

# A Survey on Deep Learning-Based Lane Detection Algorithms for Camera and LiDAR

Min-Hyeok Sun<sup>✉</sup>, Seung-Hyun Kong<sup>✉</sup>, Senior Member, IEEE,  
and Dong-Hee Paek<sup>✉</sup>, Graduate Student Member, IEEE

**Abstract**—Lane detection algorithm (LDA) is a crucial and necessary component for autonomous vehicles to ensure safe driving in various environments. Deep learning-based lane detection algorithms (DL-LDAs) have gained significant attention recently, and there have been a number of DL-LDAs, introduced in the literature, showing a continuous performance improvement in lane detection. In general, DL-LDAs are composed of pre-processing, lane feature extraction, lane detection head, and an optional lane fitting. For a systematic overview of various DL-LDAs, we provide detailed explanations for each functional component of DL-LDAs. Moreover, this paper presents the first survey to comprehensively analyze various DL-LDAs using camera and LiDAR, such as 2D (2-Dimensional) and 3D DL-LDAs using camera images and DL-LDAs using LiDAR point cloud or sensor fusion. In addition to the analysis, we present recent public lane detection benchmarks for DL-LDAs and discussions concerning technical issues that need to be addressed in future DL-LDA studies.

**Index Terms**—Lane detection, deep learning, autonomous driving, camera, LiDAR, sensor fusion.

## I. INTRODUCTION

LANE detection algorithms (LDAs) that recognize various lanes and road marks have been widely commercialized for advanced driver assistance systems (ADAS), such as lane keeping assist system (LKAS) and adaptive cruise control (ACC). In the advanced autonomous driving systems of level 3 or higher, according to the Society of Automotive Engineers International (SAE) [1], LDAs produce valuable lane information (e.g., lane locations and drivable area) that enables a vehicle to safely plan local routes for cautious maneuvers, such as lane change and overtaking.

Due to the high resolution and wide usage of camera in the vehicles, LDAs using camera images have been actively studied. In general, lanes are marked with eye-catching colors (e.g., white and yellow) and have long and thin line

Received 30 July 2023; revised 21 December 2023, 13 May 2024, 9 September 2024, and 22 November 2024; accepted 7 January 2025. Date of publication 7 April 2025; date of current version 2 June 2025. This work was supported by the National Research Foundation of Korea (NRF) Grant funded by Korea Government [Ministry of Science and ICT (MSIT)] under Grant 2021R1A2C3008370. The Associate Editor for this article was Q. Zhang. (Corresponding author: Seung-Hyun Kong.)

Min-Hyeok Sun is with Hyundai Motor Company, Seoul 06797, South Korea (e-mail: mhsun@hyundai.com).

Seung-Hyun Kong and Dong-Hee Paek are with the CCS Graduate School of Mobility, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea (e-mail: skong@kaist.ac.kr; donghee.paek@kaist.ac.kr).

Digital Object Identifier 10.1109/TITS.2025.3554695

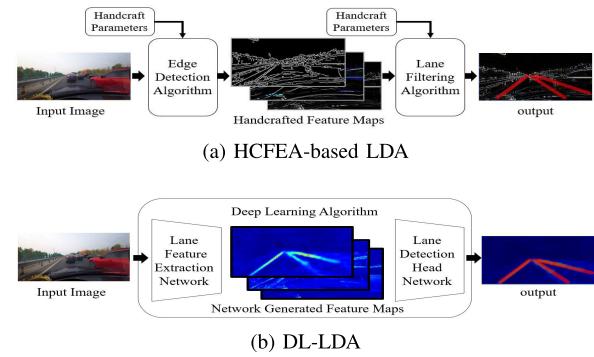


Fig. 1. Overall structures of two LDA methods; (a) shows the HCFEA-based LDA, where HCFEA is used for lane feature extraction and lane fitting. (b) illustrates the DL-LDA that exploits deep learning algorithm for lane detection.

shapes. To recognize lanes of these characteristics, hand-crafted feature extraction algorithms (HCFEAs), such as HSV color feature extraction [2], Sobel edge detection [3], Canny edge detection [4], [5], [6], Hough transformation [7], [8], [9], and Kalman filtering [10], [11], [12], have been widely used, as shown in Fig. 1a. These HCFEAs perform well in clear weather conditions, but degrade severely in adverse conditions, such as rain, snow, or backlight, where lanes and roads may not be clearly distinguished. This maybe because HCFEAs require different optimal hyper-parameters for particular weather and illumination conditions. However, manually configuring hyper-parameters suitable to every condition presents a substantial challenge. As a result, these HCFEA-based LDAs cannot guarantee consistent performance for various road environments [13], [14], [15], [16].

Deep learning algorithms offer a promising solution to the problems of HCFEAs, since deep learning algorithms can learn optimal parameters for various environmental conditions from a large amount of data. In the past, the usage of deep learning algorithms was restricted by limited hardware performance and lack of large-scale datasets. However, recent improvements in computing hardware, such as graphic processing unit (GPU) and neural processing unit (NPU), and the availability of multiple large-scale open datasets, such as COCO [17] and ImageNet [18], have enabled the use of deep learning algorithms for lane detection. Since 2016, a number of deep learning-based lane detection algorithms (DL-LDAs) [19], [20] have demonstrated superior robustness and accuracy compared to traditional HCFEA-based LDAs. And some survey stud-

ies have provided comprehensive analyses of DL-LDAs to highlight the recent research trends and accelerate DL-LDA studies.

Early survey studies [14], [21], [22], [23] focus on HCFEA-based LDAs. The study in [21] provides a thorough analysis of HCFEAs and an overview of LDAs. And [22] analyzes various existing lane detection studies and discusses the potential utilization of multiple sensors, for example, monocular camera, stereo camera, and LiDAR. In [14], the authors provide an analysis of extended concept of LDA studies, lane tracking. And [23] covers complete technical aspects and explains detailed block diagrams of lane detection systems. All of these studies [14], [21], [22], [23] provide a meaningful overview of LDAs, however, these studies do not include the analysis of DL-LDAs, because deep learning algorithms were not widely used at the time of their publication.

On the other hand, studies [15], [16], [24], [25], [26] explain new developments in DL-LDAs compared to the traditional HCFEA-based LDAs. In [24], the authors classify DL-LDAs with respect to the level of lane detection, such as ego-lane detection and all-roads lane detection. The study in [15] introduces various LDAs in two categories: traditional LDAs and recent DL-LDAs. And [25] classifies DL-LDAs into classification-based, object detection-based, and segmentation-based DL-LDAs, and explains representative studies in each class. In [16], the authors provide an analysis of broad range of deep learning algorithms for DL-LDAs. The study in [26] is one of the most recent studies in lane detection and provides analysis of DL-LDA studies with literature statistics. These studies [15], [16], [24], [25], [26] are worth noticing for the development of DL-LDAs, but have some drawbacks. For example, [15], [24], [26] lack detailed explanation about DL-LDAs, and the classification criteria for DL-LDAs in [25] are not appropriate for recent DL-LDAs. In addition, [16] focuses only on the methods to extract lane detection features but does not discuss the overall process of the lane detection systems. In fact, these studies were published before the year 2020, and, thus, do not discuss the latest DL-LDAs introduced in recent years. Moreover, all of the previous survey studies have not addressed new topics in DL-LDA, such as 3D lane detection and lane detection using LiDAR point cloud or sensor fusion, which have gained significant attention recently.

In this paper, we present the most comprehensive and up-to-date survey of recent DL-LDA studies. Technically, we focus on the overall functional processes of DL-LDAs, which include pre-processing, lane feature extraction, lane detection head, and optional lane fitting. We conduct a detailed analysis of the key ideas in each process of DL-LDAs. Furthermore, this paper covers various studies other than the 2D lane detection using camera Images, for example, 3D lane detection using camera Images, lane detection using LiDAR point cloud, and lane detection using sensor fusion, which have been recently proposed but not discussed in other surveys. We compare dataset characteristics, technical issues, and problem-solving approaches of a number of recent DL-LDA studies, in order to derive deeper and broader insights into recent DL-LDAs. The comprehensive analy-

sis provided in this paper will be necessary to develop advanced DL-LDAs and to identify next research directions. The contribution of this survey paper can be summarized as follows:

- We present the most up-to-date and comprehensive analysis of DL-LDA studies compared to the existing survey papers on lane detection systems.
- We analyze various DL-LDAs from functional perspective, so we organize subsections based on the sequential process of DL-LDAs. And we classify and analyze the methods in each process and provide key insights based on technical and methodological concepts.
- We provide the first survey of DL-LDA studies other than the conventional 2D lane detection using camera Images, for example, 3D lane detection using camera Images, lane detection using LiDAR point cloud, and lane detection using sensor fusion, all of which are at the leading edge of DL-LDAs.
- We provide the latest benchmarks and summary tables of DL-LDAs, enabling readers to identify the strengths, drawbacks, and features of each study to facilitate technical comparisons.

We organize this paper into seven sections to address the functional components of DL-LDAs. Section II presents various sensors and datasets that are used for DL-LDAs. Section III explains the step-by-step process of 2D DL-LDAs using camera images in subsections and provides detailed explanations of each process. Section IV analyzes recent studies other than the 2D lane detection using camera Images. Section V introduces widely-used performance evaluation metrics for DL-LDAs and presents the most recent benchmark tables with some analysis. Section VI discusses the technical issues to be addressed in the future DL-LDAs, and, finally, Section VII draws the conclusion of this paper.

## II. SENSORS AND DATASET FOR LANE DETECTION

In this section, we provide an overview of sensors, such as camera and LiDAR, and datasets used for DL-LDA developments. In subsection II-A, we discuss two representative sensors that are widely used for lane detection: camera and LiDAR, while in subsection II-B, we explore various open datasets with details of their data types and characteristics.

### A. Sensors for Lane Detection

1) *Camera*: Camera captures visual information of the front scene and develops RGB (Red-Green-Blue) intensity maps with high resolution. This makes camera very useful to distinguish lane marks, such as solid lines, dotted lines, and double yellow lines, from the road surface, and to figure out the lane curvature. Therefore, camera has been the most widely used sensor for DL-LDAs. However, camera is susceptible to poor light conditions, such as nighttime and early morning in addition to the adverse weather conditions, such as rain and snow. Moreover, camera projects 3D objects in 3D space onto a 2D (front-view) plane, which inherently results in a loss of depth information and causes a challenge in precisely detecting 3D shapes of various lanes.

TABLE I  
THE OPEN DATASETS FOR DL-LDAs

Datasets type	Name	Released Year	Location of Data Capture	Image Resolution	Number of Frames	Maximum Number of Lanes	Instance-Level Lane Annotation	Other Road Mark Annotation	Including Diverse Road Types	Time-Variant Data Correction	Weather-Variant Data Correction	Real 3D X, Y Annotation	Real 3D Z Annotation
Datasets for 2D Lane Detection with Camera Images	[27] Caltech Lanes	2008	2 regions in USA	640×480	1.2K	4	X	X	X	X	X	X	X
	[28] KITTI Roads	2013	Karlsruhe, Germany	1242×375	0.6K	2	X	X	X	X	X	X	X
	[29] ELAS *	2017	Regions in Brazil	640×480	15K	4	X	O	O	X	X	O	X
	[35] TuSimple	2017	San Diego, USA	1280×720	6.4K	↑ 4	X	X	X	X	X	X	X
	[32] VPGNet *	2017	Seoul, Korea	1280×728	21K	↑ 4	X	O	X	O	O	X	X
	[19] CULane	2018	Beijing, China	1640×590	133K	4	O	X	O	O	X	X	X
	[33] LLAMAS	2019	Unknown	1280×717	10K	↑ 4	O	X	X	X	X	X	X
	[36] BDD100K	2020	4 regions in USA	720×480	100K	↑ 4	X	O	O	O	O	X	X
	[30] ApolloScape	2020	4 regions in China	3384×2710	110K	↑ 4	X	O	O	O	O	O	O
	[37] CurveLanes	2020	Regions in China	2560×1440	150K	↑ 4	O	X	O	O	O	X	X
Datasets for 3D Lane Detection with Camera Images	[38] VIL-100	2021	Unknown	1920×1080	10K	↑ 4	O	O	O	O	O	X	X
	[39] Synthetic 3D Lanes	2020	Unity simulator	360×480	10K	↑ 4	X	X	O	X	X	O	O
	[31] ONCE-3DLanes	2022	Regions in China	1920×1020	211K	↑ 4	X	X	O	O	O	O	O
Datasets for Lane detection using LiDAR Point Cloud	[40] OpenLane	2022	3 regions in USA	1920×1280	200K	↑ 4	O	O	O	O	O	O	O
	[41] Deep Lane *	2018	North America	(point cloud data)	55K	↑ 4	X	X	O	X	X	O	X
	[42] RoadNet *	2020	Unknown	(point cloud data)	5.2K	2	O	X	X	X	X	O	X
	[20] K-Lane	2022	Deajeon, Korea	(point cloud data)	15K	↑ 4	O	X	O	O	X	O	X

Datasets with mark \* require an email request to access.

2) **LiDAR:** Light detection and ranging (LiDAR) sensor measures the Time-of-Flight (ToF) of the returning infrared pulses that are reflected on objects in the field of view. And the ToF can be translated into the distance between the sensor and objects. In practice, LiDAR collects measurements of the surrounding 3D space, in all azimuth directions and within a window of elevation angles, almost simultaneously. The LiDAR point cloud (i.e., LiDAR measurements) provides accurate 3D information of every line of sight (LOS) object within the measurement range. In addition, LiDAR can measure the reflectivity of the object's surface, which depends on the color, roughness, material, etc. Because of the reflectivity, LiDAR measurements can be used to distinguish the painted lane marks from the road surface, which is used for lane detection.

### B. Datasets for Lane Detection

To train and evaluate DL-LDAs, a sufficient amount of sensor data with precise lane annotations is essential and necessary. However, data collection and annotation require a significant amount of labor. Thankfully, several studies have built large-scale open (to public) datasets that contain various sensor data (i.e., measurements) and precise lane annotations for DL-LDA developments. This subsection presents an overview of the lane detection datasets and classifies the datasets by their characteristics, as in Table I.

#### 1) Dataset for 2D Lane Detection Using Camera Images:

a) *Caltech lanes:* The Caltech Lanes dataset [27], introduced in 2008, was widely used for early studies of HCFEA-based LDA. The dataset is collected on the roads in two regions in the United States and offers lane annotations on four different lanes, including two ego lanes and their adjacent lanes. However, this dataset is not designed for DL-LDAs, since the size of the dataset, 1,224 frames, is not sufficient to train a DL-LDA.

b) *Kitti roads:* The Kitti Roads dataset [28], collected at the Karlsruhe region of Germany in 2013, offers camera images and annotations for lane detection. This dataset also provides evaluation metrics for performance comparison of various LDAs. However, this dataset has a small amount of data (579 frames) and only contains data from rural roads in

a day-light condition, which means a low data diversity to be used for DL-LDA developments.

c) *ELAS:* The ELAS dataset [29], released in 2017, provides lane annotations for approximately 15,000 frames of road images in Brazil. Especially, the dataset offers annotations for different types of lane marks, such as solid white lines, dotted white lines, solid yellow lines, and dotted yellow lines. And this dataset is collected in diverse areas, such as highways, rural, and urban areas. However, this dataset is collected in ideal illumination conditions (for example, a clear daytime) and does not provide annotations for individual lane instances.

d) *TuSimple:* TuSimple dataset was made public for the lane detection challenge at the computer vision and pattern recognition (CVPR) conference in 2017, and, since then, a number of studies are using this dataset for lane detection performance evaluation. The dataset provides 6,408 frames in highways with up to five lane annotations. However, TuSimple dataset has an insufficient amount of data to train DL-LDAs and only provides data in good illumination conditions (i.e., daytime).

e) *VPGNet:* The VPGNet dataset [32], released in 2017, provides lane data collected at multiple locations in Seoul, South Korea. In addition to the lane annotations, this dataset includes annotations about 17 different types of road marks, such as crosswalks, safety zones, etc. This dataset provides data captured in various illumination and weather conditions, however, this dataset does not provide instances-level lane annotations.

f) *CULane:* The CULane dataset [19], opened in 2018, is one of the most widely used datasets for DL-LDAs. Specifically, the CULane dataset covers various road types, and the performance evaluation benchmark categorizes data frames into normal, crowded, night, no line, shadow, arrow, dazzle light, curve, and cross-roads. Moreover, CULane contains the largest amount of data at the time of publication, 133,235 high-capacity data captured in multiple areas of Beijing, China, and provides up to four individual lane instance annotations, including ego lanes and two adjacent lanes (to the ego lanes), as shown in Fig. 2a.

g) *LLAMAS:* The LLAMAS dataset [33], released in 2020, provides one of the most precise spline annotations for

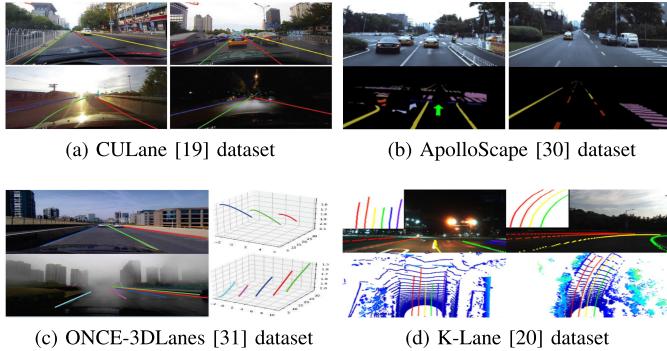


Fig. 2. Sample frames of various public lane detection datasets. Each sample frame shows camera images or LiDAR point cloud and corresponding lane annotations.

individual lanes and dashed lane mark segments of each lane. The dataset offers 100,000 frames of camera images on the highways with automatically generated lane annotations using GraphSLAM [34] and high definition (HD) maps. However, this dataset does not include challenging scenes, such as scenes with low lane visibility and scenes in heavy traffic.

*h) BDD100K:* The BDD100K dataset [36], published in 2020, is a large-scale multi-task dataset for various tasks related to autonomous driving, such as object detection, semantic segmentation, drivable area segmentation, and lane detection. For the lane detection task, this dataset provides approximately 100,000 frames that are collected in New York City and San Francisco, Berkeley, and the Bay Area in California in various driving conditions, such as daytime, nighttime, rain, and snow. And, for each frame, this dataset provides annotations for lanes and 11 types of road marks. However, these annotations are not at the lane instance level.

*i) ApolloScape:* The ApolloScape dataset [30], released in 2020, provides the most diverse lane annotation to lane mark types. Specifically, the dataset offers annotations of 27 different types of lane marks in each frame. The dataset contains approximately 160,000 frames collected in four different regions of China, including urban and suburban areas and highways in various illumination conditions. In addition, this dataset also provides high-quality data for the development of other DL algorithms for autonomous driving, such as semantic segmentation and instance segmentation.

*j) CurveLanes:* The CurveLanes dataset [37], published in 2020, is a large-scale dataset of approximately 150,000 frames collected on various roads in China. This dataset stands out from existing datasets, since most of the frames contain at least one curved lane. In addition, CurveLanes provides instance-level lane annotations for the frames captured in various illumination and weather conditions, such as day, night, and rainy conditions. Due to these unique characteristics, the CurveLanes dataset is widely used for the development of DL-LDAs capable of accurate detection of lanes with different curvatures.

*k) VIL-100:* The VIL-100 dataset [38], introduced in 2021, is specially designed for DL-LDAs that exploit the continuity of lanes between consecutive frames in the video sequence. Therefore, different from existing datasets that support lane detection with instantaneous images, this dataset provides 100 video sequences, where each video sequence

captures about 10 seconds of road scene at 10 fps (image frame per second). In each frame, the dataset provides annotations for instance-level lanes and 10 different types of road marks on various roads, such as highways and local roads, in various illumination and weather conditions.

### 2) Dataset for 3D Lane Detection:

*a) Synthetic 3D lanes:* The Synthetic 3D Lanes dataset [39], published in 2020, provides image data and corresponding 3D lane annotations for the first time. Since capturing the data and generating 3D lane annotations in the virtual simulation environment is much easier than those in the real world, the Synthetic 3D Lanes dataset consists of scenes from the virtually simulated real-world roads in Silicon Valley, USA, with a sufficient quantity of high-quality 3D lane data. In detail, this dataset has a total of 10,500 frames collected at various times of days, with 6,000 frames in highways, 1,500 in urban areas, and 3,000 in residential areas.

*b) ONCE-3DLanes:* The ONCE-3DLanes dataset [31], released in 2022, offers 211,000 real-world images and corresponding 3D lane annotations, collected in various illumination and weather conditions, as shown in Fig. 2c. This dataset is generated based on the ONCE [43] 3D object detection dataset. In detail, the authors separate the LiDAR point cloud on the road surface from each frame of the ONCE dataset, project the separated LiDAR point cloud onto 2D forward-facing camera images, add 2D lane annotations, and then back-project the generated 2D lane annotations to the 3D LiDAR point cloud to acquire 3D lane annotations.

*c) OpenLane:* The OpenLane dataset [40], introduced in 2022, is generated by adding lane annotations to the Waymo Open dataset that is used for 3D object detection and other autonomous driving algorithms. Therefore, OpenLane dataset provides large-scale data for 3D lane detection that contains 2D and 3D lane annotations on both forward-facing camera images and LiDAR point cloud data, respectively. Specifically, OpenLane includes approximately 200,000 frames collected at three regions in the United States, and provides instance-level lane annotations, 14 different types of road marks, and lane tracking information. The OpenLane dataset has a relatively even data distribution compared to other datasets, since the dataset is collected in various driving conditions.

### 3) Dataset for Lane Detection Using LiDAR Point Cloud:

*a) DeepLane:* The DeepLane dataset [41], introduced in 2018, is the first lane detection dataset of LiDAR point cloud. The dataset provides lane annotations on 2D bird's-eye-view (BEV) projected images of LiDAR point cloud from precisely aligned five consecutive frames, which results in denser point cloud than a point cloud frame obtained with a single LiDAR. However, aligning consecutive frames for lane detection in actual autonomous vehicles may not be practical because of high computational resources and runtime requirements.

*b) RoadNet:* The RoadNet dataset [42], released in 2020, includes 5,200 frames of 2D BEV images derived from the projection of LiDAR point cloud onto the horizontal plane. The dataset includes class annotations for left, right, and other lanes. However, since the dataset consists of aligned multiple consecutive frames, a DL-LDA trained with the dataset needs

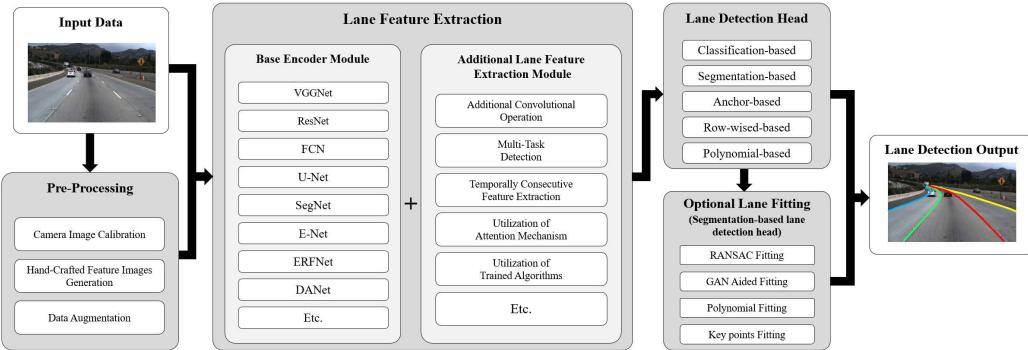


Fig. 3. The overall flow diagram of DL-LDAs. First, sensor data is input to the pre-processing to facilitate lane feature extraction. Second, various base encoder modules and additional lane feature extraction modules are used in the lane feature extraction to obtain comprehensive lane features for lane detection. Third, these extracted features are then processed by lane detection head and an optional lane fitting to generate final lane detection results.

an inference input in the same format to the dataset, which may be impractical for real-time lane detection.

c) *K-Lane*: The K-Lane dataset [20], published in 2022, is a useful dataset for the development of DL-LDAs using LiDAR point cloud in practice. Unlike other datasets for lane detection using LiDAR point cloud, this dataset provides front-view camera images, LiDAR point clouds, and lane annotations for up to six lanes on each frame of the projected LiDAR BEV images that are obtained using a 64-channel LiDAR, as shown in Fig. 2d. Since the dataset consists of 15,000 LiDAR point cloud frames collected in various road and illumination conditions, the K-Lane dataset provides a data diversity to facilitate autonomous driving in practice. Consequently, K-Lane highlights the potential of LiDAR point cloud for lane detection in real-world environments.

### III. 2D LANE DETECTION USING CAMERA IMAGES

Camera expresses the incoming visual information in the RGB channels, which provides rich color, texture, and geometric features of the scene more precisely than other sensors. Thanks to these advantages, 2D lane detection using camera images has been a highly active study area. In this section, we aim to provide a detailed analysis of DL-LDAs that perform 2D lane detection using camera images.

The 2D LDAs using camera images have three main processes and an optional process, as shown in Fig. 3: (1) pre-processing, (2) lane feature extraction, (3) lane detection head, and (4) an optional lane fitting. In the pre-processing, various operations on the input image are conducted to enable lane feature extraction, effectively. The lane feature extraction entails identifying the clues for lane detection on the input image. While previous studies have employed HCFEAs to extract lane features, a number of methods using deep learning algorithms have been proposed recently for lane feature extraction. In the lane detection head, the lane shape is determined based on the extracted features. Some DL-LDAs further utilize lane fitting to improve the lane detection accuracy. Detailed explanations for each of the four processes are provided in the following subsections.

#### A. Pre-Processing

Pre-processing is located at the forefront of DL-LDAs and performs various operations on the input image to improve

the efficiency of the following lane feature extraction network. Although pre-processing has a positive effect on feature extraction for lane detection, pre-processing increases the computational complexity of DL-LDAs.

Camera image calibration [44] is a widely used operation in the pre-processing for many computer vision tasks. Since the cameras capture the front incoming light through the lens, the lens curvature causes some inherent distortion. In general, the wider the field of view (FOV) of a camera sensor, the greater the curvature of the lens and thus, more distortion. This distortion can negatively affect lane detection accuracy as it transforms straight lines into curved and wavy forms. To correct this distortion, studies such as [28] suggest camera calibration prior to the extraction of features for lane detection.

Another pre-processing operation generates diverse hand-crafted feature images using HCFEAs. For example, the DL-LDAs in [45] and [46] apply HCFEAs to create black-and-white edge detection images, which can be utilized as image masks to assist lane feature extractions in the following process. By using both the input image and the image masks, the lane feature extraction can focus on pixels on the input image that show the presence of edges highlighted by the masks.

Data augmentation is an effective pre-processing operation to train DL-LDAs. In order to achieve a sufficient lane detection performance, training DL-LDAs requires a large amount of high-quality and refined data. However, collecting and annotating a large amount of data is a time-consuming task. To tackle this, most studies employ various data augmentation operations to generate extra data and annotation using the existing ones. For example, image rotation and flipping can generate multiple extra training data from a pair of image and annotation. In addition, to make the dataset diverse, CycleGAN-ERFNet [47] proposes a method that uses generative adversarial network (GAN) [48]. Specifically, CycleGAN-ERFNet introduces CycleGAN to generate new images as shown in Fig. 4. CycleGAN conducts style transfer that alters color and texture while preserving the scene content on the image. As a result, the images generated by CycleGAN can increase the diversity and quantity of the dataset, effectively.

#### B. Lane Feature Extraction

After pre-processing, DL-LDAs perform the second main process: lane feature extraction that searches for clues to

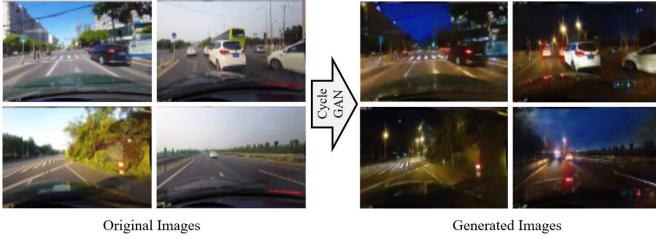


Fig. 4. Examples of data augmentation using CycleGAN in [47]. CycleGAN modifies the color and texture of daytime images to enhance the diversity and quantity of the dataset.

detect lanes from a given input. Before utilizing deep learning algorithms for LDAs, HCFEAs, such as HSV color feature extraction [2], Sobel edge detection [3], Canny edge detection [4], [5], [6], Hough transformation [7], [8], [9], and Kalman filtering [10], [11], [12], have been used for this process. Although these HCFEAs work well in moderate conditions (e.g., good illumination and weather conditions with less occlusion), they may not guarantee consistent detection performance in various illumination and weather conditions [49].

Deep learning algorithms provide a promising solution to the problems of HCFEAs, since deep learning algorithms can learn optimal parameters for various environmental conditions. Among various deep learning algorithms, convolution neural network (CNN) [50] is one of the most effective algorithms to extract features from an input image. CNN utilizes convolution operations with 2D (or 3D) kernels sliding over the input image, which enables algorithms to extract the feature of texture and spatial relationships between neighboring pixels. In addition, using the max pooling operation, CNN can compress the features within the pooling kernel window to extract global high-semantic features. These global high semantic features are used to understand the meaning of the corresponding area on the output feature map and provide evidence to determine whether the area belongs to the lane or background. In recent years, active studies on CNN have led to the emergence of new types of deep neural networks, including AlexNet [51], VGGNet [52], GoogLeNet [53], ResNet [54], and others. Consequently, DL-LDAs utilize these types of CNN as the base encoder modules. Furthermore, several studies have introduced additional lane feature extraction modules to explore lane features comprehensively further than the base encoder modules. In this subsection, we first discuss the various base encoder modules that are commonly used for DL-LDAs, and then discuss various additional lane feature extraction modules.

*1) Base Encoder Module:* Base encoder module, placed at the forefront of the lane feature extraction, is responsible for extracting spatial, textural, and geometric characteristics of the objects on the input image, which facilitates the exploration of local features. The output generated by the CNN layers in this module is condensed through pooling operations and enables the acquisition of global features. The base encoder module leverages both local and global features to produce high semantic features that serve as foundational clues for determining lane shapes on the input image.

VGGNet [52] is one of the most widely used deep neural networks for the base encoder module, due to its simplicity

and effectiveness. Its deep architecture comprises a series of convolutional layers using  $3 \times 3$  kernels and pooling layers, which enables VGGNet to extract complex and hierarchical features from the input images. ResNet [54] introduces residual connections to address the vanishing gradient problem that occurs in deeper layers of VGGNet or other deep neural networks. The residual connections lessen the vanishing gradient problem in the backpropagation and, thus, allow a deep neural network to be trained deeper for enhanced performance. Fully convolutional networks (FCN) [55] utilizes  $1 \times 1$  kernels in the convolutional layers, which makes FCN efficiently extract pixel-wise high-semantic features for semantic segmentation.

The above networks utilize max-pooling operation to obtain global features for segmentation. However, this operation halves the size of the input image (or feature maps) to compress various local features, and, thus, produces an output of low quality segmentation feature map. To generate the high quality segmentation feature map, the encoder-decoder structure is widely used for the base encoder modules. The encoder reduces the spatial dimension of the input image while extracting high-level semantic features. On the other hand, the decoder progressively recovers the spatial dimension of feature maps and restores high-level spatial features. The encoder-decoder structure allows to effectively combine both high-level semantic features and spatial features and, then, to accurately delineate object boundaries and preserve fine details on the output feature map.

U-Net [56] has such an encoder-decoder structure that is useful for segmentation tasks. Specifically, U-Net incorporates a series of symmetric contracting and expanding paths. In the expanding path, the network concatenates lower-layer feature maps (which include high-level spatial features) with deeper-layer feature maps (which include high-level semantic features). This concatenation allows the output feature map to contain comprehensive and detailed semantic and spatial features essential for detecting lane boundaries on the complex road image. SegNet [57] uses stored max-pooling indices rather than stored entire feature maps to exploit an encoder-decoder structure, with a focus on optimized memory usage and computational efficiency. Specifically, in the encoding phase, SegNet stores max-pooling indices from each pooling layer, and, in the decoding phase, these indices are utilized to up-sample feature maps, and to reconstruct the spatial information.

Feature pyramid network (FPN) [58] also utilizes an encoder-decoder structure to consider both semantic and spatial features across various scales. The difference between FPN and former encoder-decoder structures is that while the latter concatenates feature maps at the final decoder layer and performs lane detection with a single detection head, FPN employs several detection heads for each feature map generated at different level layers. Therefore DL-LDAs using FPN can utilize the lane predictions from both the high semantic and spatial feature maps.

While the encoder-decoder structure can provide high-quality segmentation feature maps, their complex structures increase the computational cost, which affects the real-time performance of DL-LDAs. Since real-time performance is one

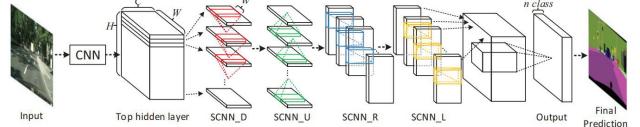
of the most crucial aspects of LDAs, this problem should not be overlooked. In this respect, ENet [59] has been often used for the base encoder module of DL-LDAs. ENet employs an initial block followed by a series of bottleneck layers, which enables the network (i.e., ENet) to efficiently learn complex features with reduced computational cost. ERFNet [60] is another efficient network for real-time semantic segmentation tasks. Specifically, ERFNet introduces factorized residual layer, which combines the advantages of residual connections and factorized convolutions [61]. This structure shows a reduced number of parameters, while maintaining a balance between the accuracy and the computational complexity. In addition, ERFNet utilizes dilated convolutions to increase the receptive field without increasing the number of parameters or computational cost.

Other than the conventional base encoder modules discussed above, a number of neural networks that are developed for various segmentation tasks (e.g., semantic segmentation, instance segmentation, and panoptic segmentation) can be used for the base encoder modules of DL-LDAs. Readers who are interested in additional details of these encoder base modules may refer to the papers [62], [63], [64].

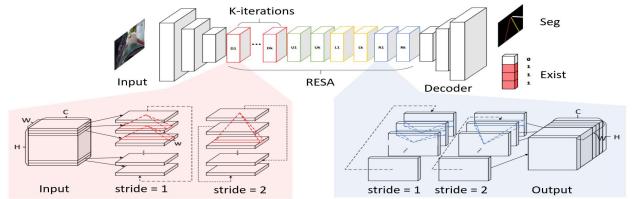
*2) Additional Lane Feature Extraction Module:* The base encoder modules discussed in the previous subsection are designed to extract high semantic features from the input image and can be applied to a wide range of image processing tasks. However, using only such high semantic features may be insufficient for precise lane detection. In fact, the presence of the existing base encoder modules are limited to their receptive fields, which restricts the full exploration of various global features useful for lane detection. However, there are multiple inherent clues from these global features that are useful for detecting lanes in DL-LDAs, but not easily detected using the base encoder modules. One example of these global features is that all of the lanes starting from the bottom of the image converge towards the vanishing point (or vanishing line) in each frame. Another is that lanes typically exhibit global characteristics such as parallel lines and constant arrangement. By exploring these global features, DL-LDAs can enhance the overall lane detection performance and even predict occluded lanes in challenging conditions. To leverage these global features, a number of DL-LDA studies have presented additional ideas to extract various lane features that are hardly explored by the conventional base encoder modules. In the following subsections, we discuss the additional lane feature extraction modules to the base encoder module.

*a) Additional convolution operation:* Several studies have proposed ideas to advance conventional convolution operations that can extract lane-specific features from the input image effectively.

For example, SCNN [19] proposes Spatial CNN to effectively extract relationship features between the pixels of a given input image. Unlike the layer-by-layer convolution operation in the standard CNNs, Spatial CNN applies slice-by-slice convolution operation to each feature map, as shown in Fig. 5a. Instead of generating the new feature map after the convolution operation, the slice-by-slice convolution operation parses the features of each pixel to the adjacent pixels in a specific



(a) Structure of SCNN [19]



(b) Structure of RESA [65]

Fig. 5. Structures of SCNN [19] and RESA [65]. SCNN employs Spatial CNN that performs slice-by-slice convolution to transfer and to provide features from pixels to adjacent pixels. RESA applies multiple strides to the Spatial CNN to provide diverse features to the pixels farther apart.

direction (up, down, right, left) within the input feature map. As a result, each feature is combined with various features of the surrounding pixels. This operation allows Spatial CNN to effectively extract global features from objects that have long shapes, such as lanes or walls. The study in [46] applies Spatial CNN at the end of the base encoder module to extract rich spatial features for lane detection. However, using Spatial CNN requires a high computational cost. RESA [65] proposes a method that extracts relationship features more efficiently than Spatial CNN. In detail, RESA applies Spatial CNN to the feature map of smaller size compared to SCNN [19] with multiple strides as shown in Fig. 5b. As a result, the features of each pixel to be directly sent to pixels that are much farther away than adjacent pixels.

Since lanes typically have a line shape (i.e., long and thin), SpinNet [66] employs line-shaped kernels (e.g.,  $1 \times 9$  or  $1 \times 25$  size) to effectively extract the line shape features. Due to their own shape, the kernels can more effectively explore features related to lane marks compared to traditional square-shaped kernels (e.g.,  $3 \times 3$  and  $5 \times 5$  sizes). In addition, inspired by the fact that lanes are typically not vertical but rather diagonal, SpinNet rotates feature maps and applies CNN with line-shaped kernels to explore diagonally tilted lane features.

*b) Multi-Task detection:* Utilizing only lane-specific semantic features (e.g., white or yellow color and the line shape) on the input image maybe not appropriate to detect lanes precisely, when the lane segments are unclear or indistinguishable. For example, lane marks could be occluded by other vehicles or peeled off. To address this problem, some DL-LDAs utilize various features from other than lanes, such as drivable area, vanishing point (or vanishing line), and the locations of nearby vehicles.

EELane [67] applies a multi-task detection approach introduced in OverFeat [68], such that the neural network is simultaneously trained for vehicle detection and lane detection. The experiment results demonstrate that training a neural network to perform multiple independent tasks simultaneously can improve the accuracy for each of the tasks. To effectively detect occluded lanes, VPGNet leverages the fact that all of

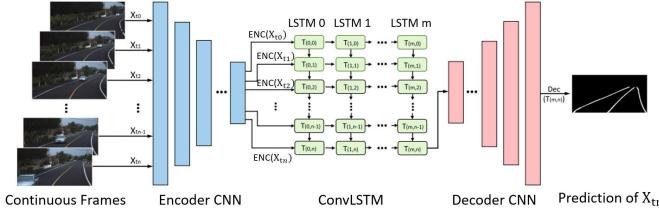


Fig. 6. Examples of temporally consecutive feature extraction. UNet-ConvLSTM [72] utilizes convolutional LSTM to extract features from five (temporally) consecutive frames.

the lanes converge toward the vanishing point on the input image. Therefore, VPGNet [32] exploits multi-task detection that searches for the vanishing point on the input image while detecting lanes at the same time. GLCNet [69] employs a multi-task detection that detects ego lane boundaries and the lane area, simultaneously. In detail, GLCNet suggests interlinks between these two tasks to deliver the concatenated feature map of the two tasks to the two branches (i.e., tasks). Therefore, this interlink enables GLCNet to leverage the features learned from one task to improve the accuracy of the other task, particularly when one task fails to perform proper detection.

c) *Temporally consecutive feature extraction:* Lane marks are piece-wise continuous on temporally consecutive input images from a front camera. Therefore, extracting such temporally consecutive features can benefit DL-LDAs, and applying Recurrent Neural Networks (RNN), such as Long Short-Term Memory (LSTM) [70] and Gated Recurrent Units (GRU) [71], is effective in the detection of the temporally consecutive features of lanes.

U-Net-ConvLSTM [72] extracts temporally consecutive features using the convolutional LSTM (ConvLSTM) [73], as shown in Fig. 6. Instead of multilayer perceptron (MLP) in LSTM, ConvLSTM employs CNN for better image feature extraction. In detail, U-Net-ConvLSTM takes five consecutive frames (e.g., current and four previous frames) as an input to the CNN base encoder module to generate high semantic feature maps. Then, the ConvLSTM is sequentially applied to extract temporal features. Finally, the obtained feature maps (by ConvLSTM) are decoded to the original input size and lane detection is performed. ST-RNN [74] exploits temporally consecutive features acquired using the ConvLSTM, too. In fact, ST-RNN applies not only ConvLSTM but also Spatial CNN to focus on both temporally and spatially consecutive features of lanes. U-Net-ConvGRU [75] employs convolutional GRU (ConvGRU) [76], [77], [78] in place of LSTM for a better time efficiency. Since GRU has a similar structure to LSTM but uses only two gates (update and reset gates), when compared to the ConvLSTM that uses three gates (input, output, and forget gates), ConvGRU requires lower computational cost and memory, which is suitable for real-time DL-LDA.

d) *Utilization of trained algorithms:* Several studies propose methods to exploit pre-trained networks to improve the detection performance of DL-LDAs or to lessen the training cost for DL-LDAs. Transfer learning and knowledge distillation are two such methods.

Training deep learning algorithms requires a large amount of data and corresponding annotations, in general. However,

before various large-scale datasets for lane detection are published, early lane detection studies suffer from the lack of training data. To tackle such a difficulty, some studies exploit transfer learning. Transfer learning utilizes a part or full of a pre-trained neural network for similar tasks to the lane detection and we only need to train the new layers added to the pre-trained network for lane detection. For example, in the study [79], the DL-LDA applies transfer learning to tackle the training problem caused by the absence of large datasets for lane detection. In detail, the DL-LDA [79] utilizes a pre-trained network with multiple road scene segmentation datasets, such as CamVid [80], Cityscapes [81], and GTA5 [82], for the base encoder module of lane detection. This pre-trained network enables the DL-LDA to learn various road features closely related to lane detection. The experiment results shown in [79] demonstrate that the DL-LDA with transfer learning converges faster and achieves higher lane detection accuracy compared to the DL-LDAs trained from scratch.

Recently, since a number of datasets for lane detection are available, utilizing pre-trained networks to mitigate the insufficient training data problem is no longer necessary. Instead, some studies utilize pre-trained networks to optimize DL-LDA for better real-time performance. For example, ERFNet-IntRa-KD [83] and PINet [84] apply knowledge distillation [85] to build DL-LDAs for higher real-time performance. In detail, knowledge distillation trains a compact student network, that is designed to improve the real-time performance with limited computational resources, to imitate a teacher network, that has better detection performance than the student algorithm but has a complex structure. The student network learns the lane detection algorithm of the teacher network and, then, detects lanes precisely.

e) *Utilization of attention mechanism:* Recently, there has been great interest in using the attention mechanism, such as [86], [87], and [88], in deep learning networks. The attention mechanism is similar to human cognitive mechanism. When we recognize objects on an image, we first scan the entire image quickly and then make our attention to an area, where careful observation is required to recognize the objects. Similarly, the attention mechanism weights network parameters accordingly to focus on specific areas of the feature map, because of which the attention mechanism can improve DL-LDA performance.

A number of studies [89], [90], [91], [92], [93], [94], [95], [96] have introduced schemes to utilize the attention mechanisms for DL-LDAs. In general, the computational complexity of the attention mechanism is proportional to the square of the input data size. Therefore, to utilize the attention mechanism efficiently, the DL-LDAs use the downsized feature maps generated by a based encoder module, instead of using the raw images as the input to the attention mechanism. For example, [89] generates channel-wise and spatial-wise attention maps using CBAM [88] in the intermediate layers of the base encoder module, as shown in Fig. 7a. In [40], [90], [91], [92], [93], [94], and [95], transformers are applied to utilize the relationship between the local and global feature map for precise lane detection, so that lane detection performance can

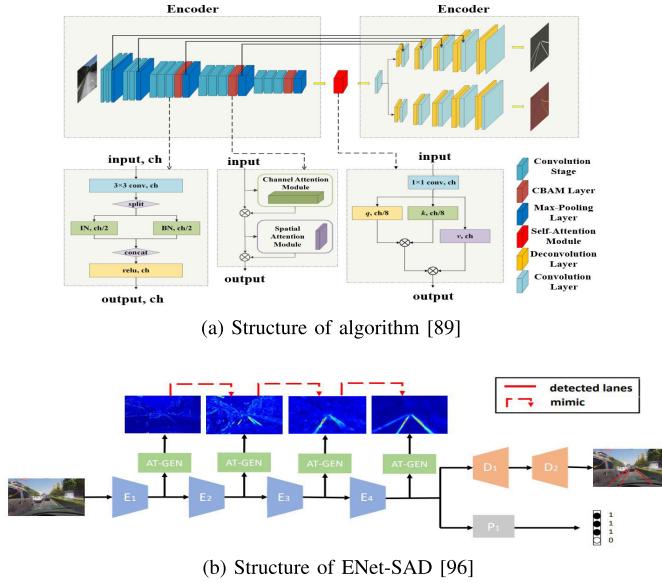


Fig. 7. Examples of attention mechanism utilization. The study [89] applies channel-wise attention and self-attention and selects the points to focus on within the feature map. On the other hand, ENet-SAD [96] trains low-level attention maps to imitate higher-level attention maps through self-attention distillation, which enables the precise generation of attention maps.

be improved even when lanes are not identified using local features alone. Similarly, LaneATT [93] and CondLaneNet [95] utilize the transformer to enhance the lane feature detection within the anchor regions on the feature map. In [96], ENet-SAD introduces self-attention distillation (SAD) to generate precise attention maps for lane detection, as shown in Fig. 7b. In general, the feature maps from shallow layers lack high semantic features, which makes shallow layers difficult to generate clear attention maps. However, in deeper layers, feature maps contain rich highly semantic features that lead to more precise attention maps than shallow layers. SAD guides shallower layers to imitate the attention maps of deeper layers, therefore, shallow layers can produce more precise attention maps.

### C. Lane Detection Head

Lane detection head is the final main process of DL-LDAs. After lane features are extracted and the output feature map is generated from the input image through lane feature extraction process, the lane detection head is applied to finally detect lanes using the output feature map. In this subsection, we analyze various methods that realize the lane detection head and classify them into the following five categories.

1) *Classification-Based Head Methods*: Before implementing deep learning algorithms to lane detection heads, LDAs such as [97] and [98] apply HCFEA-based head methods that employ HCFEAs to output final lane detection results. These HCFEA-based detection head methods require human-specified hyper-parameters that depend on the illumination and weather conditions of the input scenes for precise lane detection. As a result, these methods cannot completely distinguish whether the particular features on the output feature map correspond to lanes or not, without human aid. Moreover, these methods cannot provide robust detection performance across various input scenes, since setting the hyper-parameters that

fit all kinds of conditions is almost impossible. To overcome this problem, [67], [99], [100], [101], [102], [103] introduce classification-based head methods based on deep learning algorithms that can determine if the features pertain to the lanes without the need of human-specified parameters.

For example, the DL-LDA in [100] first extracts lane features using Sobel filtering [104] and Hough transformation [105] on the inverse perspective mapping (IPM) transformed BEV image. Then, the DL-LDA utilizes a CNN classification algorithm to determine whether the detected lane lines are dotted, solid, or double solid lines. Although this network produces the final lane detection results using deep learning algorithms, the DL-LDA still utilizes the handcrafted features generated by HCFEAs. On the other hand, Deeplanes [101] applies CNN to extract features and to identify row indices that represent the lanes on the input images from the left and right side cameras of a vehicle.

However, Deeplanes can detect only one lane for an input image. To detect multiple lanes on an input image, DL-LDAs in [99], [102], and [103] divide the output feature map into several patches, classify lane patches and non-lane patches, and connect the lane patches to detect multiple lanes at once. In detail, DVCNN [99] classifies the patches on the BEV output feature map after applying HCFEAs. The DL-LDA in [102] searches for the patches that contain lanes and applies lane fitting to the lane patches using a linear regression. Similarly, STLNet [103] divides the input BEV image into multiple patches and predicts the presence, type, and boundary of lanes in each patch using a deep learning algorithm.

2) *Segmentation-Based Head Methods*: Segmentation-based head methods utilize semantic segmentation to classify lane segments and, then, combine the lane segments to detect lanes. These methods train DL-LDAs to detect pixels of the lane marks on the output feature map which is obtained with a semantic segmentation network. To extract rich lane features for lane segmentation, [19], [32], [45], [47], [65], [69], [72], [75], [79] introduce various additional lane feature extraction modules discussed in the subsection III-B. For instance, SCNN [19] and RESA [65] employ Spatial CNN to explore additional lane features in the nearby pixels. VPGNet [32] and GLCNet [69] enhance lane detection accuracy using the multi-task learning that extracts various features from other than lanes, such as drivable area and the vanishing point. U-Net-ConvLSTM and U-Net-ConvGRU [75] utilize additional lane feature extraction modules that extract temporally consecutive features between the current and previous frames using ConvLSTM and ConvGRU, respectively. These DL-LDAs leverage the additional module to effectively enhance the lane feature extraction. However, relying only on the segmentation output may be insufficient for precise and robust lane detection, since it is often difficult for DL-LDAs to identify lane pixels, when a lane is occluded by vehicles or blurred due to the snow or rain.

To improve the precision and robustness of DL-LDAs at the same time, [66], [84], [91], [106], [107], [108], [109], [110], [111], [112] introduce optional lane fitting process, as shown in Fig. 8. Because of the optional lane fitting process, DL-LDAs can filter incorrectly predicted lane segments and fit

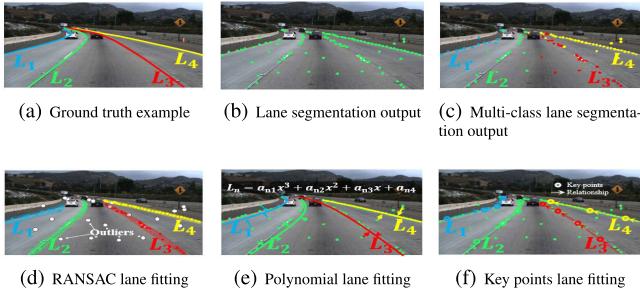


Fig. 8. Examples of optional lane fitting process; (a) represents the input image and ground truth lane annotations, (b) shows the lane segments outputs, (c) the algorithm predicts lane segments separately using multi-class segmentation, (d) RANSAC algorithm distinguishes inlier and outlier points and performs clustering, (e) polynomial coefficients are calculated with the lane segments for lane fitting, and (f) key points are selected from the lane segment, and the connections are determined from the relationship between the key points.

the incomplete shapes of lanes (such as that shown in Fig. 8c) into complete shapes as shown in Fig. 8d, 8e and 8f. For example, [106] employs RANSAC algorithm to remove false alarm pixels (i.e., pixels incorrectly predicted as lane pixels) in the lane segmentation output. EL-GAN [107] exploits GAN [48] for optional lane fitting, where the generator network is trained to create a lane fitted output image using the segmentation result and lane annotations, and the discriminator network is trained to distinguish the generator output image from the ground truth lane annotations. As a result of training the generator and discriminator adversarially, the generator can produce the precise lane fitted image.

A few studies [66], [108], [109], [110] apply polynomial lane fitting to the lane segmentation output and find coefficients of polynomials that match the lane shape, as shown in Fig. 8e. In these studies, the authors connect incomplete lane segments using polynomials and demonstrate improved DL-LDA detection accuracy even when the lanes are significantly occluded. LMD [108] conducts polynomial lane fitting with third-degree polynomials after clustering lane segments. DLFNet [109] and LaneNet [110] introduce the least squares-based polynomial lane fitting; DLFNet [109] generates weight map for each lane and performs 2D least squares-based polynomial lane fitting on the weight maps multiplied by the lane segmentation output. LaneNet [110] performs least squares-based polynomial lane fitting after transforming the lane segmentation output to the BEV image. This is because polynomial lane fitting becomes easier on the BEV image, where the lane shapes are straightened more than the lane shapes on the front-view image.

Lane fitting with a single polynomial struggles to fit zigzag and highly curved lanes. To tackle this problem, Bézier-LaneNet [113] utilizes Bézier curves for lane fitting that allows a high degree of freedom in representing a highly curved line. On the other hand, SpinNet [66] introduces multiple local polynomials for lane fitting. In detail, SpinNet generates multiple intersection points of the local polynomials and a row window around the ground truth lane pixel. Then, the intersection points are weighted using the confidence to find an (weighted) averaged point, and the averaged points from multiple rows are connected to develop a fitted lane. In a summary, this approach employs multiple local polynomials

to fit highly curved lanes, and achieves more precise detection of highly curved lanes than the conventional lane fitting using a single polynomial.

Some studies [84], [111], [112], [114] suggest key points lane fitting, which is inspired by the idea that a lane can be represented as a composition of multiple connected key points, as shown in Fig. 8f. These methods first detect key points on the output feature map and then perform lane fitting using polynomial regression or clustering algorithms. PINet [84] presents a lane detection method based on key point estimation; The key points are first identified using the obtained output feature map from the hourglass network [115] base encoder module and then used for lane detection with spline curve fitting. However, identifying and connecting the key points without considering the geometric features of the lane are not sufficient to determine accurate lane shapes.

Therefore, LaneAF [111], FoloLane [112], and RCLane [114] seek additional geometric relationships between selected key points and surrounding lanes or other key points, to improve the accuracy of key points lane fitting. In detail, LaneAF [111] generates affinity fields, that consist of horizontal and vertical fields, to find the geometric relationships between multiple key points and adjacent lanes on the output feature map. The horizontal field shows unit vectors in the direction to the adjacent lane centers, and the vertical field shows unit vectors in the direction to the next set of lane centers along the same lane. Consequently, LaneAF implements a new clustering algorithm using the affinity fields to group key points and perform lane detection. Similar to the affinity fields of LaneAF, FoloLane [112] explores geometric relationships using the local geometry construction maps that represent vectors showing the local geometry of the nearby key points in up, down, and horizontal directions. In the case of RCLane [114], relay chain prediction is introduced to fit the lane shapes, which conducts point non-maximum suppression to select key points. Since the key points are treated as relay stations, RCLane finds lane shapes by iteratively connecting these key points.

3) *Anchor-Based Head Methods:* Anchor-based head methods [37], [93], [94], [116], [117], [118] generate multiple predefined anchor lines and regress the horizontal offsets between the anchor lines and lane annotations. These methods offer several advantages over segmentation-based head methods. First, anchor-based head methods have low computational cost, since the methods perform offset regression only for a predetermined number of anchor lines, rather than classifying every pixel on the output feature map as in the segmentation-based head methods. Second, the methods can detect arbitrary number of lanes and distinguish each lane instance using separate anchor lines. Third, anchor-based head methods exhibit robustness in situations, where the lane shapes are not clear. These methods show higher robustness in challenging conditions (for example, occluded lanes by vehicles and unclear lane segments due to rain and snow) than the segmentation-based head methods, which is due to the guidance provided by pre-defined anchor lines.

LineCNN [116] adopts line proposal units that create anchor lines with various slopes for pixels at the boundaries of the

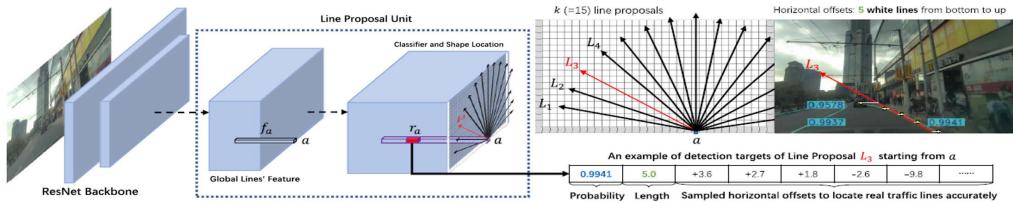


Fig. 9. Structure of the anchor-based head method used for LineCNN. LineCNN generates anchor lines for pixels at the boundaries of the output feature map and then regresses the horizontal distance to detect a lane.

output feature map, as shown in Fig. 9. LineCNN then applies  $1 \times 1$  convolutions on the feature map regions that include or are adjacent to the anchor lines to regress the horizontal offsets between the anchor lines and lane annotations. Finally, LineCNN outputs lane detection results after non-maximum suppression (NMS) operation to eliminate overlapping anchor lines based on the intersection over union (IoU) and confidence score. PointLaneNet [117] introduces anchor-based head methods for DL-LDA, too. In detail, PointLaneNet generates vertical anchor lines for every pixel on the output feature map, and calculates the horizontal offsets, lane ending positions, and the confidence score using  $1 \times 1$  convolutions for each pixel. Both studies demonstrate more robust lane detection using the anchor-based head methods than the segmentation-based head methods through experiment. However, these two DL-LDAs still have two problems: (1) The two DL-LDAs use simple straight anchor lines, which is not appropriate to regress the horizontal distance of highly-curved and zigzag lanes. (2) The two DL-LDAs regress the entire lane shapes using features from a single pixel. However, these features are often insufficient to regress the horizontal offsets for distant and long lanes, when the receptive field is not wide enough to cover an entire lane.

To overcome the two problems with the anchor-based head methods, studies [37], [93], [118] propose new ways to generate anchor lines or explore additional global features that are useful to fit the anchor lines precisely. Curvelane-NAS [37] generates multiple small anchor lines at the various levels of feature maps and clusters them to detect lanes. In detail, Curvelane-NAS utilizes anchor lines generated from the high-level (high semantic) feature maps to perform regression of the overall lane shape and other anchor lines from the low-level (high resolution) feature maps to precisely fit the curvature and local shapes of lanes. SIIC-Net [118] also exploits multiple anchor lines and introduces an eigenlane space for curve fitting with k-means clustering. As a result, Curvelane-NAS and SIIC-Net demonstrate precise detection of highly curved lanes.

In general, lanes show a regular pattern and are highly correlated with other lanes. And LaneATT [93] assumes that these global features may be crucial to fit the anchor lines especially to detect highly occluded or faded lanes. Therefore, LaneATT introduces anchor-based feature pooling to search for local features, which is similar to the RoI (Region of Interest) projection in Fast R-CNN [119]. Then, LaneATT uses an attention mechanism that find correlation (i.e., global features) between the local features of the anchor line of interest

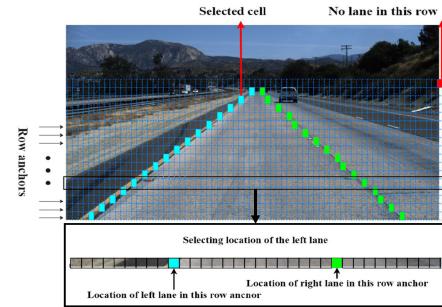


Fig. 10. Illustration of the row-wise-based head method in [122]. The method selects lane pixels in each row using MLP and uses the selected lane pixels to produce a lane detection result.

and those of the other anchor lines. Afterward, LaneATT highlights (with learnable weights) specific global features that are related to lane features, and the highlighted global features enable the DL-LDA to regress the anchor line precisely.

**4) Row-Wise-Based Head Methods:** In the forward-facing camera images, lanes appear as diagonal lines stretching from the bottom of the image to the vanishing point. These lines include pixels in every row of the input image. Row-wise-based head methods in [95], [120], [121], [122], and [123] select the specific pixels that represent a particular lane line in each row and cluster selected pixels to detect lanes, as shown in Fig. 10. The row-wise-based head methods require a selection of only one pixel per row, which makes the computational cost of DL-LDAs lower than the segmentation-based head methods and anchor-based head methods. As a result, these methods are widely used for DL-LDAs that require fast processing.

Row-wise-based head methods and anchor-based head methods are similar in that both of the methods detect lane shapes horizontally. However, while the anchor-based head methods use pre-defined lane shapes to regress the horizontal offset of each lane segment, the row-wise-head methods perform independent lane segment selection for each row without relying on pre-defined lane shapes.

The DL-LDA in [120] selects pixels in each row and connects them to detect a lane. The experiment results show that the row-wise-based head methods have an advantage in detecting long and thin shape objects (such as lane lines) over the segmentation-based head methods. In addition, the experiment results demonstrate that DL-LDAs with these methods can operate on embedded hardware (for example, Nvidia Drive PX2) because of low computational cost. In the study [121], E2E-LMD compresses the output feature map in the horizontal direction for row-wise feature extraction

and extracts additional features to select lane pixels in each row. To distinguish each lane instance, E2E-LMD constructs multiple branches that select pixels for each lane instance and generate confidence scores to determine if the selected pixel is a lane pixel. Different from the E2E-LMD, UFLD [122] flattens the entire output feature map, and applies an MLP to select lane pixels in each row to make the DL-LDA structure lighter than E2E-LMD. And UFLDv2 [123], which is a follow-up DL-LDA of UFLD [122], applies MLP to select lane pixels in row-wise and column-wise directions to enhance the lane detection accuracy of UFLD.

The DL-LDAs in [120], [121], [122], and [123] introduce row-wise-based head methods and demonstrate remarkable improvements in runtime speed. However, these methods utilize a predefined number of MLP branches to detect individual lanes, which causes two problems: (1) They can detect the same or less number of lanes than that of MLP branches. (2) Since these branches exploit the same output feature map, it is difficult to identify lane instances with the lane pixels in the same row. CondLaneNet [95] tackles these problems. CondLaneNet first generates heatmaps based on the extracted features and performs instance segmentation to detect any number of lane instances in the given input. Then, CondLaneNet regresses the dynamic kernel parameters that can be optimized for each lane instance. The trained dynamic kernel parameters for each lane are used for conditional convolution operation in each branch to generate individual feature maps, which makes CondLaneNet easily distinguish the individual lane instance. As a result, CondLaneNet improves detection accuracy over the existing methods, while maintaining low computational cost.

5) *Polynomial-Based Head Methods*: The polynomial-based head methods [124], [125], [126], [127] are different from polynomial lane fitting in the segmentation-based head methods. In detail, while the polynomial lane fitting in the segmentation-based head methods employs heuristic algorithms to produce polynomials using the lane segments in the output, the polynomial-based head methods directly predict the polynomial coefficients using the output feature map without lane segmentation. Therefore, since the polynomial-base head methods only have to output the coefficients of polynomials, these methods have low computational cost and show excellent real-time performance, which is similar to row-wise-based head methods.

PolyLaneNet [124] exploits MLP to produce lane detection results in the form of polynomials. The MLP predicts (1) the vertical coordinates of the starting and vanishing points of each lane, (2) the coefficients of the second-order polynomial equations that may represent lanes, and (3) the confidence scores of these polynomials to reduce false alarms. In [125], LSTR generates a predefined number of polynomials using the transformer [86] and utilizes Hungarian algorithm for bipartite matching to determine if the generated polynomials correspond to the presence of lanes. While PolyLaneNet and LSTR show fast runtime because of the low computational cost, both of them reveal severe degradation in detecting highly curved or zigzag lanes because of using a single second-order polynomial for a lane.

Therefore, [126], [127] propose ways to improve the detection performance of the polynomial-based head methods. For example, PRNet [126] effectively detects complex and highly curved lanes using piece-wise polynomials, which results in more accurate lane detection performance than PolyLaneNet and LSTR. In detail, instead of predicting a single polynomial for a lane, PRNet divides the lane into multiple points, calculates a piece-wise polynomial between the points, and connects them. This is a similar approach to the Curvelane-NAS [37] and SIIC-Net [118] that utilize multiple small anchor lines to detect highly curved lanes. On the other hand, MKD-E2E [127] introduces another approach to improve the detection performance further than the former polynomial-based head methods. Specifically, MKD-E2E conducts multi-task learning that trains the polynomial-based head method using knowledge distillation of the segmentation-based head method. Consequently, the polynomial-based head method of MKD-E2E can exploit the lane shape features obtainable with the segmentation-based head methods, which results in improved detection accuracy.

#### IV. STUDIES OTHER THAN 2D LANE DETECTION USING CAMERA IMAGES

In general, there are two technical problems in the DL-LDAs for 2D lane detection using camera images. First, it is difficult to acquire 3D lane shape that can be critical for safe local path planning on the winding and inclined roads.

Typically, autonomous driving systems convert the 2D lane detection result into an expression in BEV domain. However, this conversion may cause inaccurate lane shapes, particularly under conditions of uneven roads and near the vanishing points of camera images. Second, camera images are highly susceptible to poor illumination conditions, such as night-time and backlight. To tackle these problems, recent studies suggest DL-LDAs for 3D lane detection using camera images, DL-LDAs using LiDAR point cloud, and DL-LDAs using sensor fusion. In this section, we analyze DL-LDAs at the leading edge other than the 2D lane detection using camera images.

##### A. 3D Lane Detection Using Camera Images

Extracting 3D lane features from 2D camera images is a challenging task due to the lack of depth information. Nevertheless, some recent DL-LDAs can detect 3D lane shapes using the slope features of lanes that are obtained from the BEV image generated by the IPM transformation. The IPM transformation provides a matrix derived using the height and pitch angle of the installed camera sensor and represents lanes in parallel on the BEV image, assuming a flat road condition. However, for the roads with slopes, lanes are represented as trapezoidal or inverse trapezoidal shapes on the BEV image, as shown in Fig. 11. Therefore, it becomes possible for DL-LDAs to determine the 3D lane shapes when slope features of the lanes on the BEV image are obtained.

1) *3D Lane Detection With BEV Images*: The study [128] proposes the first DL-LDA for 3D lane detection using camera images. In detail, 3D-LaneNet [128] first applies the IPM transformation to the front-view feature maps (to produce

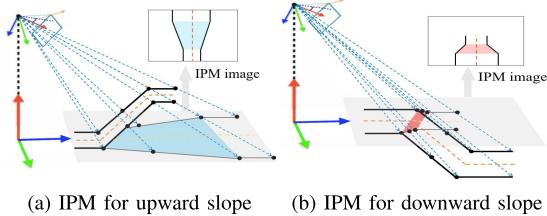


Fig. 11. Examples of lane features on the BEV image according to the lane slope. For lanes on the upward slope, the lane width on the BEV image expands as the height increases and the road appears like an inverse trapezoid, as shown in the blue region. Conversely, for lanes on the downward slope, the lane width on the BEV image narrows as the height decreases and the road appears like a trapezoid, as shown in the red region.

BEV feature maps) at the intermediate layers in the base encoder module. Then, the IPM transformed feature maps are fed into another base encoder module branch to extract 3D lane features. Finally, the DL-LDA concatenates the front-view and BEV feature maps and detects 3D lane shapes using anchor-based head method. However, since the shape of a 3D lane has a complex curvature in the 3D space, it is difficult to fit the lane using a single polynomial as in the 2D lane detection, which results in a degraded 3D lane detection accuracy. Therefore, the follow-up study, 3D-LaneNet+ [129] applies a segmentation-based head method and key points lane fitting to detect the highly curved 3D lane shapes precisely. In the study [39], Gen-LaneNet suggests another segmentation-based head method to perform better 3D lane detection than 3D-LaneNet. Specifically, Gen-LaneNet conducts lane segmentation on the front-view image with the segmentation-based head method. Then, Gen-LaneNet transforms the front-view feature map and the lane segmentation results into BEV, adds the segmentation results as the BEV masks to the transformed BEV feature map, and then extracts additional features for precise 3D lane detection. On the other hand, in [130], the authors suggest an attention mechanism called dual attention (DA) module to enhance the detection accuracy of 3D-LaneNet and Gen-LaneNet. The DA comprises lane-to-lane attention and longitudinal pixel-wise attention, where the lane-to-lane attention explores feature relationship between the current anchor lane and the other lanes, and the longitudinal pixel-wise attention explores the feature relationship between pixels within the same anchor.

These studies [39], [128], [129], [130] present novel ideas for 3D lane detection using only the BEV image from a single camera. However, the direct utilization of a BEV image encounters the following problems that cause degradation in 3D lane detection accuracy. First, DL-LDAs show difficulties in recognizing the details at distant areas on the BEV image (or feature map), since the pixels of the objects at a distance on the IPM transformed BEV image are blurred and distorted. This is because the corresponding front-view image contains fewer pixels for the distant area than the pixels for the close area to the camera. Second, since the shape of the lane marks on the BEV image may show the lane slope roughly, it is difficult to expect a high degree of accuracy in detecting lane elevation.

To tackle these problems, [40], [131], [132] introduce advanced methods. For example, in the study [131], the

authors try to mitigate the distortion problem that occurs, when Gen-LaneNet applies the IPM transformation to the lane segmentation output to generate BEV masks. Since the generated BEV mask is blurred near the vanishing point, the mask would not fit precisely with the lane mark, which results in increased uncertainty (i.e., fluctuation) in extracting precise lane shape features. As a solution, [131] suggests the geometry supervision that restricts lane mark boundaries in the IPM transformed mask to have equal thickness. Persformer [40] exploits the attention mechanism to improve the distortion due to the IPM transformation for precise 3D lane detection. In detail, Persformer supplements the BEV feature maps with additional correlation features obtained with the attention mechanism, where the BEV feature maps are used for the query and the front-view feature maps for the key and the value. BEVFormer [132] utilizes multi-view images collected by multiple cameras to produce a lane detection result with better precision than when a single BEV image is used. In addition, BEVFormer enhances the lane detection accuracy by concatenating BEV images over consecutive time frames.

*2) 3D Lane Detection Without BEV Images:* Despite the studies to predict precise lane height information, such as [40], [131], and [132], there remain inaccuracies in the BEV features caused by the IPM transformation, which hinders accurate lane height prediction. Therefore, studies in [31], [133], and [134] propose 3D lane detection methods without using inaccurate IPM transformed BEV images. SALAD [31] introduces depth estimation instead of using the IPM transformed BEV image. In detail, SALAD estimates the depth information of each lane segment and calculates 3D lane segment position through inverse projection. LATR [133] utilizes transformers for 3D lane detection from 2D camera images. Specifically, LATR generates lane segment queries using a lane-area query generator and then produces 3D lane detection result using cross-attention with dynamic 3D ground positional embedding. This embedding creates a flat 2D BEV canvas which is then projected onto a 2D image. In this process, LATR is trained to learn the projection matrix that actually enables LATR to learn 3D lane features on the input images. PETRv2 [134] introduces a 3D position encoder to learn 3D lane features, which is similar to LATR, but PETRv2 applies temporal modeling to leverage consecutive time information to extract more rich 3D features, thereby enhances 3D lane detection performance.

### B. Lane Detection Using LiDAR Point Cloud

LiDAR point cloud has multiple advantages in lane detection over camera images. One is that there is a distinctive difference in the reflectivity between the road surface and the lane marks, which is useful for lane detection. Second, the point cloud provides precise 3D information. Lastly, since LiDAR is not sensitive to illumination conditions, DL-LDAs using LiDAR point cloud show strong robustness to poor and adverse illumination conditions.

To leverage these advantages in lane detection using LiDAR point cloud, several early studies propose LDAs that employ heuristic algorithms to detect lanes from the LiDAR point cloud. For instance, [135] detects lane points with the

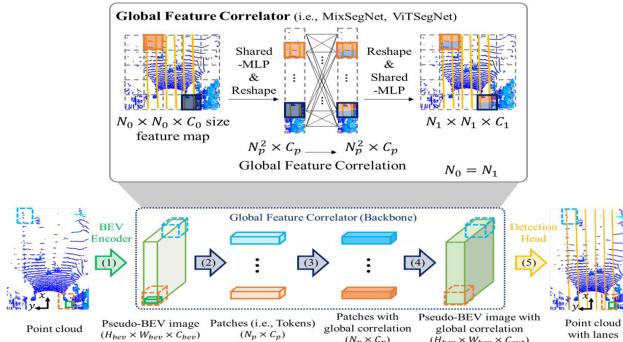


Fig. 12. Structure of LLDN-GFC [20]. LLDN-GFC introduces global feature correlator that divides feature map into multiple grids and obtains global correlation features between the grids. Then, LLDN-GFC performs lane detection using LiDAR point cloud and the global correlation features.

pre-defined threshold to the LiDAR intensity measurements, and [136] employs DBSCAN [137] to cluster lane points. However, these LDAs are not robust to various driving conditions, since determining the optimal parameters of the heuristic algorithms for all kinds of conditions is very difficult.

Therefore, the studies in [42] and [138] propose deep learning algorithms to detect lanes in the LiDAR point cloud data for the first time. RoadNet [42] aligns pre-recorded five consecutive point cloud frames to generate a dense point cloud frame, which is then used to develop a projected BEV image of the dense point cloud. Then, RoadNet applies a segmentation-based head method and polynomial lane fitting to determine lanes on the projected BEV image. DSANet [138] performs lane detection using pre-recorded point cloud data, similar to RoadNet. RoadNet and DSANet demonstrate the usefulness of the LiDAR point cloud for lane detection. However, these studies may be not suitable for real-time systems, since the algorithms require pre-recorded data.

LLDN-GFC [20] presents the first real-time DL-LDA using LiDAR point cloud. As shown in Fig. 12, LLDN-GFC inputs the projected BEV image of LiDAR point cloud into the base encoder module. Then, LLDN-GFC introduces the global feature correlator (GFC) to divide the output feature map into grid patches and to extract various global correlation features. Since the features include correlation between all of the patches, the features are used to detect the lane segments. As a result, LLDN-GFC achieves the state-of-the-art (SoTA) performance when it is published, and the authors provide the first public LiDAR point cloud lane detection dataset, K-Lane.

Since LLDN-GFC employs a segmentation-based head method, the DL-LDA has a large computational cost and accuracy degradation when there are occluded lane in the input data. The study [139], which is a follow-up study of [20], proposes the DL-LDA that has a less computational cost but, at the same time, achieves better detection accuracy than LLDN-GFC. Specifically, the DL-LDA maintains the GFC feature extractor of LLDN-GFC but replaces the segmentation-based head method with the row-wise-based head method to increase the real-time performance. Moreover, to further improve the lane detection accuracy than LLDN-GFC, the DL-DLA introduces an additional refinement head method after conducting the row-wise based head method. The

refinement head method first collects features of each lane proposal from the output feature map and generates a set of lane tokens. Then, the attention mechanism [86] is employed to find the correlations between the lane tokens. Since lanes generally have a consistent shape and distance between them, these correlations could serve as beneficial supplementary features to find the miss-detected points in each lane proposal. The DL-LDA demonstrates improved detection accuracy while having 30% less GFLOPs, compared with LLDN-GFC.

### C. Lane Detection Using Sensor Fusion

Several studies propose sensor fusion of camera and LiDAR to improve the lane detection performance. For example, in low-light conditions such as at nighttime and dawn, lane markings may not be visible with the camera images. However, DL-LDAs using sensor fusion can achieve reliable lane detection using the features found from the other sensor that compensate for the missing lane features.

Depending on where the sensor data is combined in the deep neural network, sensor fusion can be classified into early, late, and intermediate fusions. Early fusion combines the two sensor data (i.e., camera images and LiDAR point cloud) prior to the feature extraction and extracts fusion features from the early combined data. In contrast, late fusion extracts each sensor feature from separate feature extraction branches and combines the two extracted features (at a late layer) to produce fusion features. The intermediate fusion combines (or exchanges) features at an intermediate layer of each feature extraction branch for the two sensors. In general, DL-LDAs using sensor fusion employ one of the three fusion methods to improve lane detection accuracy.

The DL-LDA in [41] introduces an early fusion of camera and LiDAR to increase lane detection accuracy near the vanishing point on the BEV image. Specifically, The DL-LDA transforms the front-view camera image into the BEV image and performs early fusion between the transformed BEV image and the LiDAR point cloud. Then, the fusion features from the lane feature extraction are used to detect lanes. The experiment results demonstrate that the accurate distance information from the LiDAR point cloud near the vanishing point improves the overall lane detection accuracy.

Since early fusion directly aligns the raw data from two different sensors and inputs aligned data into a single branch of lane feature extraction, it can simplify the structure of the fusion algorithm with a low computational cost. However, this fusion method is susceptible to alignment errors between two sensor data, which may cause performance degradation of the DL-LDAs. Therefore, this method requires precise alignment of the two sensor data, which is not easy to achieve in practice.

Consequently, several studies have proposed DL-DLAs that utilize the intermediate fusion method to overcome the drawback of the early fusion method. For example, FusionLane [142] applies intermediate fusion of the camera image and the LiDAR point cloud and utilizes the segmentation-based head method, which is implemented with Deeplab V3+ [146] to output the lane detection results. In the study [143], LidCamNet also applies an intermediate

TABLE II

SUMMARY OF VARIOUS DL-LDAs. IN THE TABLE, C, S, A, R, AND P RESPECTIVELY STAND FOR CLASSIFICATION, SEGMENTATION, ANCHOR, ROW-WISE, AND POLYNOMIAL-BASED DETECTION HEAD. B AND NB STAND FOR 3D LANE DETECTION WITH AND WITHOUT BEV IMAGES RESPECTIVELY

Algorithm	Categories	Published Year	Utilizing Datasets	Instance Level Detection	Summary
EELane [67]	2D (C)	2015	Own dataset	o	Applies multi-task detection for lane detection
DVCNN [99]	2D (C)	2016	Own dataset	x	Divides the input image into several patches after applying HCFEAs
DeepLanes [101]	2D (C)	2016	Own dataset	o	Detects two ego lanes in side-view images
Lane Detection and ... [100]	2D (C)	2018	KITTI, Own dataset	o	Applies HCFEAs and CNN classification using the BEV image
Lane Marking Detection ... [102]	2D (C)	2018	KITTI	x	Detect lane patches and applies linear regression for lane fitting
SLTNet [103]	2D (C)	2018	Caltech / TuSimple	o	Divides the input BEV image into several patches and classify them
VGPNet [32]	2D (S)	2017	VPGNet	o	Employs multi-task learning that detects lanes and vanishing point
End-to-End Lane ... [79]	2D (S)	2017	Own dataset	o	Introduces transfer learning for lane detection
SCNN [19]	2D (S)	2018	CULane	o	Proposes Spatial CNN and releases CULane dataset
GLC-Net [69]	2D (S)	2018	KITTI, CULane	x	Interlinks lane detection task and drivable zone detection task
Traffic Lane Detection ... [106]	2D (S)	2018	Own dataset	x	Introduces RANSAC algorithm for optimal lane fitting process
EL-GAN [107]	2D (S)	2018	TuSimple	o	Utilizes GAN for optional lane fitting process
LMD [108]	2D (S)	2018	CamVid, KITTI	x	Applies polynomial lane fitting for optional lane fitting process
SpinNet [66]	2D (S)	2019	CULane	o	introduces line-shaped kernels to effectively extract lane features
ENet-SAD [96]	2D (S)	2019	TuSimple, CULane,	x	Generates improved attention maps using self-attention distillation
DLFNet [109]	2D (S)	2019	TuSimple	o	Employs weighted least squares for polynomial lane fitting
LaneNet [110]	2D (S)	2019	TuSimple	o	Conducts instance segmentation and least squares lane fitting
Ripple-GAN [45]	2D (S)	2020	TuSimple	x	Extracts HCFEAs features and integrates with the input image
Multitask Attention Network ... [46]	2D (S)	2020	Caltech, KITTI, TuSimple, CULane	o	Employs HCFEA for pre-processing, and Spatial CNN to extract spatial information
CycleGAN-ERFNet [47]	2D (S)	2020	CULane	o	Introduces CycleGAN to augment the input data
U-Net-ConvLSTM [72]	2D (S)	2020	TuSimple, CULane	x	Extracts temporally consecutive features using ConvLSTM
A Lane Detection ... [89]	2D (S)	2020	TuSimple	o	Employs CBAM and self-attention in lane feature extraction
AMSC [90]	2D (S)	2020	TuSimple, CULane	o	Utilizes self-attention and channel attention to enlarge the receptive fields and emphasize the important features
ERFNet-IntRa-KD [83]	2D (S)	2020	CULane, LLAMAS, ApolloScape	x	Applies knowledge distillation for smaller network
RESA [65]	2D (S)	2021	TuSimple, CULane	o	Improves Spatial CNN with recurrent feature shift aggregator
U-Net-ConvGRU [75]	2D (S)	2021	TuSimple, LLAMAS	x	Extracts temporally consecutive features in the continuous frames
ERFNet-H&VESA [92]	2D (S)	2021	TuSimple, CULane BDD100K	o	Applies extended self-attention module to recognize the lanes in occluded area
PINet [84]	2D (S)	2021	TuSimple, CULane	o	Employs hourglass base encoder module and knowledge distillation
LaneAF [111]	2D (S)	2021	TuSimple, CULane, LLAMAS	o	Proposes optional lane fitting process that utilizes vertical and horizontal affinity fields
FoloLane [112]	2D (S)	2021	TuSimple, CULane	o	Explores global geometric relationships using local geometry construction maps
GANet [91]	2D (S)	2022	TuSimple, CULane	o	Proposes Key points based lane fitting method and deformable convolution to integrate adjacent key points feature
BézierLaneNet [113]	2D (S)	2022	TuSimple, CULane LLAMAS	o	Utilizes Bézier curves for polynomial lane fitting
RCLane [114]	2D (S)	2022	TuSimple, CULane LLAMAS, CurveLanes	o	Applies NMS to get key points and relay chain approach to construct global lane shape
MLDA [140]	2D (S)	2022	TuSimple, CULane	o	Introduces domain adaptation in DL-LDA
ST-RNN [74]	2D (S)	2023	TuSimple	x	Utilizes both spatially and temporally consecutive features
Line-CNN [116]	2D (A)	2019	TuSimple, CULane	x	Proposes anchor-based head method for DL-LDA

TABLE II

(Continued.) SUMMARY OF VARIOUS DL-LDAs. IN THE TABLE, C, S, A, R, AND P RESPECTIVELY STAND FOR CLASSIFICATION, SEGMENTATION, ANCHOR, ROW-WISE, AND POLYNOMIAL-BASED DETECTION HEAD. B AND NB STAND FOR 3D LANE DETECTION WITH AND WITHOUT BEV IMAGES RESPECTIVELY

PointLaneNet [117]	2D (A)	2019	TuSimple, CULane	o	Generates anchor lines in each feature map grid pixels
CurveLane-NAS [37]	2D (A)	2020	CULane, CurveLanes	o	Proposes vertical anchor integration and point blending search
LaneATT [93]	2D (A)	2021	TuSimple, CULane, LLAMAS	x	Introduces anchor-based feature pooling and attention mechanism to find global features
CLRNet [94]	2D (A)	2022	TuSimple, CULane, LLAMAS	o	Refines the anchor lines using cross layer refinement
SIIC-Net [113]	2D (A)	2022	TuSimple, CULane, LLAMAS	x	Applies K-means clustering in eigenlane space to fit the lanes precisely
Reliable Multilane Detection ... [120]	2D (R)	2018	TuSimple	o	Exploits MLP branches to detect individual lane row-wisely
E2E-LMD [121]	2D (R)	2020	TuSimple, CULane	o	Compresses the output feature map in horizontal direction
UFLD [122]	2D (R)	2020	TuSimple, CULane	o	Flattens the output feature map, and applies MLPs row-wisely
CondLaneNet [95]	2D (R)	2021	TuSimple, CULane, CurveLanes	o	Applies conditional convolution that utilizes dynamic kernels
UFLDv2 [123]	2D (R)	2022	TuSimple, CULane, LLAMAS, CurveLanes	o	Selects lane pixels in both row-wise and column-wise direction
PRNet [126]	2D (P)	2020	TuSimple, CULane	o	Predicts piece-wise polynomials and connects them
PolyLaneNet [124]	2D (P)	2021	TuSimple, CULane, LLAMAS, ELAS	o	Predicts second-order polynomial coefficients to detect lanes
LSTR [125]	2D (P)	2021	TuSimple	o	Generate polynomials using transformer and applies Hungarian algorithm to find the existing lanes
MKD-E2E [127]	2D (P)	2023	TuSimple, CULane, CurveLanes	o	Employs knowledge distillation between segmentation and polynomial-based head methods
3D-LaneNet [128]	3D (B)	2019	Own dataset	o	Explores BEV lane features and regress $x$ , $y$ and $z$ using anchor-based head method
Gen-LaneNet [39]	3D (B)	2020	Synthetic-3D-lanes	o	Generates virtual BEV image and introduces two-stage approach for sophisticated 3D lane detection
3D-LaneNet+ [129]	3D (B)	2020	Own dataset	o	Extracts the 3D lane features in corresponding feature map grids
Robust Monocular 3D ... [130]	3D (B)	2021	Synthetic-3D-lanes	o	Applies dual attention module which consists of lane-to-lane attention and longitudinal pixel-wise attention
Reconstruct from Top ... [131]	3D (B)	2022	Synthetic-3D-lanes	o	Obtains 3D lane features using feature projection
PersFormer [40]	3D (B)	2022	OpenLane	o	Introduces cross attention to enhance the 3D lane features
BEVFormer [132]	3D (B)	2023	Other dataset	o	Utilizes multi-view BEV images
SALAD [31]	3D (NB)	2022	Synthetic-3D-lanes, ONCE-3DLanes	o	Detects 3D lane using lane segmentation and depth estimation
LATR [133]	3D (NB)	2023	Synthetic-3D-lanes, OpenLane	o	Applies dynamic 3D ground positional embedding
PETRv2 [134]	3D (NB)	2023	OpenLane	o	Introduces 3D position encoder and temporal modeling
Fast LiDAR-based ... [141]	LiDAR	2017	KITTI / Own dataset	x	Utilizes the point cloud projected BEV image for lane detection
LiDAR-based Deep ... [42]	LiDAR	2020	Own dataset	o	Classifies lane segments from the point cloud projected BEV image and determines lane shapes through the polynomial fitting
LLDN-GFC [20]	LiDAR	2022	K-Lane	o	Introduces global feature correlator to segment the lane points
2D (R) LiDAR ... [139]	LiDAR	2022	K-Lane	o	Applies row-wise-based head method to improve computation and detection performance
DSANet [138]	LiDAR	2023	Own dataset	o	Employs Dual-level shape attention to consider both local and global shapes in lane feature extraction
Deep Multi-Sensor ... [41]	Fusion	2018	DeepLane	x	Transforms the front-view camera image into a BEV to integrate with LiDAR point cloud.
FusionLane [142]	Fusion	2020	Own dataset	x	Performs lane and road mark detection using segmentation-based head method using camera and LiDAR sensors data
LidCamNet [143]	Fusion	2019	Own dataset	x	Integrates the camera image and LiDAR point cloud by projecting the point cloud into the image
BEVFusion [144]	Fusion	2023	Other dataset	o	Fuses Lidar feature maps and BEV feature maps
CFECA [145]	Fusion	2021	Own dataset	x	Applies efficient channel-wise attention module to the fusion of camera and LiDAR data

fusion of the front-view camera image and the LiDAR point cloud image that are obtained by projecting the LiDAR point cloud into the 2D front-view image plane. In particular, LidCamNet exchanges features from the separate base encoder modules for the two sensor inputs through the cross-connections. BEVFusion [144] transforms the features extracted from multi-view camera images to the BEV domain, concatenates the extracted features with the features from the LiDAR point cloud, and extracts fused BEV features from the concatenated features to finally detect lanes. In a summary, FusionLane, LidCamNet and BEVFusion can flexibly integrate features from two different sensors using the intermediate fusion method and can produce robust lane detection results even in the presence of sensor alignment errors.

All of the sensor fusion methods above simply concatenate raw data or feature maps at the intermediate layers. However, the concatenation operation does not consider the relationship between the features of different sensors. In fact, the relationship features are useful to identify important features on the fusion feature map. Recently, the attention mechanism shows high usefulness in extracting the relationship features and weighting the area corresponding to the important features on the feature map. Therefore, considering the relationship features to which a DL-LDA should pay more attention, sensor fusion using the attention mechanism can produce more comprehensive fusion features. CFECA [145] introduces a fusion method that applies channel-wise attention to extract fusion features of camera images and LiDAR point cloud data and demonstrates performance improvement.

## V. EVALUATION METRICS AND PERFORMANCE ANALYSIS USING PUBLIC LANE DETECTION BENCHMARKS

In this section, we introduce widely used evaluation metrics in the recent studies for the performance comparison of DL-LDAs. Furthermore, we discuss the most updated three public lane detection benchmarks and a brief description of DL-LDA performance improvements.

### A. Evaluation Metrics

The TuSimple [35] benchmark metric, introduced for the lane detection challenge in 2017, has been used as an official evaluation metric for DL-LDAs. The metric evaluates DL-LDA performance in two aspects: (1) pixel-level detection (*Accuracy*) and (2) lane-level detection (*FPR*, *FNR*, *F1* score).

Pixel-level detection metric is to evaluate how well a DL-LDA has detected pixels corresponding to lanes in the input data. To evaluate the pixel-level detection performance, *Accuracy* is calculated as follows.

$$\text{Accuracy} = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}}. \quad (1)$$

where  $C_{clip}$  and  $S_{clip}$  represent the lane pixels predicted by the DL-LDA and the corresponding ground truth lane pixels, respectively. The TuSimple benchmark metric considers predicted lane pixels as correct if the pixels are within a specified threshold to the lane annotation. Then, the metric sums up the

number of correctly predicted lane pixels across all clips and divides the sum by the total number of pixels in the ground truth lanes to determine *Accuracy* as in (1).

On the other hand, the lane-level detection metric uses false positive rate (*FPR*), false negative rate (*FNR*), and *F1* score to determine whether a DL-LDA has recognized lanes correctly based on the pixel-level detection result. To figure out the *FPR*, *FNR*, and *F1* score, the metric first compares the rate of correctly detected lane pixels to total pixels in the predicted lane with a predefined threshold. And when the rate is higher than the threshold, the metric regards the predicted lane as detected (i.e., true positive), and otherwise the predicted lane is regarded as falsely detected (i.e., false positive) or missed detection (i.e., false negative). Therefore, the metric can count the total number of true positive lanes (*TP*), false negative lanes (*FN*), and false positive lanes (*FP*). Consequently, the total number of predicted lanes  $N_{pred}$  and the total number of ground truth lanes  $N_{gt}$  are calculated as following.

$$N_{pred} = TP + FP \quad (2)$$

$$N_{gt} = TP + FN. \quad (3)$$

Then, the metric calculates the *FPR* and *FNR* that represent the rate of *FP* in  $N_{pred}$  and the rate of *FN* in  $N_{gt}$ , respectively, as below.

$$FPR = \frac{FP}{N_{pred}} \quad (4)$$

$$FNR = \frac{FN}{N_{gt}}. \quad (5)$$

Subsequently, the benchmark metric calculates *Precision*, *Recall*, and finally determines the *F1* score as following.

$$Precision = \frac{TP}{N_{pred}} = 1 - FPR \quad (6)$$

$$Recall = \frac{TP}{N_{gt}} = 1 - FNR \quad (7)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (8)$$

The CULane [19] and LLAMAS [33] benchmarks, released in 2018 and 2019, respectively, adopt the TuSimple benchmark metric. However, the detection performance *F1* score is quantified with the *Precision* and *Recall* in a slightly different way from the TuSimple benchmark. Specifically, the two benchmark metrics of the CULane and LLAMAS utilize lane IoU in the determination of *TP*, *FP*, and *FN*. Since lane IoU can recognize the area mismatch between the ground truth lane and the predicted lane, these two metrics can determine the rate of success or failure of the detection, which provides more reasonable performance evaluation than the rate of the number of correctly predicted lane pixels to the number of total pixels in the detected lane. However, as lanes have a line (i.e., long and thin) shape, determining the exact IoU is a difficult task. Therefore, the CULane and LLAMAS benchmark metrics assume a 30-pixel thickness of lane lines for both predicted lane lines and the ground truth lane lines, which is then used for the lane IoU calculation. In general, a predefined threshold is used to distinguish positive lanes from negative lanes, and in the CULane benchmark, the lane

TABLE III

TABLE OF TUSIMPLE DATASET BENCHMARK [35], WHERE S, M, L, AND D IN THE ALGORITHMS REPRESENT RESNET-18, RESNET-34, RESNET-122, AND DLANET-34, RESPECTIVELY. THE TOP 5 VALUES ARE EMPHASIZED IN BOLD FONTS

Algorithm	Year	Accuracy	False Positive (FPR)	False Negative (FNR)	F1 Score	Runtime (ms) / FPS (Hz)	Computing resource
SCNN [19]	2018	96.53	0.0617	<b>0.0180</b>	95.97	133 / 8.9	GTX Titan black
EL-GAN [107]	2018	96.39	0.0412	0.0336	96.26	100 / 10	Unknown
ISS-based [96]	2018	96.50	0.0851	0.0269	94.31	-	Unknown
ENet-SAD [96]	2019	96.64	0.0602	0.0205	95.92	13.3 / 75	GTX Titan X
Line-CNN [116]	2019	<b>96.87</b>	0.0442	0.0197	96.79	33.3 / 30	GTC Titan X
PointLaneNet [117]	2019	96.34	0.0467	0.0518	95.07	14.1 / 71	GTX 1080 ti
LaneNet [110]	2019	96.38	0.0780	0.0244	94.80	19.0 / 52.6	Unknown
U-Net-ConvLSTM [72]	2019	<b>97.30</b>	0.0416	<b>0.0186</b>	96.98	5.80 / 172	GTX Titan X
PINet [84]	2020	96.75	0.0310	0.0250	97.20	33.3 / 30	RTX 3090
E2E-LMD [121]	2020	96.02	0.0321	0.0428	96.25	-	Unknown
UFLD-S [122]	2020	95.87	<b>0.0191</b>	0.0392	87.87	<b>3.2 / 313</b>	GTX 1080 ti
UFLD-M [122]	2020	96.06	<b>0.0190</b>	0.0385	87.91	5.9 / 170	GTX 1080 ti
RESA [65]	2020	96.82	0.0363	0.0248	96.94	-	RTX 2080 ti
Multitask Attention ... [46]	2020	96.51	0.2393	0.0316	85.21	-	GTX 1080 ti
PRNet [126]	2020	<b>97.18</b>	0.0397	<b>0.0172</b>	97.14	9.1 / 110	GTX 1080 ti
FoloLane [112]	2021	<b>96.92</b>	0.0447	0.0238	96.61	25 / 40	Tesla V100
LaneATT-S [93]	2021	95.57	0.0356	0.0301	96.71	<b>4 / 250</b>	RTX 2080 ti
LaneATT-M [93]	2021	95.63	0.0353	0.0291	96.78	5.8 / 171	RTX 2080 ti
LaneATT-L [93]	2021	96.10	0.0564	0.0217	96.06	38.5 / 26	RTX 2080 ti
CondLaneNet-S [95]	2021	95.48	0.0218	0.0380	97.00	-	RTX 2080
CondLaneNet-M [95]	2021	95.37	0.0220	0.0382	96.98	-	RTX 2080
CondLaneNet-L [95]	2021	96.54	0.0201	0.0350	97.24	-	RTX 2080
PolyLaneNet [124]	2021	93.36	0.0942	0.0933	90.62	8.7 / 115	GTX Titan V
LSTR [125]	2021	96.18	0.0291	0.0338	96.85	<b>2.4 / 420</b>	GTX 1080 ti
LaneAF-D [111]	2021	95.62	0.0280	0.0418	96.51	-	GTX Titan X maxwell
CLRNet-S [94]	2022	96.84	0.0228	<b>0.0192</b>	<b>97.89</b>	8.4 / 119	GTX 1080 ti
CLRNet-M [94]	2022	<b>96.87</b>	0.0227	0.0208	<b>97.82</b>	9.7 / 103	GTX 1080 ti
CLRNet-L [94]	2022	96.83	0.0237	0.0238	97.62	21.7 / 46	GTX 1080 ti
GANet-S [91]	2022	95.95	<b>0.0197</b>	0.0262	<b>97.70</b>	6.5 / 153	Tesla V100
GANet-M [91]	2022	95.87	<b>0.0199</b>	0.0264	<b>97.68</b>	7.9 / 127	Tesla V100
GANet-L [91]	2022	96.44	0.0263	0.0247	97.45	15.9 / 63	Tesla V100
MLDA [140]	2022	89.70	0.0295	<b>0.0184</b>	97.60	-	GTX 1080 ti
ERPNet-H&VESA [92]	2022	99.12	0.0331	0.0450	96.09	8.1 / 123	RTX 2080 ti
SIC-Net [118]	2022	95.62	0.0320	0.0399	96.40	10 / 100	Unknown
UFLDv2-S [123]	2022	95.65	0.0306	0.0461	96.16	<b>3.2 / 312</b>	RTX 3090
UFLDv2-M [123]	2022	95.56	0.0318	0.0437	96.22	5.9 / 169	RTX 3090
RCLane-S [114]	2022	96.49	0.0221	0.0257	97.52	21.9 / 45.6	Unknown
RCLane-M [114]	2022	96.51	0.0224	0.0236	97.61	22.8 / 43.8	Unknown
RCLane-L [114]	2022	96.58	0.0228	0.0227	<b>97.64</b>	40.8 / 24.5	Unknown
MKD-E2E-S [127]	2023	95.96	0.0229	0.0430	96.69	<b>2.8 / 355</b>	RTX 2080 ti
MKD-E2E-M [127]	2023	96.27	<b>0.0179</b>	0.0380	97.19	<b>4.7 / 215</b>	RTX 2080 ti

TABLE IV

TABLE OF LLAMAS DATASET BENCHMARK [33], WHERE S, M, L, AND D IN THE ALGORITHMS REPRESENT RESNET-18, RESNET-34, RESNET-122, AND DLANET-34, RESPECTIVELY. THE TOP 3 VALUES ARE EMPHASIZED IN BOLD FONTS

Algorithm	Year	F1 Score	Precision	Recall
LaneAF-D [111]	2021	<b>96.07</b>	<b>96.91</b>	95.26
PolyLaneNet [124]	2021	88.40	88.87	87.93
LaneATT-S [93]	2021	93.46	96.62	90.24
LaneATT-M [93]	2021	93.74	<b>96.79</b>	90.88
LaneATT-L [93]	2021	93.54	<b>96.82</b>	95.26
UFLDv2-S [123]	2022	94.58	95.29	93.88
UFLDv2-M [123]	2022	94.95	95.75	94.17
CLRNet-S [94]	2022	96.00	-	-
CLRNet-D [94]	2022	<b>96.12</b>	-	-
BézierLaneNet-S [113]	2022	94.91	95.71	94.13
BézierLaneNet-M [113]	2022	95.17	95.89	94.46
RCLane-S [114]	2022	96.05	<b>96.70</b>	<b>95.42</b>
RCLane-M [114]	2022	96.03	96.62	<b>95.45</b>
RCLane-L [114]	2022	<b>96.13</b>	<b>96.79</b>	<b>95.48</b>

IoU threshold of 0.5 is set. Furthermore, to evaluate the performance in various road conditions comprehensively, the CULane benchmark metric calculates individual *F1* scores for various scenes (e.g., normal, crowded, night, etc.), as shown in Table V.

### B. Public Lane Detection Benchmarks

Table III, IV, and V present the most widely used benchmarks for 2D lane detection using camera images, TuSimple [35], LLAMAS [33], and CULane [19], respectively. And

Table VI presents the Synthetic 3D Lanes dataset benchmark for 3D lane detection using camera images. Unfortunately, due to the lack of diverse experiment results, we cannot conduct a comprehensive analysis of the benchmarks for lane detection using LiDAR point cloud and lane detection using sensor fusion. However, we expect that these areas of study will also be actively conducted, and various study performance can be systematically compared and evaluated using public benchmarks in the near future.

Table III, IV, V, and VI show the experiment results of various DL-LDAs that are listed in the chronological order of the paper publication. Note that some of the DL-LDAs in Table II, that do not provide experiment results using the official metrics in subsection V-A, are excluded from the tables. Additionally, the runtime and FPS presented in the tables are estimated based on the information provided in the respective papers. Generally, runtime and FPS can vary according to the hardware, algorithm implementation, and memory management. Therefore, it is not advisable to compare DL-LDA running speed based solely on the runtime and FPS shown in Table III, IV, V, and VI. To facilitate a relative comparison of the runtime and FPS of the DL-LDAs, Table III and V list the use of GPU hardware resources in the last column.

TuSimple benchmark has been used by a number of studies to evaluate DL-LDAs, as shown in Table III. However, the TuSimple benchmark has a relatively small number of frames (6,408 frames) only for highways, which may cause training DL-LDAs prone to overfit. Therefore, most evaluation results of DL-LDAs with TuSimple benchmark show high accuracy, and the performance enhancements are not dramatically noticeable over time. Therefore, TuSimple benchmark maybe not useful to evaluate the effectiveness of the proposed ideas of DL-LDA studies. Nevertheless, recent algorithms have shown gradual improvement in both *FPR* and *FNR*, which leads to an overall improvement in *F1* score.

The LLAMAS benchmark, shown in Table IV, has about 100,000 frames for highways, which makes this benchmark more useful to evaluate the general performance of DL-LDAs than TuSimple. Up to present, the number of studies using the LLAMAS benchmark has been relatively small compared to other benchmarks, which makes it difficult to comprehensively compare DL-LDA performance. However, since recent studies actively using LLAMAS benchmark, this benchmark is expected to be widely used for DL-LDA performance evaluation on the highways in the future.

The CULane benchmark in Table V includes various road conditions, such as crowded and night-time, and, therefore, the CULane benchmark can be useful for performance evaluation and comprehensive comparison of DL-LDAs. The example outputs of various DL-LDAs are demonstrated in the bottom of Fig 13. The DL-LDAs with segmentation-based head methods inherently require high computational cost and, consequently, longer runtime. In addition, because segmentation-based head methods do not often consider various geometric features of lanes, the methods show lower detection accuracy than the other head methods. However, it is found that segmentation-based head methods can handle various lane semantic features, and, therefore, the DL-LDAs

TABLE V

TABLE OF CULANE DATASET BENCHMARK [19], WHERE S, M, L, AND D IN THE ALGORITHMS REPRESENT RESNET-18, RESNET-34, RESNET-122, AND DLANET-34, RESPECTIVELY. THE TOP 5 VALUES ARE EMPHASIZED IN BOLD FONTS

Algorithm	Year	Total	Normal	Crowded	Night	No line	Shadow	Arrow	Dazzle light	Curve	Crossroad (FP)	Runtime (ms)	FPS (Hz)	Computing resource
SCNN [19]	2018	71.60	90.60	69.70	66.10	43.40	66.90	84.10	58.50	64.40	1990	176.0	5.7	GTx Titan black
GLCNet [69]	2018	73.10	89.70	76.50	68.70	35.10	65.50	82.20	67.40	63.20	-	-	-	GTx Titan X
SpinNet [66]	2019	74.20	90.50	71.70	68.40	43.20	72.90	85.00	62.00	50.70	-	-	-	GTx 1080 ti
ENet-SAD [96]	2019	70.80	90.10	68.80	66.00	41.60	65.90	84.00	60.20	67.70	1998	13.4	74.6	GTx Titan X
CycleGAN-ERFNet [47]	2020	73.90	91.80	71.80	69.40	46.10	76.20	87.80	66.40	67.10	2346	-	-	Unknown
PINet [84]	2020	74.40	90.30	72.30	67.70	49.80	68.40	83.70	66.30	65.60	1427	30.3	33.0	RTX 3090
E2E-LMD [121]	2020	74.00	91.00	73.10	67.90	46.60	74.10	85.80	64.50	71.90	2022	-	-	Unknown
UFLD-S [122]	2020	68.40	87.70	66.00	62.10	40.20	62.80	81.00	58.40	57.90	1743	<b>3.1</b>	<b>322.6</b>	GTx 1080 ti
UFLD-M [122]	2020	72.30	90.70	70.20	66.70	44.40	69.30	85.70	59.50	69.50	2037	5.7	175.4	GTx 1080 ti
Multitask Attention ... [46]	2020	-	90.20	69.70	67.30	44.70	68.50	84.80	59.70	69.60	1933	-	-	GTx 1080 ti
CurveLane-S [37]	2020	71.40	88.30	68.60	66.20	47.90	68.00	82.50	63.20	66.00	2817	-	-	Tesla V100
CurveLane-M [37]	2020	73.50	90.20	70.50	68.20	48.80	69.30	85.70	65.90	67.50	2359	-	-	Tesla V100
CurveLane-L [37]	2020	74.80	90.70	72.30	68.90	49.40	70.10	85.80	67.70	68.90	1746	-	-	Tesla V100
FoloLane [112]	2021	78.80	92.70	77.80	74.50	52.10	79.30	89.00	<b>75.20</b>	69.40	1569	25.0	40.0	Tesla V100
LaneATT-S [93]	2021	75.09	91.11	72.96	68.95	48.35	70.91	85.49	65.72	63.37	1170	<b>4.0</b>	<b>250.0</b>	RTX 2080 ti
LaneATT-M [93]	2021	76.68	92.14	75.03	70.72	49.39	78.15	88.38	66.47	67.72	1330	5.8	172.4	RTX 2080 ti
LaneATT-L [93]	2021	77.02	91.74	76.16	70.81	50.46	76.31	86.29	69.47	64.05	1264	38.5	26.0	RTX 2080 ti
CondLaneNet-S [95]	2021	78.14	92.87	75.79	73.23	52.39	80.01	89.37	70.72	72.40	1364	<b>4.5</b>	<b>220.3</b>	RTX 2080
CondLaneNet-M [95]	2021	78.74	93.38	77.14	73.92	51.85	79.93	89.89	71.17	73.88	1387	6.6	152.0	RTX 2080
CondLaneNet-L [95]	2021	79.48	93.47	77.44	74.80	<b>54.13</b>	<b>80.91</b>	90.16	70.93	75.21	1201	17.2	58.1	RTX 2080
RESA [65]	2021	75.30	92.10	73.10	69.90	47.70	72.80	88.30	69.20	70.30	1530	28.0	35.7	RTX 2080 ti
LaneAF-D [111]	2021	77.41	91.80	75.61	73.03	51.38	79.12	86.88	71.78	72.70	1360	-	-	GTx Titan X maxwell
BézierLaneNet-S [113]	2022	73.67	90.22	71.55	68.70	45.30	70.91	84.09	62.49	58.98	<b>996</b>	4.7	212.8	RTX 2080 ti
BézierLaneNet-M [113]	2022	75.57	91.59	73.20	69.90	48.05	76.74	87.16	69.20	62.45	<b>888</b>	6.7	149.3	RTX 2080 ti
CLRNet-S [94]	2022	79.58	93.30	78.33	<b>75.11</b>	53.14	79.66	<b>90.25</b>	<b>73.71</b>	71.56	1321	8.4	119.0	GTx 1080 ti
CLRNet-D [94]	2022	<b>80.47</b>	<b>93.73</b>	<b>79.59</b>	<b>75.51</b>	<b>54.48</b>	<b>82.51</b>	<b>90.62</b>	<b>75.30</b>	74.13	1155	10.6	94.3	GTx 1080 ti
GANet-S [91]	2022	78.79	93.24	77.16	72.75	53.59	77.88	89.62	71.24	75.92	1240	6.5	153.0	Tesla V100
GANet-M [91]	2022	79.39	<b>93.73</b>	77.92	73.67	52.63	79.49	<b>90.37</b>	71.64	<b>76.32</b>	1368	7.9	127.0	Tesla V100
GANet-L [91]	2022	<b>79.63</b>	<b>93.67</b>	<b>78.66</b>	73.85	53.38	78.32	89.86	71.82	<b>77.37</b>	1352	15.9	63.0	Tesla V100
PRNet-S [126]	2020	74.80	90.80	72.30	69.20	47.60	70.60	85.20	64.20	67.20	<b>1113</b>	9.0	111.1	GTx 1080 ti
PRNet-ERFNet [126]	2020	76.40	92.00	74.70	70.50	51.70	76.00	87.80	68.40	70.00	2114	14.3	69.9	GTx 1080 ti
ERFNet-H&VESA [92]	2022	74.20	92.00	73.10	69.50	45.80	75.10	88.10	63.10	68.80	2001	8.1	123.5	RTX 2080 ti
SIIC-Net [118]	2022	77.20	91.70	76.00	71.80	52.20	74.10	87.70	69.80	62.90	1509	10.0	100.0	Unknown
UFLDv2-S [123]	2022	74.70	91.80	73.30	65.30	47.60	75.10	87.90	65.30	68.50	2075	<b>3.2</b>	<b>312.5</b>	RTX 3090
UFLDv2-M [123]	2022	76.00	92.50	74.80	70.80	49.20	75.50	88.80	65.50	70.10	1910	6.4	156.3	RTX 3090
RCLane-S [114]	2022	79.52	93.41	77.93	74.33	<b>53.84</b>	80.31	89.04	<b>73.32</b>	75.66	1298	21.9	45.7	Unknown
RCLane-M [114]	2022	<b>80.03</b>	93.59	<b>78.77</b>	73.96	52.77	<b>84.37</b>	<b>90.31</b>	72.44	<b>78.39</b>	<b>907</b>	22.8	43.9	Unknown
RCLane-L [114]	2022	<b>80.50</b>	<b>94.01</b>	<b>79.13</b>	<b>75.10</b>	<b>53.94</b>	<b>81.16</b>	<b>90.51</b>	<b>72.92</b>	<b>79.66</b>	<b>931</b>	40.8	24.5	Unknown
MKD-E2E-S [127]	2023	78.57	93.09	77.63	<b>75.07</b>	58.59	76.87	85.98	57.60	73.44	2463	<b>2.8</b>	<b>358.2</b>	RTX 2080 ti
MKD-E2E-M [127]	2023	<b>79.69</b>	<b>93.62</b>	<b>78.93</b>	<b>76.36</b>	<b>61.81</b>	<b>80.56</b>	88.58	61.41	<b>76.56</b>	3102	4.7	214.8	RTX 2080 ti

TABLE VI

TABLE OF SYNTHETIC 3D LANES DATASET BENCHMARK [39]. THE TOP VALUES ARE EMPHASIZED IN BOLD FONTS

Algorithm	Balanced Scenes	Rarely observed scenes	Scenes with visual variations
	<i>F1</i> Score / AP	<i>F1</i> Score / AP	<i>F1</i> Score / AP
3D-LaneNet [128]	86.4 / 89.3	72.0 / 74.6	72.5 / 74.9
Gen-LaneNet [39]	88.1 / 90.1	78.0 / 79.0	85.3 / 87.2
Robust Monocular 3D ... [130]	91.0 / 93.2	84.1 / 85.8	85.4 / 87.4
Performer [40]	<b>92.9</b> / -	87.5 / -	<b>89.6</b> / -
Reconstruct from Top ... [131]	92.8 / <b>94.7</b>	<b>87.8</b> / <b>89.5</b>	87.2 / <b>89.1</b>

with the segmentation-based head methods have small *FP* in the crossroad scenes, where lanes look different from normal lanes.

On the other hand, row-wise-based head methods and polynomial-based head methods show significant improvement in runtime, because pixel-wise segmentation is not required. In particular, UFLD [122] and LSTR [125] achieve the best runtime, while the detection accuracy is nearly comparable to the conventional segmentation-based head methods. CondLaneNet [95] also applies the row-wise-based head method and utilizes lane-specific dynamic kernel parameters in the conditional convolution operations for lane feature extraction, which achieves both desirable runtime and detection accuracy as shown in the last column of Fig. 13. In the case of anchor-based head methods, such

as CurveLaneNas [37], LaneATT [93], and CLRNet [94], improvement in both runtime and detection performance compare to segmentation-based head methods are observed. In particular, CLRNet applies cross-layer refinement to anchor lines, which allows utilization of both low-level and high-level semantic features. These semantic features enable CLRNet to refine the anchor line more precisely than other anchor-based head methods. As a result, CLRNet achieves remarkable lane detection performance in the CULane benchmark as demonstrated in the fourth column of Fig. 13 and, at the same time, shows impressive runtime performance.

The Synthetic 3D Lanes benchmark in Table VI evaluates 3D DL-LDAs using camera images. The benchmark compares the *F1* score and the Average Precision (AP), which represents the area under the *Precision* curve according to the *Recall*. In the table, balanced scenes include typical environments, rarely observed scenes contain complex urban road environments, and scenes with visual variations cover environments that occur illumination change, such as at dawn and twilight. As demonstrated in Fig. 14, since lanes are represented in different shapes on the BEV image based on the lane slope, 3D-LaneNet [128] utilizes these slope features of the lanes to perform 3D lane detection. Gen-LaneNet [39] exploits front-view image features additional to BEV image therefore, achieves better lane detection performance compared to 3D-LaneNet, as shown in the second column of Fig. 14. Performer [40] introduces attention mechanisms to enhance 3D lane detection accuracy further. The DL-LDA [131]

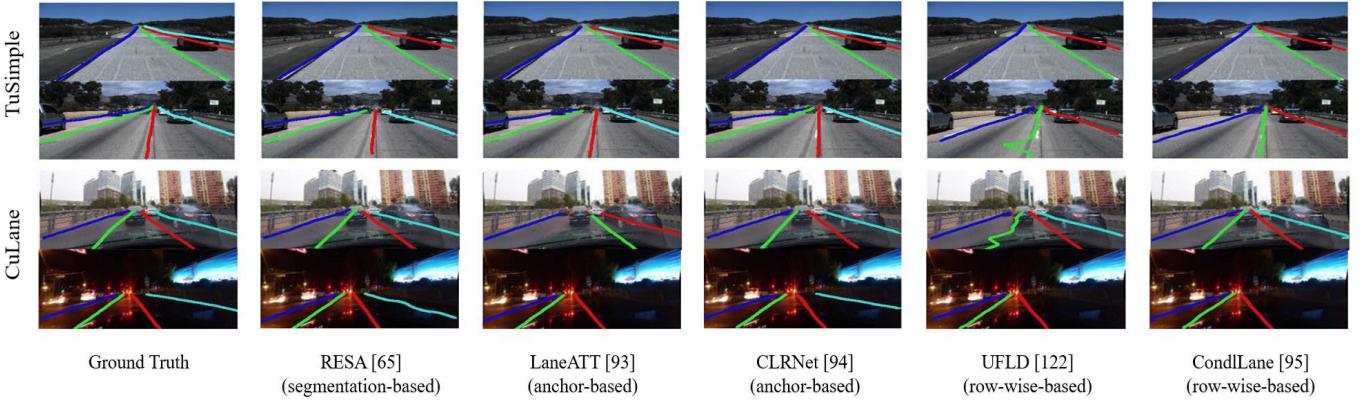


Fig. 13. The example outputs of various 2D DL-LDAs using camera images. The images are from [94]. The top two rows demonstrate sample frames from the TuSimple dataset, and the remaining rows show the CULane dataset. Each lane instance is painted in a different color. DL-LDAs with segmentation-based head methods, such as RESA, can accurately recognize lanes in environments with diverse semantic features. However, they face challenges identifying lane segments when there are lane occlusions by vehicles. On the other hand, anchor-based head methods like LaneATT and CLRNet can detect lanes even in environments lack of semantic information with the assistance of pre-defined anchor lines. Row-wise head methods, such as UFLD and CondLane, show fast runtime. But, as shown in the fifth column, UFLD may produce inaccurate lane shapes, as they select individual lane segments per row independently.

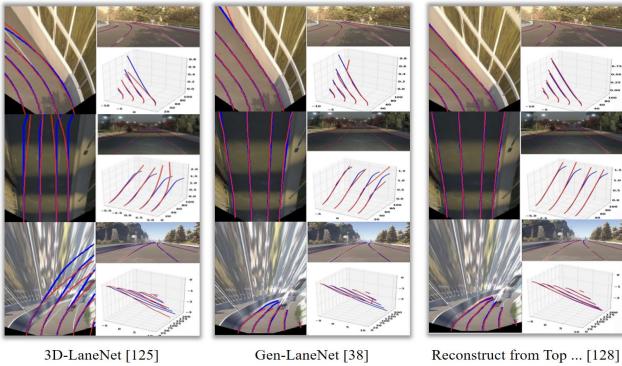


Fig. 14. The example outputs of three different 3D DL-LDAs using camera images. The images are from [39]. Each row shows example frames in the 3D Synthetic Lanes dataset while each column displays the results from DL-LDAs. In the result, the left image represents the BEV image, the top right is the front-view image, and the bottom right displays the 3D lane detection results. Ground truth is marked in blue, and inferences are indicated with red lines. In the BEV image of each study, the ground truth lines are generated with different IPM methods, therefore the shapes of the ground truth lines are different, however the 3D lane information is the same in all three studies.

handles the distortion that occurs during IPM transformation and achieves remarkable 3D lane detection performance in the benchmark, as demonstrated in the last column of Fig. 14.

## VI. TECHNICAL ISSUES TO BE ADDRESSED

The study [14], published in 2018, emphasizes the absence of suitable lane detection datasets for DL-LDA development. However, recent large-scale datasets and standardized benchmark metrics have resolved these issues, and the advancements in deep learning algorithms have significantly improved the performance of DL-LDAs. As a result, DL-LDA studies have made significant strides and become active in recent years. Nevertheless, there are still technical issues that need to be improved or resolved, and we present the technical issues to be addressed in the future DL-LDA studies in this section.

### A. Generalized Lane Detection Performance Across Datasets

In a number of studies, DL-LDAs are trained using only one or two datasets, which causes the DL-LDAs to overfit

specific environments. For this reason, expecting the same or similar performance of the trained DL-LDAs for another lane detection dataset can be impractical. This problem becomes evident when the datasets for training and testing are from different countries, as there can be differences on the road, for example, road traffic direction and lane shapes. Therefore, to achieve a generalized lane detection performance in diverse environments, further studies on domain adaptation [140] or training and testing the DL-LDA for diverse datasets are necessary. For example, FoloLane [112] and CLRNet [94] are trained with CULane dataset and tested with TuSimple dataset.

### B. Robustness to Various Driving Environments

Current DL-LDAs have demonstrated good performance in normal driving conditions, but they still struggle in challenging conditions, such as adverse weather, shadows, and the absence of lines. Especially, when it rains or snows, lane markings in camera images or LiDAR point cloud are often unclear and occluded. While we may not expect the same level of performance in both normal and adverse conditions, the accuracy gap between the normal and adverse weather conditions is still unacceptable. Unfortunately, few studies tackle this issue intensively. Therefore, future DL-LDA studies need to devise advanced feature extraction methods for adverse weather conditions. As a potential direction for future studies, 4D radar sensors have recently gained attention in various perception studies such as [147] and [148] for their ability to obtain dense 3D information robustly in adverse weather conditions. Therefore, in future lane detection studies, we expect that radar sensors are considered for sensor fusion to enhance the accuracy and robustness of lane detection.

### C. Insufficient Real-Time Performance

Ensuring DL-LDAs to operate in real-time is crucial for high speed vehicles. However, most DL-LDAs are computationally expensive and real-time processing is challenging. Although, recent DL-LDAs have shown improvement in real-time processing using row-wise and anchor-based head methods, the

DL-LDAs still require a large computational capacity in the vehicle. Therefore, for the commercialization of autonomous driving, a sufficiently high run-time speed and low cost are critical issues to be resolved.

#### D. Lane Detection Other Than Front-View Directions

Most DL-LDAs perform lane detection only on the front-view image. However, studies for detecting lanes in the left, right, and rear directions or integrating lane detection results from surrounding directions have not been studied yet. In general, surrounding lane detection can improve the detection performance of the front-view lanes by utilizing lane features from different views, when occlusion or broken lanes are observed on the forward-facing image. Furthermore, lane detection in surrounding views can improve local path planning and construction of high-definition (HD) map, as demonstrated in [149].

## VII. CONCLUSION

In this paper, we have presented a survey of diverse and comprehensive DL-LDA studies introduced in the literature to present. Specifically, we have focused on the overall process of DL-LDAs and provided detailed analysis of each process. In addition, We have presented surveys on studies other than the 2D lane detection using camera images for the first time. This paper has provided the most updated benchmarks, too, which enables readers to analyze the strengths and weaknesses of DL-LDAs. On the other hand, we have presented current technical issues to be resolved for the commercialization of DL-LDAs for autonomous driving vehicles. We expect that this paper provides necessary insights into the architecture and essential algorithms of DL-LDAs that are required for the development of advanced and real-time DL-LDAs to ensure robust, safe, and highly reliable autonomous driving.

## REFERENCES

- [1] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-Road Motor vehicles*, Standard J3016\_201806, SAE International, 2018.
- [2] K.-B. Kim and D. H. Song, "Real time road lane detection with RANSAC and HSV color transformation," *J. Inf. Commun. Converg. Eng.*, vol. 15, no. 3, pp. 187–192, Jan. 2017.
- [3] J. He, H. Rong, J. Gong, and W. Huang, "A lane detection method for lane departure warning system," in *Proc. Int. Conf. Optoelectron. Image Process.*, vol. 1, Nov. 2010, pp. 28–31.
- [4] D. Obradović, Z. Konjović, E. Pap, and I. J. Rudas, "Linear fuzzy space based road lane model and detection," *Knowl.-Based Syst.*, vol. 38, pp. 37–47, Jan. 2013.
- [5] B.-S. Shin, J. Tao, and R. Klette, "A superparticle filter for lane detection," *Pattern Recognit.*, vol. 48, no. 11, pp. 3333–3345, Nov. 2015.
- [6] H. Guan, W. Xingang, W. Wenqi, Z. Han, and W. Yuanyuan, "Real-time lane-vehicle detection and tracking system," in *Proc. Chin. Control Decis. Conf. (CCDC)*, May 2016, pp. 4438–4443.
- [7] R. K. Satzoda, S. Sathyaranayana, T. Srikanthan, and S. Sathyaranayana, "Hierarchical additive Hough transform for lane detection," *IEEE Embedded Syst. Lett.*, vol. 2, no. 2, pp. 23–26, Jun. 2010.
- [8] K. Ghazali, R. Xiao, and J. Ma, "Road lane detection using H-Maxima and improved Hough transform," in *Proc. 4th Int. Conf. Comput. Intell., Modelling Simulation*, Sep. 2012, pp. 205–208.
- [9] G. Liu, F. Wörgötter, and I. Markelic, "Combining statistical Hough transform and particle filter for robust lane detection and tracking," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 993–997.
- [10] A. Mammeri, A. Boukerche, and G. Lu, "Lane detection and tracking system based on the MSER algorithm, Hough transform and Kalman filter," in *Proc. 17th ACM Int. Conf. Modelling, Anal. Simulation Wireless Mobile Syst.*, Sep. 2014, pp. 259–266.
- [11] Y. Huang, Y. Li, X. Hu, and W. Ci, "Lane detection based on inverse perspective transformation and Kalman filter," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 2, pp. 643–661, 2018.
- [12] K. Zhao, M. Meuter, C. Nunn, D. Müller, S. Müller-Schneiders, and J. Pauli, "A novel multi-lane detection and tracking system," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 1084–1089.
- [13] D. Vajak, M. Vranješ, R. Grbić, and D. Vranješ, "Recent advances in vision-based lane detection solutions for automotive applications," in *Proc. IEEE Int. Symp. ELMAR*, Sep. 2019, pp. 45–50.
- [14] Y. Xing et al., "Advances in vision-based lane detection: Algorithms, integration, assessment, and perspectives on ACP-based parallel vision," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 3, pp. 645–661, May 2018.
- [15] N. Xu, "Lane detection for driverless vehicle: A survey," *Int. Core J. Eng.*, vol. 6, no. 9, pp. 225–235, 2020.
- [16] Y. Zhang, Z. Lu, X. Zhang, J.-H. Xue, and Q. Liao, "Deep learning in lane marking detection: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 5976–5992, Jul. 2022.
- [17] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [19] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 1–8.
- [20] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, "K-lane: LiDAR lane dataset and benchmark for urban roads and highways," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4449–4458.
- [21] S. Yenikaya, G. Yenikaya, and E. Düven, "Keeping the vehicle on the road: A survey on on-road lane detection systems," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 1–43, Oct. 2013.
- [22] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 727–745, 2014.
- [23] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognit.*, vol. 73, pp. 216–234, Jan. 2018.
- [24] D. Liang, Y.-C. Guo, S.-K. Zhang, T.-J. Mu, and X. Huang, "Lane detection: A survey with new results," *J. Comput. Sci. Technol.*, vol. 35, no. 3, pp. 493–505, May 2020.
- [25] J. Tang, S. Li, and P. Liu, "A review of lane detection methods based on deep learning," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107623.
- [26] N. J. Zakaria, M. I. Shapai, R. A. Ghani, M. N. M. Yassin, M. Z. Ibrahim, and N. Wahid, "Lane detection in autonomous vehicles: A systematic review," *IEEE Access*, vol. 11, pp. 3729–3765, 2023.
- [27] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 7–12.
- [28] J. Fritsch, T. Kuhnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2013, pp. 1693–1700.
- [29] R. F. Berriel, E. de Aguiar, A. F. de Souza, and T. Oliveira-Santos, "Ego-lane analysis system (ELAS): Dataset and algorithms," *Image Vis. Comput.*, vol. 68, pp. 64–75, Dec. 2017.
- [30] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.
- [31] F. Yan et al., "ONCE-3DLanes: Building monocular 3D lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 17143–17152.
- [32] S. Lee et al., "VPGNet: Vanishing point guided network for lane and road marking detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1947–1955.
- [33] K. Behrendt and R. Sousson, "Unsupervised labeled lane markers using maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 832–839.

- [34] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4372–4378.
- [35] TuSimple. (2017). *Tusimple Benchmark*. [Online]. Available: <https://github.com/TuSimple/tusimple-benchmark>
- [36] F. Yu et al., "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2636–2645.
- [37] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "CurveLane-NAS: Unifying lane-sensitive architecture search and adaptive point blending," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 689–704.
- [38] Y. Zhang et al., "VIL-100: A new dataset and a baseline model for video instance lane detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15661–15670.
- [39] Y. Guo et al., "Gen-LaneNet: A generalized and scalable approach for 3D lane detection," in *Proc. 16th Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 666–681.
- [40] L. Chen et al., "PersFormer: 3D lane detection via perspective transformer and the OpenLane benchmark," in *Proc. Eur. Conf. Comput. Vis.*, Jul. 2022, pp. 550–567.
- [41] M. Bai, G. Matiyus, N. Homayounfar, S. Wang, S. K. Lakshminarayanan, and R. Urtasun, "Deep multi-sensor lane detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3102–3109.
- [42] P. Martinek, G. Pucea, Q. Rao, and U. Sivalingam, "LiDAR-based deep neural network for reference lane generation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 89–94.
- [43] J. Mao et al., "One million scenes for autonomous driving: ONCE dataset," 2021, *arXiv:2106.11037*.
- [44] T. A. Clarke and J. G. Fryer, "The development of camera calibration methods and models," *Photogrammetric Rec.*, vol. 16, no. 91, pp. 51–66, Apr. 1998.
- [45] Y. Zhang, Z. Lu, D. Ma, J.-H. Xue, and Q. Liao, "Ripple-GAN: Lane line detection with ripple lane line detection network and Wasserstein GAN," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1532–1542, Mar. 2021.
- [46] Q. Wang, T. Han, Z. Qin, J. Gao, and X. Li, "Multitask attention network for lane detection and fitting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1066–1078, Mar. 2022.
- [47] T. Liu, Z. Chen, Y. Yang, Z. Wu, and H. Li, "Lane detection in low-light conditions using an efficient data enhancement: Light conditions style transfer," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1394–1399.
- [48] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [49] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.
- [50] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
- [51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [53] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [55] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
- [56] T. Falk et al., "U-net: Deep learning for cell counting, detection, and morphometry," *Nature Methods*, vol. 16, no. 1, pp. 67–70, Jan. 2019.
- [57] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
- [58] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.
- [59] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*.
- [60] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2017.
- [61] J. Jin, A. Dundar, and E. Culurciello, "Flattened convolutional neural networks for feedforward acceleration," 2014, *arXiv:1412.5474*.
- [62] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, Sep. 2020.
- [63] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, Apr. 2019.
- [64] A. Garcia-Garcia, S. Orts-Escalano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Appl. Soft Comput.*, vol. 70, pp. 41–65, Sep. 2018.
- [65] T. Zheng et al., "RESA: Recurrent feature-shift aggregator for lane detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3547–3554.
- [66] R. Fan, X. Wang, Q. Hou, H. Liu, and T.-J. Mu, "SpinNet: Spinning convolutional network for lane boundary detection," *Comput. Vis. Media*, vol. 5, no. 4, pp. 417–428, Dec. 2019.
- [67] B. Huval et al., "An empirical evaluation of deep learning on highway driving," 2015, *arXiv:1504.01716*.
- [68] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*.
- [69] J. Zhang, Y. Xu, B. Ni, and Z. Duan, "Geometric constrained joint lane segmentation and lane boundary detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 486–502.
- [70] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [71] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.
- [72] 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 Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Jan. 2020.
- [73] X. Shi, Z. Chen, H. Wang, D. Yeung, W. K. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2015, pp. 802–810.
- [74] Y. Dong, S. Patil, B. van Arem, and H. Farah, "A hybrid spatial–temporal deep learning architecture for lane detection," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 38, no. 1, pp. 67–86, Jan. 2023.
- [75] J. Zhang, T. Deng, F. Yan, and W. Liu, "Lane detection model based on spatio-temporal network with double convolutional gated recurrent units," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6666–6678, Jul. 2022.
- [76] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," 2015, *arXiv:1511.06432*.
- [77] Q. Lai, W. Wang, H. Sun, and J. Shen, "Video saliency prediction using spatiotemporal residual attentive networks," *IEEE Trans. Image Process.*, vol. 29, pp. 1113–1126, 2020.
- [78] S. Valipour, M. Siam, M. Jagersand, and N. Ray, "Recurrent fully convolutional networks for video segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 29–36.
- [79] J. Kim and C. Park, "End-to-end ego lane estimation based on sequential transfer learning for self-driving cars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 30–38.
- [80] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.
- [81] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
- [82] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 102–118.
- [83] Y. Hou, Z. Ma, C. Liu, T.-W. Hui, and C. C. Loy, "Inter-region affinity distillation for road marking segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12483–12492.

- [84] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, “Key points estimation and point instance segmentation approach for lane detection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8949–8958, Jul. 2022.
- [85] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, *arXiv:1503.02531*.
- [86] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.
- [87] A. Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale,” 2020, *arXiv:2010.11929*.
- [88] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.
- [89] W. Li, F. Qu, J. Liu, F. Sun, and Y. Wang, “A lane detection network based on IBN and attention,” *Multimedia Tools Appl.*, vol. 79, nos. 23–24, pp. 16473–16486, Jun. 2020.
- [90] D. Xiao, X. Yang, J. Li, and M. Islam, “Attention deep neural network for lane marking detection,” *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105584.
- [91] J. Wang et al., “A keypoint-based global association network for lane detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1382–1391.
- [92] M. Lee, J. Lee, D. Lee, W. Kim, S. Hwang, and S. Lee, “Robust lane detection via expanded self attention,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1949–1958.
- [93] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, “Keep your eyes on the lane: Real-time attention-guided lane detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 294–302.
- [94] T. Zheng et al., “CLRNet: Cross layer refinement network for lane detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 898–907.
- [95] L. Liu, X. Chen, S. Zhu, and P. Tan, “CondLaneNet: A top-to-down lane detection framework based on conditional convolution,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3773–3782.
- [96] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, “Learning lightweight lane detection CNNs by self attention distillation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1013–1021.
- [97] J. Kim and M. Lee, “Robust lane detection based on convolutional neural network and random sample consensus,” in *Proc. Int. Conf. Neural Inf. Process.*, 2014, pp. 454–461.
- [98] Y. Ma, V. Havaryimana, J. Bai, and Z. Xiao, “Vision-based lane detection and lane-marking model inference: A three-step deep learning approach,” in *Proc. 9th Int. Symp. Parallel Architectures, Algorithms Program. (PAAP)*, Dec. 2018, pp. 183–190.
- [99] B. He, R. Ai, Y. Yan, and X. Lang, “Accurate and robust lane detection based on dual-view convolutional neural network,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 1041–1046.
- [100] W. Song, Y. Yang, M. Fu, Y. Li, and M. Wang, “Lane detection and classification for forward collision warning system based on stereo vision,” *IEEE Sensors J.*, vol. 18, no. 12, pp. 5151–5163, Jun. 2018.
- [101] A. Gurgelian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, “DeepLanes: End-to-end lane position estimation using deep neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 38–45.
- [102] Y. Tian et al., “Lane marking detection via deep convolutional neural network,” *Neurocomputing*, vol. 280, pp. 46–55, Mar. 2018.
- [103] Y. Huang, S. Chen, Y. Chen, Z. Jian, and N. Zheng, “Spatial-temporal based lane detection using deep learning,” in *Proc. 14th Artif. Intell. Appl. Innov.*, Jan. 2018, pp. 143–154.
- [104] M. Fu, X. Wang, H. Ma, Y. Yang, and M. Wang, “Multi-lanes detection based on panoramic camera,” in *Proc. 11th IEEE Int. Conf. Control Autom. (ICCA)*, Jun. 2014, pp. 655–660.
- [105] J. Wang, Y. Wu, Z. Liang, and Y. Xi, “Lane detection based on random Hough transform on region of interesting,” in *Proc. IEEE Int. Conf. Inf. Autom.*, Jun. 2010, pp. 1735–1740.
- [106] J. Zang, W. Zhou, G. Zhang, and Z. Duan, “Traffic lane detection using fully convolutional neural network,” in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Nov. 2018, pp. 305–311.
- [107] M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, and M. Hofmann, “ELGAN: Embedding loss driven generative adversarial networks for lane detection,” in *Proc. 15th Eur. Conf. Comput. Vis. Workshops*, Jan. 2019, pp. 256–272.
- [108] P.-R. Chen, S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, “Efficient road lane marking detection with deep learning,” in *Proc. IEEE 23rd Int. Conf. Digit. Signal Process. (DSP)*, Nov. 2018, pp. 1–5.
- [109] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, and L. Van Gool, “End-to-end lane detection through differentiable least-squares fitting,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 905–913.
- [110] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, “Towards end-to-end lane detection: An instance segmentation approach,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 286–291.
- [111] H. Abualsaud, S. Liu, D. B. Lu, K. Situ, A. Rangesh, and M. M. Trivedi, “LaneAF: Robust multi-lane detection with affinity fields,” *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7477–7484, Oct. 2021.
- [112] Z. Qu, H. Jin, Y. Zhou, Z. Yang, and W. Zhang, “Focus on local: Detecting lane marker from bottom up via key point,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14122–14130.
- [113] Z. Feng, S. Guo, X. Tan, K. Xu, M. Wang, and L. Ma, “Rethinking efficient lane detection via curve modeling,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17062–17070.
- [114] S. Xu et al., “RCLane: Relay chain prediction for lane detection,” in *Proc. 17th Eur. Conf. Comput. Vis.*, Jan. 2022, pp. 461–477.
- [115] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Chamc, Switzerland. Springer, 2016, pp. 483–499.
- [116] X. Li, J. Li, X. Hu, and J. Yang, “Line-CNN: End-to-end traffic line detection with line proposal unit,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 248–258, Jan. 2020.
- [117] Z. Chen, Q. Liu, and C. Lian, “PointLaneNet: Efficient end-to-end CNNs for accurate real-time lane detection,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 2563–2568.
- [118] D. Jin, W. Park, S.-G. Jeong, H. Kwon, and C.-S. Kim, “Eigenlanes: Data-driven lane descriptors for structurally diverse lanes,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17163–17171.
- [119] R. Girshick, “Fast R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
- [120] S. Chougule, N. Koznek, A. Ismail, G. Adam, V. Narayan, and M. Schulze, “Reliable multilane detection and classification by utilizing CNN as a regression network,” in *Proc. 15th Eur. Conf. Comput. Vis. Workshops*, Jan. 2019, pp. 740–752.
- [121] S. Yoo et al., “End-to-end lane marker detection via row-wise classification,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1006–1007.
- [122] Z. Qin, W. Huanyu, and X. Li, “Ultra fast structure-aware deep lane detection,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 276–291.
- [123] Z. Qin, P. Zhang, and X. Li, “Ultra fast deep lane detection with hybrid anchor driven ordinal classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 2555–2568, May 2022.
- [124] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, “PolyLaneNet: Lane estimation via deep polynomial regression,” in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 6150–6156.
- [125] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, “End-to-end lane shape prediction with transformers,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3694–3702.
- [126] B. Wang, Z. Wang, and Y. Zhang, “Polynomial regression network for variable-number lane detection,” in *Proc. 16th Eur. Conf. Comput. Vis.*, Jan. 2020, pp. 719–734.
- [127] H. Pan, X. Chang, and W. Sun, “Multitask knowledge distillation guides end-to-end lane detection,” *IEEE Trans. Ind. Informat.*, vol. 19, no. 9, pp. 9703–9712, Sep. 2023.
- [128] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi, “3D-LaneNet: End-to-end 3D multiple lane detection,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 2921–2930.
- [129] N. Efrat, M. Bluvstein, S. Oron, D. Levi, N. Garnett, and B. El Shlomo, “3D-LaneNet+: Anchor free lane detection using a semi-local representation,” 2020, *arXiv:2011.01535*.
- [130] Y. Jin, X. Ren, F. Chen, and W. Zhang, “Robust monocular 3D lane detection with dual attention,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3348–3352.

- [131] C. Li, J. Shi, Y. Wang, and G. Cheng, "Reconstruct from top view: A 3D lane detection approach based on geometry structure prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4369–4378.
- [132] C. Yang et al., "BEVFormer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 17830–17839.
- [133] Y. Luo et al., "LATR: 3D lane detection from monocular images with transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 7907–7918.
- [134] Y. Liu et al., "PETRv2: A unified framework for 3D perception from multi-camera images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 3239–3249.
- [135] P. Lindner, E. Richter, G. Wanielik, K. Takagi, and A. Isogai, "Multi-channel LiDAR processing for lane detection and estimation," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2009, pp. 1–6.
- [136] D. Cáceres Hernández, V.-D. Hoang, and K.-H. Jo, "Lane surface identification based on reflectance using laser range finder," in *Proc. IEEE/SICE Int. Symp. Syst. Integr.*, Dec. 2014, pp. 621–625.
- [137] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, Aug. 1996, vol. 96, no. 34, pp. 226–231.
- [138] R. Liu et al., "Learning to detect 3D lanes by shape matching and embedding," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4280–4288.
- [139] D.-H. Paek, K. T. Wijaya, and S.-H. Kong, "Row-wise LiDAR lane detection network with lane correlation refinement," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 4328–4334.
- [140] C. Li, B. Zhang, J. Shi, and G. Cheng, "Multi-level domain adaptation for lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 4379–4388.
- [141] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast LiDAR-based road detection using fully convolutional neural networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1019–1024.
- [142] R. Yin, Y. Cheng, H. Wu, Y. Song, B. Yu, and R. Niu, "FusionLane: Multi-sensor fusion for lane marking semantic segmentation using deep neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1543–1553, Feb. 2022.
- [143] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "LiDAR-camera fusion for road detection using fully convolutional neural networks," *Robot. Auto. Syst.*, vol. 111, pp. 125–131, Jan. 2019.
- [144] Z. Liu et al., "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 2774–2781.
- [145] X. Zhang, Z. Li, X. Gao, D. Jin, and J. Li, "Channel attention in LiDAR-camera fusion for lane line segmentation," *Pattern Recognit.*, vol. 118, Oct. 2021, Art. no. 108020.
- [146] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.
- [147] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, "K-radar: 4D radar object detection for autonomous driving in various weather conditions," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 3819–3829.
- [148] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, "Enhanced K-radar: Optimal density reduction to improve detection performance and accessibility of 4D radar tensor-based object detection," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2023, pp. 1–6.
- [149] L. Qiao, W. Ding, X. Qiu, and C. Zhang, "End-to-end vectorized HD-map construction with piecewise Bézier curve," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 13218–13228.



**Min-Hyeok Sun** received the B.S. degree in automotive engineering from Kookmin University, Seoul, South Korea, in 2022, and the M.S. degree from the CCS Graduate School of Mobility, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2024. He is currently with Hyundai Motor Company, South Korea. His research interests include autonomous vehicle, perception, and robotics.



**Seung-Hyun Kong** (Senior Member, IEEE) received the B.S. degree in electronics engineering from Sogang University, Seoul, South Korea, in 1992, the M.S. degree in electrical and computer engineering from Polytechnic University (merged to NYU), New York, in 1994, and the Ph.D. degree in aeronautics and astronautics from Stanford University, Palo Alto, in 2005. From 1997 to 2004 and from 2006 to 2010, he was with companies, including Samsung Electronics (Telecommunication Research Center), South Korea, and Qualcomm (Corporate Research and Development Department), San Diego, USA, for advanced technology research and development in mobile communication systems, wireless positioning, and assisted GNSS. Since 2010, he has been a Faculty Member with the CCS Graduate School of Mobility, Korea Advanced Institute of Science and Technology (KAIST). Since he joined KAIST as a Faculty Member in 2010, he has been working on various research and development projects in advanced intelligent transportation systems, such as robust GNSS-based navigation for urban environment, deep learning and reinforcement learning algorithms for autonomous vehicles, sensor fusion, and vehicular communication systems (V2X). He is currently an Associate Professor with the CCS Graduate School of Mobility, KAIST. He has authored more than 100 papers in peer-reviewed journals and conference proceedings and holds 12 patents. His research group won the President Award (of South Korea) in the 2018 International Student Autonomous Driving Competition host by the Korean Government. He has served as an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the Program Chair for IPNT, South Korea, from 2017 to 2019, the Program Co-Chair for IEEE ITSC2019, New Zealand, and the General Chair for IEEE IV2024 in Jeju, South Korea.



**Dong-Hee Paek** (Graduate Student Member, IEEE) received the B.S. degree in robotics from Kwangwoon University, South Korea, in 2019, and the M.S. degree from the Robotics Program, Korea Advanced Institute of Science and Technology (KAIST), where he is currently pursuing the Ph.D. degree with the CCS Graduate School of Mobility. His research interests include autonomous vehicle, reinforcement learning, and robotics.