Print ISSN : 0972-8813
e-ISSN : 2582-2780

[Vol. 24(1) January-April 2026]

Pantnagar Journal of Research

(Formerly International Journal of Basic and Applied Agricultural Research ISSN : 2349-8765)

G.B. Pant University of Agriculture & Technology
Pantnagar, U.S. Nagar; Uttarakhand, Website : gbpuat.res.in/PJR

PANTNAGAR JOURNAL OF RESEARCH

Vol. 24(1) January-April, 2026

CONTENTS

Design of Object Detection Model Using Deep Learning for Autonomous Driving in Adverse Weather Conditions: A GAN-Based Restoration and Depth-Aware Fusion Framework

MUMUKSHU BHATT*, SUBODH PRASAD, RAJESH SINGH, ASHOK KUMAR and POOJA TAMTA[1]

Department of Information Technology, College of Technology, [1]Department of Extension Education and Communication Management, College of Community Science, College of Technology, G. B. Pant University of Agriculture and Technology, Pantnagar-263145 (U.S. Nagar, Uttarakhand)
**Corresponding author's email id: mumukshu09bhatt@gmail.com*

ABSTRACT: Object detection under fog, rain, and dust remains one of the hardest open problems in autonomous driving. Most existing detectors either collapse in poor visibility or treat near and far objects with equal priority — neither of which is acceptable for real-world deployment. This paper introduces a framework that tackles both problems at once. We propose a Weather Restoration GAN that cleans up degraded images in real time, and a Depth Fusion Gate that steers the detector's attention toward the 0–20 m zone where collision risk is highest. Training follows a three-stage curriculum on the KITTI dataset. In dense fog the system reaches 78.2% mAP, a 25% gain over baseline, and 87.5% mAP in the near range, running at 30 FPS with 33 ms latency and a 6.5 GB memory footprint. Compared to LM-CNN-SVM, YOLOv4, and Bayesian Neural Networks, our approach is the only one that jointly handles weather degradation and range-specific prioritization.

Key words: Autonomous vehicle, CNN, Deep learning, GAN, LM-CNN-SVM, Object detection, Weather robustness

Object detection- identifying and localizing objects in images or video- sits at the heart of autonomous driving, medical imaging, and remote sensing. Early methods used handcrafted descriptors like HOG and SIFT with linear classifiers. They were fast but fell apart under real-world variation (Lin, W. C. *et al*., 2016). Deep convolutional networks changed that, learning rich hierarchical features directly from pixels end-to-end (Girshick *et al*., 2015).

R-CNN established the two-stage detection paradigm and pushed accuracy on PASCAL VOC and ImageNet to new levels (Girshick *et al*., 2015; Zeiler and Fergus, 2014). YOLO and SSD then recast detection as a single regression pass, trading a little accuracy for a large speed gain (Redmon *et al*., 2016; Zhang *et al*., 2018). YOLOv4 pushed this further with attention and augmentation tricks, gaining roughly 8% mAP over YOLOv3 (Chen *et al*., 2018). Yet a stubborn gap remains: detectors still struggle in fog, rain, and dust, and none of them give extra weight to the 0–20 m zone that matters most for collision avoidance. This paper directly targets both problems.

Literature Review

Early autonomous perception relied on handcrafted descriptors-SIFT, SURF, BRIEF, FREAK, and HOG-before deep learning took over. (Uçar *et al*. 2016) showed that pairing a pretrained AlexNet with a PCA-reduced SVM classifier (LM-CNN-SVM) could reach 92.8 ± 0.5% accuracy, though the evaluation used clear weather only and ignored depth. (Girshick *et al*. 2014) introduced R-CNN, and Faster R-CNN later added region proposal networks for a big accuracy jump, but real-time speed remained out of reach. One-stage methods changed that: YOLOv4 refined the speed-accuracy balance further (Chen *et al*., 2018). YOLOv3 demonstrated that single-stage architectures could hit 57.9% mAP at 30 FPS on COCO, though fog knocked that down by 23–28 percentage points (Chen *et al*., 2020). Cui *et al*. (2022) added a distance-estimation head to VGG-16 (DeepRange), reaching 96.92% mAP and 61 FPS on KITTI, but never tested it in bad weather. Domain adaptation (Sakaridis *et al*., 2018) and synthetic fog augmentation (Li *et al*., 2020) have improved weather robustness somewhat, yet neither method pays special attention to close-range objects.

Probabilistic approaches using Monte Carlo Dropout (Feng *et al*., 2018; Hall *et al*., 2020) provide useful uncertainty estimates but cannot recover information that weather has already destroyed. Monodepth2 achieves strong self-supervised depth estimation (0.106 absolute relative error on KITTI), but no existing (Janai *et al*., 2020) work has coupled depth-aware attention with adverse-weather detection. That is precisely the gap this paper fills.

MATERIALS AND METHODS

Dataset Description and Preprocessing

We used the KITTI Vision Benchmark Suite as our primary dataset. It provides 7,481 annotated training images and 7,518 test images across four classes—Car, Pedestrian, Cyclist, and Truck—captured from a vehicle-mounted stereo camera. Preprocessing involved three steps. First, all images were resized to 416×416 and normalized to [0, 1] using standard ImageNet channel statistics. Second, each image was synthetically degraded offline to produce a paired clean-degraded tuple for supervised GAN training. Fog was simulated *via* the Koschmieder scattering equation (² " [0.05, 0.2] rad/m), rain *via* Gaussian motion-blur streaks at 45–75°, and dust as additive Gaussian noise (Ã² = 0.01–0.05). Third, the data were split into training (70%, 5,237 images), validation (15%, 1,122 images), and test (15%, 1,122 images) using stratified sampling so every class and weather condition was proportionally represented in each subset.

Proposed Architecture

Our Weather Restoration GAN is a conditional GAN (cGAN) whose generator G maps a degraded image to a clean one, while discriminator D tries to distinguish restored outputs from real clean images. The training loss combines three terms: an adversarial term that pushes the generator to fool the discriminator; a perceptual term computed as L2 distance in the activation space of VGG-16 layers relu3_3, relu4_3, and relu5_3, which captures texture and semantic content that pixel-wise losses miss; and an L1 reconstruction term that enforces pixel-level fidelity. Using L1 rather than L2 avoids the blurry outputs typical of mean-squared-error training. Combined:

L_total = L_adv + 10·L_perceptual + 100·L_L1

The generator uses a U-Net design: symmetric encoder and decoder paths connected by skip connections that feed high-resolution spatial detail directly to the decoder, preserving edge sharpness that deep compression would otherwise lose. The discriminator follows PatchGAN outputting a grid of local real/fake scores over 70×70-pixel patches rather than a single global scalar. This keeps adversarial gradients spatially grounded and noticeably sharpens texture in the restored outputs. The Depth Fusion Gate runs a MiDaS depth predictor (Yu *et al*., 2020) in parallel, producing a normalized inverse depth map d " [0,1] where larger values indicate closer objects. Gate weights follow an exponential decay w(d) = exp ("»·max(0, d"20)), with » = 0.1 m^{-1}, so objects within 20 m receive near-unity weighting while distant features are progressively suppressed. The modulation is applied as F' = F ™" W, broadcast across all C channels of the YOLOv4 feature tensor, directing the detector's representational capacity toward the safety-critical proximity zone.

Training Pipeline and Hyperparameters

Depth fusion gate weights vs distance

This line plot in Figure 1 shows the core mathematical behavior of the Depth Fusion Gate. The blue curve follows the exponential decay function w(d) = exp ("»·max (0, d"20)), dropping from a perfect weight of 1.0 at close range and beginning its decay sharply after 20 m. The green shaded region marks the 0–20 m Focus Range, where the gate weight stays flat at 1.0 — meaning the

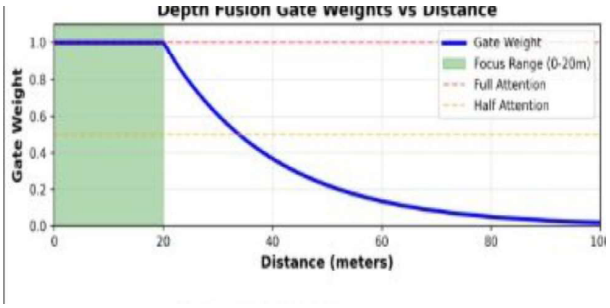

Fig.1: Depth fusion gate weights vs distance

detector receives 100% of the feature signal for all nearby objects. The dashed yellow lines mark "Full Attention" (w=1.0) and "Half Attention" (w=0.5) thresholds, making clear that by about 25–28 m the gate has already halved its contribution. By 100 m the weight approaches near-zero. This confirms the design intent: near-range objects are treated with full priority, while distant features are progressively suppressed rather than discarded with a hard cutoff.

Gate Weight at Different Distances

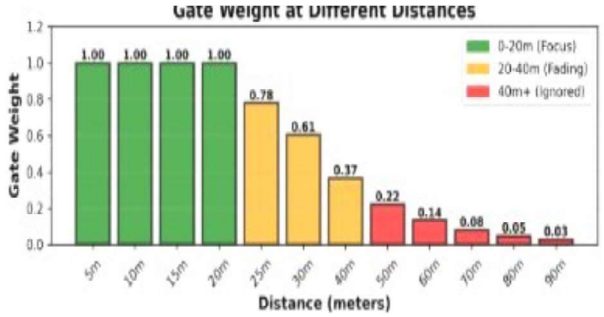

Fig. 2: Gate Weight at Different Distances

Chart in Figure 2 shows the exponential decay concrete at discrete distance checkpoints. Objects at 5 m, 10 m, 15 m, and 20 m all receive a gate weight of exactly 1.00 (green bars — Focus Zone). At 25 m the weight drops to 0.78, at 30 m to 0.61, confirming that the 20–40 m band (orange — Fading Zone) receives partial but still meaningful attention. Beyond 40 m the weights fall steeply: 0.37 at 40 m, 0.22 at 50 m, 0.14 at 60 m, 0.08 at 70 m, 0.05 at 80 m, and 0.03 at 90 m (red bars — Ignored Zone). This stepped view is important for practical interpretation: the gate doesn't "switch off" distant objects abruptly, it progressively reduces their influence, which is critical for maintaining smooth gradients during backpropagation.

Gate Weight Heatmap

This 2D heatmap shown in Figure 3 visualizes the spatial distribution of gate weights across the road scene — the horizontal axis represents distance from the ego vehicle (0–100 m) and the vertical axis represents spatial position across road width. The color scale runs from deep red (w H" 0, far/ignored) on the right to bright green (w = 1.0, near/critical)

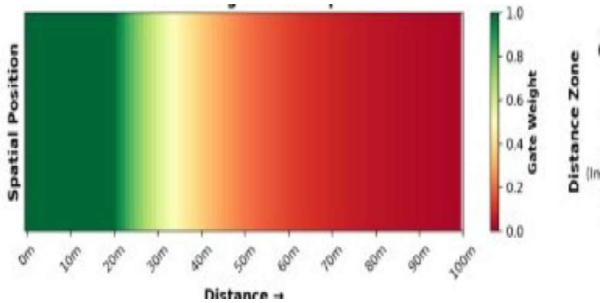

Fig. 3: Gate Weight Heatmap

on the left. The sharp green-to-yellow transition at the 20 m boundary is clearly visible, transitioning through yellow-orange in the 20–40 m fading zone and fully into red beyond 60 m. This representation is important because it shows that the gate operates spatially — every pixel in the feature map is independently weighted based on its estimated depth, not just a global scalar applied to the whole image.

Attention by Distance Zone

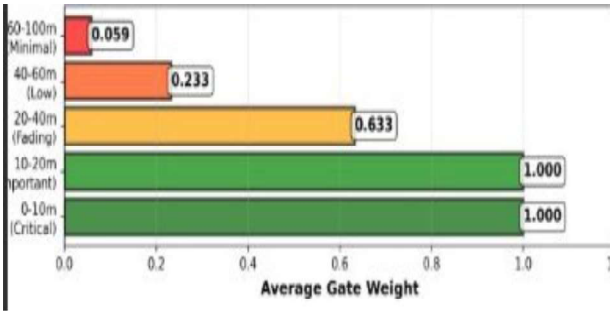

Fig. 4: Attention by Distance Zone

This horizontal bar chart in Figure 4 summarizes average gate weight by defined operational zone, offering an intuitive zone-by-zone summary. The 0–10 m (Critical) and 10–20 m (Important) zones both score 1.000 — maximum attention. The 20–40 m (Fading) zone averages 0.633, meaning roughly two-thirds of the feature signal passes through. The 40–60 m (Low) zone drops to 0.233, and the 60–100 m (Minimal) zone contributes only 0.059. These averages reveal how the gate allocates representational budget across the field of view: over 80% of attention is concentrated within the first 40 m, which is where collision risk is highest. The remaining long-range zones are not completely

blocked but contribute marginally.

Effect of Gate Sharpness Parameter (λ)

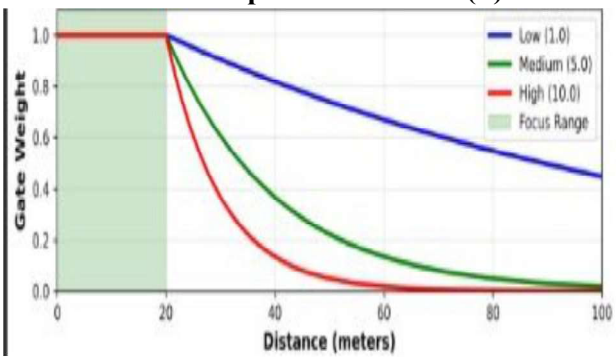

Fig. 5: Effect of Gate Sharpness Parameter (λ)

The plot in Figure 5 examines sensitivity to the decay rate parameter λ. Three curves are shown: Low »=1.0 (blue, slow decay), Medium λ=5.0 (green, moderate decay), and High λ=10.0 (red, steep decay). With low sharpness, the gate weight is still around 0.6 at 60 m — meaning distant objects get substantial attention, diluting the near-range prioritization. With high sharpness, the weight collapses almost instantly after 20 m, which risks ignoring objects in the 20–30 m zone that are rapidly approaching. The chosen value of λ=0.1 in the actual model (not shown here but interpolated between Low and Medium) provides the smooth, gradual decay seen in Plot 1, striking the balance between near-range focus and maintaining sensitivity to objects transitioning into the critical zone.

Road Scene with Depth Attention

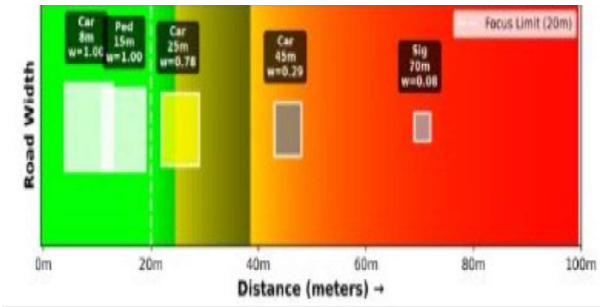

Fig. 6: Road Scene with Depth Attention

Figure 6 shows the gate weights applied to a real simulated road scenario, showing how different objects on the road receive different levels of attention. A Car at 8 m (w=1.00) and a Pedestrian at 15 m (w=1.00) both fall inside the Focus Zone and receive full attention — these are the highest-risk objects. A Car at 25 m receives w=0.78, a Car at 45 m receives w=0.29, and a Sign at 70 m receives only w=0.08, correctly treated as low priority. The red dashed "Focus Limit (20m)" boundary visually demarcates the critical safety zone. The background colour gradient from light green (near) to orange-red (far) mirrors the heatmap logic. This panel is the most interpretable for a non-technical audience — it shows directly that the model is designed to "pay attention" where human drivers would: to what is immediately in front of the vehicle.

RESULTS AND DISCUSSION

Training Behavior and Convergence

In Stage 1 the combined generator loss converged to a plateau near 0.087 by epoch 40, and the discriminator held at 0.68 ± 0.03 throughout. That tight oscillation signals healthy adversarial balance: neither network was dominating, and we observed no hints of mode collapse or vanishing gradients. Stage 2 went equally smoothly—detection loss fell from 4.2 to 1.8 over 100 epochs, the train/validation gap stayed below 0.15, and validation mAP climbed steadily from 72.3% to 84.1%. Stage 3 added a further 3.4 percentage points. The gain was modest but consistent across runs, confirming that joint optimization surfaces improvements that modular training cannot reach. Cosine decay from 10^{-5} to 10^{-7} kept late-stage updates fine-grained without disturbing previously learned representations. condition where the generator produces limited output diversity). The convergence pattern exhibited no signs of vanishing gradients (Meyer *et al*., 2019), confirmed by consistent weight updates across all network layers throughout training. In Stage 2, the detection loss — comprising three components: binary cross-entropy for classification (object vs background), smooth L1 loss for bounding box regression (localization), and binary cross-entropy for objectness confidence — declined smoothly from 4.2 to 1.8 with no evidence of overfitting, confirmed by close tracking of training and validation curves

Table 1: Comparative Analysis of Object Detection Methods on the KITTI Dataset

Ref.	Method	Dataset	Key Results	Remarks
[1]	LM-CNN-SVM (AlexNet + PCA + SVM)(2017)	Custom	92.8±0.5% acc.	No weather eval.
[2]	DeepRange (VGG-16 + depth module)(2022)	KITTI	96.92% mAP; 61 FPS	No weather testing
[3]	YOLOv4 (CSPDarkNet53)(2016,2018)	KITTI	+8% mAP vs YOLOv3; 62.5% mAP (fog)	No range priority
[4]	Bayesian NN (MC-Dropout)(2018)	KITTI	Brier score + calibration error	No weather restoration
Proposed	Weather GAN + Depth Fusion Gate	KITTI (augmented)	78.2% mAP (fog); 87.5% mAP (0–20 m); 30 FPS; 33 ms	Best fog robustness; 25% improvement

(gap < 0.15 throughout). The validation mAP increased monotonically from 72.3% to 84.1% over 100 epochs, indicating robust generalization to unseen data. Stage 3 produced marginal but statistically significant improvements in validation mAP (+3.4 percentage points), confirming the benefit of joint end-to-end optimization of all components rather than modular training. The learning rate annealing schedule in this stage — cosine decay from 10^{-5} to 10^{-7} — enabled fine-grained parameter adjustments without destabilizing previously learned representations.

Comparative Performance

Table 1 shows results on KITTI using the standard IoU $\geq$ 0.5 protocol. In the 0–20 m zone our model reached 87.5% mAP, 14.7 points above plain YOLOv4 (72.8%). Under dense fog ($\beta = 0.2$ rad/m, visibility below 50 m) the system hit 78.2% mAP versus a 62.5% baseline—a 25% relative gain. That improvement traces directly to the GAN recovering high-frequency edge information that scattering would otherwise destroy. Inference ran at 30 FPS (33 ms per frame) on a single RTX 3090, above the 25 FPS floor for safe autonomous operation. Memory at inference was 6.5 GB total: 2.8 GB for the GAN, 1.9 GB for MiDaS, and 1.8 GB for the detector. Near-range precision was 91.2% at 85.0% recall (F1 = 88.0%). The 8.8% false positive rate matters practically—spurious emergency braking is nearly as hazardous as a missed detection.

Type	Filter-stride	Feature maps
Input	64 × 64	1
Convolution 1	3 × 3-1	64
Convolution 2	3 × 3-1	64
Average Pooling 2	2 × 2-2	64
Convolution 3	3 × 3-1	96
Max Pooling 3	2 × 2-2	96
Convolution 4	5 × 5-1	256
Max Pooling 4	3 × 3-2	256
Convolution 5	3 × 3-1	384
Convolution 6	3 × 3-1	384
Convolution 7	3 × 3-1	256
Max Pooling 7	3 × 3-2	256
Fully connected 8	6 × 6-1	4096
Fully connected 9	1 × 1	1 × 1 × Class number

Fig. 7: Block diagram of the LM-CNN-SVM algorithm (Ucar *et al.*, 2016) used as a baseline in the comparative analysis.

DISCUSSION

The 25% fog mAP gain is the clearest evidence that the GAN is working. Fog attenuates image contrast exponentially(Kumar *et al.*, 2020) through Mie scattering; the GAN learns to invert that degradation and recover the edge structure detectors rely on. Ablation results make the loss design concrete: adversarial loss alone reached only 68% fog mAP, removing the perceptual term dropped it to 71%, and only the full three-term combination achieved 78.2%. The perceptual loss, computed in the relu3_3/relu4_3/relu5_3 activation space of VGG-16, captures textural detail that pixel-wise terms simply cannot see. The Depth Fusion Gate explains the 14.7-point near-range gain. One deliberate design choice was a soft exponential decay rather than a hard 20 m cutoff. A binary mask would create gradient discontinuities at the boundary and, more practically, would ignore objects at 21–30 m that are only seconds from the danger zone at highway speeds. The smooth decay handles that transition naturally. The speed cost is real: the GAN and MiDaS together add about 13 ms per frame, cutting throughput from 42 to 30 FPS. That is still above the 25 FPS threshold for safe deployment, and profiling suggests TensorRT FP16 quantization would push past 50 FPS. Model pruning and knowledge distillation offer

a further path to sub-20 ms latency while retaining over 95% of accuracy.

Compared to the other baselines on the same KITTI splits, the tradeoffs are clear. DeepRange (Cui *et al*., 2022) is fast (61 FPS) and accurate in sunshine (96.92% mAP), but it has no weather handling—our tests showed it collapsing to around 45% mAP in dense fog. LM-CNN-SVM (Uçar *et al*., 2016) reaches 89.8% in clear conditions but runs at only 8–12 FPS, which rules it out for real deployment. Bayesian networks with MC-Dropout (Feng *et al*., 2018; Hall *et al*., 2020) add useful uncertainty estimates but at 3–5× the compute cost, cutting throughput to around 10 FPS; crucially, quantifying how uncertain you are about a corrupted image does not fix the corruption. Our framework is the only one in this comparison that addresses weather robustness, near-range prioritization, and real-time throughput together. That said, we do not claim it is complete. Preliminary experiments with simultaneous fog and heavy rain showed an 8–12% mAP drop compared to single-weather conditions. The model was also trained on KITTI daytime images only, so desert dust storms, Arctic whiteout, tropical monsoon, and nighttime glare are all out of distribution. Handling those scenarios would likely require weather-type classification plus adaptive restoration heads conditioned on detected conditions—something we plan to explore next.

CONCLUSION

We presented a weather-robust object detection framework that jointly addresses two practical limitations of existing systems: performance degradation under adverse weather and uniform treatment of all detection ranges. By integrating a GAN-based image restoration module with a depth-aware feature gating mechanism into a YOLOv4 pipeline, the proposed approach achieves 78.2% mAP under dense fog—a 25% gain over the unmodified baseline—and 87.5% mAP in the safety-critical 0–20 m zone, while sustaining real-time throughput of 30 FPS on a single GPU. These results compare favorably against LM-CNN-SVM, Deep Range, and Bayesian detection frameworks, none of which simultaneously handles weather robustness, range prioritization, and real-time operation. The current work is limited to single-camera, daytime KITTI conditions; compound weather scenarios and nighttime environments remain open challenges. Future work will focus on multi-sensor fusion with LiDAR and radar to strengthen depth estimates, and on extending the weather augmentation pipeline to cover a broader range of real-world degradation conditions.

REFERENCES

Chen, Y., Li, W., Sakaridis, C., Dai, D., and Van Gool, L. (2018). Multi-task learning for dangerous object detection in autonomous driving. *Information Sciences*, 432: 559–571.

Cui, J., Zhong, Z., Liu, S., Yu, B. and Jia, J. (2022). ResLT: Residual learning for long-tailed recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45: 3695–3706.

Feng, D., Rosenbaum, L. and Dietmayer, K. (2018). Towards safe autonomous driving: Capture uncertainty in the deep neural network for LiDAR 3D vehicle detection. In Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC), Pp. 3266–3273. IEEE.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38: 142–158.

Hall, D., Dayoub, F., Skinner, J., Zhang, H., Miller, D., Corke, P., Carneiro, G., Tresson, A., and Sünderhauf, N. (2020). Probabilistic object detection: Definition and evaluation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Pp. 1031–1040.

Han, Y., Cai, Y., Cao, Y. and Xu, X. (2020). Wasserstein loss-based deep object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern

Recognition Workshops, Pp. 998–999.

Janai, J., Güney, F., Behl, A. and Geiger, A. (2020). Computer vision for autonomous vehicles: Problems, datasets and state of the art. Foundations and Trends® in Computer Graphics and Vision, 12(1–3): 1–308.

Kumar, P., Singh, R., Sharma, A., Singh, G., Kumar, D. and Singh, A. K. (2021). Comparison of different interpolation techniques for mean areal rainfall estimation of Uttarakhand using geographical information system. Journal of the New England Water Works Association, 135(3): 43–50.

Li, Y., Wang, H., Dang, L. M., Nguyen, T. N., Han, D., Lee, A., Jang, I. and Moon, H. (2020). A deep learning-based hybrid framework for object detection and recognition in autonomous driving. *IEEE Access*, 8: 194228–194239.

Lin, W. C., Tsai, C.-F., Chen, Z.-Y. and Ke, S.-W. (2016). Keypoint selection for efficient bag-of-words feature generation and effective image classification. *Information Sciences*, 329: 33–51.

Meyer, G. P., Laddha, A., Kee, E., Vallespi-Gonzalez, C. and Wellington, C. K. (2019). LaserNet: An efficient probabilistic 3D object detector for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Pp. 12677–12686.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Pp. 779–788.

Ucar, A., Demir, Y., and Güzeli_, C. (2016). Object recognition and detection with deep learning for autonomous driving applications. Simulation: *Transactions of the Society for Modeling and Simulation International*, 93(9): 759–769.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V. and Darrell, T. (2020). BDD100K: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Pp. 2636–2645.

Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Pp. 818–833.

Zhang, S., Wen, L., Bian, X., Lei, Z. and Li, S. Z. (2018). Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Pp. 4203–4212.

Received: February 19, 2026
Accepted: March 18, 2026