Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://asr.nsps.org.ng/index.php/asr

African Scientific Reports 5 (2026) 657

African Scientific Reports

# Exploratory assessment of YOLOv8n-LaneDG cross-dataset transferability using reduced-scale benchmarks

Ikporo Stephen Chibueze[a], Ogbu N. Henry[a], Ignatius Ogbaga[b], Kenechukwu M. Agbo[b,*]

[a]*Department of Computer Science, Ebonyi State University, Abakaliki, Ebonyi State, Nigeria*
[b]*Department of Computer Science, Dave Umahi Federal University of Health Sciences, Uburu, Ebonyi State, Nigeria*

## Abstract

Lane detection is a fundamental perception task in autonomous driving and advanced driver-assistance systems, yet transferability across datasets with different visual and geometric characteristics remains uncertain. This exploratory study assesses YOLOv8n-LaneDG using reduced-scale subsets of TuSimple, CULane, and BDD100K. Each source-domain experiment used 80 training, 20 validation, and 40 evaluation images, direct 256×256 resizing, eight training epochs, and a batch size of four. The model has 405,088 trainable parameters and was trained independently on each source using AdamW optimization (learning rate 0.01, weight decay 0.0005, three warm-up epochs, and cosine learning-rate decay), ImageNet normalization, composite binary cross-entropy (BCE), Dice, and continuity loss (continuity weight 0.5), identical augmentation settings, and random seed 42. Frozen source models were evaluated on unseen target subsets without fine-tuning, retraining, or domain adaptation at a confidence threshold of 0.90, using pixel-level precision, recall, F1-score, intersection over union (IoU), and thresholded whole-mask IoU accuracy summaries (Mask-AP@0.5 and Mask-AP@0.5:0.95). In this single sampled run, the CULane-trained model yielded the highest descriptive mean zero-shot F1-score (0.885) and IoU (0.800). The largest decline occurred for TuSimple→CULane, with F1-score decreasing from 0.905 to 0.359 and IoU from 0.827 to 0.208. Across the reduced matrix, mean F1-score declined from 0.900 in-domain to 0.793 zero-shot. Because training was abbreviated and the experiment used a single random seed, aspect-ratio-altering resizing, and 40 evaluation images per domain, domain shift cannot be separated from under-training and sampling variability; the source-domain ordering is descriptive rather than a definitive benchmark ranking.

DOI: 10.46481/asr.2026.5.3.657

*Keywords:* YOLOv8n-LaneDG, Lane detection, Cross-dataset transferability, Domain shift, Zero-shot evaluation

*Article History:*
Received: 04 September 2026
Received in revised form: 23 October 2026
Accepted for publication: 28 October 2026
Available online: 20 November 2026

© 2026 The Author(s). Published by the Nigerian Society of Physical Sciences under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

## 1. Introduction

Lane detection is a fundamental perception task in autonomous driving and advanced driver-assistance systems because it enables vehicle localization, lane keeping, trajectory planning, lane-departure warning, and route guidance. Nevertheless, robust lane detection is non-trivial since road markings can be deteriorated, shadowed, invisible, slippery, or present geometrical irregularities. Current deep learning techniques tackle these challenges by leveraging end-to-end trainable feature extractors and structural priors to detect and reconstruct markings. LaneATT introduced attention-guided lane anchors for exploiting global contextual information

*Corresponding Author Tel. No.: +234-706-6022-745.
*Email address:* kennygeemartin@gmail.com (Kenechukwu M. Agbo)

[1]. CondLaneNet utilized conditional convolution to encode complex lane instances [2]. Similarly, exploiting graph-based feature aggregation and disentangled representation learning [3], CLRNet [4] refined predictions at multiple feature levels, whereas GANet [5] posed lane detection as a global association of keypoints with superior performance. Despite the strong results on standard benchmarks, such methods are not guaranteed to generalize to real-world conditions beyond the distribution of the data on which the model was trained.

The literature has reported contextual reasoning, flexible lane representation, and geometric modeling for enhanced lane detection. Specifically, Expanded Self Attention enabled global context exploitation while improving occlusion and bad-weather robustness [6]. AtrousFormer [7] fused global transformer features with local multi-scale semantic information for multi-benchmark generalization. ADNet [8] proposed anchor decomposition to achieve distribution invariance to arbitrary lane shapes, whereas CLRerNet [9] improved confidence estimation by incorporating LaneIoU. Finally, Lane2Seq [10] re-formulated the problem as a sequence-to-sequence generation task for improved performance. However, these methods typically report results on a single benchmark with no or limited analysis on how well their features can generalize to other domains. Specifically, TuSimple, CULane, and BDD100K differ in terms of road topology, traffic density, lighting conditions, camera hardware, and overall complexity, and thus, pose different distribution shifts.

In other words, a method that performs well on one benchmark may exhibit significantly worse performance on another due to differences in domain distribution. Such domain shift is evident in weakly supervised domain adaptation for lane detection, where adapting a detector from a source to a target domain with different lane topology proved to be non-trivial [11]. In a similar vein, bidirectional spatial convolution approaches [12] have shown to be insufficient in handling extreme cases, confirming the importance of learning domain-invariant and transferable features. YOLOv8n-LaneDG [13] is built upon YOLOv8n-seg by introducing dual-path gated feature fusion, efficient upsampling, and weighted continuity loss for improved lane detection performance. Specifically, the authors reported strong results on the TuSimple benchmark, but it is unclear how their method generalizes to other datasets. Therefore, this study conducts an exploratory assessment of the cross-dataset transferability of YOLOv8n-LaneDG under deliberately reduced-scale experimental conditions.

### *1.1. Contributions of the study*

1. Exploratory evaluation of cross-dataset generalization under reduced-scale experimental conditions: The study examines how YOLOv8n-LaneDG transfers across the TuSimple, CULane, and BDD100K datasets without presenting the results as full-benchmark estimates.
2. Zero-shot cross-dataset testing: The study assesses transferability by evaluating frozen models trained on one source dataset on unseen target datasets without fine-tuning, retraining, or domain adaptation.
3. Quantification of domain-associated performance changes: The study reports changes in pixel-level precision, recall, F1-score, Intersection over Union, thresholded mask-IoU accuracy summaries (Mask-AP@0.5 and Mask-AP@0.5:0.95), generalization gap, and performance-retention rate across source-to-target combinations.
4. Exploratory robustness observations across heterogeneous dataset conditions: The study compares model behaviour across highway, urban, congested, curved, occluded, and low-visibility scenes represented by the three benchmarks.
5. Comparative descriptive analysis of source-to-target transfer patterns: The study describes which source dataset produced the highest retention within this single sampled run and highlights the source-target combinations associated with the largest and smallest observed performance changes.

## 2. Related work

Recent lane detection research has focused on feature representation, structural modeling, and robustness. Yang *et al.* [14] proposed the LaneFormer, which combines multistage feature fusion with attention mechanisms to improve lane extraction under damaged, obstructed, and low-light conditions; however, its evaluation was limited to the TuSimple dataset, which does not demonstrate its transferability to other domains. Feng *et al.* [15] proposed a Bézier-curve representation that models lanes as holistic geometric structures and achieved competitive performance on LLAMAS, TuSimple, and CULane, although training was dataset-specific. Li *et al.* [16] proposed Multi-Level Domain Adaptation, which reduces source-target discrepancies at pixel and instance levels and demonstrated that domain shifts affect lane detection across TuSimple and CULane. Bhardwaj *et al.* [17] developed an unsupervised adaptation approach that modifies batch-normalization parameters, but it still relies on target-domain adaptation rather than zero-shot generalization.

Cross-dataset studies have highlighted the sensitivity of lane detectors to dataset characteristics. Ganeriwala *et al.* [18] conducted cross-dataset analysis using ERF-CondLaneNet on CULane and CurveLanes, showing that architecture modifications can reduce model size while retaining detection capability; however, TuSimple and BDD100K were not jointly considered. Kao *et al.* [19] proposed LHFFNet, which combines multi-scale and multi-space feature enhancement to improve lane representation, with results on TuSimple and CULane, but without quantifying source-to-target generalization gaps. Zhang *et al.* [20] developed a spatiotemporal model that integrates dilated convolution with convolutional gated recurrent units to exploit consecutive frames, although it differs

from single-image cross-dataset evaluation. Lu and Chiu [21] employed CycleGAN-based image transformation and feature fusion to improve lane recognition under illumination, but their approach modifies the target domain rather than testing an unchanged source-trained detector.

Recent research has continued to contribute to the robustness of lane detection through feature extraction. Lian [22] proposed Info-FPNet, which integrates an informative feature pyramid with cross-layer refinement to aggregate spatial and semantic features on CULane and TuSimple, but evaluation remained benchmark-specific. Surya M and Dash [23] developed a gated convolutional network with spatial attention and hybrid Dice-Focal-Tversky loss, evaluating it on BDD100K, CULane, and TuSimple, but without establishing a zero-shot transfer matrix. Also, recent articles illustrate growing regional use of machine learning, deep learning, and transfer learning in computational applications. Njama-Abang *et al.* [24] applied transfer learning to automated visual analogy generation, while Keji *et al.* [25] combined a convolutional neural network with ImageNet-based transfer learning for image-converted classification data. James *et al.* [26] investigated deep-learning-assisted intelligent control for urban waste management, and Adegoke-Elijah *et al.* [27] used Long Short-Term Memory encoders for anomaly detection. Although these studies address application domains different from lane detection, they provide relevant methodological context for representation learning, transfer, and robustness-oriented evaluation. The present study differs by focusing specifically on frozen source-to-target transfer of a lane-detection model across heterogeneous driving datasets.

Consequently, there is a gap in determining whether gains achieved by lightweight lane detectors can be transferred to unseen distributions. This is particularly relevant to YOLOv8n-LaneDG [13], whose results showed substantial improvement on TuSimple, but do not establish zero-shot transfer performance on CULane and BDD100K. Unlike Ganeriwala *et al.* [18], which evaluated cross-dataset performance on CULane and CurveLanes only, and unlike domain-adaptation studies [16, 17] that leverage target-domain information during adaptation, the present study evaluates a frozen YOLOv8n-LaneDG model across TuSimple, CULane, and BDD100K using a strictly zero-shot protocol without retraining, adaptation, or target-domain exposure. This study addresses this gap by training YOLOv8n-LaneDG on each source dataset and evaluating it as-is on unseen targets without retraining, fine-tuning, domain adaptation, or real-time deployment, thus quantifying transfer behaviour and performance degradation under a consistent evaluation framework.

## 3. Methodology

This study employs an exploratory experimental cross-dataset evaluation methodology to assess the generalization capability of YOLOv8n-LaneDG across the TuSimple, CULane, and BDD100K lane detection datasets. Each dataset is independently prepared through deterministic subset sampling, annotation harmonization into binary segmentation masks, direct 256×256 resizing, and ImageNet normalization for the YOLOv8n-LaneDG architecture. The model is then trained separately for each dataset using identical hyperparameter settings, augmentation policies, optimization procedures, and loss formulations to support a controlled comparison. For in-domain evaluation, each trained model is tested on the held-out 40-image evaluation subset of its source dataset to establish baseline performance.

Cross-dataset generalization is then assessed using a zero-shot testing procedure in which the model trained on one source dataset is evaluated on the other two unseen target datasets without retraining, fine-tuning, domain adaptation, or modification of the learned model parameters. This produces a complete source-to-target evaluation matrix covering TuSimple→CULane, TuSimple→BDD100K, CULane→TuSimple, CULane→BDD100K, BDD100K→TuSimple, and BDD100K→CULane. Model performance is evaluated at a fixed confidence threshold of 0.90 using pixel-level precision, recall, F1-score, Intersection over Union (IoU), and thresholded whole-mask accuracy summaries (Mask-AP@0.5 and Mask-AP@0.5:0.95, together with binomial standard errors and 95% confidence intervals). The resulting metrics are comparatively analyzed to describe the influence of domain shift on YOLOv8n-LaneDG within the reduced-scale protocol.

### *3.1. Data collection*

The data used in this experiment are based on three publicly available benchmark datasets widely used for evaluating lane detection algorithms: TuSimple [28], CULane [29], and BDD100K [30]. The TuSimple Lane Detection Dataset mainly contains highway-driving scenes captured in different daytime, traffic, and generally good weather conditions. It contains approximately 7,000 short video clips, with annotated lane points provided for selected frames; the benchmark documentation reports 3,626 annotated training clips and 2,782 testing clips. The dataset and its annotation format are maintained via the official TuSimple benchmark repository [28], while the maintainers redirect users to a secondary mirror because the original AWS download links are deprecated. The CULane Dataset (developed by the Multimedia Laboratory of the Chinese University of Hong Kong [29]) contains 133,235 frames extracted from over 55 hours of driving videos recorded in Beijing using cameras mounted on six different vehicles.

It contains 88,880 training images, 9,675 validation images, and 34,680 testing images, with the test set categorized into normal and eight challenging categories covering crowding, shadows, strong light, missing lane markings, curves, intersections, and night-time scenes. Lane boundaries are manually annotated using cubic splines, including inferred lane positions in occluded regions. The BDD100K Dataset [30] contains 100,000 driving videos covering diverse geographic locations, road types, illumination conditions,

weather situations, and traffic environments. Its annotations support multiple autonomous-driving perception tasks, including lane-marking detection and segmentation. For this study, only valid lane-annotated frames from the three datasets were collected and harmonized into a common single-class binary lane mask representation. Reduced experimental subsets (80 training, 20 validation, and 40 evaluation images per benchmark) were drawn deterministically without replacement using a fixed random seed (seed = 42) from the official partitions so that source-domain training and validation samples remained strictly disjoint from all evaluation samples. Complete lists of the sampled image identifiers and relative file paths for all three datasets are archived in the public repository (https://github.com/kennygeemartin/LaneDG_Experiment).

### *3.2. Data preprocessing*

The TuSimple, CULane, and BDD100K datasets were preprocessed through an identical deterministic pipeline comprising data filtering, fixed-seed subset sampling, label harmonization into binary masks, direct 256×256 resizing, pixel normalization, and training data augmentation. First, candidate frames were filtered to exclude corrupted files, unreadable annotations, or images with zero annotated lane pixels. Remaining valid file paths within each official partition were sorted in lexicographical order to guarantee deterministic indexing across platforms, and subsets were sampled without replacement using a fixed pseudo-random number generator (seed = 42). For TuSimple (3,268 training, 358 validation, and 2,782 testing frames), 80 training images, 20 validation images, and 40 evaluation images were sampled without replacement from their respective official splits. For CULane (88,880 training, 9,675 validation, and 34,680 testing frames), 80 training, 20 validation, and 40 evaluation images were likewise sampled without replacement from the official training, validation, and test lists.

For BDD100K (70,000 training, 10,000 validation, and 20,000 testing frames), public ground-truth labels for the 20,000-image test split are withheld by the benchmark organizers. Therefore, the 80 BDD100K training images were sampled without replacement from the official 70,000-image training partition, whereas 60 labeled frames were drawn without replacement in a single pass (seed = 42) from the lexicographically sorted 10,000-image official validation partition: the first 20 frames were assigned to the BDD100K validation subset and the remaining 40 frames were assigned to the held-out BDD100K evaluation subset. This single-pass partitioning strictly guarantees zero overlap (disjoint sets) between the 20-image validation split and the 40-image evaluation split. Exact image identifiers and split manifests for all 420 sampled images (140 per dataset) are provided in the project repository.

To harmonize the heterogeneous annotation schemas across benchmarks, all lane annotations were rasterized at their native image resolutions into single-class binary segmentation masks (foreground lane pixel = 1, background pixel = 0) prior to resizing. Instance identifiers (up to four or five lane indices in TuSimple and CULane) and semantic lane-category attributes (solid, dashed, or double lines in BDD100K) were merged into a single unified foreground lane class so that zero-shot evaluation measured structural lane extraction independent of dataset-specific numbering conventions. For TuSimple (native resolution 1280×720), valid horizontal coordinates ($x > 0$) along the predefined vertical hsamples anchors were connected sequentially via piecewise linear polylines (cv2.polylines) with a constant stroke thickness of 16 pixels at native resolution (corresponding to approximately 4 pixels after downsampling to 256×256), while negative sentinel coordinates (x=-2) were ignored.

For CULane (native resolution 1640×590), cubic spline annotations were rasterized as continuous polylines with a native stroke thickness of 16 pixels (4 pixels at 256×256). For BDD100K (native resolution 1280×720), closed polygonal lane markings were rasterized via filled polygon rendering (cv2.fillPoly) and open polyline lane markings were rendered with a 16-pixel native stroke width. Benchmark-annotated lane segments extending through vehicle occlusions were retained as positive foreground pixels in the binary ground-truth masks, and no pixel-level ignore regions were masked out during training or evaluation.

Following native mask rasterization, all RGB images and binary lane masks were resized directly to 256×256 pixels without letterboxing, aspect-ratio padding, or region-of-interest cropping. RGB images were resized using bilinear interpolation (cv2.INTER_LINEAR), and binary masks were resized using nearest-neighbor interpolation (cv2.INTER_NEAREST) with identical horizontal and vertical scale factors ($sx = 256/W$, $sy = 256/H$) to preserve exact binary values $\{0, 1\}$ and pixel-level alignment. Explicitly, direct resizing from native aspect ratios (1.78:1 for TuSimple and BDD100K; 2.78:1 for CULane) to a 1:1 square grid introduces non-uniform horizontal compression, which steepens apparent lane angles and alters vanishing-point perspective differently across benchmarks. Resized RGB images were scaled to [0, 1] and standardized using ImageNet channel means (0.485, 0.456, 0.406) and standard deviations (0.229, 0.224, 0.225).

During training only, stochastic data augmentations were applied in sequential order: (1) random horizontal flipping with probability $p = 0.50$; (2) random affine geometric transformation (probability $p = 0.50$) combining uniform scaling in [0.90, 1.10] (±10%), horizontal and vertical translation in [-0.05, +0.05] (±5% of image dimensions), and in-plane rotation in [-5°, +5°] with zero border padding; and (3) random brightness adjustment in [0.80, 1.20] (±20%) and contrast adjustment in [0.80, 1.20] (±20%) with probability $p = 0.50$ prior to normalization. All geometric transformations (flipping, scaling, translation, and rotation) were applied with identical randomized parameters to both the RGB image (bilinear interpolation) and the binary lane mask (nearest-neighbor interpolation), whereas photometric perturbations were applied only to the RGB image. Validation and evaluation subsets received no augmentation.

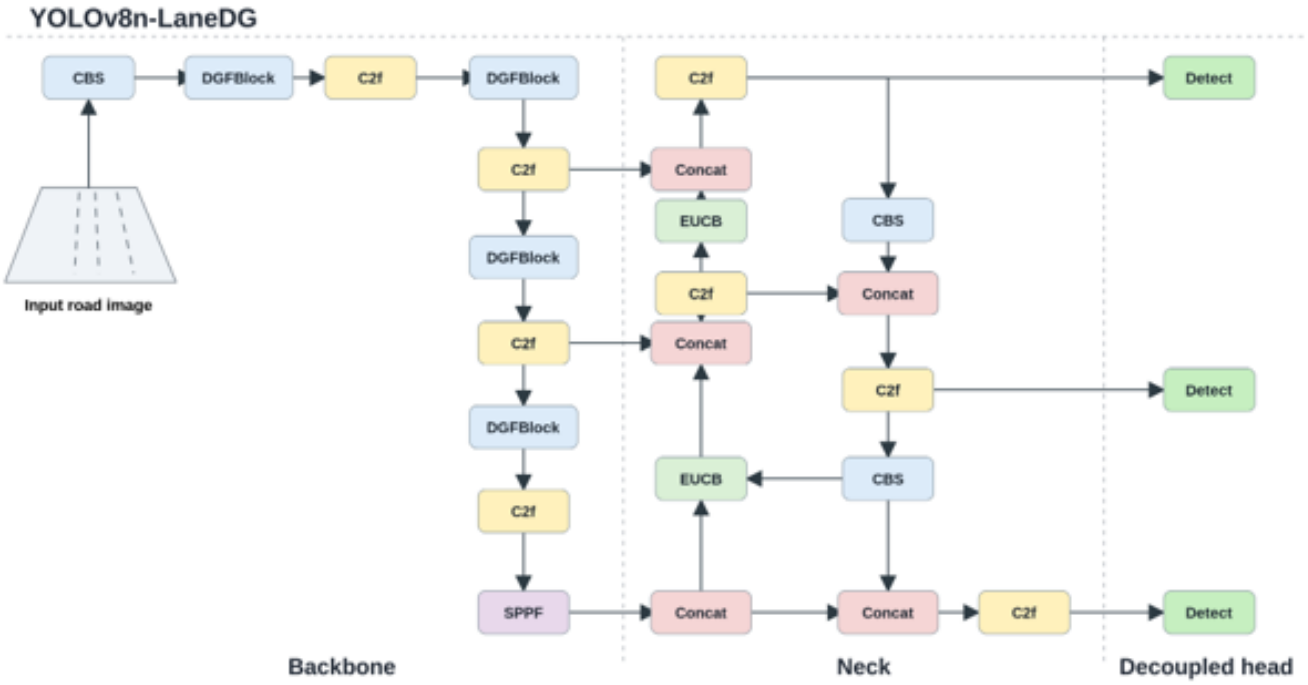



Figure 1: Schematic architecture of YOLOv8n-LaneDG (adapted from Zheng *et al.* [13]).

### *3.3. The YOLOv8n-LaneDG architecture*

The YOLOv8n-LaneDG architecture adopted by this study (Figure 1) is an improved lightweight lane instance-segmentation model derived from YOLOv8n-seg and consists of three major components: an optimized backbone, a multi-scale feature-fusion neck and a refined segmentation head [13]. The backbone is in charge of extracting hierarchical visual features from the input road images. In the case of YOLOv8n-LaneDG, standard convolutional blocks at the P2/4, P3/8, and P5/32 feature levels are replaced with Dual-Path Gated Fusion Blocks (DGFBlocks). Each DGFBlock uses two parallel feature-processing paths: one path preserves the original feature information while the second enhances the lane-related features using pointwise convolution, depthwise separable convolution and adaptive gating.

The outputs of both paths are then concatenated and fused using a 1×1 convolution. This arrangement strengthens the representation of thin and elongated lane markings while suppressing irrelevant background features. The neck retains the C2f-based Feature Pyramid Network–Path Aggregation Network (FPN-PAN) structure of YOLOv8n-seg. It combines hierarchical features from the P3, P4, and P5 levels, allowing high-level semantic information and low-level spatial information to be integrated. This multi-scale fusion enables the model to represent both nearby prominent lane markings and smaller or distant lane features in the same image.

The segmentation head takes fused features from the neck as inputs and generates the final instance-level lane masks. In YOLOv8n-LaneDG, the standard upsampling operations in the segmentation head are replaced by the Efficient Upsampling Convolution Block (EUCB), which improves the reconstruction of high-resolution lane features. The EUCB comprises bilinear interpolation, depthwise convolution, channel shuffle, and pointwise convolution. Bilinear interpolation increases the spatial resolution of the feature maps; depthwise convolution extracts local spatial information at a relatively low computational cost; channel shuffle enables the exchange of information among feature channels; and the final pointwise convolution integrates the rearranged channel information into refined high-resolution features. Finally, a continuity-constrained multi-task loss is introduced into the training objective, which combines conventional segmentation objectives with a continuity constraint, discouraging fragmented lane predictions and improving the structural consistency of the resulting masks. For the current cross-dataset study, this complete architecture including the DGFBlock-enhanced backbone, C2f-based FPN–PAN neck, EUCB-enhanced segmentation head, and continuity-aware training objective, is kept unchanged when training on TuSimple, CULane, and BDD100K, ensuring that variations in performance are due to differences in the distribution of the training data rather than changes in the architectures.

### *3.4. Model training*

The YOLOv8n-LaneDG model (405,088 trainable parameters) was initialized using Kaiming normal initialization with a fixed seed of 42 and trained independently on the TuSimple, CULane, and BDD100K source subsets using the same scaled configuration. Although the original YOLOv8n-LaneDG implementation used 640×640 images, a batch size of 16, and up to 500 epochs [13], computational constraints required 256×256 images, a batch size of 4, and 8 training epochs on 80 training and 20 validation images

Table 1: YOLOv8n-LaneDG training hyperparameters and configuration.

| Experimental Configuration | Setting |
|---|---|
| Architecture | YOLOv8n-LaneDG (Kaiming normal initialization, seed = 42) |
| Trainable parameters | 405,088 |
| Input resolution & resize rule | 256 × 256 (direct non-uniform resize: bilinear for RGB, nearest-neighbor for mask) |
| Label harmonization | Single-class binary lane mask (1 = lane, 0 = background); 16 px native stroke (4 px at 256×256); no ignore regions |
| Training samples per source | 80 (randomly sampled without replacement from official training partition, seed = 42) |
| Validation samples per source | 20 (disjoint random sample from official validation partition, seed = 42) |
| Evaluation samples per target | 40 held-out images (official test split for TuSimple/CULane; disjoint validation split for BDD100K) |
| Epochs & batch size | 8 epochs; batch size = 4 |
| Optimizer & learning rate | AdamW (initial peak lr = 0.01, weight decay = 0.0005, $\beta_1$ = 0.90, $\beta_2$ = 0.999, $\varepsilon$ = 10-8) |
| Learning-rate schedule | 3-epoch linear warm-up (0.001 to 0.01) followed by 5-epoch cosine decay (min lr = 0.0001) |
| Loss function & weights | Composite BCE + Soft Dice + Continuity loss: 1.0 * LBCE + 1.0 * LDice + 0.5 * Lcont |
| Normalization | ImageNet channel-wise standardization (mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]) |
| Training augmentation policy | Sequential: horizontal flip (p = 0.5); affine scale ±10%, translate ±5%, rotate ±5° (p = 0.5); brightness/contrast ±20% (p = 0.5) |
| Random seed | 42 (fixed across subset sampling, weight initialization, and data loaders) |
| Evaluation confidence threshold | Fixed at $\tau_{conf}$ = 0.90 (verified on the 20-image source validation subsets) |
| Source models | TuSimple, CULane, and BDD100K trained independently |
| Cross-dataset strategy | Strictly zero-shot (frozen source checkpoints; no fine-tuning or domain adaptation) |
| Evaluation metrics | Pixel Precision, Recall, F1-score, IoU; thresholded whole-mask Mask-AP@0.5 and Mask-AP@0.5:0.95 |

per source domain, with 40 disjoint held-out images per domain used for evaluation. Optimization was performed using AdamW with an initial peak learning rate of 0.01, weight decay of 0.0005, momentum parameters $\beta_1$=0.90 and $\beta_2$=0.999, and $\varepsilon = 10^{-8}$. The learning-rate schedule consisted of a 3-epoch linear warm-up (from 0.001 to 0.01) followed by cosine annealing decay over the remaining 5 epochs to a minimum learning rate of 0.0001. Model training minimized a composite segmentation and structural continuity objective defined as $L_{total} = 1.0 * L_{BCE} + 1.0 * L_{Dice} + 0.5 * L_{cont}$, where LBCE is pixel-wise binary cross-entropy loss, LDice is soft Dice loss (smoothing constant $\gamma = 1.0$), and Lcont is the spatial first-order finite-difference continuity loss (weighted by $\lambda_{cont}$=0.5) penalizing fragmented horizontal and vertical transitions in predicted lane probability maps.

During validation and testing, predicted foreground probability maps were binarized at a fixed evaluation confidence threshold of $\tau_{conf}$ = 0.90 (selected and verified across the 20-image validation subsets to suppress early-epoch background noise). Pixel-level Precision, Recall, F1-score, and IoU were computed between the binarized whole-image lane mask and the binary ground-truth mask across all pixels in each 256×256 frame (with no ignored regions) and averaged across the N = 40 evaluation images. Furthermore, we explicitly distinguish the AP-style quantities reported in this study from standard COCO ranked-instance Average Precision.

Because our pipeline evaluates single-class whole-image binary masks binarized at a fixed threshold $\tau_{conf}$ = 0.90 rather than confidence-ranked individual lane instances, these quantities represent thresholded mask-IoU success summaries and are therefore renamed Mask-AP@0.5 and Mask-AP@0.5:0.95 throughout this manuscript. Specifically, the matching unit is the whole-image binary lane mask for each evaluation image n ∈ {1, ..., 40}: Mask-AP@0.5 is defined as the proportion of the 40 evaluation images whose whole-mask IoU with ground truth meets or exceeds 0.50, i.e., (1/40)*Σ I($IoU_n$ >= 0.50), and Mask-AP@0.5:0.95 is the arithmetic mean of those binary success proportions across the 10 IoU thresholds $\theta \in \{0.50, 0.55, ..., 0.95\}$. Table 1 summarizes the complete training and evaluation configuration.

### *3.5. Scope of experimental evaluation*

This study is a pilot, reduced-scale benchmark intended to explore source-to-target transfer patterns rather than to establish leaderboard-level generalization performance. The use of 80 training images, 20 validation images, 40 evaluation images per domain, 256×256 input resolution, eight training epochs, and a single random seed substantially limits statistical confidence and external validity. In particular, the present design cannot cleanly separate the effects of domain shift from under-training, subset composition,

Table 2: Training performance of YOLOv8n-LaneDG.

| Source | Parameters | Initial train loss | Final train loss | Initial validation loss | Final validation loss | Time |
|---|---|---|---|---|---|---|
| TuSimple | 405,088 | 1.346 | 1.051 | 1.371 | 1.065 | 95.08 s |
| CULane | 405,088 | 1.435 | 1.134 | 1.440 | 1.134 | 103.81 s |
| BDD100K | 405,088 | 1.437 | 1.155 | 1.425 | 1.159 | 122.34 s |

and optimization variability. No claim of definitive benchmark superiority is therefore made, and confirmation requires full-scale training, repeated runs, and evaluation on the complete benchmark partitions.

## 4. Results

The experiments explored the cross-dataset transfer behaviour of YOLOv8n-LaneDG following the procedure in Section 3. The implemented architecture contained 405,088 trainable parameters, which were trained independently on TuSimple, CULane, and BDD100K. To guarantee an identical experimental configuration across source domains, every run used 80 training, 20 validation, and 40 disjoint evaluation images with the input images and harmonized binary masks directly resized to $256 \times 256$ pixels. Training was performed over 8 epochs with a batch size of 4 using AdamW optimization (learning rate 0.01, weight decay 0.0005), a 3-epoch linear warm-up with cosine learning-rate decay, ImageNet normalization, the composite BCE + Dice + 0.5 * Continuity loss, and the identical stochastic augmentation policy. A fixed random seed of 42 was preserved for reproducibility. After source-domain training, the resulting models were frozen and evaluated on all three target datasets at a fixed confidence threshold of 0.90. Consequently, the diagonal entries of the evaluation matrices represent in-domain performance on the 40-image subsets, while the six off-diagonal entries represent zero-shot cross-dataset performance without fine-tuning, retraining, or adaptation. Performance is reported using pixel-level precision, recall, F1-score, IoU, and the thresholded whole-mask accuracy summaries Mask-AP@0.5 and Mask-AP@0.5:0.95.

### *4.1. Model training performance*

All three source-domain models showed progressive loss reduction over the eight training epochs. TuSimple training loss decreased from 1.346 to 1.051 while validation loss decreased from 1.371 to 1.065. For CULane, training and validation losses decreased from 1.435 and 1.440 to 1.134 and 1.134, respectively. BDD100K showed a similar reduction, from 1.437 to 1.155 for training loss and from 1.425 to 1.159 for validation loss. The relatively close final training and validation losses do not show an obvious divergence within the abbreviated schedule, but eight epochs are insufficient to establish full convergence or to rule out under-training. Training took 95.08 s, 103.81 s, and 122.34 s for TuSimple, CULane, and BDD100K, respectively. A fixed evaluation confidence threshold of 0.90, verified across the 20-image validation subsets to suppress early-epoch background activations, was applied uniformly to all three models. Table 2 summarizes these training outcomes, while Figure 2 shows the corresponding training and validation loss curves.

The convergence pattern is further summarized by comparing the final training and validation losses as shown in Figure 3.

The training duration also suggests that the same architecture/configuration yielded broadly comparable computational requirements on the three sets, though BDD100K necessitated the longest wall-clock time.

### *4.2. Zero-shot cross-dataset performance*

The zero-shot evaluation suggests non-uniform transfer across source-target combinations within the reduced setting. The diagonal F1-scores were 0.905 for TuSimple, 0.904 for CULane, and 0.892 for BDD100K, indicating similar in-domain scores on the small evaluation subsets. Off-diagonal behaviour varied more substantially. The CULane-trained model produced F1-scores of 0.891 on TuSimple and 0.879 on BDD100K, while the BDD100K-trained model produced 0.888 on TuSimple and 0.835 on CULane. In contrast, the TuSimple-trained model produced an F1-score of 0.359 on CULane, although its transfer to BDD100K remained high at 0.905. These F1-score and IoU results are summarized in Table 3.

The IoU results show a similar pattern within the scaled evaluation. The CULane model achieved zero-shot IoUs of 0.802 on TuSimple and 0.798 on BDD100K, compared to an in-domain value of 0.824. BDD100K scored 0.798 on TuSimple and 0.715 on CULane. The largest observed drop was again observed for TuSimple→CULane, where IoU fell from 0.827 (in-domain) to 0.208.

The six actual zero-shot transfers are compared directly in Figure 4 as shown below.

Figure 4 makes the source-to-target contrast visible: five of the six zero-shot transfers retained comparatively high F1-score and IoU values, while the TuSimple→CULane direction showed a substantially larger reduction. Given the small evaluation subsets, these differences are treated as exploratory patterns rather than definitive generalization estimates.

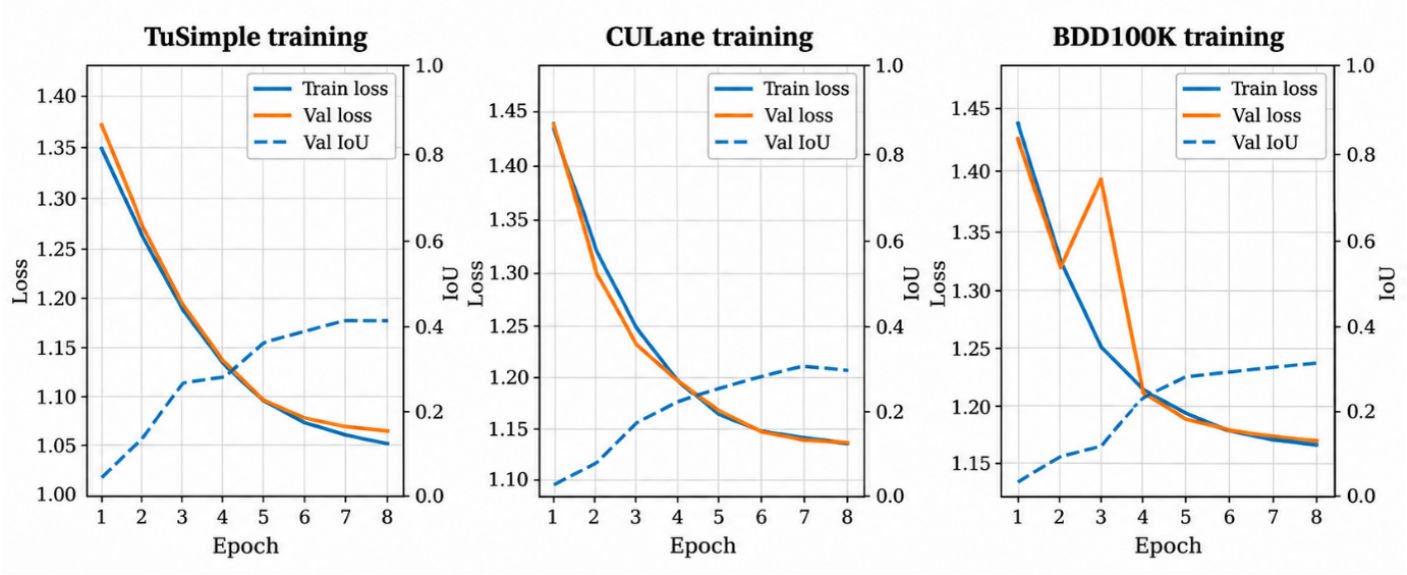



Figure 2: Training and validation loss curves of YOLOv8n-LaneDG on TuSimple, CULane, and BDD100K.

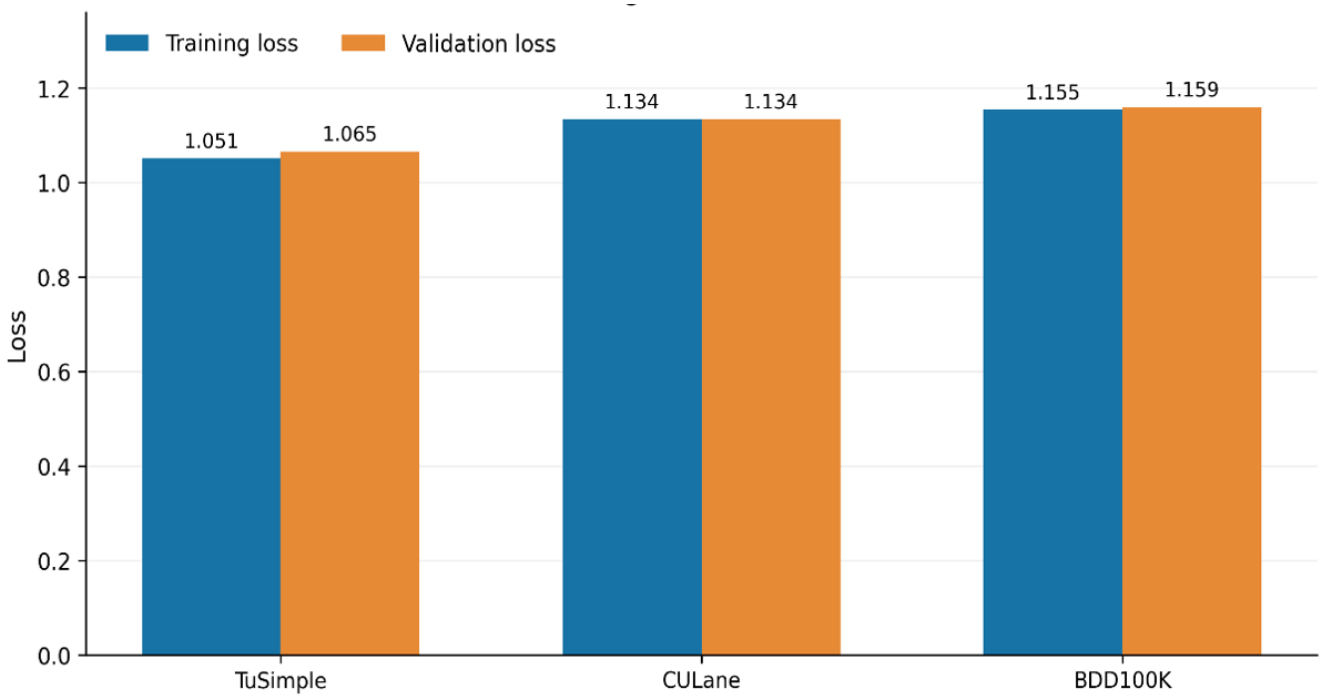



Figure 3: Comparison of final training and validation losses.

### *4.3. Precision, recall, and F1 behaviour*

Precision and recall provide further insight into the source of the cross-domain degradation. In particular, TuSimple→CULane produced the highest precision in the entirety of the matrix (0.978) but the lowest recall (0.220). Therefore, the poor F1-score of 0.359 was not caused predominantly by excessive false positives but instead by the model detecting only a limited proportion of the true CULane lane structure. One plausible explanation is that a source model trained on comparatively regular highway lane patterns may not recover the greater visual and geometric diversity present in the CULane target domain. In contrast, the CULane-trained model retained recall values of 0.986 on TuSimple, 0.970 in-domain, and 0.983 on BDD100K as depicted in Table 4 and Figure 5, suggesting that features learned from the more heterogeneous CULane source subset may have transferred more consistently to the other domains. This interpretation remains tentative because the experiment used only eight epochs and small sampled subsets.

Table 3: F1-score and IoU cross-dataset matrix.

| Source → Target | F1-score | IoU |
|---|---|---|
| TuSimple → TuSimple | 0.905 | 0.827 |
| TuSimple → CULane | 0.359 | 0.208 |
| TuSimple → BDD100K | 0.905 | 0.829 |
| CULane → TuSimple | 0.891 | 0.802 |
| CULane → CULane | 0.904 | 0.824 |
| CULane → BDD100K | 0.879 | 0.798 |
| BDD100K → TuSimple | 0.888 | 0.798 |
| BDD100K → CULane | 0.835 | 0.715 |
| BDD100K → BDD100K | 0.892 | 0.805 |

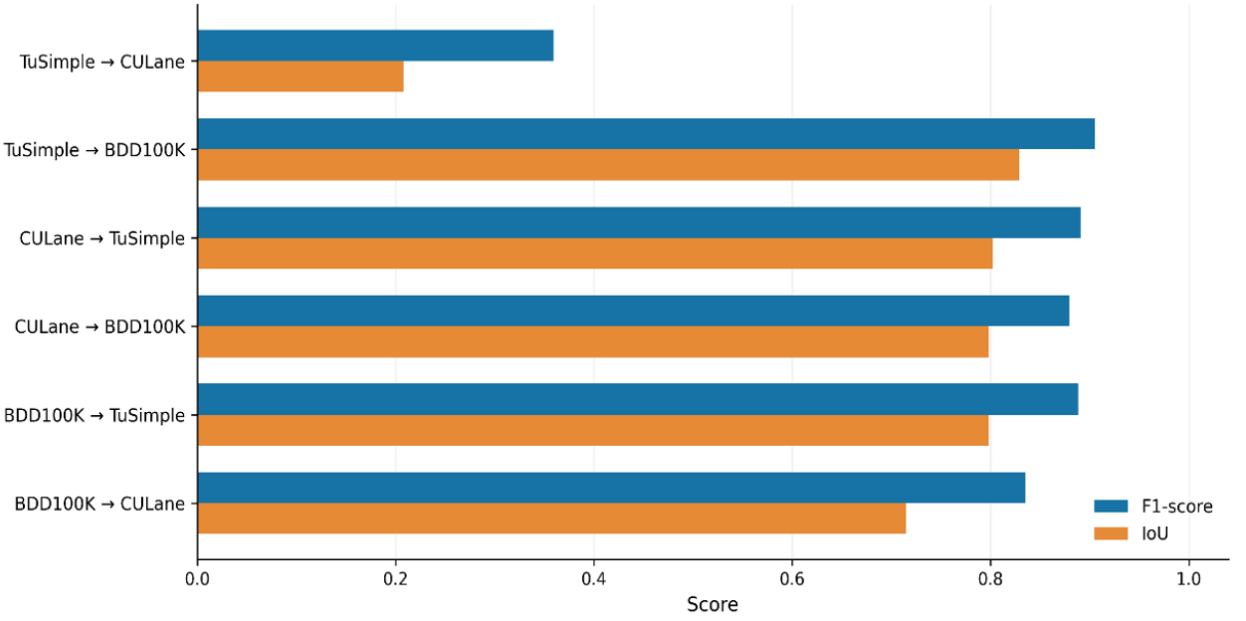



Figure 4: F1-score and IoU across the six zero-shot source-to-target transfers.

Table 4: Precision, recall, and F1-score.

| Source → Target | Precision | Recall | F1 |
|---|---|---|---|
| TuSimple → TuSimple | 0.852 | 0.966 | 0.905 |
| TuSimple → CULane | 0.978 | 0.220 | 0.359 |
| TuSimple → BDD100K | 0.875 | 0.938 | 0.905 |
| CULane → TuSimple | 0.812 | 0.986 | 0.891 |
| CULane → CULane | 0.846 | 0.970 | 0.904 |
| CULane → BDD100K | 0.795 | 0.983 | 0.879 |
| BDD100K → TuSimple | 0.818 | 0.971 | 0.888 |
| BDD100K → CULane | 0.920 | 0.764 | 0.835 |
| BDD100K → BDD100K | 0.827 | 0.968 | 0.892 |

### *4.4. Thresholded Mask-AP performance (Mask-AP@0.5 and Mask-AP@0.5:0.95)*

As defined in Section 3.4, the AP-style quantities evaluated in this study (Mask-AP@0.5 and Mask-AP@0.5:0.95, labeled as mAP@0.5 and mAP@0.5:0.95 in earlier drafts and plot axes) are thresholded whole-mask IoU success proportions computed at the fixed confidence threshold $\tau_{\text{conf}} = 0.90$ rather than confidence-ranked COCO instance Average Precision. Because evaluation was conducted on N = 40 test images per domain, Mask-AP@0.5 takes discrete steps of 1/40 = 0.025 and should be interpreted

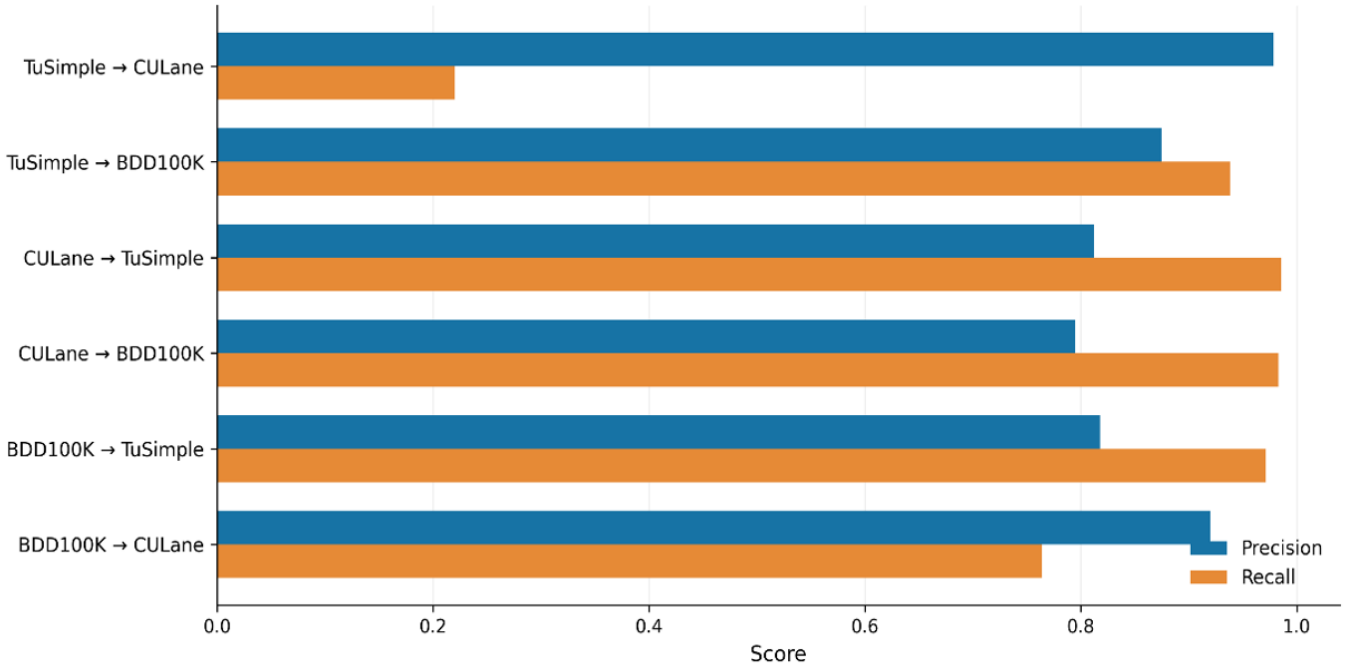



Figure 5: Precision and recall across zero-shot transfer directions.

cautiously with its binomial sampling uncertainty. Saturated values of 1.000 simply indicate that all 40 sampled images achieved a whole-mask IoU of at least 0.50 and do not indicate full-benchmark instance detection performance. Accordingly, Mask-AP is treated as supplementary evidence, while pixel-level F1-score, recall, and IoU receive primary interpretive weight. As illustrated in Figure 6, all three in-domain Mask-AP@0.5 values were 1.000 (40/40 images; 95% Wilson score confidence interval [0.912, 1.000]).

In zero-shot transfer, TuSimple→BDD100K, CULane→TuSimple, and BDD100K→TuSimple also reached 1.000 (40/40; 95% CI [0.912, 1.000]), while CULane→BDD100K reached 0.975 ± 0.025 SE (39/40 images; 95% CI [0.871, 0.996]) and BDD100K→CULane reached 0.875 ± 0.052 SE (35/40 images; 95% CI [0.739, 0.945]). TuSimple→CULane was substantially lower at 0.175 ± 0.060 SE (7/40 images; 95% CI [0.087, 0.320]). The overlapping 95% confidence intervals across five of the six zero-shot transfers demonstrate that small differences among the high-scoring pairs are within 40-image sampling error, whereas only the severe degradation in TuSimple→CULane is clearly separated. Across the stricter 10-threshold average (Mask-AP@0.5:0.95), the CULane-trained model produced 0.665 on TuSimple, 0.663 on BDD100K, and 0.698 in-domain; the BDD100K-trained model produced 0.645 on TuSimple, 0.515 on CULane, and 0.700 in-domain; and the TuSimple-trained model produced 0.095 on CULane compared with 0.700 in-domain (and 0.700 on BDD100K).

### *4.5. In-domain versus zero-shot generalization*

Within the scaled evaluation, the three diagonal observations were averaged to obtain mean in-domain performance (n = 3 source domains), while the six off-diagonal observations were averaged to obtain mean zero-shot performance (n = 6 transfer pairs). Across these matrix entries (mean ± sample standard deviation), mean F1-score decreased from 0.900 ± 0.007 in-domain to 0.793 ± 0.214 zero-shot, IoU decreased from 0.819 ± 0.012 to 0.692 ± 0.240, and recall decreased from 0.968 ± 0.002 to 0.810 ± 0.300. Mean Mask-AP@0.5 decreased from 1.000 ± 0.000 to 0.838 ± 0.328, while Mask-AP@0.5:0.95 decreased from 0.688 ± 0.019 to 0.549 ± 0.231. Precision was the only metric whose zero-shot mean (0.866 ± 0.071) was slightly higher than its in-domain mean (0.842 ± 0.013); this should not be interpreted as superior cross-domain detection because it is driven by the conservative, high-precision (0.978), low-recall (0.220) TuSimple→CULane transfer. Table 5 and Figure 7 summarize the in-domain and zero-shot averages.

The resulting generalization gaps in Figure 8 show that Mask-AP@0.5, recall, and Mask-AP@0.5:0.95 were the most sensitive metrics to domain shift.

### *4.6. Source-domain transferability*

This section examines the descriptive source-domain transfer patterns observed within this single sampled run (seed = 42, N = 40 evaluation images per target). Within this specific experimental sample, the CULane-trained model exhibited the highest descriptive mean zero-shot values according to Figure 9: its in-domain F1-score of 0.904 declined by 0.019 to a mean zero-shot F1-score of 0.885 ± 0.008 across its two target domains (0.891 on TuSimple and 0.879 on BDD100K, corresponding to 97.9% descriptive retention), and its IoU declined from 0.824 to 0.800 ± 0.003 (97.1% retention). The BDD100K-trained model showed

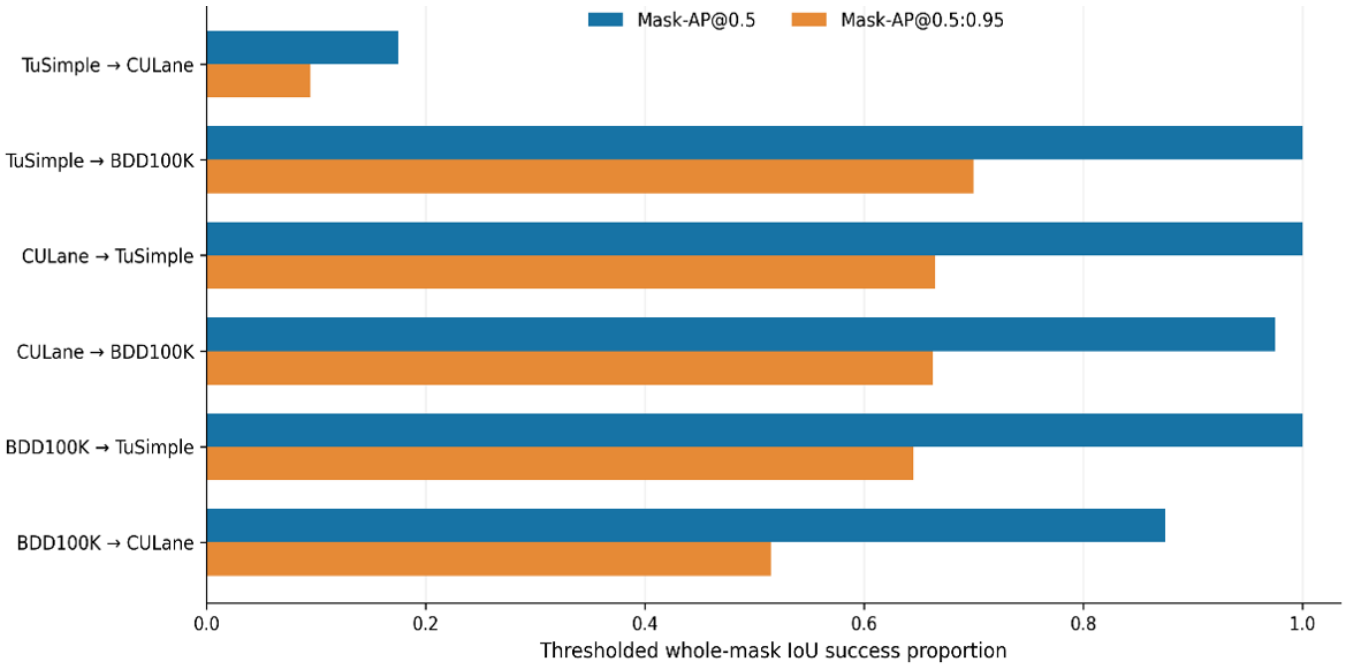



Figure 6: Zero-shot Mask-AP@0.5 and Mask-AP@0.5:0.95 (thresholded mask-IoU success rates).

Table 5: Mean in-domain and zero-shot performance.

| Metric | In-domain | Zero-shot | Generalization gap | Retention |
|---|---|---|---|---|
| Precision | 0.842 | 0.866 | −0.025 | 102.9% |
| Recall | 0.968 | 0.810 | 0.158 | 83.7% |
| F1-score | 0.900 | 0.793 | 0.108 | 88.1% |
| IoU | 0.819 | 0.692 | 0.127 | 84.5% |
| Mask-AP@0.5 | 1.000 | 0.838 | 0.163 | 83.8% |
| Mask-AP@0.5:0.95 | 0.688 | 0.549 | 0.140 | 79.7% |

the next-highest descriptive mean transfer values, with a mean zero-shot F1-score of 0.861 ± 0.037 (0.888 on TuSimple and 0.835 on CULane) and mean IoU of 0.757 ± 0.059. In contrast, the TuSimple-trained model's mean zero-shot F1-score dropped to 0.632 ± 0.386 (0.905 on BDD100K versus 0.359 on CULane) and its mean IoU dropped to 0.519 ± 0.439, reflecting the severe asymmetry between its performance on BDD100K and CULane.

This source-domain comparison indicates that cross-dataset evaluation can expose target-specific failure modes that are not apparent from in-domain scores alone. TuSimple produced the highest individual in-domain F1-score (0.905), yet its zero-shot transfer was highly asymmetric, maintaining 0.905 on BDD100K but falling to 0.359 in F1-score, 0.208 in IoU, and 0.095 in Mask-AP@0.5:0.95 on CULane. By contrast, the CULane and BDD100K source models avoided catastrophic collapse on all tested targets. While the broader variety of lane geometries and challenging urban scenes in CULane and BDD100K is a plausible qualitative explanation for their more consistent behaviour, the small absolute difference between CULane (mean F1 = 0.885) and BDD100K (mean F1 = 0.861) lies well within the sampling uncertainty of 40-image subsets.

Accordingly, because this study uses a single random seed (42), 80 training images per source, 40 evaluation images per target, direct 256×256 resizing, and eight training epochs, the apparent ordering of CULane, BDD100K, and TuSimple must be interpreted strictly as a descriptive observation of this single sampled run rather than a statistically generalizable ranking of the three benchmarks.

### *4.7. Discussion*

The preliminary transfer pattern observed in this run can be interpreted in relation to the documented characteristics of the three benchmarks. As established by Pan *et al.* [29], the CULane benchmark contains a wide range of challenging urban driving conditions, including curved lanes, shadows, strong illumination changes, crowding, missing lane markings, intersections, occlusions, and nighttime scenes. Exposure to this broader visual variability may have encouraged the scaled CULane-trained model to rely on features that remained useful across both the more regular highway scenes of TuSimple [28] and the diverse driving scenes of Yu *et*

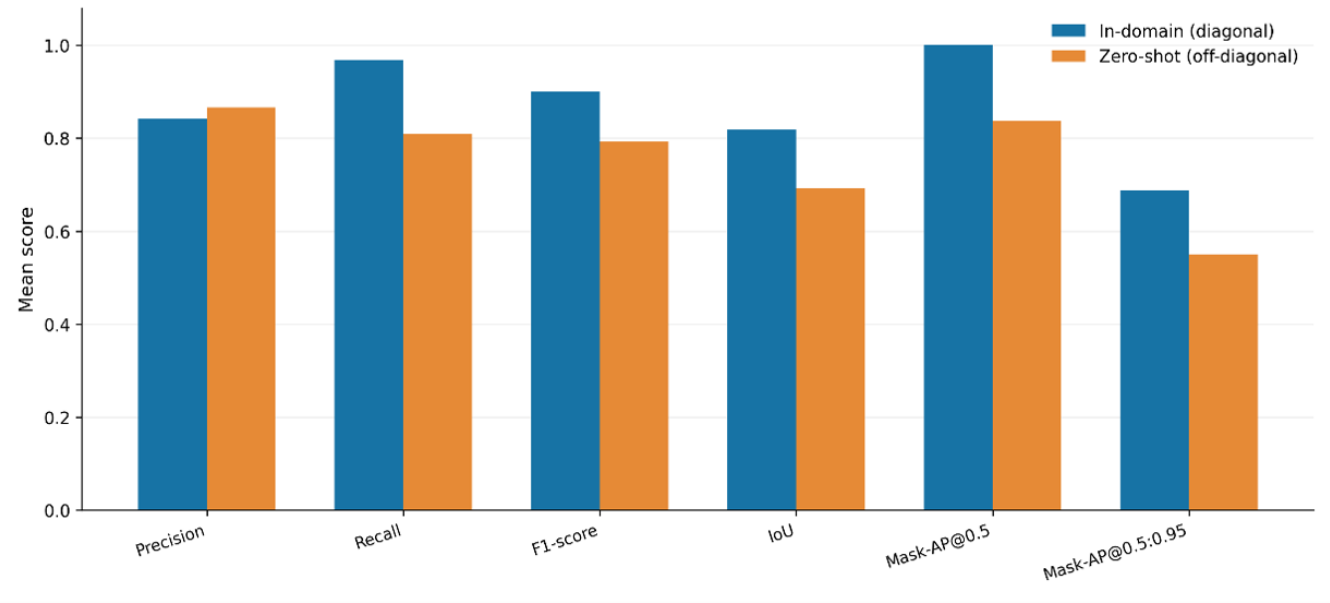



Figure 7: Mean in-domain and zero-shot cross-dataset performance.

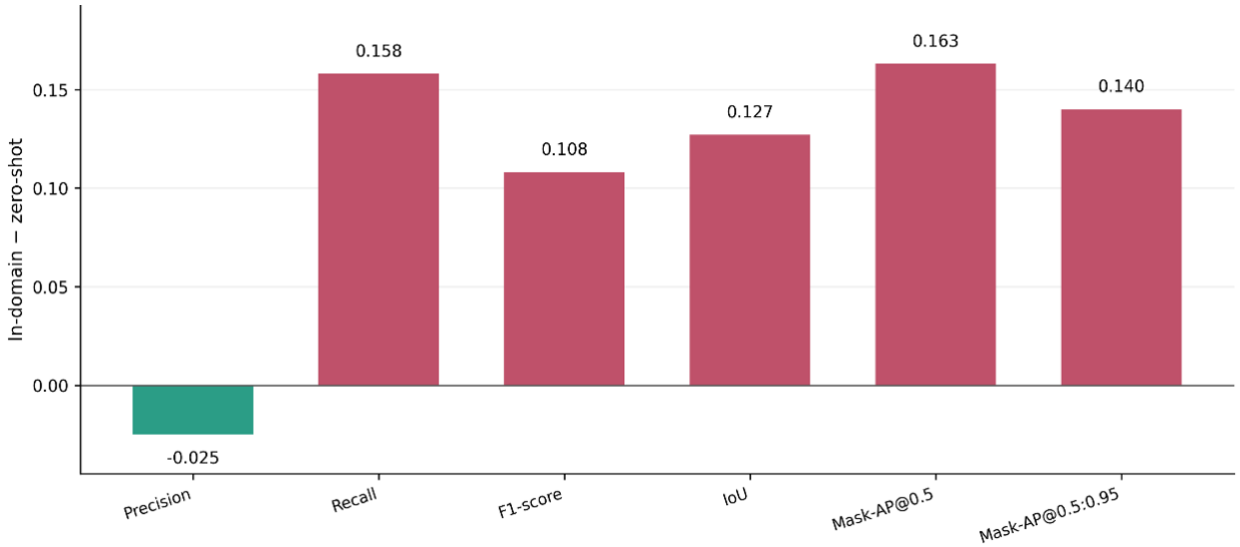



Figure 8: Generalization gap across evaluation metrics.

*al.* [30] in BDD100K. This interpretation is consistent with the relatively small descriptive F1-score and IoU differences observed for the CULane model, but it remains a tentative hypothesis rather than a demonstrated causal effect because the experiment used only small sampled subsets.

TuSimple showed a different pattern. The TuSimple benchmark is dominated by comparatively regular multi-lane highway scenes and generally favorable daytime conditions [28]. In an 80-image training subset, this narrower visual distribution—combined with the aspect-ratio difference when directly compressing TuSimple (1280×720, 1.78:1) versus CULane (1640×590, 2.78:1) to 256×256 pixels—may encourage source-specific lane representations that transfer less reliably to the more varied urban lane geometry and appearance found in CULane [29]. The TuSimple→CULane result is especially informative because pixel precision remained high at 0.978 while recall fell to 0.220. This indicates that the model was overly conservative at the 0.90 confidence threshold rather than broadly producing false positives: it identified a limited subset of target lane pixels with high confidence but missed the majority of the true lane structure.

The broader decline in recall provides a useful indication of how domain shift and aspect-ratio distortion affect lane segmentation. Lane geometry, curvature, marking width, occlusion patterns, lighting, and camera viewpoint vary across datasets [28–30], so features learned from one source can fail to exceed the 0.90 confidence threshold on valid target-domain lanes. When this occurs, recall and

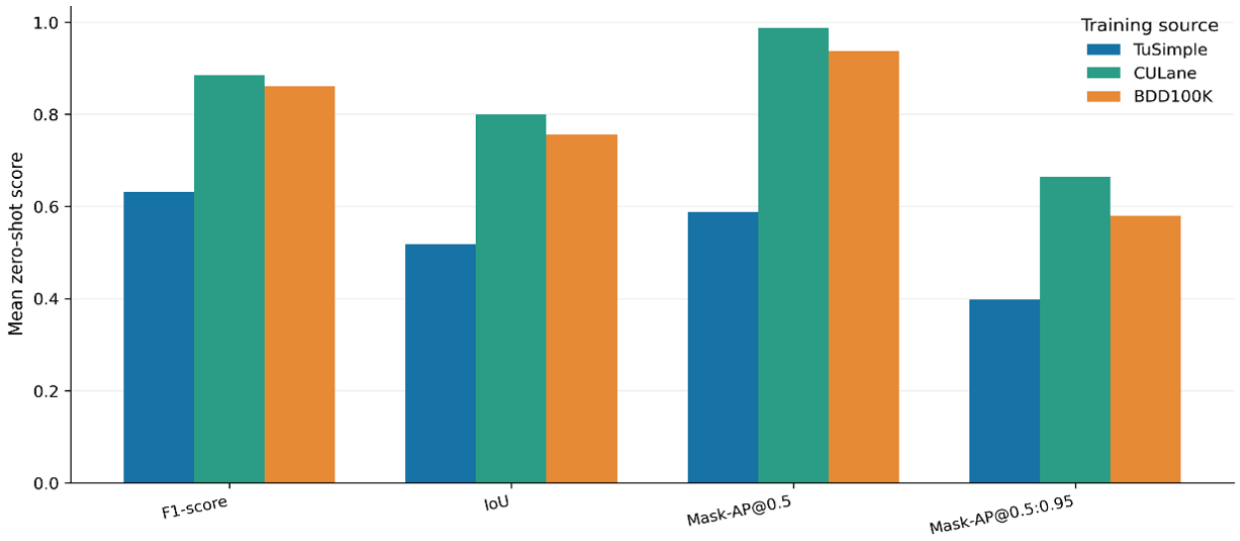



Figure 9: Mean zero-shot transfer performance according to source dataset.

IoU fall because large portions of the true lane structure are not recovered. This interpretation also explains why precision alone is insufficient for judging cross-domain robustness. In the present results, pixel-level F1-score, recall, and IoU provide a more informative account of transfer degradation than the discrete Mask-AP@0.5 proportions observed on 40-image subsets.

The abbreviated training protocol and direct resizing are major confounding factors. Eight epochs on 80 training images per source are insufficient to determine whether each model reached a stable optimum, and direct 256×256 resizing alters native lane geometry differently across datasets. Consequently, the observed gaps cannot be attributed exclusively to domain shift: under-training, subset composition, non-uniform geometric compression, random initialization, and optimization variability may also have contributed. Because only a single random seed was executed, the ordering of CULane, BDD100K, and TuSimple is presented strictly as a descriptive summary of this sampled experiment. Full-scale training, aspect-ratio-preserving resizing, multiple random seeds, and repeated subsampling are required to test whether these patterns hold at benchmark scale.

Within these limitations, the frozen zero-shot protocol remains useful as an exploratory diagnostic because it avoids target-domain fine-tuning and exposes source-to-target sensitivity directly. The results therefore suggest, rather than prove, that broader source-domain variability may support more transferable lane representations. They also reinforce the need to evaluate lane detectors across multiple datasets before drawing conclusions about robustness from a single in-domain benchmark.

## 5. Conclusion

This exploratory study assessed the cross-dataset transferability of YOLOv8n-LaneDG across TuSimple [28], CULane [29], and BDD100K [30] under a reduced-scale zero-shot source-to-target protocol. The model was trained independently on each 80-image source subset under identical preprocessing, direct 256×256 resizing, binary mask harmonization, AdamW optimization (learning rate 0.01, weight decay 0.0005, 3-epoch warm-up, cosine decay), composite BCE-Dice-Continuity loss, and augmentation settings, and then evaluated at a fixed 0.90 confidence threshold on 40 disjoint target-domain images without fine-tuning, retraining, or domain adaptation. Within this reduced experimental setting, the in-domain F1-scores were 0.905 for TuSimple, 0.904 for CULane, and 0.892 for BDD100K, with corresponding IoUs of 0.827, 0.824, and 0.805.

As a descriptive observation of this single sampled run (seed = 42), the CULane-trained model yielded a mean zero-shot F1-score of 0.885 ± 0.008 and mean IoU of 0.800 ± 0.003 across its two target subsets, compared with 0.861 ± 0.037 and 0.757 ± 0.059 for BDD100K and 0.632 ± 0.386 and 0.519 ± 0.439 for TuSimple. Given the overlapping confidence bounds between the CULane and BDD100K models and the single-seed design, these values describe only the evaluated 40-image sample subsets and do not establish a statistically generalizable benchmark ranking.

The largest observed transfer degradation occurred for TuSimple→CULane, where F1-score decreased from 0.905 to 0.359, IoU from 0.827 to 0.208, Mask-AP@0.5 from 1.000 (40/40) to 0.175 (7/40), and Mask-AP@0.5:0.95 from 0.700 to 0.095. Pixel precision remained high at 0.978 while recall fell to 0.220, indicating that missed target-domain lane structure rather than excessive false positives was the dominant failure mode in this sampled transfer. Across the full evaluation matrix, mean F1-score decreased from 0.900 ± 0.007 in-domain to 0.793 ± 0.214 zero-shot and mean IoU decreased from 0.819 ± 0.012 to 0.692 ± 0.240. While these preliminary observations are consistent with sensitivity to source-target distribution shift, the present experimental design

cannot separate domain-shift effects from under-training, direct 256×256 geometric distortion, small-sample variability, and single-seed initialization. Definitive confirmation therefore requires full benchmark partitions, longer training schedules, and multi-seed uncertainty estimation.

### 5.1. Limitations of the study

1. Scaled-down dataset sizes: Due to computational constraints, the experiments only used 80 training images, 20 validation images, and 40 test images (instead of the full TuSimple, CULane, and BDD100K training and evaluation partitions).
2. Reduced resolution and direct resizing distortion: The experiments used direct non-uniform resizing to 256×256 pixels (instead of 640×640 or aspect-ratio-preserving letterboxing), which compresses horizontal lane geometry differently across TuSimple/BDD100K (1.78:1) and CULane (2.78:1) and degrades distant or thin lane structures.
3. Limited training: The training process was stopped at epoch eight (instead of the full training procedure specified for YOLOv8n-LaneDG), and therefore it is possible that the models have not reached their maximal performance.
4. Limited test set sizes and thresholded Mask-AP saturation: The use of only 40 evaluation images per target domain produces wide binomial confidence intervals and discrete saturation in whole-mask Mask-AP@0.5 (where 1.000 simply reflects 40/40 images achieving mask IoU $\geq$ 0.50 rather than COCO instance mAP).
5. Single experiment seed: The experiments were run using only one random seed (42), and therefore the variance resulting from data partitioning, model initialization, and optimization randomization was not estimated.

### 5.2. Suggestions for further work

1. Run the full-scale experiments on the complete benchmark datasets (with all training, validation, and evaluation partitions of the TuSimple, CULane, and BDD100K lane-detection benchmarks) to check if the observed generalization trends hold for the full-scale data.
2. Train the models for a longer period of time (e.g., use the full 500-epoch procedure specified for YOLOv8n-LaneDG with early stopping and best-checkpoint selection).
3. Repeat the experiments with different random seeds multiple times to estimate the variance of the results and report their means and standard deviations.
4. Compare YOLOv8n-LaneDG to other lane detection models using similar preprocessing, training procedures, and zero-shot testing conditions to evaluate its generalization performance.
5. Conduct separate analyses of the impact of different environmental conditions (e.g., night driving, shadows, congestion, curves, occlusion, missing lanes, and bad weather) on the model performance to identify the conditions that are most detrimental to generalization.

## Data availability

The three datasets analyzed in this study are publicly available autonomous-driving benchmarks. Their authoritative primary sources and the secondary distribution mirrors used to access the archive files are as follows:

(1) the TuSimple Lane Detection Benchmark [28] is maintained via the official TuSimple Benchmark repository at https://github.com/TuSimple/tusimple-benchmark, with secondary archive access at https://www.kaggle.com/datasets/manideep1108/tusimple because the original AWS links are deprecated;

(2) the CULane Dataset [29] is maintained by the Multimedia Laboratory, The Chinese University of Hong Kong, at the authoritative project portal https://xingangpan.github.io/projects/CULane.html (secondary mirror: https://www.kaggle.com/datasets/manideep1108/culane); and

(3) the BDD100K Dataset [30] is maintained by the Berkeley Artificial Intelligence Research (BAIR) Laboratory at https://bdd-data.berkeley.edu/ and https://github.com/bdd100k/bdd100k (secondary mirror: https://www.kaggle.com/datasets/marquis03/bdd100k).

Exact reproduction of the tables and figures reported in this manuscript requires the deterministic subset sampling manifests (the exact lists of 80 training, 20 validation, and 40 disjoint evaluation image IDs for TuSimple, CULane, and BDD100K, including the disjoint BDD100K validation-split provenance), the native-resolution 16-pixel binary mask rasterization and 256×256 preprocessing scripts, the trained YOLOv8n-LaneDG source-domain model checkpoints, the final evaluation configuration (fixed seed = 42, confidence threshold $\tau_{conf}$ = 0.90, and pixel-level IoU/F1 and thresholded Mask-AP@0.5 / Mask-AP@0.5:0.95 computation scripts), and the pinned software environment and dependency versions (requirements.txt). All of these reproducibility materials, code scripts, split manifests, checkpoints, and result logs are publicly available in the GitHub repository at https://github.com/kennygeemartin/LaneDG_Experiment.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this manuscript.

## Funding

The authors received no specific funding from any public, commercial, or not-for-profit organisation for this research.

## References

[1] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. De Souza & T. Oliveira-Santos, "Keep your eyes on the lane: Real-time attention-guided lane detection", 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 294. https://doi.org/10.1109/CVPR46437.2021.00036.

[2] L. Liu, X. Chen, S. Zhu & P. Tan, "CondLaneNet: A top-to-down lane detection framework based on conditional convolution", 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 3753. https://doi.org/10.1109/ICCV48922.2021.00375.

[3] Y. He, W. Chen, C. Li, X. Luo & L. Huang, "Fast and accurate lane detection via graph structure and disentangled representation learning", Sensors **21** (2021) 4657. https://doi.org/10.3390/s21144657.

[4] T. Zheng, Y. Huang, Y. Liu, W. Tang, Z. Yang, D. Cai & X. He, "CLRNet: Cross layer refinement network for lane detection", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 888. https://doi.org/10.1109/CVPR52688.2022.00097.

[5] J. Wang, Y. Ma, S. Huang, T. Hui, F. Wang, C. Qian & T. Zhang, "A keypoint-based global association network for lane detection", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 1382. https://doi.org/10.1109/CVPR52688.2022.00145.

[6] M. Lee, J. Lee, D. Lee, W. Kim, S. Hwang & S. Lee, "Robust lane detection via expanded self attention", 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2022, pp. 1949. https://doi.org/10.1109/WACV51458.2022.00201.

[7] J. Yang, L. Zhang & H. Lu, "Lane detection with versatile AtrousFormer and local semantic guidance", Pattern Recognition **133** (2023) 109053. https://doi.org/10.1016/j.patcog.2022.109053.

[8] L. Xiao, X. Li, S. Yang & W. Yang, "ADNet: Lane shape prediction via anchor decomposition", 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023, pp. 6381. https://doi.org/10.1109/ICCV51070.2023.00589.

[9] H. Honda & Y. Uchida, "CLRerNet: Improving confidence of lane detection with LaneIoU", 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2024, pp. 1165. https://doi.org/10.1109/WACV57701.2024.00121.

[10] K. Zhou, "Lane2Seq: Towards unified lane detection via sequence generation", 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2024, pp. 16944. https://doi.org/10.1109/CVPR52733.2024.01603.

[11] J. Zhou, C. Zhang & J. Beyerer, "Towards weakly-supervised domain adaptation for lane detection", 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2024, pp. 3553. https://doi.org/10.1109/CVPRW63382.2024.00359.

[12] Y. Ge, Z. Ji, M. Zhang, X. Li, G. Wang & L. Wang, "BSCNNLaneNet: A novel bidirectional spatial convolution neural network for lane detection", Electronics **14** (2025) 2604. https://doi.org/10.3390/electronics14132604.

[13] K. Zheng, J. Jiang, Z. Liang, Y. Yoo & Y. Wang, "Lane detection algorithm based on improved YOLOv8", Computers **15** (2026) 561. https://doi.org/10.3390/computers15090561.

[14] Y. Yang, H. Peng, C. Li, W. Zhang & K. Yang, "LaneFormer: Real-time lane exaction and detection via transformer", Applied Sciences **12** (2022) 9722. https://doi.org/10.3390/app12199722.

[15] Z. Feng, S. Guo, X. Tan, K. Xu, M. Wang & L. Ma, "Rethinking efficient lane detection via curve modeling", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 17062. https://doi.org/10.1109/CVPR52688.2022.01655.

[16] C. Li, B. Zhang, J. Shi & G. Cheng, "Multi-level domain adaptation for lane detection", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), New Orleans, LA, USA, 2022, pp. 4380. https://doi.org/10.1109/CVPRW56347.2022.00484.

[17] K. Bhardwaj, Z. Wan, A. Raychowdhury & R. A. Goldhahn, "Real-time fully unsupervised domain adaptation for lane detection in autonomous driving", 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 2023, pp. 1. https://doi.org/10.23919/DATE56975.2023.10136937.

[18] P. Ganeriwala, S. Bhattacharyya & R. Muthalagu, "Cross dataset analysis and network architecture repair for autonomous car lane detection", 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 2023, pp. 1. https://doi.org/10.1109/IV55152.2023.10186721.

[19] Y. Kao, S. Che, S. Zhou, S. Guo, X. Zhang & W. Wang, "LHFFNet: A hybrid feature fusion method for lane detection", Scientific Reports **14** (2024) 16353. https://doi.org/10.1038/s41598-024-66913-1.

[20] J. Zhang, B. Wang, H. Naeem & S. Dai, "Robust spatiotemporal lane detection model", Transportation Research Record **2679** (2025) 2213. https://doi.org/10.1177/03611981241260696.

[21] E. H.-C. Lu & W.-C. Chiu, "Lane detection based on CycleGAN and feature fusion in challenging scenes", Vehicles **7** (2025) 2. https://doi.org/10.3390/vehicles7010002.

[22] G. Lian, "Robust lane detection based on informative feature pyramid network in complex scenarios", Electronics **14** (2025) 3179. https://doi.org/10.3390/electronics14163179.

[23] M. Rahul Surya & S. K. Dash, "Robust lane detection in autonomous driving using gated convolutional networks with spatial attention", Results in Engineering **30** (2026) 110129. https://doi.org/10.1016/j.rineng.2026.110129.

[24] O. Njama-Abang, D. U. Ashishie, E. A. Edim & M. A. Agana, "Development of a visual analogy model using transfer learning techniques", African Scientific Reports **4** (2025) 304. https://doi.org/10.46481/asr.2025.4.3.304.

[25] A. A. Keji, O. Fakeye, N. N. Onochie & O. Sangotoki, "Predicting long-term deposit customers using convolutional neural network and data conversion technique", African Scientific Reports **3** (2024) 192. https://doi.org/10.46481/asr.2024.3.3.192.

[26] G. James, A. Ekong, E. Abraham, E. Oduobuk, N. Michael, V. Ufford & O. Ebong, "An enhanced control solutions for efficient urban waste management using deep learning algorithms", African Scientific Reports **3** (2024) 183. https://doi.org/10.46481/asr.2024.3.3.183.

[27] A. Adegoke-Elijah, T. O. Ojewumi & K. O. Jimoh, "ECG anomaly detection: A deep learning perspective with LSTM encoders", African Scientific Reports **4** (2025) 359. https://doi.org/10.46481/asr.2025.4.3.359.

[28] TuSimple, "TuSimple Lane Detection Challenge", TuSimple Benchmark (2017). Available online: https://github.com/TuSimple/tusimple-benchmark.

[29] X. Pan, J. Shi, P. Luo, X. Wang & X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding", Proceedings of the AAAI Conference on Artificial Intelligence **32** (2018) 7276. https://doi.org/10.1609/aaai.v32i1.12301.

[30] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan & T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 2633. https://doi.org/10.1109/CVPR42600.2020.00271.