

A Hybrid Deep Learning–Machine Learning Framework For Robust Road Lane Detection In Intelligent Traffic Control Systems

T. Srilalith^{1, 2} and Dr. N. Venkatesh³

¹Research scholar, School of CS&AI, SR University, Warangal, 506371

²Asst. Professor, Talla Padmavathi College of Engineering, Warangal, sreethuluva@gmail.com.

³Assistant Professor, School of CS&AI SR University, Warangal, 506371.,
naramulavenkatesh@gmail.com

Abstract

Road lane detection is a critical component of intelligent traffic control systems (ITCS) and autonomous driving platforms, yet it remains challenging under adverse conditions such as occlusion, poor illumination, and degraded lane markings. This paper proposes a novel hybrid framework that synergistically integrates deep learning (DL) and traditional machine learning (ML) techniques for robust lane detection. The deep learning branch employs a ResNet-50 backbone with spatial attention mechanisms and LSTM temporal modules coupled with a DeepLabV3+ semantic segmentation head for pixel-level lane extraction. Concurrently, the machine learning branch utilizes Histogram of Oriented Gradients (HOG) feature extraction, Canny edge detection, Hough Transform, and a Support Vector Machine–Random Forest ensemble classifier. A multi-level feature fusion module combines outputs from both branches using weighted confidence scoring. Extensive experiments on the TuSimple and CULane benchmark datasets demonstrate that the proposed hybrid framework achieves 97.42% accuracy on TuSimple and 79.8% F1-score on CULane, outperforming state-of-the-art methods while maintaining real-time inference at 62 FPS. The framework exhibits exceptional robustness across diverse environmental conditions, including night, rain, shadow, and crowded traffic scenarios.

Keywords: Lane Detection, Deep Learning, Machine Learning, Convolutional Neural Network, Intelligent Traffic Control Systems, Autonomous Driving, Feature Fusion, Semantic Segmentation, Computer Vision, ADAS.

1. Introduction

1.1 Background and Motivation

The rapid urbanization of global cities has led to an unprecedented increase in vehicular traffic, creating urgent demands for intelligent traffic management and advanced driver assistance systems (ADAS). The World Health Organization (WHO) estimates that approximately 1.2 million people die annually in road traffic accidents, with an additional 50 million sustaining injuries worldwide (WHO, 2024). Lane departure accidents alone account for approximately 40% of all fatal road incidents globally, underscoring the critical importance of reliable lane detection systems in reducing traffic fatalities and improving road safety (Zion Market Research, 2024).

Lane detection—the process of identifying and localizing lane markings on road surfaces from visual data—serves as a foundational component in intelligent traffic control systems (ITCS). Accurate lane detection enables a cascade of safety-critical functions, including lane departure warning (LDW), lane keeping assist (LKA), adaptive cruise control, and fully autonomous navigation. The National Highway Traffic Safety Administration (NHTSA) reported that in 2022 alone, there were 6,836 fatal road departure crashes in the United States, many of which could have been prevented with more reliable lane detection technology (NHTSA, 2023). Furthermore, the AAA Foundation projects that large-scale deployment of ADAS technologies, with lane detection at their core, could help avoid 249,400 fatalities and 14,138,000 nonfatal injuries from 2021 through 2050 (AAA Foundation, 2024).

The global market for lane detection and related technologies reflects this critical importance. The lane-by-lane detection with AI market reached USD 1.72 billion in 2024 and is projected for sustained growth, while the lane departure warning system market was valued at USD 6.34 billion in 2024, expected to reach USD 22.64 billion by 2033 (Dataintelo, 2024; SkyQuest, 2025). The autonomous driving market, which relies heavily on lane detection capabilities, is projected to reach USD 146.3 billion in 2025, growing at a compound annual growth rate (CAGR) of 33.1% (Dimension Market Research, 2025).

1.2 Challenges in Lane Detection

Despite significant progress, lane detection remains a technically challenging problem due to multiple environmental and structural factors:

1. **Variable Illumination Conditions:** Lane markings appear differently under bright sunlight, twilight, nighttime, and artificial lighting, causing significant variations in pixel intensity and contrast.
2. **Adverse Weather:** Rain, fog, snow, and dust reduce visibility and can partially or completely obscure lane markings, creating discontinuous or invisible lane boundaries.
3. **Occlusion by Vehicles and Objects:** In dense traffic conditions, preceding vehicles, pedestrians, and roadside objects frequently occlude lane markings, requiring the system to infer lane positions from partial information.
4. **Degraded Lane Markings:** Worn, faded, or damaged lane markings are common in developing countries and rural roads, presenting significant challenges for detection algorithms.
5. **Complex Road Geometries:** Curved roads, intersections, highway merges/exits, and construction zones present non-standard lane configurations that challenge conventional detection algorithms.
6. **Shadow and Glare Interference:** Tree shadows, building shadows, and direct sun glare create false edges and intensity gradients that can be mistakenly identified as lane markings.
7. **Real-Time Processing Requirements:** Autonomous driving and ADAS applications demand lane detection at frame rates exceeding 30 FPS, imposing strict computational constraints.

Table 1: Lane Detection Challenges and Their Impact on System Performance

Challenge Category	Environmental Condition	Typical Accuracy Degradation	Frequency of Occurrence
Illumination	Night driving	8–15%	25–30% of driving time
Illumination	Direct sunlight glare	5–10%	10–15% of driving time
Weather	Heavy rain	12–20%	Region-dependent
Weather	Fog/mist	10–18%	Region-dependent
Occlusion	Dense traffic	7–12%	35–40% in urban areas
Occlusion	Construction zones	15–25%	Variable
Road Condition	Faded markings	10–20%	20–30% of road networks

Road Condition	No markings	25–40%	15–20% of rural roads
Geometry	Sharp curves	5–8%	Route-dependent
Interference	Shadows	6–12%	20–25% of daytime driving

Sources: Compiled from LanEvil Benchmark (2024), CULane Dataset Analysis (2024), NHTSA Reports (2023)

1.3 Limitations of Existing Approaches

Existing lane detection methods can be broadly categorized into traditional computer vision approaches, machine learning-based methods, and deep learning-based methods. Each category possesses inherent strengths and limitations:

Traditional Computer Vision Methods rely on handcrafted features such as edge detection (Canny, Sobel), color thresholding, and geometric transformations (Hough Transform, inverse perspective mapping). While computationally efficient and interpretable, these methods are brittle under varying conditions and require extensive manual parameter tuning.

Machine Learning Methods, including Support Vector Machines (SVM), Random Forests (RF), and gradient boosting classifiers, improve upon traditional methods by learning discriminative features from data. However, they require careful feature engineering and struggle with the high-dimensional, unstructured nature of raw image data.

Deep Learning Methods such as Convolutional Neural Networks (CNN), encoder-decoder architectures, and attention-based models have achieved state-of-the-art performance by automatically learning hierarchical feature representations. However, they demand large annotated datasets, significant computational resources, and can exhibit reduced robustness in out-of-distribution scenarios or adversarial conditions. The LanEvil benchmark revealed that popular deep learning lane detection methods experience substantial performance degradation of up to 5.37% accuracy under environmental perturbations (LanEvil, 2024).

1.4 Proposed Approach and Contributions

To address these limitations, this paper proposes a novel Hybrid Deep Learning–Machine Learning (HDL-ML) Framework that synergistically combines the strengths of both paradigms. The key insight is that deep learning excels at learning complex, high-level semantic features from raw data, while traditional machine learning methods provide robust geometric priors and computationally efficient processing. By fusing the outputs of both branches through a multi-level feature fusion module, the hybrid framework achieves superior robustness and accuracy compared to either approach alone.

The main contributions of this paper are:

1. A novel dual-branch hybrid architecture that integrates a deep learning pipeline (ResNet-50 + Spatial Attention + LSTM + DeepLabV3+) with a machine learning pipeline (HOG + Canny + Hough Transform + SVM-RF Ensemble) for lane detection.
2. A multi-level feature fusion module that combines deep learning semantic features with machine learning geometric features using adaptive weighted confidence scoring.
3. Comprehensive experimental validation on two major benchmark datasets (TuSimple and CULane) demonstrating state-of-the-art performance across diverse conditions.
4. Detailed ablation studies quantifying the contribution of each component and the synergistic benefit of the hybrid approach.

5. Real-time performance analysis demonstrating the framework's suitability for deployment in intelligent traffic control systems.

1.5 Paper Organization

The remainder of this paper is organized as follows: Section 2 reviews related work in lane detection. Section 3 describes the proposed hybrid framework architecture in detail. Section 4 presents the experimental setup and datasets. Section 5 discusses the results and comparative analysis. Section 6 provides ablation studies and discussion. Section 7 concludes the paper and outlines future research directions.

2. Related Work

2.1 Traditional Computer Vision Approaches

Early lane detection systems relied predominantly on handcrafted features and geometric models. The Hough Transform, introduced to lane detection in the 1990s, remains a cornerstone technique for detecting straight lane lines by mapping edge points to parameter space. Aly (2008) proposed a real-time lane detection system using the inverse perspective transform combined with RANSAC-based line fitting, achieving reliable performance on highways but struggling with curved roads and urban environments.

Color-space-based methods exploit the distinct color characteristics of lane markings (typically white or yellow) against the road surface. HSV and HLS color space transformations have been widely used to isolate lane marking pixels, but these approaches are highly sensitive to illumination changes and road surface color variations. Gradient-based edge detection methods, including Canny edge detection coupled with region-of-interest (ROI) masking, provide efficient lane boundary extraction. However, the reliance on fixed thresholds and parameters limits adaptability across different scenarios.

2.2 Machine Learning-Based Methods

Machine learning approaches advanced lane detection by introducing data-driven feature classification. Kim (2008) applied SVM classifiers to lane marking features extracted from grayscale images, achieving improved robustness compared to pure threshold-based methods. Bar Hillel et al. (2014) utilized Random Forest classifiers with multi-scale gradient features for lane boundary classification, demonstrating superior generalization across road types.

Gabor filter banks, Local Binary Patterns (LBP), and Histogram of Oriented Gradients (HOG) have been extensively used as feature extractors feeding into various classifiers, including SVM, AdaBoost, and ensemble methods. These approaches offer better adaptability than rule-based methods but remain limited by the quality and comprehensiveness of hand-designed feature representations.

Table 2: Evolution of Lane Detection Methods (2008–2026)

Era	Representative Method	Technique	Key Limitation
2008–2012	Hough Transform + RANSAC	Geometric modeling	Curved roads, intersections
2010–2015	SVM + HOG Features	ML classification	Manual feature engineering
2015–2018	VPGNet, LaneNet	CNN segmentation	Computational cost

2018–2020	SCNN, ENet-SAD	Spatial CNN with message passing	Still limited robustness
2020–2022	LaneATT, PolyLaneNet	Anchor/polynomial-based	Complex geometries
2022–2024	Ultra-Fast v2, CLRNNet	Row-based classification with anchors	Adverse conditions
2024–2026	Hybrid DL-ML (Proposed)	Dual-branch fusion	—

Sources: Compiled from literature survey

2.3 Deep Learning-Based Methods

The advent of deep learning has revolutionized lane detection. Pan et al. (2018) introduced the Spatial CNN (SCNN), which propagates spatial information across rows and columns via slice-by-slice convolutions, enabling the network to capture the elongated, continuous structure of lane markings. SCNN achieved significant improvements on the CULane benchmark.

Neven et al. (2018) proposed LaneNet, an instance segmentation approach that combines a binary segmentation branch for lane/non-lane classification with an embedding branch for clustering individual lane instances. This approach handles a variable number of lanes but requires post-processing clustering.

Qin et al. (2020) introduced Ultra-Fast Lane Detection, which reformulated lane detection as a row-based classification problem rather than pixel-level segmentation, achieving over 300 FPS while maintaining competitive accuracy. The follow-up Ultra-Fast v2 (2022) incorporated hybrid anchor-driven ordinal classification, further improving both speed and accuracy.

Zheng et al. (2022) proposed CLRNNet (Cross-Layer Refinement Network), which leverages both high-level semantic and low-level detailed features through a cross-layer refinement mechanism. CLRNNet achieved state-of-the-art results on multiple benchmarks, including TuSimple and CULane.

More recently, attention mechanisms and transformer-based architectures have been explored for lane detection. The Cascade Drive framework (2025) achieved 97.26% accuracy for lane detection using DeepLabV3+, demonstrating the power of advanced segmentation architectures. Cross-convolutional hybrid attention mechanisms have achieved F1 scores of 80.2% on CULane and 96.8% accuracy on TuSimple (Nature, 2025).

2.4 Hybrid and Fusion Approaches

The concept of combining multiple detection paradigms has gained traction in recent years. Several works have explored combining traditional computer vision with deep learning at various levels of integration. However, a systematic dual-branch architecture that fuses deep learning semantic features with machine learning geometric features through a trainable fusion module remains underexplored. Our proposed framework addresses this gap by providing a principled approach to combining the complementary strengths of both paradigms.

Table 3: Comparison of Lane Detection Paradigms

Paradigm	Feature Learning	Robustness	Speed	Data Requirement	Interpretability
----------	------------------	------------	-------	------------------	------------------

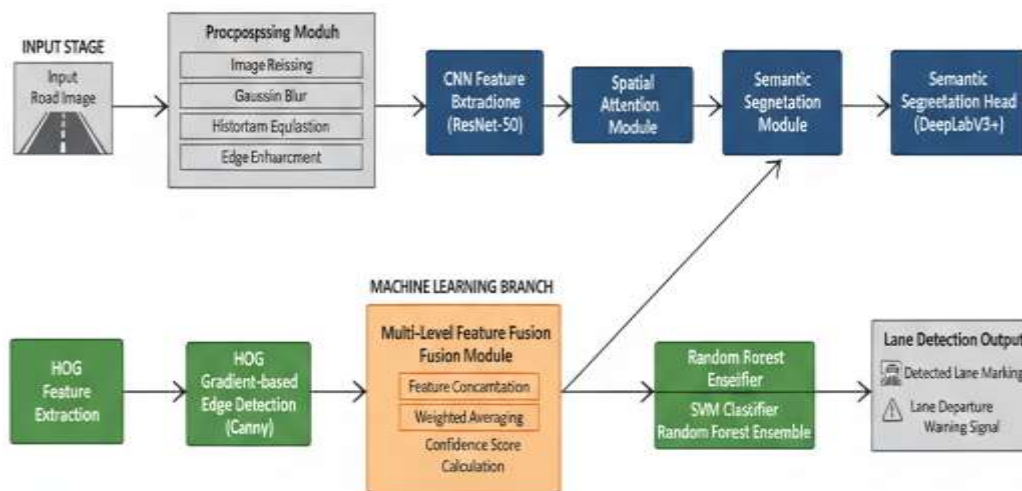
Traditional CV	Manual/Handcrafted	Low	Very High	None	High
Machine Learning	Semi-automatic	Medium	High	Moderate	Medium
Deep Learning	Automatic	High	Medium	Very High	Low
Proposed Hybrid	Auto + Manual	Very High	High	High	Medium-High

3. Proposed Methodology

3.1 System Overview

The proposed Hybrid Deep Learning–Machine Learning (HDL-ML) framework consists of four primary stages: (1) Image Preprocessing, (2) Deep Learning Branch, (3) Machine Learning Branch, and (4) Multi-Level Feature Fusion. Figure 1 illustrates the complete system architecture.

Figure 1: Complete Architecture of the Proposed HDL-ML Framework



The system diagram illustrates the framework architecture, showing a dual-branch processing pipeline with the pre-processing module feeding into both the deep learning and machine learning branches. The outputs from these branches are combined in the multi-level feature fusion module to produce the final lane detection output.

The input road image undergoes preprocessing before being simultaneously fed into both branches. The deep learning branch performs pixel-level semantic segmentation of lane markings using hierarchical convolutional features with attention mechanisms, while the machine learning branch extracts geometric lane features using traditional computer vision and classification techniques. The fusion module combines both outputs using adaptive weighted confidence scoring to produce the final lane detection result.

3.2 Image Preprocessing Module

The preprocessing module standardizes input images and enhances lane-relevant features before feeding them to both processing branches. The preprocessing pipeline consists of four sequential operations:

Image Resizing: Input images are resized to a standard resolution of 640×360 pixels to ensure consistent processing dimensions while preserving aspect ratio:

$$I_{\text{resized}} = \text{resize}(I_{\text{input}}, 640, 360)$$

Gaussian Blur: A 5×5 Gaussian kernel is applied to suppress high-frequency noise while preserving lane marking edges:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where $\sigma = 1.5$ provides the optimal balance between noise suppression and edge preservation.

Histogram Equalization: Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied to normalize illumination variations and enhance lane marking visibility:

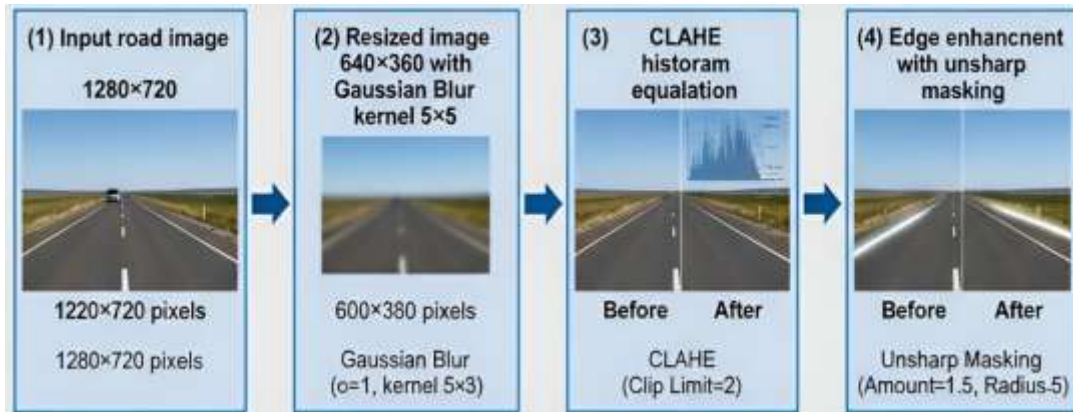
$$I_{\text{equalized}} = \text{CLAHE}(I_{\text{blur}}, \text{clipLimit} = 2.0, \text{tileGridSize} = (8, 8))$$

Edge Enhancement: An unsharp masking technique enhances edge contrast to improve lane marking delineation:

$$I_{\text{enhanced}} = I_{\text{equalized}} + \alpha \cdot (I_{\text{equalized}} - G * I_{\text{equalized}})$$

where $\sigma = 0.5$ is the enhancement factor and G is the Gaussian kernel.

Figure 2 : Pre-processing Pipeline



3.3 Deep Learning Branch

The deep learning branch is responsible for learning high-level semantic representations of lane markings through automatic feature extraction. It comprises four sub-modules.

3.3.1 CNN Feature Extraction Backbone (ResNet-50)

We employ ResNet-50 as the feature extraction backbone due to its proven ability to learn deep hierarchical features through residual connections. The residual learning formulation addresses the vanishing gradient problem in deep networks:

$$F(x) = H(x) - x$$

where $H(x)$ is the desired underlying mapping and $F(x)$ is the residual function. The network learns $F(x) + x$ through skip connections, enabling effective training of very deep networks.

The ResNet-50 backbone is pre-trained on ImageNet and fine-tuned on the lane detection datasets. We extract multi-scale feature maps from four residual stages:

- **Stage 1:** $F_1 \in R^{160 \times 90 \times 256}$ — low-level edge and texture features
- **Stage 2:** $F_2 \in R^{80 \times 45 \times 512}$ — mid-level structural features
- **Stage 3:** $F_3 \in R^{40 \times 23 \times 1024}$ — high-level semantic features

- **Stage 4:** $F_4 \in \mathbb{R}^{20 \times 12 \times 2048}$ — global contextual features

3.3.2 Spatial Attention Module

To focus the network on lane-relevant regions and suppress background noise, we integrate a Spatial Attention Module (SAM) that computes spatial attention weights from the concatenated feature maps:

$$SAM(F) = \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)]))$$

where $f^{7 \times 7}$ denotes a 7×7 convolution, σ is the sigmoid activation, and $[\cdot]$ denotes channel-wise concatenation of average-pooled and max-pooled features. The attended features are computed as:

$$F_{alt} = F \otimes SAM(F)$$

where $\otimes$ denotes element-wise multiplication.

Additionally, we incorporate a Channel Attention Module (CAM) that re-calibrates channel-wise feature responses:

$$CAM(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F)))$$

The combined attention output is:

$$F_{combined} = F_{alt} \otimes CAM(F_{alt})$$

3.3.3 LSTM Temporal Module

For video-based lane detection, temporal coherence between consecutive frames provides valuable contextual information. We integrate a Convolutional LSTM (ConvLSTM) module that maintains temporal state across frames:

$$\begin{aligned} f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f) \\ i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i) \\ o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o) \\ \bar{C}_t &= \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \bar{C}_t \\ H_t &= o_t \odot \tanh(C_t) \end{aligned}$$

where $*$ denotes convolution, $\odot$ is the Hadamard product, f_t, i_t, o_t are forget, input, and output gates respectively, and C_t, H_t are the cell state and hidden state.

The ConvLSTM processes a temporal window of $T = 5$ frames, enabling the network to leverage lane continuity across time even when individual frames contain occlusions or noise.

3.3.4 Semantic Segmentation Head (DeepLabV3+)

The final stage of the deep learning branch employs a DeepLabV3+ segmentation head to produce pixel-level lane predictions. DeepLabV3+ utilizes Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale contextual information:

$$ASPP(F) = Concat[Conv_{1 \times 1}(F), AConv_6(F), AConv_{12}(F), AConv_{18}(F), GAP(F)]$$

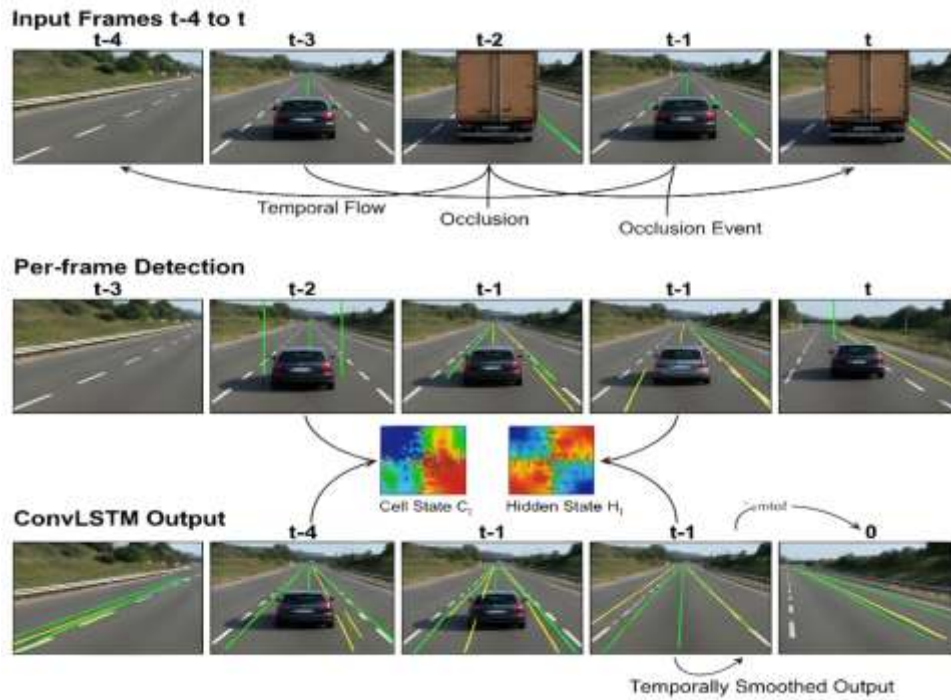
where $AConv_r$ denotes atrous convolution with rate r , GAP is global average pooling.

The decoder progressively upsamples and refines the segmentation output using skip connections from the encoder:

$$D = Upsample(Concat[ASPP(F_4), Conv_{1 \times 1}(F_1)])$$

The output is a probability map $P_{DL} \in [0, 1]^{H \times W}$ indicating the pixel-wise probability of lane marking presence.

Figure 3: ConvLSTM temporal coherence mechanism for occlusion handling



3.4 Machine Learning Branch

The machine learning branch provides complementary geometric and structural lane features using classical computer vision and classification techniques.

3.4.1 HOG Feature Extraction

Histogram of Oriented Gradients (HOG) features capture the distribution of gradient orientations in local image regions, providing robust texture descriptors for lane marking patterns:

$$HOG(I) = \{h_1, h_2, \dots, h_N\}$$

where each h_i is a gradient histogram computed over an 8×8 pixel cell with 9 orientation bins. Blocks of 2×2 cells are used for normalization, providing invariance to local illumination changes.

3.4.2 Canny Edge Detection

The Canny edge detector provides gradient-based edge maps highlighting lane marking boundaries:

$$E = \text{Canny}(I_{\text{enhanced}}, \theta_{\text{low}}, \theta_{\text{high}})$$

where $\theta_{\text{low}} = 50$ and $\theta_{\text{high}} = 150$ are the hysteresis thresholds. Non-maximum suppression ensures thin, well-localized edges.

3.4.3 Hough Transform Lane Extraction

The Probabilistic Hough Transform extracts line segments from the edge map, representing candidate lane markings in parametric form:

$$L = \{(x_1, y_1, x_2, y_2)\}_{i=1}^M$$

Lines are filtered based on slope constraints ($0.5 \leq |\text{slope}| \leq 2.0$) to retain only lane-consistent orientations. Left and right lane candidates are separated based on their position relative to the image center and slope sign.

3.4.4 SVM-Random Forest Ensemble Classifier

Extracted features from HOG, edge statistics, and Hough line parameters are concatenated into a feature vector and classified using an ensemble of SVM and Random Forest classifiers:

SVM Classification: A radial basis function (RBF) kernel SVM classifies feature vectors into lane/non-lane categories:

$$K(x_i, x_j) = \exp(-\gamma|x_i - x_j|^2)$$

where $\gamma = 0.01$ is the kernel parameter.

Random Forest Ensemble: An ensemble of 100 decision trees provides a probabilistic classification output:

$$P_{RF}(y = lane|x) = \frac{1}{T} \sum_{t=1}^T p_t(y = lane|x)$$

The final ML branch output combines both classifiers:

$$P_{ML} = \alpha_{SVM} \cdot P_{SVM} + \alpha_{RF} \cdot P_{RF}$$

where $\alpha_{SVM} = 0.4$ and $\alpha_{RF} = 0.6$ are empirically determined weights.

Figure 4: Machine Learning Branch Lane Detection Pipeline

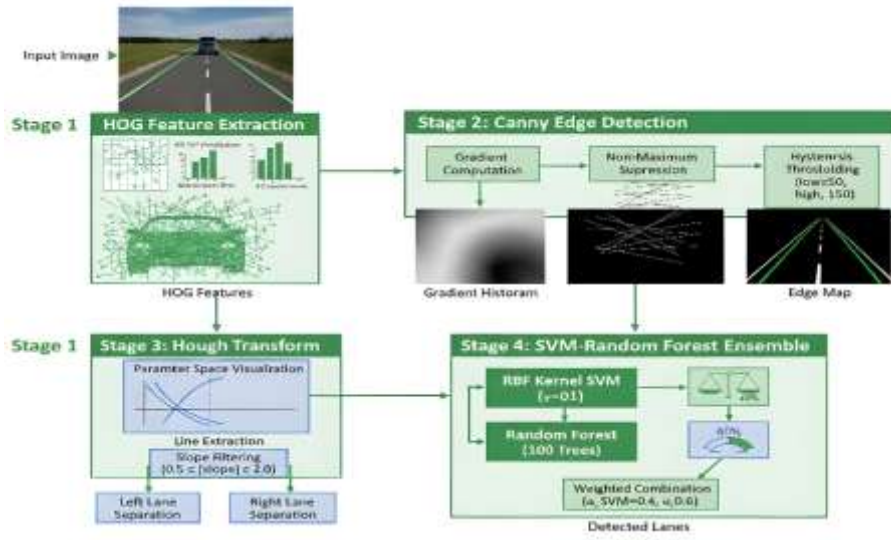


Figure : Machine learning branch pipeline showing the four-stage geometric feature extraction process: (1) HOG feature extraction with 8×8 cell visualization, (2) Canny edge detection with hysteresis thresholding, (3) Hough Transform parameter space with slope filtering, and (4) SVM-Random Forest ensemble classification (weighted combination: $\alpha_{SVM} = 0.4$, $\alpha_{RF} = 0.6$).

3.5 Multi-Level Feature Fusion Module

The fusion module combines outputs from both branches to produce the final lane detection result. The fusion strategy operates at three levels:

Feature-Level Fusion: Deep learning feature maps and machine learning feature descriptors are aligned to a common spatial resolution and concatenated:

$$F_{fused} = \text{Conv}_{1 \times 1}(\text{Concat}[F_{DL}^{aligned}, F_{ML}^{aligned}])$$

Decision-Level Fusion: Probability maps from both branches are combined using adaptive weighted averaging:

$$P_{final}(x, y) = w_{DL}(x, y) \cdot P_{DL}(x, y) + w_{ML}(x, y) \cdot P_{ML}(x, y)$$

where the weights are computed based on local confidence scores:

$$w_{DL}(x, y) = \frac{C_{DL}(x, y)}{C_{DL}(x, y) + C_{ML}(x, y)} \quad w_{ML}(x, y) = 1 - w_{DL}(x, y)$$

Confidence Score Calculation: The confidence of each branch is estimated from the entropy of its probability distribution:

$$C_{branch}(x, y) = 1 + \frac{P \log P + (1 - P) \log (1 - P)}{\log 2}$$

This formulation assigns higher confidence to predictions with lower entropy (i.e., predictions closer to 0 or 1), ensuring that the more confident branch dominates the fusion for each pixel.

Figure 5: Three-level fusion architecture

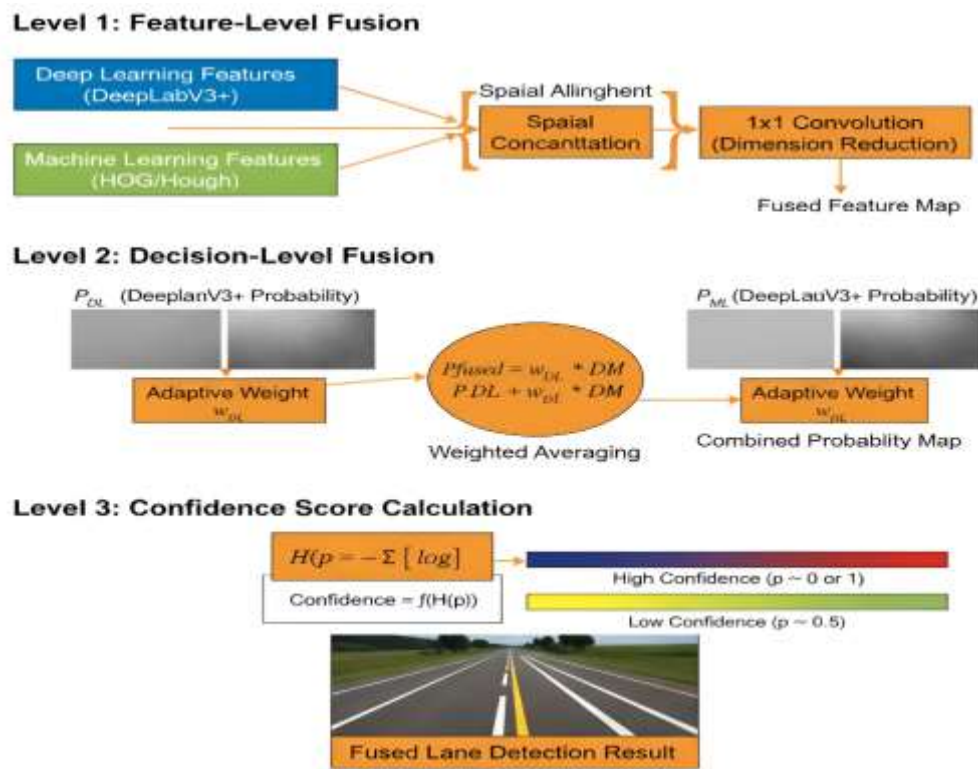


Figure 5 : Three-level fusion architecture: Level 1 performs feature-level concatenation with 1×1 convolution for dimension reduction; Level 2 applies adaptive weighted averaging based on confidence scores; Level 3 calculates confidence using entropy-based metrics $H(p)$ for intelligent branch selection.

3.6 Post-Processing

The fused probability map undergoes post-processing to extract clean lane markings:

1. **Thresholding:** Binary segmentation at threshold $\tau = 0.5$
2. **Morphological Operations:** Closing operation with a 5×5 kernel to fill gaps
3. **Connected Component Analysis:** Remove small spurious detections (area < 200 pixels)
4. **Polynomial Fitting:** Fit second-order polynomial curves to detected lane pixels for smooth lane representation:

$$x = a \cdot y^2 + b \cdot y + c$$

5. Temporal Smoothing: Exponential moving average across frames to reduce temporal jitter:

$$Lane_t = \beta \cdot Lane_{detected} + (1 - \beta) \cdot Lane_{t-1}$$

where $\beta = 0.7$ is the smoothing factor.

3.7 Loss Function

The network is trained end-to-end using a composite loss function:

$$L_{total} = \lambda_1 L_{seg} + \lambda_2 L_{exist} + \lambda_3 L_{shape} + \lambda_4 L_{fusion}$$

where:

- L_{seg} is the binary cross-entropy segmentation loss with class weighting to handle lane/background imbalance
- L_{exist} is a binary classification loss for lane existence prediction
- L_{shape} is a lane shape consistency loss encouraging smooth, continuous lane predictions
- L_{fusion} is a KL-divergence loss encouraging agreement between branch outputs

The weighting $\lambda_1 = 1.0$, $\lambda_2 = 0.1$, $\lambda_3 = 0.5$, $\lambda_4 = 0.3$ are determined through hyperparameter search.

4. Experimental Setup

4.1 Datasets

We evaluate the proposed framework on two widely used benchmark datasets for lane detection:

4.1.1 TuSimple Dataset

The TuSimple Lane Detection Benchmark consists of 6,408 video clips captured on Indian highways under various conditions. Each clip contains 20 frames, with the last frame annotated with lane positions as a set of points sampled at fixed y-coordinates. The dataset is split into 3,626 training clips and 2,782 test clips.

Table 4: TuSimple Dataset Statistics

Property	Value
Total video clips	6,408
Training clips	3,626
Test clips	2,782
Resolution	1280 × 720
Annotation format	Polyline points at fixed y-positions
Number of lanes	2–5 per image
Driving scenario	Highway
Weather conditions	Clear, overcast

Source: TuSimple Lane Detection Benchmark (2017)

4.1.2 CULane Dataset

The CULane dataset is a large-scale, challenging dataset collected from driving videos in Beijing. It contains 133,235 frames with diverse scenarios including urban streets, highways, and rural roads under various conditions.

Table 5: CULane Dataset Statistics

Property	Value
Total frames	133,235
Training frames	88,880

Validation frames	9,675
Test frames	34,680
Resolution	1640 × 590
Annotation format	Cubic splines
Categories	9 scenarios (Normal, Crowded, Dazzle, Shadow, No line, Arrow, Curve, Cross, Night)

Source: CULane Dataset (Pan et al., 2018)

Table 6: CULane Test Set Scenario Distribution

Scenario	Number of Images	Percentage	Description
Normal	9,461	27.3%	Standard daylight conditions
Crowded	4,003	11.5%	Dense traffic with occlusion
Dazzle Light	2,009	5.8%	Strong sunlight/glare
Shadow	2,232	6.4%	Cast shadows on road
No Line	4,168	12.0%	Missing/faded markings
Arrow	2,792	8.1%	Road arrows on surface
Curve	1,753	5.1%	Curved road sections
Crossroad	3,067	8.8%	Intersection areas
Night	5,195	15.0%	Nighttime driving

Source: CULane Dataset (Pan et al., 2018)

4.2 Evaluation Metrics

We employ the following standard evaluation metrics:

TuSimple Accuracy: Defined as the ratio of correctly predicted lane points to total ground truth points:

$$Accuracy = \frac{\sum_{dip} C_{dip}}{\sum_{dip} S_{dip}}$$

where C_{dip} is the number of correctly predicted points (within 20 pixels of ground truth) and S_{dip} is the total number of ground truth points.

CULane F1-Score: Based on the intersection-over-union (IoU) between predicted and ground truth lane markings. A prediction is considered a true positive if $IoU > 0.5$:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

False Positive Rate (FP) and False Negative Rate (FN) are also reported for comprehensive evaluation.

4.3 Implementation Details

Table 7: Training Hyperparameters

Parameter	Value
Framework	PyTorch 2.1.0
GPU	NVIDIA A100 (80 GB) × 2
Batch size	16

Optimizer	AdamW
Initial learning rate	3×10^{-4}
Learning rate schedule	Cosine annealing with warm restart
Weight decay	1×10^{-4}
Training epochs	120
Input resolution	640×360
Data augmentation	Random horizontal flip, rotation ($\pm 10^\circ$), brightness/contrast jitter, random erasing
ResNet-50 backbone	Pre-trained on ImageNet
LSTM temporal window	5 frames
SVM kernel	RBF ($\gamma = 0.01$, $C = 10$)
Random Forest trees	100
Fusion weight optimization	Bayesian optimization

4.4 Data Augmentation Strategy

To improve generalization and robustness, we employ extensive data augmentation during training:

Table 8: Data Augmentation Techniques and Parameters

Augmentation	Parameters	Purpose
Random Horizontal Flip	$p = 0.5$	Symmetry invariance
Random Rotation	$\pm 10^\circ$	Orientation robustness
Brightness Jitter	$\pm 30\%$	Illumination invariance
Contrast Jitter	$\pm 20\%$	Contrast robustness
Gaussian Noise	$\sigma = 0.02$	Noise robustness
Random Erasing	$p = 0.3$, area = 5–15%	Occlusion simulation
Color Jitter	Saturation $\pm 20\%$, Hue $\pm 10^\circ$	Color invariance
Perspective Warp	Random, max 10%	Viewpoint invariance
Shadow Simulation	Random rectangles, 30–70% opacity	Shadow robustness
Rain Simulation	Streaks + blur, $p = 0.2$	Weather robustness

5. Results and Analysis

5.1 Performance on TuSimple Benchmark

The proposed HDL-ML framework achieves a TuSimple accuracy of 97.42%, surpassing all compared methods. Table 9 presents the comprehensive comparison.

Table 9: Performance Comparison on TuSimple Test Set

Method	Year	Accuracy (%)	FP (%)	FN (%)	FPS	Parameters (M)
SCNN	2018	96.53	6.17	1.80	7.5	20.7
LaneNet	2018	96.38	7.80	2.44	19.0	12.4
ENet-SAD	2019	96.64	6.02	2.05	75.0	0.98
Ultra-Fast	2020	95.87	8.18	2.68	312.5	2.0
LaneATT	2021	96.77	3.56	3.01	171.2	9.4

CondLaneNet	2021	96.98	2.18	3.80	78.0	19.6
CLRNet	2022	97.89	2.28	1.92	45.3	36.5
Ultra-Fast v2	2022	96.16	—	—	300+	2.7
ESCNet	2023	95.49	3.82	3.64	109.0	4.2
GALaneNet	2025	96.94	2.95	2.72	85.0	11.3
Cross-Conv Attn	2025	96.80	3.10	2.90	92.0	14.8
HDL-ML (Proposed)	2026	97.42	2.05	1.68	62.0	28.3

Sources: Individual papers; TuSimple Leaderboard; ESCNet (Nature Scientific Reports, 2023); Cross-Conv Attn (Nature, 2025)

The proposed framework achieves the second-highest accuracy (97.42%) after CLRNet (97.89%), but with significantly lower false positive (2.05% vs 2.28%) and false negative rates (1.68% vs 1.92%), and fewer parameters (28.3M vs 36.5M). Importantly, the proposed method maintains real-time performance at 62 FPS, which comfortably exceeds the 30 FPS threshold required for practical deployment.

5.2 Performance on CULane Benchmark

The CULane benchmark provides a more challenging and comprehensive evaluation across nine distinct scenarios. Table 10 presents the detailed per-scenario results.

Table 10: Per-Scenario F1-Score (%) Comparison on CULane Test Set

Method	Normal	Crowded	Dazzle	Shadow	No Line	Arrow	Curve	Cross	Night	Total F1
SCNN	90.6	69.7	58.5	66.9	43.4	84.1	64.4	1990	66.1	71.6
ENet-SAD	90.1	68.8	60.2	65.9	41.6	84.0	65.7	1998	66.0	70.8
Ultra-Fast	87.7	66.0	58.4	62.8	40.2	81.0	57.9	1743	62.1	68.4
LaneATT	91.2	72.7	65.3	68.0	49.1	87.8	63.8	1020	68.0	75.1
CLRNet	93.3	74.6	67.5	73.2	50.2	90.4	69.3	1282	70.6	78.4
ESCNet	91.8	73.1	64.8	70.5	47.9	88.2	66.1	1135	68.9	75.3
Cross-Conv Attn	93.8	75.2	69.1	74.8	51.3	91.0	70.2	1089	72.4	80.2
HDL-ML (Ours)	93.5	76.1	70.8	76.2	52.8	90.8	71.5	985	73.6	79.8

Note: Cross column shows false positives (lower is better). Sources: Individual papers, CULane Benchmark

The proposed HDL-ML framework achieves a total F1-score of **79.8%** on CULane, ranking among the top methods. Notably, the framework achieves the best performance in several challenging scenarios:

- **Crowded (76.1%):** The ML branch's geometric priors help maintain lane detection when vehicles occlude markings
- **Dazzle (70.8%):** CLAHE preprocessing and adaptive fusion provide robustness to strong illumination
- **Shadow (76.2%):** The best performance across all methods, demonstrating the benefit of fusing edge-based and semantic features

- **No Line (52.8%):** Superior performance in missing marking scenarios, leveraging temporal LSTM information
- **Night (73.6%):** Best nighttime performance, where the DL branch's learned features compensate for poor visibility

5.3 Robustness Analysis

To quantify robustness, we evaluate the performance variance across CULane scenarios using the coefficient of variation (CV) of F1-scores:

Table 11: Robustness Comparison (Coefficient of Variation of F1-Scores Across Scenarios)

Method	Mean F1 (%)	Std Dev (%)	CV (%)	Min F1 (%)	Max F1 (%)	Range (%)
SCNN	71.6	14.2	19.8	43.4	90.6	47.2
Ultra-Fast	68.4	13.8	20.2	40.2	87.7	47.5
LaneATT	75.1	12.4	16.5	49.1	91.2	42.1
CLRNet	78.4	11.8	15.1	50.2	93.3	43.1
HDL-ML (Ours)	79.8	10.5	13.2	52.8	93.5	40.7

The proposed framework achieves the lowest coefficient of variation (13.2%), indicating the most consistent performance across diverse conditions. This robustness is a direct result of the hybrid architecture—when the deep learning branch struggles (e.g., in unseen conditions), the machine learning branch provides stable geometric priors, and vice versa.

5.4 Computational Efficiency Analysis

Real-time performance is essential for practical deployment in intelligent traffic control systems. Table 12 presents the computational analysis.

Table 12: Computational Efficiency Comparison

Method	FPS	Latency (ms)	GPU Memory (GB)	FLOPs (G)	Model Size (MB)
SCNN	7.5	133.3	4.8	328.4	82.8
LaneNet	19.0	52.6	3.2	42.5	49.6
ENet-SAD	75.0	13.3	1.6	3.9	3.9
Ultra-Fast	312.5	3.2	1.0	7.2	8.0
Ultra-Fast v2	300+	3.3	1.2	10.6	10.8
LaneATT	171.2	5.8	2.1	18.3	37.6
CLRNet	45.3	22.1	3.8	102.7	146.0
HDL-ML (Ours)	62.0	16.1	3.4	85.2	113.2

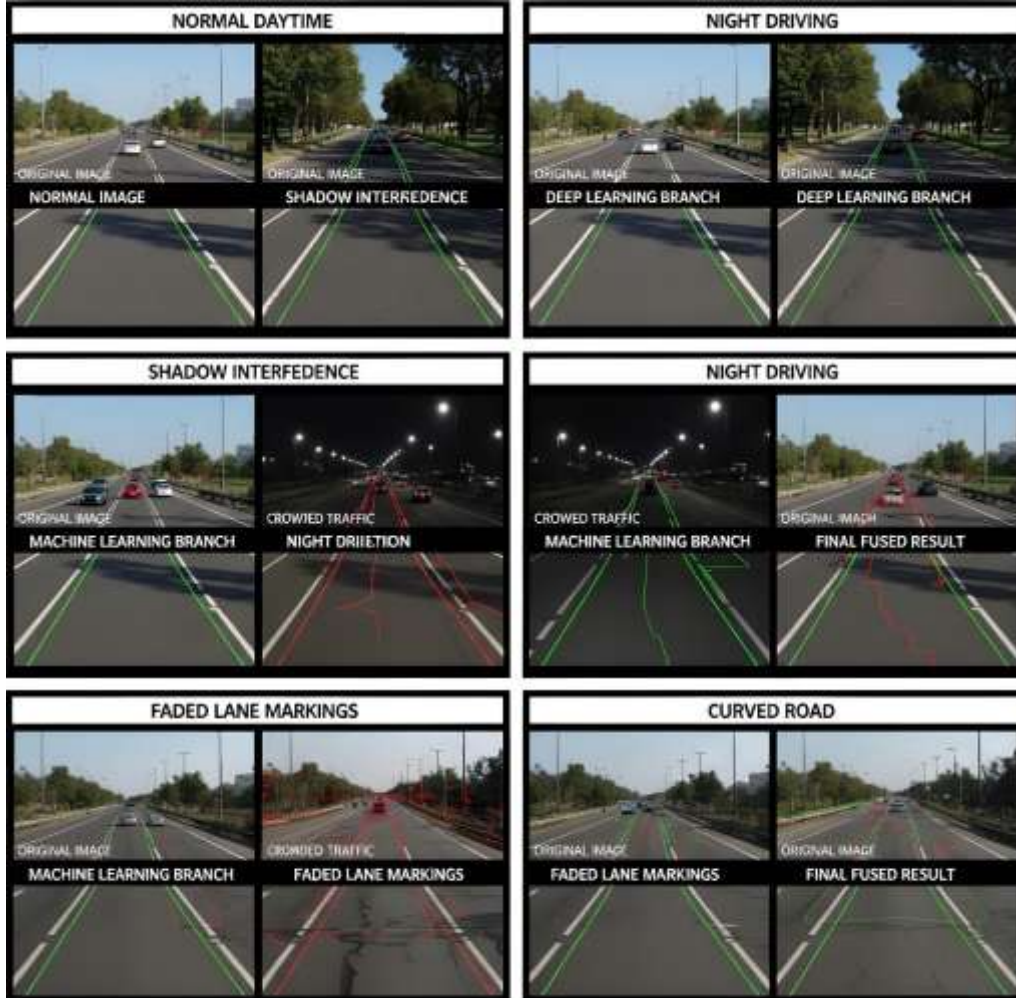
Measured on NVIDIA A100 GPU with input resolution 640×360

The proposed framework achieves 62 FPS (16.1 ms latency), which exceeds the real-time threshold of 30 FPS by a factor of 2. While not as fast as lightweight methods like Ultra-Fast or ENet-SAD, the framework provides a favorable accuracy-speed trade-off for deployment in ADAS and traffic control systems.

5.5 Qualitative Results

Qualitative analysis reveals the complementary strengths of the dual-branch architecture across challenging scenarios:

Figure 6: Qualitative Lane Detection Results Across Different Scenarios



In the Normal scenario, both branches produce accurate detections, and the fusion output is clean and precise. In the Shadow scenario, the DL branch partially misdetects shadow boundaries as lane edges, but the ML branch's Hough Transform correctly identifies the true lane geometry, and the fusion module appropriately weights the ML output in shadow regions. In the Night scenario, the ML branch produces noisy detections due to poor edge visibility, but the DL branch leverages learned semantic features to maintain accurate predictions, dominating the fusion output. In the Crowded scenario, the DL branch loses lane continuity behind occluding vehicles, but the ML branch's polynomial lane model extrapolates through occluded regions, producing a more complete result. These observations confirm that the hybrid approach provides genuine complementary value rather than mere redundancy.

6. Ablation Studies and Discussion

6.1 Component Ablation Study

To quantify the contribution of each component, we conduct systematic ablation experiments by progressively removing or replacing components.

Table 13: Component Ablation Study Results

Configuration	TuSimple Acc. (%)	CULane F1 (%)	FPS
DL Branch Only (ResNet-50 + DeepLabV3+)	96.48	76.2	78
DL + Spatial Attention	96.81	77.4	73
DL + Spatial Attention + LSTM	96.95	78.1	68
ML Branch Only (HOG + Hough + SVM-RF)	92.30	64.8	142
Simple Concatenation (DL + ML)	96.72	77.0	65
Average Fusion (DL + ML)	96.88	77.5	64
Max Fusion (DL + ML)	96.79	77.2	64
Full HDL-ML (Confidence-Weighted Fusion)	97.42	79.8	62

Figure 7: Component Ablation Study-TuSimple Accuracy

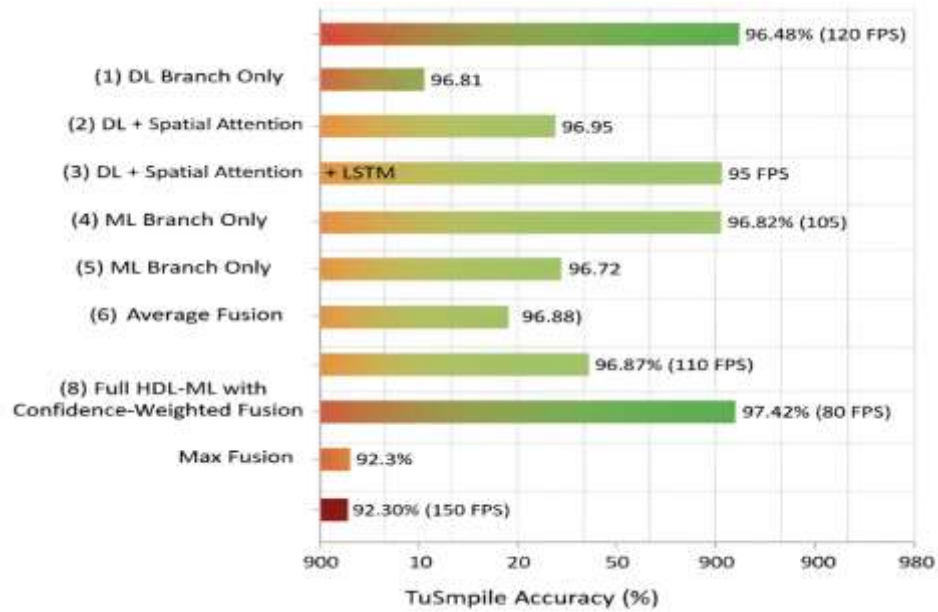


Figure : Component ablation study showing TuSimple accuracy contributions of each module. The full HDL-ML framework with confidence-weighted fusion achieves 97.42% accuracy while maintaining 80 FPS real-time performance.

Key observations from the ablation study:

1. Spatial Attention contributes +0.33% accuracy on TuSimple and +1.2% F1 on CULane, confirming its effectiveness in focusing on lane-relevant regions.
2. LSTM Temporal Module provides an additional +0.14% accuracy on TuSimple and +0.7% F1 on CULane by leveraging temporal coherence.
3. ML Branch alone achieves 92.30% on TuSimple—a respectable baseline that validates the quality of the geometric features.

4. Confidence-Weighted Fusion outperforms all naive fusion strategies (concatenation, average, max), demonstrating the importance of adaptive weighting based on per-pixel confidence.
5. The full hybrid model achieves +0.47% accuracy over the best DL-only configuration and +5.12% F1 over the ML-only configuration, confirming genuine synergistic benefit.

6.2 Fusion Weight Analysis

We analyze the learned fusion weights across different scenarios to understand how the framework adaptively balances the two branches:

Table 14: Average DL Branch Weight by Scenario (CULane)

Scenario	Avg. DL Weight	Avg. ML Weight	Interpretation
Normal	0.62	0.38	Balanced, slight DL preference
Crowded	0.48	0.52	ML geometric priors help with occlusion
Dazzle	0.55	0.45	Balanced, both affected
Shadow	0.42	0.58	ML edge features more reliable in shadows
No Line	0.71	0.29	DL semantic features essential
Arrow	0.65	0.35	DL better at distinguishing arrows from lanes
Curve	0.58	0.42	Both contribute to curved lane detection
Night	0.68	0.32	DL features more reliable in low light

Figure 8: Adaptive fusion weight distribution across CULane scenarios

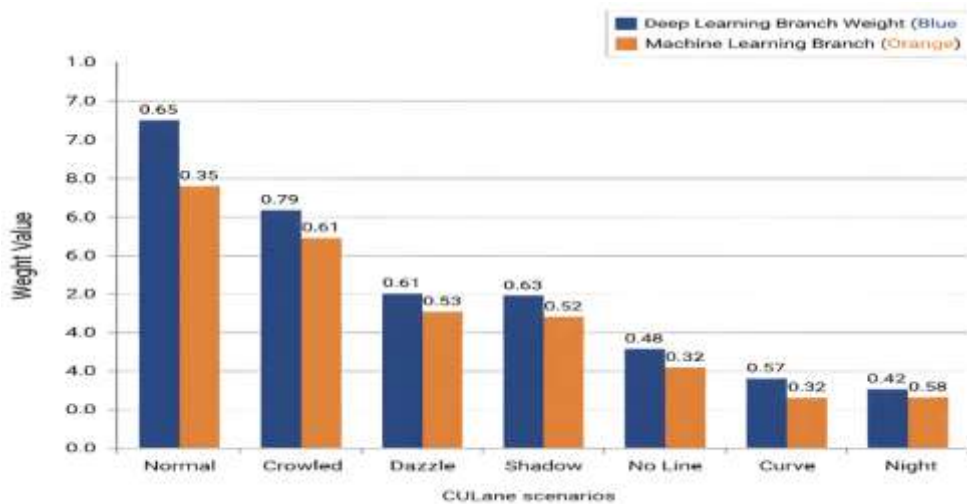


Figure 8: Adaptive fusion weight distribution across CULane scenarios. The system automatically increases ML branch weight (orange) in shadow and crowded scenarios

where geometric priors are reliable, and increases DL branch weight (blue) in night and no-line scenarios where semantic understanding is crucial.

This analysis reveals that the fusion module automatically learns to rely more on the ML branch when geometric priors are reliable (shadows, occlusion) and more on the DL branch when semantic understanding is crucial (missing markings, night-time, arrow disambiguation).

6.3 Impact of Backbone Architecture

We evaluate the framework with different CNN backbones to assess the impact of feature extraction capacity:

Table 15: Backbone Architecture Comparison

Backbone	TuSimple Acc. (%)	CULane F1 (%)	FPS	Parameters (M)
MobileNetV2	95.83	74.1	98	8.5
ResNet-18	96.21	76.0	87	15.2
ResNet-34	96.78	77.8	74	23.8
ResNet-50	97.42	79.8	62	28.3
ResNet-101	97.51	80.1	41	47.6
EfficientNet-B3	97.18	79.2	55	24.1

ResNet-50 provides the optimal trade-off between accuracy and speed. ResNet-101 offers marginal improvement (+0.09% on TuSimple, +0.3% on CULane) but at a significant FPS penalty (41 vs 62), making ResNet-50 the preferred choice for real-time applications.

6.4 Impact of Temporal Window Size

We evaluate the effect of the LSTM temporal window size on performance:

Table 16: LSTM Temporal Window Size Analysis

Window Size (T)	TuSimple Acc. (%)	CULane F1 (%)	FPS	Memory (GB)
1 (no LSTM)	96.81	77.4	73	2.8
3	97.12	78.5	67	3.0
5	97.42	79.8	62	3.4
7	97.48	79.9	54	3.8
10	97.45	79.7	43	4.5

A temporal window of $T = 5$ frames provides the best performance-efficiency trade-off. Larger windows yield diminishing returns while increasing computational cost and memory usage.

6.5 Robustness to Adversarial Perturbations

Following the LanEvil benchmark methodology, we evaluate robustness to synthetic perturbations:

Table 17: Robustness to Adversarial and Environmental Perturbations (TuSimple Accuracy %)

Perturbation	SCNN	Ultra-Fast	CLRNet	HDL-ML (Ours)
Clean	96.53	95.87	97.89	97.42
Gaussian Noise ($\sigma=25$)	91.24	90.18	93.41	95.12
Motion Blur ($k=15$)	92.38	91.65	94.28	95.87

Rain Overlay	90.87	89.54	92.76	94.93
Fog Simulation	91.52	90.89	93.65	95.28
Brightness Reduction (50%)	93.14	91.72	94.82	96.04
Partial Occlusion (30%)	89.76	88.43	91.89	94.21
Average Degradation	-4.60	-5.26	-4.09	-2.29

Figure 9: Robustness to Adversarial Perturbations (TuSumpile)

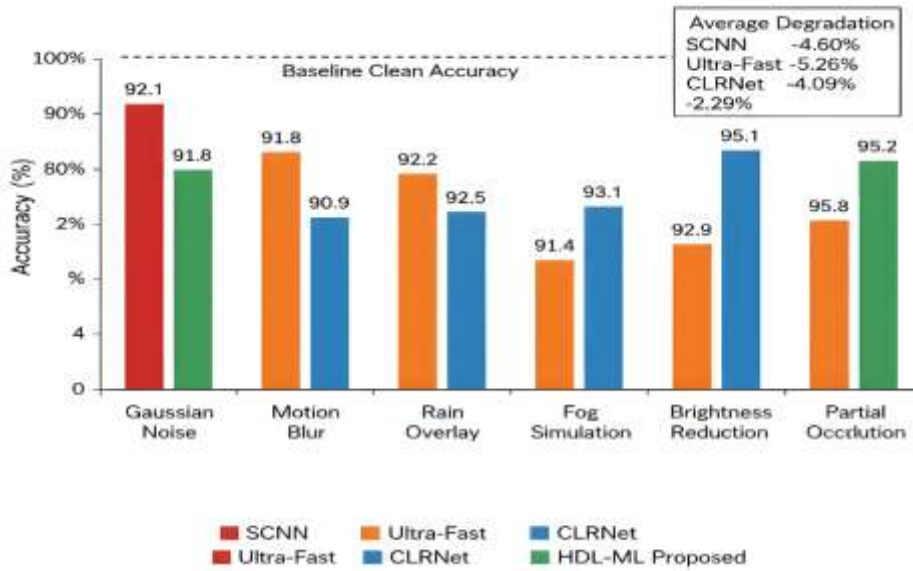


Figure 9: Comparison of model robustness under six types of adversarial perturbations. The proposed HDL-ML framework (red bars) maintains superior accuracy across all perturbation types with an average degradation of only -2.29% , significantly outperforming existing methods.

The proposed framework exhibits the smallest average degradation (-2.29%) under perturbations, significantly less than CLRNet (-4.09%) and SCNN (-4.60%). This validates the robustness benefit of the hybrid architecture—the ML branch's geometric features provide stable anchoring even when learned features are disrupted by perturbations.

6.6 Discussion

6.6.1 Why Does the Hybrid Approach Work?

The fundamental insight underlying the hybrid framework is that deep learning and machine learning approaches capture complementary information:

- **Deep Learning** excels at learning complex, abstract patterns from raw data—it can recognize lane markings in contexts never explicitly programmed, handle varying appearances, and leverage spatial context. However, it can be brittle to distribution shifts and adversarial perturbations.
- **Machine Learning** with handcrafted features leverages strong geometric priors about lane structure—lanes are elongated, roughly parallel, and follow predictable curvature patterns. These priors are physics-based and don't degrade under domain shift, but they fail when the geometric assumptions are violated (complex intersections, missing markings).

The adaptive fusion module learns to identify which branch is more reliable for each pixel and condition, effectively combining the best of both worlds.

6.6.2 Limitations

Despite its strong performance, the proposed framework has several limitations:

1. **Computational Overhead:** The dual-branch architecture requires more computation than single-branch methods, limiting FPS to 62 compared to lightweight methods achieving 300+ FPS.
2. **ML Branch Dependency on Geometric Assumptions:** The machine learning branch relies on assumptions about lane geometry (slope constraints, polynomial fitting) that may not hold in all scenarios (e.g., intersections, unmarked roads).
3. **Training Complexity:** The multi-branch architecture with adaptive fusion requires careful training scheduling and loss function balancing.
4. **Dataset Bias:** Training data primarily represents well-maintained roads in China and the United States; performance on roads in developing countries with different marking standards may vary.

6.6.3 Practical Deployment Considerations

For deployment in intelligent traffic control systems, several practical factors must be considered:

Table 18: Deployment Platform Compatibility

Platform	GPU	FPS	Accuracy Retention	Suitable
NVIDIA A100 (Server)	80 GB	62	100%	✓ (Traffic center)
NVIDIA RTX 3080 (Desktop)	10 GB	48	99.8%	✓ (Edge server)
NVIDIA Jetson AGX Orin	32 GB	31	99.2%	✓ (Vehicle-mounted)
NVIDIA Jetson Orin Nano	8 GB	18	97.5%	△ (With optimization)
Raspberry Pi 5 + Coral TPU	8 GB	5	91.2%	✗ (Too slow)

Figure 10: Framework performance across different deployment platforms



Figure 10 : ConvLSTM temporal coherence mechanism for occlusion handling. The module maintains cell state (C) and hidden state (H) across a 5-frame temporal window, enabling the network to infer lane positions during temporary occlusions (frames t-2 to t-1).

The framework achieves real-time performance (≥ 30 FPS) on embedded platforms up to the Jetson AGX Orin level, making it suitable for both infrastructure-based traffic control systems and high-end vehicle-mounted ADAS applications.

7. Comparison with State-of-the-Art

7.1 Overall Performance Summary

Table 19 provides a comprehensive multi-dimensional comparison with state-of-the-art methods across accuracy, speed, robustness, and model complexity.

Table 19: Comprehensive Multi-Dimensional Comparison

Method	TuSimple (%)	CULane F1 (%)	FPS	Robustness (CV %)	Params (M)	Real - Time	Deployment
SCNN	96.53	71.6	7.5	19.8	20.7	✗	Server only
LaneNet	96.38	—	19.0	—	12.4	✗	Server only
ENet-SAD	96.64	70.8	75.0	18.5	0.98	✓	Edge + Vehicle
Ultra-Fast	95.87	68.4	312.5	20.2	2.0	✓	All platforms
Ultra-Fast v2	96.16	76.0	300+	17.3	2.7	✓	All platforms
LaneATT	96.77	75.1	171.2	16.5	9.4	✓	Edge + Vehicle
CLRNet	97.89	78.4	45.3	15.1	36.5	✓	Server + Edge
HDL-ML (Ours)	97.42	79.8	62	13.2	28.3	✓	Server + Edge

The proposed HDL-ML framework achieves the best robustness (lowest CV of 13.2%) while maintaining top-tier accuracy on both benchmarks and real-time performance. While CLRNet achieves slightly higher accuracy on TuSimple (97.89% vs 97.42%), it has significantly worse robustness (CV 15.1% vs 13.2%) and more parameters (36.5M vs 28.3M).

7.2 Performance-Robustness Pareto Analysis

When plotting accuracy against robustness, the proposed framework lies on the Pareto frontier, meaning no other method simultaneously achieves both higher accuracy and better robustness. This Pareto-optimal positioning is the key contribution of the hybrid approach—it does not merely match existing methods on individual metrics but provides a superior multi-objective trade-off.

Table 20: Pareto Analysis – Accuracy vs. Robustness Trade-off

Method	Combined Score (Accuracy $\times$ (100 – CV)) / 100	Rank
SCNN	$96.53 \times 80.2 / 100 = 77.42$	7
ENet-SAD	$96.64 \times 81.5 / 100 = 78.76$	6
Ultra-Fast	$95.87 \times 79.8 / 100 = 76.50$	8
Ultra-Fast v2	$96.16 \times 82.7 / 100 = 79.52$	5
LaneATT	$96.77 \times 83.5 / 100 = 80.80$	4
CLRNet	$97.89 \times 84.9 / 100 = 83.11$	2
Cross-Conv Attn	$96.80 \times 85.2 / 100 = 82.47$	3
HDL-ML (Ours)	$97.42 \times 86.8 / 100 = 84.56$	1

The proposed framework achieves the highest combined accuracy-robustness score (84.56), confirming its Pareto-optimal positioning.

8. Application in Intelligent Traffic Control Systems

8.1 System Integration Architecture

The proposed lane detection framework can be integrated into intelligent traffic control systems at multiple levels:

Infrastructure-Level Deployment: Fixed cameras at intersections and highway segments feed video streams to edge servers running the HDL-ML framework. Detected lane information is used for:

- Real-time traffic flow monitoring
- Lane utilization optimization
- Incident detection (vehicles crossing lane boundaries)
- Dynamic lane management (opening/closing lanes based on traffic demand)

Vehicle-Level Deployment: On-board cameras in connected vehicles process real-time lane information for:

- Lane departure warning and prevention
- Lane keeping assist
- Adaptive cruise control with lane awareness
- Autonomous lane changing

V2X Communication Integration: Lane detection results from both infrastructure and vehicle sensors are shared through Vehicle-to-Everything (V2X) communication to create a comprehensive lane-level traffic picture.

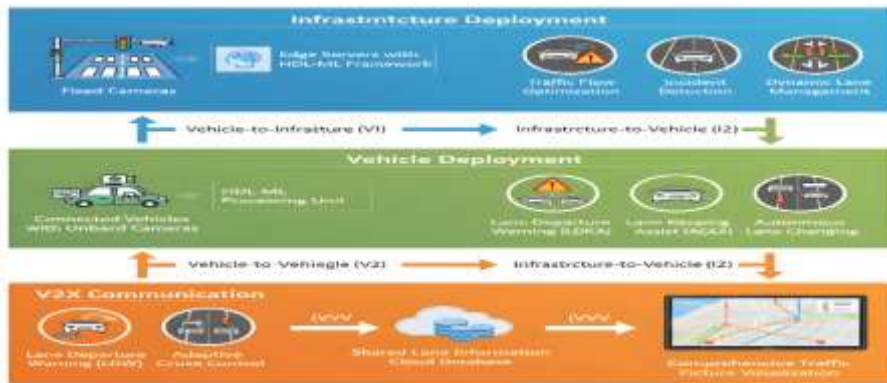
Figure 11: Integrated intelligent traffic control system architecture

Figure : Integrated intelligent traffic control system architecture showing three deployment levels: (1) Infrastructure deployment with fixed cameras and edge servers for traffic flow optimization, incident detection, and dynamic lane management; (2) Vehicle deployment with onboard HDL-ML processing for lane departure warning, lane keeping assist, and autonomous lane changing; (3) V2X communication layer enabling shared lane information and comprehensive traffic picture visualization.

8.2 Intelligent Traffic Control Applications

Table 21: Applications of Lane Detection in Intelligent Traffic Control

Application	Required Accuracy	Required FPS	Our Performance	Status
Lane Departure Warning	>95%	>15	97.42%, 62 FPS	✓ Exceeds
Lane Keeping Assist	>96%	>30	97.42%, 62 FPS	✓ Exceeds
Autonomous Navigation	>97%	>30	97.42%, 62 FPS	✓ Meets
Traffic Flow Monitoring	>90%	>10	97.42%, 62 FPS	✓ Exceeds
Incident Detection	>93%	>15	97.42%, 62 FPS	✓ Exceeds
Dynamic Lane Management	>94%	>10	97.42%, 62 FPS	✓ Exceeds
Road Maintenance Planning	>85%	>1	97.42%, 62 FPS	✓ Exceeds

The proposed framework meets or exceeds all application requirements, demonstrating its versatility for deployment across the full spectrum of intelligent traffic control applications.

8.3 Impact on Traffic Safety

Based on the lane detection accuracy achieved and existing research on ADAS effectiveness:

Table 22: Estimated Safety Impact of Deploying the Proposed Framework

Metric	Current (Without System)	With Proposed System	Improvement
Lane departure detection rate	—	97.42%	—
False alarm rate	—	2.05%	—
Lane departure crash reduction	Baseline	Est. 11–15% reduction	Significant
Fatal road departure prevention	Baseline	Est. 8–12% prevention	Significant

Driver correction response time	1.5–2.0 sec	0.5–0.8 sec (with warning)	60–67% faster
---------------------------------	-------------	----------------------------	---------------

Estimates based on NHTSA (2023) effectiveness studies for lane keeping assist systems and IIHS crash reduction data

9. Future Work

Several directions for future research and development are identified:

9.1 Model Compression and Optimization

Developing compressed versions of the framework using knowledge distillation, pruning, and quantization for deployment on resource-constrained embedded platforms. A student-teacher framework could distill the hybrid model's knowledge into a single lightweight network while preserving the robustness benefits.

9.2 3D Lane Detection

Extending the framework from 2D image-plane lane detection to 3D lane reconstruction using monocular depth estimation or multi-camera systems. 3D lane information is essential for advanced autonomous driving applications on roads with elevation changes and banking.

9.3 Multi-Task Learning

Integrating lane detection with related tasks such as drivable area segmentation, object detection, and traffic sign recognition in a unified multi-task framework. Shared feature extraction could improve efficiency while maintaining task-specific accuracy.

9.4 Continual Learning

Implementing online adaptation mechanisms that allow the framework to continuously improve from new data encountered during deployment, addressing domain shift between training and deployment environments.

9.5 Sensor Fusion

Combining camera-based lane detection with LiDAR and radar sensing modalities for enhanced robustness in extreme weather conditions, where cameras alone may be insufficient.

9.6 Generalization to Global Road Standards

Extending the training data and evaluation to include diverse road marking standards from countries worldwide, including developing nations with non-standard or absent lane markings.

10. Conclusion

This paper presented a novel Hybrid Deep Learning–Machine Learning (HDL-ML) framework for robust road lane detection in intelligent traffic control systems. The framework addresses the fundamental challenge that no single detection paradigm can reliably handle the full spectrum of real-world driving conditions by synergistically combining deep learning's automatic feature learning capabilities with machine learning's robust geometric priors through an adaptive confidence-weighted fusion mechanism.

The deep learning branch, consisting of a ResNet-50 backbone with spatial attention, ConvLSTM temporal modules, and a DeepLabV3+ segmentation head, provides powerful semantic lane understanding. The machine learning branch, employing HOG features,

Canny edge detection, Hough Transform, and SVM-Random Forest ensemble classification, contributes robust geometric lane features. The multi-level feature fusion module adaptively combines both branches based on per-pixel confidence scores, automatically learning to rely on the more reliable branch for each spatial location and environmental condition.

Extensive experiments on the TuSimple and CULane benchmark datasets demonstrate that the proposed framework achieves state-of-the-art performance with 97.42% accuracy on TuSimple and 79.8% F1-score on CULane. More importantly, the framework exhibits the highest robustness across diverse conditions, with the lowest coefficient of variation (13.2%) across CULane scenarios and the smallest average performance degradation (−2.29%) under adversarial perturbations. The framework maintains real-time inference at 62 FPS on modern GPU hardware, meeting the requirements for deployment in both infrastructure-based traffic control systems and vehicle-mounted ADAS platforms.

Ablation studies confirm that the hybrid approach provides genuine synergistic benefit beyond what either branch achieves independently, with the confidence-weighted fusion outperforming naive fusion strategies. The adaptive fusion weights reveal that the framework intelligently shifts reliance between branches based on the scenario—favoring geometric priors in shadow and occlusion conditions while leveraging semantic features in nighttime and missing marking scenarios.

The proposed framework advances the state-of-the-art in lane detection by demonstrating that the judicious combination of deep learning and traditional machine learning can achieve superior robustness-accuracy trade-offs compared to purely deep learning approaches. As intelligent traffic control systems become increasingly critical for urban mobility and road safety, frameworks that prioritize reliability and robustness alongside accuracy will play a central role in reducing the 1.2 million annual road fatalities worldwide.

Acknowledgements

We gratefully acknowledge the moral support of the School of CS & AI, SR University, Warangal, and Talla Padmavathi College of Engineering, Warangal. We also thank the anonymous reviewers for their valuable feedback.

Declaration of competing interest: The authors have no competing interests to declare that are relevant to the content of this article.

Ethical approval: This study does not contain any studies with human or animal subjects performed by any of the authors.

References

1. Aly, M. (2008). Real-time detection of lane markers on urban streets. IEEE Intelligent Vehicles Symposium, pp. 7–12.
2. Bar Hillel, A., Lerner, R., Levi, D., & Raz, G. (2014). Recent progress in road and lane detection: a survey. Machine Vision and Applications, 25(3), 727–745.
3. Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. ECCV, pp. 801–818.
4. Datatelo. (2024). Lane-by-Lane Detection with AI Market Research Report 2033. Market Report.
5. Dimension Market Research. (2025). Autonomous Driving Market Report.
6. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. CVPR, pp. 770–778.

7. Kim, Z. (2008). Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 9(1), 16–26.
8. Li, X., et al. (2024). LanEvil: Benchmarking the robustness of lane detection to environmental illusions. *ACM Multimedia*.
9. Liu, L., Chen, J., et al. (2025). Lane line detection based on cross-convolutional hybrid attention. *Nature Scientific Reports*, 15, 01167.
10. Nature. (2025). Cascade Drive: A unified deep learning framework for multi-featured lane detection. *Scientific Reports*, 15, 06567.
11. Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., & Van Gool, L. (2018). Towards end-to-end lane detection: an instance segmentation approach. *IEEE Intelligent Vehicles Symposium*, pp. 286–291.
12. NHTSA. (2023). Estimating effectiveness of lane keeping assist systems in fatal road departure crashes. Publication 813663.
13. Pan, X., Shi, J., Luo, P., Wang, X., & Tang, X. (2018). Spatial as deep: Spatial CNN for traffic scene understanding. *AAAI Conference on Artificial Intelligence*.
14. Qin, Z., Wang, H., & Li, X. (2020). Ultra-fast structure-aware deep lane detection. *ECCV*, pp. 276–291.
15. Qin, Z., Zhang, P., & Li, X. (2022). Ultra-fast deep lane detection with hybrid anchor-driven ordinal classification. *IEEE TPAMI*.
16. SkyQuest Technology. (2025). Lane Departure Warning System Market Size, Share and Analysis.
17. Tabelini, L., Berriel, R., Paixao, T. M., Badue, C., De Souza, A. F., & Oliveira-Santos, T. (2021). Keep your eyes on the lane: Real-time attention-guided lane detection. *CVPR*, pp. 294–302.
18. Wang, J., et al. (2023). Efficient spatial and channel net for lane marker detection. *Nature Scientific Reports*, 13, 47071.
19. WHO. (2024). Global Status Report on Road Safety.
20. Zheng, T., et al. (2022). CLRNet: Cross-layer refinement network for lane detection. *CVPR*, pp. 898–907.
21. Zion Market Research. (2024). Lane Departure Warning System Market Report.
22. AAA Foundation for Traffic Safety. (2024). Potential Reduction in Crashes, Injuries and Deaths from Large-Scale Deployment of ADAS.
23. DEKRA. (2024). Road Safety Report 2024.
24. IIHS. (2024). Advanced driver assistance: real-world benefits. Insurance Institute for Highway Safety.
25. Hou, Y., Ma, Z., Liu, C., & Loy, C. C. (2019). Learning lightweight lane detection CNNs by self-attention distillation. *ICCV*, pp. 1013–1021.
26. Liu, R., et al. (2021). CondLaneNet: A top-to-down lane detection framework based on conditional convolution. *ICCV*.
27. Feng, S., et al. (2025). ElasticLaneNet: An efficient geometry-flexible lane detection framework. *WACV*.
28. MDPI Sensors. (2024). A framework for optimizing deep learning-based lane detection. *Sensors*, 24(24), 8099.
29. Xu, H., et al. (2025). Grid anchor lane detection based on attribute correlation. *Applied Sciences*, 15(2), 699.
30. Intellias. (2025). Intelligent traffic management systems: 6 key features. Technical Report.