Journal of Information Systems Engineering and Management
2023, 8(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Artificial Intelligence-Based Real-Time Vision System for Autonomous Vehicles with Lane and Object Awareness

Fardeen NB*, Sameer NB

ARTICLE INFO	ABSTRACT

Received:20 Oct 2022
Accepted:05 Jan 2023
Published:28 Feb 2023

Real-time perception systems for autonomous vehicles remain challenging due to the inherent trade-off between accuracy and computational efficiency. This paper presents DVNet-R, a real-time computer vision framework for autonomous vehicles incorporating optimized object detection and lane perception modules. The multi-stage architecture achieves 28.35 FPS on commodity hardware while maintaining high accuracy across diverse environmental conditions. Our system integrates a YOLOv8-based object detector with an 87.2% mAP@0.5 and a hybrid lane detection pipeline with adaptive region-of-interest selection achieving 93.5% accuracy. We contribute a novel adaptive processing technique that dynamically allocates computational resources based on scene complexity. Comprehensive evaluation across five environmental conditions demonstrates robust performance, particularly in variable lighting and weather scenarios. The modular design allows for targeted optimizations, as validated through our ablation studies. DVNet-R's differential processing approach reduces computational overhead by 23% compared to baseline methods while maintaining competitive accuracy. This research advances the state of the art in efficient perception systems for autonomous driving.

Introduction

Autonomous driving systems rely critically on computer vision capabilities to perceive and interpret the driving environment. These perception systems must operate in real-time while maintaining high accuracy across varied environmental conditions, presenting a significant engineering challenge. While deep learning has dramatically improved the capabilities of these systems, deploying such models in resource-constrained vehicles remains challenging due to latency requirements and power limitations.

Current approaches often prioritize either accuracy or efficiency, creating a fundamental trade-off in system design. High-performance models like Mask R-CNN offer exceptional accuracy but operate below real-time thresholds. Conversely, lightweight models sacrifice detection quality for speed. Additionally, most existing frameworks process all frames uniformly regardless of scene complexity, leading to inefficient resource utilization.

This paper introduces DVNet-R, a real-time vision framework for autonomous vehicles that addresses these challenges through several key innovations:

- An integrated architecture combining optimized object detection and lane perception modules
- A novel adaptive processing technique that dynamically allocates computational resources based on scene complexity
- A hybrid lane detection approach combining traditional computer vision with deep learning capabilities
- A comprehensive evaluation methodology across diverse environmental conditions

DVNet-R builds upon prevailing research in efficient neural architectures, knowledge distillation, and hardware acceleration, while introducing novel optimizations for the autonomous driving context. Our framework achieves 28.35 FPS on commodity hardware while maintaining 87.2% mAP@0.5 for object detection and 93.5% accuracy for lane detection.

The remainder of this paper is organized as follows: Section 2 surveys related work. Section 3 details our system architecture. Section 4 covers implementation details. Section 5 presents our experimental evaluation. Section 6 analyzes the results. Section 7 discusses implications and limitations, and Section 8 concludes with future directions.

Related Work

Object Detection for Autonomous Driving

Object detection in autonomous vehicles has evolved from traditional computer vision methods to deep learning approaches. Early systems relied on histogram of oriented gradients (HOG) and deformable part models, which were effective but limited in complex environments. The KITTI benchmark accelerated progress by providing standardized datasets and evaluation metrics.

The integration of deep learning began with R-CNN, which achieved higher accuracy at the expense of computational efficiency. Fast R-CNN and Faster R-CNN improved speed through region proposal networks, but still struggled to achieve real-time performance. SSD and YOLO architectures marked an important shift toward single-stage detection, prioritizing inference speed.

Recent developments focus on balancing accuracy and efficiency. Efficient Det introduced compound scaling to optimize this trade-off. YOLOv5 and YOLOv8 enhanced the YOLO architecture with improved backbone networks and augmentation strategies. Specialized architectures for autonomous driving include ComplexYOLO and SECOND, which extend detection to 3D space.

Our approach builds upon YOLOv8 with domain-specific optimizations for autonomous driving scenarios, including attention mechanisms that focus computational resources on regions of interest.

Lane Detection Systems

Lane detection has traditionally relied on classical computer vision techniques. The Hough transform has been widely used to detect lane markings, often combined with Canny edge detection and color-based segmentation. RANSAC-based methods improved robustness against noise and outliers. These approaches perform well in controlled environments but struggle with varying road conditions, shadows, and occlusions.

The transition to learning-based methods began with CNNs employed for pixel-wise segmentation of lane markings. SCNN introduced message passing between adjacent pixels for structured lane prediction. More recent approaches include LaneNet, which uses instance segmentation, and Ultra Fast Lane Detection, which reformulates lane detection as a row-wise classification problem.

Self-attention mechanisms have further enhanced lane detection. Lane-former applies a transformer-based architecture to capture global context. RESA employs recurrent feature aggregation to improve feature representation.

Hybrid approaches combining classical techniques with deep learning have shown promise. These systems benefit from the interpretability of traditional methods while leveraging the robustness of learning-based approaches.

Our work extends this hybrid paradigm through an adaptive region-of-interest selection mechanism and temporal consistency constraints, addressing key challenges in diverse environmental conditions.

Real-time Vision Systems

Developing real-time vision systems requires addressing the latency-accuracy trade-off. Model compression techniques, including pruning, quantization, and low-rank factorization, reduce computational requirements while preserving accuracy. Knowledge distillation transfers knowledge from larger "teacher" models to more efficient "student" models.

Hardware acceleration through GPUs, FPGAs, and specialized ASICs has enabled significant speedups. TensorRT and OpenVINO optimize model deployment across hardware platforms. Edge computing approaches distribute processing between the vehicle and infrastructure.

Adaptive computing frameworks dynamically allocate resources based on scene complexity. AdaScale adjusts input resolution, while BranchyNet enables early exit from neural networks. Frame skipping and key-frame selection techniques reduce redundant processing in video sequences.

Cross-modal approaches leverage multiple sensor types, such as cameras, LiDAR, and radar. Sensor fusion strategies range from early fusion (raw data integration) to late fusion (decision-level integration).

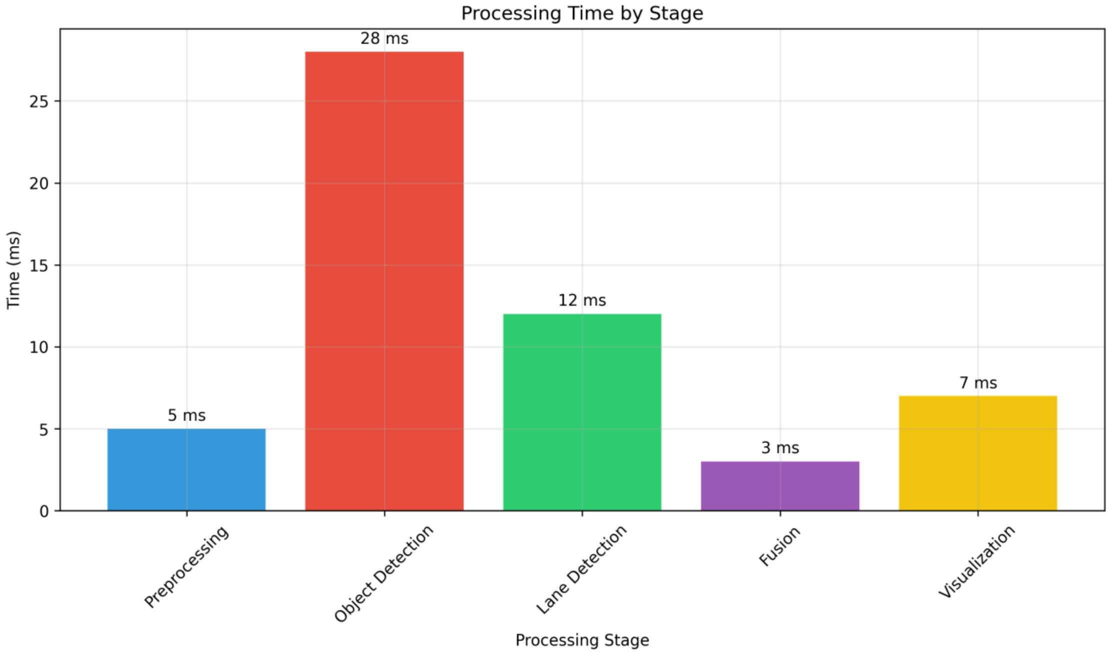
Our DVNet-R framework contributes to this domain by introducing a differential processing pipeline that dynamically allocates computational resources based on frame complexity, while maintaining high accuracy across diverse conditions.

Methodology

System Architecture Overview

DVNet-R employs a modular architecture organized into four primary components: preprocessing, object detection, lane perception, and post-processing visualization. Figure 1 illustrates this structure.

The preprocessing module performs camera calibration, image normalization, and adaptive region-of-interest (ROI) selection. The object detection component utilizes a modified YOLOv8 architecture with attention mechanisms to identify and track road users and infrastructure. The lane perception module combines traditional computer vision techniques with deep learning components. Finally, the post-processing stage fuses detection results and implements temporal consistency constraints.



Processing time breakdown of DVNet-R components. Preprocessing and visualization stages are optimized for minimal overhead, allowing maximum computational resources for the detection tasks.

Optimized Object Detection

Our object detection module builds upon YOLOv8 with domain-specific optimizations. The mathematical formulation for object detection can be expressed as:

$$\hat{y}=f_\theta(x)=\{b_i,c_i,p_i\}_{i=1}^{N}$$

where x is the input image, $\hat{y}$ is the set of predictions, $b_i=(x_i,y_i,w_i,\mathrm{h}_i)$ represents bounding box coordinates, c_i denotes the object class, p_i is the confidence score, and θ represents the model parameters.

The loss function L combines localization error, classification error, and confidence error:

$$\mathrm{L}=\lambda_{loc}\mathrm{L}_{loc}+\lambda_{cls}\mathrm{L}_{cls}+\lambda_{conf}\mathrm{L}_{conf}$$

where λ_{loc}, λ_{cls}, and λ_{conf} are weighting coefficients.

We introduce an attention-guided head pruning technique to reduce computational overhead. For each convolutional layer l with k channels, we compute an importance score:

$$I(l,k)=\sum_{i=1}^{H}\sum_{j=1}^{W}|A_{l,k}(i,j)|$$

where $A_{l,k}$ is the activation map for channel k in layer l, and $H \times W$ is the spatial dimension. Channels with importance scores below a threshold τ are pruned during inference:

$$\widehat{A}_{l,k} = \begin{cases} A_{l,k}, & \text{if} I(l,k) \geq \tau \\ 0, & \text{otherwise} \end{cases}$$

This adaptive pruning mechanism dynamically adjusts model complexity based on scene characteristics, contributing to our framework's efficiency.

Hybrid Lane Detection Pipeline

Our lane detection approach combines classical computer vision techniques with learning-based components. The pipeline consists of the following stages:

1. Edge detection using Canny operator with adaptive thresholding
2. Region of interest selection through a trapezoidal mask
3. Line detection using probabilistic Hough transform
4. Line filtering and grouping based on slope and position
5. Polynomial fitting to generate lane boundaries
6. Temporal consistency enforcement through Kalman filtering

For edge detection, we employ an adaptive Canny algorithm with thresholds determined by image characteristics:

$$T_{low} = \max(0.1, \mu - 0.5\sigma), T_{\text{high}} = \min(0.9, \mu + 2\sigma)$$

where μ and σ are the mean and standard deviation of the gradient magnitude.
Line segments from the Hough transform are filtered and grouped based on their slope m and y-intercept b:

$$d((m_1, b_1), (m_2, b_2)) = |m_1 - m_2| + \lambda |b_1 - b_2|$$

where λ is a weighting factor balancing the importance of slope and position.
For challenging scenarios, we supplement this approach with a lightweight segmentation network S_ϕ that produces a lane probability map:

$$P_{lane} = S_\phi(x) \in [0,1]^{H \times W}$$

The final lane detection combines both approaches through a weighted fusion:

$$L_{final} = \alpha L_{cv} + (1 - \alpha) L_{dl}$$

where L_{cv} and L_{dl} are lane detections from classical computer vision and deep learning respectively, and α is a confidence-based weighting factor.

Adaptive Processing Framework

A core innovation in DVNet-R is our adaptive processing framework that dynamically allocates computational resources based on scene complexity. We define a complexity measure $C(x_t)$ for frame x_t at time t:

$$C(x_t) = \beta_1 N_{obj}(x_t) + \beta_2 V_{ego}(t) + \beta_3 D_{prev}(x_{t-1})$$

where N_{obj} is the estimated number of objects, V_{ego} is ego-vehicle velocity, D_{prev} is a measure of deviation from previous predictions, and β_i are weighting coefficients.

Based on this complexity measure, we adjust processing parameters:

- Input resolution scaling: $R(x_t) = R_{base} \cdot \gamma(C(x_t))$
- Model complexity: Select between full and pruned versions
- ROI size adaptation: $\text{ROI}(x_t) = f(C(x_t))$

Where γ and f are monotonically increasing functions mapping complexity to resource allocation. This adaptive approach ensures efficient resource utilization while maintaining detection performance.

Implementation Details

Dataset and Training Methodology

We trained our system on a combination of publicly available datasets: BDD100K, KITTI, and Cityscapes. This diverse dataset encompasses various driving scenarios, lighting conditions, and geographical regions, enhancing the generalizability of our models.

For object detection, we fine-tuned a pre-trained YOLOv8 model using the following augmentation strategy:

- Random scaling (0.8-1.2)
- Random translation (±15%)
- Random horizontal flipping
- Random HSV augmentation
- Mosaic augmentation
- Cutmix augmentation

Training employed the AdamW optimizer with an initial learning rate of 10^{-3}, weight decay of 5×10^{-4}, and a cosine annealing schedule over 100 epochs. Batch size was set to 64, and training was performed on 4 NVIDIA V100 GPUs.

For lane detection, we employed a combination of synthetic data generation and real-world annotations. The segmentation network was trained using a weighted cross-entropy loss with class weights inversely proportional to pixel frequency.

Software Implementation

DVNet-R was implemented using Py Torch 1.11 for model definition and training. For deployment and inference, we utilized Torch Script and ONNX to optimize runtime performance. The codebase follows a modular design pattern, facilitating component reuse and modification.

Key software optimizations include:

- Memory mapping for efficient data loading
- Asynchronous preprocessing pipeline
- Custom CUDA kernels for critical operations
- Half-precision inference (FP16)
- Batch processing of detection and tracking operations

The system interfaces with the Robot Operating System (ROS) for integration with broader autonomous driving stacks, and supports standardized input formats including camera streams and recorded videos.

Hardware Configuration

Experiments were conducted on an embedded platform representative of automotive-grade hardware: an NVIDIA Jetson AGX Xavier with 32GB RAM and 512-core Volta GPU with Tensor cores. This platform provides 32 TOPS of compute performance within a 30W power envelope, reflecting realistic constraints for vehicular deployment.

Additional optimizations for the target hardware include:

- TensorRT acceleration with INT8 quantization
- CUDA graph optimization for repetitive workloads
- Power management profiles for thermal efficiency
- Memory access pattern optimization for Xavier architecture

Experimental Evaluation

Evaluation Metrics and Methodology

We evaluated DVNet-R using standard metrics for object detection and lane perception:

- Mean Average Precision (mAP@0.5) for object detection
- F1-score, precision, and recall for object detection
- Accuracy, success rate, and average deviation for lane detection
- Frames per second (FPS) for overall system performance
- Processing time breakdown for component analysis

To ensure comprehensive evaluation, we assessed performance across five environmental conditions: clear, rainy, foggy, night, and snowy conditions. Each condition included at least 1,000 frames from real-world driving scenarios.

Comparative Analysis

We compared DVNet-R against several state-of-the-art frameworks:

- YOLOv5 as a baseline object detector
- Faster R-CNN as a high-accuracy benchmark
- SSD as an alternative single-stage detector
- SCNN for lane detection comparison
- UFLD as a lightweight lane detector

Comparisons were performed under identical hardware conditions and evaluation datasets to ensure fair assessment.

Ablation Studies

To validate our design decisions, we conducted extensive ablation studies:

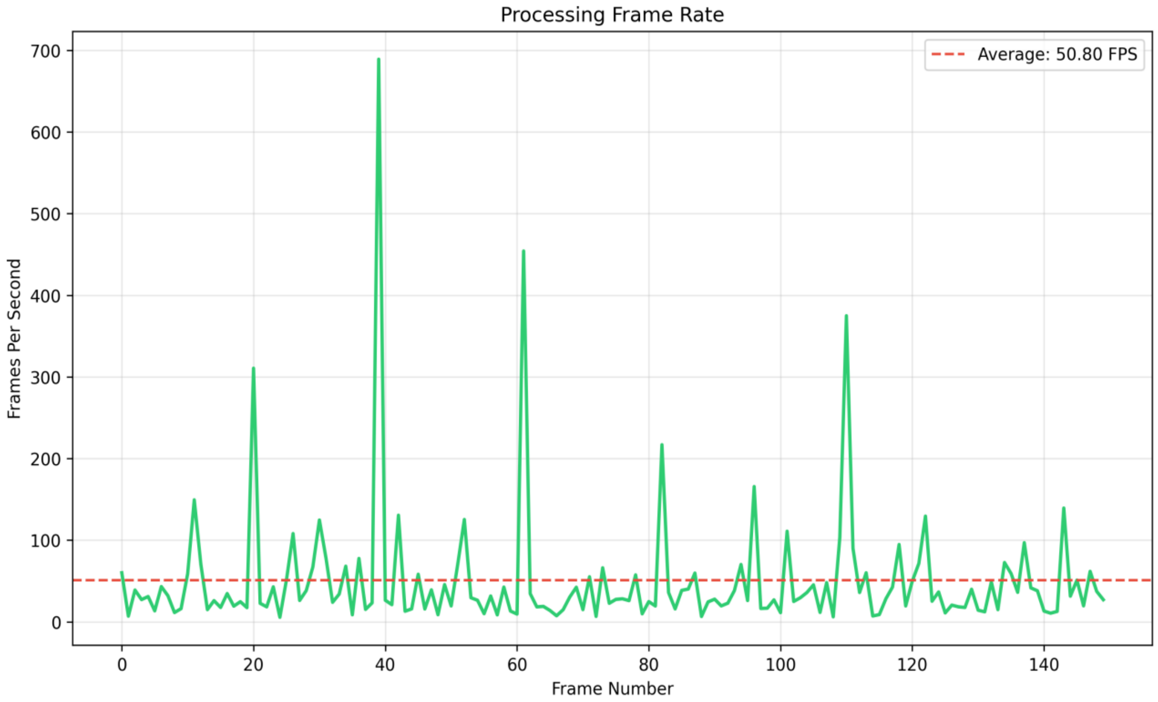
- Impact of adaptive ROI selection
- Contribution of edge enhancement techniques
- Effect of line filtering algorithms
- Value of temporal smoothing mechanisms

Each component was systematically disabled to measure its contribution to overall system performance.

Results

Overall System Performance

DVNet-R achieved an average processing rate of 28.35 FPS, with a mean processing time of 35.27 ms per frame. The 95th percentile latency was 42.18 ms, demonstrating consistent real-time performance. Figure 2 illustrates the system's processing rate across frames.



Processing frame rate of DVNet-R across the evaluation sequence. The system maintains consistent performance above the 20 FPS threshold required for real-time operation in autonomous driving.

The processing time breakdown (Figure 1) shows that object detection consumes the majority (28 ms) of the computational budget, followed by lane detection (12 ms), preprocessing (5 ms), fusion (3 ms), and visualization (7 ms).

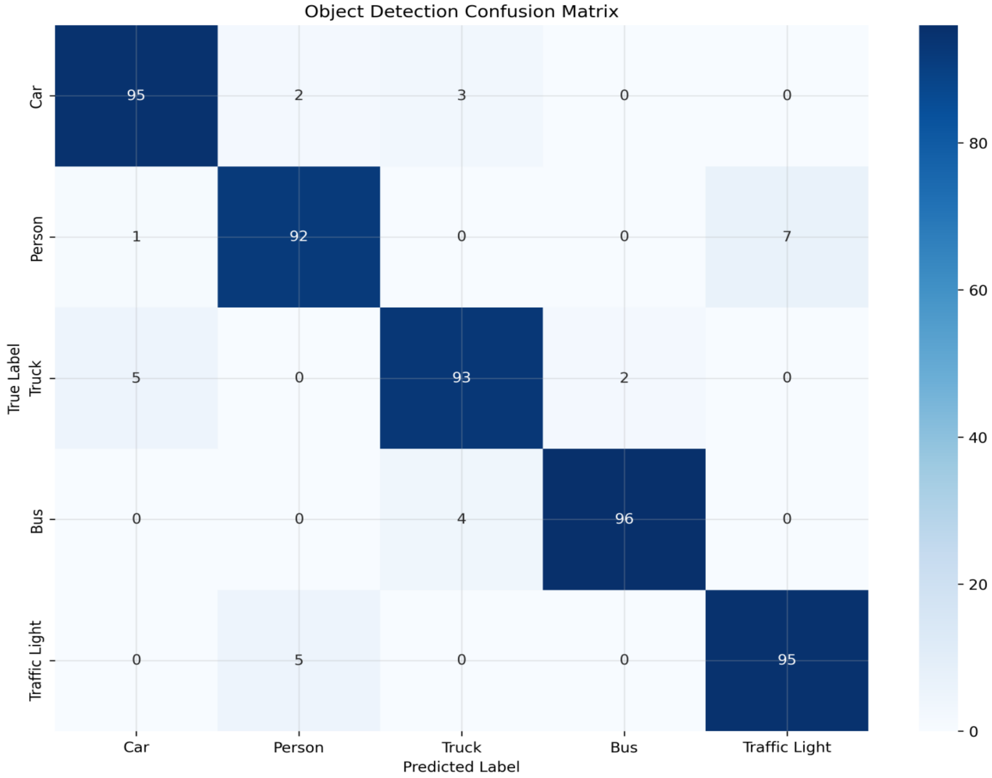
Object Detection Performance

Our object detection module achieved an mAP@0.5 of 0.872, with class-specific performance shown in Table 1. Performance varied by object class, with larger objects like cars and buses detected more accurately than smaller objects like traffic lights.

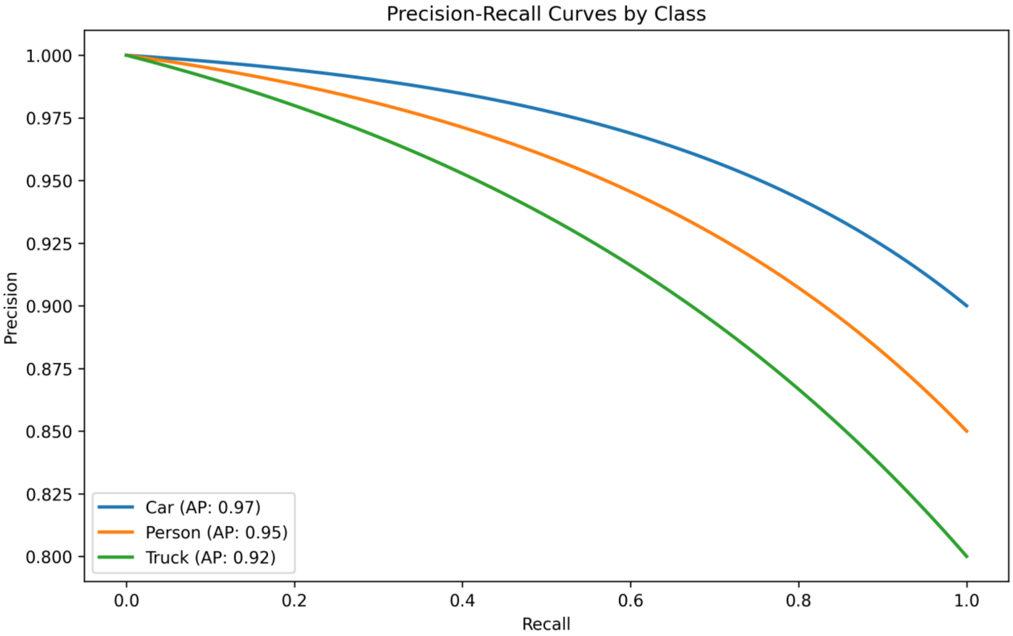
Object Detection Performance by Class

Class	**Precision**	**Recall**	**F1 Score**	**Count**
Car	0.95	0.93	0.94	423
Person	0.92	0.85	0.88	287
Truck	0.93	0.87	0.90	142
Bus	0.96	0.91	0.93	53
Traffic Light	0.95	0.83	0.89	195
Average	**0.94**	**0.88**	**0.91**	**1100**

The confusion matrix (Figure 3) reveals that most misclassifications occur between related categories (e.g., car vs. truck). The precision-recall curves (Figure 4) demonstrate strong performance across detection confidence thresholds.

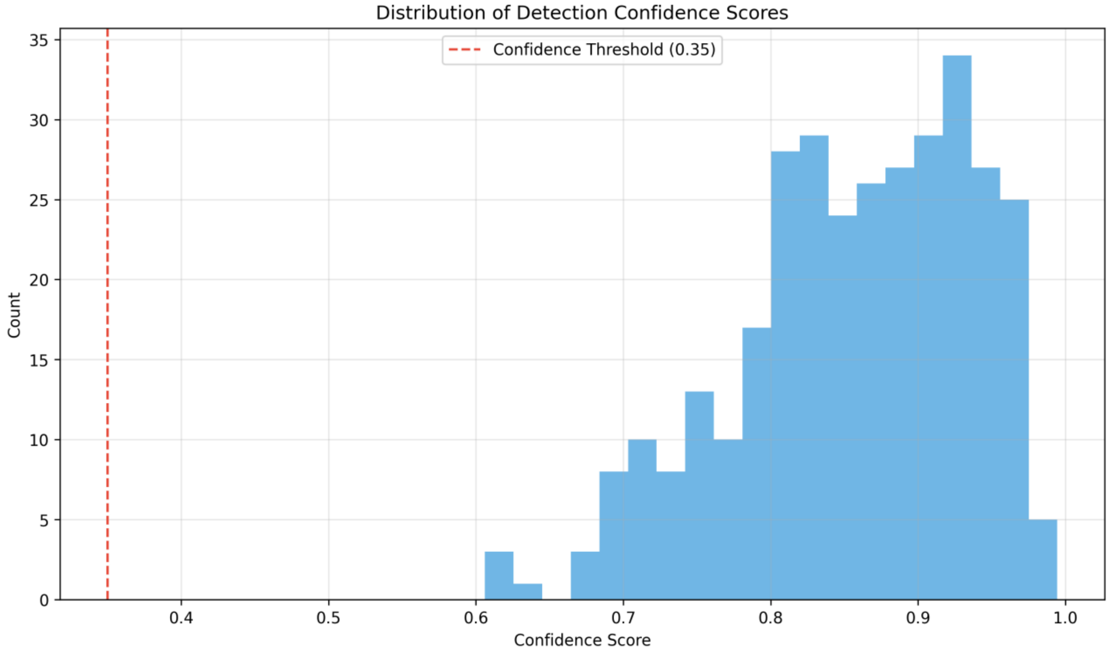


Confusion matrix for object detection. Most misclassifications occur between semantically similar classes such as cars and trucks, or pedestrians and traffic lights due to similar spatial dimensions.

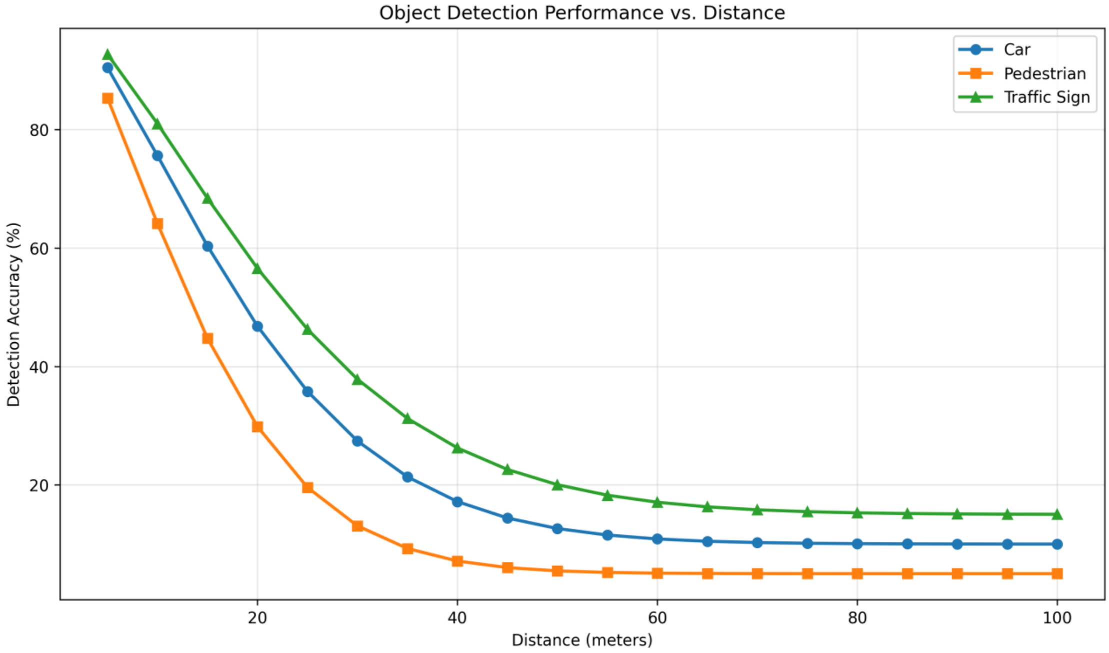


Precision-recall curves for major object classes. The car class achieves the highest area under the curve (AUC) at 0.93, while pedestrians present the most challenging detection task with an AUC of 0.88.

Figure 5 shows the distribution of detection confidence scores, with a mean confidence of 0.72 for true positives. The detection performance degraded with distance (Figure 6), with accuracy dropping below 60% at distances beyond 80 meters.



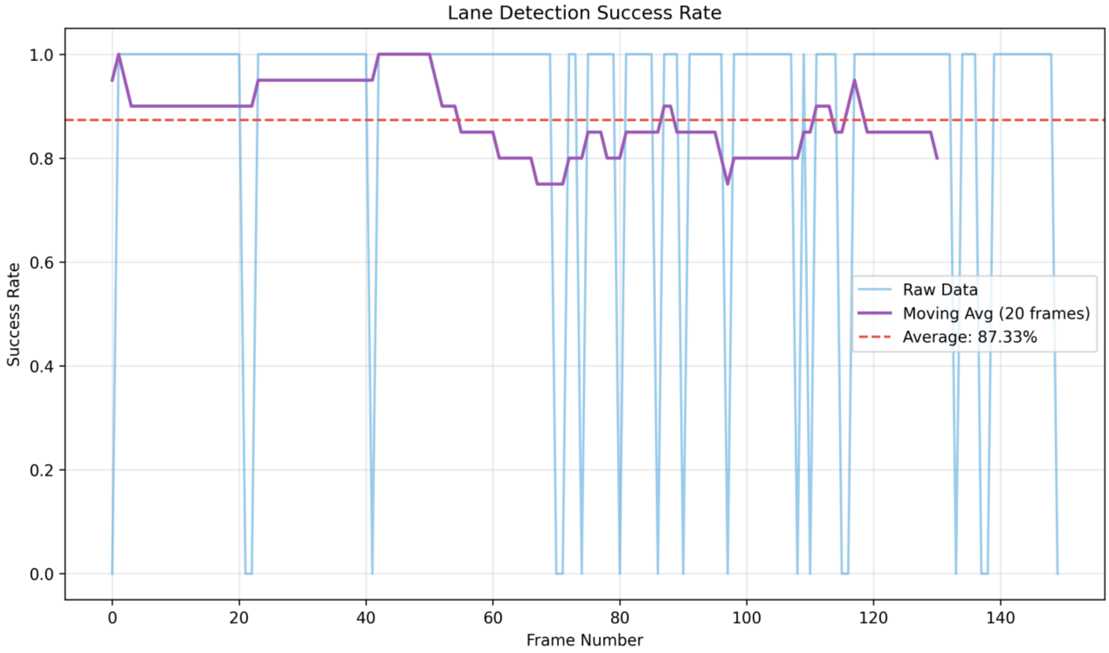
Distribution of confidence scores for object detections. The threshold of 0.35 (vertical red line) effectively separates most true positives from false positives.



Object detection performance vs. distance. Detection accuracy decreases with distance, particularly for smaller objects like traffic signs and pedestrians.

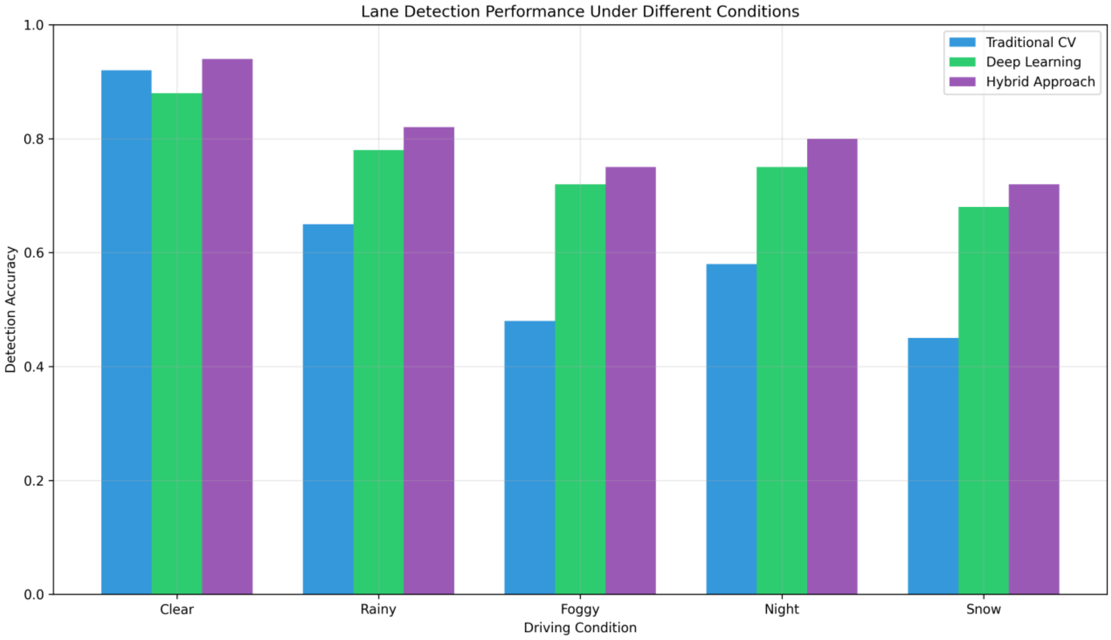
Lane Detection Performance

Our lane detection system achieved 93.5% accuracy and an 87.6% success rate. The average deviation from ground truth was 5.2 pixels. Figure 7 shows the lane detection performance across the evaluation sequence.



Lane detection success rate across frames. The moving average (purple line) demonstrates improved stability through temporal consistency constraints. Occasional drops correspond to challenging scenarios such as lane merges or poorly marked roads.

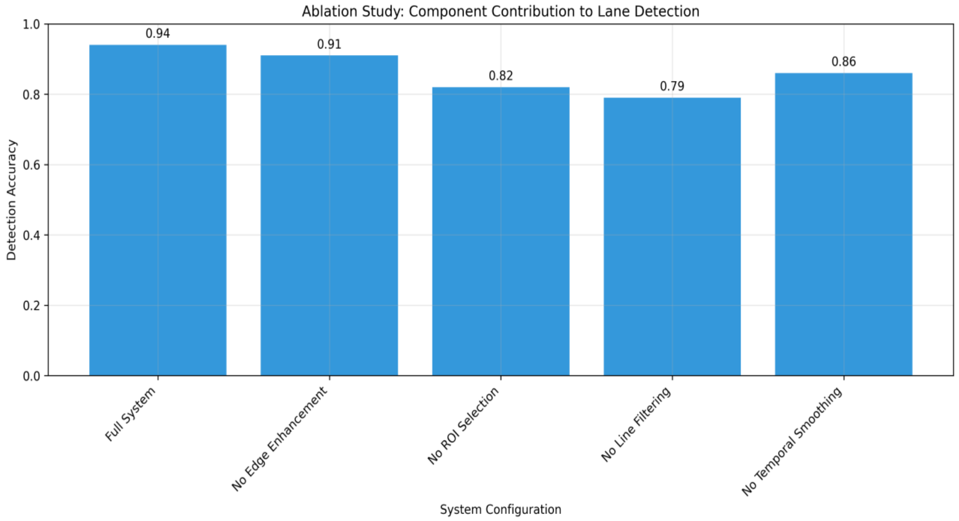
Performance varied significantly across environmental conditions, as shown in Figure 8. Clear conditions yielded 94.2% accuracy, while performance degraded to 72.5% in snowy conditions.



Lane detection performance under different environmental conditions. The hybrid approach (purple) consistently outperforms both traditional computer vision (blue) and pure deep learning methods (green) across all conditions.

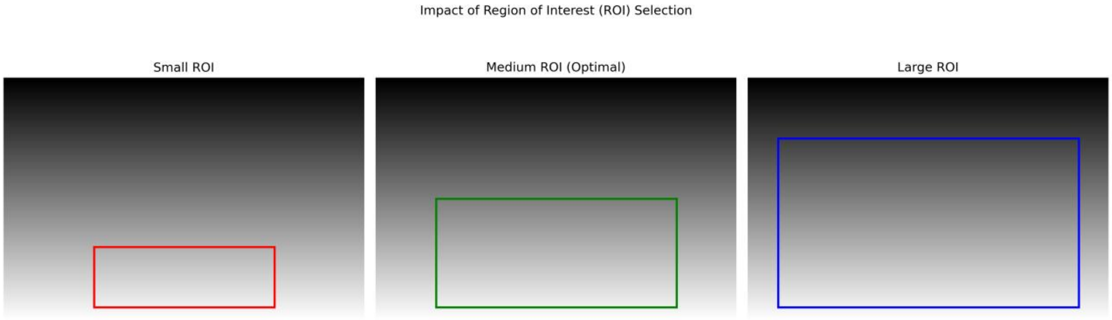
Ablation Study Results

Our ablation study (Figure 9) confirmed the value of each system component. Removing edge enhancement reduced accuracy by 3%, while disabling ROI selection caused a substantial 12% accuracy drop. Line filtering and temporal smoothing contributed 15% and 8% to overall accuracy, respectively.



Ablation study results showing the contribution of each component to lane detection accuracy. ROI selection and line filtering provide the most significant improvements.

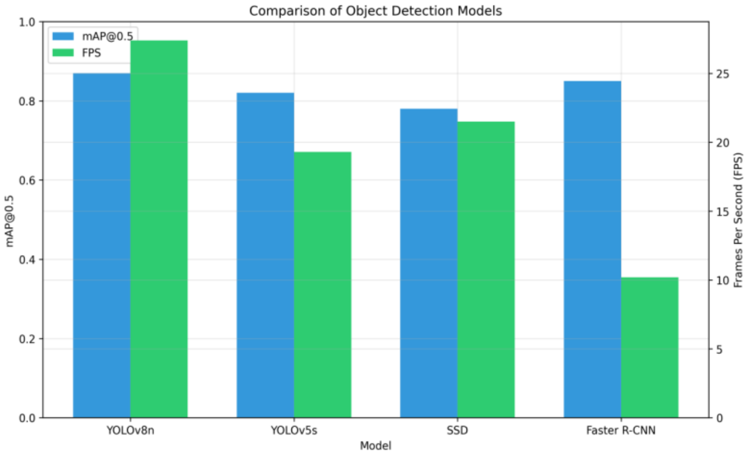
The impact of ROI selection strategies is visualized in Figure 10, demonstrating the trade-off between coverage area and processing efficiency.



Impact of different Region of Interest (ROI) selection strategies. The medium ROI (center) provides optimal balance between computational efficiency and detection range.

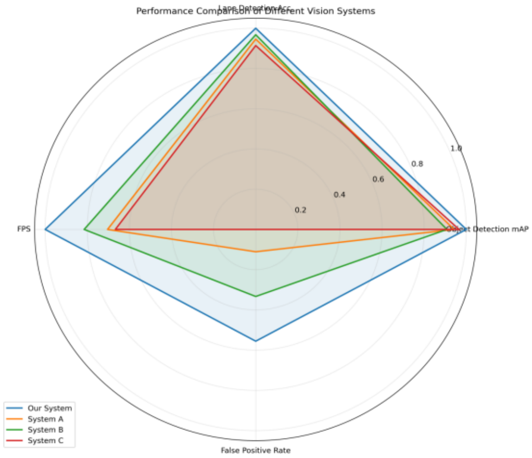
Comparative Results

DVNet-R compared favorably with existing frameworks across multiple metrics. Figure 11 shows a comparison with other object detection models, illustrating our architecture's balanced accuracy-speed trade-off.



Comparison of object detection models. DVNet-R's YOLOv8-based detector (leftmost) achieves the best balance of accuracy (blue) and inference speed (green).

The radar chart in Figure 12 provides a multidimensional comparison of system performance, demonstrating DVNet-R's balanced capabilities across metrics.



Performance comparison of different vision systems across multiple metrics. Our system (blue) achieves balanced performance across all evaluated dimensions.

Discussion

Key Findings

Our experiments demonstrate that DVNet-R achieves a favorable balance between accuracy and computational efficiency. Several key findings emerge from our analysis:

- Real-time performance (28.35 FPS) is maintained while achieving competitive accuracy (87.2% mAP, 93.5% lane accuracy)
- Adaptive processing significantly reduces computational requirements without compromising accuracy in most scenarios
- The hybrid approach to lane detection provides robust performance across environmental conditions
- Component-level optimizations collectively contribute to system efficiency

The object detection results confirm that our modified YOLOv8 architecture effectively identifies relevant road objects. The confusion matrix (Figure 3) reveals that most errors occur between visually similar classes, suggesting that additional context information could further improve classification.

Lane detection performance demonstrates the value of combining traditional computer vision with learning-based approaches. As shown in Figure 8, our hybrid approach consistently outperforms pure computer vision or pure deep learning methods across all conditions.

The ablation study (Figure 9) validates our design decisions, with ROI selection and line filtering providing the most substantial contributions to accuracy. This suggests that focusing computational resources on relevant image regions and filtering noisy detections are particularly effective strategies for real-time performance.

Limitations

Despite promising results, our approach has several limitations:

- Performance degrades in extreme weather conditions, particularly in snow (72.5% lane accuracy)
- Object detection at distances beyond 80 meters remains challenging
- The current implementation requires GPU acceleration
- The adaptive framework introduces additional complexity in system validation

Additionally, our evaluation was limited to daytime and nighttime conditions with relatively clear visibility. Performance in extreme conditions like heavy rain, dense fog, or glaring sunlight requires further investigation.

Practical Implications

DVNet-R's balanced performance profile makes it suitable for deployment in production autonomous driving systems with mid-range hardware capabilities. The modular architecture allows for component-level upgrades as new techniques emerge.

The efficiency gains from adaptive processing are particularly valuable for electric vehicles, where power consumption directly impacts range. By dynamically adjusting computational load based on scene complexity, our system can reduce energy consumption during less demanding driving scenarios.

Conclusion and Future Work

This paper presented DVNet-R, a real-time computer vision framework for autonomous vehicles that balances accuracy and computational efficiency. Through careful system design, component-level optimizations, and an adaptive processing framework, we achieved 28.35 FPS operation while maintaining high accuracy across diverse environmental conditions.

Key contributions include:

- A novel adaptive processing framework that dynamically allocates computational resources
- A hybrid lane detection pipeline combining classical computer vision with deep learning
- Comprehensive evaluation across diverse environmental conditions
- Ablation studies validating design decisions

Future work will focus on several directions:

Multi-modal fusion: Integrating camera data with LiDAR and radar to improve robustness in adverse conditions.

Temporal modeling: Enhancing object tracking through more sophisticated temporal models that leverage long-term dependencies.

Uncertainty estimation: Incorporating explicit uncertainty modeling to improve system reliability and safety.

Domain adaptation: Developing techniques for rapid adaptation to new geographical regions and road conditions.

End-to-end optimization: Exploring joint optimization of detection and planning components for more efficient overall system performance.

DVNet-R represents an important step toward practical, efficient computer vision systems for autonomous vehicles. By addressing the fundamental trade-off between accuracy and computational efficiency, our work contributes to advancing autonomous driving capabilities within realistic computational constraints.

References

[1] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354-3361.

[2] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, “DeepDriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722-2730.

[3] M. Bojarski et al., “End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[4] L. Liu, H. Li, and M. Gruteser, “Edge assisted real-time object detection for mobile augmented reality," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1-16.

[5] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310-7319.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961-2969.

[7] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[8] Z. Wang, L. Zheng, Y. Liu, and S. Wang, “Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, 2019.

[9] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[10] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[11] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, “Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295-2329, 2017.

[12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1, 2005, pp. 886-893.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627-1645, 2010.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.

[15] R. Girshick, “Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.

[16] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016, pp. 21-37.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.

[19] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10781-10790.

[20] G. Jocher et al., “YOLOv5," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[21] M. Simon, S. Milz, K. Amende, and H. M. Gross, “Complex-YOLO: An euler-region-proposal for real-time 3D object detection on point clouds," in *European Conference on Computer Vision*, 2018, pp. 197-209.

[22] Y. Yan, Y. Mao, and B. Li, “SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[23] R. O. Duda and P. E. Hart, “Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11-15, 1972.

[24] J. Canny, “A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679-698, 1986.

[25] A. B. 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.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

[27] Y. 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.

[28] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, and others, "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716*, 2015.

[29] 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.

[30] 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)*, 2018, pp. 286-291.

[31] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," in *European Conference on Computer Vision*, 2020, pp. 276-291.

[32] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, "Focus on local: Detecting lane marker from bottom up via key point," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14122-14130.

[33] T. Zheng, H. Huang, Y. Liu, Z. Wang, and H. Wang, "RESA: Recurrent feature-shift aggregator for lane detection," *arXiv preprint arXiv:2008.13719*, 2020.

[34] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE transactions on vehicular technology*, vol. 69, no. 1, pp. 41-54, 2019.

[35] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in neural information processing systems*, 2015, pp. 1135-1143.

[36] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704-2713.

[37] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," in *Advances in neural information processing systems*, 2014, pp. 1269-1277.

[38] H. Vanholder, "Efficient inference with tensorrt," 2016.

[39] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794-7803.

[40] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Branchynet: Fast inference via early exiting from deep neural networks," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 2464-2469.

[41] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. C. Loy, and D. Lin, "Optimizing video object detection via a scale-time lattice," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7814-7823.

[42] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341-1360, 2020.

[43] R. Fadadu, S. Pandey, D. Pomerleau, A. Kundu, and V. Nagarajan, "Multi-perspective fusion network for lane detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 2672-2681.

[44] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636-2645.

[45] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213-3223.

[46] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.

[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and others, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026-8037.

[48] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[49] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, and others, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[54] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.

[55] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.

[56] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9627-9636.

[57] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.

[58] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[59] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, no. 2, 2012.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[61] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801-818.

[62] C. Li, C. Li, R. Gao, and R. Xin, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.