PALETTE – Patrol Ai with Lane-based Encoded Track Transition Estimation

Dong Seoung Seo*, Jun Hyeok Lee**, Dong Hoon Oh**
*Dept. of Automotive Engineering, Hanyang Univ.
** Dept. of Mechanical Engineering, Hanyang Univ.,

(Received June 19, 2025)

Key Words: Lane-based navigation, UGV, YOLO, FSM, simulation-to-real transfer, autonomous driving

Abstract: In this study, we propose **PALETTE**, a lane-based autonomous patrol system designed for Unmanned Ground Vehicles (UGVs) that utilizes color-encoded tracks for navigation. The system employs a lightweight YOLO-based perception module and a Finite State Machine (FSM) for robust decision-making. To address the domain gap between virtual and real environments, we introduce three training strategies—simulation-only, real-only, and hybrid simNreal data—and compare their performance. Our system is validated through real-world and simulated track environments constructed to mirror each other. We analyze performance using standard metrics (F1 score, mAP) and a novel behavior-based accuracy metric based on state transitions. The results show that the hybrid training strategy significantly improves robustness across diverse lighting and environmental conditions.

1. Introduction

1.1 Color-Coded Road Markings in Korea

In South Korea, color-coded lane markings are extensively used to convey navigational and safety information. Yellow lines indicate opposing traffic, white lines separate lanes in the same direction, and special-purpose lanes such as bus lanes (blue), bicycle lanes (green), and safety zones (red) are clearly distinguished by their unique colors. These visual cues are designed to be immediately interpretable by human drivers. Inspired by this visual infrastructure, we adopt a color-lane encoding scheme for indoor UGV navigation. Our approach reinterprets such color semantics not for human drivers, but for autonomous systems, turning lane colors into machine-readable instructions.

Fig. 1 Color-Coded Road Markings

1.2 Motivation and Problem Definition

Traditional autonomous navigation systems rely heavily on GPS, LiDAR, or SLAM algorithms, which may be cost-prohibitive or overkill for **indoor patrol scenarios** in constrained environments like factories, warehouses, or research labs. Furthermore, small-scale autonomous systems often lack the computational and sensory resources needed for such complex approaches. To address these limitations, we propose a low-cost, lightweight system using a monocular camera and a color-based visual pipeline. The robot infers its behavioral state from the color of the lane it sees and adjusts its motion accordingly using a Finite State Machine (FSM). This approach enables structured, interpretable patrol logic without external localization.

1.3 Contribution and Overview of PALETTE

We present **PALETTE (Patrol AI with Lane-based Encoded Track Transition Estimation)**, a color-lane-driven autonomous navigation system for UGVs. Our contributions include: A lane color-based encoding scheme for defining patrol and transition logic. FSM-based decision system that governs patrol state transitions. A YOLO-based lightweight perception module trained on **simulated, real, and mixed datasets**. A real-world and simulation environment designed for consistent benchmarking. A novel performance metric based on **search state frequency** to quantify behavior reliability. Through systematic training and evaluation across domains, PALETTE demonstrates the feasibility and robustness of semantic color-lane navigation for UGVs operating in structured environments.

2. Methodology

2.1 System Architecture

The PALETTE system is composed of both hardware and software modules, integrated to support autonomous patrol of a UGV based on color-coded lanes. The architecture consists of:

UGV Platform: A 4WD differential-drive robot, controlled via PWM in real hardware and via position control in simulation.

Onboard Processing: NVIDIA Jetson Nano running the YOLOv8 model and FSM-based control logic in real-time.

Camera Module: A CSI camera mounted to face downward, providing visual input for lane detection.

Visual Detection: A YOLOv8 engine trained to detect colored lane segments (yellow, blue, red, white, green, purple) representing patrol or transition instructions.

Finite State Machine (FSM): Encodes patrol logic and defines transitions between patrol zones or recovery behaviors.

TCP Communication: A bridge between Jetson and the simulation PC for state synchronization and motion visualization.

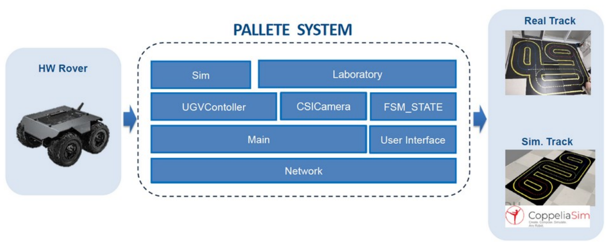

Fig. 2 PALETTE System

2.2 Environment and Lane Encoding

To ensure a fair comparison between simulation and real-world performance, we constructed two identical 5m × 5m tracks: **Real Track** - Constructed in a laboratory (HYU B2 210) using colored masking tape and black paper to create sharp color contrast. **Simulation Track** - Built in CoppeliaSim using a customized URDF of the UGV and the Bullet 2.78 physics engine. Patrol Zones - Yellow (Zone Y), Blue (Zone B), Red (Zone R). **Bridge Transitions** - White (common connector), Green and Purple (trigger zone transitions). The camera detects lane colors and passes this information to the FSM to determine current behavior and whether to transition to a new state.

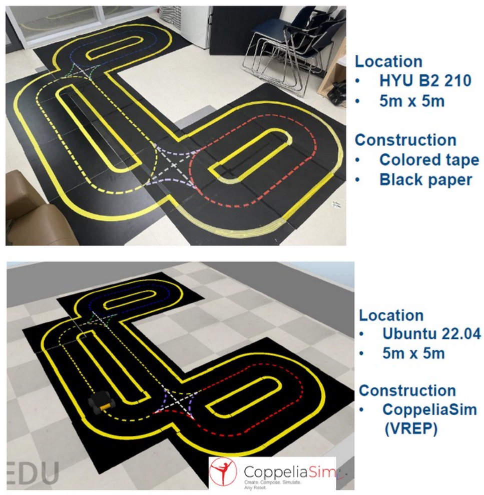

Fig. 3 Real Track and Sim. Track

2.3 Control Strategy and FSM Logic

Control is governed by two components: A steering value s derived from the detected lane position modulates wheel speeds: v is the base speed and α is a gain coefficient.

$$L = clip(v * (1 - \alpha s)) \quad (1)$$
$$R = clip(v * (1 + \alpha s)) \quad (2)$$

The FSM defines current behavior (e.g., YELLOW_LOOP) and transitions based on color inputs and detection stability (e.g., green or purple must persist over 2+ frames to confirm transition). Search states are entered when no valid lane is detected, and recovery behavior is triggered.

3. Experiments

3.1 Dataset Collection and Preparation

To evaluate the PALETTE system across different domains, we constructed three datasets tailored for color lane detection. The **simulation dataset** consisted of 3,821 frames generated from a CoppeliaSim environment, where the lighting and background conditions were consistent and noise-free. The **real-world dataset** contained 3,840 frames captured from an indoor 5m × 5m physical track, constructed using colored tape and black paper. To enhance generalization, we created a third dataset, **simNreal**, by combining equal portions of the two domains for a total of 7,660 samples. All datasets were annotated with bounding boxes for six lane color classes: yellow, blue, red (patrol zones), and white, green, purple (transition cues). Each dataset was used to train a separate YOLOv8 model under the same training configuration. A common evaluation set of 249 images—120 from simulation and 129 from the real track—was used to assess all three models.

3.2 Object Detection Performance

The object detection results highlighted the trade-offs between domain specialization and generalization. The **model trained only on real data** performed best in terms of precision and recall on real-world test images, achieving an F1 score of 0.868 and mAP@50 of 0.909. However, its performance declined when applied to simulation data, showing limited transferability. The **simulation-only model**, on the other hand, struggled with detecting key lane colors such as green, white, and purple in the real domain, resulting in a lower F1 score of 0.802. The **simNreal hybrid model** achieved balanced results across both domains, with an F1 score of 0.841 and a notable improvement in purple-line detection—about a 40% increase over the real-only model. These findings demonstrate that incorporating domain-diverse training data leads to better generalization, especially for complex or underrepresented lane cues.

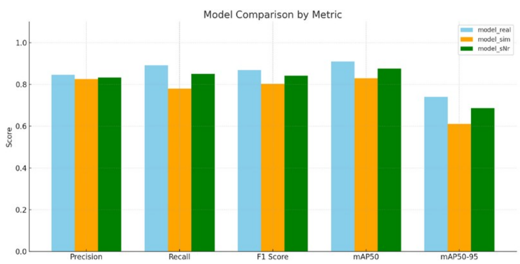

Fig. 4 Inference Score of Three Models

Model	Best Epoch	Precision	Recall	F1 Score	mAP50	mAP50-95
model_real	40	0.845	0.892	0.868	0.909	0.74
model_sim	26	0.825	0.78	0.802	0.828	0.611
model_simNreal	31	0.833	0.85	0.841	0.876	0.687

Table. 1 Inference Score of Three Models

3.3 Behavioral Evaluation Using FSM Logs

To evaluate how detection accuracy translated into real-world behavior, we measured each model's effect on the robot's patrol performance. Specifically, we counted the number of times the robot entered the **"SEARCH" state**, which is triggered when no lane is detected. This metric serves as a proxy for perception-induced navigation failure. The **simulation-only model** exhibited unstable behavior, entering the search state 94 times during an 80-second test run and ultimately failing to complete the patrol. The **real-only model** completed the patrol in 80 seconds but required four search-state recoveries. In contrast, the **simNreal model** completed the loop in just 68 seconds with only two brief search-state episodes, reflecting more stable and consistent behavior throughout the patrol.

Model	Search State Coun t	Duration (s)
model_simNreal	2	68
model_real	4	80
model_sim (fail)	94	80

Table. 2 Reverse Driving for Acc. Metric

3.4 Real-Time Control and Synchronization

The PALETTE system's robustness was further enhanced through dynamic motion control and simulation synchronization. The robot's wheel velocities were continuously adjusted based on steering values derived from detected lane geometry. A clipping function prevented excessive speed differences, ensuring smoother turns and avoiding sharp divergence. In parallel, a TCP socket transmitted FSM states, steering gains, and left/right wheel speeds from the Jetson Nano to a PC-based CoppeliaSim visualization. This setup provided a synchronized, interpretable visual feedback loop, aiding in both debugging and system evaluation.

4. Conclusion

In this work, we presented PALETTE, a vision-based autonomous patrol system for UGVs that utilizes color-encoded lane markings and FSM-driven behavior logic. Through the construction of matched real and simulated environments, we demonstrated that lane color can effectively serve as both navigational and semantic control input. Our experiments showed that while models trained solely on real or simulated data perform well in their respective domains, the hybrid simNreal model consistently achieved higher robustness and behavioral stability. The proposed system highlights the feasibility of low-cost, interpretable autonomous navigation in structured indoor environments, and lays the groundwork for future extensions involving multi-agent coordination, LED-based path encoding, or adaptive state learning.

Appendix A. Track Design

To evaluate the system under consistent visual constraints, we constructed a 5m × 5m closed-loop track in both simulation and the real world. The real track was built in the HYU B2–210 laboratory using high-contrast colored masking tape for lane markings and black matte paper for background suppression. Each patrol zone (yellow, blue, red) and transition zone (white, green, purple) was laid out to form a circular or figure-eight loop. The simulated track was implemented in CoppeliaSim with the same layout, camera angle, and lighting assumptions.

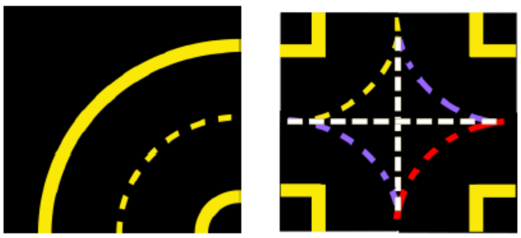

Fig. 5 Track Images for Simulation Enviroment

Appendix B. Transition Table

The Finite State Machine (FSM) logic used in PALETTE consists of 10 states. Each transition is triggered by consistent detection of a specific color class (green or purple) over at least two consecutive frames to avoid false triggers. The FSM also enforces a minimum time in each state to prevent unstable flip-backs.

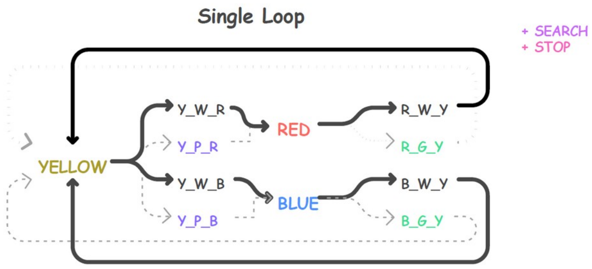

Fig. 6 Scenario #1

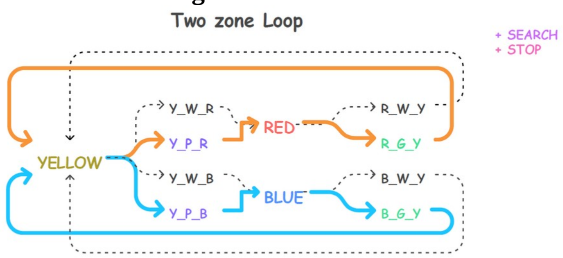

Fig. 7 Scenario #2

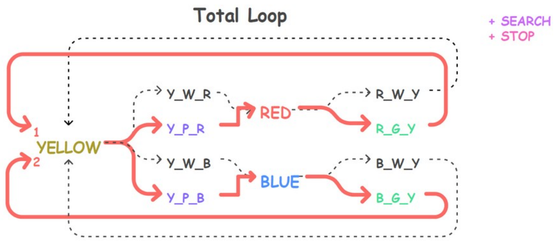

Fig. 8 Scenario #3

Reference

[1] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, and others, “**Ultralytics YOLOv5: Cutting-edge object detection**,” 2020

[2] J. Kim and C. Kim, “Robust Lane Detection Based on Convolutional Neural Network and Color Filtering,” Journal of Advanced Transportation, vol. 2017, Article ID 5898103, 2017.

[3] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed., Springer, 2017.

[4] S. James, A. J. Davison, and E. Johns, “**Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task**,” in *Proc. Conf. on Robot Learning (CoRL)*, 2017. [Online]. Available: https://arxiv.org/abs/1707.02267

[5] H. Jung, B. Kim, and J. Kim, “**Vision-Based Indoor Navigation Using Color-Coded Landmarks and FSM-Based Planning**,” *IEEE Sensors Journal*, vol. 20, no. 15, pp. 8541–8549, Aug. 2020,