BOLT- Behavioral Object Locomotion & Tracking For Unitree Go2 Robot

EAS 563 AI Capstone under Prof. David Doermann
Submitted by

Sahil Shivaji Sawant
UB ID: 50606712
Email: sahilshi@buffalo.edu

Introduction:

Quadruped robots are increasingly being deployed in real-world environments that are dynamic, unstructured, and unpredictable. To operate effectively in such settings, these robots must possess robust perception and tightly coordinated motion control. The Unitree Go2 robot demonstrates strong locomotion capabilities; however, it lacks an integrated autonomous vision-to-motion pipeline capable of sustained, closed-loop object tracking and following ability.

The B.O.L.T (Behavioral Object Locomotion & Tracking) project addresses this limitation by enabling real-time, vision-guided object following. The system allows the Go2 robot to continuously detect, track, and move toward a target object (here a multi-colored ball) using a low-latency perception-to-control loop.
This capability is critical for applications such as search and rescue, service robotics, and warehouse automation, where robots must respond quickly and reliably to moving targets.

Problem Statement:

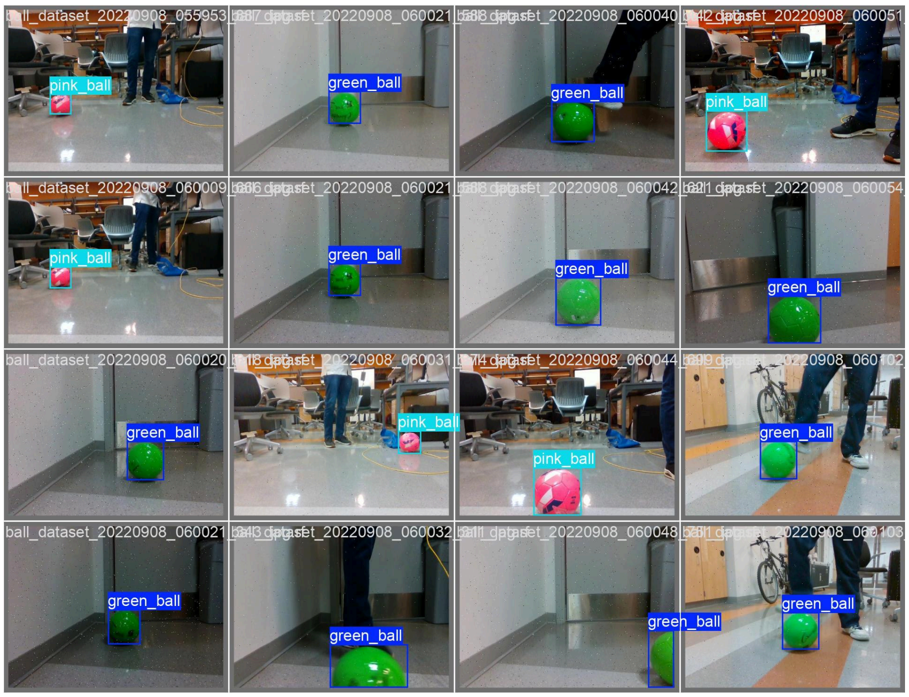
While the Unitree Go2 platform provides reliable locomotion and basic perception, it does not natively support **autonomous behavioral following based on continuous visual feedback**. Key technical challenges include:

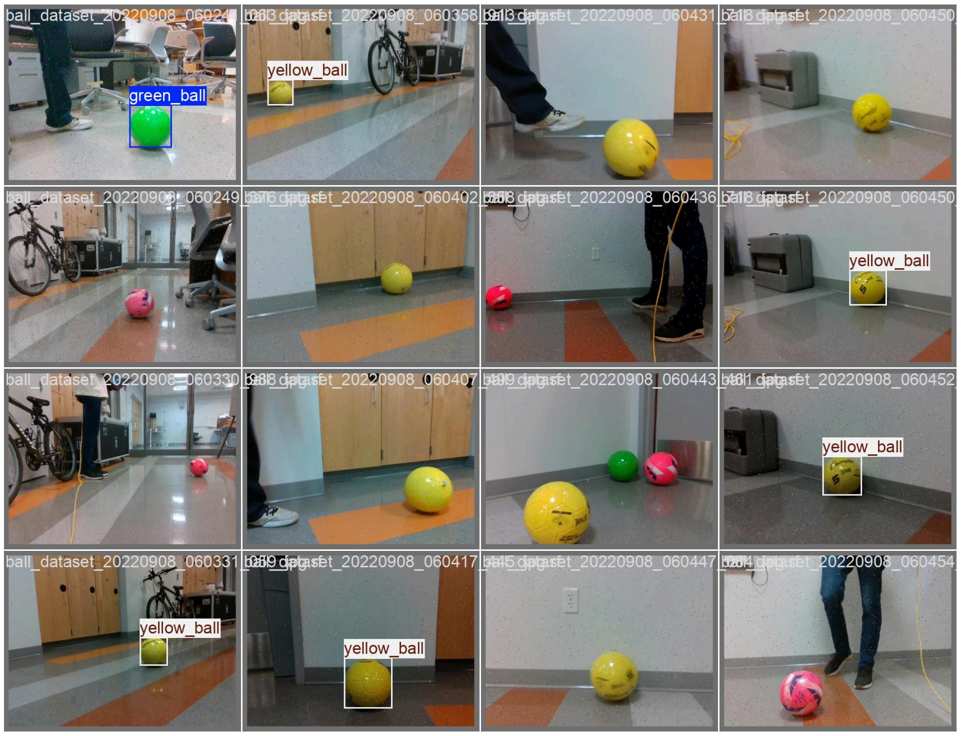
- **Perception-to-Control Gap:** Frame-based 2D vision detections must be transformed into smooth, temporally consistent velocity commands suitable for stable quadruped locomotion. Direct coupling leads to oscillations, overshoot, and loss of target tracking.
- **Latency and Synchronization:** Vision pipelines often operate at higher update rates than motion controllers, causing the robot to act on stale perception data during movement and reducing tracking responsiveness.
- **Environmental Robustness:** Monocular vision is sensitive to lighting variations, shadows, and background clutter, making reliable object detection and distance estimation difficult in uncontrolled environments.
- **Onboard Software Integration:** Deployment on embedded hardware requires strict compatibility across ROS/ROS2, CUDA, TensorRT, deep learning frameworks, and device drivers. Frequent SDK and library upgrades introduce breaking changes, necessitating repeated system reconfiguration and validation.

These challenges motivate a **decoupled, low-latency perception and control architecture**, as adopted in the proposed methodology, enabling reliable real-time object tracking under practical embedded system constraints.

Dataset Preparation

A custom dataset was created by manually capturing 20–30 images per object class using the onboard camera to reflect real deployment conditions. Images were annotated in Roboflow by drawing bounding boxes around each target object. To improve generalization, the dataset was augmented using geometric and photometric transformations. The final augmented dataset was used to train a YOLOv8 object detection model, optimized for real-time inference on edge hardware.





Limitations of Traditional Approaches

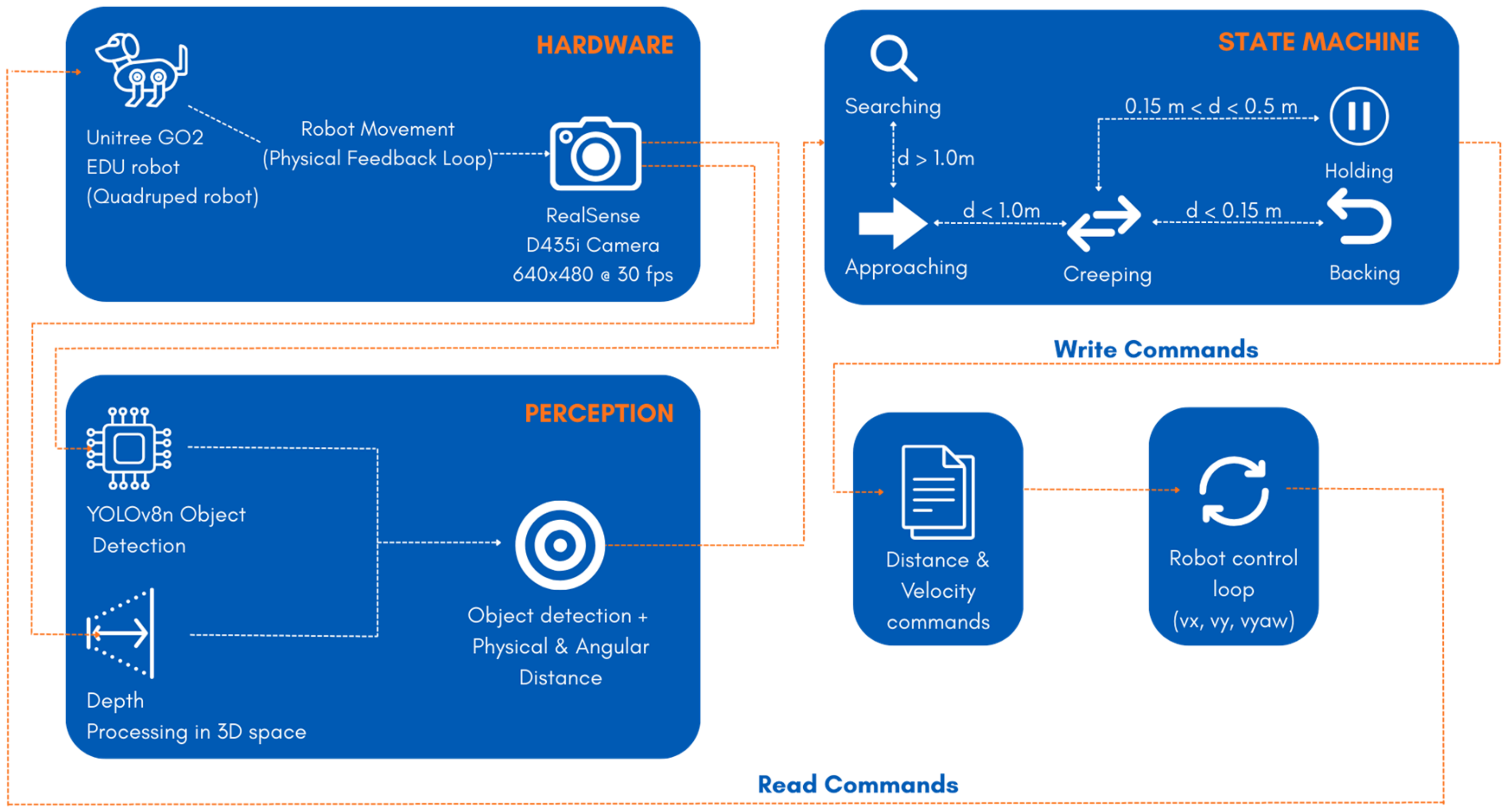
Several limitations in existing robotic perception and control paradigms motivated the design of the BOLT system:

- **2D Vision Systems:** Monocular vision provides only image-plane information, making accurate depth and distance estimation unreliable, particularly under poor, dynamic, or inconsistent lighting conditions.
- **SLAM-Based Systems:** While effective in structured environments, SLAM approaches require extensive calibration, stable visual features, and high-end computational resources, reducing their suitability for lightweight, deployable robotic platforms.
- **Tightly Coupled Architectures:** In many systems, perception pipelines operate faster than motion controllers, leading to control decisions based on outdated sensory inputs. This mismatch results in missed targets, delayed reactions, and degraded tracking performance during motion.

Hardware and Control Constraints

In addition to algorithmic limitations, the system design is constrained by hardware and control challenges:

- **Non-Linear Quadruped Dynamics:** Translating high-level velocity commands into stable, coordinated leg motion is inherently complex due to the non-linear dynamics of quadruped locomotion, requiring careful control abstraction.
- **Camera Calibration Sensitivity:** Accurate intrinsic calibration, particularly focal length estimation, is critical for reliable real-world distance and angle computation. Small calibration errors can lead to significant deviations in motion behavior.


HARDWARE
Unitree GO2 EDU robot (Quadruped robot)
Robot Movement (Physical Feedback Loop)
RealSense D435i Camera 640x480 @ 30 fps
STATE MACHINE
Searching
d > 1.0m
Approaching
d < 1.0m
Creeping
0.15 m < d < 0.5 m
Holding
d < 0.15 m
Backing
Write Commands
PERCEPTION
YOLOv8n Object Detection
Depth Processing in 3D space
Object detection + Physical & Angular Distance
Distance & Velocity commands
Robot control loop (vx, vy, vyaw)
Read Commands

System Overview and Methodology
The BOLT pipeline adopts a decoupled architecture, separating perception from motion control to ensure up-to-date sensory feedback drives robot behavior.

Perception Pipeline
- A monocular RGB camera mounted on the Unitree Go2 captures real-time video frames.
- A quantized YOLOv8n object detection model runs on a Jetson Nano, optimized using ONNX for edge deployment.
- Object distance and angle are estimated using a focal-length–based geometric method.

Control Pipeline
- The estimated distance and angle are fed into a state-based control system.
- Each state generates velocity commands (vx, vyaw) that are sent to the robot.
- The entire system is managed using the ROS / ROS2 framework, ensuring reliable communication between perception and locomotion modules.

Closed-Loop Execution and System Flow

As illustrated in the above **figure**, the BOLT system operates as a **fully closed-loop perception–control pipeline**, integrating hardware, perception, decision-making, and locomotion into a continuous feedback cycle.

The process begins with the **RGB camera mounted on the Unitree Go2**, which captures real-time visual input while the robot is in motion. These frames are processed by the perception module, where a **YOLOv8-based object detector** identifies the target (Multi color balls))and estimates its relative **distance and angular offset** using geometric depth approximation and depth camera. This perception output represents the robot's current understanding of the environment.

The estimated distance and orientation are then passed to the **state-based decision layer**, which determines the appropriate behavioral state. Based on the active state, **velocity commands** are generated and written to the robot control interface (`vx`, `vy`, `vyaw`). The robot executes these commands, physically moves in the environment, and produces new camera observations, thereby closing the feedback loop.

This closed-loop execution ensures that **every control decision is driven by the most recent sensory input**, allowing the robot to immediately adapt to changes in target position, motion, or visibility.

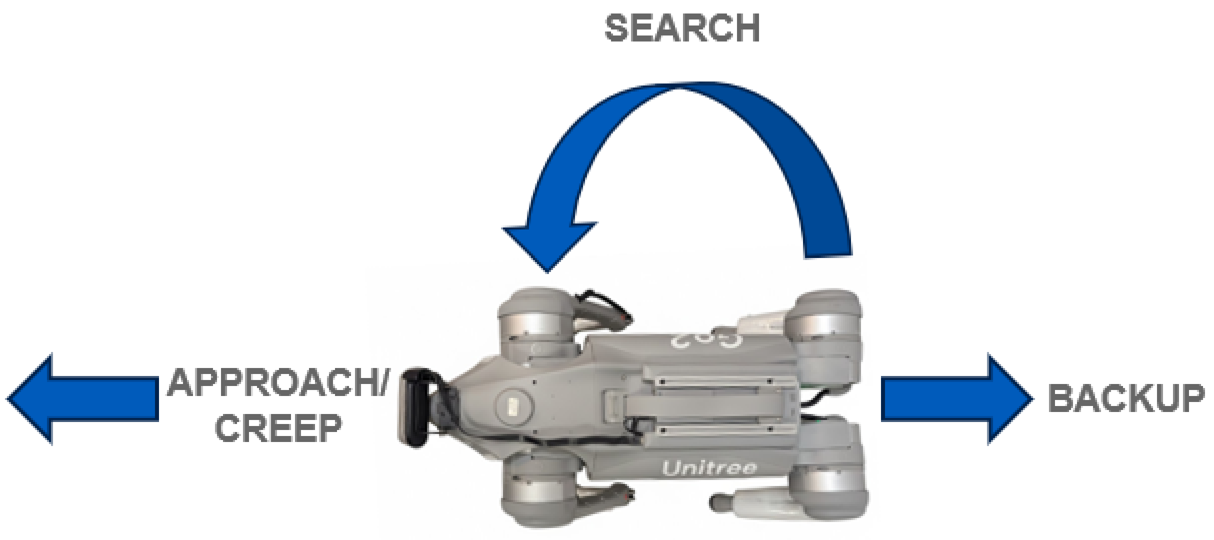
State-Based Control System

At the core of the control architecture is a **finite-state machine (FSM)**, shown in the top-right portion of the diagram. The FSM abstracts noisy perception outputs into a small set of well-defined, interpretable behaviors, ensuring smooth transitions and stable locomotion.

The system transitions between states based on the **estimated distance to the target**:

- **SEARCH:** When no valid target is detected, the robot enters a search behavior and rotates in place (`vyaw = 0.4`) to scan the environment and reacquire the object.
- **APPROACH:** Once the target is detected at a large distance (`d > 1.0 m`), the robot moves forward at a higher velocity (`vx = 0.30`) to rapidly close the gap.
- **CREEP:**As the robot approaches the target (`d < 1.0 m`), forward velocity is reduced (`vx = 0.15`) to allow precise positioning and prevent overshooting.
- **HOLD:** When the robot reaches the desired following range (`0.15 m < d < 0.5 m`), it maintains position with minimal movement (`vx = 0`), continuously correcting small deviations using visual feedback.
- **BACKUP:** If the robot gets too close to the target (`d < 0.15 m`), a safety behavior is triggered, and the robot reverses (`vx = -0.15`) to reestablish a safe distance.

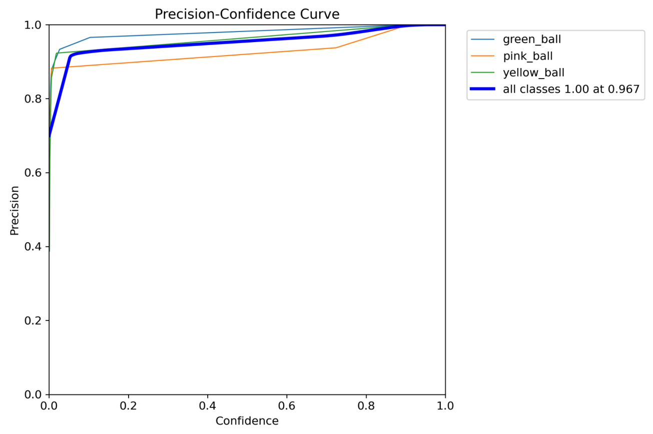
Transitions between states occur continuously based on updated distance estimates, ensuring smooth behavior without abrupt motion changes.

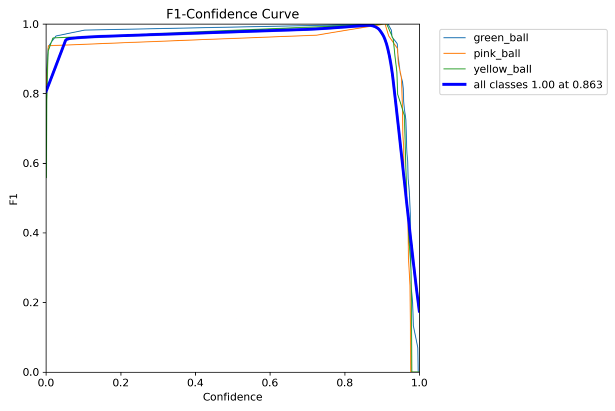


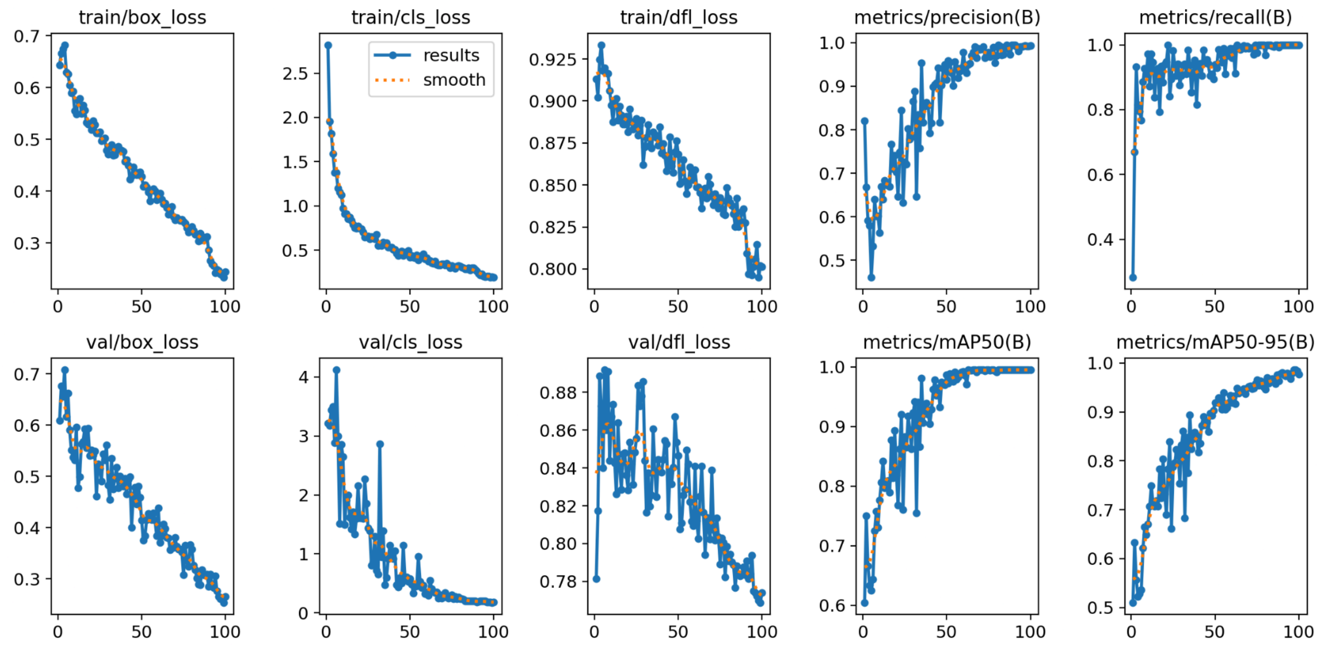
Results and Performance Evaluation

The BOLT system demonstrates **highly reliable real-time object detection and tracking**, as evidenced by both perception metrics and end-to-end system performance. Evaluation of the YOLOv8-based perception module shows **near-perfect detection quality across all target classes (green, pink, and yellow balls)**.

The **precision–confidence and recall–confidence curves** indicate that the detector maintains high precision and recall across a wide range of confidence thresholds. The **F1–confidence curve** peaks near optimal confidence values, confirming a well-balanced trade-off between false positives and false negatives. The model achieves an overall **mAP@0.5 of 0.995**, demonstrating consistent detection accuracy across all classes.







From a systems perspective, the closed-loop perception–control pipeline achieved the following performance:

- **Average end-to-end latency:** 48 ms
- **Worst-case latency:** 61 ms
- **Target tracking success rate:** >90% in indoor environments
- **Operational stability:** Robust across varied lighting conditions and backgrounds

By eliminating reliance on SLAM-based localization and adopting a **decoupled perception and control architecture**, the system consistently maintains **sub-50 ms average latency**, enabling responsive and stable quadruped navigation during active tracking.

Limitations

The system is constrained by **limited detection range (< 1.2m)** and camera field-of-view, which can lead to target loss during rapid motion. Performance may degrade under occlusion and motion blur, particularly during aggressive maneuvers.

Future Applications and Scope

Future work includes voice-guided target selection with re-localization, and deployment in challenging or outdoor environments to improve robustness and applicability.

Learning Outcomes

This project provided practical experience in deploying models on edge devices, robotic system design, hardware and network integration, and end-to-end autonomous system deployment.

References
Unitree Robotics SDK2: https://github.com/unitreerobotics/unitree_sdk2

Acknowledgments
A2IL Lab, Dept. Of CSE, University at Buffalo for their guidance and support.

Helpful Links
How to guide: Link
Github Repo: Link
Video demonstration: Link 1 , Link 2
Poster: Link