1 Introduction

This report summarizes the progress of Team SPARK for February 2025. The team focused on improving lane-keeping, optimizing camera processing speed, and enhancing decision-making using object detection. Significant progress was made in integrating YOLOv8-based traffic sign detection and refining vehicle control logic. However, challenges such as real-time processing delays on RPi remain, affecting high-speed performance.

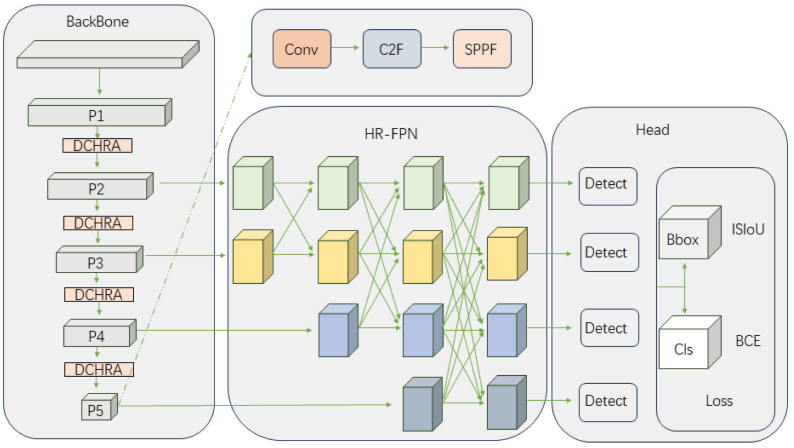
2 Planned activities

Activity	Members	Type of Activity
1. Vehicle control via .py script (decision-making)	Tai, Hieu, Long	Development
2. Improve lane-keeping	Hieu, Long, Nhi	Development
3. Optimize camera processing speed for real-time control	Tai, Ha, Long	Optimization
4. Implement obstacle avoidance behavior	Nhi, Ha, Hieu	Development
5. Document progress and record video	All	Reporting

3 Status of planned activities

Activity	Status	Implementation	Difficulties
1. Vehicle control via .py script (decision-making)	Ongoing	Developed initial logic for decision-making based on traffic rules in documentation, integrating object detection results into decision-making.	Processing speed limitations on RPi cause delays in real-time decision-making, requiring further optimization.

YOLOv8 Architecture Overview for Decision-making



- **Backbone**: YOLOv8 uses CSPDarknet53, an improved Darknet variant with Cross-Stage Partial (CSP) connections, enhancing feature extraction and gradient flow.
- **Neck**: Incorporates Path Aggregation Network (PANet) to improve multi-scale feature fusion.

● **Head**: Features multiple detection heads for bounding boxes, class probabilities, and objectness scores. ● **Detection Head**: Uses dynamic anchor assignment and an IoU-based loss for better accuracy and handling of overlapping objects.			
2. Improve lane-keeping	Ongoing	Using the camera to detect lanes and control the steering to follow the detected path, including turning based on the lane.	Lane-keeping is stable at low speeds, but delays in camera processing on RPi affect control at higher speeds, requiring optimization for high performance.
3. Optimize camera processing speed for real-time control	Ongoing	Working on reducing delay in transferring camera data to RPi for faster response times.	
4. Implement obstacle avoidance behavior	Postponed	Planned for the next phase after improving lane-keeping and decision-making.	Pending lane-keeping optimization.

4 General status of the project

The team has achieved significant progress in developing core functionalities:

Accomplishments:

- Improved lane-keeping performance, with good accuracy at low speeds in real-world testing.
- Started optimizing camera processing speed to improve real-time decision-making.
- Integrated YOLOv8-based traffic sign detection into the system, enabling the car to recognize signs dynamically.

Challenges:

- Lane-keeping at high speeds is unstable due to slow processing time between the camera and RPi.

5 Upcoming activities

- Continue improving lane-keeping to match high performance.
- Try with Jetson Nano as a replacement for Raspberry Pi to achieve high performance.
- Continue to develop a program for vehicles to run on pre-determined path.