remote sensing

MDPI

Article

InstLane Dataset and Geometry-Aware Network for Instance Segmentation of Lane Line Detection

Qimin Cheng [1,*], Jiajun Ling [1], Yunfei Yang [2], Kaiji Liu [1], Huanying Li [1] and Xiao Huang [3]

[1] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China; jiajunling@hust.edu.cn (J.L.); d202381129@hust.edu.cn (K.L.); m202172459@hust.edu.cn (H.L.)
[2] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China; yang-yf21@mails.tsinghua.edu.cn
[3] Department of Environmental Sciences, Emory University, Atlanta, GA 30322, USA; xiao.huang2@emory.edu
* Correspondence: chengqm@hust.edu.cn

Abstract: Despite impressive progress, obtaining appropriate data for instance-level lane segmentation remains a significant challenge. This limitation hinders the refinement of granular lane-related applications such as lane line crossing surveillance, pavement maintenance, and management. To address this gap, we introduce a benchmark for lane instance segmentation called InstLane. To the best of our knowledge, InstLane constitutes the first publicly accessible instance-level segmentation standard for lane line detection. The complexity of InstLane emanates from the fact that the original data are procured using cameras mounted laterally, as opposed to traditional front-mounted sensors. InstLane encapsulates a range of challenging scenarios, enhancing the generalization and robustness of the lane line instance segmentation algorithms. In addition, we propose GeoLaneNet, a real-time, geometry-aware lane instance segmentation network. Within GeoLaneNet, we design a finer localization of lane proto-instances based on geometric features to counteract the prevalent omission or multiple detections in dense lane scenarios resulting from non-maximum suppression (NMS). Furthermore, we present a scheme that employs a larger receptive field to achieve profound perceptual lane structural learning, thereby improving detection accuracy. We introduce an architecture based on partial feature transformation to expedite the detection process. Comprehensive experiments on InstLane demonstrate that GeoLaneNet can achieve up to twice the speed of current State-Of-The-Artmethods, reaching 139 FPS on an RTX3090 and a mask AP of 73.55%, with a permissible trade-off in AP, while maintaining comparable accuracy. These results underscore the effectiveness, robustness, and efficiency of GeoLaneNet in autonomous driving.

Keywords: lane line detection; autonomous driving; instance segmentation; ground remote sensing

check for updates

Citation: Cheng, Q.; Ling, J.; Yang, Y.; Liu, K.; Li, H.; Huang, X. InstLane Dataset and Geometry-Aware Network for Instance Segmentation of Lane Line Detection. *Remote Sens.* **2024**, *16*, 2751. https://doi.org/10.3390/rs16152751

Academic Editors: Mohammad Aldibaja, Sufyan Ali Memon and Andrea Masiero

Received: 9 May 2024
Revised: 22 July 2024
Accepted: 27 July 2024
Published: 28 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

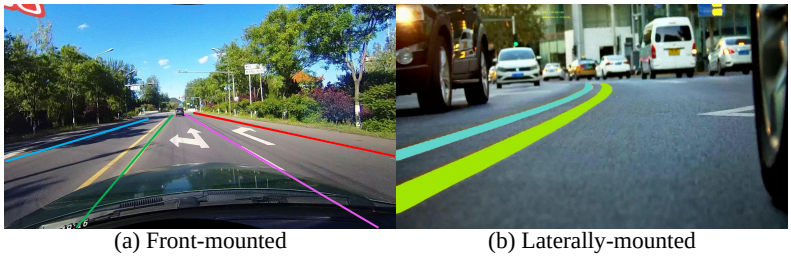
Autonomous Driving (AD) has revolutionized vehicle safety and efficiency through sophisticated technologies such as HD map-guided motion planning [1], advanced image-to-point cloud registration [2], and LIDAR maps [3]. These innovations enhance a vehicle's ability to navigate complex environments and support real-time applications.

Central to the effectiveness of ADAS is the capability of precise lane line detection [4–16]. This technology extends beyond typical driving-related tasks such as lane departure warnings and solid-line crossing surveillance. It can also be leveraged for infrastructure purposes such as pavement maintenance and management, among others. However, the task of lane line detection still poses significant challenges due to various factors: (1) data acquisition conditions vary greatly, encompassing scenarios with low luminance, inclement weather, and diverse camera angles; (2) road scenarios can be complex, incorporating a variety of lane line types, particularly curved and multi-lanes, and are often compounded by interfer-

ence from irrelevant road signs, markings, and occlusions; (3) the necessity for real-time detection imposes requirements for lightweight, high-efficiency network architectures.

Significant strides have been made in this field to address the aforementioned challenges associated with lane line detection. The predominant focus of recent work has been on developing deep learning-based approaches, as deep neural networks have proven their mettle in powerful feature representation capabilities for most visual-based tasks. Deep learning-based lane detection methodologies can broadly be categorized into three types: curve structure-based, keypoint detection-based, and segmentation-based algorithms. Curve structure-based algorithms, including PolyLaneNet [5] and BézierLaneNet [6], interpret lane detection as a regression problem for mathematical curve description. Keypoint detection-based algorithms, such as Line-CNN [7] and UFLD [8], are designed to yield a more precise representation by identifying specific key points or anchor locations, thereby addressing optimization problems inherent in curve-based algorithms. Meanwhile, segmentation-based algorithms, such as LaneNet [9] and SCNN [10], treat lane detection as an instance or semantic segmentation issue. These approaches have the distinct advantage of expressing the detected lane as an explicit mask and avoiding complex post-processing steps, such as curve fitting. However, resulting from the requirement to evaluate the association of each individual pixel with a lane, segmentation-based algorithms have a critical limitation in their relatively lower efficiency.

Benchmark datasets are widely recognized as critical catalysts for advancing empirical progress in deep learning. Notable benchmark datasets for lane detection include CULane [10], TuSimple [17], Apollo Scape [18] and BDD100K [19], among others. TuSimple is known for its clear highway scenarios, suitable for sunny and cloudy conditions. CULane, on the other hand, provides coverage of eight distinct challenging scenarios. While both Tusimple and CULane provide labeled lane data, Apollo Scape and BDD100K offer a more comprehensive range of traffic annotations. Notably, most publicly available benchmarks draw data from commercial front-mounted sensors. This data acquisition method is convenient and economical. At the same time, the acquired images are suitable for some popular applications of Intelligent Transportation Systems (ITS), like centerline detection and urban scene segmentation. However, these images might be inappropriate for granular lane-related applications such as lane line crossing surveillance, blind spot detection, and pavement maintenance from certain perspectives. For this reason, some manufacturers and researchers are focusing on alternative data acquisition methods. For instance, the Tesla Model 3 has eight cameras, while the Heyxpeng G9 boasts eleven impressive cameras around the vehicle. Figure 1 compares the images captured from front-mounted and laterally-mounted cameras, respectively. Most of the existing datasets use a series of points [10,17,20–22] or semantic mask labels [11,18,19] to describe the location of lane lines for front-mounted images in Figure 1a. However, it is evident that respective instance-level labels for laterally-mounted scenes in Figure 1b are more appropriate for the lane lines with a certain area. However, these kinds of benchmarks remain quite scarce, which hampers the development of the fine-grained applications mentioned above.



(a) Front-mounted (b) Laterally-mounted

Figure 1. Comparison of lane lines captured by front-mounted and laterally-mounted cameras.

In an effort to surmount the limitations inherent in existing lane benchmarks, for instance, segmentation, we introduce a novel benchmark, i.e., InstLane. The innovation of InstLane is that, firstly, unlike the classical datasets, which are collected from front cameras,

InstLane is a high-resolution dataset of 4096 × 2160 pixels collected from lateral sensors, which enriches the relevant data for scene understanding in autonomous driving. The unique advantages of lateral perspective data include improved blind spot detection and enhanced lane line monitoring, which are critical for addressing specific driving challenges. Lateral data captures areas that front-facing sensors often miss, making it invaluable for detecting objects in blind spots and monitoring lane lines in more detail. The integration of lateral perspective data into remote sensing methods complements traditional techniques by providing a more detailed view of lane markings and road infrastructure. This enhanced view is beneficial for accurate road maintenance and infrastructure assessment, ultimately contributing to the development of more reliable autonomous driving systems. Secondly, it is a dataset annotated with instance-level masks, which is of great significance for developing lane line detection and derivative applications of autonomous driving. Finally, with various challenging scenarios like different light and shadows, it can be used for robustness evaluation. Furthermore, to meet the need for both generalization and real-time application compatibility, we propose an innovative instance-level lane segmentation network, i.e., Geometry Lane Network (GeoLaneNet). GeoLaneNet boasts several key advantages, which can be encapsulated in three main aspects: (1) GeoLaneNet enables deeper perceptual learning by ingeniously fusing features with different scales of the receptive field. This refined fusion ensures that the network adequately captures local and global contexts. Compared with previous methods, GeoLaneNet has different fine-grained representations with large receptive fields and achieves superior results through various receptive field feature acquisitions and fusion. (2) It mitigates the prevalent issue of under- or over-filtering of proto instances in dense lane scenes, typically due to non-maximum suppression (NMS). Compared with previous approaches, it does not require modifying network branches or complex re-training. It distinguishes different instances while maintaining real-time detection, achieving better results than those rigid NMS-improvement approaches. (3) The network enhances segmentation efficiency by introducing partial feature transformation between different frames, allowing faster computation and improved performance.

This study contributes significantly to the field of autonomous driving technology, specifically lane detection, in three primary ways:

1. We introduce a novel instance segmentation benchmark for lane line detection, termed InstLane. Distinctively characterized by its challenging and intricate scenarios, InstLane encompasses a variety of data acquisition conditions and encompasses interference from unrelated road information. The objective of this benchmark is to enhance the generalization and robustness of lane detection algorithms.
2. We propose GeoLaneNet, a highly efficient, instance-level lane line segmentation network. Through the strategic design of a large receptive field to improve the perception of lane structures, the utilization of geometric features to realize finer instance localization, and the introduction of a partial feature transform to boost speed, GeoLaneNet significantly enhances detection efficiency while maintaining detection accuracy.
3. We conduct a thorough set of experiments, including comparative studies, ablation studies, and visualization of results, to assess the effectiveness, robustness, and efficiency of GeoLaneNet. These studies serve to verify the significant performance improvements that GeoLaneNet delivers in the realm of lane detection.

The rest of this paper is organized as follows. Section 2 comprehensively reviews related work, including representative approaches and public datasets. Section 3 introduces the InstLane dataset. Section 4 presents the proposed GeoLaneNet in detail. Section 5 presents the experimental results, analysis, and discussions, followed by conclusions and prospects in Section 6.

2. Related Work

This section provides a succinct survey of traditional lane detection datasets. Additionally, we categorize lane detection methodologies into three distinct categories.

2.1. Lane Dataset

Caltech Lane [21] offers scenes characterized by shadows cast by various elements such as trees, buildings, and moving vehicles. It encompasses 1200 images, each with dimensions of 640 × 480 pixels. Notably, all images in the dataset were captured on urban streets during sunny weather conditions, thus providing a specific context for lane detection analysis.

Road Marking [23] consists of approximately 1400 labeled images featuring a diverse range of road markings. Each image in this dataset is labeled with bounding boxes to denote specific features. The dataset encompasses a variety of road scenes captured under different illumination conditions, offering a diverse set of scenarios for road marking analysis.

TuSimple [17] comprises more than 6400 images with dimensions of 1280 × 720 pixels. These images were captured under good and medium weather conditions, offering a range of environmental scenarios. A defining characteristic of the TuSimple dataset is its emphasis on highway scenes where lane lines are distinctly visible and easily identifiable, thus providing a focused context for lane detection studies.

CULane [10] is a comprehensive collection comprising more than 133,000 images of 1640 × 590 pixels. What sets CULane apart is its inclusion of eight challenging scenarios beyond ordinary driving conditions, namely crowded environments, nighttime settings, scenarios with absent lane lines, areas with shadow, arrow markings, dazzling light, curved routes, and cross scenes. Given its broad array of weather conditions, time periods, and lighting environments, such as sunny and rainy days or nighttime, and its diversity of scenes, including urban roads, suburban roads, and highways, the CULane dataset has been primarily utilized for evaluating the robustness of various lane detection methodologies.

CurveLanes [22] includes more than 150,000 high-resolution images of 2560 × 1440 pixels. A distinguishing feature of this dataset is that 90% of its images contain curved lane lines. Additionally, CurveLanes includes more complex scenarios, such as "Y-shaped" crossings, offering a unique resource for the analysis of lane detection in more intricate road environments.

DeepLanes [12], different from the datasets above, is a large lane dataset collected by laterally-mounted cameras, which cannot only be used to estimate the position of vehicles in the lane but also the robustness of the driver assistant system. It includes over 80,000 manually labeled images and over 40,000 semi-artificial labeled images.

DET [11] comprises more than 5400 images of 1280 × 800 pixels each, gathered from event cameras installed at various positions on the vehicle. It is designed to address the challenge of lane detection in environments with low light or rapid changes in ambient lighting conditions. As such, DET provides a unique resource for developing and testing lane detection algorithms under these demanding scenarios.

Most public lane datasets are collected from the front-mounted cameras. Unlike them, DeepLanes uses a laterally mounted camera for photography, and DET takes an event camera for photography, further broadening lane detection application scenarios. In addition, it is worth mentioning that most datasets are labeled with points or semantic labels. However, instance-level datasets for lane segmentation are still lacking. Therefore, we presented the InstLane dataset, and detailed information is shown in Table 1 and Section 3.

Table 1. Summary of the attributes of representative lane line datasets.

Dataset	Num.	Resolution	Areas	Challenging Scenes	Camera	Label
Caltech Lanes [21]	1.2 k	640 × 480	Urban Road	——	Front-RGB	Points
Road Marking [23]	1.4 k	800 × 600	Superhighway	Cloudy, Twilight, night	Front-RGB	BBoxes

Table 1. *Cont.*

Dataset	Num.	Resolution	Areas	Challenging Scenes	Camera	Label
TuSimple [17]	6.4 k	1280 × 720	Superhighway	——	Front-RGB	Points
Llamas [20]	100 k	1276 × 717	Superhighway	——	Front-RGB	Points
BDD100K [19]	100 k	1280 × 720	Urban Road, Rural Road, Superhighway	Rainy, Snowy, Night	Front-RGB	Semantic labels
CULane [10]	133 k	1640 × 590	Urban Road, Rural Road, Superhighway	8 complex scenes	Front-RGB	Points
CurveLanes [22]	150 k	2560 × 1440	Urban Road	Curve Lanes	Front-RGB	Points
ApolloScape [18]	170 k	3384 × 2710	Urban Road	Various weathers, Complex scenes	Front-RGB	Semantic labels
DeepLanes [12]	120 k	360 × 240	Urban Road	——	Laterally-RGB	——
DET [11]	5.4 k	1280 × 800	Urban Road	——	Front-Event	Semantic labels
InstLane (ours)	7.5 k	4096 × 2160	Urban Road	shadow, dazzle light grayscale with noise	Laterally -RGB	Instance labels

2.2. Lane Detection

Lane detection algorithms based on keypoint detection have widespread use, the primary methodology being to regress the key points or classify the anchor locations. Some of the notable works in this field are as follows. Ref. [7] pioneered the perspective of considering lane detection as an object detection problem and designed a keypoint anchor for lane lines, a method referred to as Line-CNN. A limitation of this approach is its reliance on the FasterRCNN [24] as the object detector, which results in slower detection speeds. Ref. [13] utilized the same anchor representation as Line-CNN. To address the low-efficiency issue observed in Line-CNN, they designed a light attention module to fuse global features, which exhibits superior performance in challenging scenarios. Contrary to anchor-based methods, refs. [8,14] divided the image into dense grids, considering lane detection as a classification problem to increase speed. However, as the number of lane lines and line anchors increases, the model size correspondingly enlarges. Distinct from the CNN-based methodologies, ref. [16] employed a Transformer [25] to extract deeper lane features. They proposed row and column self-attention mechanisms to better learn the shape of lane lines. Among these approaches, refs. [8,16] stand out due to their superior portability and swift inference speed.

Curve-based lane detection algorithms output mathematical curve descriptions of lane lines directly. The advantage is that no post-processing is required to estimate the whole lane line. Representative works include the following: Ref. [15] used the least-square method to fit the lane curve and achieve end-to-end training and prediction by back-propagation of the fitting process. Similarly, refs. [5,15] also regarded lane detection as a polynomial regression problem. However, different from LSFLaneNet [15], PolyLaneNet [5] directly predicted the polynomial coefficients by constructing anchors in fully connected layers, which simplifies the method and achieves a faster frame rate than LSFLaneNet. However, PolyLaneNet did not perform well. To achieve better performance with the curve-based method, ref. [6] used the Bézier curve for lane representation and proposed

the feature flip fusion module. In general, curve-based lane detection algorithms fit the lane line into quadratic, cubic polynomial, and Bézier curves. However, a limited number of mathematical parameters is not sufficient to fit complex lane scenes.

Image segmentation-based algorithms treat lane detection as pixel-level classification problems, necessitating categorizing each pixel into its corresponding instance or semantic category. Ref. [9] integrated semantic segmentation and vectorial pixel representation, which were subsequently clustered to derive lane line instances. However, this simple clustering approach proved unsatisfactory for the detection of complex scenes. To address challenging scenes, such as occluded conditions, ref. [10] designed a network employing convolution in four distinct directions, which facilitated the recognition of structures with continuous physical lane line extensions. Despite this advancement, the complexity of the computations resulted in a slow process, failing to meet real-time requirements. To expedite detection, ref. [26] implemented a lightweight module for Self-Attention Distillation (SAD) on ENet [27], dealing with occluded conditions by utilizing varying attention maps in different layers. ENet-SAD accomplished a detection speed ten times faster and higher accuracy compared to SCNN [10]. Ref. [28] considered the problem of mismatched lane position distributions and proposed a unified viewpoint transformation for better alignment. Ref. [29] incorporated a cosine metric to better discriminate foreground features for segmentation. Despite these developments, image segmentation-based algorithms continue to grapple with efficiency challenges due to the vast quantity of pixels involved.

In light of the above, we contemplated the concepts inherent in certain instance segmentation algorithms. These algorithms, developed for natural scenes, have demonstrated impressive real-time performance. We aim to explore their potential application for lane detection. Among these methods, contour-based techniques such as DeepSnake [30], PolarMask [31], Dance [32] and E2EC [33] employ a gradual adjustment strategy. Non-maximum suppression (NMS)-based algorithms currently form a mainstream category in the detection field. SOLOv2 [34], for instance, reframes the segmentation problem into a position classification problem, bypassing the preceding object detection process by conducting dynamic convolution. YolactEdge [35] significantly improves efficiency by utilizing features of keyframes to estimate those of non-keyframes, thus saving computational resources in comparison with YOLACT [36]. Recently, NMS-Free methods such as K-Net [37] and SparseInst [38] have attained higher accuracy by forgoing NMS post-processing. Nevertheless, the efficiency of both K-Net and SparseInst remains less than optimal, underscoring the need for continued development in this area.

3. InstLane Dataset

The original data utilized in the creation of InstLane was obtained through laterally mounted cameras situated across various road types within the Haidian District, Beijing, China. A total of 30 video clips were meticulously collected for this purpose. The dataset encompasses a multitude of challenging scenes, including scenarios with varying illumination conditions such as night-time, dazzling lights, and uneven shadows, as well as complex traffic road situations like curved, dashed lines and intricate zebra crossings. To enhance the noise immunity and interference resilience of the algorithms, InstLane also offers grayscale scenes.

The construction process of InstLane proceeded as follows: For regular scenes involving straight lanes captured during the daytime, an image was extracted every 30 frames. Conversely, for challenging scenes, such as those mentioned earlier, an image was extracted every 10 frames to capture crucial details. Our sampling perspective is from the side of the vehicle, which differs from existing public datasets (such as TuSimple and CULane) that use a driver's perspective. For near-distance side-view captures, the lane lines appear as regions with certain areas, whereas the lane lines seen from a driver's perspective can be simplified into a sequence of elongated points, as shown in Figure 2. Therefore, we cannot represent side-view lane lines with simple points but rather need masks to precisely describe their positions. As a result, we adopted instance segmentation methods

to manually label the masks of different lane line instances. Although this is indeed a time-consuming process, we believe it provides richer and more accurate information, improving the model's performance and generalization.

InstLane consists of a comprehensive collection of 7541 images, each with dimensions of 4096 × 2160 pixels, following the COCO [39] instance-label formats. A summary of InstLane and the number of images categorized by different environments can be found in Table 2. Examples showcasing the aforementioned scenarios are illustrated in Figure 3.

Table 2. Images amount in different environments in InstLane.

Dataset	Env	Num.	Total
train	Day	1073 (14.23%)	6787
	Night	1004 (13.31%)	
	Gray	4710 (62.46%)	
testval	Day	108 (1.43%)	754
	Night	111 (1.47%)	
	Gray	535 (7.09%)	



(a) Instance annotations in InstLane (b) point annotations in CULane

Figure 2. Examples of different annotation formats. Line lines in (**a**) are annotated with distinct instance labels in different colors, while those in CULane [10] (**b**) are annotated solely by points.



(a) complex road scenarios (b) dazzling scenarios (c) noisy scenarios

Figure 3. Examples of challenging scenes in InstLane.

4. Methodology

4.1. Pipeline of GeoLaneNet

The pipeline of GeoLaneNet is illustrated in Figure 4. The whole architecture is composed of three main parts: Deep Perception Learning Module, YolactEdge-based Coarse Detection, and Geometry-Aware Fine Localization. The original image is first down-sampled. The input image is first down-sampled to 550 × 550 to ensure network inference speed. Then it is fed into the Deep Perception Learning Module for high-level feature extraction. Later, the feature maps C3, C4, and C5 are used for coarse detection of proto-instance generation. Finally, the proto-instances are fed into the Geometry-Aware Fine Localization module, which are selected or filtered by Cluster NMS and Score Filter with their unique geometric features to obtain the final detection results.

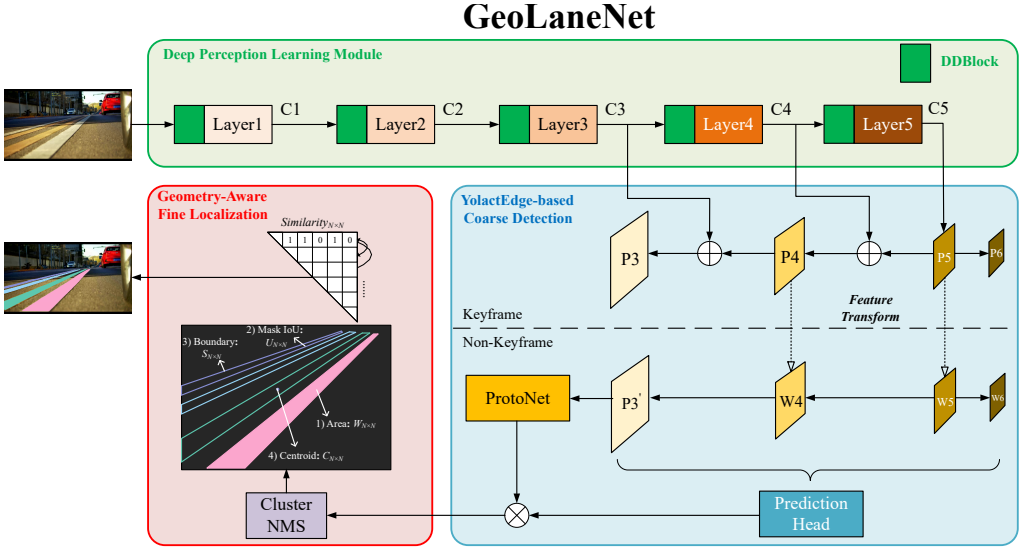


Figure 4. Architecture of GeoLaneNet.

- Deep Perception Learning Module: This module aims to facilitate the identification of semantic characteristics at deeper and higher levels and better perceive the long and large-scale lane lines. It is achieved through the fusion of various reception fields.
- YolactEdge-based Coarse Detection: This module aims to speed up the generation of proto-instances by introducing the partial feature transformation strategy between key frames and non-key frames in YolactEdge.
- Geometry-Aware Fine Localization: This module is designed to achieve a fine-grained localization task, in which various geometric features are subtly utilized to alleviate the problem of omission or multiple detections in dense lane scenarios due to NMS.

4.2. Deep Perception Learning Module

The importance and effectiveness of the large receptive field have been demonstrated in many visual tasks, e.g., classification [40,41], segmentation [40–43], and object detection [41,44]. For segmentation tasks, lane lines are considered large-scale objects. Therefore, enlarging the receptive field of the network allows the network to "see" large-scale lane lines faster and to learn and better adjust the structure representations. Therefore, a plug-and-play module, Depthwise Dilated Block (DDBlock), is designed to achieve deep perceptual lane structural learning via an expanding receptive field.

There are two main ideas for enlarging the receptive field. One is to make the network deeper [40,42,45] by constantly stacking convolution or pooling layers and to reduce the resolution of feature maps. Stacking more convolutions enhances the interaction of information between neighboring regions, which in turn enlarges the receptive field. The other is to start from the design of convolution kernels [41,44], including using large kernel size or dilated settings, directly making the coverage area of the convolution kernel larger

to "see" a larger area. Homoplastically, a plug-and-play module, Depthwise Dilated Block (DDBlock), is designed to achieve deep perceptual lane structural learning via an expanding receptive field.

The proposed DDBlock aims to fuse different fine-grained receptive field features from different layers via different settings of kernel size and dilated rate. However, there are some differences between it and the current methods.

(1) It is different from the rigid way of altering independent dilated rate [40,42,45] or kernel size [41,44]. The former can enlarge the receptive field by crossing regions with limited parameters and operations but suffer from the poor representation of the depth features because of few feature sampled points. The latter achieves a large receptive field and a better representation through larger convolution kernels but also brings more parameters and operations and makes real-time performance worse, which is unacceptable for the autonomous driving system. Therefore, DDBlock considers their characteristics comprehensively and adjusts both parameters simultaneously, obtaining a variety of receptive field features compared to an independent parameter.

(2) Different from D-LinkNet [45], which only learns the receptive field in the deepest layer, we integrate the receptive field features acquired in different ways in each layer. It can obtain comparable receptive field features in all layers to more fully conduct the mining of semantic information.

(3) Moreover, DDBlock adjusts inversely for the two parameters, which has two benefits. One is that different modules can use different settings to control the number of parameters and calculation of the whole network, thus maintaining a large receptive field and real-time performance simultaneously. The other is that the setting is not arbitrary but fully considers the respective characteristics of different levels. We use a combination of large kernels with small dilated rates, small kernels with large dilated rates for shallow layers, and small kernels with small dilated rates for deep layers, which can maintain relatively large and reasonable rerceptive fields in all layers.

The specific settings for convolution kernel size (K) and dilated rate (D) of DDBlock are shown in Table 3. In our experiments, the lightweight model ResNet18[46] is selected as the backbone to meet real-time requirements. Additionally, depthwise convolution is introduced to reduce the parameters of a large kernel. The structure of DDBlock in layer1 of ResNet18 is shown in Figure 5 to illustrate the fusion way, and then, finally, the architecture of the backbone is shown in Figure 6.

Table 3. Configurations of kernel size (K) and dilated rate (D) in different conv blocks of different layers.

Layer	Configurations									
	①		②		③		④		⑤	
	K	***D***	***K***	***D***	***K***	***D***	***K***	*D*	***K***	***D***
1	31	1	23	2	15	4	7	8	3	16
2	23	1	15	2	7	4	3	8		
3	15	1	7	2	3	4				
4	7	1	3	2						

Explanation: Each layer contains a subset of configurations, with Layer1 having all five configurations (① to ⑤), Layer2 having the first four (① to ④), Layer3 having the first three (① to ③), and Layer4 having the first two (① and ②).

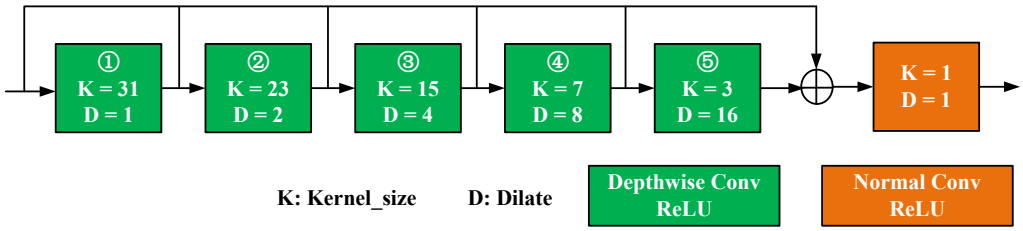


Figure 5. Architecture of DDBlock in layer1 of ResNet18.

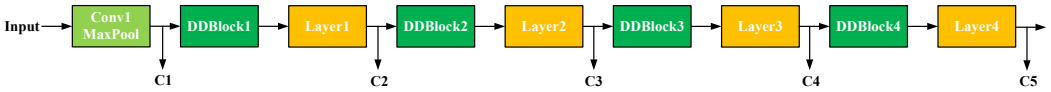


Figure 6. Architecture of the backbone (Examples of DDBlock in ResNet18).

4.3. YolactEdge-based Coarse Detection

As mentioned above, YolactEdge-style architecture [35] is characterized by its surprising inference speed in natural scenes. For this reason, the ideas of partial feature transformation and instance generation are introduced into the proposed GeoLaneNet to reduce computational complexity.

- Partial Feature Transform: This part uses features of keyframes to predict features of non-keyframes. YolactEdge employs a method to exploit temporal redundancy by processing keyframes in detail and using partial feature transformations for non-keyframes. Specifically, all feature maps in the backbone (C1∼C5) and FPN (P3∼P6) for keyframes and several feature maps (C1∼C3 in the backbone and P3, P6 in FPN) for non-keyframes are computed from the previous Deep Perception Learning Module:

$$P_i = C_i + up(P_{i+1}), i = 3, 4, 5$$

where $up()$ is upsampling. Then, W4 and W5 in FPN of non-keyframes are transformed from P4 and P5 features of the previous keyframe by FeatFlowNet proposed also in [35].

$$W_i = FeatFlowNet(P_i), i = 4, 5$$

- Instances Generation: This part is composed of ProtoNet and Prediction Head. ProtoNet is used to generate a certain number of prototype masks. The prediction Head is used for mask coefficient generation. Finally, these prototype masks are linearly combined based on mask coefficients to obtain the proto-instance segmentation results:

$$M = \sigma(PC^T)$$

where $P \in N^{h \times w \times k}$ is k prototype masks, h and w are the size of the box, $C \in N^{n \times k}$ is k mask coefficients, n is the number of objects predicted in the detection branch, and σ is the sigmoid function. This approach significantly reduces computational overhead while maintaining high accuracy in instance segmentation. Thus, the computation of the backbone and FPN in non-keyframes can be significantly reduced, greatly improving the speed of inference.

For more technical details, please refer to [35].

4.4. Geometry-Aware Fine Localization

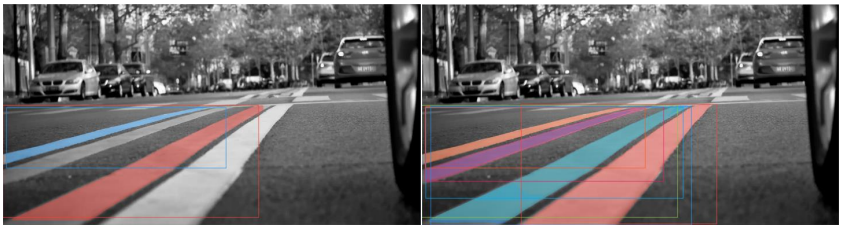
This module manages to mitigate the "NMS Dilemma" in dense scenarios of multiple lane lines, which are pervasive on urban roads and also present a challenge in lane detection.

NMS is an indispensable part of many object detection algorithms. It is used to filter out most detection results with low confidence and high overlap. It involves two basic thresholds: the score threshold and the overlap threshold. The score threshold is used to

filter out most results with low confidence, while the overlap threshold handles cases with high overlap.

"NMS Dilemma" refers to the sensitivity of detecting objects to the NMS overlap threshold setting, which is common in detection-based visual tasks. Take lane line detection as an instance. Setting an overly small NMS overlap threshold may result in filtering out one of the adjacent lane line instances incorrectly. Therefore, we need to maintain a relatively large NMS overlap threshold to ensure that results are not lost. However, a large NMS overlap threshold may lead to multiple bounding boxes with high overlap for a single lane line instance, requiring further processing to select the most appropriate result. Figure 7a,b show these two cases separately. Thus, it is necessary to set a reasonable threshold to suppress repetitive boxes, eventually making each instance correspond to one prediction box.

In ordinary object detection-intensive scenarios, the "NMS Dilemma" has little effect on practical requirements. However, in complex traffic scenarios, omission lane detection may pose serious safety hazards, making the "NMS dilemma" an urgent issue that needs to be addressed.



(a) NMS overlap threshold of 0.5 (b) NMS overlap threshold of 0.85

Figure 7. Demonstration of "NMS Dilemma". Different colored bounding boxes represent different results detected at this threshold.

To mitigate the "NMS Dilemma", a Geometry-Aware Fine Localization module is presented, which is composed of Cluster NMS (with SPM and Dist) [47] and a Score Filter by Geometry Features. The former is used to penalize proto-instances in each cluster by the geometric information of the overlapping area and central point distance. While the latter is used to calculate the instance geometric features, which assists in re-evaluating the relationship between the individual instances by constructing a similarity weight matrix.

Cluster NMS [47] divides the detected boxes through implicit clustering into clusters and implements parallel operations on the GPU to achieve acceleration. In addition, Cluster NMS can easily introduce geometric factors to better identify different instances to improve AP. Geometric factors include Weighted Coordinates, Score Penalty Mechanism (SPM), and Normalized Central Point Distance (Dist), which can be further exploited to improve the average precision.

Score Filter by Geometry Features: A geometric-aware strategy is applied to rescore the proto-instances, and the following four factors are considered:

(1) Area of mask: The smaller the difference between two mask areas, the greater the probability that they belong to the same instance. The area similarity factor weights between the two instances are shown below:

$$w_{ij} = w_{ji} = \begin{cases} \arctan\left(\left|A^i - A^j\right|^{-1}\right), & \left|A^i - A^j\right| \le \delta_A \\ 0, & \left|A^i - A^j\right| > \delta_A \end{cases}$$
$$W_{N\times N} = \frac{2}{\pi}\left(w_{ij}\right)_{N\times N} \quad (1)$$

where δ_A is a pre-set threshold, A^i is the area of the i-th mask, which is very simple to find by calculating the number of pixels where the pixel values are equal to 1. $\frac{2}{\pi}arctan(\cdot)$ are used to constrain the range between 0 and 1.

(2) Mask IoU: Mask IoU is defined as the overlapping degree between two masks. It equals 1 when the two masks totally overlap and 0 when they do not overlap at all.

$$u_{ij} = u_{ji} = \begin{cases} 1, & IoU_{mask}^{ij} > \delta_m \\ 0, & IoU_{mask}^{ij} \le \delta_m \end{cases}, U_{N\times N} = \left(u_{ij}\right)_{N\times N} \tag{2}$$

where $sum(\cdot)$ is the operation of the area calculation of a mask. N is the number of masks. Mask IoU between any two masks is first computed, and the threshold δ_m for classification is set. Then two masks corresponding to the same instance can be considered when $mask\ IoU > \delta_m$:

$$IoU_{mask}^{ij} = \frac{\text{sum}\left(M^i \cap M^j\right)}{\text{sum}\left(M^i \cup M^j\right)} = \frac{\text{sum}\left(M^i \cap M^j\right)}{\text{sum}\left(M^i\right) + \text{sum}\left(M^j\right) - \text{sum}\left(M^i \cap M^j\right)} \tag{3}$$

(3) Coordinate distribution of mask: We measure the coordinate distribution of the mask by calculating the similarity between the two unequal two-dimensional sets of masks. The coordinates of the two masks are first flattened into 1D vectors, and the ratio of the elements of the intersection to the two 1D vectors is calculated separately:

$$m^i = f\left\{\text{argwhere}\left(M^i > 0\right)\right\}, m^j = f\left\{\text{argwhere}\left(M^j > 0\right)\right\} \tag{4}$$

$$s_{ij} = \frac{C\left(m^i \cap m^j, m^i\right)}{2 \times \text{len}\left(m^i\right)} \tag{5}$$

where $f(\cdot)$ is the flatten operation, $C\left(m^i \cap m^j, m^i\right)$ calculates the number of elements of the intersection of m^i and m^i in m^i, s_{ij} calculates the sum of the frequencies of each element of the intersection, and $len(\cdot)$ calculates the length of the vector. The higher the similarity, the closer the spatial distribution of the two mask pixels.
Since this operation on all the pixels of the two masks is relatively large, edge detection on the masks can effectively reduce the computation and achieve a similar performance. Therefore, M^i and M^j are replaced by E^i and E^j in Equation (4), where E^i and E^j represent the results of edge detection of M^i and M^j. In addition, because s_{ij}! $= s_{ji}$, the calculation of the average is needed to regulate the similarity of the two boundaries:

$$e^i = f\left\{\text{argwhere}\left(E^i > 0\right)\right\}, e^j = f\left\{\text{argwhere}\left(E^j > 0\right)\right\} \tag{6}$$

$$g_{ij} = \frac{C\left(e^i \cap e^j, e^i\right)}{2 \times \text{len}\left(e^i\right)} \tag{7}$$

$$\begin{aligned} s_{ij} = s_{ji} &= \frac{g_{ij} \times \text{len}\left(e^i\right) + g_{ji} \times \text{len}\left(e^j\right)}{\text{len}\left(e^i\right) + \text{len}\left(e^j\right)} \\ S_{N\times N} &= \left(s_{ij}\right)_{N\times N} \end{aligned} \tag{8}$$

where e^i, e^j are the set of edge points of i-th and j-th masks.

(4) Centroid of mask: Clustering the centers of these original instances using DBSCAN can further distinguish their true belongings. To avoid the clustering errors resulting from directly calculating the center of the prediction boxes, as illustrated in Figure 8, we use the centroids of the lanes for clustering. The calculation formula of the centroid is shown in Equations (9) and (10):

$$x_0 = \frac{\sum_{(x,y)\in \text{ mask}} xf(x,y)}{\sum_{(x,y)\in \text{ mask}} f(x,y)} \xrightarrow{\text{Binary Img}} x_0 = \frac{\sum_{(x,y)\in \text{ mask}} x}{A_{\text{mask}}} \tag{9}$$

$$y_0 = \frac{\sum_{(x,y)\in \text{ mask}} yf(x,y)}{\sum_{(x,y)\in \text{ mask}} f(x,y)} \xrightarrow{\text{Binary Img}} y_0 = \frac{\sum_{(x,y)\in \text{ mask}} y}{A_{\text{mask}}} \tag{10}$$

where $f(x,y) = \begin{cases} 1, & (x,y) \in \text{ mask} \\ 0, & (x,y) \notin \text{ mask} \end{cases}$ in binary images, and A_{mask} is the area of mask.

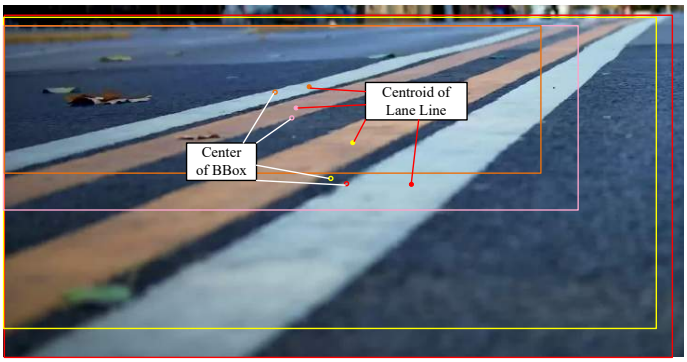


Figure 8. Center of bounding boxes (hollow core), and centroid of the lane-line (solid core).

$C_{N\times N}$ are constructed according to the centroid clustering results, where c_{ij} equals to 1 when centroids of the i-th mask and the j-th mask belong to the same cluster. Otherwise, we set it to 0.

To sum up, the whole process of Score Filter by Geometry Features is summarized as follows:

1. Make preliminary judgments of classification based on the different areas of masks and calculate $W_{N\times N}$.
2. Calculate mask IoU directly at pairwise marks and calculate $U_{N\times N}$.
3. Cluster the masks by the distribution of the coordinate of contours of masks and calculate $S_{N\times N}$.
4. Calculate the centroids of all masks and use a clustering algorithm like DBSCAN to cluster and obtain $C_{N\times N}$.
5. Add them all up to obtain a similarity matrix:

$$Sim = \alpha W_{N\times N} + \beta U_{N\times N} + \gamma S_{N\times N} + \xi C_{N\times N} \tag{11}$$

$$sim_{ij} = \begin{cases} 1, & \text{sim}_{ij} > \delta \text{ and } sim_{ji} > \delta \\ 0, & \text{else} \end{cases} \tag{12}$$

where α, β, γ, ξ are hyper parameters. Then, use Equation (12) to binarize the matrix by the threshold value.

6. Combine the scores of the masks corresponding to 1 and select the one with the highest score as the result of the instance.

4.5. Differences to Other Methods

In general, this module addresses the issue of highly overlapping boxes in dense scenarios. Current solutions can be categorized into three types:

- NMS Improvement Methods: Techniques like Soft-NMS [48] and Softer-NMS [49] modify the suppression strategy by decaying the confidence score of neighboring predictions rather than discarding them outright.
- Network and Loss-based Methods: Approaches such as [50,51] incorporate specific loss functions during training to ensure proposals are closely aligned with ground truth. AdaptiveNMS [52] focuses on learning optimal NMS thresholds, while [53] introduces a branch and uses EMDLoss to handle multiple predictions per proposal.

- NMS-Free Approaches: Methods like K-Net [37] and SparseInst [38] avoid NMS altogether by directly recognizing instances, though they often suffer from slower speeds compared to our method.

The characteristics and differences of the Geometry-Aware Fine Localization Module and these methods can be summarized as follows:

- Enhanced Geometric Distinction: Unlike NMS-improvement methods that rely on a single geometric parameter to address overlapping instances, our approach uses a comprehensive set of lane line geometric features, including area, mask Intersection over Union (IoU), coordinate distribution, and geometric centroid. This multi-faceted geometric analysis enables more precise distinction between adjacent instances, overcoming the limitations of single-parameter approaches.
- Simplified Integration: Unlike network and loss-based methods that require modifications to network architecture or training processes, our approach operates in the post-processing stage, following network output. This design avoids complex retraining and parameter adjustments, offering a plug-and-play solution that can be easily integrated across various platforms.
- Superior Real-Time Performance: While NMS-free approaches can directly recognize instances without NMS, they often exhibit slower performance. Our method, by contrast, maintains high-speed processing suitable for real-time lane detection scenarios, ensuring it meets the stringent demands of practical applications.

5. Experimental Results and Analysis

In this section, ablation and comparative experiments are conducted to verify the effectiveness and efficiency of our proposed GeoLaneNet. Finally, visualization results of representative baselines of examples in InstLane are given. It should be noted that, in popular open-source datasets, the lane line captured by front-mounted sensors is thin and long. In InstLane, the area of lane markings in our research is relatively large, and we are concerned more with the quality of the generation of lane marking instances. Therefore, in the experiment, we only use GeoLaneNet and the current instance segmentation SOTA method for natural scenes for comparison.

5.1. Experimental Settings and Metrics

Training and experiments are performed on NVIDIA RTX3090 and Intel I9-13900K. The operating system and deep learning platforms used are Ubuntu 22.04 and PyTorch 1.13.0. The configuration settings are as follows. In the experiment, the learning rate is set to 0.001, and it decreases by 0.1 after every 20 epochs. The batch size is set to 32, and the number of epochs is set to 200. The hyperparameters α, β, γ, ξ, δ in Equations (11) and (12) are set to be 0.25, 0.25, 0.25, 0.25, 0.75, respectively. The threshold of NMS is set to 0.75. The input size is set to 550×550. We follow the standard protocol where the overall performance, in terms of mask average precision (AP), is measured by averaging over multiple intersection-over-union (IoU) thresholds, ranging from 0.5 to 0.95, with an interval of 0.05. In the experiment involving ResNet18-DDBlock, we loaded pretrained weights on ImageNet [54] for the parts of the original ResNet18, and the parameters of DDBlock were randomly initialized.

5.2. Ablation Study

5.2.1. Deep Perception Learning Module

To evaluate the effectiveness of the Deep Perception Learning Module, DDBlocks are added layer by layer into the original ResNet18. MACs, model parameters, AP, and FPS are measured on the modified Resnet18, together ResNet50, D-LinkNet34(Enc) [45] and ConvNext-Small [55]. The results are shown in Table 4.

From the results, the FPS of modified ResNet18 is much faster than ResNet50, while the performance in terms of AP is very close, showing that enlarging the receptive field can improve AP and maintain rapid detection.

Table 4. Comparisons of the results for different numbers of DDBlocks in ResNet18, together with ResNet50, D-LinkNet34(Encoder) and ConvNext-Small.

Model	MACs(G)	Params(M)	AP(%)	FPS
ResNet18 (baseline)	40.71	17.58	64.69	269
ResNet18+DDBlock 1	42.95	17.70	+1.13	252
ResNet18+DDBlock 1, 2	43.52	17.82	+1.38	235
ResNet18+DDBlock 1, 2, 3	43.69	17.96	+2.26	211
ResNet18+DDBlock 1, 2, 3, 4	43.78	18.25	+2.44	198
ResNet50	55.91	30.60	+2.76	153
D-LinkNet34 (Enc) [45]	64.49	46.57	+2.68	158
ConvNext-Small [55]	80.91	56.10	+2.89	89

In addition, we visualize the effective receptive field (ERF) of the proposed ResNet18-DDBlock and several typical backbones. The results are illustrated in Figure 9. It is evident that our approach yields a larger effective receptive field than some lightweight backbones (e.g., ResNet18 [46], ShuffleNetV2_x1.0 [56] and MobileNetV3_large [57]) and brings an analogous effective receptive field of ConvNext_small [55], D-LinkNet34 (encoder) [45] and ResNet50 [46]. This substantiates that ResNet18-DDBlock can effectively increase the size of ERF while preserving detection speed.

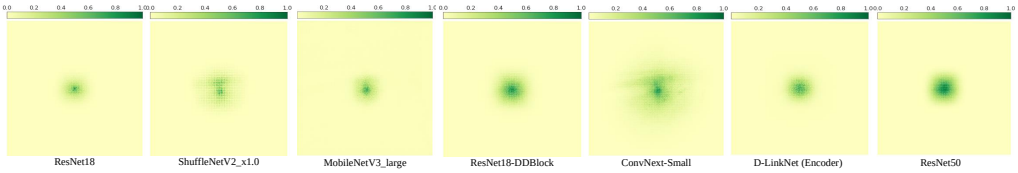


Figure 9. Visualization of effective receptive field (ERF) of ResNet18 and 50 [46], ShuffleNetV2 [56], MobileNetV3 [57], ConvNext-Small [55] and D-LinkNet34 (Encoder) [45]. Wider dark green area corresponds to a larger ERF.

5.2.2. Geometry-Aware Fine Localization

The effectiveness is verified by different settings of NMS with or without Score Filter by Geometry Features. ResNet18-DDBlock is used as a backbone. The results are shown in Table 5. As is demonstrated, Cluster NMS with SPM and Distance penalty terms have higher AP than Cluster NMS, particularly higher than that of traditional NMS, Fast NMS [36], Cluster NMS (score SPM).

Table 5. Mask AP results of optional NMS with or without score filter by geometry features. The backbone is ResNet18-DDBlock.

NMS	Score Filter by Geometry Features	AP (%)
Fast NMS [36]		67.13
	✓	71.79
Traditional NMS		67.28
	✓	**72.02**
Soft NMS [48]		68.52
	✓	72.93
Cluster NMS [47]		69.37
	✓	**73.29**
Cluster NMS (SPM) [47]		70.57
	✓	**73.50**
Cluster NMS (SPM & Dist) [47]		71.85
	✓	**73.55**

5.2.3. Ablation Study of Proposed Modules

Table 6 shows the effectiveness of the Deep Perception Learning Module, the Geometry-Aware Fine Localization, and the combined final accuracy. As shown, Geometry-Aware Fine Localization's contribution is bigger than that of the Deep Perception Learning Module.

Table 6. Experiments of proposed modules of GeoLaneNet on InstLane. Baseline stands for YolactEdge-ResNet18 with Fast NMS.

Deep Perception Learning Module	Geometry-Aware Fine Localization	AP (%)	FPS
		64.69	269
✓		67.13 (+2.44)	198
	✓	72.24 (+7.55)	175
✓	✓	73.55 (+8.86)	139

5.3. Comparison with other Baselines

We compare the proposed GeoLaneNet with several different types of baselines that perform well on natural classical instance segmentation, including contour-based algorithms of PolarMask [31] and Dance [32], NMS-free algorithms of SparseInst [38] and K-Net [37], NMS-based algorithms of Mask RCNN [58], CenterMask [59], SOLOv2 [34], YOLACT [35], and original YolactEdge [35]. The results are shown in Table 7. It can be observed that GeoLaneNet reaches AP of 73.55%, close to SparseInst and K-Net with a much higher FPS. Additionally, we measured the GPU memory metric of models by `torch.cuda.memory_allocated()` to evaluate the real runtime environments, and the results show that our GeoLaneNet is GPU-friendly. In general, GeoLaneNet can achieve two to three times the speed of the NMS-Free methods while maintaining a tolerable loss in AP.

Table 7. Comparison results of GeoLaneNet with baselines for natural instance segmentation.

Model	Contour	NMS	Backbone	AP(%)	FPS	GPU Mem(M)
PolarMask [31]	✓	✓	ResNet50	53.64	40	135
Dance [32]	✓	✓	ResNet50	64.66	56	174
SparseInst [38]			ResNet50	**78.08**	67	128
K-Net [37]			ResNet50	**76.50**	43	146
MaskRCNN [58]		✓	ResNet50	56.64	36	175
CenterMask [59]		✓	VoVNet39 [60]	62.35	44	134
SOLOv2 [34]		✓	ResNet18	70.50	49	**90**
YOLACT [36]		✓	ResNet50	65.52	62	153
YolactEdge [35]		✓	ResNet50	64.69	**159**	145
GeoLaneNet		✓	ResNet18-DDBlock	73.55	**139**	**98**

The result of bolding and underlining the **font** at the same time indicates the 1st in the result comparison; bolding the **font** only indicates the 2nd place in the result; and underlining the font only indicates the 3rd place.

5.4. Visualization Results

Figure 10 shows the examples of instance segmentation results in the testval set of InstLane of GeoLaneNet. The results show that the lane lines can also be detected well in the challenging scenes of exposure environments, uneven illumination environments, and high noise environments, which fully demonstrates the effectiveness and robustness of GeoLaneNet.



Figure 10. Visualization examples of the testval set in InstLane outputted by GeoLaneNet. (**a**) shows normal scenes. (**b**) shows the dazzle light scenes at nighttime. (**c**) shows scenes of uneven illumination with shadow. (**d**) shows grayscale scenes with more noise and inapparent lane lines.

Visualization results in dense lane line scenes of some baselines and our GeoLaneNet are presented in Figure 11. The contour-based method PolarMask does not detect and segment such large objects as lane lines. There is an "NMS Dilemma" in the results of NMS-based algorithms such as SOLOv2 and YolactEdge. NMS-free algorithms such as SparseInst and K-Net and our GeoLaneNet all mitigate the "NMS Dilemma" in dense lane line scenarios.

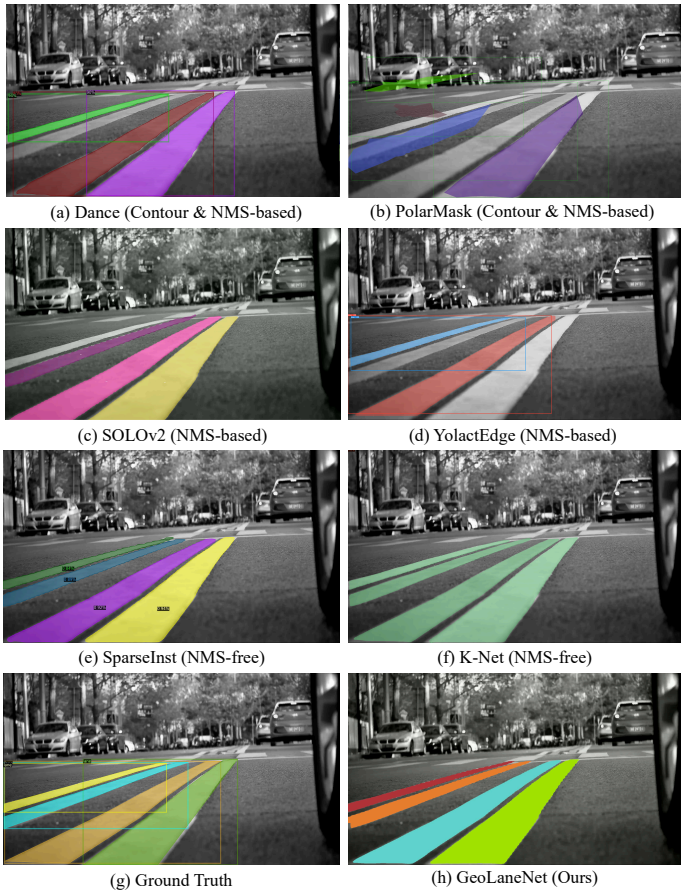


Figure 11. Visualization results of different approaches in dense lane lines scenes.

5.5. Discussion of Limitations

Even though our algorithm achieves good results that balance both accuracy and speed, and mitigates the shortcomings of NMS-based algorithms in the face of the NMS dilemma. Our algorithm still has some flaws and we hope to improve it in the future.

Firstly, our algorithm is an improvement upon existing NMS-based instance segmentation algorithms and still relies on the NMS algorithm. Therefore, the NMS dilemma is an unavoidable issue. While our module significantly optimizes or mitigates the occurrence of this situation, it cannot be completely avoided.

Secondly, our module introduces multiple geometric feature considerations to mitigate the NMS dilemma caused by using a single NMS overlap threshold. However, incorporating multiple geometric factors introduces additional hyperparameters. These hyperparameters can make the algorithm less flexible.

Lastly, the Geometry-Aware Fine Localization module in our algorithm uses serial processing rather than parallel matrix operations, which can slow down the inference speed. Despite the fact that our current experimental results show the speed to be twice that of the state-of-the-art, future efforts should focus on achieving matrix processing parallelization to further improve inference speed.

6. Conclusions and Prospects

In this paper, we have introduced InstLane, the first-ever instance-level dataset designed explicitly for lane line detection. It is intended to serve as a benchmark for evaluating instance segmentation-based methodologies in the field. We have also unveiled GeoLaneNet, a novel, high-performance instance segmentation algorithm for lane line detection. This network is uniquely designed to leverage features of varying depths and receptive fields, thereby enhancing detection capabilities. Furthermore, it introduces feature transformation between different frames to accelerate inference speed and fully exploit

geometric features, thereby mitigating the common issue known as the "NMS Dilemma". The performance of GeoLaneNet was rigorously evaluated and compared against several baseline models. The results conclusively demonstrated its effectiveness, robustness, and efficiency in lane line detection. For future works, we aim to introduce semi-supervised guidance for automatically labeling lane lines, thereby addressing the challenge of labor-intensive manual work. Through this approach, we aim to streamline the lane line detection process further and contribute to the evolution of autonomous driving technology.

Author Contributions: Conceptualization, Q.C. and J.L.; methodology, J.L.; software, J.L.; validation, J.L.; formal analysis, J.L.; investigation, J.L. and Y.Y.; resources, J.L. and Y.Y.; data curation, J.L., H.L., and K.L.; writing—original draft preparation, J.L.; writing—review and editing, Q.C., J.L., and X.H.; visualization, J.L.; supervision, Q.C.; project administration, Q.C. and J.L.; funding acquisition, Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under Grants No. 42271352.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be provided by request from emails of nonprofit institutions such as universities and research institutes.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Guo, X.; Cao, Y.; Zhou, J.; Huang, Y.; Li, B. HDM-RRT: A Fast HD-Map-Guided Motion Planning Algorithm for Autonomous Driving in the Campus Environment. *Remote Sens.* **2023**, *15*, 487. [CrossRef]
2. Yan, S.; Zhang, M.; Peng, Y.; Liu, Y.; Tan, H. AgentI2P: Optimizing Image-to-Point Cloud Registration via Behaviour Cloning and Reinforcement Learning. *Remote Sens.* **2022**, *14*, 6301. [CrossRef]
3. Aldibaja, M.; Suganuma, N.; Yanase, R. 2.5D Layered Sub-Image LIDAR Maps for Autonomous Driving in Multilevel Environments. *Remote Sens.* **2022**, *14*, 5847. [CrossRef]
4. Ling, J.; Chen, Y.; Cheng, Q.; Huang, X. Zigzag Attention: A Structural Aware Module For Lane Detection. In Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 4175–4179. [CrossRef]
5. Tabelini, L.; Berriel, R.; Paixao, T.M.; Badue, C.; De Souza, A.F.; Oliveira-Santos, T. Polylanenet: Lane estimation via deep polynomial regression. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; IEEE: Piscataway, NJ, USA, 2020; pp. 6150–6156.
6. Feng, Z.; Guo, S.; Tan, X.; Xu, K.; Wang, M.; Ma, L. Rethinking Efficient Lane Detection via Curve Modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 17062–17070.
7. Li, X.; Li, J.; Hu, X.; Yang, J. Line-CNN: End-to-End Traffic Line Detection With Line Proposal Unit. In *IEEE Transactions on Intelligent Transportation Systems*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–11.
8. Qin, Z.; Wang, H.; Li, X. Ultra fast structure-aware deep lane detection. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 276–291.
9. Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Towards end-to-end lane detection: An instance segmentation approach. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 286–291.
10. Pan, X.; Shi, J.; Luo, P.; Wang, X.; Tang, X. Spatial as deep: Spatial cnn for traffic scene understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 7276–7283.
11. Cheng, W.; Luo, H.; Yang, W.; Yu, L.; Chen, S.; Li, W. Det: A high-resolution dvs dataset for lane extraction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 15–20 June 2019; pp. 1666–1675.
12. Gurghian, A.; Koduri, T.; Bailur, S.V.; Carey, K.J.; Murali, V.N. DeepLanes: End-To-End Lane Position Estimation Using Deep Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, Nevada, USA, 26 June–1 July 2016; pp. 38–45.
13. Tabelini, L.; Berriel, R.; Paixao, T.M.; Badue, C.; Souza, A.; Oliveira-Santos, T. Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 294–302.
14. Qin, Z.; Zhang, P.; Li, X. Ultra Fast Deep Lane Detection With Hybrid Anchor Driven Ordinal Classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *46*, 2555–2568. [CrossRef] [PubMed]

15. Van Gansbeke, W.; De Brabandere, B.; Neven, D.; Proesmans, M.; Van Gool, L. End-to-end lane detection through differentiable least-squares fitting. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Repulic of Korea, 27–28 October 2019; pp. 905–913.
16. Han, J.; Deng, X.; Cai, X.; Yang, Z.; Xu, H.; Xu, C.; Liang, X. Laneformer: Object-aware Row-Column Transformers for Lane Detection. *arXiv* **2022**, arXiv:2203.09830.
17. Tusimple. 2019. Available online: https://github.com/TuSimple/tusimple-benchmark (accessed on 11 May 2023).
18. Wang, P.; Huang, X.; Cheng, X.; Zhou, D.; Geng, Q.; Yang, R. The apolloscape open dataset for autonomous driving and its application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2702–2719.
19. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2636–2645.
20. Behrendt, K.; Soussan, R. Unsupervised Labeled Lane Markers Using Maps. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Repulic of Korea, 27–28 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 832–839.
21. Aly, M. Real time detection of lane markers in urban streets. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 7–12.
22. Xu, H.; Wang, S.; Cai, X.; Zhang, W.; Liang, X.; Li, Z. Curvelane-NAS: Unifying lane-sensitive architecture search and adaptive point blending. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin, Germany, 2020; pp. 689–704.
23. Wu, T.; Ranganathan, A. A practical system for road marking detection and recognition. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 25–30.
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; NIPS'17; pp. 6000–6010.
26. Hou, Y.; Ma, Z.; Liu, C.; Loy, C.C. Learning lightweight lane detection cnns by self attention distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Repulic of Korea, 27 October–2 November 2019; pp. 1013–1021.
27. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv* **2016**, arXiv:1606.02147.
28. Wen, T.; Yang, D.; Jiang, K.; Yu, C.; Lin, J.; Wijaya, B.; Jiao, X. Bridging the Gap of Lane Detection Performance Between Different Datasets: Unified Viewpoint Transformation. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 6198–6207. [CrossRef]
29. Sun, Y.; Li, J.; Xu, X.; Shi, Y. Adaptive Multi-Lane Detection Based on Robust Instance Segmentation for Intelligent Vehicles. *IEEE Trans. Intell. Veh.* **2023**, *8*, 888–899. [CrossRef]
30. Peng, S.; Jiang, W.; Pi, H.; Li, X.; Bao, H.; Zhou, X. Deep snake for real-time instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8533–8542.
31. Xie, E.; Sun, P.; Song, X.; Wang, W.; Liu, X.; Liang, D.; Shen, C.; Luo, P. Polarmask: Single shot instance segmentation with polar representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12193–12202.
32. Liu, Z.; Liew, J.H.; Chen, X.; Feng, J. Dance: A deep attentive contour model for efficient instance segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 345–354.
33. Zhang, T.; Wei, S.; Ji, S. E2EC: An End-to-End Contour-based Method for High-Quality High-Speed Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4443–4452.
34. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. Solov2: Dynamic and fast instance segmentation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17721–17732.
35. Liu, H.; Soto, R.A.R.; Xiao, F.; Lee, Y.J. Yolactedge: Real-time instance segmentation on the edge. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; IEEE: Piscataway, NJ, USA, 2021, pp. 9579–9585.
36. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Repulic of Korea, 27 October–2 November 2019; pp. 9157–9166.
37. Zhang, W.; Pang, J.; Chen, K.; Loy, C.C. K-net: Towards unified image segmentation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 10326–10338.
38. Cheng, T.; Wang, X.; Chen, S.; Zhang, W.; Zhang, Q.; Huang, C.; Zhang, Z.; Liu, W. Sparse Instance Activation for Real-Time Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4433–4442.
39. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.

40. Yu, F.; Koltun, V.; Funkhouser, T. Dilated Residual Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
41. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11963–11975.
42. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
43. Mehta, S.; Rastegari, M.; Caspi, A.; Shapiro, L.; Hajishirzi, H. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 552–568.
44. Li, Y.; Hou, Q.; Zheng, Z.; Cheng, M.M.; Yang, J.; Li, X. Large Selective Kernel Network for Remote Sensing Object Detection. *arXiv* **2023**, arXiv:2303.09030.
45. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 182–186.
46. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
47. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [CrossRef]
48. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS–improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5561–5569.
49. He, Y.; Zhu, C.; Wang, J.; Savvides, M.; Zhang, X. Bounding box regression with uncertainty for accurate object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
50. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Occlusion-aware R-CNN: Detecting Pedestrians in a Crowd. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
51. Wang, X.; Xiao, T.; Jiang, Y.; Shao, S.; Sun, J.; Shen, C. Repulsion Loss: Detecting Pedestrians in a Crowd. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
52. Liu, S.; Huang, D.; Wang, Y. Adaptive NMS: Refining Pedestrian Detection in a Crowd. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
53. Chu, X.; Zheng, A.; Zhang, X.; Sun, J. Detection in Crowded Scenes: One Proposal, Multiple Predictions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
54. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
55. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 11966–11976. [CrossRef]
56. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
57. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
58. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE international conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
59. Lee, Y.; Park, J. Centermask: Real-time anchor-free instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13906–13915.
60. Lee, Y.; Hwang, J.w.; Lee, S.; Bae, Y.; Park, J. An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 15–20 June 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.