sensors

MDPI

Review

Vehicle Detection Algorithms for Autonomous Driving: A Review

Liang Liang [1], Haihua Ma [1,2,*], Le Zhao [1,2], Xiaopeng Xie [1], Chengxin Hua [1], Miao Zhang [1,2] and Yonghui Zhang [1]

[1] College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China; liangliang@stu.haut.edu.cn (L.L.); lezhao@haut.edu.cn (L.Z.); xiaopengxie@stu.haut.edu.cn (X.X.)

[2] Key Laboratory of Grain Information Processing and Control of Ministry of Education, Henan University of Technology, Zhengzhou 450001, China

[*] Correspondence: mahaihua@haut.edu.cn

Abstract: Autonomous driving, as a pivotal technology in modern transportation, is progressively transforming the modalities of human mobility. In this domain, vehicle detection is a significant research direction that involves the intersection of multiple disciplines, including sensor technology and computer vision. In recent years, many excellent vehicle detection methods have been reported, but few studies have focused on summarizing and analyzing these algorithms. This work provides a comprehensive review of existing vehicle detection algorithms and discusses their practical applications in the field of autonomous driving. First, we provide a brief description of the tasks, evaluation metrics, and datasets for vehicle detection. Second, more than 200 classical and latest vehicle detection algorithms are summarized in detail, including those based on machine vision, LiDAR, millimeter-wave radar, and sensor fusion. Finally, this article discusses the strengths and limitations of different algorithms and sensors, and proposes future trends.

Keywords: autonomous driving; vehicle detection; sensors; sensor fusion; deep learning

check for updates

Citation: Liang, L.; Ma, H.; Zhao, L.; Xie, X.; Hua, C.; Zhang, M.; Zhang, Y. Vehicle Detection Algorithms for Autonomous Driving: A Review. *Sensors* **2024**, *24*, 3088. https://doi.org/10.3390/s24103088

Academic Editor: Enrico Meli

Received: 27 March 2024
Revised: 7 May 2024
Accepted: 10 May 2024
Published: 13 May 2024

1. Introduction

The advancement in technology is gradually permeating into the lives of people, with autonomous driving being at the forefront. In particular, autonomous vehicles (AVs) can eliminate 94% of road accidents caused by human error and distracted driving [1]. Against this backdrop, automated driving systems (ADSs) have emerged with the promise of preventing accidents, reducing emissions, transporting the mobility-impaired, and taking the stress out of driving [2]. Autonomous vehicles can be classified into six levels based on the degree of human intervention and attention required, denoted as L0 to L5, each signifying different degrees of autonomy [3]. Currently, most IVs can only achieve partial autonomous driving functions, such as lane-keeping, intelligent speed limitation, adaptive cruise control, etc. The realization of fully automated driving still has a long way to go.

Autonomous driving systems typically consist of three components: environmental perception, behavioral decision making, and motion planning and control [4]. Environmental perception serves as the prerequisite and foundation of autonomous driving [5]. In particular, robust and reliable vehicle detection has been a topic of great interest [6]. Vehicle detection, the ability of a vehicle to perceive its surrounding vehicles in real-world driving scenarios, holds critical importance across various domains, including intelligent transportation, military defense, security surveillance, and autonomous driving [7]. According to traffic accident statistics, the main threat to drivers often comes from other vehicles [8]. Therefore, the efficient sensing and accurate recognition of the surrounding environment are paramount for ensuring the safety of self-driving vehicles. In order for autonomous vehicles to function effectively, they must be aware of other vehicles in a timely manner, allowing them to formulate safe and reliable plans [9]. Given the potential for closing speeds between vehicles, this necessitates the ability to accurately detect vehicles.

Moreover, the performance of vehicle detection directly influences the quality of decision making and control in autonomous vehicles. Detecting vehicles using different sensors is a significant challenge due to the various characteristics of vehicles, such as size, occlusion, orientation, and shadows [10]. Additionally, the time-sensitive nature of vehicle detection, requiring faster processing than other applications, further complicates the task [11]. Therefore, precise vehicle detection is crucial for the automation and intelligence of vehicles. With the rapid development of deep learning, sensor technologies, and the Internet of Things (IoT), more and more new methods and technologies have emerged and are gradually being applied in the field of vehicle detection [12,13]. This paper focuses on sensors and summarizes more than 200 classical and latest vehicle detection algorithms in recent years. This paper also analyzes the tasks, evaluation metrics, and existing public datasets for vehicle detection and presents the future trends of vehicle detection.

The rest of this article is organized as follows: Section 2 describes the tasks, evaluation metrics, and existing public datasets for vehicle detection. Then, Section 3 introduces vision-based vehicle detection algorithms, focusing on the application of deep learning methods. Next, vehicle detection methods based on radar and LiDAR are delineated in Section 4. Section 5 provides an integrated analysis of Sections 3 and 4, encompassing the implementation of various sensor fusion techniques. Section 6 discusses the different sensors and algorithms for vehicle detection, and offers future trends. Finally, Section 7 is the conclusion.

2. Preliminaries for Vehicle Detection

Intelligent vehicles provide drivers with information regarding safety, assistance, and comfort. In environmental perception, demands arise in complex road scenarios to detect and assess various targets in real time and evaluate the effectiveness of detection indicators. This section mainly describes the detection tasks and metrics for intelligent vehicles and introduces some public datasets for vehicle detection.

2.1. Tasks

Vehicle detection is of crucial importance in the environment perception framework of intelligent vehicle systems. It facilitates positioning and classifying diverse entities, including pedestrians, non-motorized vehicles, traffic signage, and lane demarcations within road environments. Vehicle detection is divided into 2D object detection and 3D object detection, and both of them are widely applied in vehicle detection tasks. Two-dimensional object detection serves as a fundamental technique in the realm of vehicle detection. It entails utilizing a 2D bounding box within the visual field of intelligent vehicles to select detected objects, and then the selected objects are classified and positioned. The 3D object detection system displays the specific position of the vehicles in the camera coordinate system. This process requires 3D bounding boxes to select the detected objects, followed by their classification and localization [14].

2.2. Evaluation Metrics

Metrics such as Intersection over Union (IoU), precision (P), recall (R), F1-score, Average Precision (AP), and mean Average Precision (mAP) are commonly utilized to evaluate detection accuracy of algorithms. IoU quantifies the degree of overlap between predicted bounding boxes and ground truth bounding boxes. Precision indicates the proportion of predicted positive samples that are true positives, whereas recall signifies the proportion of actual positive samples that are correctly identified. F1-score offers a combined measure of precision and recall. These metrics are computed using specific formulas:

$$\mathrm{IoU} = \frac{\mathrm{p_b} \cap \mathrm{g_b}}{\mathrm{p_b} \cup \mathrm{g_b}} \tag{1}$$

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{4}$$

where $\mathrm{p_b}$ and $\mathrm{g_b}$ represent predicted boxed and ground truth boxes, respectively; TP denotes the count of positive cases accurately identified as true samples; FP indicates the tally of negative cases erroneously identified as true samples; and FN represents the number of positive cases mistakenly classified as false samples. Additionally, AP measures the precision performance for an individual class, while mAP offers the mean of AP values across all classes. These metrics are formulated using the following equations:

$$\mathrm{AP} = \int_0^1 \mathrm{P(R)dR} \tag{5}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{n}} \sum_{i=1}^{\mathrm{n}} \mathrm{P(i)\Delta R(i)} \tag{6}$$

where n is the number of categories in detection targets.

In the realm of assessing detection speed, parameters, FLOPs (Floating Point Operations), and FPS (Frames Per Second) are essential metrics. Parameters denote the total count of trainable parameters involved in the model training, often used to gauge the size of the model. FLOPs quantify the model's complexity. The lower value of FLOPs indicates reduced computational load for model inference. FPS assesses the detection speed by indicating the number of frames processed per second.

2.3. Datasets

Datasets are indispensable for training models in each task of autonomous driving. High-quality annotated data, such as images, videos, and sensor data from various scenarios, are the basis for training autonomous driving systems and machine learning models, and are manually labeled to indicate information about correct behavior, object detection, and environment perception. In order to propel and invigorate the field of autonomous driving, industry organizations and researchers have produced many high-quality datasets. Table 1 summarizes some essential information from these datasets. We list a number of items, including year, location, scene, category, annotation, 3D boxes, and application scenarios.

Table 1. Datasets for vehicle detection. Sc. stands for Scenes, Cl. for Classes, An. for Annotations, and 3Db. for 3D boxes.

Dataset	Year	Loc.	Sc.	Cl.	An.	3Db.	Application Scenarios
KITTI [15]	2012	Karlsruhe (DE)	22	8	15 k	200 k	Multiple application scenarios.
Cityscapes [16]	2016	50 cities	-	30	25 k	-	Mainly oriented to segmentation tasks.
Oxford RobotCar [17]	2016	Central Oxford (UK)	-	-	-	-	Multimodal joint calibration tasks can be conducted.
Vistas [18]	2017	Global	-	152	25 k	-	Globally constructed dataset for autonomous driving.
BDD100K [19]	2018	San Fransisco and New York (US)	100 k	10	100 k	-	The total volume of data is enormous, nearly 2 terabytes.
ApolloScape [20]	2018	4 cities in CN	-	8–35	144 k	70 k	It contains many extensive and richer labels.

Table 1. *Cont.*

Dataset	Year	Loc.	Sc.	Cl.	An.	3Db.	Application Scenarios
KAIST [21]	2018	South Korea	-	3	8.9 k	-	Primarily targets SLAM tasks, emphasizing the provision of examples in complex scenarios.
Waymo open [22]	2019	6 cities in US	1 k	4	200 k	12 M	Focused on computer vision tasks, and utilizes data collected in all-weather conditions.
Lyft L5 [23]	2019	California (US)	366	-	-	55 k	More than 1000 h of driving record data.
Argoverse [24]	2019	Pittsburgh and Miami (US)	1 k	-	22 k	993 k	Focus on two tasks: 3D tracking and action prediction.
D^2-City [25]	2019	5 cities in CN	1 k	12	700 k	-	Suitable for detection and tracking tasks.
H3D [26]	2019	San Francisco (US)	160	8	27 k	1.1 M	It is a large-scale full-surround 3D multi-object detection and tracking dataset.
nuScenes [27]	2019	Boston (US), Singapore	1 k	23	40 k	1.4 M	It was taken in dense traffic and highly challenging driving situations.
CADC [28]	2020	Waterloo (CA)	75	10	7 k	-	Focused on constructing a dataset for driving in snowy conditions.
A2D2 [29]	2020	3 cities in DE	-	14	12 k	43 k	Perception for autonomous driving.
A*3D [30]	2020	Singapore	-	7	39 k	230 k	With a significant diversity of the scene, time, and weather.
RADIATE [31]	2021	UK	7	8	-	-	Focus on tracking and scene understanding using radar sensors in adverse weather.
ACDC [32]	2021	Switzerland	4	19	4.6 k	-	A larger semantic segmentation dataset on adverse visual conditions.
KITTI-360 [33]	2022	Karlsruhe (DE)	-	37	-	68 k	An extension of the KITTI dataset. It established benchmarks for tasks relevant to mobile perception.
SHIFT [34]	2022	8 cities	-	23	2.5 M	2.5 M	A synthetic driving dataset for continuous multi-task domain adaptation.
Argoverse 2 [35]	2023	6 cities in US	250 k	30	-	-	The successor to the Argoverse 3D tracking dataset. It is the largest ever collection of LiDAR sensor data.
V2V4Real [36]	2023	Ohio (US)	-	5	20 k	240 k	The first large-scale real-world multimodal dataset for V2V perception.

3. Vehicle Detection Algorithms Based on Machine Vision

Machine vision systems are considered a promising research field with broad applications in various detection scenarios. Machine-vision-based sensors are the earliest and most widely used sensors for vehicle detection [37]. These types of sensors are typically referred to as passive sensors since they solely capture images of the objects without the need for specialized illumination projection devices. Vision-based sensors typically have access to a rich set of perception information from the traffic environment, such as textures, colors, lane markings, obstacle identifications, and semantics. In the past few decades, the rapid development of computer information and sensor technology has led to the widespread adoption of sensor combinations based on multiple visual modalities. For example, companies like Tesla [38] and Mobileye [39] have embraced pure vision solutions for intelligent vehicle environment perception. According to the different principles of existing algorithms, vehicle detection techniques based on machine vision can be categorized into three components: traditional-based, machine learning-based, and deep learning-based techniques [4].

3.1. Traditional-Based Methods for Vehicle Detection

Inherent appearance features of vehicles can be exploited in traditional-based vehicle detection methods that typically include two main steps: hypothesis generation (HG) and hypothesis verification (HV). In the HG stage, the system extracts a region of interest (ROI) based on the appearance features of the detected vehicle. And then in the HV stage, the system confirms whether the ROI contains vehicle targets. In other words, HG is the backbone of the process, while HV is the further verification and validation of the generated hypotheses, both of which are necessary. Depending on the traffic scenarios, the appearance features of vehicles can generally be categorized into the following six common types.

- Color: Due to the continuity and concentration of the color distribution of the vehicles in the image, the vehicles can be separated from the image background by applying different color channels and setting appropriate segmentation thresholds [40,41]. However, techniques based on color features are susceptible to variations in illumination and specular reflections [42].
- Symmetry: Most cars have symmetrical rear ends. By leveraging this feature, we can search for regions with high symmetry on ROI in the image to obtain vehicle information, resulting in the identification of vehicle objects and non-vehicle objects. Moreover, symmetry can not only help to optimize the bounding boxes of vehicles, but also be employed to confirm if the ROI includes targets for vehicles in the HV stages [43]. However, the computation of symmetry increases the overhead of time and reduces detection efficiency.
- Edges: Vehicle features such as silhouettes, bumpers, rear windows, and license plates exhibit strong linear textures in both vertical and horizontal directions [44]. Extracting these typical edge features from the image allows for a further determination of the car's bounding box [45,46]. However, the edge lines may tend to overlap with some texture lines of the image background, which may lead to the appearance of false positives in particular scenes.
- Texture: Typically, road textures exhibit a relatively uniform distribution, whereas textures on car surfaces tend to be less uniform due to the presence of highly varied regions. We can indirectly perform vehicle detection by distinguishing the difference between these two conditions [47]. However, relying on feature textures to detect vehicles may result in low detection accuracy.
- Shadows: In bright daylight, the vehicles traveling on the road cast stable shadows underneath. The shadowed region clearly exhibits a lower gray value compared to the remaining road areas. Utilizing segmentation thresholds enables the extraction of the underlying shadow as the ROI for vehicles during the HG stage [48,49]. However, the application scenarios of this approach are relatively limited.

- Taillights: Vehicle detection at night is often achieved through the use of taillight features due to the noticeable color (usually red). It is easy to extract information from it through image processing techniques [50,51]. However, this method is only effective for detecting vehicles at night.

Traditional vehicle detection methods have the advantages of a low cost, a fast detection speed, and simple working principles. However, these methods are based on prior knowledge and hence are mainly susceptible to interference from other objects. An effective approach is to use the fusion of multiple features for detection [52]. In addition, the extracted ROI can be used as basic feature information and then modeled using machine learning or deep learning methods [48,53].

3.2. Machine Learning-Based Methods for Vehicle Detection

With the rapid development of computer technologies, machine learning (ML) has become a popular issue in the realm of vehicle detection. ML, an essential branch within the fields of artificial intelligence (AI) and computer science, is dedicated to using data and algorithms to emulate how humans learn. An ML model transforms and encodes vehicle images using manually designed features and applies a particular mapping method to convert high-dimensional image space data to low-dimensional image space data. The model is then trained continuously to receive a final model for vehicle detection. Typically, vehicle detection using machine learning models can be divided into two key steps: first, the input image is processed to obtain the ROI; second, the extracted image features are fed into a classifier for training and optimization.

3.2.1. Feature Extraction

Ease to extract and identify, while preserving stable vehicle characteristics when the vehicle attitude and type change, is a necessary quality of an effective feature extraction technique. The histogram of oriented gradients (HOG) is one of the popular methods for feature extraction in the field of object detection. It initially achieved significant success in pedestrian detection [54], and has since expanded to other application domains, such as vehicle detection and face recognition. Many researchers have improved upon the HOG algorithm, such as two HOG vectors [55], the pyramid of HOG [56], and symmetry HOG [57]. The deformable part model (DPM) employs the improved HOG descriptor and adopts a multi-component strategy [58]. The Haar-like vector is also a fundamental descriptor used for face detection and was later extended to vehicle detection [59]. Some other feature extraction methods are also frequently used for vehicle detection, such as a local binary pattern (LBP) [60], Gabor filters [61], and sped-up robust features (SURFs) [62]. Moreover, some studies have shown that the fusion of multiple feature descriptors may result in a richer representation of vehicles [63,64].

3.2.2. Classifier

An ML classifier can distinguish vehicle and non-vehicle targets based on the local features collected from the image. In general, a classifier needs to be trained on well-labeled datasets first, with boundaries drawn between positive samples and negative samples. AdaBoost, K-nearest neighbor (KNN), Naive Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT) are the more commonly used classifiers for vehicle detection. The selection of a classifier requires the consideration of both its generalization ability and fitting accuracy. The generalization ability determines how well the model adapts to new data, while the fitting accuracy measures whether the classifier has a sufficient fit on the training data to identify patterns and associated information accurately. Ensemble learning is a classic machine learning method, combining predictions from multiple base classifiers to enhance overall predictive performance [65,66]. Different studies on feature engineering and classifiers for vehicle detection are shown in Table 2. Vehicle detection methods based on machine learning typically require scanning the entire image to extract features. However, this process increases computational costs and time consumption. Ref. [67]

notes that more than half of the image area contains no vehicle information. The use of traditional-based feature extraction combined with a classifier has proved to be an effective approach to address this difficulty. For instance, the ROI for the vehicle was extracted from the image by utilizing the shadow. Then, Haar-like features and the AdaBoost classifier were employed to detect the vehicles from the ROI [55].

Table 2. Different studies on feature engineering and classifiers for vehicle detection.

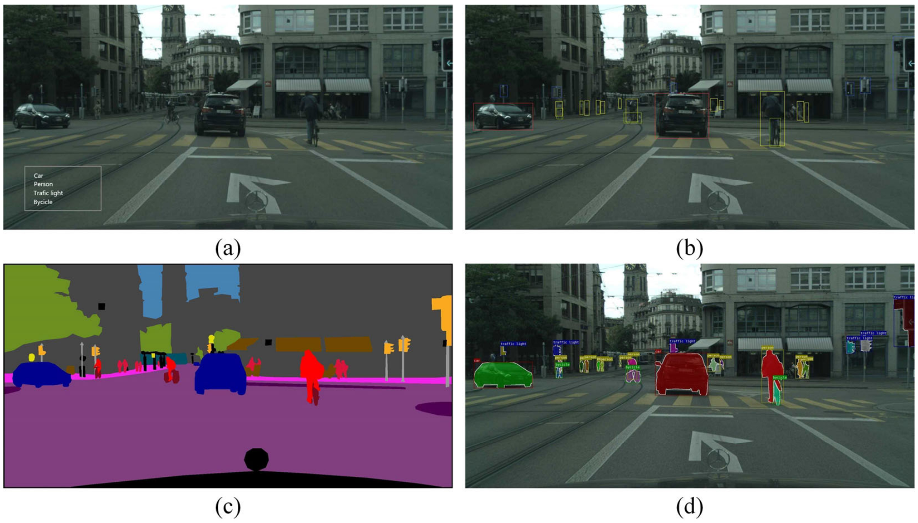
Feature	Classifier	Dataset	Accuracy	Reference
HOG	Adaboost	GTI vehicle database and real traffic scene videos	98.82%	[55]
HOG	GA-SVM	1648 vehicles and 1646 non-vehicles	97.76%	[56]
HOG	SVM	420 road images from real on-road driving tests	93.00%	[57]
HOG	SVM	GTI vehicle database and another 400 images from real traffic scenes	93.75%	[68]
Haar-like	Adaboost	Hand-labeled data of 10,000 positive and 15,000 negative examples	-	[69]
SURF	SVM	2846 vehicles from 29 vehicle makes and models	99.07%	[70]
PCA	SVM	1051 vehicle images and 1051 nonvehicle images	96.11%	[71]
SIFT	SVM	880 positive samples and 800 negative samples	-	[72]

3.3. Deep Learning-Based Methods for Vehicle Detection

ML-based methods typically rely on manually designed feature extractors and classifiers, which, to some extent, limit the representational capacity of the models. With the rise in deep learning, especially the introduction of convolutional neural networks (CNNs), great progress has been achieved in object detection [73]. Object detection is a pivotal subtask in the field of computer vision, often closely associated with object classification, semantic segmentation, and instance segmentation. Object classification refers to recognizing the different object classes present in an image, while target detection further determines the relative positions of the objects on this basis and locates them by means of bounding boxes. Semantic segmentation is a technique that assigns each pixel to a semantic category label. Instance segmentation, on the other hand, is an extension of semantic segmentation with the goal of distinguishing between different object instances. Figure 1 illustrates the comparison of them. Vehicle detection frameworks under deep learning techniques can be divided into two types: object-detection-based models and segmentation-based models.

3.3.1. Object-Detection-Based Methods

Object detection holds significant potential across diverse applications such as image recognition and video surveillance. In general, object-detection-based models are classified as anchor-based detectors, anchor-free detectors, and end-to-end detectors, as illustrated in Figure 2.



(a) (b)

(c) (d)

Figure 1. Relationship between different vehicle detection algorithms: (**a**) object classification, (**b**) object detection, (**c**) semantic segmentation, (**d**) instance segmentation [16].

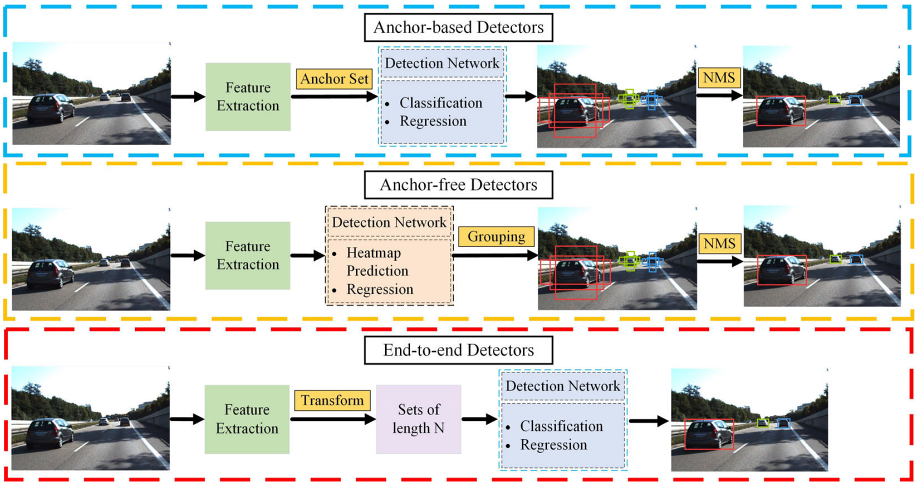


Figure 2. Vehicle detection methods in different detectors [15].

(1) Anchor-Based Detectors

In anchor-based models, predefined bounding boxes are used to detect target objects. Depending on whether region proposals are utilized, anchor-based detectors fall into two types: two-stage and one-stage.

Two-stage: In vehicle detection, vehicle region proposals are first generated followed by classifying and regressing vehicle targets of interest from region proposals. R-CNN series [73–76], SPP-Net [77], R-FCN [78], FPN [79], and Cascade R-CNN [80] are examples of typical two-stage detectors. Faster R-CNN [73] consists of a separate region proposal network and R-CNN [74] to detect objects, considerably lowering the running time consumed by the detection network. Two-stage methods refine anchors multiple times, resulting in more accurate results compared to one-stage methods.

One-stage: The method predicts the center and bounding boxes of vehicles by placing anchors on the feature map. Typical representatives of one-stage detectors include SSD [81], M2Det [82], RetinaNet [83], and part of the YOLO series [84–88]. YOLOv1 [89] is the pioneering work of the YOLO family of algorithms. As an anchor-free model, it laid the foundation for subsequent YOLO algorithms. From YOLOv2 to YOLOv5, all versions use the anchor-based approach and continue introducing new techniques through each iteration, all of which have improved detection performance. YOLOv4 extensively tests and applies some commonly used tricks in deep learning algorithms to achieve the optimal balance between detection speed and accuracy. YOLOv5 continues the style of the YOLO series of algorithms, and has a strong advantage in the deployment of mobile devices. The innovative YOLOv7 introduces efficient layer aggregation networks (ELANs) as a backbone, and re-parameterized convolutions are employed to accelerate the inference speed. Although one-stage algorithms exhibit lower detection accuracy compared to two-stage algorithms, they hold an advantage in terms of detection speed.

(2) Anchor-Free Detectors

The anchor-free model predicts the center point or keypoints of an object directly and clusters them into a single entity to obtain bounding boxes. The keypoint-based approach involves detecting critical features of the target or the interrelations among these features to determine the target's position and shape. These critical features may include the corners and center of the object. Some models such as CornerNet [90], RepPoints [91], CenterNet [92], ExtremeNet [93], and Grid R-CNN [94] are keypoint-based. CornerNet identifies an object's bounding box by detecting a pair of keypoints. CenterNet advances this approach by utilizing a triplet of keypoints instead of a pair. This modification aims to enhance both precision and recall in object detection tasks. The center-based approach determines the target's bounding box by predicting its center point and positional offset with respect to the center point. Some classical center-based models include YOLOv1 [89], FSAF [95], FCOS [96], GA-RPN [97], FoveaBox [98], YOLOX [99], YOLOv8 [100], and YOLOv9 [101]. FCOS considers all locations within the object bounding box as positives, utilizing four distances and a novel center score for object detection. GA-RPN defines the pixels within the central region of the object as positives, predicting object proposal locations, widths, and heights for Faster R-CNN. Anchor-free detectors are usually more computationally efficient compared to anchor-based detectors. YOLOv8 adopts a novel C2F module that enriches the gradient flow and employs a decoupled head for regression. YOLOv9 introduces generalized ELAN based on YOLOv7 and proposes programmable gradient information to accommodate customized network structures. It is expected for YOLOv9 to become the industry standard for anchor-free detectors in the near future.

(3) End-To-End Detectors

Anchor-based methods rely on proposals or anchors, whereas anchor-free methods utilize center points or keypoints. They indirectly predict a set of bounding boxes by regression and classification tasks. The efficacy of their performances is notably shaped by non-maximum suppression procedures aimed at consolidating near-identical forecasts, by the formulation of anchor sets, and by the heuristics governing the allocation of target boxes to anchors. End-to-end detectors analyze an input image to directly determine the location and category of a target without the need for complicated pre-processing or post-processing procedures. Some models such as DeFCN [102], Sparse R-CNN [103], and DETR [104] are end-to-end-based. DeFCN is based on FOCS [96] and introduces a Prediction-aware One-To-One (POTO) label assignment for classification. Sparse R-CNN re-evaluates the design process of RPN and provides a fixed sparse set of learned object proposals (total length of N) to the object recognition head to perform classification including location. DERT is a new style of neural network based on Transformer [105] for end-to-end detection. Unlike traditional convolutional networks, Transformer-based networks use self-attention mechanisms for encoding and decoding, and can model global feature information. The encoder–decoder architecture was initially proposed for machine translation tasks and has since been widely used in various deep learning models [106]. In vehicle detection, the role of the encoder is to encode the features of input images and map them to high-dimensional vector representations. The decoder is responsible for mapping the encoded features to the output space, which includes categories and positions of the vehicles. DETR transforms the target detection task into an unordered ensemble prediction challenge. It feeds the extracted feature sequences into both the encoder and decoder of the Transformer, yielding an unordered set of length N as output. Each element within the set comprises the object’s category and coordinates. Deformable DETR [107], Anchor-DETR [108], and RT-DETR [109] are also some excellent algorithms based on improved DETR. End-to-end detectors can simplify the vehicle detection process and have an auspicious future.

The Microsoft Common Objects in Context (MS COCO) dataset is widely recognized as one of the most authoritative datasets in the field of object detection. It encompasses 80 object categories, with a total of 2.5 million labeled instances across 328 k images. As a benchmark, we have conducted the performance comparison of various deep models on the MS COCO dataset, as illustrated in Table 3.

Table 3. Comparison of detection performances on MS COCO dataset.

Model	**Backbone**	**AP**	**AP_{50}**	**AP_{75}**	**AP_S**	**AP_M**	**AP_L**
Anchor-based two-stage							
Faster RCNN [73]	VGG-16	21.9	42.7	-	-	-	-
R-FCN [78]	ResNet-101	29.9	51.9	-	10.8	32.8	45.0
CoupleNet [110]	ResNet-101	34.4	54.8	37.2	13.4	38.1	50.8
Mask RCNN [76]	ResNeXt-101	39.8	62.3	43.4	22.1	43.2	51.2
DetNet [111]	DetNet-59	40.3	62.1	43.8	23.6	42.6	50.0
Soft-NMS [112]	ResNet-101	40.8	62.4	44.9	23.0	43.4	53.2
G-RMI [113]	-	41.6	61.9	45.4	23.9	43.5	54.9
Cascade R-CNN [80]	Res101-FPN	42.8	62.1	46.3	23.7	45.5	55.2
SNIP [114]	DPN-98	45.7	67.3	51.5	29.3	48.8	57.1
Anchor-based one-stage							
SSD [81]	VGG-16	28.8	48.5	30.3	10.9	31.8	43.5
DSSD [115]	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
M2Det [82]	VGG-16	33.5	52.4	35.6	14.4	37.6	47.6
RefineDet [116]	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4
RetinaNet [83]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2
YOLOv2 [84]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
YOLOv3 [85]	DarkNet-53	33.0	57.9	34.4	18.3	35.4	41.9
YOLOv4 [86]	CSPDarkNet-53	41.2	62.8	44.3	20.4	44.4	56.0
YOLOv5 [87]	CSPDarkNet-53	49.0	67.3	-	-	-	-
YOLOv7 [88]	ELAN	52.9	71.1	57.5	36.9	57.7	68.6
Anchor-free keypoint-based							
CornerNet [90]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
RepPoints [91]	Res101-DCN	45.9	66.1	49.0	26.6	48.6	57.2
CenterNet [92]	Hourglass-104	44.9	62.4	48.1	25.6	47.4	57.4
ExtremeNet [93]	Hourglass-104	40.2	55.5	43.2	20.4	43.2	53.1
Grid R-CNN [94]	ResNeXt-DCN	43.2	63.0	46.6	25.1	46.5	55.2
Anchor-free center-based							
FSAF [95]	ResNeXt-101	42.9	63.8	46.3	26.6	46.2	52.7
FCOS [96]	ResNeXt-101	43.2	62.8	46.6	26.5	46.2	53.3
GA-RPN [97]	ResNet-50	39.8	59.2	43.5	21.8	42.6	50.7
FoveaBox [98]	ResNeXt-101	42.1	61.9	45.2	24.9	46.8	55.6
YOLOX [99]	CSPDarkNet-53	50.0	68.5	54.5	29.8	54.5	64.4
YOLOv6 [117]	EfficientRep	52.8	70.3	57.7	34.4	58.1	70.1
YOLOv8 [100]	DarkNet-53	52.9	69.8	57.5	35.3	58.3	69.8
YOLOv9 [101]	GELAN	53.0	70.2	57.8	36.2	58.5	69.3
End-to-end-based							
DeFCN [102]	-	38.6	57.6	41.3	-	-	-
Sparse R-CNN [103]	ResNet-50	42.8	61.2	45.7	26.7	44.6	57.6
DETR [104]	ResNet-50	43.3	63.1	45.9	22.5	47.3	61.1
Deformable DETR [107]	ResNet-50	46.2	65.2	50.0	28.8	49.2	61.7
Anchor-DETR [108]	ResNet-101	45.1	65.7	48.8	25.8	49.4	61.6
Efficient-DETR [118]	ResNet-101	45.7	64.1	49.5	28.8	49.1	60.2
RT-DETR [109]	ResNet-101	54.3	72.7	58.6	36.0	58.8	72.1

3.3.2. Segmentation-Based Methods

Semantic segmentation is generally considered to be more precise and accurate than target-level vehicle detection methods. It attempts to assign a label or category to each pixel in an image and has a greater ability to identify a collection of pixels from different categories and show the position and contour information of vehicles [16], making it important for autonomous vehicles' environmental perception. Semantic segmentation can be categorized into fully supervised algorithms and weakly supervised algorithms. Weakly supervised learning is a method that utilizes partial, inaccurate, or noisily labeled data for model training [119]. Although this method requires less annotated data and has relatively lower costs, the presence of noise or even mislabeling can impact the accuracy of detection. In autonomous driving, such inaccurate detection significantly affects its performance and safety in real-world scenarios. The fully supervised algorithms are almost always used in most scenarios due to the low security of weakly supervised algorithms.

Traditional vehicle semantic segmentation methods rely on region classification. The principle of these methods is similar to the two-stage detectors, in which vehicle candidate regions are first extracted by a region proposal network, and then a trained classifier assigns labels to each pixel within the candidate regions. DeepMask [120] is a CNN-based model that outputs a class-agnostic segmentation mask, followed by the likelihood score that the patch lies in the center of the vehicles. SharpMask [121] employs a top–down refinement method to generate high-fidelity masks to augment the feed-forward network. MultipathNet [122] makes three improvements on the Fast R-CNN [75] and incorporates DeepMask proposals for detection. Mask R-CNN [76] extends Faster R-CNN [73] to detect different scales and overlapping vehicles in an image with anchor boxes. However, these methods rely on generating candidate regions, which limits the ability to deploy vehicle detection in real time.

The use of pixel-level classification methods helps to improve the issue. A full convolution network (FCN) is a classical algorithm that was first proposed in 2015 [123]. The model replaces the fully connected layers with convolutional layers and uses skip architecture to fuse feature information. SegNet [124] builds on this with an encoder–decoder network. The decoder network maps the low-resolution representation of the encoder to full input resolution feature maps and performs non-linear upsampling in the max-pooling step of the corresponding encoder. Google Labs improved FCN with four separate proposed algorithms. DeepLabv1 [125] introduces the CRF model and atrous convolution to extract image information. DeepLabv2 [126] is built on DeepLabv1 with the backbone of Resnet [127] and an atrous spatial pyramid pooling (ASPP) module. DeepLabv3 [128] combines ideas from DeepLabv1 and DeepLabv2 to segment objects at multiple scales. DeepLab3+ [129] adopts Xception [130] as the backbone and introduces depthwise separable convolution to replace some of the convolutional and pooling layers. DeepLab series can effectively increase the filter's receptive field. Nevertheless, the Deeplab series requires a high computational cost to deploy in real scenarios.

To further improve the speed and accuracy of vehicle semantic segmentation, some researchers have proposed feature fusion models. These models use multiscale convolution to better access the deep contextual information of an image through a cross-layer structure and reduce computational consumption to some extent. RefineNet [131] efficiently fuses high-level features with finer-grained low-level features to prevent image resolution degradation. PSPNet [132] proposes a pyramid pooling module that exploits global context information by the context aggregation of different regions. ICNet [133] incorporates multi-resolution branches by proper label guidance and introduces the cascade feature fusion unit for fast and high-quality segmentation. In addition, some scholars have attempted to use a generative adversarial network (GAN) for vehicle semantic segmentation [134,135]. However, these methods are unstable during training and fine-tuning, and they are prone to cause the model to collapse and fall into a local optimum.

The Transformer-based architecture is also being applied to semantic vehicle detection as a powerful feature extractor. SERT [136] utilizes ViT [137] as its backbone while integrating multiple CNN decoders to enlarge feature resolution. SegFormer [138] designs a novel hierarchical structured Transformer block to acquire multiscale features and uses MLPs to simply aggregate the features from different layers for decoding. SeaFormer [139] employs a squeeze axial and detail-enhanced attention module to achieve the optimal trade-off between segmentation accuracy and latency on ARM-based mobile devices.

Generally, semantic segmentation-based vehicle detection methods require high computational complexity, which can lead to slower inference speed than that of other vehicle detection algorithms. Therefore, the design and deployment of lightweight models is where the need for the future lies, which requires both speed and accuracy. Recently, there has been a lot of research into lightweight vehicle semantic segmentation models. ESPNet [140] employs efficient convolutional modules, which are 22 times faster and 180 times smaller than existing state-of-the-art vehicle semantic segmentation networks. DFANet [141] begins with a solitary lightweight backbone and progressively consolidates discriminative features through a cascade of sub-networks and sub-stages. Experiments show that the model attained 1.7 GFLOPs at a speed of 160 FPS on one NVIDIA Titan X GPU and a 0.703 mIoU (mean IoU) on the Cityscapes dataset. LEDNet [142] utilizes an asymmetric encoder–decoder architecture and achieved 71 FPS and a 0.706 mIoU on the Cityscapes dataset with NVIDIA Titan X. Lightweight deployment capabilities will be a key technology for researchers to consider in the field of autonomous driving.

4. Vehicle Detection Algorithms Based on Radar and LiDAR

Vehicle detection using radar and LiDAR is a key component of modern advanced driver assistance systems (ADASs) and self-driving vehicles. LiDAR and radar differ from the visible light images captured by cameras that they acquire information about the distance and shape of the target. Both are now widely used in autonomous driving systems for intelligent vehicles.

4.1. Millimeter-Wave Radar-Based Methods for Vehicle Detection

The millimeter-wave radar sensor operates by utilizing millimeter-wave frequencies in wireless radio wave detection. Its principle lies in the emission and reception of millimeter-wave signals, extracting parameters such as distance, velocity, direction, size, and trajectory of objects through techniques like time-of-flight measurements and Doppler effects. Compared to camera-based and LiDAR-based sensors, millimeter-wave radar is more resilient and allows vehicle detection in harsher weather conditions. In addition, these radars can also acquire accurate vehicle depth information, thus facilitating the perception of autonomous driving. The radar-based vehicle detection process is shown in Figure 3. Depending on the type of output signal, millimeter-wave radar is generally categorized into target-level radar and image-level radar. Figure 4 presents an example of this.

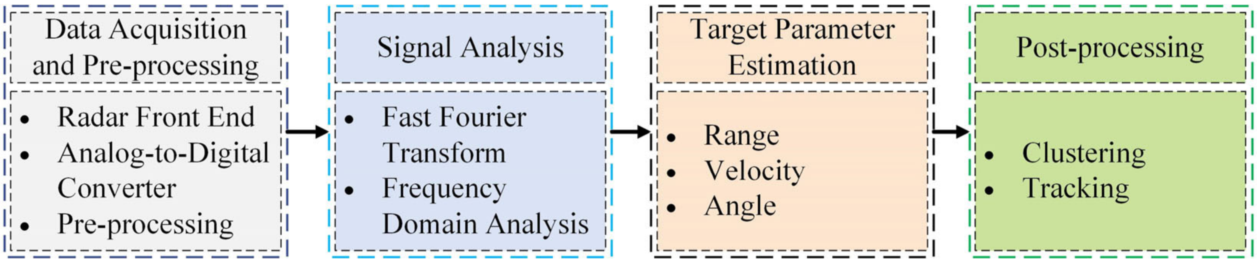


Figure 3. The radar-based vehicle detection process.

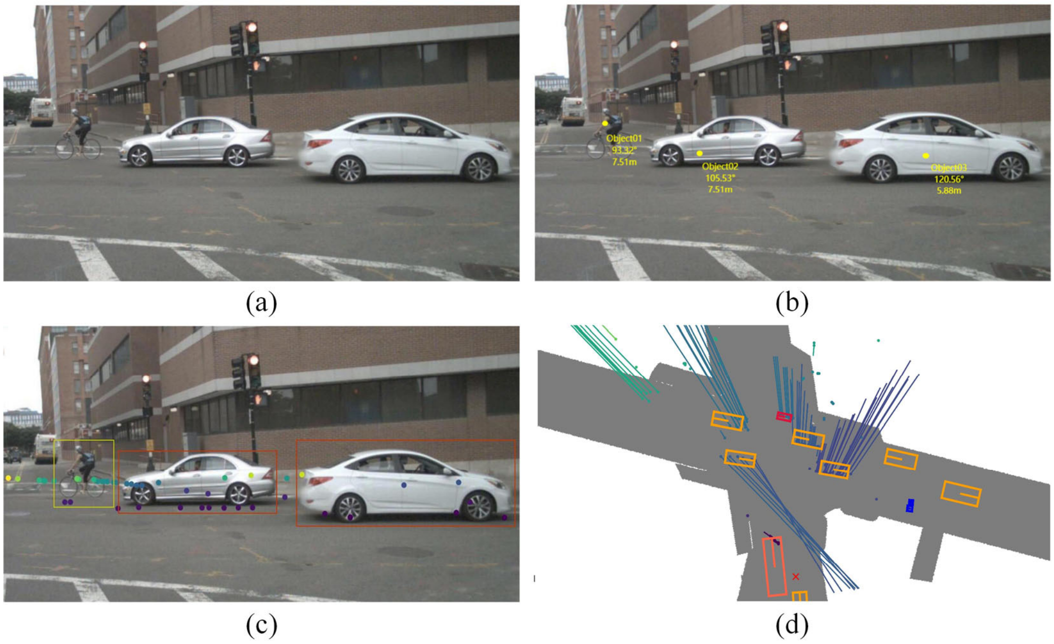


Figure 4. An example of target-level radar and image-level radar for vehicle detection: (**a**) original image, (**b**) target detection results (target-level), (**c**) projection map (image-level), (**d**) point cloud map (image-level) [27].

4.1.1. Target-Level Radar

The target-level radar is oriented to the output target and can transform received echo signals directly into target information, such as distance, speed, and angle of the detected vehicles. Radar detection results can be classified into three categories: moving targets, stationary targets, and false targets. Dynamic vehicles, bicyclists, and pedestrians are the most common moving targets, while stationary targets mainly include parked automobiles, streetlamps, roadside trees, road guardrails, and curbs. False targets are invalid owing to interference or background noise. The radar itself does not have the discriminatory ability to classify detected targets. Therefore, there is a need to eliminate the interference of stationary and false targets on vehicle detection as much as possible. According to research, false target signals only remain for a short time when detected and can be eliminated by the Kalman filter [143], multiple-hypothesis target [144], and iterative adaptive approach [145]. In addition, some scholars have found that the radar cross-section (RCS) and signal-to-noise ratio (SNR) of stationary vehicles are much smaller than those of moving vehicles [146]. RCS refers to the extent to which an object is detected by radar, while SNR is the ratio of the desired signal power to the noise power. According to the characteristics of the motion state, a specific threshold value is set for the RCS and SNR of the radar, which can separate the moving vehicle targets from the stationary ones. Recognition and classification can be achieved by an ML-based classifier such as SVM and deep belief network (DBN).

The target-level radar provides information regarding the vehicle's position and motion status, which is crucial for the environmental perception of autonomous driving. However, it lacks the ability to depict the vehicle's contour and type. Furthermore, the detection accuracy of radar is not satisfactory when the vehicle is moving slowly or stationary. Hence, depending solely on target-level radar as a vehicle sensor is inappropriate.

4.1.2. Image-Level Radar

Image-level radar is increasingly being applied in autonomous driving due to the need for high-resolution imaging. It not only provides information on the speed and motion status of the target but also generates an imaging map of the radar signals. In general, radar image formats can be categorized into four types: projection maps, range–Doppler–azimuth maps, point cloud maps, and SAR maps. By projecting the reflection intensity of a radar detection target onto the image, the reflection intensity map can be produced [12].

The generation of range–Doppler–azimuth maps requires the use of Fourier transform and time–frequency domain analysis techniques combined with distance measurements, Doppler shifts, and azimuth estimation algorithms. Both projection maps and range–Doppler–azimuth maps are 2D imaging maps that can be represented with deep learning algorithms, such as CNN [147], FCN [148], and LSTM [149].

Point cloud maps represent spatial data composed of a collection of points in the 3D coordinate system. Machine learning and deep learning algorithms are often used to model point cloud information for vehicle classification and detection. It has been reported that the radar point cloud can be clustered together using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, allowing the obtained clustered vectors to describe the vehicle features [150]. The vehicle targets were then classified by SVM. However, the detection performance of millimeter-wave radar is limited by its low resolution. To improve this problem, a GAN architecture has been designed for recovering high-frequency shapes from original low-resolution millimeter-wave heatmaps [151]. According to this research, a CNN-based point cloud segmentation algorithm is utilized to detect vehicle targets, which can accurately reconstruct cars in real scenes with low visibility. Li et al. [152] developed a method to enhance radar perception with temporal information. They used the temporal relational layers of successive ego-centric bird-eye-view radar image frames for radar object recognition. Synthetic aperture radar (SAR) is a technique that produces fine-resolution coherent images from a resolution-limited radar system. SAR obtains image data by processing the reflected echoes, which can be used for vehicle detection with deep learning algorithms, such as CNN [153] and YOLO [154].

Millimeter-wave radar has been widely used in the field of autonomous driving for its robustness and anti-interference. With the continuous development of radar technology, how to further improve the resolution of radar will become a key research direction in the future.

4.2. LiDAR-Based Methods for Vehicle Detection

LiDAR is an optical technology that senses distance by measuring the time lapse between an emitted laser pulse and the detection of a reflected light pulse. In the process, LiDAR feeds back the geometric information about the object, such as size and 3D coordinates. The point cloud, composed of a collection of 3D points, can express the sensory information of the transportation environment. Compared with cameras and millimeter-wave radar, LiDAR has higher detection accuracy and can more accurately acquire information about the surroundings of the vehicle [4]. Moreover, it is insensitive to changes in light intensity, making it more applicable to vehicle detection in autonomous driving. LiDAR-based vehicle detection methods can be divided into two categories: traditional and deep learning.

4.2.1. Traditional-Based Methods

Traditional methods rely on the construction of feature engineering and data processing. The traditional LiDAR vehicle detection algorithm is shown in Figure 5. For vehicle detection in a single frame point cloud, the raw image needs to be pre-processed, downsampled, ground-segmented, and clustered for feature extraction, respectively. Due to the sparse nature of LiDAR point cloud data, it is often necessary to convert 3D LiDAR data to 2D or 2.5D data to improve computational efficiency. These conversion techniques include a graph method [155], range image [156], and occupancy map [157]. Some irrelevant point cloud information can be eliminated to optimize the traffic environment sensing system. Studies have shown that the point cloud information of road pavement is significantly different from vehicles and other obstacles. It has been reported that road point clouds can be eliminated by setting specific feature thresholds to reduce the amount of computation and improve detection in real time [158,159]. However, these methods fail to perform well when dealing with special road sections, such as potholes or steep slopes. In this regard, scholars have proposed some fitting algorithms to solve the problem by partitioning the uneven pavement into a combination of several smooth planes, such as Markov random fields (MRFs) [160], random sampling consensus (RANSAC) [161], and Gaussian process regression (GPR) [162]. Next, the point cloud information with the same features is grouped with a clustering algorithm to highlight the attributes of the target. DBSCAN and K-means are classical clustering algorithms that divide points into clusters by density and distance, respectively. The clustered results are generally fed into the machine learning-based classifier, and then the vehicle detection is performed.

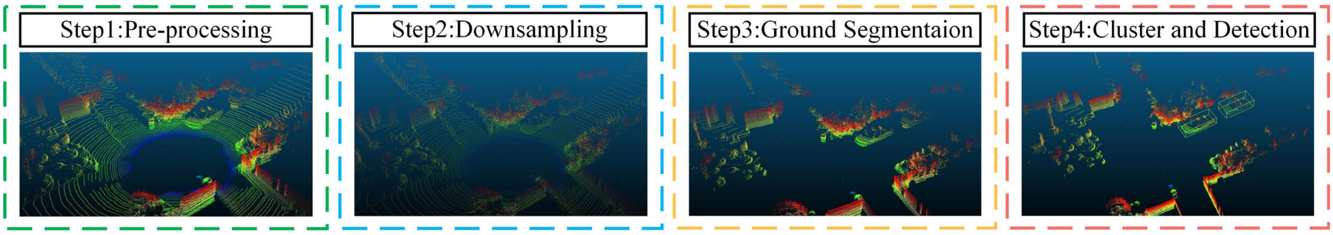


Figure 5. Traditional-based vehicle detection process using LiDAR point cloud.

Traditional methods are simple to implement but are overly dependent on a priori knowledge in processing point cloud information. In addition, the process of vehicle detection requires multiple steps and cannot meet the real-time requirements in real scenarios.

4.2.2. Deep Learning-Based Methods

With the rapid development of computer vision, LiDAR deep learning-based algorithms for vehicle detection demonstrate superiority in detection speed and accuracy. Most of these methods adopt an end-to-end approach, which facilitates the improvement in real-time vehicle detection. In contrast to the construction of feature engineering, deep learning algorithms can automatically learn complex point cloud information and extract high-level representation from deep networks. Based on the principle of the algorithms, LiDAR deep learning-based methods can be classified as point-based, projection-based, and voxel-based. The visual representations of these methods are shown in Figure 6.

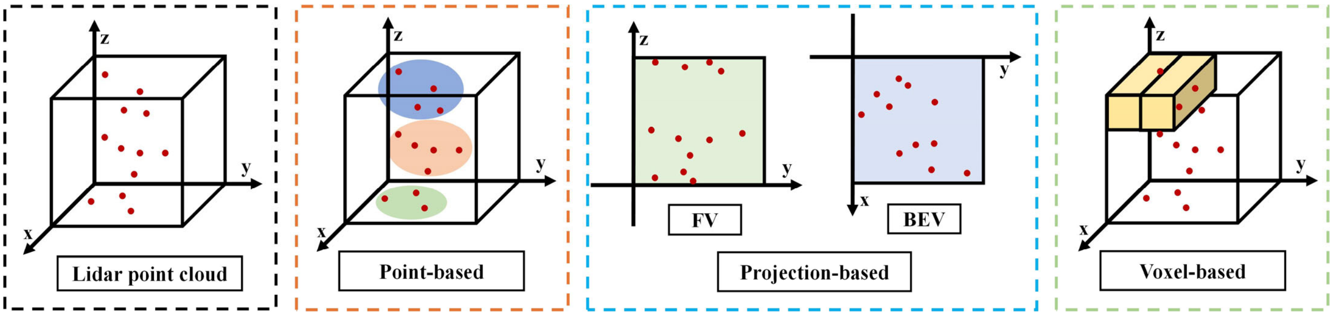


Figure 6. Spatial data representation of LiDAR point cloud.

(1) Point-Based Methods

Point-based methods perform 3D detection techniques of raw point cloud data to obtain vehicle targets. The primary characteristic of point cloud data is their insensitivity to the arrangement order of points. This implies that we can process point cloud data in any order. Vote3deep [163] employs a feature-centric voting scheme for constructing convolutional layers, which leverage the sparsity inherent in point cloud data. PointNet [164] is a classical algorithm presented in 2017. This method designs a novel neural network that processes point cloud information directly while respecting the permutation invariance of the input points. However, the design of PointNet fails to capture the local structure created by metric space points, restricting its ability to gather fine-grained information. PointNet++ [165] improves on PointNet to fully extract global and local vehicle features. It utilizes the PointNet network recursively on the set of input points through a hierarchical approach and adaptively combines features from multiple scales at the learning layer. PointRCNN [166] references the feature point extraction method of PointNet++ and utilizes a two-stage framework for detection and segmentation. Stage one generates 3D proposals through a bottom–up approach, and in the second stage, the convergence points of each proposal are converted to canonical coordinates. These models have excellent detection accuracy but are time-consuming. To achieve a reasonable balance between accuracy and efficiency, scholars propose a single-stage anchor-free-based detection method named 3DSSD [167]. This method adopts a fusion sampling strategy in downsampling to enable detection on fewer representative points, which yields the inference speed of 25+ FPS.

Point-based methods maximize the use of raw information from point clouds in space, which is effective for vehicle detection. However, the target is usually represented by only some of the points, resulting in a loss of spatial information between neighboring localized ones.

(2) Projection-Based Methods

Projection-based methods project a 3D point cloud onto a 2D plane to create a front view (FV) or bird's eye view (BEV), which reduces the complexity of modeling point cloud data and requires fewer computational resources. DeepthCN [168] proposes a vehicle detection system based on hypothesis generation and hypothesis verification. The data input to the system is first subjected to ground segmentation and point cloud segmentation, and then it is projected onto a Dense-depth Map and detected by ConvNet. RT3D [169] projects the 3D point cloud onto the BEV and applies R-FCN for feature map extraction. BirdNet [170] projects laser information into a novel cell encoding for BEV, and then employs a CNN-based network to estimate the location and heading of the object, which is mapped through post-processing to 3D orientation detection. BirdNet+ [171] discards the post-processing step of BirdNet and achieves state-of-the-art results through an end-to-end detection framework for direct inference of oriented 3D boxes in BEV images.

LiDAR is expensive but is widely used as it can acquire 3D information of the environment with high accuracy. In projection-based methods, the feature map is eventually projected as a 2D map, which is similar to 2D target detection. Consequently, some scholars have proposed pseudo-LiDAR representations, which essentially mimic LiDAR signals. Wang et al. [172] converted depth maps to pseudo-LiDAR maps and improved the detection accuracy of within 30 m to 74% in the KITTI dataset. However, pseudo-LiDAR-based methods generally require the depth estimation of the 2D map before performing 3D object detection, resulting in two separate steps. Qian et al. [173] introduced a framework based on the differentiable change in the representation module that allows end-to-end training of the entire pseudo-LiDAR pipeline. Pseudo-L [174] presents three novel methods for virtual view generation, including image-level generation methods, feature-level generation, and a feature clone. Furthermore, a disparity-wise dynamic convolution is proposed, which alleviates the feature degradation caused by depth estimation errors.

The success of projection-based methods is essentially the maturation of 2D detection algorithms. However, dimensionality reduction in the 3D point cloud information will inevitably result in the loss of spatial depth information, reducing the detection accuracy.

(3) Voxel-Based Methods

The point cloud data collected by LiDAR are typically dense, yet a majority of points are concentrated within specific spatial regions. This leads to the sparsity of data, making direct processing of the entire point cloud complex and inefficient. In order to better describe the distribution of the point clouds in three dimensions, the challenge is solved by dividing the point clouds into regular grids of voxels of a specific size. VoxelNet [175] is a representative algorithm that combines PointNet and CNN to present an end-to-end trainable architecture. The model represents voxel points through a voxel feature encoding

(VFE) layer, which enables efficient parallel processing of the voxel grid. Voxel RCNN [176] designs a voxel ROI pooling to further refine the features of the BEV region proposal network. PV-RCNN [177] summarizes the 3D scene with 3D voxel CNNs into a small collection of keypoints and then employs ROI grid points to extract richer contextual information. PV-RCNN++ [178] proposes two improvements based on PV-RCNN, sectorized proposal-centric sampling and vector pool aggregation, which can generate more efficient keypoints and better aggregate local point features, respectively. VoxelNeXt [179] directly uses a sparse convolutional network to detect and track 3D objects entirely through voxel features without switching to a dense detection header and NMS post-processing, resulting in a better trade-off between speed and accuracy. Other voxel-based methods include MA-MFFC [180], PDV [181], and SAT-GCN [182].

The voxel-based approach converts the point cloud data into a 3D voxel grid, which offers the advantages of high processing efficiency and good spatial information retention. Nevertheless, some information might be lost using this method. Table 4 summarizes the performance of different LiDAR-based deep learning models.

Table 4. The performance of different LiDAR-based deep learning models on KITTI dataset.

Model	**Car AP (IoU = 0.7)**			**FPS**	**Year**	**Reference**
	Easy	**Moderate**	**Hard**			
Point-based						
Vote3deep	76.79	68.24	63.23	0.9	2017	[163]
PointRCNN	85.95	75.76	68.32	3.8	2019	[166]
STD	86.61	77.63	76.06	-	2019	[183]
Part-A2	85.94	77.95	72.00	-	2020	[184]
3DSSD	88.36	79.57	74.55	26.3	2020	[167]
SASSD	88.75	79.79	74.16	24.9	2020	[185]
Pyramid RCNN	87.03	80.30	76.48	8.9	2021	[186]
ST3D	-	-	74.61	-	2021	[187]
SASA	88.76	82.16	77.16	27.8	2022	[188]
PointDistiller	88.10	76.90	73.80	-	2023	[189]
DCGNN	89.65	79.80	74.52	9.0	2023	[190]
Projection-based						
DeepthCN	37.59	23.21	18.01	-	2017	[168]
RT3D	72.85	61.64	64.38	11.2	2018	[169]
BirdNet	88.92	67.56	68.59	9.1	2018	[170]
PIXOR	81.70	77.05	72.95	10.8	2018	[191]
Complex-YOLO	67.72	64.00	63.01	59.4	2018	[192]
BirdNet+	70.14	51.85	50.03	10.0	2020	[171]
E2E-PL	79.60	58.80	52.10	-	2020	[173]
Pseudo-L	23.74	17.74	15.14	-	2022	[174]
Ri-Fusion	85.62	75.35	68.31	26.0	2023	[193]
Voxel-based						
3DFCN	84.20	75.30	68.00	-	2017	[194]
VoxelNet	77.47	65.11	57.73	30.3	2018	[175]
Second	83.13	73.66	66.20	20.0	2018	[195]
PV-RCNN	90.25	81.43	76.82	12.5	2020	[177]
HVNet	87.21	77.58	71.79	31.3	2020	[196]
TANet	83.81	75.38	67.66	28.8	2020	[197]
Voxel RCNN	90.09	81.62	77.06	25.0	2021	[176]
MA-MFFC	92.60	84.98	83.21	7.1	2022	[180]
PDV	90.43	81.86	77.49	7.4	2022	[181]
SAT-GCN	79.46	86.55	78.12	8.2	2023	[182]
BSAODet	88.89	81.74	77.24	-	2023	[198]

4.2.3. Point Cloud Segmentation-Based Methods

In the context of road scenes, semantic segmentation labels each point to a predefined category, such as pedestrians, vehicles, trees, etc. Vehicle detection algorithms based on point cloud segmentation can be divided into two categories: traditional methods and deep learning-based methods. Traditional segmentation approaches rely on prior knowledge and feature engineering, like region growing [199], clustering [200], and model fitting [201]. The design of feature engineering entails substantial time investment, while the determination of segmentation boundaries through thresholding is prone to errors. Moreover, achieving pixel-level segmentation poses significant challenges. These factors collectively contribute to the limitations of traditional methods in point-cloud-based vehicle detection. Presently, deep learning-based point cloud segmentation has achieved remarkable performance in accuracy and speed. These approaches are classified as point-based, projection-based, and voxel-based.

(1) Point-Based Methods

These methods directly process 3D point cloud information for vehicle detection. PointNet [164] and PointNet++ [165] are the most representative models in this domain. They leverage shared Multi-Layer Perceptrons (shared MLPs) and pooling to integrate global and local features, while employing MLPs to assign semantic labels to individual points. However, these point sampling methods exhibit poor scalability with respect to the scale of point clouds. Additionally, employing max-pooling to group local points may result in robustness in complex scenes. Some scholars have attempted to address these issues. RandLA-Net [202] integrates random point sampling with a local feature aggregation module to increase the receptive field of each 3D point, rendering it suitable for the semantic segmentation of large-scale point clouds on a per-point basis. S3Net [203] utilizes sparse mechanisms to construct modules, thereby providing rich contextual information for feature maps. Direct processing of point clouds can also be achieved through point convolution. KPConv [204] identifies a set of pivotal points in the spatial domain, employing kernel functions to compute the weighting coefficients for each point, thereby effecting a transformation of the features. Landrieu et al. [205] proposes a framework for a large-scale point cloud based on the concept of a superpoint graph (SPG). The SPG facilitates the provision of compact yet rich contextual information, which can increase the performance of point cloud segmentation.

(2) Projection-Based Methods

The sparsity and lack of structure in point clouds pose challenges for feature extraction using CNN in 3D space. Projection-based point cloud segmentation converts 3D point clouds into 2D BEV maps, FV maps, and RV maps and then uses CNN for feature extraction, followed by reconstructing the original 3D scenes. SqueezeSeg [206] employs spherical projection to transform point clouds into front-view representations. Then, SqueeNet [207] is used to output a point-wise label map, which is refined by a conditional random field. SqueezeSegv2 [208] introduces a context aggregation module to enhance SqueezeNet and proposes an adaptive training approach to reduce the distribution gap between simulated data and real data. SqueezeSegv3 [209] introduces spatially adaptive convolution, which employs different filters for various positions in the image. In order to minimize the loss of information due to dimensional changes during projection, some scholars have proposed multi-view projection methods. GVCNN [210] is a typical representative algorithm. It groups feature subgraphs from different viewpoints based on discriminative weight and then aggregates descriptions of each group through pooling.

(3) Voxel-Based Methods

The sparsity and lack of structure of point clouds greatly affects their representation capability. Voxel-based point cloud segmentation transforms point clouds into structured voxels and employs 3D networks for semantic segmentation. In this process, the depth information is fully utilized at the expense of resolution. VoxNet [211] is a pioneer in this approach, utilizing 3D CNN to process the voxels of the occupied girds. Subsequently, a number of voxel-based algorithms have been applied to vehicle detection, such as SegCloud [212], Kd-Net [213], and SPVNAS [214]. These models require a lot of computational resources and memory space, which is detrimental to real-time 3D segmentation. PointGrid [215] is a hybrid model that integrates a point and grid, using simple points to quantify local features for each grid unit. Further, considering the geometric spatial properties of 3D point clouds, some scholars have attempted to optimize the network structure using octrees, such as OctNet [216] and O-CNN [217]. Recently, Hou et al. [218] proposed applying knowledge distillation to the semantic segmentation of LiDAR for model compression. The method suggests inter-point and inter-pixel affinity distillation and uses a difficulty-aware sampling strategy for difficult hypervoxels.

Projection-based methods and voxel-based methods lead to the loss of point cloud information. In contrast, point-based methods effectively retain the original point cloud data but incur higher computational costs. In recent years, point cloud segmentation methods have emerged as a significant research focus for LiDAR-based vehicle detection. Some of these algorithms are not specifically designed for vehicle detection but serve as general network architectures that can be adapted to this domain through retraining. Overall, the deep learning-based and the point cloud segmentation-based approach to achieve vehicle detection can be summarized as the process shown in Figure 7.

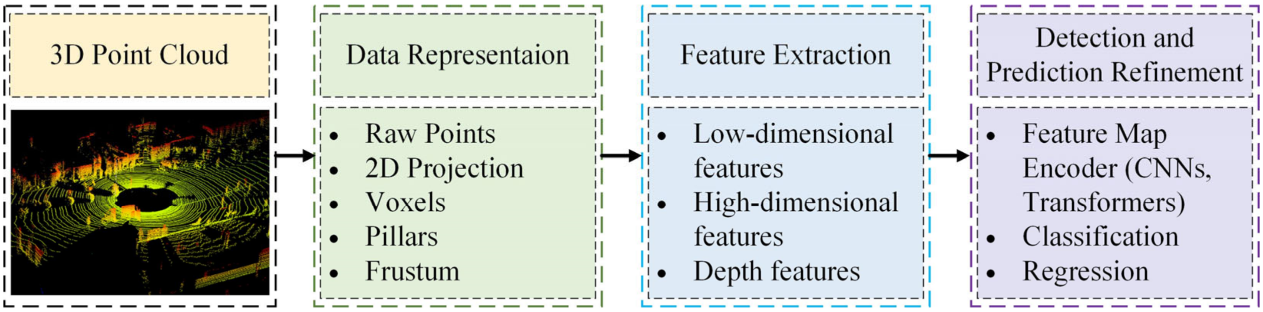


Figure 7. Pipeline for vehicle detection based on LiDAR deep models [15].

5. Vehicle Detection Algorithms Based on Multi-Sensor Fusion

Sections 3 and 4 summarize the vehicle detection algorithms under three typical sensors: a camera, millimeter-wave radar, and LiDAR. It can be noted that each sensor has its advantages and limitations related to its functional operation. A number of metrics for these sensors are summarized in detail in Table 5. In autonomous driving, environmental perception presents a multifaceted challenge. The integration of multiple sensors facilitates synergistic advantages, leading to augmented information acquisition. In general, multi-sensor fusion can be categorized into stereo vision-based, fusion of millimeter-wave radar and vision-based, fusion of LiDAR and vision-based, and multi-sensor-based.

Table 5. Comparative analysis of different sensors. Numbers "1"–"5" denote the level from extremely low to low, medium, high, and extremely high, respectively.

Sensor	Camera	Radar	LiDAR
Silhouette Representation	5	2	3
Color Perception	5	1	1
Velocity Measurement	2	5	2
Angle Resolution	5	3	4
Range Resolution	2	4	5
Object Detection	5	3	4
Object Classification	5	1	3
Field of View	3	4	4

Table 5. *Cont.*

Sensor	Camera	Radar	LiDAR
Adaptability to Complex Weather	2	5	2
Sensor Size	2	2	4
Cost	3	1	5

5.1. Stereo Vision-Based Methods for Vehicle Detection

Stereo vision is a technique that utilizes two or more cameras to simultaneously capture images of a scene from different perspectives in order to obtain depth information and position information of a target in space. Based on the idea of estimating a target's parameters through the changes in the corresponding points in the disparity map, this technique draws inspiration from the search mechanism of the human eyes. According to the principle of the algorithms, stereo vision techniques can be classified into appearance-based methods and motion-based methods, shown as Figure 8.

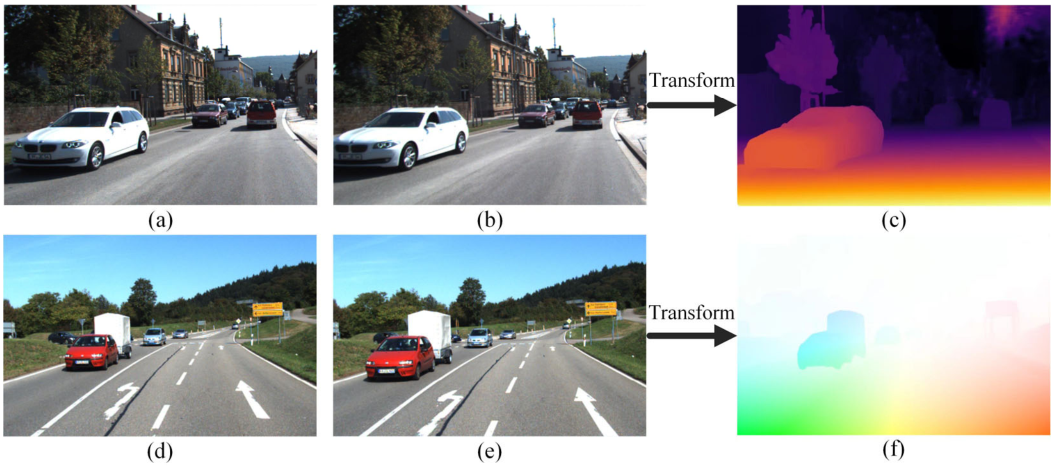


Figure 8. An example of appearance-based methods and motion-based methods for vehicle detection: (**a**) left-eye view, (**b**) right-eye view, (**c**) disparity map (appearance-based); (**d**) a frame from the video stream, (**e**) the subsequent frame from the video stream, (**f**) the optical flow map (motion-based) [15].

(1) Appearance-Based Methods

These methods rely on the extraction of salient vehicle appearance features to achieve vehicle detection. The U-V disparity is a common method used in vehicle detection [219]. Xie et al. [220] proposed a stereo vision segmentation algorithm based on a cascaded framework. For a corrected binocular image pair disparity map, a probabilistic approach is used to compute the U-V disparity, and outliers are removed using RANSAC to obtain the road region. Ma et al. [221] used a nonparametric and refined U-V disparity mapping method to obtain the road ROI, and then utilized an adjacent disparity similarity algorithm to complement and extract the target region for vehicle detection. Observing that the depth information of vehicles is constantly changing in stereo vision, some scholars used depth information clustering to detect vehicles [222]. In recent years, researchers have reported methods for vehicle environmental perception using machine learning and deep learning, which have achieved satisfactory detection results [223,224].

(2) Motion-Based Methods

This type of algorithm relies heavily on the optical flow information. Optical flow information refers to the displacement of pixel points between consecutive image frames due to the motion of an object. Kale et al. [225] make use of optical flow in conjunction with motion vector estimation for object detection and tracking in a sequence of frames. Sengar et al. [226] utilized a Gaussian filter to remove noise from each frame and detected moving targets by calculating the optical flow between three consecutive frames. Some scholars have proposed the fusion of vehicle depth information with optical flow information through occupancy grids to strengthen representation capability and further enhance detection efficiency. Chen et al. [227] optimized the classical optical flow algorithm at a single resolution on a regular grid. Yin et al. [228] presented GeoNet, a joint unsupervised learning framework. This method combines depth, optical flow, and self-motion estimation for image reconstruction loss, and inference for static and dynamic scene parts.

5.2. Fusion of Radar and Vision-Based Methods for Vehicle Detection

Millimeter-wave radar exhibits strong adaptability to complex environments and offers motion and depth information about vehicles. With their high-resolution imaging capabilities, cameras excel in detection and classification tasks. The integration of these two sensors is a typical configuration in many mature autonomous driving systems [14]. Figure 9 illustrates the three fusion levels that are often present for radar and vision-based methods, including the data level, decision level, and feature level.

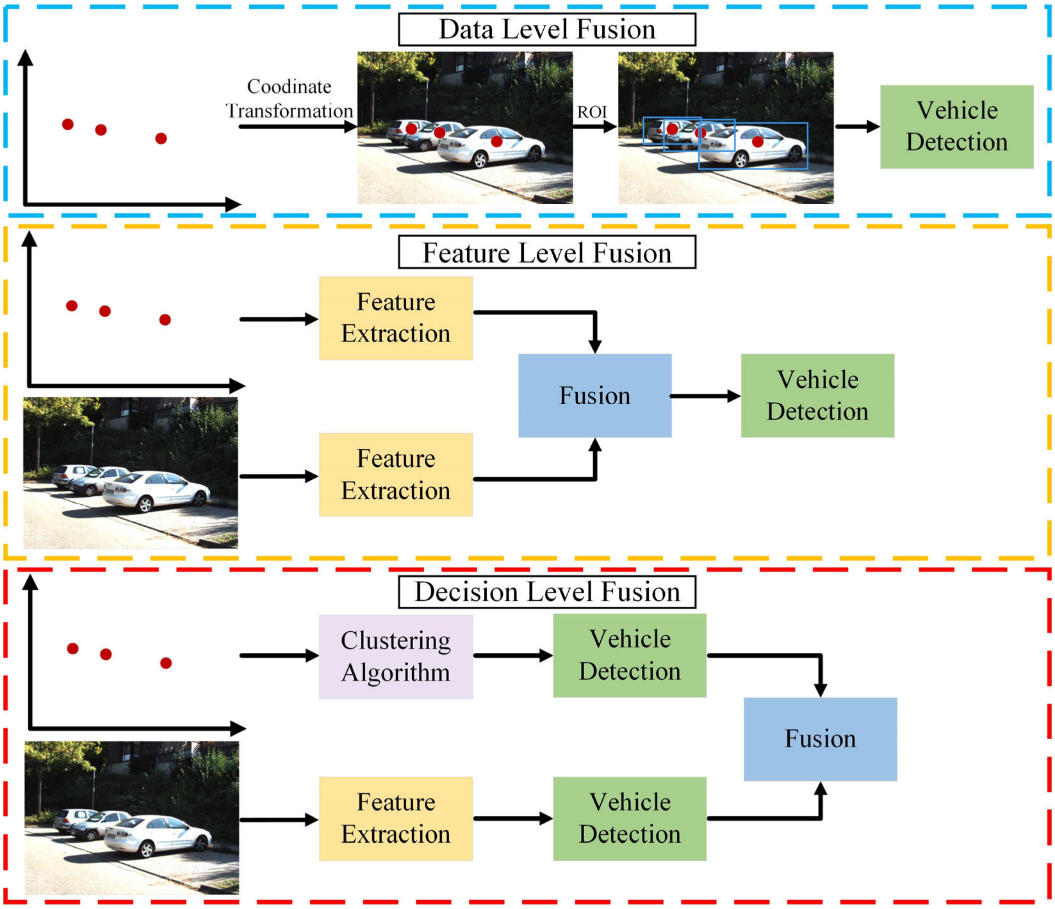


Figure 9. Different methods of radar–camera fusion for vehicle detection [15].

(1) Data-Level Fusion

Data-level fusion is a well-established method for vehicle detection by fusing camera data and radar data. Although this scheme is not currently a mainstream method, its fusion idea is worthwhile. Specifically, data-level fusion first generates ROI through radar, then visual images analyze these regions, and finally vehicle targets are obtained through detectors. Wang et al. [229] employed a fusion strategy of visual attention mechanisms to detect vehicles via an adaptive thresholding algorithm. Wang et al. [230] optimized the fusion approach to achieve the balance between vehicle detection accuracy and speed.

Craft [231] focuses on the spatial properties of the camera and radar, adaptively fusing spatial contextual information between the two.

(2) Feature-Level Fusion

Feature-level fusion is a processing approach for features extracted from raw data. This type of fusion is applicable to a collection of features extracted from multiple sensors or data sources. For the feature extraction of radar information, radar points can be converted into image format. Then, for the feature maps obtained from each sensor, deep models (e.g., CNNs and Transformers) can be used to achieve feature fusion. Lekic et al. [232] introduced a conditional multi-generator generative adversarial network. The model can qualitatively and quantitatively convert environmental features detected by radar sensors into visually appealing images. Chang et al. [233] presented a novel spatial attention fusion method for vehicle detection. Starting with the sensor features, the method fuses the vision features by applying an attention weight matrix. Zhou et al. [234] contributed to multimodal fusion 3D object detection by narrowing the view disparity in different sensor features.

(3) Decision-Level Fusion

Decision-level fusion is the highest level of image fusion in which the independent detection results from each of the camera and radar sensors are integrated. The advantage of radar lies in its ability to accurately measure the longitudinal distance of a target, while the camera provides a broader field of view. Combining radar and a camera can fully utilize these two types of information, thereby improving the accuracy and reliability of target detection and tracking. Zhong et al. [235] reported a Kalman-filter-based camera–radar fusion system, which strikes a balance between performance and energy efficiency and demonstrates the competitiveness of the software–hardware ecosystem. Bai et al. [236] correlated the respective detections of the radar and camera in the image plane to generate a random finite set with an object type. The model is then refined using a Gaussian mixture probability hypothesis density algorithm. Sengupta et al. [237] combined the Hungarian algorithm and triple Kalman filtering for object tracking, significantly reducing the false-negative rate and providing a promising direction for autonomous perception. Table 6 summarizes the performance of some of the latest different radar–camera-based deep learning models.

Table 6. Performance of some of the latest radar–camera-based models on the nuScenes dataset.

Model	**Metrics**							**FPS**	**Year**	**Reference**
	mAP	**NDS**	**mATE**	**mASE**	**mAOE**	**mAVE**	**mAAE**			
CenterFusion	32.6	44.9	63.1	26.1	51.6	61.4	11.5	-	2021	[238]
CRAFT	41.1	52.3	46.7	26.8	45.3	51.9	11.4	4.1	2023	[231]
RCBEV	40.6	45.6	48.4	25.7	58.7	70.2	14.0	-	2023	[234]
MVFusion	45.3	51.7	56.9	24.6	37.9	78.1	12.8	-	2023	[239]
CRN	57.5	62.4	46.0	27.3	44.3	35.2	18.0	7.2	2023	[240]

5.3. Fusion of LiDAR and Vision-Based Methods for Vehicle Detection

Compared to millimeter-wave radar, LiDAR possesses a superior manufacturing process, which allows it to deliver higher precision and resolution in imaging technology. As a result, the integration of these two sensors is considered to be an outstanding method for environmental perception. Figure 10 illustrates the three fusion levels that are often present for LiDAR and vision-based methods, including the data level, decision level, and feature level.

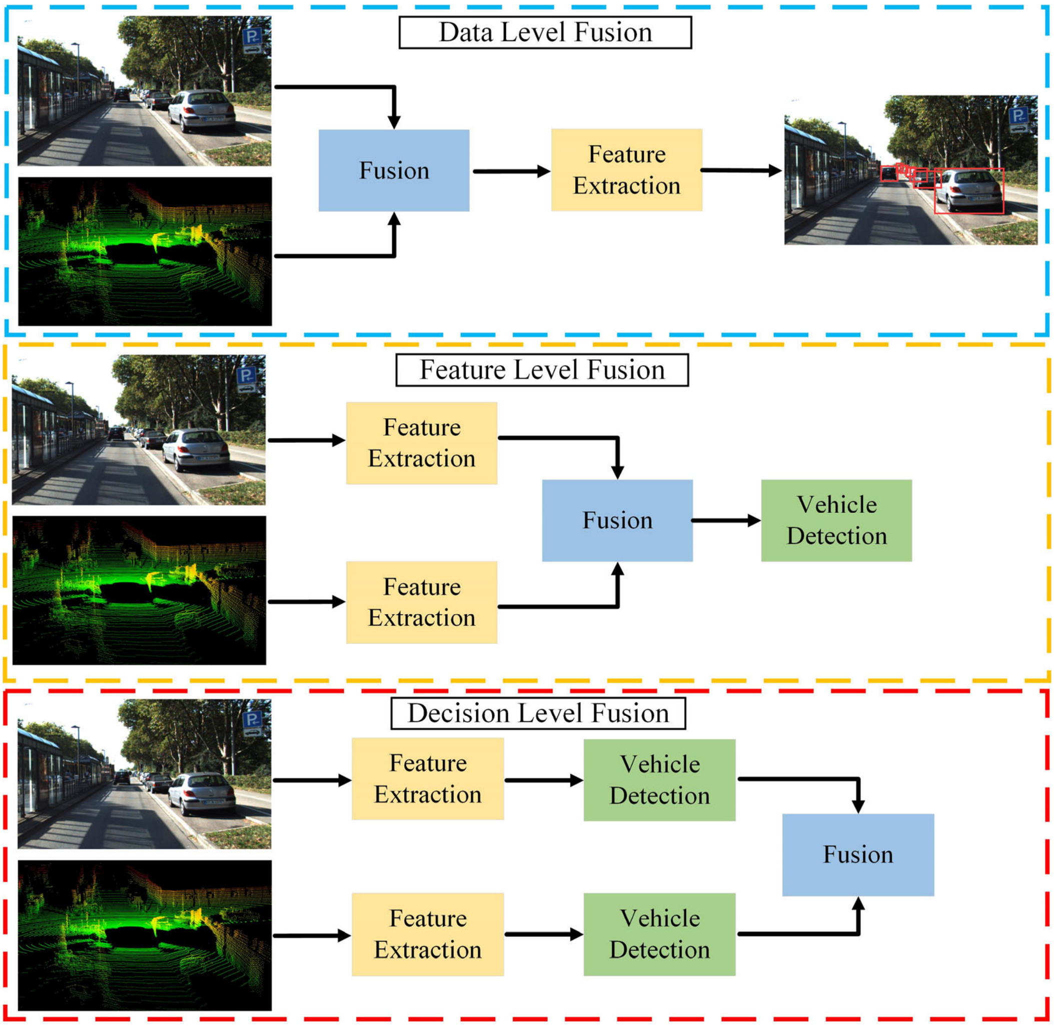


Figure 10. Different methods of LiDAR–camera fusion for vehicle detection [15].

(1) Data-Level Fusion

The raw data captured by LiDAR and a camera are inherently different in structure, which renders data-level fusion complex and potentially detrimental to the quality of information representation. As a result, this approach is currently no longer universally applicable in the context of environmental perception.

(2) Feature-Level Fusion

This approach integrates data extracted from LiDAR and a camera through a feature extraction mechanism. MV3D is a typical feature-level fusion framework [241]. This model takes the FV and BEV of the LiDAR, along with an image from a camera, as inputs, and projects the 3D proposals generated in the BEV into three views. Then, a feature fusion network is employed to fuse the feature maps obtained from ROI pooling of each view for vehicle detection. Ku et al. [242] used RPN to achieve multimodal feature fusion on high-resolution feature maps, followed by 3D regression and classification. Ku et al. proposed a method to accurately estimate the 3D bounding box [243]. The scheme first employs a 2D detector, and then lifts the 2D region to 3D to generate stymied proposals for 3D bounding box estimation. Zhao et al. [244] used LiDAR data to generate region proposals, and fed the generated ROIs from candidates into a CNN for vehicle detection. An et al. [245] leveraged the geometric consistency between 3D and 2D local regions, integrating manually crafted 2D features with attention-based voxel features to enhance the accuracy of 3D object detection. Li et al. [246] initially employed residual modules to extract image features and sparse convolutions to extract BEV map features from a LiDAR point cloud. Subsequently, these features were fused and fed into the RPN to detect vehicles. Liang et al. [247] converted multi-view images and LiDAR point clouds into BEV maps separately, and then used a dynamic fusion module to fuse the two feature maps for 3D target detection, which overcame the limitation of over-reliance on LiDAR data. Liu et al. [248] constructed a shared BEV map, thus preserving the semantic density of the camera and the geometry of the LiDAR. In addition, it can be extended to multi-task, multi-sensor frameworks. Wu et al. [249] proposed VirConvNet, a fast and efficient virtual point-based 3D object detection backbone network based on the VirConv operator, which reduces redundant computation and noise interference. The application of feature-level fusion improves the representation capability and thus helps to improve the accuracy of vehicle detection. However, the use of multiple feature extraction modules to obtain feature maps from sensors may lead to a decrease in speed.

(3) Decision-Level Fusion

A fusion algorithm is applied to the objects detected by the camera or LiDAR, respectively, in decision-level fusion. Typical fusion algorithms include the Kalman filter, Bayesian, etc. Oh et al. [250] used a CNN to fuse the classification outputs of an independent unary classifier. This classifier utilizes more than two pre-trained convolutional layers to consider local-to-global features as data representations. Guan et al. [251] proposed a decision-level object detection method based on Dempster–Shafer evidence theory. They complemented and converted the 2D LiDAR sparse depth map to a dense depth map, then used YOLOv3 [85] for vehicle target detection on both an RGB image and depth map, and then fused the two results to receive the information of vehicles. The decision-level approach has excellent robustness and ensures the normal operation of the system even if one of the sensors fails. At present, there is less research on decision-level methods compared to feature-level methods. Table 7 summarizes the performance of different LiDAR–camera-based deep learning models.

Table 7. Performance overview of LiDAR–camera-based models on KITTI dataset.

Model	Fusion	Car AP_{3D} (IoU = 0.7)			FPS	Year	Reference
		Easy	Moderate	Hard			
MV3D	Feature	-	-	-	2.8	2017	[241]
AVOD-FPN	Feature	81.94	71.88	66.38	10.0	2018	[242]
PointFusion	Feature	77.92	63.00	53.27	0.8	2018	[252]
ContFuse	Feature	82.54	66.22	64.04	16.7	2018	[253]
F-PointNet	Decision	83.76	70.92	63.65	-	2018	[243]
IPOD	Decision	84.10	76.40	75.30	-	2018	[254]
MMF	Feature	86.81	76.75	68.41	12.5	2019	[255]
F-ConvNet	Decision	85.88	76.51	68.08	-	2019	[256]
SIFRNet	Feature	85.62	72.05	64.19	-	2020	[257]
PointPainting	Feature	92.45	88.11	83.36	-	2020	[258]
EPNet	Feature	88.94	80.67	77.15	-	2020	[259]
F-PointPillars	Feature	88.90	79.28	78.07	14.3	2021	[260]
Fast-CLOCs	Feature	89.11	80.34	76.98	13.0	2022	[261]
SFD	Feature	91.73	84.76	77.92	-	2022	[262]
VPFNet	Feature	91.02	83.21	78.20	10.0	2022	[263]
FocalsConv	Feature	92.26	85.32	82.95	6.3	2022	[264]
VFF	Feature	92.31	85.51	82.92	-	2022	[265]
EPNet++	Feature	91.37	81.96	76.71	-	2022	[266]
PA3DNet	Feature	90.49	82.57	77.88	47.6	2023	[267]
DVF-PF	Feature	90.99	82.40	77.37	-	2023	[268]
LoGoNet	Feature	92.04	85.04	84.31	-	2023	[269]
VirConvNet	Feature	95.81	90.29	88.10	10.9	2023	[249]
VoxelNextFusion	Feature	90.40	82.03	79.86	18.5	2024	[270]

5.4. Multi-Sensor-Based Methods for Vehicle Detection

The original intentions behind the design of each sensor are different, and the fusion of multiple sensors to achieve multimodal perception is a promising trend for vehicle detection in autonomous driving in the future. Chavez-Garcia et al. [271] proposed a complete perception fusion architecture based on an evidence framework that combines composite representation with uncertainty management to solve the detection and tracking problem of moving targets. Yi et al. [272] presented a spatial calibration algorithm based on a multi-sensor system. They fused LiDAR, radar, and a camera to detect and recognize targets. In the realm of autonomous driving, environmental perception and vehicle detection typically entail the aggregation of information from disparate sensors. Therefore, multi-sensor fusion solutions can also be extended to encompass research on the fusion of one or two sensors.

6. Discussion and Future Trends

Sections 3–5 of this paper provide a detailed overview of the mainstream self-driving vehicle detection algorithms. In this section, we will discuss the algorithms for different sensors and explore the future trends of vehicle detectors.

6.1. Discussion

(1) Machine Vision

Camera sensors typically use stereo vision technology to acquire depth information of objects by comparing the disparity between images captured by two cameras to calculate the distance of vehicles from the cameras. Once the camera sensor captures the image, vehicle detection algorithms such as CNNs are employed to identify vehicles in the image and determine their positions and bounding boxes. The bounding boxes provide information about the size and orientation of the vehicles. By analyzing images from consecutive frames to recognize the displacement of objects, their speed and direction can thereby be determined. In the realm of machine-vision-based vehicle detection, deep learning approaches have taken the lead. Due to the powerful fitting and representation capabilities of deep models, they can extract deeper feature information, and thus are proven to be the optimal choice for vehicle detection. The vehicle detection algorithm based on semantic segmentation possesses a finer-grained representation compared to object-detection-based methods, and achieves higher precision. However, the trade-off in detection speed is a matter worth considering. The research on vehicle detection algorithms for camera sensors presents a diverse landscape, with each method having advantages and disadvantages. However, given the critical importance of speed metrics in autonomous driving, the design of algorithms must prioritize real-time performance.

(2) Millimeter-Wave Radar

Millimeter-wave radar can transmit millimeter-scale electromagnetic pulse wave energy and analyze the echo signal to receive the position and motion status of the vehicles. By utilizing antenna array processing, the millimeter-wave radar can derive angular data from a vehicle's reflection points. Upon encountering an object, the emitted millimeter-wave signal undergoes partial absorption and reflection, with the reflected signal returning to the sensor. By measuring the time difference between the transmitted and received signals, the distance between the object and the sensor is determined. This allows these points to be located in 3D space when combined with the time of flight. Millimeter-wave radar, as an inexpensive sensor, is capable of operating in all-weather conditions. However, millimeter-wave radar has low resolution and cannot provide information on the type and size of vehicles, which are indispensable requirements for vehicle detection. Therefore, millimeter-wave radar is usually used as an auxiliary sensor or in combination with other sensors. Nevertheless, its adaptability to complex weather conditions has led to its widespread application in the field of autonomous driving.

(3) LiDAR

LiDAR is an active sensor that plays a crucial role in detecting vehicles for autonomous driving due to its high precision and optical stability. The operation principle of a LiDAR sensor includes a laser emission, reception, and analysis. Firstly, a short-pulse laser beam is generated by the laser emitter to record its round-trip time, resulting in distance information. Subsequently, through the rotation or scanning of the sensor, LiDAR can acquire reflected signals in different directions, thus constructing a point cloud in three-dimensional space to achieve vehicle localization. By identifying and extracting parameters from the point cloud information, the size and shape of vehicles can be obtained. Methods based on deep learning and point cloud segmentation have been widely applied in vehicular radar systems. However, the sparse and unstructured nature of a point cloud leads to high computational costs when processing raw data directly. Researchers have adopted various data representation methods to structure the original point cloud and then utilize deep models for vehicle detection. This approach sacrifices some point cloud information in exchange for higher detection efficiency. LiDAR can function as an independent sensor or be fused with cameras for perception, representing a classic environmental perception scheme.

(4) Sensor Settings

Table 8 shows the sensor solutions of some autonomous driving manufacturers. We can see that Tesla and Xpeng have opted for a combination of cameras, millimeter-wave radar, and ultrasonic radar, rather than using LiDAR as their fundamental perception sensor. Other manufacturers have opted for LiDAR sensors, with Waymo in particular using four LiDAR sensors. It can be concluded that the mainstream trend in the field of environmental perception for autonomous vehicles is the fusion of radar and a vision sensing solution. Radars and cameras possess complementary characteristics, allowing for better perception in real-world environments.

Table 8. Autonomous driving sensor solutions of some manufacturers.

Company	Autonomous Driving System	Sensor Settings	Link
Baidu	Apollo	13 cameras 5 mmWave radars 2 LiDAR	https://www.apollo.auto/ (accessed on 12 May 2024)
Tesla	Autopilot	8 cameras mmWave radars 12 ultrasonic radars	https://www.tesla.com/ (accessed on 12 May 2024)
Waymo	Waymo Driver	29 cameras 6 mmWave radars 4 LiDAR	https://waymo.com/ (accessed on 12 May 2024)
NIO	Aquila	11 cameras 4 mmWave radars 1 LiDAR 12 ultrasonic radars	https://www.nio.cn/ (accessed on 12 May 2024)
Xpeng	XPILOT	13 cameras 12 ultrasonic radars 5 mmWave radars	https://www.xiaopeng.com/ (accessed on 12 May 2024)

6.2. Future Trends

(1) Balancing Speed and Accuracy of Algorithms

The performance of vehicle detection algorithms directly influences the perception capability of autonomous driving systems. In this field, the speed and accuracy of algorithms are the core issues in environmental perception for intelligent vehicles. Many algorithms exhibit either high precision or fast speed in practical deployment, but few can simultaneously achieve both aspects. Some scholars attempt to enhance one aspect of performance at the expense of another, which often lacks robustness in real driving scenarios. The performance of deep models depends on multiple prerequisites, with the design of the backbone network being considered one of the most critical factors. For general vehicle-embedded chips, lower computational complexity and faster processing speed are prioritized. Hence, exploring the effectiveness of network architectures while ensuring low algorithmic complexity is an intriguing research direction. One of the future research focuses will be on designing superior backbone network architectures to achieve a balance between accuracy and speed.

(2) Multi-Sensor Fusion Strategy

Vehicle detection algorithms based on machine vision, millimeter-wave radar, or LiDAR each possess distinct advantages and drawbacks. Inherent limitations persist in vehicle detection algorithms reliant on a singular sensor modality, rendering them unavoidably constrained. Therefore, multi-sensor fusion to realize cooperative perception will be a hot research topic in the future. Currently, industrial-grade autonomous driving is typically deployed with a multi-sensor fusion strategy, which is proven to be effective. Nonetheless, fusion techniques for vehicle detection still encounter challenges such as immaturity and representation disparities. To enhance the representation capabilities of multi-sensor fusion, further consideration in design schemes and protocols is necessary. For algorithms, research on enhanced fusion algorithms is imperative to maximize the utilization of non-redundant multiscale information for a collaborative perception of multiple sensors.

(3) Multi-tasking Algorithms

Existing vehicle detection methods, such as target detection and semantic segmentation, are experimented in specific traffic scenarios. Different algorithms are often optimized for varying usage contexts. However, practical vehicle detection in real-world traffic environment scenes confronts challenges of diversity and complexity, including adverse weather conditions such as fog, night-time, snow, and rain. Presently, most algorithms are tailored to specific scenes, lacking a universal approach capable of adaptive detection across various environments. Therefore, coping with vehicle detection in complex environments becomes an inevitable trend in the future. In future research, the integration of diverse algorithms into a framework adaptable to dynamic traffic conditions should be pursued. This can not only enhance detection speed and accuracy but also, more importantly, augment the adaptability and robustness of the vehicle detection system, mitigating the occurrence of traffic accidents due to perception failures.

(4) Unsupervised Learning

Today, almost all mainstream vehicle detection algorithms are based on supervised learning. These methods require large amounts of well-labeled data to train the model, and tend to have outstanding performance in test sets. However, these models require large computational resources for dataset training, which is a time-consuming and laborious task. Additionally, models based on supervised learning exhibit certain limitations in terms of generalization; when confronted with scenes divergent from the training sets, the detection accuracy often suffers. Hence, one feasible direction for the future is the development of semi-supervised or weakly supervised vehicle detection algorithms to address this issue. These algorithms can make better use of unlabeled data and thus achieve more accurate vehicle detection across a broader range of scenarios.

7. Conclusions

Autonomous driving technology is gradually changing the way people commute and reshaping transportation systems. Vehicle detection, the capability to perceive vehicles in real driving scenarios, has long been a topic of great interest in the field of autonomous driving. In this paper, we have provided a comprehensive review of vehicle detection algorithms for autonomous driving. We started by introducing the tasks, evaluation metrics, and datasets. Second, a detailed analysis of various detection methods was presented, such as machine-vision-based, millimeter-wave-radar-based, LiDAR-based, and sensor-fusion-based approaches. Finally, we delved into various sensor modalities and their associated detection algorithms, emphasizing the crucial balance between precision, speed, and environmental adaptability, and provided an outlook on future research directions. The main contribution of this work is to summarize and analyze over 200 classical as well as state-of-the-art vehicle detection algorithms in an organized manner, which helps researchers to have a deeper and more comprehensive understanding of this field. In the future, more systematic and comprehensive perception techniques will become the research hotspots. Sensor fusion strategies, multi-task algorithms, and unsupervised learning methods show a very promising trend. At the same time, striking a more reasonable balance between detection speed and detection accuracy has posed a challenge for researchers. Capturing these key elements not only enhances vehicle detection efficiency but also promotes the development of autonomous driving technology.

Author Contributions: Literature search, L.L., X.X., C.H. and Y.Z.; writing, L.L. and H.M.; original draft, L.L. and H.M.; chart drawing, L.L.; conceptualization, L.L., H.M. and X.X.; review and editing, H.M. and L.Z.; supervision, H.M., L.Z. and M.Z.; translating, L.L., H.M. and C.H.; project administration, H.M. and L.Z.; funding acquisition, H.M. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Key R&D and promotion projects in Henan Province under Grant 242102240117, the Natural Science Project of Zhengzhou Science and Technology Bureau under Grant 22ZZRDZX43, and the Research and Practice Project of Undergraduate Education and Teaching Reform in Henan University of Technology under Grant JXYJ2023010.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available: the pictures in Figure 1 were derived from Cityscapes [16]; the pictures in Figure 4 were sourced from nuScenes [27]; Figures 2 and 7–10 display pictures obtained from KITTI [15].

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

References

1. Kukkala, V.K.; Tunnell, J.; Pasricha, S.; Bradley, T. Advanced driver-assistance systems: A path toward autonomous vehicles. *IEEE Consum. Electron. Mag.* **2018**, *7*, 18–25. [CrossRef]
2. Crayton, T.J.; Meier, B.M. Autonomous vehicles: Developing a public health research agenda to frame the future of transportation policy. *J. Transp. Health* **2017**, *6*, 245–252. [CrossRef]
3. Shadrin, S.S.; Ivanova, A.A. Analytical review of standard Sae J3016 «taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles» with latest updates. *Avtomob. Doroga Infrastrukt.* **2019**, *3*, 10.
4. Karangwa, J.; Liu, J.; Zeng, Z. Vehicle detection for autonomous driving: A review of algorithms and datasets. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11568–11594. [CrossRef]
5. Alam, F.; Mehmood, R.; Katib, I.; Altowaijri, S.M.; Albeshri, A. TAAWUN: A decision fusion and feature specific road detection approach for connected autonomous vehicles. *Mob. Netw. Appl.* **2023**, *28*, 636–652. [CrossRef]
6. Sivaraman, S.; Trivedi, M.M. A review of recent developments in vision-based vehicle detection. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, 23–26 June 2013; pp. 310–315.
7. Bouguettaya, A.; Zarzour, H.; Kechida, A.; Taberkit, A.M. Vehicle detection from UAV imagery with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6047–6067. [CrossRef] [PubMed]
8. Gormley, M.; Walsh, T.; Fuller, R. Risks in the Driving of Emergency Service Vehicles. *Ir. J. Psychol.* **2008**, *29*, 7–18. [CrossRef]
9. Chadwick, S.; Maddern, W.; Newman, P. Distant vehicle detection using radar and vision. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8311–8317.
10. Sivaraman, S.; Trivedi, M.M. Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1773–1795. [CrossRef]
11. Sun, Z.; Bebis, G.; Miller, R. On-Road Vehicle Detection: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 694–711.
12. Wang, Z.; Zhan, J.; Duan, C.; Guan, X.; Lu, P.; Yang, K. A review of vehicle detection techniques for intelligent vehicles. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 3811–3831. [CrossRef]
13. Liu, Q.; Li, Z.; Yuan, S.; Zhu, Y.; Li, X. Review on vehicle detection technology for unmanned ground vehicles. *Sensors* **2021**, *21*, 1354. [CrossRef] [PubMed]

14. Wei, Z.; Zhang, F.; Chang, S.; Liu, Y.; Wu, H.; Feng, Z. Mmwave radar and vision fusion for object detection in autonomous driving: A review. *Sensors* **2022**, *22*, 2542. [CrossRef] [PubMed]
15. The KITTI Vision Benchmark Suite. Available online: https://www.cvlibs.net/datasets/kitti (accessed on 7 May 2024).
16. Cityscapes Dataset. Available online: https://www.cityscapes-dataset.com (accessed on 7 May 2024).
17. Oxford Radar RobotCar Dataset. Available online: https://oxford-robotics-institute.github.io/radar-robotcar-dataset (accessed on 7 May 2024).
18. Mapillary Vistas Dataset. Available online: https://www.mapillary.com (accessed on 7 May 2024).
19. Berkeley DeepDrive. Available online: http://bdd-data.berkeley.edu (accessed on 7 May 2024).
20. ApolloScape Advanced Open Tools and Datasets for Autonomous Driving. Available online: https://apolloscape.auto (accessed on 7 May 2024).
21. KAIST Multispectral Pedestrian Detection Benchmark. Available online: http://multispectral.kaist.ac.kr (accessed on 7 May 2024).
22. Waymo Open Dataset. Available online: https://waymo.com/open (accessed on 7 May 2024).
23. Self-Driving Motion Prediction Dataset. Available online: https://github.com/woven-planet/l5kit (accessed on 7 May 2024).
24. Argoverse 1. Available online: https://www.argoverse.org/av1.html (accessed on 7 May 2024).
25. D^2-City. Available online: https://www.v7labs.com/open-datasets/d2-city (accessed on 7 May 2024).
26. H3D Honda 3D Dataset. Available online: https://usa.honda-ri.com//H3D (accessed on 7 May 2024).
27. nuScenes. Available online: https://www.nuscenes.org/nuscenes (accessed on 7 May 2024).
28. Canadian Adverse Driving Conditions Dataset. Available online: http://cadcd.uwaterloo.ca (accessed on 7 May 2024).
29. Audi Autonomous Driving Dataset. Available online: https://www.a2d2.audi/a2d2/en.html (accessed on 7 May 2024).
30. A*3D: An Autonomous Driving Dataset in Challeging Environments. Available online: https://github.com/I2RDL2/ASTAR-3D (accessed on 7 May 2024).
31. Heriot-Watt RADIATE Dataset. Available online: https://pro.hw.ac.uk/radiate (accessed on 7 May 2024).
32. ACDC DATASET. Available online: https://acdc.vision.ee.ethz.ch (accessed on 7 May 2024).
33. KITTI-360: A Large-Scale Dataset with 3D&2D Annotations. Available online: https://www.cvlibs.net/datasets/kitti-360 (accessed on 7 May 2024).
34. SHIFT DATASET: A Synthetic Driving Dataset For Continuous Multi-Task Domain Adaptation. Available online: https://www.vis.xyz/shift (accessed on 7 May 2024).
35. Argoverse 2. Available online: https://www.argoverse.org/av2.html (accessed on 7 May 2024).
36. V2v4real: The First Large-Scale, Real-World Multimodal Dataset for Vehicle-to-Vehicle (V2V) Perception. Available online: https://mobility-lab.seas.ucla.edu/v2v4real (accessed on 7 May 2024).
37. Bertozzi, M.; Broggi, A.; Fascioli, A. Vision-based intelligent vehicles: State of the art and perspectives. *Robot. Auton. Syst.* **2000**, *32*, 1–16. [CrossRef]
38. Endsley, M.R. Autonomous driving systems: A preliminary naturalistic study of the Tesla Model S. *J. Cognit. Eng. Decis. Making* **2017**, *11*, 225–238. [CrossRef]
39. Yoffie, D.B. *Mobileye: The Future of Driverless Cars*; Harvard Business School Case; Harvard Business Review Press: Cambridge, MA, USA, 2014; pp. 421–715.
40. Russell, A.; Zou, J.J. Vehicle detection based on color analysis. In Proceedings of the 2012 International Symposium on Communications and Information Technologies (ISCIT), Gold Coast, Australia, 2–5 October 2012; pp. 620–625.
41. Shao, H.X.; Duan, X.M. Video vehicle detection method based on multiple color space information fusion. *Adv. Mater. Res.* **2012**, *546*, 721–726. [CrossRef]
42. Chen, H.-T.; Wu, Y.-C.; Hsu, C.-C. Daytime preceding vehicle brake light detection using monocular vision. *IEEE Sens. J.* **2015**, *16*, 120–131. [CrossRef]
43. Teoh, S.S.; Bräunl, T. Symmetry-based monocular vehicle detection system. *Mach. Vis. Appl.* **2012**, *23*, 831–842. [CrossRef]
44. Tsai, W.-K.; Wu, S.-L.; Lin, L.-J.; Chen, T.-M.; Li, M.-H. Edge-based forward vehicle detection method for complex scenes. In Proceedings of the 2014 IEEE International Conference on Consumer Electronics-Taiwan, Taipei, Taiwan, 26–28 May 2014; pp. 173–174.
45. Mu, K.; Hui, F.; Zhao, X.; Prehofer, C. Multiscale edge fusion for vehicle detection based on difference of Gaussian. *Optik* **2016**, *127*, 4794–4798. [CrossRef]
46. Nur, S.A.; Ibrahim, M.; Ali, N.; Nur, F.I.Y. Vehicle detection based on underneath vehicle shadow using edge features. In Proceedings of the 2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 25–27 November 2016; pp. 407–412.
47. Creusot, C.; Munawar, A. Real-time small obstacle detection on highways using compressive RBM road reconstruction. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Republic of Korea, 28 June–1 July 2015; pp. 162–167.
48. Chen, X.; Chen, H.; Xu, H. Vehicle detection based on multifeature extraction and recognition adopting RBF neural network on ADAS system. *Complexity* **2020**, *2020*, 8842297. [CrossRef]
49. Ibarra-Arenado, M.; Tjahjadi, T.; Pérez-Oria, J.; Robla-Gómez, S.; Jiménez-Avello, A. Shadow-based vehicle detection in urban traffic. *Sensors* **2017**, *17*, 975. [CrossRef]

50. Kosaka, N.; Ohashi, G. Vision-based nighttime vehicle detection using CenSurE and SVM. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2599–2608. [CrossRef]
51. Satzoda, R.K.; Trivedi, M.M. Looking at vehicles in the night: Detection and dynamics of rear lights. *IEEE Trans. Intell. Transp. Syst.* **2016**, *20*, 4297–4307. [CrossRef]
52. Gao, L.; Li, C.; Fang, T.; Xiong, Z. Vehicle detection based on color and edge information. In Proceedings of the Image Analysis and Recognition: 5th International Conference, Berlin/Heidelberg, Germany, 25–27 June 2008; pp. 142–150.
53. Pradeep, C.S.; Ramanathan, R. An improved technique for night-time vehicle detection. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Karnataka, India, 19–22 September 2018; pp. 508–513.
54. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
55. Yan, G.; Yu, M.; Yu, Y.; Fan, L. Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification. *Optik* **2016**, *127*, 7941–7951. [CrossRef]
56. Khairdoost, N.; Monadjemi, S.A.; Jamshidi, K. Front and rear vehicle detection using hypothesis generation and verification. *Signal Image Process.* **2013**, *4*, 31. [CrossRef]
57. Cheon, M.; Lee, W.; Yoon, C.; Park, M. Vision-based vehicle detection system with consideration of the detecting location. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1243–1252. [CrossRef]
58. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645. [CrossRef]
59. Wen, X.; Shao, L.; Fang, W.; Xue, Y. Efficient feature selection and classification for vehicle detection. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *25*, 508–517.
60. Ojala, T.; Pietikainen, M.; Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel, 9–13 October 1994; pp. 582–585.
61. Feichtinger, H.G.; Strohmer, T. *Gabor Analysis and Algorithms: Theory and Applications*; Springer: New York, NY, USA, 2012.
62. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 404–417.
63. Yang, X.; Yang, Y. A method of efficient vehicle detection based on HOG-LBP. *Comput. Eng.* **2014**, *40*, 210–214.
64. Arunmozhi, A.; Park, J. Comparison of HOG, LBP and Haar-like features for on-road vehicle detection. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; pp. 0362–0367.
65. Webb, G.I.; Zheng, Z. Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 980–991. [CrossRef]
66. Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; Ma, Q. A survey on ensemble learning. *Front. Comput. Sci.* **2020**, *14*, 241–258. [CrossRef]
67. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
68. Ali, A.M.; Eltarhouni, W.I.; Bozed, K.A. On-road vehicle detection using support vector machine and decision tree classifications. In Proceedings of the 6th International Conference on Engineering & MIS 2020, Istanbul, Turkey, 4–6 July 2020; pp. 1–5.
69. Sivaraman, S.; Trivedi, M.M. Active learning for on-road vehicle detection: A comparative study. *Mach. Vis. Appl.* **2014**, *25*, 599–611. [CrossRef]
70. Hsieh, J.-W.; Chen, L.-C.; Chen, D.-Y. Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 6–20. [CrossRef]
71. Sun, Z.; Bebis, G.; Miller, R. Monocular precrash vehicle detection: Features and classifiers. *IEEE Trans. Image Process.* **2006**, *15*, 2019–2034. [PubMed]
72. Ho, W.T.; Lim, H.W.; Tay, Y.H. Two-stage license plate detection using gentle Adaboost and SIFT-SVM. In Proceedings of the 2009 First Asian Conference on Intelligent Information and Database Systems, Quang Binh, Vietnam, 1–3 April 2009; pp. 109–114.
73. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; pp. 91–99.
74. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
75. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
76. 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.
77. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
78. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
79. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

80. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
81. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
82. Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L.; Ling, H. M2det: A single-shot object detector based on multi-level feature pyramid network. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 9259–9266.
83. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
84. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
85. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
86. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
87. Ultralytics YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 7 May 2024).
88. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
89. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
90. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
91. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. Reppoints: Point set representation for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 9657–9666.
92. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 6569–6578.
93. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 850–859.
94. Lu, X.; Li, B.; Yue, Y.; Li, Q.; Yan, J. Grid r-cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7363–7372.
95. Zhu, C.; He, Y.; Savvides, M. Feature selective anchor-free module for single-shot object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 840–849.
96. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
97. Wang, J.; Chen, K.; Yang, S.; Loy, C.C.; Lin, D. Region proposal by guided anchoring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2965–2974.
98. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Li, L.; Shi, J. Foveabox: Beyound anchor-based object detection. *IEEE Trans. Image Process.* **2020**, *29*, 7389–7398. [CrossRef]
99. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
100. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972.
101. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
102. Wang, J.; Song, L.; Li, Z.; Sun, H.; Sun, J.; Zheng, N. End-to-end object detection with fully convolutional network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 10–25 June 2021; pp. 15849–15858.
103. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C. Sparse r-cnn: End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 10–25 June 2021; pp. 14454–14463.
104. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
105. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
106. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
107. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
108. Wang, Y.; Zhang, X.; Yang, T.; Sun, J. Anchor detr: Query design for transformer-based detector. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 22 February–1 March 2022; pp. 2567–2575.
109. Lv, W.; Xu, S.; Zhao, Y.; Wang, G.; Wei, J.; Cui, C.; Du, Y.; Dang, Q.; Liu, Y. Detrs beat yolos on real-time object detection. *arXiv* **2023**, arXiv:2304.08069.

110. Zhu, Y.; Zhao, C.; Wang, J.; Zhao, X.; Wu, Y.; Lu, H. Couplenet: Coupling global structure with local parts for object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4126–4134.
111. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Detnet: A backbone network for object detection. *arXiv* **2018**, arXiv:1804.06215.
112. 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.
113. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7310–7311.
114. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3578–3587.
115. Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
116. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4203–4212.
117. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
118. Yao, Z.; Ai, J.; Li, B.; Zhang, C. Efficient detr: Improving end-to-end object detector with dense prior. *arXiv* **2021**, arXiv:2104.01318.
119. Zhou, Z.H. A brief introduction to weakly supervised learning. *Natl. Sci. Rev.* **2017**, *5*, 44–53. [CrossRef]
120. Pinheiro, P.O.O.; Collobert, R.; Dollár, P. Learning to segment object candidates. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 1990–1998.
121. Pinheiro, P.O.; Lin, T.-Y.; Collobert, R.; Dollár, P. Learning to refine object segments. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 75–91.
122. Zagoruyko, S.; Lerer, A.; Lin, T.-Y.; Pinheiro, P.O.; Gross, S.; Chintala, S.; Dollár, P. A multipath network for object detection. *arXiv* **2016**, arXiv:1604.02135.
123. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
124. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
125. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
126. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
127. 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.
128. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
129. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
130. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
131. Lin, G.; Milan, A.; Shen, C.; Reid, I. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1925–1934.
132. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
133. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. Icnet for real-time semantic segmentation on high-resolution images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 405–420.
134. Luc, P.; Couprie, C.; Chintala, S.; Verbeek, J. Semantic segmentation using adversarial networks. *arXiv* **2016**, arXiv:1611.08408.
135. Souly, N.; Spampinato, C.; Shah, M. Semi supervised semantic segmentation using generative adversarial network. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5688–5696.
136. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 10–25 June 2021; pp. 6881–6890.
137. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
138. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; pp. 12077–12090.

139. Wan, Q.; Huang, Z.; Lu, J.; Yu, G.; Zhang, L. Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation. *arXiv* **2023**, arXiv:2301.13156.
140. 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.
141. Li, H.; Xiong, P.; Fan, H.; Sun, J. Dfanet: Deep feature aggregation for real-time semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9522–9531.
142. Wang, Y.; Zhou, Q.; Liu, J.; Xiong, J.; Gao, G.; Wu, X.; Latecki, L.J. Lednet: A lightweight encoder-decoder network for real-time semantic segmentation. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1860–1864.
143. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina: Chapel Hill, NC, USA, 1995.
144. Kim, C.; Li, F.; Ciptadi, A.; Rehg, J.M. Multiple hypothesis tracking revisited. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4696–4704.
145. Roberts, W.; Stoica, P.; Li, J.; Yardibi, T.; Sadjadi, F.A. Iterative adaptive approaches to MIMO radar imaging. *IEEE J. Sel. Top. Signal Process.* **2010**, *4*, 5–20. [CrossRef]
146. Pang, S.; Zeng, Y.; Yang, Q.; Deng, B.; Wang, H.; Qin, Y. Improvement in SNR by adaptive range gates for RCS measurements in the THz region. *Electronics* **2019**, *8*, 805. [CrossRef]
147. Major, B.; Fontijne, D.; Ansari, A.; Teja Sukhavasi, R.; Gowaikar, R.; Hamilton, M.; Lee, S.; Grzechnik, S.; Subramanian, S. Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1–9.
148. Sligar, A.P. Machine learning-based radar perception for autonomous vehicles using full physics simulation. *IEEE Access* **2020**, *8*, 51470–51476. [CrossRef]
149. Akita, T.; Mita, S. Object tracking and classification using millimeter-wave radar based on LSTM. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 1110–1115.
150. Zhao, Z.; Song, Y.; Cui, F.; Zhu, J.; Song, C.; Xu, Z.; Ding, K. Point cloud features-based kernel SVM for human-vehicle classification in millimeter wave radar. *IEEE Access* **2020**, *8*, 26012–26021. [CrossRef]
151. Guan, J.; Madani, S.; Jog, S.; Gupta, S.; Hassanieh, H. Through fog high-resolution imaging using millimeter wave radar. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11464–11473.
152. Li, P.; Wang, P.; Berntorp, K.; Liu, H. Exploiting temporal relations on radar perception for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 17071–17080.
153. Huang, Z.; Pan, Z.; Lei, B. Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data. *Remote Sens.* **2017**, *9*, 907. [CrossRef]
154. Kim, W.; Cho, H.; Kim, J.; Kim, B.; Lee, S. YOLO-based simultaneous target detection and classification in automotive FMCW radar systems. *Sensors* **2020**, *20*, 2897. [CrossRef] [PubMed]
155. Douillard, B.; Underwood, J.; Kuntz, N.; Vlaskine, V.; Quadros, A.; Morton, P.; Frenkel, A. On the segmentation of 3D LIDAR point clouds. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2798–2805.
156. Wen, M.; Cho, S.; Chae, J.; Sung, Y.; Cho, K. Range image-based density-based spatial clustering of application with noise clustering method of three-dimensional point clouds. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1735–1754. [CrossRef]
157. Lee, S.-M.; Im, J.J.; Lee, B.-H.; Leonessa, A.; Kurdila, A. A real-time grid map generation and object classification for ground-based 3D LIDAR data using image analysis techniques. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 2253–2256.
158. Reymann, C.; Lacroix, S. Improving LiDAR point cloud classification using intensities and multiple echoes. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5122–5128.
159. Bogoslavskyi, I.; Stachniss, C. Efficient online segmentation for sparse 3D laser scans. *PFG-J. Photogramm. Remote Sens. Geoinf. Sci.* **2017**, *85*, 41–52. [CrossRef]
160. Byun, J.; Na, K.-I.; Seo, B.-S.; Roh, M. Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle. In *Field and Service Robotics: Results of the 9th International Conference*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; Volume 105, pp. 49–60.
161. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
162. Asvadi, A.; Premebida, C.; Peixoto, P.; Nunes, U. 3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robot. Auton. Syst.* **2016**, *83*, 299–311. [CrossRef]
163. Engelcke, M.; Rao, D.; Wang, D.Z.; Tong, C.H.; Posner, I. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1355–1361.

164. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
165. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5099–5108.
166. Shi, S.; Wang, X.; Li, H. Pointrcnn: 3d object proposal generation and detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 770–779.
167. Yang, Z.; Sun, Y.; Liu, S.; Jia, J. 3dssd: Point-based 3d single stage object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11040–11048.
168. Asvadi, A.; Garrote, L.; Premebida, C.; Peixoto, P.; Nunes, U.J. DepthCN: Vehicle detection using 3D-LIDAR and ConvNet. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
169. Zeng, Y.; Hu, Y.; Liu, S.; Ye, J.; Han, Y.; Li, X.; Sun, N. Rt3d: Real-time 3-d vehicle detection in lidar point cloud for autonomous driving. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3434–3440. [CrossRef]
170. Beltrán, J.; Guindel, C.; Moreno, F.M.; Cruzado, D.; Garcia, F.; De La Escalera, A. Birdnet: A 3d object detection framework from lidar information. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3517–3523.
171. Barrera, A.; Guindel, C.; Beltrán, J.; García, F. Birdnet+: End-to-end 3d object detection in lidar bird's eye view. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–6.
172. Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8445–8453.
173. Qian, R.; Garg, D.; Wang, Y.; You, Y.; Belongie, S.; Hariharan, B.; Campbell, M.; Weinberger, K.Q.; Chao, W.-L. End-to-end pseudo-lidar for image-based 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 5881–5890.
174. Chen, Y.-N.; Dai, H.; Ding, Y. Pseudo-stereo for monocular 3d object detection in autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 887–897.
175. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4490–4499.
176. Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; Li, H. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 2–9 February 2021; pp. 1201–1209.
177. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10529–10538.
178. Shi, S.; Jiang, L.; Deng, J.; Wang, Z.; Guo, C.; Shi, J.; Wang, X.; Li, H. PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection. *Int. J. Comput. Vis.* **2023**, *131*, 531–551. [CrossRef]
179. Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; Jia, J. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 21674–21683.
180. Liu, M.; Ma, J.; Zheng, Q.; Liu, Y.; Shi, G. 3D Object Detection Based on Attention and Multi-Scale Feature Fusion. *Sensors* **2022**, *22*, 3935. [CrossRef]
181. Hu, J.S.; Kuai, T.; Waslander, S.L. Point density-aware voxels for lidar 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8469–8478.
182. Wang, L.; Song, Z.; Zhang, X.; Wang, C.; Zhang, G.; Zhu, L.; Li, J.; Liu, H. SAT-GCN: Self-attention graph convolutional network-based 3D object detection for autonomous driving. *Knowl.-Based Syst.* **2023**, *259*, 110080. [CrossRef]
183. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. Std: Sparse-to-dense 3d object detector for point cloud. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 1951–1960.
184. Shi, S.; Wang, Z.; Shi, J.; Wang, X.; Li, H. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 2647–2664. [CrossRef] [PubMed]
185. He, C.; Zeng, H.; Huang, J.; Hua, X.-S.; Zhang, L. Structure aware single-stage 3d object detection from point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11873–11882.
186. Mao, J.; Niu, M.; Bai, H.; Liang, X.; Xu, H.; Xu, C. Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 10–25 June 2021; pp. 2723–2732.
187. Yang, J.; Shi, S.; Wang, Z.; Li, H.; Qi, X. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 10–25 June 2021; pp. 10368–10378.
188. Chen, C.; Chen, Z.; Zhang, J.; Tao, D. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 22 February–1 March 2022; pp. 221–229.

189. Zhang, L.; Dong, R.; Tai, H.-S.; Ma, K. Pointdistiller: Structured knowledge distillation towards efficient and compact 3d detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 21791–21801.
190. Xiong, S.; Li, B.; Zhu, S. DCGNN: A single-stage 3D object detection network based on density clustering and graph neural network. *Complex Intell. Syst.* **2023**, *9*, 3399–3408. [CrossRef]
191. Yang, B.; Luo, W.; Urtasun, R. Pixor: Real-time 3d object detection from point clouds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7652–7660.
192. Simony, M.; Milzy, S.; Amendey, K.; Gross, H.-M. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 197–209.
193. Zhang, X.; Wang, L.; Zhang, G.; Lan, T.; Zhang, H.; Zhao, L.; Li, J.; Zhu, L.; Liu, H. RI-Fusion: 3D object detection using enhanced point features with range-image fusion for autonomous driving. *IEEE Trans. Instrum. Meas.* **2022**, *72*, 1–13. [CrossRef]
194. Li, B. 3d fully convolutional network for vehicle detection in point cloud. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1513–1518.
195. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [CrossRef] [PubMed]
196. Ye, M.; Xu, S.; Cao, T. Hvnet: Hybrid voxel network for lidar based 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1631–1640.
197. Liu, Z.; Zhao, X.; Huang, T.; Hu, R.; Zhou, Y.; Bai, X. Tanet: Robust 3d object detection from point clouds with triple attention. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 11677–11684.
198. Xiao, W.; Peng, Y.; Liu, C.; Gao, J.; Wu, Y.; Li, X. Balanced Sample Assignment and Objective for Single-Model Multi-Class 3D Object Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 5036–5048. [CrossRef]
199. Wang, X.; Zou, L.; Shen, X.; Ren, Y.; Qin, Y. A region-growing approach for automatic outcrop fracture extraction from a three-dimensional point cloud. *Comput. Geosci.* **2017**, *99*, 100–106. [CrossRef]
200. Sun, S.; Li, C.; Chee, P.W.; Paterson, A.H.; Jiang, Y.; Xu, R.; Robertson, J.S.; Adhikari, J.; Shehzad, T. Three-dimensional photogrammetric mapping of cotton bolls in situ based on point cloud segmentation and clustering. *ISPRS-J. Photogramm. Remote Sens.* **2020**, *160*, 195–207. [CrossRef]
201. Zhao, B.; Hua, X.; Yu, K.; Xuan, W.; Chen, X.; Tao, W. Indoor point cloud segmentation using iterative gaussian mapping and improved model fitting. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7890–7907. [CrossRef]
202. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; Markham, A. Randla-net: Efficient semantic segmentation of large-scale point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11108–11117.
203. Cheng, R.; Razani, R.; Ren, Y.; Bingbing, L. S3Net: 3D LiDAR sparse semantic segmentation network. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 14040–14046.
204. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 6411–6420.
205. Landrieu, L.; Simonovsky, M. Large-scale point cloud semantic segmentation with superpoint graphs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4558–4567.
206. Wu, B.; Wan, A.; Yue, X.; Keutzer, K. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1887–1893.
207. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
208. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4376–4382.
209. Xu, C.; Wu, B.; Wang, Z.; Zhan, W.; Vajda, P.; Keutzer, K.; Tomizuka, M. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 1–19.
210. Feng, Y.; Zhang, Z.; Zhao, X.; Ji, R.; Gao, Y. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 264–272.
211. Maturana, D.; Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
212. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. Segcloud: Semantic segmentation of 3d point clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 537–547.
213. Klokov, R.; Lempitsky, V. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 863–872.

214. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching efficient 3d architectures with sparse point-voxel convolution. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 685–702.
215. Le, T.; Duan, Y. Pointgrid: A deep network for 3d shape understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 9204–9214.
216. Riegler, G.; Osman Ulusoy, A.; Geiger, A. Octnet: Learning deep 3d representations at high resolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3577–3586.
217. Wang, P.-S.; Liu, Y.; Guo, Y.-X.; Sun, C.-Y.; Tong, X. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.* **2017**, *36*, 1–11. [CrossRef]
218. Hou, Y.; Zhu, X.; Ma, Y.; Loy, C.C.; Li, Y. Point-to-voxel knowledge distillation for lidar semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 14–24 June 2022; pp. 8479–8488.
219. Hu, Z.; Uchimura, K. UV-disparity: An efficient algorithm for stereovision based scene analysis. In Proceedings of the IEEE Proceedings. Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005; pp. 48–54.
220. Xie, Y.; Zeng, S.; Zhang, Y.; Chen, L. A cascaded framework for robust traversable region estimation using stereo vision. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 3075–3080.
221. Ma, W.; Zhu, S. A Multifeature-Assisted Road and Vehicle Detection Method Based on Monocular Depth Estimation and Refined UV Disparity Mapping. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16763–16772. [CrossRef]
222. Lefebvre, S.; Ambellouis, S. Vehicle detection and tracking using mean shift segmentation on semi-dense disparity maps. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, New York, NY, USA, 3–7 June 2012; pp. 855–860.
223. Neumann, D.; Langner, T.; Ulbrich, F.; Spitta, D.; Goehring, D. Online vehicle detection using Haar-like, LBP and HOG feature based image classifiers with stereo vision preselection. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 773–778.
224. Xie, Q.; Long, Q.; Li, J.; Zhang, L.; Hu, X. Application of intelligence binocular vision sensor: Mobility solutions for automotive perception system. *IEEE Sens. J.* **2023**, *24*, 5578–5592. [CrossRef]
225. Kale, K.; Pawar, S.; Dhulekar, P. Moving object tracking using optical flow and motion vector estimation. In Proceedings of the 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida, India, 2–4 September 2015; pp. 1–6.
226. Sengar, S.S.; Mukhopadhyay, S. Detection of moving objects based on enhancement of optical flow. *Optik* **2017**, *145*, 130–141. [CrossRef]
227. Chen, Q.; Koltun, V. Full flow: Optical flow estimation by global optimization over regular grids. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4706–4714.
228. Yin, Z.; Shi, J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1983–1992.
229. Wang, T.; Zheng, N.; Xin, J.; Ma, Z. Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications. *Sensors* **2011**, *11*, 8992–9008. [CrossRef] [PubMed]
230. Wang, X.; Xu, L.; Sun, H.; Xin, J.; Zheng, N. On-road vehicle detection and tracking using MMW radar and monovision fusion. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2075–2084. [CrossRef]
231. Kim, Y.; Kim, S.; Choi, J.W.; Kum, D. Craft: Camera-radar 3d object detection with spatio-contextual fusion transformer. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 1160–1168.
232. Lekic, V.; Babic, Z. Automotive radar and camera fusion using generative adversarial networks. *Comput. Vis. Image Underst.* **2019**, *184*, 1–8. [CrossRef]
233. Chang, S.; Zhang, Y.; Zhang, F.; Zhao, X.; Huang, S.; Feng, Z.; Wei, Z. Spatial attention fusion for obstacle detection using mmwave radar and vision sensor. *Sensors* **2020**, *20*, 956. [CrossRef] [PubMed]
234. Zhou, T.; Chen, J.; Shi, Y.; Jiang, K.; Yang, M.; Yang, D. Bridging the view disparity between radar and camera features for multi-modal fusion 3d object detection. *IEEE Trans. Intell. Veh.* **2023**, *8*, 1523–1535. [CrossRef]
235. Zhong, Z.; Liu, S.; Mathew, M.; Dubey, A. Camera radar fusion for increased reliability in ADAS applications. *Electron. Imaging* **2018**, *17*, 258. [CrossRef]
236. Bai, J.; Li, S.; Huang, L.; Chen, H. Robust detection and tracking method for moving object based on radar and camera data fusion. *IEEE Sens. J.* **2021**, *21*, 10761–10774. [CrossRef]
237. Sengupta, A.; Cheng, L.; Cao, S. Robust multiobject tracking using mmwave radar-camera sensor fusion. *IEEE Sens. Lett.* **2022**, *6*, 1–4. [CrossRef]
238. Nabati, R.; Qi, H. Centerfusion: Centre-based radar and camera fusion for 3d object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; pp. 1527–1536.
239. Wu, Z.; Chen, G.; Gan, Y.; Wang, L.; Pu, J. Mvfusion: Multi-view 3d object detection with semantic-aligned radar and camera fusion. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 2766–2773.
240. Kim, Y.; Shin, J.; Kim, S.; Lee, I.-J.; Choi, J.W.; Kum, D. Crn: Camera radar net for accurate, robust, efficient 3d perception. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Vancouver, BC, Canada, 17–24 June 2023; pp. 17615–17626.

241. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3d object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
242. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S.L. Joint 3d proposal generation and object detection from view aggregation. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–8.
243. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum pointnets for 3d object detection from rgb-d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 918–927.
244. Zhao, X.; Sun, P.; Xu, Z.; Min, H.; Yu, H. Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications. *IEEE Sens. J.* **2020**, *20*, 4901–4913. [CrossRef]
245. An, P.; Liang, J.; Yu, K.; Fang, B.; Ma, J. Deep structural information fusion for 3D object detection on LiDAR–camera system. *Comput. Vis. Image Underst.* **2022**, *214*, 103295. [CrossRef]
246. Li, J.; Li, R.; Li, J.; Wang, J.; Wu, Q.; Liu, X. Dual-view 3d object recognition and detection via lidar point cloud and camera image. *Robot. Auton. Syst.* **2022**, *150*, 103999. [CrossRef]
247. Liang, T.; Xie, H.; Yu, K.; Xia, Z.; Lin, Z.; Wang, Y.; Tang, T.; Wang, B.; Tang, Z. Bevfusion: A simple and robust lidar-camera fusion framework. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 10421–10434.
248. Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.L.; Han, S. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 2774–2781.
249. Wu, H.; Wen, C.; Shi, S.; Li, X.; Wang, C. Virtual Sparse Convolution for Multimodal 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and, and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 21653–21662.
250. Oh, S.-I.; Kang, H.-B. Object detection and classification by decision-level fusion for intelligent vehicle systems. *Sensors* **2017**, *17*, 207. [CrossRef] [PubMed]
251. Guan, L.; Chen, Y.; Wang, G.; Lei, X. Real-time vehicle detection framework based on the fusion of LiDAR and camera. *Electronics* **2020**, *9*, 451. [CrossRef]
252. Xu, D.; Anguelov, D.; Jain, A. Pointfusion: Deep sensor fusion for 3D bounding box estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 244–253.
253. Liang, M.; Yang, B.; Wang, S.; Urtasun, R. Deep continuous fusion for multi-sensor 3D object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 641–656.
254. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. Ipod: Intensive point-based object detector for point cloud. *arXiv* **2018**, arXiv:1812.05276.
255. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-task multi-sensor fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7345–7353.
256. Wang, Z.; Jia, K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 1742–1749.
257. Zhao, X.; Liu, Z.; Hu, R.; Huang, K. 3D Object Detection Using Scale Invariant and Feature Reweighting Networks. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 9267–9274.
258. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. Pointpainting: Sequential fusion for 3d object detection. In Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4603–4611.
259. Huang, T.; Liu, Z.; Chen, X.; Bai, X. EPNet: Enhancing point features with image semantics for 3D object detection. In *Computer Vision—ECCV*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 35–52.
260. Paigwar, A.; Sierra-Gonzalez, D.; Erkent, Ö.; Laugier, C. Frustum-pointpillars: A multi-stage approach for 3d object detection using rgb camera and lidar. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual Event, 11–17 October 2021; pp. 2926–2933.
261. Pang, S.; Morris, D.; Radha, H. Fast-CLOCs: Fast camera-LiDAR object candidates fusion for 3D object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 187–196.
262. Wu, X.; Peng, L.; Yang, H.; Xie, L.; Huang, C.; Deng, C.; Liu, H.; Cai, D. Sparse Fuse Dense: Towards High Quality 3D Detection with Depth Completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5408–5417.
263. Zhu, H.; Deng, J.; Zhang, Y.; Ji, J.; Mao, Q.; Li, H.; Zhang, Y. VPFNet: Improving 3D Object Detection with Virtual Point based LiDAR and Stereo Data Fusion. *IEEE Trans. Multimedia* **2022**, *25*, 5291–5304. [CrossRef]
264. Chen, Y.; Li, Y.; Zhang, X.; Sun, J.; Jia, J. Focal Sparse Convolutional Networks for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 5428–5437.
265. Li, Y.; Qi, X.; Chen, Y.; Wang, L.; Li, Z.; Sun, J.; Jia, J. Voxel field fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1120–1129.
266. Liu, Z.; Huang, T.; Li, B.; Chen, X.; Wang, X.; Bai, X. Epnet++: Cascade bi-directional fusion for multi-modal 3d object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 8324–8341. [CrossRef]

267. Wang, M.; Zhao, L.; Yue, Y. PA3DNet: 3-D vehicle detection with pseudo shape segmentation and adaptive camera-LiDAR fusion. *IEEE Trans. Ind. Inf.* **2023**, *19*, 10693–10703. [CrossRef]
268. Mahmoud, A.; Hu, J.S.K.; Waslander, S.L. Dense Voxel Fusion for 3D Object Detection. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 663–672.
269. Li, X.; Ma, T.; Hou, Y.; Shi, B.; Yang, Y.; Liu, Y.; Wu, X.; Chen, Q.; Li, Y.; Qiao, Y.; et al. LoGoNet: Towards Accurate 3D Object Detection with Local-to-Global Cross-Modal Fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 17524–17534.
270. Song, Z.; Zhang, G.; Xie, J.; Liu, L.; Jia, C.; Xu, S.; Wang, Z. VoxelNextFusion: A Simple, Unified and Effective Voxel Fusion Framework for Multi-Modal 3D Object Detection. *arXiv* **2024**, arXiv:2401.02702.
271. Chavez-Garcia, R.O.; Aycard, O. Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 525–534. [CrossRef]
272. Yi, C.; Zhang, K.; Peng, N. A multi-sensor fusion and object tracking algorithm for self-driving vehicles. *Proc. Inst. Mech. Eng. Part D-J. Automob. Eng.* **2019**, *233*, 2293–2300. [CrossRef]

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.