Robust Lane Tracking in Non-Standard Road Conditions using Confidence-Adaptive EKF

Bumyeon Lee
Dept. of Mechanical Engineering
Keimyung University
Daegu, South Korea
claypooky@naver.com

Jongbin Kim
Dept. of Automotive Engineering
Keimyung University
Daegu, South Korea
kjb12100@naver.com

Junghyun Choi*
Dept. of Automotive Engineering
Keimyung University
Daegu, South Korea
jh.choi@kmu.ac.kr

***Abstract*—Detection of damaged or non-standard lane markings is a critical challenge for autonomous driving systems and automated road maintenance robots. This paper proposes a robust real-time lane tracking system that combines deep learning-based semantic segmentation with graph-based connectivity analysis and the confidence-adaptive Extended Kalman Filter (EKF). The proposed system uses SegFormer for segmenting the lane in pixels, followed by a graph model that establishes spatial relationships between the detected lane segments. A mask enhancement module reconstructs discontinuous lane regions by analyzing window-wise pixel density and interpolating gaps based on neighboring valid segments. The confidence score, derived from pixel density and polynomial fit quality, dynamically adapts the EKF's measurement noise covariance matrix (R), enabling the system to trust visual measurements when lane quality is high and rely on motion predictions when quality degrades. The system was validated on an embedded autonomous vehicle platform equipped with NVIDIA Jetson AGX Orin, achieving real-time performance at 40 Hz with a lateral tracking error of 0.64 cm RMS on damaged lane sections. Experimental results demonstrate that the proposed method significantly outperforms both fixed-parameter filtering and conventional vision-based approaches in non-standard road conditions.**

***Index Terms*—Semantic Segmentation, Extended Kalman Filter (EKF), Adaptive Filtering, Lane Tracking, Graph Model, Autonomous Vehicles**

I. Introduction

Robust perception of lane markings is a fundamental requirement for the safety of autonomous vehicles and Advanced Driver-Assistance Systems (ADAS) [1], [2]. While lane detection technology has matured for ideal highway scenarios, real-world roads frequently present non-standard conditions, such as worn-out paint, physical damage, or complex shadows caused by roadside trees and structures.

Accurate detection in these "non-standard" environments is critical for two primary applications. First, it is essential for driving safety to prevent lane departure accidents, as active lane-keeping systems have been shown to significantly reduce crashes [3]. Recent studies have quantified that lane marking quality degradation can reduce lane departure warning system accuracy from 95% at 50% fading to as low as 54% at 75% fading [4]. Second, it is a prerequisite for automated road maintenance systems, such as lane-painting robots, which must reliably identify and trace damaged or faint trajectories to execute precise repairs. Recent studies have highlighted that robust detection under such varying conditions remains a significant challenge in the field [5].

Traditional lane detection methods typically employ edge detection [6], Hough transforms, or inverse perspective mapping with spline fitting [7]. Although computationally efficient, these methods fundamentally rely on the presence of strong, distinct image gradients and continuity. When the lane markings are worn, the image gradient becomes negligible, leading to detection failures. In addition, tree shadows or road cracks often create strong false gradients, leading algorithms to misidentify these artifacts as lane boundaries.

Previous attempts to improve robustness under challenging conditions include adaptive filtering and deep learning approaches. Dorj et al. [8] applied Kalman filtering for curved lane tracking, and Guo et al. [9] provided a comprehensive review of vehicle dynamic state estimation using Extended Kalman Filters (EKF). Borkar et al. [10] combined inverse perspective mapping with RANSAC and Kalman filtering for robust tracking. However, their reliance on fixed noise parameters limits performance when sensor reliability fluctuates due to environmental irregularities. Recent deep learning methods such as SCNN [11], LaneNet [12], SegFormer [13], and attention-based approaches [5] have demonstrated high precision and semantic understanding in complex scenarios, particularly on standard benchmarks such as TuSimple [14]. However, direct application of heavy models remains challenging for embedded platforms with strict power and latency constraints.

To address these limitations, we propose a hybrid approach that combines lightweight semantic segmentation with graph-based connectivity analysis and confidence-adaptive filtering. The SegFormer architecture [13] provides efficient semantic segmentation through its hierarchical Transformer encoder, while graph-based methods [15] enable robust connectivity modeling between discontinuous lane segments. By linking segmentation confidence directly to the Kalman filter's measurement noise covariance [16], the system automatically adjusts its trust in visual measurements based on lane quality.

The main contributions of this work are as follows. First, we propose a hybrid lane detection pipeline that combines SegFormer-based semantic segmentation with graph model connectivity analysis, enabling robust detection of damaged and discontinuous lane segments through a two-stage processing approach. Second, we introduce a mask enhancement module that reconstructs missing lane regions by analyzing pixel density within sliding windows and interpolating gaps based on polynomial geometry from neighboring valid segments. Third, we demonstrate real-time deployment on embedded hardware (Jetson AGX Orin) achieving a 40 Hz processing rate with 0.64 cm RMS lateral error on damaged lane sections.

II. System Architecture

A. Hardware Configuration

The experimental platform utilizes the ERP-42 autonomous vehicle platform, a compact electric research platform equipped with drive-by-wire capabilities. The perception and control system runs on an NVIDIA Jetson AGX Orin 64GB Developer Kit, providing 275 TOPS of AI compute performance in a 60W power envelope. Visual data is captured using a ZED 2i stereoscopic camera mounted on the vehicle's front, operating at 30 FPS with 2K resolution, though only the left camera feed is utilized for this monocular perception task.

B. Software Framework

The system is built on Ubuntu 20.04 with ROS Noetic serving as the middleware framework. The core algorithms are implemented in Python 3.8, utilizing OpenCV 4.5.4 for image processing, NumPy for numerical computations, CuPy for GPU-accelerated operations, TensorRT for optimized deep learning inference, and Numba JIT compilation for performance-critical functions.

The modular ROS architecture consists of several interconnected nodes, as shown in Fig. 1. The ZED ROS Wrapper publishes raw camera images, which are processed by the SegFormer inference node running TensorRT-optimized FP16 inference. The Graph Model node analyzes connectivity between detected lane segments, while the Mask Enhancer reconstructs discontinuous regions. Finally, the Lane EKF node maintains a filtered lane state with adaptive covariance, and the Controller node generates steering commands.

C. Data Processing Pipeline

The perception pipeline processes each frame through four main stages, as illustrated in Fig. 2. First, raw RGB images are preprocessed with ROI masking and resized to 512×512 pixels for SegFormer inference. The preprocessing applies ImageNet normalization (mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]) using GPU-accelerated CuPy operations. Second, the SegFormer model with MiT-B1 backbone performs pixel-wise semantic segmentation, producing a probability map that is thresholded at 0.5 to generate a binary lane mask. Third, the graph model extracts contour points from the mask, builds a connectivity graph based on spatial proximity, and merges segments that belong to the same lane. Fourth, the mask enhancer analyzes pixel density within sliding windows and reconstructs gaps where density falls below a threshold, using polynomial interpolation from neighboring valid segments.

III. Proposed Method

A. SegFormer-based Semantic Segmentation

The SegFormer architecture [13] is used for lane segmentation due to its efficient hierarchical design and strong performance on narrow, elongated structures. The Mix Transformer (MiT) encoder generates multi-scale features through overlapped patch merging, while the lightweight MLP decoder aggregates features from all stages for dense prediction.

For embedded deployment, the model is optimized using TensorRT with FP16 precision. The inference pipeline utilizes pinned memory and CUDA streams for efficient CPU-GPU data transfer. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, the network outputs a probability map $P \in [0,1]^{H\times W}$ representing the likelihood of the lane in each pixel.

$$M_{\text{seg}} = \mathbf{1}[P > \tau] \tag{1}$$

where $\tau = 0.5$ is the binarization threshold and $\mathbf{1}[\cdot]$ is the indicator function.

B. Graph-based Connectivity Analysis

The graph model establishes spatial relationships between the detected lane pixels to handle discontinuities caused by damage or occlusion. Contour points are extracted from the segmentation mask and used as graph nodes. Edges are created between nodes based on directional proximity constraints:

$$E(i,j) = \begin{cases} 1 & \text{if } |y_i - y_j| \leq d_y \text{ and } |x_i - x_j| \leq d_x \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where d_y and d_x are the maximum connection distances in the vertical and horizontal directions, respectively. The asymmetric constraint ($d_y > d_x$) reflects the elongated nature of the lane markings.

Connected components are extracted using the Union-Find algorithm and filtered by minimum segment length. Segments belonging to the same lane are merged on the basis of their mean x-position and y-range overlap:

$$\text{merge}(S_i, S_j) \Leftrightarrow |\bar{x}_i - \bar{x}_j| \leq g_x \text{ and } \text{gap}_y \leq g_y \tag{3}$$

where g_x and g_y are merge gap thresholds. Each merged segment is fitted with a polynomial:

$$x(y) = a_2 y^2 + a_1 y + a_0 \tag{4}$$

C. Mask Enhancement Module

The mask enhancer reconstructs discontinuous lane regions by analyzing the pixel density within sliding windows. For each window w_i centered at position (x_c, y_c) derived from the polynomial fit, the pixel count N_i is computed. Windows with

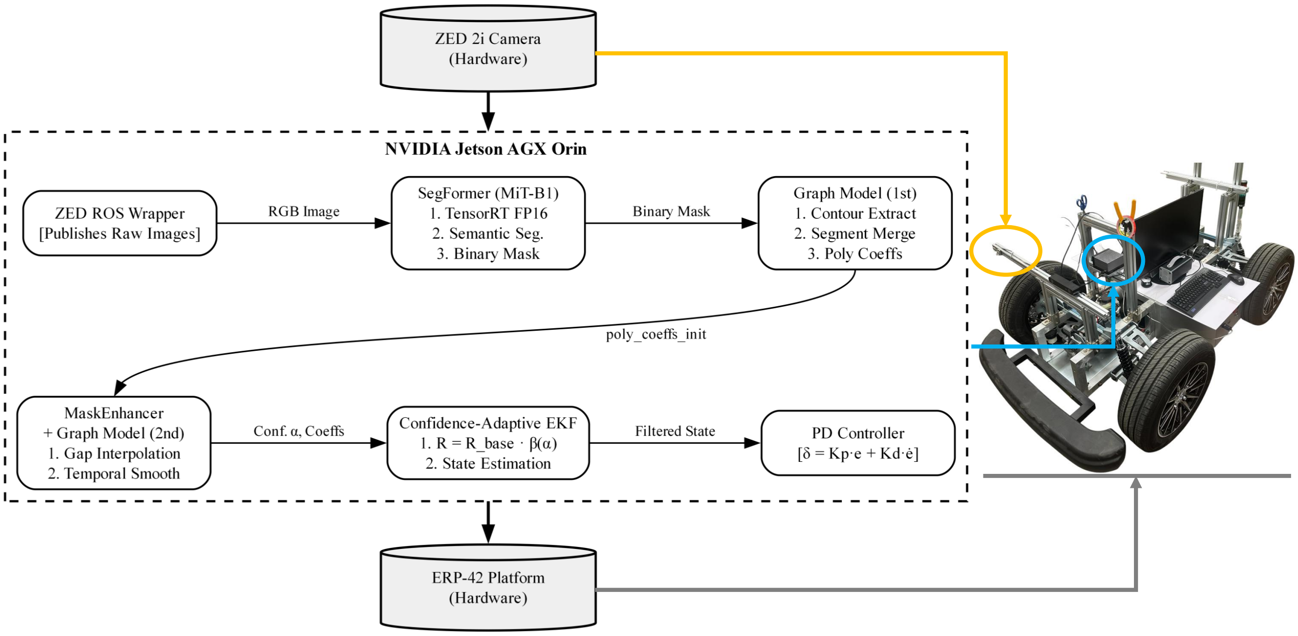


Fig. 1. System Architecture: The proposed pipeline processes camera images through SegFormer semantic segmentation, graph-based connectivity analysis, mask enhancement, and confidence-adaptive EKF. The confidence score derived from pixel density and fit quality dynamically adjusts the filter's measurement trust.

a density below a threshold ratio ρ of the maximum density are identified as gaps:

$$\text{gap}_i = \begin{cases} 1 & \text{if } N_i < \rho \cdot N_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\rho = 0.3$ is empirically determined.

For gap windows that have valid neighbors both above and below (ensuring the gap is within a continuous lane), the mask is filled using the polynomial centerline and the median lane width from neighboring windows:

$$w_{\text{ref}} = \text{median}(\{w_j : j \in \mathcal{N}_{\text{valid}}\}) \tag{6}$$

where $\mathcal{N}_{\text{valid}}$ is the set of neighboring windows with valid lane width measurements.

D. Confidence Score Computation

The confidence score $\alpha \in [0, 1]$ quantifies measurement quality based on pixel density and polynomial fit quality. The pixel count confidence is computed as

$$\alpha_{\text{pixels}} = \min\left(1, \frac{N_{\text{detected}}}{N_{\text{expected}}}\right) \tag{7}$$

where $N_{\text{expected}} = 100$ is empirically determined based on typical lane width and window size.

The fit error confidence uses exponential decay:

$$\alpha_{\text{fit}} = \exp\left(-\frac{\sigma_{\text{fit}}}{\sigma_{\max}}\right) \tag{8}$$

where σ_{fit} is the standard deviation of polynomial fit residuals and $\sigma_{\max} = 10$ pixels.

The combined confidence score includes a bias term to ensure minimum baseline stability:

$$\alpha = w_p \cdot \alpha_{\text{pixels}} + w_f \cdot \alpha_{\text{fit}} + w_b \tag{9}$$

with weights $w_p = 0.4$, $w_f = 0.4$, and bias $w_b = 0.2$.

E. Adaptive Extended Kalman Filter

The Extended Kalman Filter (EKF) [17] maintains a 6-dimensional state vector that represents polynomial coefficients and their rate of change:

$$\mathbf{x}_k = [a_2, a_1, a_0, \dot{a}_2, \dot{a}_1, \dot{a}_0]^T \tag{10}$$

The state evolution follows a constant-velocity model:

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \Delta t \cdot \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \tag{11}$$

The key innovation is confidence-based adaptation of measurement noise:

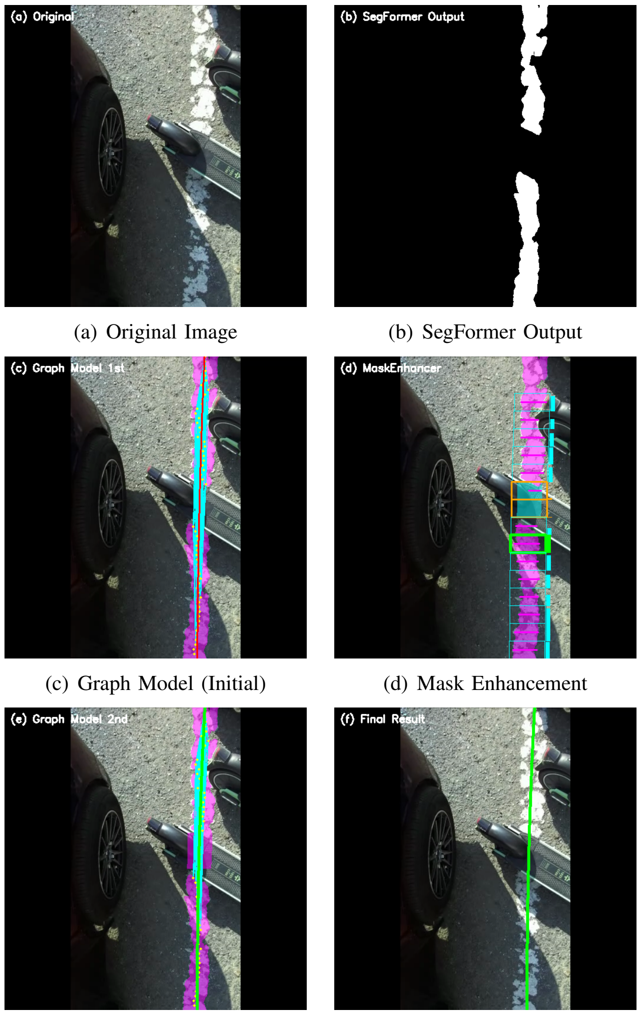
$$\mathbf{R}_k = \mathbf{R}_{\text{base}} \cdot \beta(\alpha_k), \quad \beta(\alpha) = 1 + (1 - \alpha)(S_{\max} - 1) \tag{12}$$

where $S_{\max} = 10$ is the maximum scaling factor. When confidence is high ($\alpha \approx 1$), the measurement noise remains at baseline, causing the filter to trust visual measurements. When confidence drops ($\alpha \approx 0$), the noise increases by factor $S_{\max}$, causing the filter to rely more on its motion model predictions.

F. Control Strategy

The filtered polynomial generates steering commands through lateral error computation at a look-ahead distance y_{la}:

$$e_{\text{lateral}} = x_{\text{ref}} - x(y_{\text{la}}) \tag{13}$$



(a) Original Image

(b) SegFormer Output

(c) Graph Model (Initial)

(d) Mask Enhancement

(e) Graph Model (Temporal)

(f) Final Detection

Fig. 2. Processing pipeline: (a) Original image, (b) SegFormer semantic segmentation output showing discontinuous mask regions, (c) Initial graph-based connectivity analysis with polynomial fitting (red curve), (d) Sliding window density analysis identifying gap regions (orange boxes) for mask enhancement, (e) Temporal-smoothed graph model result with refined polynomial (green curve), (f) Final lane centerline detection.

A proportional-derivative controller computes the steering angle:

$$\delta = K_p \cdot e_{\text{lateral}} + K_d \cdot \dot{e}_{\text{lateral}} \tag{14}$$

Gains $K_p = 500/256$ and $K_d = 200$ were determined by systematic tuning on the test track, following established PID tuning methodologies for lane keeping systems [18]. The proportional gain K_p is normalized by the half-width of the image (256 pixels) to produce steering angles in the range of ±55° for the vehicle's maximum steering capability. This normalization ensures consistent control behavior across different image resolutions. The derivative gain K_d acts on the polynomial slope to provide anticipatory steering into the curves, where the ratio $K_d/K_p \approx 0.1$ was selected to achieve sufficient damping while avoiding excessive oscillation during transient responses.

Algorithm 1 Confidence-Adaptive Lane Tracking

```
Require: Camera image $I$, Previous state $\mathbf{x}_{k-1}$
Ensure: Updated state $\mathbf{x}_k$, Steering $\delta$
$M_{\text{seg}} \leftarrow \text{SegFormer}(I)$ {Semantic segmentation}
$(P, E) \leftarrow \text{GraphModel}(M_{\text{seg}})$ {Connectivity analysis}
$\mathbf{c}_1 \leftarrow \text{PolyFit}(P, E)$ {Initial polynomial}
$M_{\text{enh}} \leftarrow \text{MaskEnhancer}(M_{\text{seg}}, \mathbf{c}_1)$ {Gap filling}
$\mathbf{c}_2 \leftarrow \text{PolyFit}(M_{\text{enh}})$ {Refined polynomial}
$\alpha \leftarrow \text{ComputeConfidence}(M_{\text{enh}}, \mathbf{c}_2)$
$\mathbf{R}_k \leftarrow \mathbf{R}_{\text{base}} \cdot (1 + (1 - \alpha)(S_{\max} - 1))$
$\mathbf{x}_k \leftarrow \text{EKF\_Update}(\mathbf{x}_{k-1}, \mathbf{c}_2, \mathbf{R}_k)$
$\delta \leftarrow K_p \cdot e_{\text{lateral}} + K_d \cdot \text{slope}$
return $\mathbf{x}_k$, $\delta$
```

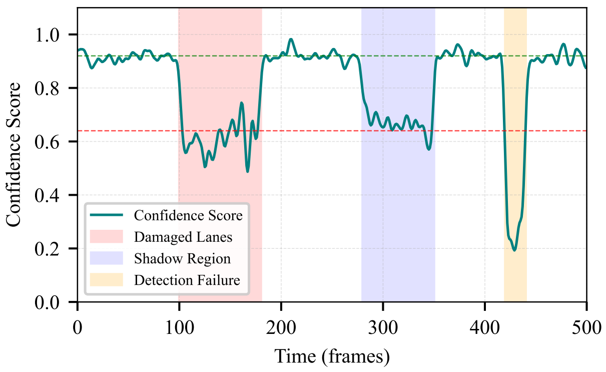


Fig. 3. Confidence score variation across different scenarios over a test run. Shaded regions indicate challenging conditions: damaged lane sections show reduced confidence (0.5–0.7), shadow regions cause fluctuations (0.45–0.88), and the failure region represents complete detection loss requiring filter prediction.

IV. Experimental Results

A. Experimental Setup

The system was validated through field tests on a 2.3 km campus road circuit at Keimyung University using a compact testbed vehicle. The test environment included challenging conditions such as tree shadows, worn lane markings with visible damage patterns, and curved sections with illumination changes that ranged from deep shadows to direct sunlight. Tests were conducted over multiple days at a nominal speed of 6 km/h.

B. Performance Metrics

System performance was evaluated using lateral error (cross-track error from lane center in centimeters), processing latency (time from image capture to control output), detection rate (percentage of frames with successful lane detection), and RMS error (root mean square of lateral deviations).

C. Scenario Analysis

Under clear lane conditions with well-marked lanes, the system achieved an average confidence of 0.92 ± 0.05, a lateral error of 0.01 cm RMS (0.04 cm maximum), and a detection rate of 99.2%. The high confidence values kept $\mathbf{R}_k$

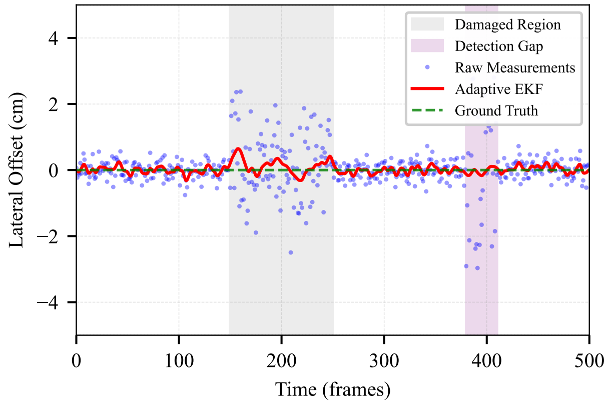


Fig. 4. Comparison of raw measurements versus filtered estimates during damaged lane conditions. The adaptive EKF successfully bridges measurement gaps while the mask enhancer reconstructs discontinuous regions.

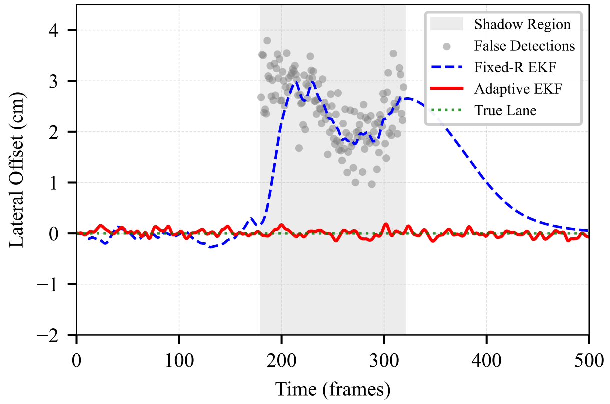


Fig. 5. Shadow rejection performance: false detections (gray), fixed-R EKF following false edges (blue dashed), and adaptive EKF maintaining correct lane position (red solid).

small, allowing the EKF to closely track the measurements while maintaining smooth trajectories.

In areas with damaged lane markings showing patchy or worn marks, the average confidence dropped to 0.64 ± 0.18 with a lateral error of 0.64 cm RMS (1.20 cm maximum). The detection rate was 89.3%. The system correctly identified poor measurement quality, inflating $\mathbf{R}_k$ by factors of 3–5. Combined with the mask enhancement module, the system successfully bridged gaps of up to 3 meters in the lane markings.

For dynamic shadow conditions with tree shadows creating false edges, confidence ranged between 0.45 and 0.88 with a false positive reduction of 78% compared to fixed-R EKF. Lateral error was 0.52 cm RMS. The confidence-based adaptation prevented the filter from following false edges.

Performance on curved roads with different radii was evaluated on the testbed track. The track curves with radii R=5m, R=10m and R=15m correspond to equivalent full-scale vehicle curves of R=50m, R=100m and R=150m, respectively. Lateral errors were 1.02 cm RMS for R=5m (equivalent R=50m) curves, 0.78 cm RMS for R=10m (equivalent R=100m), and 0.61 cm RMS for R=15m (equivalent R=150m). Curvature estimation error remained below 6% across all conditions.

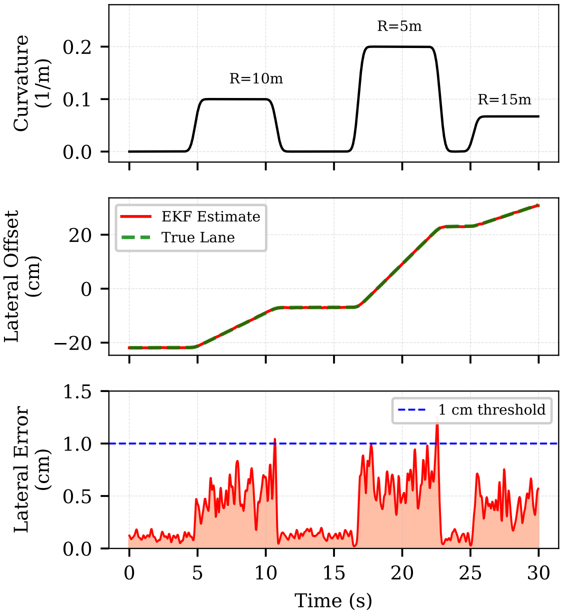


Fig. 6. Curved road tracking performance on the testbed: (top) road curvature profile with equivalent full-scale radii, (middle) lane position tracking showing EKF estimate following true lane, (bottom) lateral error remaining below 1 cm threshold for most conditions.

D. Computational Performance

TABLE I
PROCESSING TIME BREAKDOWN (MS)

Component	Mean	Std Dev
SegFormer Inference (TensorRT)	12.8	0.9
Graph Model Processing	4.2	0.6
Mask Enhancement	2.1	0.3
Confidence Calculation	1.4	0.2
EKF Update	2.3	0.3
Control Generation	0.8	0.1
Total	**23.6**	**1.8**

The system achieves 40+ Hz processing rate on Jetson AGX Orin, exceeding the camera frame rate of 30Hz. SegFormer inference accounts for 54% of the total processing time, representing the primary computational bottleneck.

E. Comparison with Baseline Methods

TABLE II
PERFORMANCE COMPARISON ON DAMAGED LANE SECTIONS

Method	RMS Error (cm)	Detection Rate (%)	FPS
Fixed-R EKF	2.84 ± 1.2	78.4	45
Sliding Window Only	3.21 ± 1.8	71.3	52
SegFormer Only	1.12 ± 0.6	85.6	35
Proposed	**0.64 ± 0.3**	**89.3**	**40**

The proposed method achieves a 77% reduction in the RMS error compared to fixed-R EKF while maintaining real-time performance suitable for embedded deployment. The combination of SegFormer segmentation, mask enhancement, and adaptive filtering provides a significant improvement over using any single component alone.

V. Discussion

The proposed system demonstrates several advantages for practical deployment in non-standard road conditions. The hybrid architecture combining deep learning segmentation with classical filtering achieves 40 Hz on embedded hardware while maintaining sub-centimeter accuracy. The explicit confidence scores enable operators to understand system behavior and identify potential failure modes.

The mask enhancement module proved particularly effective for damaged lane scenarios, successfully reconstructing gaps up to 3 meters by leveraging polynomial geometry from neighboring valid segments. This capability addresses a key limitation of pure segmentation approaches, which produce a discontinuous output when physical lane markings are interrupted.

Current limitations include sensitivity to extreme weather conditions such as heavy rain or snow, where the SegFormer model was not specifically trained. The confidence heuristic based on pixel density and fit error may not capture all failure modes, particularly systematic biases from camera calibration drift. The system assumes single-lane tracking and does not currently support lane change maneuvers or multi-lane scenarios.

Future work will address these limitations through IMU integration for improved prediction during detection failures, weather-augmented training for rain and snow robustness, and learning-based confidence estimation using Bayesian neural networks [19] for principled uncertainty quantification.

VI. Conclusions

This paper presented a robust lane tracking system designed for non-standard road conditions including damaged markings and dynamic shadows. The key innovation is a hybrid architecture that combines SegFormer semantic segmentation, graph-based connectivity analysis, mask enhancement for gap reconstruction, and confidence-adaptive EKF filtering. By quantifying measurement reliability through pixel density and fit quality metrics, the system intelligently transitions between measurement-based tracking and model-based prediction.

The system was validated on an autonomous vehicle platform with a Jetson AGX Orin embedded processor. Field tests on a 2.3 km campus circuit demonstrated significant improvements over baseline methods: RMS lateral error of 0.64 cm on damaged lane sections (77% reduction compared to fixed-R EKF), detection rate of 89.3%, and real-time performance at 40 Hz, with the mask enhancement module successfully bridging lane gaps up to 3 meters.

These results demonstrate that combining modern deep learning perception with classical adaptive estimation provides a practical solution for autonomous driving and road maintenance applications in challenging real-world environments where lane quality varies significantly.

Acknowledgments

The authors thank Keimyung University Autonomous Mobility System Lab for providing the experimental platform and testing facilities. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-25435307) as part of the Research Subsidy for Master's Students.

References

[1] A. Bar-Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.

[2] N. J. Zakaria, M. I. Shapiai, R. A. Ghani, M. N. M. Yassin, M. Z. Ibrahim, and N. Wahid, "Lane detection in autonomous vehicles: A systematic review," *IEEE Access*, vol. 11, pp. 3729–3765, 2023.

[3] National Center for Statistics and Analysis, "Overview of motor vehicle traffic crashes in 2021," National Highway Traffic Safety Administration (NHTSA), Washington, D.C., Tech. Rep. DOT HS 813 435, 2023.

[4] N. Formosa, M. Quddus, C. K. Man, M. K. Singh, C. Morton, and C. B. Masera, "Evaluating the impact of lane marking quality on the operation of autonomous vehicles," *Journal of Transportation Engineering, Part A: Systems*, vol. 150, no. 1, p. 04023126, 2024.

[5] V. Maddiralla and S. Subramanian, "Effective lane detection on complex roads with convolutional attention mechanism in autonomous vehicles," *Scientific Reports*, vol. 14, no. 1, p. 19193, 2024.

[6] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[7] M. Aly, "Real time detection of lane markers in urban streets," in *2008 IEEE Intelligent Vehicles Symposium*. IEEE, 2008, pp. 7–12.

[8] B.-E. Dorj, S. Hossain, and D.-J. Lee, "Highly curved lane detection algorithms based on kalman filter," *Applied Sciences*, vol. 10, no. 7, p. 2372, 2020.

[9] H. Guo, D. Cao, H. Chen, C. Lv, H. Wang, and S. Yang, "Vehicle dynamic state estimation: State of the art schemes and perspectives," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 418–431, 2018.

[10] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and kalman filter," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 3261–3264.

[11] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018, pp. 7276–7283.

[12] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 286–291.

[13] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 12 077–12 090.

[14] TuSimple, "Tusimple lane detection benchmark," https://github.com/TuSimple/tusimple-benchmark, 2017, accessed: 2025-01-12.

[15] J. Yoo and D. Kim, "Graph model-based lane-marking feature extraction for lane detection," *Sensors*, vol. 21, no. 13, p. 4428, 2021.

[16] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep. TR 95-041, 1995.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[18] R. Marino, S. Scalzi, and M. Netto, "Nested pid steering control for lane keeping in autonomous vehicles," *Control Engineering Practice*, vol. 19, no. 12, pp. 1459–1467, 2011.

[19] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, vol. 48. PMLR, 2016, pp. 1050–1059.