 3D printing. The head and tail are made of artificial chemical fibers and fixed to the shoulder strap and pelvic belt by adhesive, respectively. KSTD5589MG servomotors (manufactured by KST Digital Technology Limited, Shenzhen, China) were used to drive the limb joints. The hardware circuit board contained in the robot control system is arranged on the flexible spine and fixed by studs. The entire experimental prototype of the gecko robot weighs 620 g, and is 65 cm long and 32 cm wide. The robot prototype is shown in Figure 7.

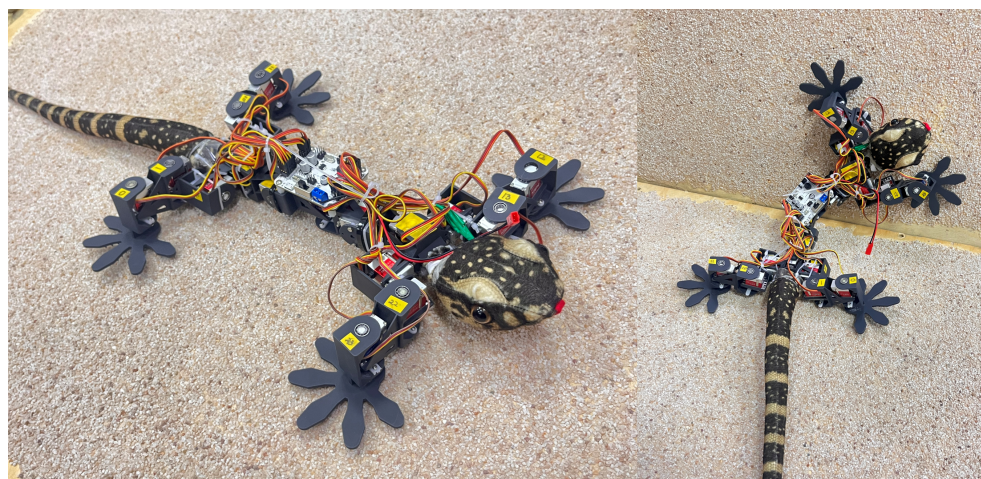


Figure 7. Assembled prototype of the bio-inspired gecko robot.

4.2. YOLOv5 Detection Performance

The model is trained using the training set. The model training uses the pre-trained model as the basis, and the network is fine-tuned through transfer learning. The hyperparameters during the experiment are shown in Table 4.

To further analyze the robustness of the selected hyperparameters, we examined the influence of key parameters on training stability and detection performance.

Learning rate. The initial learning rate was set to 0.01. Increasing the learning rate beyond this value resulted in noticeable oscillations in the training loss and reduced convergence stability. Conversely, reducing the learning rate below 0.005 slowed convergence and increased training time without yielding significant improvement in mAP. Therefore, 0.01 provides a balanced trade-off between convergence speed and stability.

Table 4. Hyperparameters of the network model.

Hyperparameter	Description	Experimental Value
Learning Rate	Controls the step size of model parameter updates during optimization	0.01
Batch Size	The number of samples processed in one forward/backward pass during training	16
Optimizer	Algorithm used to update model weights (Adam combines Momentum and RMSprop)	Adam
Momentum	Parameter that accelerates gradient descent in the relevant direction and dampens oscillations	0.937
Epochs	Complete passes through the entire training dataset during model training	100
Input Image Size	Spatial dimensions (width $\times$ height) of input images after resizing	608 $\times$ 608
Pretrained Weights	Initial model parameters trained on large datasets (COCO) for transfer learning	yolov5l.pt

Batch size. A batch size of 16 was selected considering GPU memory constraints and gradient stability. Increasing the batch size slightly improved gradient smoothness but led to higher memory consumption and longer per-epoch training time. Reducing the batch size below 8 introduced larger fluctuations in loss curves, which negatively affected convergence stability.

Number of epochs. Training for fewer than 80 epochs resulted in incomplete convergence and slightly lower mAP values. Increasing the number of epochs beyond 150 did not produce significant accuracy gains but increased training time, indicating diminishing returns.

Input image size. The input resolution of 608 $\times$ 608 was chosen as a compromise between detection accuracy and computational cost. Increasing resolution improved detection of small objects marginally but increased inference latency. Decreasing resolution reduced computational load but slightly degraded detection precision.

Overall, the selected hyperparameter configuration ensures stable convergence and balanced performance under the computational constraints of the gecko-inspired robotic platform. To provide a clearer overview of the parameter sensitivity analysis, the effects of varying key hyperparameters are summarized in Table 5. This summary illustrates the rationale for the selected configuration and its balance between convergence stability, detection accuracy, and computational efficiency.

Table 5. Sensitivity analysis and robustness assessment of key training hyperparameters.

Parameter	Current Value	Current Assessment	If Increased	If Decreased
Learning Rate	0.01	Balanced	Loss oscillation/instability	Slower convergence
Batch Size	16	Stability–memory trade-off	Higher memory usage	Gradient fluctuations
Epochs	100	Near convergence	Risk of overfitting	Incomplete convergence
Input Resolution	608 $\times$ 608	Accuracy–speed balance	Increased latency	Reduced precision
Momentum	0.937	Stable default	Possible oscillation	Slower convergence

The dataset was trained using the hyperparameters of the model mentioned above, and the model underwent a total of 100 training rounds. During the model training process, various generated images helped us intuitively understand the model's progress and performance at different stages.

Figure 8 shows the prediction results for the model validation set, where each box represents the object category identified by the model and its corresponding detection confidence score. From the figure, it can be seen that the confidence scores for most objects are at a high level. Each sub-figure represents a different image frame, showcasing object detection results in various scenes and demonstrating the model's performance under diverse conditions. This reflects the model's accuracy and reliability.



Figure 8. The prediction results of the model on the validation set.

As shown in Figure 9, the YOLOv5 training results in the dataset. The result figure displays the model loss curves in both the training set and the test set, as well as the changes in evaluation metrics such as accuracy, recall rate, average precision, and mean average precision. Box_loss represents the difference between the predicted box and the actual box. As the training progresses, this loss gradually decreases, indicating that the model's performance in spatial localization is getting better and better. Obj_loss represents the loss of target confidence. This value also gradually decreases, indicating that the model's confidence in target detection is increasing. The cls_loss represents the classification loss, which decreases as training progresses, indicating an improvement in the model's performance in classification. Val/box_loss, val/obj_loss, and val/cs_loss all show

a decreasing trend similar to the training loss, indicating that the model also performs well on the validation dataset and there is no obvious overfitting phenomenon. Recall represents the proportion of successfully detected samples among all true positive ones. This value is close to 1, indicating that the model performs very well in detecting real targets. MAP_0.5 and mAP_0.5:0.95, these two indicators reflect the comprehensive detection ability of the model under different IoU thresholds. As the training progresses, these two mAP values gradually increase, indicating a effective in the detection performance of the model under various conditions, especially when the mAP value approaches 1, indicating excellent overall performance of the model. The training curves indicate that the model exhibits stable learning behavior during both training and validation processes, with a continuous decrease in the loss curve and increasing accuracy and recall. Overall, the training results indicate stable convergence and suitability for practical object detection tasks.

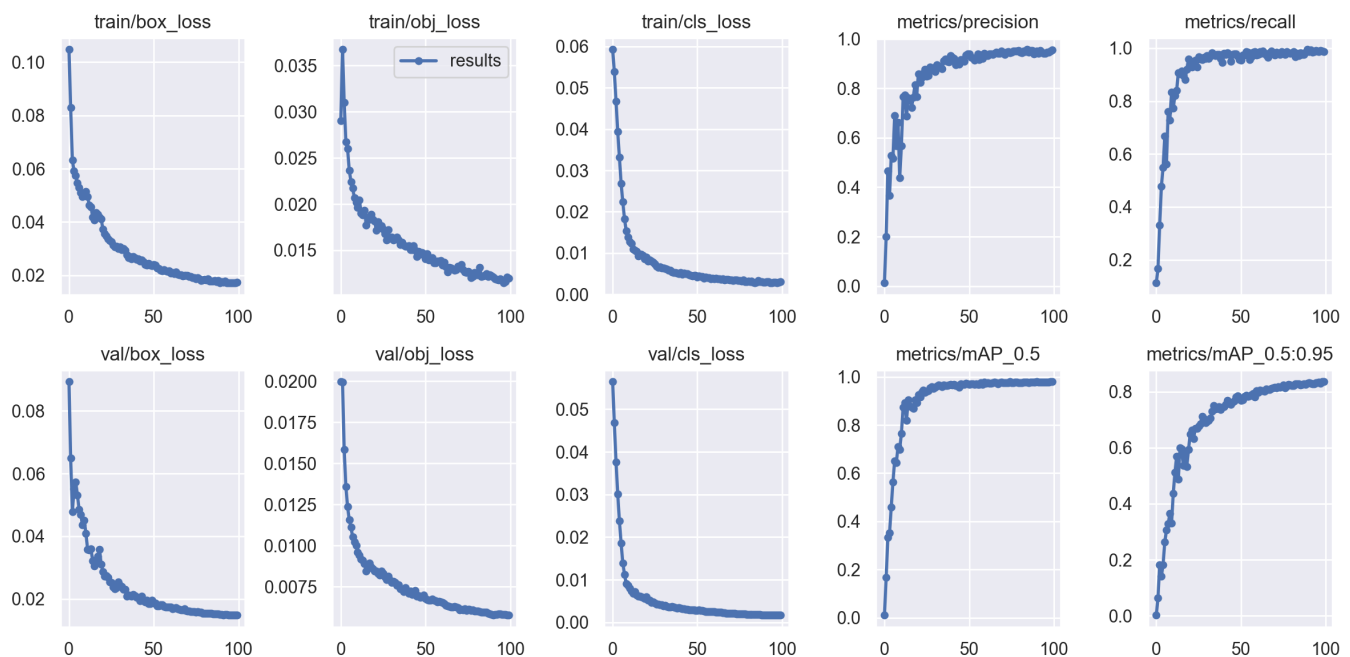


Figure 9. Training results of YOLOv5 on the obstacle detection dataset.

The two graphs in Figure 10 show the changes in precision and recall of the model relative to confidence. As the confidence level gradually increases from 0, the overall precision shows an upward trend, indicating that higher confidence thresholds lead to increased detection precision. The precision of all categories reached 1.00 at a confidence level of 0.947, indicating that all detected samples were correctly predicted at this threshold. Individual categories such as “chair” and “cabinet” perform relatively well in precision at lower confidence levels, but as they approach the high confidence threshold, precision may tend to saturate. The recall rate maintains a high level of confidence from 0 to 1, especially when it is relatively stable at low confidence, indicating that the model can detect most of the targets. The precision and recall of the model perform well in different classifications, which can strike a balance between precision and recall to meet the needs of different practical applications.

The two graphs in Figure 11 illustrate the variation of the F1-score and the precision–recall (PR) curves of the proposed YOLOv5 model under different confidence thresholds. These metrics are commonly used to evaluate detection performance and to determine an appropriate operating point for practical deployment.

As shown in Figure 11a, the F1-score of all categories increases with the confidence threshold and reaches its maximum value of 0.97 at a threshold of 0.573. Since the F1-score represents the harmonic mean of precision and recall, this operating point provides the best overall balance between false positives and missed detections. Therefore, the threshold corresponding to the maximum F1-score is considered the most suitable choice for practical obstacle detection.

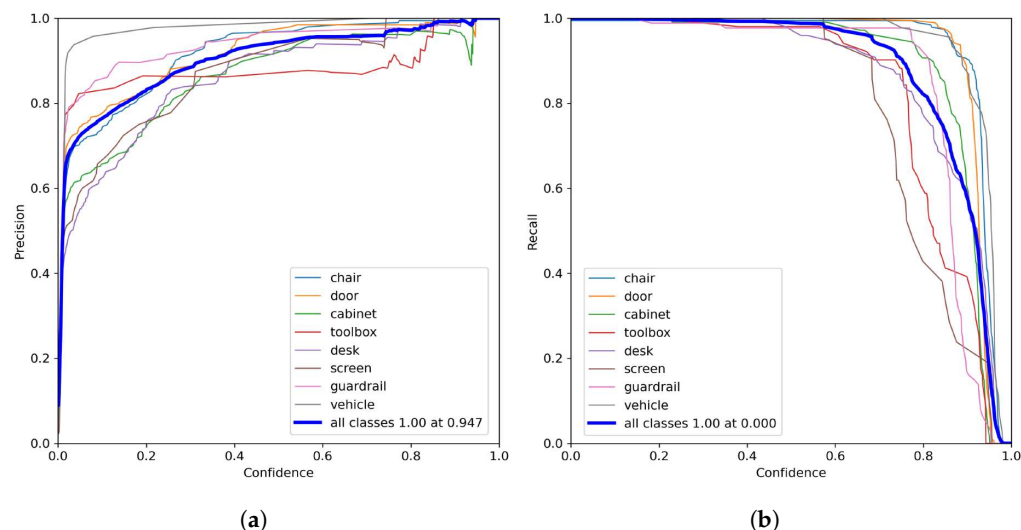


Figure 10. (a) Precision curve and (b) Recall curve of the YOLOv5 model.

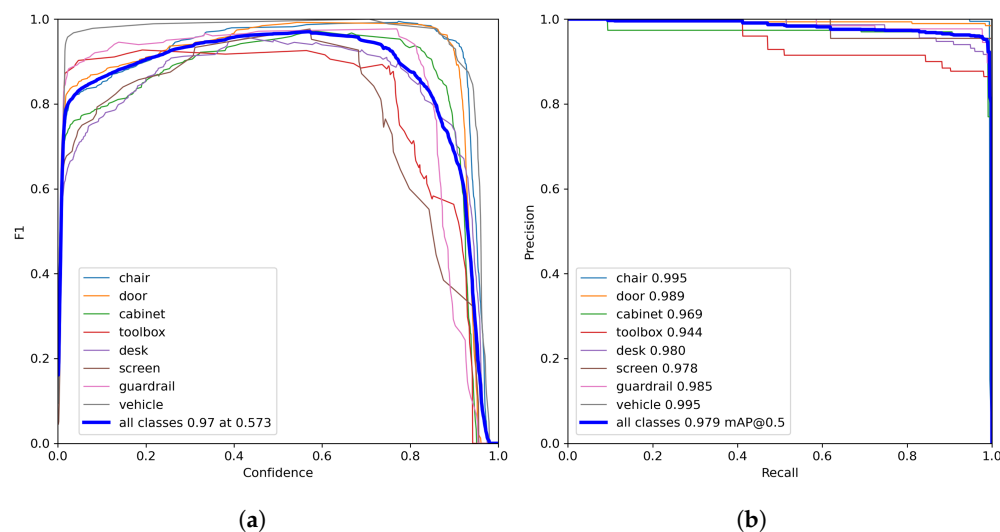


Figure 11. (a) F1 curve and (b) PR curve of the YOLOv5 model.

The PR curves in Figure 11b further demonstrate the detection capability of the model across different recall levels. Most object categories exhibit high precision even at large recall values, particularly for classes such as *chair*, *guardrail*, and *vehicle*, indicating strong robustness and consistent detection performance. The overall mAP reaches 0.979, confirming the effectiveness of the trained model.

It is also observed that when the confidence threshold is increased to 0.947, the precision of all categories reaches 1.00, meaning that all remaining predicted bounding boxes are correct at this operating point. However, this perfect precision is obtained at the expense of recall, as a large number of low-confidence detections are suppressed, resulting in missed obstacle detections.

In object detection systems, the confidence threshold directly controls the trade-off between precision and recall. A higher threshold reduces false positives but increases the risk of missed detections, whereas a lower threshold improves recall while introducing additional false alarms. The confidence threshold mainly affects detection quality during post-processing and has negligible influence on inference speed.

For robotic obstacle avoidance tasks, missed detections pose a greater safety risk than occasional false positives. Consequently, although a threshold of 0.947 achieves perfect precision, it is not selected as the operating point in practice. Instead, a moderate confidence threshold corresponding to the maximum F1-score is adopted, as it provides a more reliable balance between detection accuracy and robustness for real-time navigation. The confidence threshold does not significantly affect inference speed, as it is applied during post-processing. Its primary impact lies in detection quality rather than computational efficiency.

4.3. Comparative Experiment of Models

To further evaluate model suitability under the system constraints of the proposed robotic platform, comparative experiments were conducted at two levels: (1) comparison between different detection architectures (YOLOv5 vs. YOLOv11), and (2) ablation-style analysis among different YOLOv5 variants.

4.3.1. Comparison Between YOLOv5 and YOLOv11

We conducted training experiments using YOLOv11 under identical dataset and training settings. The objective of this comparison is not to establish a benchmark ranking, but to analyze whether more recent model architectures provide practical advantages under the system constraints of the proposed robotic platform.

Table 6 summarizes the comparative training results between YOLOv5 and YOLOv11 under identical dataset and training settings. Both models achieve comparable precision and recall values, and no significant improvement in mAP is observed with YOLOv11. However, the YOLOv11 training curves exhibit larger oscillations in the early epochs, indicating increased sensitivity to training dynamics.

Table 6. Comparison between YOLOv5 and YOLOv11 under identical training conditions.

Metric	YOLOv5	YOLOv11
Precision	≈0.98–0.99	≈0.97–0.99
Recall	≈0.97–0.99	≈0.97–0.99
mAP@0.5	≈0.99	≈0.98
mAP@0.5:0.95	≈0.82–0.85	≈0.75–0.78
Training stability	Smooth convergence	Noticeable oscillation in early epochs
Model complexity	Moderate	Higher
Deployment maturity	Widely validated	Relatively recent

4.3.2. Ablation Analysis of YOLOv5 Variants

Given that the objective of this work is to ensure stable real-time perception under the hardware and system constraints of a gecko-inspired robotic platform, model suitability is evaluated based on detection reliability, training stability, computational efficiency, and deployment maturity. Under these criteria, YOLOv5 provides a more balanced and robust solution for the current task.

To further analyze model suitability under the system constraints of the proposed gecko-inspired climbing robot, we conducted an ablation-style comparison among different YOLOv5 variants, including YOLOv5n, YOLOv5s, YOLOv5m, and YOLOv5l. The purpose of this comparison is not to perform state-of-the-art benchmarking, but to investigate

how model scale and network configuration affect detection performance and real-time feasibility within a unified system framework.

For fairness, all YOLOv5 variants were trained under identical experimental conditions, including the same input image resolution, data augmentation strategies, number of training epochs, and hyperparameter settings. Only the model scale and internal network configuration were varied. The evaluation metrics include precision, recall, mAP@0.5, and mAP@0.5:0.95, as summarized in Table 7. To provide a clearer visual comparison of the performance differences among the models, a corresponding bar chart is presented in Figure 12.

Table 7. Comparative Experimental Model Evaluation Index Table.

Model	P/%	R/%	mAP@0.5/%	mAP@0.5:0.95
Yolov5s	0.915	0.96	0.97	0.81
Yolov5n	0.926	0.95	0.96	0.78
Yolov5l	0.947	0.98	0.97	0.84
Yolov5m	0.942	0.97	0.97	0.82

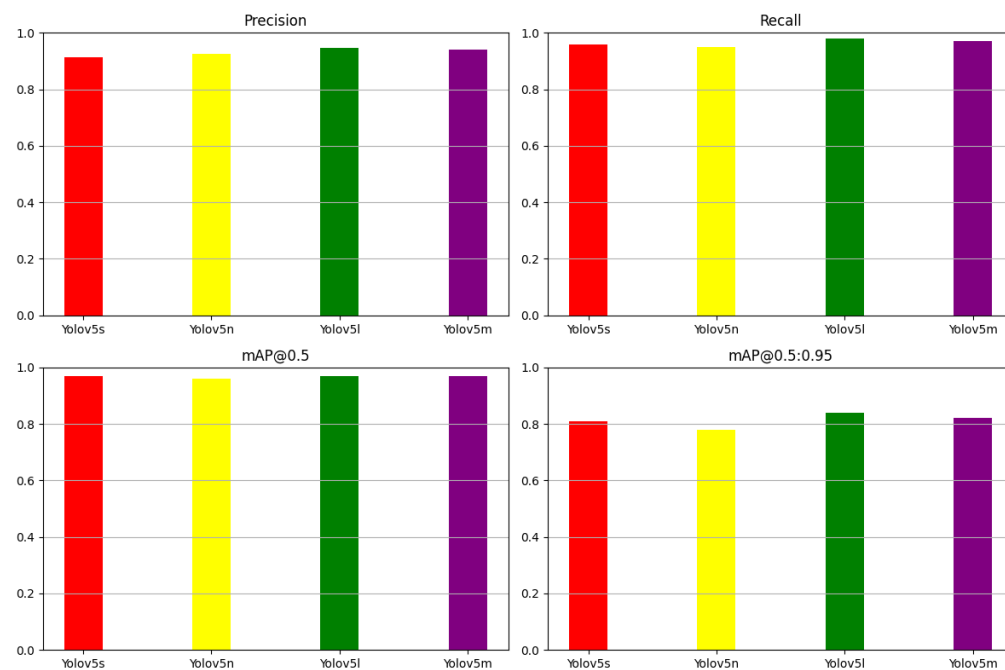


Figure 12. Bar chart of evaluation indicators for comparative experimental models.

Among them, the precision of Yolov5s is 0.915, which is the lowest among the four models. The precision of Yolov5n and Yolov5m has slightly improved, reaching 0.926 and 0.942, respectively. Yolov5l has the highest precision, 0.947, demonstrating the best classification precision. The recall rates of all models are close to 0.97 or higher, indicating that these models perform well in detecting true positive cases, especially Yolov5l has the highest recall rate, indicating its excellent ability to collect all positive cases. All models are available at mAP@0.5. The performance of the indicators is relatively consistent, ranging from 0.96 to 0.97, indicating that all models can effectively detect targets under a relatively relaxed IOU threshold. However, under stricter criteria of mAP@0.5:0.95, Yolov5l also leads with a score of 0.84, demonstrating its higher robustness. In general, Yolov5l performs excellently in all indicators.

4.4. CPG Motion Performance

The first step in the CPG control algorithm is to build the oscillation center of the imitation gecko model, as shown in Figure 13. Based on the kinematic model design, a total of 11 oscillation centers were set up, of which 8 were located in the shoulder and knee joints of the limbs. The other three are located on the flexible spine. Through the above design, the oscillation center can spontaneously and continuously generate joint control signals through learning.

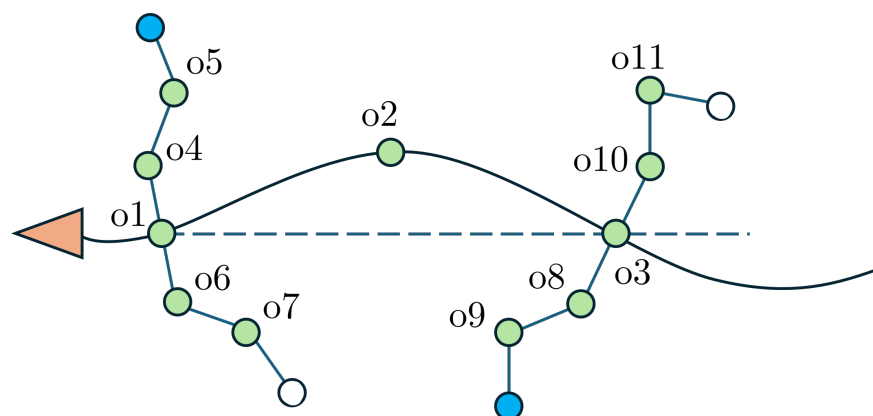


Figure 13. Construction of the gecko robot oscillation center.

The CPG signal generated through the Hopf oscillator network successfully performs diagonal gait control. The angular timing curves of the knee and shoulder joints are shown in Figures 14 and 15. Among them, the left front leg (LF) and right rear leg (RH), the right front leg (RF), and left rear leg (LH) are respectively in synchronization, with a phase difference of 0, and a phase difference of π is present between the two groups of diagonal legs, which meets the diagonal gait requirements. The knee joint controls the forward motion of the robot, with the swing phase at 0 to 30 degrees and the support phase at 40 to 60 degrees. The phase state distinction is achieved by switching the symbol of the oscillator state variable y in Equation (5). The shoulder joint controls the leg lifting action and sets the joint angle change range from 0 to 60 degrees, corresponding to the linear mapping of A_i and θ_{bias} in Equation (7). All joints oscillate at a frequency of 1 Hz ($\omega = 2\pi$ rad/s).

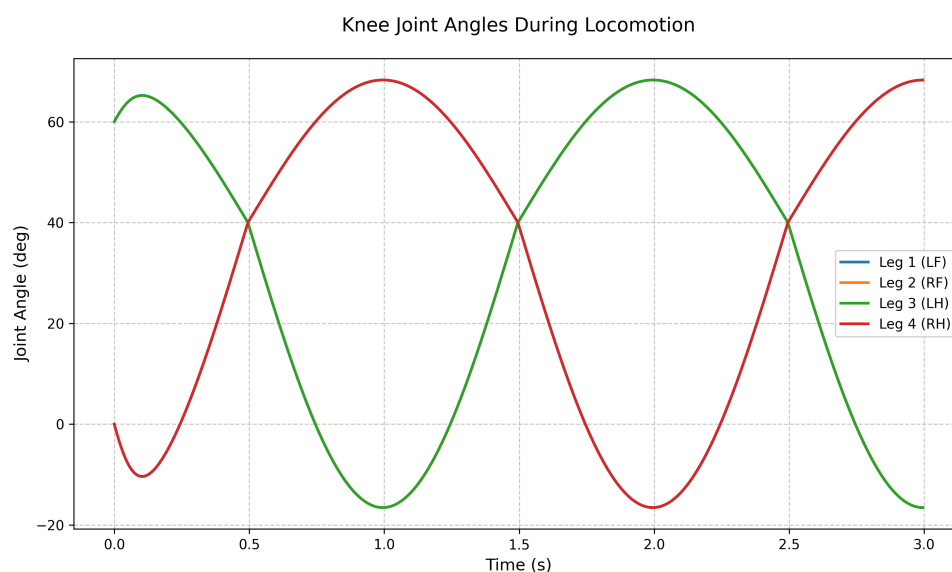


Figure 14. Angle timing curves of knee joints.

The three lateral swing joints of the spine show wave motions with a phase difference of $\pi/2$. As shown in Figure 16, the maximum deflection angles are $\pm 15^\circ$, $\pm 10^\circ$, and $\pm 8^\circ$, respectively, which can effectively assist steering.

As shown in Figure 17, the proportion of the support phase is about 60%, indicating the spatio-temporal coordination of diagonal gait. When one leg enters the swing phase, its opposite leg always remains in a supporting state, which avoids violent fluctuations in the center of gravity and increases the speed and stability of movement.

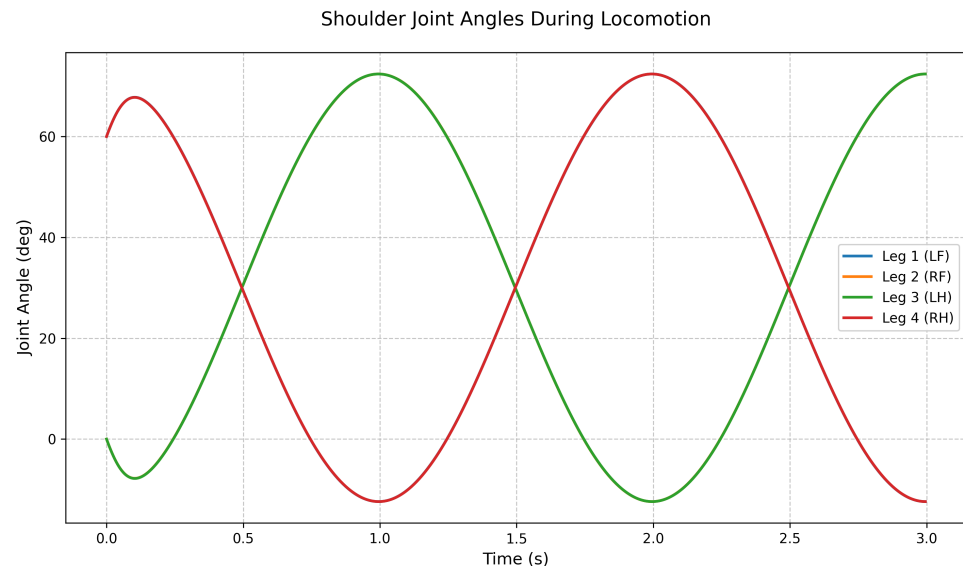


Figure 15. Angle timing curves of shoulder joints.

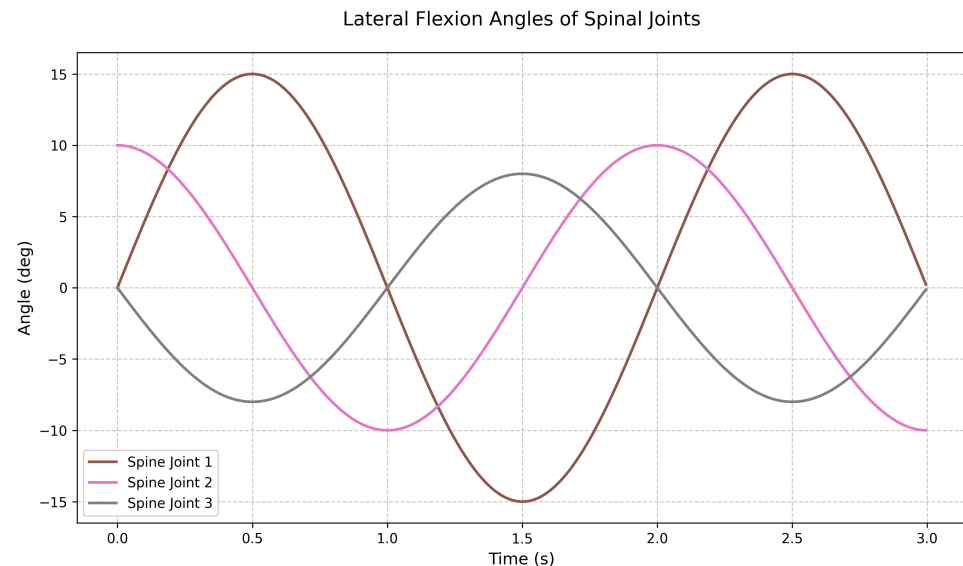


Figure 16. Angle timing curves of spine joints.

The joint angles output by the above CPG were mapped to each servo motor through PWM signals, and a crawling experiment was conducted on a horizontal plane. In the robot's straight-line motion experiment, the following Figure 18 shows the straight-line motion of the gecko in the diagonal gait. From the motion in Figure 18, it can be found that the flexible spine of the gecko-like robot can produce periodic lateral deflection, accompanied by alternating swings of the limbs back and forth. The robot can move smoothly in a straight line on the experimental platform. The experimental results also demonstrated the feasibility of the CPG control algorithm.

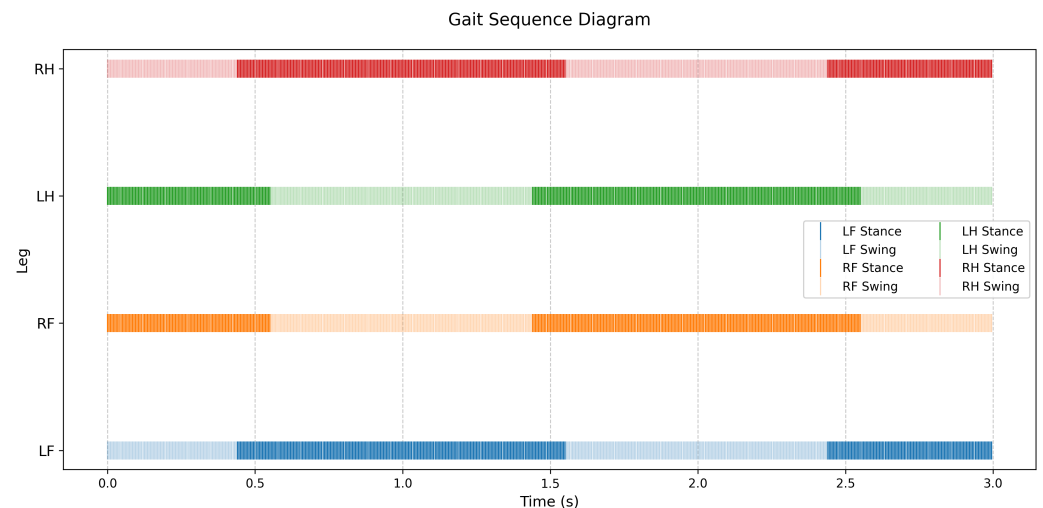


Figure 17. Gait sequence diagram.

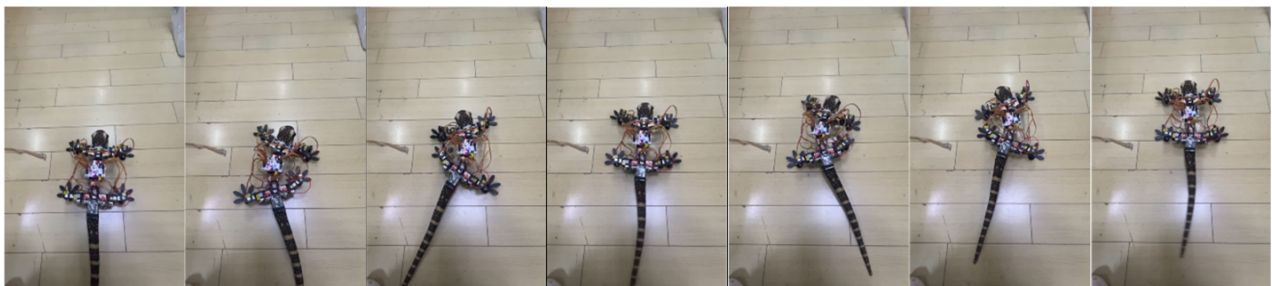


Figure 18. Straight-line crawling motion of the bio-inspired gecko robot prototype under CPG control.

4.5. Discussion on Future Integration

Based on the system architecture described in the previous section, this part discusses potential extensions toward a fully integrated perception–decision–motion closed-loop obstacle avoidance system.

The core integration concept of this study is to transform visual perception outputs into modulation of locomotion parameters within the CPG controller. Specifically, obstacle information obtained from YOLO-based detection is abstracted into low-dimensional descriptors, such as obstacle direction and relative proximity, which are mapped to oscillation amplitude, frequency, and phase offsets of the Hopf oscillator network. This parameter-level coupling provides a structured interface between perception and locomotion, enabling biologically inspired sensorimotor coordination at the framework level.

In a potential closed-loop implementation, the robot would acquire forward-view images through an onboard camera, detect obstacles in real time, and estimate their relative positions using depth information or geometric models. The system would then determine whether obstacles fall within a predefined avoidance region and adjust the CPG parameters accordingly. For example, obstacles on the left could increase the oscillation amplitude and phase advance of the right limbs to induce a rightward turn, while frontal obstacles at close range could reduce oscillation frequency to enable in-place turning. When no obstacle is detected within the critical region, the CPG maintains nominal gait generation. The resulting oscillator states $x_i(t)$ are mapped to joint angle commands and transmitted to the servo drivers. The conceptual architecture is illustrated in Figure 19.

However, it is important to emphasize that the current experiments validate the perception and locomotion subsystems independently. Real-time perception–motion closed-loop control, including software integration, timing synchronization, latency analysis, and system-level debugging, has not yet been experimentally implemented. In particular,

the spinal motion demonstrated in this study corresponds to rhythmic oscillation under preset CPG parameters, and visually triggered spinal modulation remains a conceptual extension rather than an experimentally verified capability.

From a systems perspective, transitioning from subsystem validation to true autonomous obstacle avoidance involves several major challenges. Reliable real-time mapping between perception descriptors and locomotion parameters must be established to ensure responsive motion adaptation. In addition, synchronization between perception update rates and high-frequency motor control loops is required to prevent timing conflicts and instability. Robustness mechanisms must also be developed to handle perception uncertainty, environmental disturbances, and varying surface conditions encountered in climbing scenarios. Finally, long-duration integrated experiments are necessary to evaluate stability, adaptability, and performance under continuous interaction.

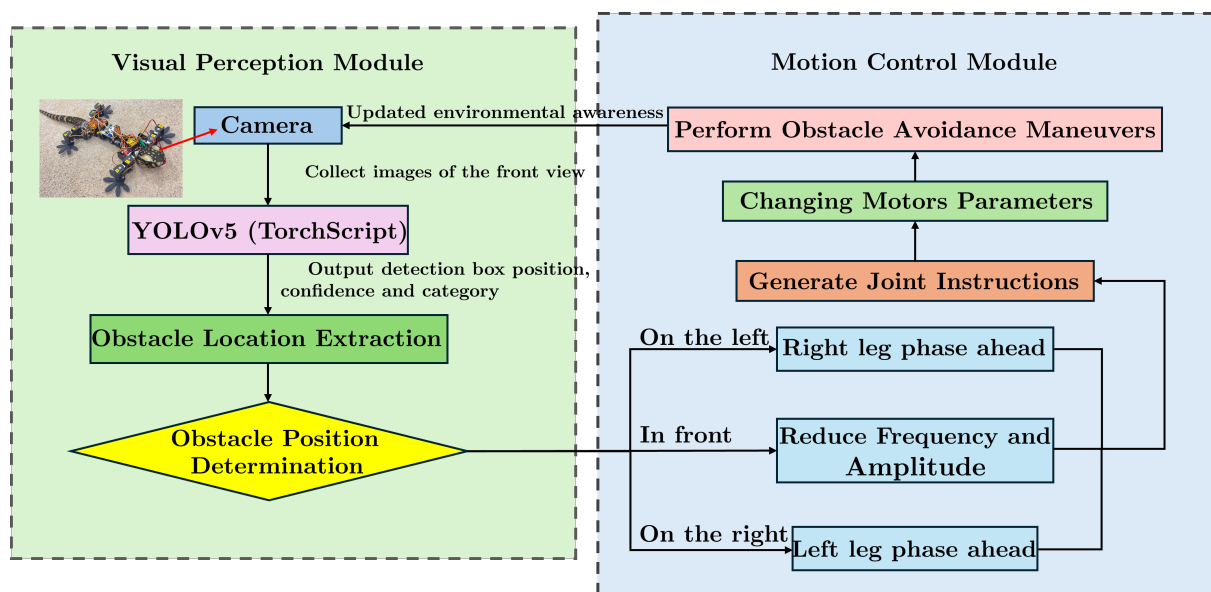


Figure 19. Vision-to-locomotion closed-loop control architecture.

These challenges define the gap between the current framework-level formulation and fully autonomous obstacle avoidance. Accordingly, the obstacle avoidance capability emphasized in this study should be interpreted as a research objective and system architecture rather than a completed autonomous function.

Future research will focus on implementing real-time perception–CPG coupling, including quantitative mapping strategies, synchronization analysis, and robustness enhancement. Adaptive or learning-based modulation mechanisms may be introduced to improve disturbance rejection and environment-dependent gait optimization. In addition, multi-modal sensing integration, such as tactile and proprioceptive feedback, will be investigated to enhance closed-loop stability in complex climbing environments.

Regarding reproducibility, the experiments are conducted using a self-collected dataset tailored to the robot-centric viewpoint of a gecko-inspired climbing robot. Public benchmark datasets are not directly applicable due to differences in scene geometry and obstacle characteristics. Detailed descriptions of data acquisition, annotation standards, training configurations, and hyperparameters are provided in the Section 3. Although the dataset is not publicly released at this stage, it can be made available upon reasonable academic request.

Overall, this work should be regarded as a foundational step toward integrated perception–motion systems for climbing robots, providing a structured framework and experimental baseline for future closed-loop development.

4.6. Limitations of the Proposed System

Despite the results obtained from subsystem validation, several limitations of the proposed framework should be acknowledged.

First, the perception and locomotion modules are evaluated independently, and real-time perception–motion closed-loop integration has not been experimentally implemented. Consequently, the obstacle avoidance capability remains at the framework level rather than a fully realized autonomous behavior.

Second, the perception–CPG coupling mechanism adopts a deterministic rule-based mapping, which does not incorporate adaptive learning, disturbance rejection, or environment-dependent optimization. This limits the system’s ability to cope with highly dynamic or uncertain environments.

Third, the experimental validation is conducted using a self-collected dataset under controlled conditions. Although it reflects the robot-centric viewpoint, further evaluation is required to assess generalization across diverse real-world environments.

Finally, long-duration integrated experiments and system-level robustness analysis under external disturbances have not yet been performed.

These limitations highlight the gap between the current subsystem-level verification and fully autonomous obstacle avoidance, and they define key directions for future research.

5. Conclusions

This paper investigates key enabling technologies for obstacle avoidance in bio-inspired gecko robots, focusing on mechanical design, vision-based perception, and CPG-based rhythmic locomotion control. Rather than claiming the realization of a fully autonomous closed-loop system, this study establishes and experimentally validates the core subsystems required for future perception–motion integration.

Experimental results demonstrate that the proposed gecko robot platform can achieve stable diagonal gait locomotion with coordinated limb–spine motion. Meanwhile, the vision-based perception module provides reliable real-time obstacle detection from the robot’s onboard perspective. These findings confirm the feasibility of combining bio-inspired mechanical structures, deep learning-based perception, and oscillator-driven gait generation within a unified robotic framework.

An important insight of this study is that obstacle avoidance in climbing robots is inherently a coupled system problem involving perception reliability, locomotion stability, and parameter interpretability. Successful integration requires careful consideration of computational constraints, interface design, and robustness under dynamic environmental conditions.

It should be emphasized that real-time perception–decision–motion closed-loop obstacle avoidance has not yet been experimentally implemented. The present contribution lies in subsystem validation and system framework formulation. Achieving true closed-loop autonomy requires addressing several challenges, including establishing a stable mapping between visual obstacle information and CPG parameters, ensuring real-time synchronization between perception and control modules, and maintaining closed-loop stability under environmental uncertainties.

Future work will focus on implementing real-time perception–CPG coupling to enable obstacle-dependent modulation of oscillation frequency, amplitude, and phase coordination. Improving environmental adaptability through enhanced visual robustness and multimodal sensing integration is also essential for practical deployment. In addition, systematic parameter sensitivity analysis and online adaptation mechanisms will be investigated to ensure reliable locomotion across diverse terrains and task conditions.

Overall, this study represents a foundational step toward robust and adaptive closed-loop obstacle avoidance in bio-inspired climbing robots, providing a structured framework upon which more advanced perception–motion integration and intelligent control strategies can be developed.

Author Contributions: Conceptualization, W.X., A.J. and B.H.S.A.; methodology, W.X., A.J. and B.H.S.A.; resources, B.H.S.A.; writing—original draft preparation, A.J., B.H.S.A. and W.X.; writing—review and editing, B.H.S.A., A.J. and W.X.; supervision, A.J. and B.H.S.A.; visualization, B.H.S.A. and A.J. All authors have read and agreed to the published version of the manuscript.

Funding: The National Natural Science Foundation of China (No. 52575339, No. 52405317) and the Natural Science Foundation of Jiangsu Province (BK20241407).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Qiu, J.; Ji, A.; Zhu, K.; Han, Q.; Wang, W.; Qi, Q.; Chen, G. A Gecko-inspired Robot with Flexible Spine Driven by Shape Memory Alloy Spring. *Soft Robot.* **2023**, *10*, 713–723. [[CrossRef](#)] [[PubMed](#)]
2. Nansai, S.; Mohan, R.E. A Survey of Wall Climbing Robots: Recent Advances and Challenges. *Robotics* **2016**, *5*, 14. [[CrossRef](#)]
3. Wang, W.; Ji, A.; Chen, G.; Ravi, S.; Shen, H.; Gorb, S.N.; Dai, Z. Kinematics of gecko climbing: The lateral undulation pattern. *Zoology* **2020**, *140*, 125768. [[CrossRef](#)] [[PubMed](#)]
4. Spenko, M.J.; Haynes, G.C.; Saunders, J.A.; Cutkosky, M.R.; Rizzi, A.A.; Full, R.J.; Koditschek, D.E. Biologically inspired climbing with a hexapedal robot. *J. Field Robot.* **2008**, *25*, 223–242. [[CrossRef](#)]
5. Autumn, K.; Buehler, M.; Cutkosky, M.; Fearing, R.; Full, R.J.; Goldman, D.; Groff, R.; Provancher, W.; Rizzi, A.A.; Saranli, U.; et al. Robotics in scansorial environments. In *Proceedings of the Unmanned Ground Vehicle Technology VII*; SPIE: Bellingham, WA, USA, 2005; Volume 5804, pp. 291–302.
6. Kim, S.; Spenko, M.; Trujillo, S.; Heyneman, B.; Mattoli, V.; Cutkosky, M.R. Whole body adhesion: Hierarchical, directional and distributed control of adhesive forces for a climbing robot. In *Proceedings of the Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy, 10–14 April 2007; pp. 1268–1273. [[CrossRef](#)]
7. Santos, D.; Heyneman, B.; Kim, S.; Esparza, N.; Cutkosky, M.R. Gecko-inspired climbing behaviors on vertical and overhanging surfaces. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 19–23 May 2008; pp. 1125–1131. [[CrossRef](#)]
8. Hawkes, E.W.; Eason, E.V.; Asbeck, A.T.; Cutkosky, M.R. The Gecko’s Toe: Scaling Directional Adhesives for Climbing Applications. *IEEE/ASME Trans. Mechatron.* **2013**, *18*, 518–526. [[CrossRef](#)]
9. Zhao, Z. Hook-Claw Type Quadruped Wall-Climbing Robot. Master’s Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2017.
10. Nan, J. Opposing-Grasp Hexapod Wall-Climbing Robot. Master’s Thesis, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2018.
11. Beck, H.K.; Schultz, J.T.; Clemente, C.J. A bio-inspired robotic climbing robot to understand kinematic and morphological determinants for an optimal climbing gait. *Bioinspir. Biomim.* **2022**, *17*, 016005. [[CrossRef](#)]
12. Zhao, Q.; Nakajima, K.; Sumioka, H.; Yu, X.; Pfeifer, R. Embodiment enables the spinal engine in quadruped robot locomotion. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 2449–2456. [[CrossRef](#)]
13. Seok, S.; Wang, A.; Chuah, M.Y.; Hyun, D.J.; Lee, J.; Otten, D.M.; Lang, J.H.; Kim, S. Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 1117–1129. [[CrossRef](#)]
14. Crespi, A.; Karakasiliotis, K.; Guignard, A.; Ijspeert, A.J. Salamandra Robotica II: An Amphibious Robot to Study Salamander-like Swimming and Walking Gaits. *IEEE Trans. Robot.* **2013**, *29*, 308–320. [[CrossRef](#)]
15. Karakasiliotis, K.; Thandiackal, R.; Melo, K.; Horvat, T.; Mahabadi, N.K.; Tsitkov, S.; Cabelguen, J.M.; Ijspeert, A.J. From cineradiography to biorobots: An approach for designing robots to emulate and study animal locomotion. *J. R. Soc. Interface* **2016**, *13*, 20151089. [[CrossRef](#)]
16. Haomachai, W.; Shao, D.; Wang, W.; Ji, A.; Dai, Z.; Manoonpong, P. Lateral Undulation of the Bendable Body of a Gecko-Inspired Robot for Energy-Efficient Inclined Surface Climbing. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7917–7924. [[CrossRef](#)]

17. Nadan, P.; Backus, S.; Johnson, A.M. LORIS: A Lightweight Free-Climbing Robot for Extreme Terrain Exploration. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 18480–18486. [\[CrossRef\]](#)
18. Baik, S.; Kim, D.W.; Park, Y.; Lee, T.J.; Ho Bhang, S.; Pang, C. A wet-tolerant adhesive patch inspired by protuberances in suction cups of octopi. *Nature* **2017**, *546*, 396–400. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Wang, L.; Zhou, Q.; Xu, S. Role of locust *Locusta migratoria manilensis* claws and pads in attaching to substrates. *Chin. Sci. Bull.* **2011**, *56*, 789–795. [\[CrossRef\]](#)
20. Kong, J.; Ji, A.; Han, Q.; Shen, H.; Liu, S.; Xiang, W.; Zhang, Q. Advances in Research of Wall-climbing Robots: From Biology to Bionics—A Review. *J. Bionic Eng.* **2025**, *22*, 945–981. [\[CrossRef\]](#)
21. Fang, Y.; Wang, S.; Bi, Q.; Cui, D.; Yan, C. Design and Technical Development of Wall-Climbing Robots: A Review. *J. Bionic Eng.* **2022**, *19*, 877–901. [\[CrossRef\]](#)
22. Trivedi, D.; Rahn, C.D.; Kier, W.M.; Walker, I.D. Soft robotics: Biological inspiration, state of the art, and future research. *Appl. Bionics Biomech.* **2008**, *5*, 99–117. [\[CrossRef\]](#)
23. Xiang, W.; Honarvar Shakibaei Asli, B. Advances in Gecko-Inspired Climbing Robots: From Biology to Robotics—A Review. *Electronics* **2025**, *14*, 2810. [\[CrossRef\]](#)
24. Simmons, R.; Lin, L.J.; Fedor, C. Autonomous task control for mobile robots. In Proceedings of the Proceedings 5th IEEE International Symposium on Intelligent Control 1990, Philadelphia, PA, USA, 5–7 September 1990; Volume 2, pp. 663–668. [\[CrossRef\]](#)
25. Elfes, A. A distributed control architecture for an autonomous mobile robot. *Artif. Intell. Eng.* **1986**, *1*, 99–108. [\[CrossRef\]](#)
26. Majorana, S.; Chua, L.O. A unified framework for multilayer high order CNN. *Int. J. Circuit Theory Appl.* **1998**, *26*, 567–592. [\[CrossRef\]](#)
27. Arena, P.; Fortuna, L.; Frasca, M.; Patane, L.; Pavone, M. Climbing obstacles via bio-inspired CNN-CPG and adaptive attitude control. In Proceedings of the 2005 IEEE International Symposium on Circuits and Systems, Kobe, Japan, 23–26 May 2005; Volume 5, pp. 5214–5217. [\[CrossRef\]](#)
28. Murphy, M.P.; Sitti, M. Waalbot: An Agile Small-Scale Wall-Climbing Robot Utilizing Dry Elastomer Adhesives. *IEEE/ASME Trans. Mechatron.* **2007**, *12*, 330–338. [\[CrossRef\]](#)
29. Meng, C. Design and Analysis of Gecko-like Robot. *Chin. J. Mech. Eng.* **2011**, *24*, 224. [\[CrossRef\]](#)
30. Srisuchinnawong, A.; Shao, D.; Ngamkajornwiwat, P.; Teerakittikul, P.; Dai, Z.; Ji, A.; Manoonpong, P. Neural Control for Gait Generation and Adaptation of a Gecko Robot. In Proceedings of the 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2–6 December 2019; pp. 468–473. [\[CrossRef\]](#)
31. Pei, X.; Liu, S.; Wei, A.; Shi, R.; Dai, Z. Bioinspired Rigid–Flexible Coupled Adaptive Compliant Motion Control of Robot Gecko for Space Stations. *Biomimetics* **2023**, *8*, 415. [\[CrossRef\]](#)
32. Sun, T.; Dai, Z.; Manoonpong, P. Robust and reusable self-organized locomotion of legged robots under adaptive physical and neural communications. *Front. Neural Circuits* **2023**, *17*, 1111285. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Zhu, Y.; Cao, A.; He, Z.; Zhou, M.; Li, R. Coordinated Locomotion Control for a Quadruped Robot with Bionic Parallel Torso. *Biomimetics* **2025**, *10*, 335. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Li, Z.; Pang, C.; Dong, C.; Zeng, X. R-YOLOv5: A Lightweight Rotational Object Detection Algorithm for Real-Time Detection of Vehicles in Dense Scenes. *IEEE Access* **2023**, *11*, 61546–61559. [\[CrossRef\]](#)
35. Khanam, R.; Hussain, M. What is YOLOv5: A deep look into the internal features of the popular object detector. *arXiv* **2024**, arXiv:2407.20892. [\[CrossRef\]](#)
36. Lavanya, G.; Pande, S.D. Enhancing real-time object detection with YOLO algorithm. In *EAI Endorsed Transactions on Internet of Things*; European Alliance for Innovation (EAI): Bratislava, Slovakia, 2023; Volume 10.
37. Wang, J.; Han, L.; Jiang, Y.; Qi, Y.; Altangerel, K. Monocular visual obstacle avoidance method for autonomous vehicles based on YOLOv5 in multi lane scenes. *Alex. Eng. J.* **2024**, *109*, 497–507. [\[CrossRef\]](#)
38. Yang, S.; Xu, A.; Li, P.; Chen, M.; Du, P.; Shao, K. Visual SLAM Algorithm Based on YOLOv5 in Dynamic Scenario. In Proceedings of the 2023 China Automation Congress (CAC), Chongqing, China, 17–19 November 2023; pp. 2640–2645. [\[CrossRef\]](#)
39. Liu, H.; Liu, W.; Alblehai, F. A Dynamic Obstacle Avoidance Method in Robotic Navigation Control Through YOLOv5-Based Object Detection. *J. Circuits Syst. Comput.* **2024**, *33*, 2450269. [\[CrossRef\]](#)
40. Adiuku, N.; Avdelidis, N.; Tang, G.; Plastropoulos, A.; Diallo, Y. Mobile Robot Obstacle Detection and Avoidance with NAV-YOLO. *Int. J. Mech. Eng. Robot. Res.* **2024**, *13*, 219–226. [\[CrossRef\]](#)
41. Abood, S.; Al-Khafaji, H.; Al-Khafaji, M. Enhancing Collision Avoidance in Mobile Robots Using YOLOv5: A Lightweight Approach for Unstructured Environments. *J. Robot. Control (JRC)* **2025**, *6*, 769–778. [\[CrossRef\]](#)
42. Matsuoka, K. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biol. Cybern.* **1985**, *52*, 367–376. [\[CrossRef\]](#)

43. Kamimura, A.; Kurokawa, H.; Yoshida, E.; Tomita, K.; Kokaji, S.; Murata, S. Distributed adaptive locomotion by a modular robotic system, M-TRAN II. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2370–2377. [[CrossRef](#)]
44. Crespi, A.; Badertscher, A.; Guignard, A.; Ijspeert, A.J. AmphiBot I: An amphibious snake-like robot. *Robot. Auton. Syst.* **2005**, *50*, 163–175. [[CrossRef](#)]

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.



DSPACE

<https://dspace.org/>

A vision–locomotion framework toward obstacle avoidance for a bio-inspired gecko robot

Xiang, Wenrui; Honarvar Shakibaei Asli, Barmak; Ji, Aihong

2026-02-02

MDPI

<https://doi.org/10.3390/electronics15040882>

Xiang W, Honarvar Shakibaei Asli B, Ji A. (2026) A vision–locomotion framework toward obstacle avoidance for a bio-inspired gecko robot. Electronics, Volume 15, Issue 4, February 2026, Article number 882