

# Radar-Camera Fusion for Object Detection and Semantic Segmentation in Autonomous Driving: A Comprehensive Review

Shanliang Yao<sup>ID</sup>, Student Member, IEEE, Runwei Guan<sup>ID</sup>, Student Member, IEEE,  
 Xiaoyu Huang<sup>ID</sup>, Student Member, IEEE, Zhuoxiao Li<sup>ID</sup>, Student Member, IEEE,  
 Xiangyu Sha<sup>ID</sup>, Student Member, IEEE, Yong Yue<sup>ID</sup>, Eng Gee Lim<sup>ID</sup>, Senior Member, IEEE,  
 Hyungjoon Seo<sup>ID</sup>, Member, IEEE, Ka Lok Man<sup>ID</sup>, Member, IEEE, Xiaohui Zhu<sup>ID</sup>, Member, IEEE,  
 and Yutao Yue<sup>ID</sup>, Member, IEEE

**Abstract**—Driven by deep learning techniques, perception technology in autonomous driving has developed rapidly in recent years, enabling vehicles to accurately detect and interpret surrounding environment for safe and efficient navigation. To achieve accurate and robust perception capabilities, autonomous vehicles are often equipped with multiple sensors, making sensor fusion a crucial part of the perception system. Among these fused sensors, radars and cameras enable a complementary and cost-effective perception of the surrounding environment regardless of lighting and weather conditions. This review aims to provide a comprehensive guideline for radar-camera fusion, particularly concentrating on perception tasks related to object detection and semantic segmentation. Based on the principles of the radar and camera sensors, we delve into the data processing process and representations, followed by an in-depth analysis and summary of radar-camera fusion datasets. In the review of methodologies in radar-camera fusion, we address interrogative questions, including “why to fuse”, “what to fuse”, “where to fuse”, “when to fuse”, and “how to fuse”, subsequently discussing various challenges and potential research

Manuscript received 12 July 2023; revised 11 August 2023; accepted 16 August 2023. Date of publication 22 August 2023; date of current version 23 February 2024. This work was supported in part by the Suzhou Municipal Key Laboratory for Intelligent Virtual Engineering under Grant SZS2022004, in part by the Key Programme Special Fund of XJTLU under Grant KSF-A-19, in part by the Suzhou Science and Technology Project under Grant SYG202122, in part by the Research Development Fund of XJTLU under Grant RDF-19-02-23, in part by the Jiangsu Engineering Research Center for Data Science and Cognitive Computing, in part by the Jiangsu Industrial Technology Research Institute (JITRI), and in part by Wuxi National Hi-Tech District (WND). (*Corresponding authors: Xiaohui Zhu; Yutao Yue.*)

Shanliang Yao, Runwei Guan, Xiaoyu Huang, Zhuoxiao Li, Xiangyu Sha, and Hyungjoon Seo are with the Faculty of Science and Engineering, University of Liverpool, L69 3BX Liverpool, U.K. (e-mail: shanliang.yao@liverpool.ac.uk; runwei.guan@liverpool.ac.uk; x.huang42@liverpool.ac.uk; zhuoxiao.li@liverpool.ac.uk; sgxsha2@liverpool.ac.uk; hyungjoon.seo@liverpool.ac.uk).

Yong Yue, Eng Gee Lim, Ka Lok Man, and Xiaohui Zhu are with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China (e-mail: yong.yue@xjtu.edu.cn; enggee.lim@xjtu.edu.cn; ka.man@xjtu.edu.cn; xiaohui.zhu@xjtu.edu.cn).

Yutao Yue is with the Institute of Deep Perception Technology, Jiangsu Industrial Technology Research Institute, Wuxi 214028, China, also with the Xi'an Jiaotong-Liverpool University-JITRI Academy of Industrial Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China, and also with the Department of Mathematical Sciences, University of Liverpool, L69 3BX Liverpool, U.K. (e-mail: yueyutao@idpt.org).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIV.2023.3307157>.

Digital Object Identifier 10.1109/TIV.2023.3307157

directions within this domain. To ease the retrieval and comparison of datasets and fusion methods, we also provide an interactive website: <https://radar-camera-fusion.github.io>.

**Index Terms**—Autonomous driving, radar-camera fusion, object detection, semantic segmentation.

## I. INTRODUCTION

AUTONOMOUS driving has excellent potential in mitigating traffic congestion and improving driving safety. Perception, akin to eyes in autonomous driving, constitutes the foundation for successive functions, such as motion prediction, path planning and maneuver control [1], [2]. To achieve optimal accuracy and robustness of the perception system, various sensors are integrated into autonomous vehicles, allowing for the utilization of their complementary and redundant characteristics [3], [4]. However, which sensors to choose and how to fuse the data between different sensors have emerged as challenging issues requiring further exploration.

Given the rich semantic information that can be perceived, cameras are widely utilized in autonomous driving for object detection, segmentation and tracking. LiDARs calculate the distance to surrounding objects by measuring the time difference of the laser beam from emission to reception via the objects. The denser the laser layers emitted by a LiDAR sensor, the clearer an object's three-dimensional (3D) contour. These complementary features provided by cameras and LiDARs have made LiDAR-camera sensor fusion a hot topic in recent years, and achieved high accuracy in two-dimensional (2D) and 3D object detection [6], [7], [8], [9], semantic segmentation [10], [11] and object tracking [12], [13]. Despite their strengths, both LiDARs and cameras suffer from the same defect of being sensitive to adverse weather conditions (e.g., rain, fog, snow) that can significantly diminish their field of view and object recognition capabilities [14]. Moreover, the high cost of LiDAR products has brought certain difficulties in promoting their widespread adoption [15].

Compared to LiDARs and cameras, radars exhibit superior effectiveness under challenging lighting and weather conditions [16], [17]. Radars can also deliver accurate velocity estimation for all detected objects depending on the Doppler effect

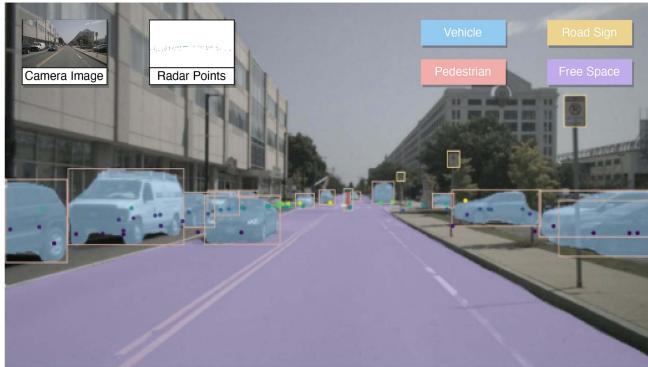


Fig. 1. Common scenario of object detection and semantic segmentation in autonomous driving. Boxes and masks represent the results of detection and segmentation, respectively. Dots indicate the location of each radar point, and the darker the dot, the closer the distance to the ego-vehicle. Image is generated from the nuScenes [5] dataset.

without requiring any temporal information [18]. With these characteristics, radars are widely used in Advanced Driving Assistance Systems (ADAS) applications, including collision avoidance, Adaptive Cruise Control (ACC), Lane Change Assist (LCA) and Automatic Emergency Braking (AEB). As depicted in Fig. 1, the integration of radar and camera data in sensor fusion enables a comprehensive perception of the surrounding environment in terms of outlines, colors, textures, ranges, and velocities. Moreover, the fusion system can operate continuously throughout the day regardless of weather and lighting conditions.

Although radar sensors are popularly applied to vehicles, few studies focus on data fusion from radars and cameras. One reason for this is the limitations of radar output data, such as low resolution, sparse point clouds, uncertainty in elevation and clutter effects. Another reason is that up to now, the datasets containing both radar and camera data for autonomous driving applications are insufficient, making it challenging for researchers to conduct in-depth analysis. Additionally, applying or adapting existing LiDAR-based algorithms to radar point clouds yields poor results due to inherent differences of point clouds between the LiDAR sensor and radar sensor [18]. Radar point clouds are significantly sparser than their LiDAR counterparts, making it inefficient to extract objects' geometry information using LiDAR-based algorithms. Although Radar Cross Section (RCS) values in the radar sensor indicate the reflective intensity from the surface of an object, they are easily affected by numerous factors and cannot be used singularly to determine the classification of the target. In addition, though aggregating multiple radar frames enhances the density of the point clouds, it also causes a delay to the whole system. In summary, radar-camera fusion perception is significant in autonomous driving as well as challenging in implementation.

#### A. Related Surveys

Most sensor fusion surveys focus on LiDAR-camera [2], [19], [20], [21], or the broader field of multi-sensor fusion, including LiDAR, camera, radar and other sensors [3], [21], [22], [23]. Specifically, in multi-sensor fusion surveys, LiDARs

and cameras are still the main research objectives. For example, Feng et al. [3] conducted a comprehensive survey on deep multi-modal object detection and semantic segmentation for autonomous driving. However, this survey mainly concentrates on fusion methods based on LiDARs and cameras, and briefly mentions some studies combining camera images and radar data.

To the best of our knowledge, [24] is the only survey that primarily focuses on radar-camera fusion for object detection in autonomous driving. However, it does not cover the radar-camera fusion dataset or the semantic segmentation task.

#### B. Contributions

With the limited focus on radar-camera fusion in existing surveys, it is challenging for researchers to gain an overview of this emerging research field. Our survey attempts to narrow this gap by providing a comprehensive review of radar-camera fusion in autonomous driving. The contributions of our review are summarized as follows:

- To the best of our knowledge, this is the first survey focusing on two fundamental perception problems for radar-camera fusion, namely, object detection and semantic segmentation.
- We present an up-to-date (2019–2023) overview of radar-camera fusion datasets and algorithms, and conduct in-depth research on “why to fuse”, “what to fuse”, “where to fuse”, “when to fuse”, and “how to fuse”.
- We analyze the critical challenges and open questions in radar-camera fusion, and put forward potential research directions.
- We provide an interactive and updated website for better retrieving and comparing the fusion datasets and methods.

## II. BACKGROUND

This section provides background information on radar-camera fusion in autonomous driving. We first introduce the working principles, sensor characteristics and data representations of the radar and camera sensors. By comparing the characteristics of the two sensors, we aim to demonstrate the importance of radar-camera fusion. Subsequently, as the perception module leverages data from specific sensors to understand the surroundings, we present basic concepts and highlight representative algorithms for two fundamental and crucial perception tasks: object detection and semantic segmentation.

#### A. Radar Sensors

*1) Working Principles:* Radar is the abbreviation of Radio Detection And Ranging, which calculates the range and velocity of the target by transmitting radio waves and receiving the reflected waves from the target [25]. In autonomous driving applications, radar typically refers to the Millimeter-Wave (MMW) radar that works in the millimeter wave band with a wavelength of 1–10 mm and frequency of 76–81 GHz. Specifically, the radar equipped in the forward direction, as well as the four corner directions, is usually a Multiple-Input Multiple-Output (MIMO) radar, while the radar on the roof is typically a mechanical

rotating radar. A MIMO radar utilizes multiple antennas and transmitters to simultaneously transmit and receive multiple signals with different frequencies. In contrast, a mechanical rotating radar operates with a single antenna that physically rotates to emit radar signals in different directions. With multiple antennas and beamforming capabilities, a MIMO radar achieves higher spatial resolution and interference reduction compared to mechanical rotating radar. While a mechanical rotating radar provides better coverage and is simpler in implementation.

Based on the Time of Flight (TOF) principle, the radar sensor calculates the range from the object by the time difference between the transmitted and reflected signals. Based on the Doppler principle, when there is a relative movement between the emitted electromagnetic wave and the detected target, the frequency of the returned wave differs from that of the emitted wave. Thus, the target's relative velocity to the radar can be measured using this frequency difference. Leveraging the array signal processing method, the azimuth angle is calculated using the signal's phase difference between parallel antennas. Since the receivers of traditional 3D (range, Doppler velocity and azimuth angle) radar sensors are only lined up in a 2D direction, targets are only detected in 2D horizontal coordinates without vertical height information. Recently, with advancements in radar technologies, 4D (range, Doppler velocity, azimuth angle and elevation angle) radar sensors have been developed with antennas arranged horizontally and vertically, enabling the measurement of elevation information. In addition, 4D is often represented as x, y, z coordinates and Doppler velocity.

*2) Sensor Characteristics:* In addition to the ability to measure range, Doppler velocity and azimuth angle, electromagnetic waves in the millimeter wave band have low atmospheric attenuation and better penetration of rain, smoke and dust [26]. These characteristics make the radar sensor work all day regardless of severe weather conditions. However, radar sensors still have certain limitations. They exhibit low angular resolution and cannot distinguish between closely located objects. The point clouds generated by radars are sparsely distributed, with only a few points on a pedestrian and a dozen points on a car. These points cannot adequately outline an object's contours, making it challenging to extract the geometric information [27], [28]. Doppler radar measurements have a limitation in that they only provide the radial component of velocity. The lack of tangential velocity makes it difficult to estimate the accurate velocity of an object in dynamic scenes [29], [30]. Besides, data produced by radars are noisy, which may arise from diverse sources such as multipath interference, electrical interference and equipment imperfections [31], [32]. Such noise reduces the precision and reliability of radar data, while also increasing the probability of false detections. Furthermore, radars are weak in the perception of stationary obstacles. Moving targets can be distinguished from the surrounding scene in one dimension of range and velocity. However, radars are highly sensitive to metal, often resulting in strong reflections from stationary objects such as manhole covers on the ground. Strong reflections from stationary objects are not filtered, resulting in a lack of detecting stationary obstacles.

*3) Data Representations:* As depicted in Fig. 2, the raw output of a radar sensor is the **ADC signal**, which refers to

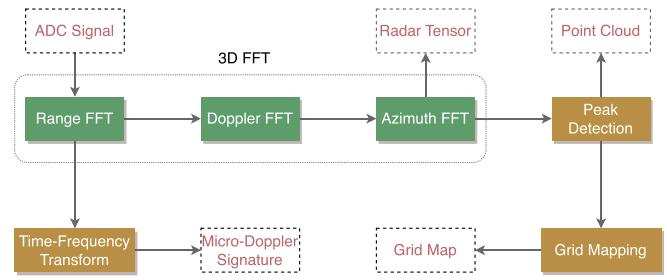


Fig. 2. Generation process of radar data representations.

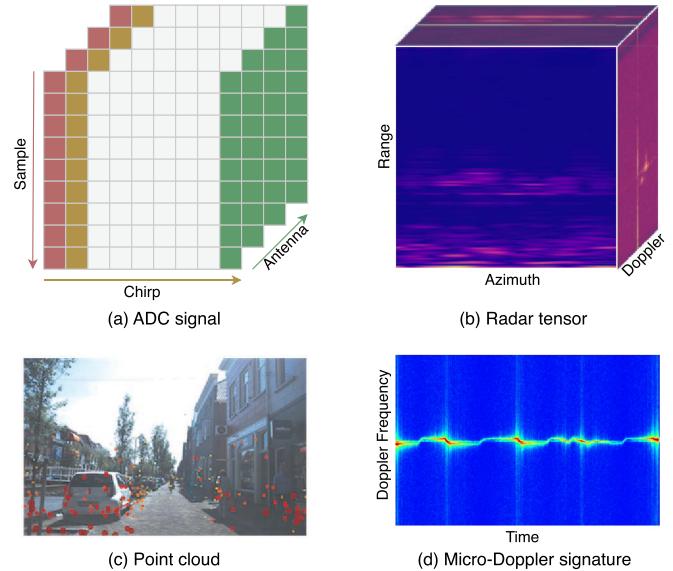


Fig. 3. Radar data representations. (a) ADC signal in the format of Simple-Chirp-Antenna tensor. (b) Radar tensor represented by a 3D Range-Azimuth-Doppler tensor. Image is generated from the CARRADA [33] dataset. (c) Point cloud projected on a 2D image plane. Image is generated from the View-of-Delft [34] dataset. (d) Micro-Doppler signature showing a pedestrian walking. Image is generated from the Open Radar Datasets [35].

the output signal of an Analog-to-Digital Converter (ADC). At this stage, the signal lacks spatial coherence between the values as all the information exists in the time domain [36]. Besides, the signal is represented in complex value which contains real part and imaginary part [37]. To represent the ADC signal in a more structured form, it is usually transformed into a 3D Sample-Chirp-Antenna (SCA) tensor, as illustrated in Fig. 3(a). Some researchers apply 3D Fast Fourier Transform (FFT) along the sample, chirp and antenna dimensions to get an image-like representation named **radar tensor** (Fig. 3(b)), describing the spatial pattern of the received echo [16], [17]. At this stage, the non-coherent combination (e.g., norm calculation) converts ADC signals composed of complex values to radar tensors that consists of real values. With these three features in Range-Azimuth-Doppler (RAD) coordinates, two forms of radar tensors are formed: one is in 2D including the Range-Azimuth (RA) tensor, Range-Doppler (RD) tensor and Azimuth-Doppler (AD) tensor; the other is the whole 3D RAD tensor, with each side consisting of a 2D tensor. Furthermore, peak detection is

carried out on the radar tensor to filter out clutter, resulting in a sparse point-like representation called the **point cloud**, as depicted in Fig. 3(a). The point cloud provides a spatially intuitive representation better suited for visualization and interpretation, yet it can not accurately indicate the outline information [38], [39]. Constant False Alarm Rate (CFAR) is the most commonly used method for peak detection, which enables the radar system to automatically adjust its sensitivity level to changes in the strength of external interference, thereby maintaining a steady false alarm rate [40], [41]. By applying grid mapping methods to point clouds accumulated over a given period, a **grid map** for identifying static objects is generated. There are two main grid maps: one is the occupancy-based grid map [42], [43], [44], which represents the obstacles and free-space derived from the radar data; the other is the amplitude-based grid map [42], [45], which displays the RCS values for each cell. However, it is essential to note that the sparsity of the point clouds still influences the accuracy of detection and segmentation performed on the grid map. In addition, some researchers perform Time-Frequency transform after the Range-FFT to obtain the **micro-Doppler signature**, which is utilized to recognize objects with tiny motion features [35], [46]. As exemplified in Fig. 3(d), the Doppler frequency of a pedestrian walking shows a periodic variation. This representation enables not only the distinction of different object categories (e.g., pedestrians, bicycles and vehicles), but also the recognition of complex object behaviors, such as gait and gesture recognition.

### B. Camera Sensors

1) *Working Principles*: The camera sensor usually consists of a lens, an image sensor, an Image Signal Processor (ISP) and an Input/Output (I/O) interface [47]. The lens collects the light reflected from the target and converges it to the image sensor. Then, the image sensor converts light waves into electrical signals and converts electrical signals to digital values via an on-chip ADC. After that, the ISP performs post-processing (e.g., noise reduction) and converts the digital values into a format of RGB data for images or videos. Finally, the image data is transferred and displayed via the I/O interface.

2) *Sensor Characteristics*: Cameras capture the rich appearance features of the objects, including colors, shapes and textures. After learning from neural networks, these features can be utilized to identify obstacles, including vehicles, pedestrians, bicycles and various traffic lights. However, cameras are passive sensors, indicating that the formation of an image requires incident light intake. When the light intake is adversely affected, such as insufficient light at night, extreme weather, water droplets or dust sticking to the lens, the imaging results will be unclear, and object detection performance may be significantly affected [48]. Besides, in autonomous driving, it is crucial to identify the distances of obstacles ahead. However, a target in three dimensions in the world coordinate system becomes a 2D target in the image coordinate system after being imaged by the camera sensor, resulting in a loss of distance information.

3) *Data Representations*: Raw data representation is the uncompressed and unprocessed format captured by the camera

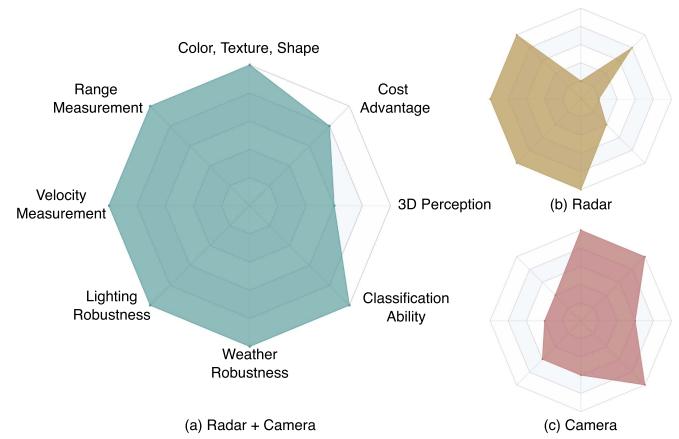


Fig. 4. Comparison of radar and camera characteristics. In these charts, each characteristic is plotted along one of the line segments radiating from the central point, with closer proximity to the vertex representing higher quality for that characteristic. (a) Fusion characteristics of radar and camera. (b) Radar characteristics. (c) Camera characteristics.

sensor. It contains all the radiance information that hits each pixel on the camera sensor during image exposure [49], [50]. After post-processing, a data representation named **RGB image** is generated, which illustrates an image as a grid of pixels, with each pixel containing a value for each of the red, green and blue color channels. In addition, some modern cameras applied in autonomous driving are able to generate specific data representations. For instance, a depth camera produces the **depth map** representation, providing information about the distance to each pixel in the scene [51]. Relying on flood-light flash laser sources, infrared cameras output the **infrared image** representation, which is able to render perception results in adverse weather and low-light conditions [52], [53]. Event cameras are bio-inspired vision sensors that generate an **event image** representation pertaining to pixel-level changes in brightness. With sub-millisecond latency, high-dynamic range, and robustness to motion blur, event cameras present considerable potential for real-time detection and tracking of objects in time-critical scenarios [54].

### C. Comparison of Radar and Camera Sensors

Through our extensive review, a clear and concise figure is designed to compare the characteristics of the two sensors, shown in Fig. 4. Specifically, the radar sensor is an active sensor and measures various information, including ranges, velocities and azimuth angles [55]. Nowadays, radars equipped in driver-assistance systems can detect up to 300 meters, with a 140° horizontal field of view and a less than 1° angular resolution [56], [57]. In addition, the radar sensor is robust to darkness and extreme weather conditions, allowing it to work throughout the day. The camera sensor is a passive sensor that provides colors, textures and shapes of objects. With a resolution of up to 2 K, the camera sensor performs much better in classification than the radar sensor. As far as system cost goes, both radars and cameras are relatively cost-effective and are mass-applied in vehicles.

To sum up, both the radar and camera have their strengths and weaknesses, and they cannot be substituted for each other. The most effective way to ensure adequate information acquisition is mutual integration. Based on their respective characteristics, complementary advantages can improve scenario understanding performance. In addition, when one of the sensors fails, the remaining one can continue working, thus increasing the reliability of the autonomous driving system. Hence, the fusion of radar and camera sensors is critical for perception accuracy and robustness in autonomous driving.

#### D. Perception Tasks

1) *Object Detection*: The object detection task involves identifying a particular object in a camera image or a radar scan, locating its position and determining its category. Generally speaking, researchers use a rectangular or cubic bounding box to encompass the object. As there is no depth channel in 2D object detection, the rectangular bounding box is expressed as  $(x, y, h, w, c)$ , where  $(x, y)$  is the bounding box center,  $h$  and  $w$  are the height and width of the bounding box, and  $c$  is the class of the object. While the cubic bounding box for 3D object detection is described as  $(x, y, z, h, w, l, \theta, c)$ , where  $(x, y, z)$  represents the center of the 3D bounding box,  $h$ ,  $w$  and  $l$  are the height, width and length of the bounding box,  $\theta$  is the object's orientation, and  $c$  is its class. Bird's Eye View (BEV) object detection is a specialized form of 3D object detection focusing on detecting objects from a top-down perspective. In this approach, height information is typically discarded, and objects are represented as 2D bounding boxes on the ground plane.

a) *Camera-based Object Detection*: In autonomous driving, camera-based object detection approaches have been widely used in detecting vehicles [58], [59], pedestrians [60], [61], [62], traffic lights [63], [64], [65], [66], and traffic signs [67], [68], [69]. According to different training steps, CNN-based object detection algorithms can be classified into two-stage and one-stage. Two-stage detection algorithms (e.g., R-CNN [70], SPPNet [71], Fast R-CNN [72], Faster R-CNN [73], FPN [74]) segregate the detection problem into two stages: the first step is to generate region proposals, and the second step is to refine the position and predict the classification of each object. Experimental results of these algorithms are high in precision and recall, but relatively slow in time. Without the region proposal generation phase, one-stage detection algorithms simultaneously predict the bounding boxes and the probability of classes within these boxes. Thus, one-stage detection algorithms are commonly faster than two-stage detection algorithms, but lower in accuracy. Some highly representative one-stage object detectors include the YOLO series [75], [76], [77], [78], [79], [80], [81], [82], SSD [83] and RetinaNet [84].

Exploiting the self-attention mechanism that enables the model to model the contextual features and their correlation, transformer-based methods have emerged as a recent breakthrough compared to CNN-based detectors. Some representative pure transformer detectors include DETR [85], Deformable DETR [86], RT-DETR [87], WB-DETR [88], Swin [89] and

YOLOS [90]. Additionally, plenty of studies have endeavored to accelerate the conventional transformer block by the combination of convolution and self-attention, aggregating the advantages of both CNN and transformer, as exemplified by Conformer [91], EdgeViTs [92], MobileViT [93], ViTAE [94] and Visformer [95].

b) *Radar-based Object Detection*: Radar-based object detection approaches have been widely used in detecting vehicles [16], [17], pedestrians [96], [97] and static objects [98]. As radar tensors are image-like representations, researchers generally utilize image-based networks (e.g., ResNet [99], Faster R-CNN [73], YOLOv4 [78]) to perform object detection on 2D RA tensors [17], [100], [101], 2D RD tensors [102], [103] and 3D RAD tensors [16], [96], [97], [104], [105]. Unlike images, radar tensors lack a physical interpretation, thereby presenting difficulties in translating the learned features from image-based algorithms to radar data. Furthermore, applying algorithms to radar tensors in real-time applications poses challenges due to the high-dimensional nature of radar tensors and the presence of noise, interference and clutter.

For radar data in the format of point clouds, various types of point-based networks are utilized to detect objects. Point-wise methods [106], [107], [108], [109], [110] directly operate on the raw point clouds and leverage LiDAR-based algorithms, such as PointNet [111], PointNet++ [112] and Frustum PointNets [113], to classify the points into distinct object classes. Grid-based methods [108], [114], [115], [116] map the 3D point clouds into grid-like structures, such as 2D image planes or 3D voxel grids. Subsequently, object detection algorithms (e.g., YOLOv3 [77], VoxelNet [117]) are applied to the grid representation to identify objects. The grid-based approaches demonstrate efficiency in handling large datasets and are frequently employed in real-time applications. Graph-based methods (e.g., RadarGNN [39], Radar-PointGNN [98]) in radar point cloud object detection employ the Graph Neural Network (GNN), where the points serve as nodes, and the relationships between the points are modeled as edges in the graph. Leveraging graph structures and algorithms, these methods effectively capture the spatial relationships and contextual information among the points, leading to improved detection performance compared to traditional point-wise methods. However, the construction and feature extraction of graphs are complex and computationally intensive, especially when handling large-scale point clouds.

2) *Semantic Segmentation*: Semantic segmentation involves clustering the basic components of input data into different semantically relevant regions. Essentially, it refers to assigning selected labels from a pre-defined set  $Y = \{y_1, y_2, \dots, y_k\}$  to each pixel in an image-based dataset  $D_i = \{d_1, d_2, \dots, d_m\}$  or each point in a point-based dataset  $D_p = \{d_1, d_2, \dots, d_n\}$ .

a) *Camera-based Semantic Segmentation*: The technique of camera-based semantic segmentation finds widespread application in the fields of free-space segmentation [118], [119], [120], [121], [122], [123], lane segmentation [120], [121], [123], [124], [125], [126], [127] and obstacle segmentation [128], [129], [130] in autonomous driving. Fully Convolutional Network (FCN) [131] is a milestone in semantic segmentation as it enables end-to-end training of deep networks for this task.

However, due to its failure to account for global contextual information, the obtained segmentation results tend to be coarse. Therefore, the encoder-decoder architecture has emerged to address this shortcoming, represented by SegNet [132], U-Net [133] and HRNet [134]. The encoder-decoder architecture typically uses an image classification network as its encoder, gradually reducing the spatial dimensions of the pooling layer. Meanwhile, the decoder gradually restores the details and spatial dimensions for segmentation purposes.

However, for encoder-decoder architectures, high-resolution representations are lost during the encoding process, reducing fine-grained information within the image. Dilated (or “atrous”) convolution structure is created to avoid decimating the input’s resolution by adding a dilation rate to standard convolutions. This architecture enlarges the receptive field without increasing the parameters and avoids the loss of information caused by repeated pooling. Some notable examples of representative networks which apply dilated convolution structure include DeepLab series [135], [136], [137], [138], ENet [139], PSPNet [140], DUC-HDC [141] and DenseASPP [142].

CNNs require multiple decoder stacks to map high-level features to the original spatial resolution. In contrast, transformer-based models can be graciously combined with a lightweight transformer decoder for segmentation mask prediction due to their global modeling capability and resolution invariance. Recently, transformer-based segmentation models (e.g., SETR [143], Segmenter [144], SegFormer [145], Lawin [146] and MaskFormer [147]) extract global contextual features based on self-attention and achieve remarkable results.

*b) Radar-based Semantic Segmentation:* Radar-based semantic segmentation is applied in the fields of vehicle segmentation [38], [148], pedestrian segmentation [44], [149], free-space segmentation [36], [150] and static object segmentation [151], [152] in autonomous driving. Similar to object detection methods in radar-based applications, network architectures vary depending on specific radar representations. These architectures also incorporate algorithms adapted from the image and point cloud domains to enable efficient processing and analysis of radar data. Segmentation on radar tensors refers to the process of dividing the tensor into discrete regions or segments based on specific criteria or properties. The goal is to identify and label different parts or objects in the radar RA tensor [153], RD tensor [154] and RAD semantic segmentation [36], [105], [148], [150], [155], enabling a more comprehensive understanding of the scene. CNN architectures like DeepLabv3+ [138] and U-Net [133] possess the ability to extract intricate features and relationships directly from the radar tensor data, thereby facilitating effective segmentation tasks.

For segmentation on radar point clouds, conventional CNN algorithms (e.g., PointNet and PointNet++) can effectively capture the spatial relationships and semantics of individual radar points to classify them into different categories or segments. These algorithms are widely utilized in point-wise semantic segmentation [38], [106], [156], [157], [158], [159] and grid-based methods [44], [151], [152], [153], [160]. However, the initial data transformation involved in these approaches may

result in information loss and sparsity in the data representation. Recent point transformer networks (e.g., Gaussian Radar Transformer (GRT) [161]) enhance performance for 3D point cloud understanding by elaborating the attention mechanism, which is able to capture complex structures in sparse point clouds. In addition to basic semantic segmentation, instance-based segmentation methods [159], [162] not only classify each point in the radar point cloud but also group nearby points together into instances.

### III. FUSION DATASETS

High-quality and large-scale data are fundamental for deep learning-based perception algorithms in autonomous driving. Datasets containing data from LiDARs and cameras, such as KITTI [163], Oxford RobotCar [164], ApolloScape [165], and Waymo [166], have been widely used for LiDAR-camera fusion in autonomous driving. As radar research continues in-depth, dozens of radar and camera datasets have been released in recent years. In this section, we analyze and summarize these datasets designed explicitly for tasks related to object detection and semantic segmentation. Fig. 5 presents clear statistics of radar-camera fusion datasets with their radar representations and dataset sizes. We also provide a table for retrieval and comparison of different datasets (see Table II).

#### A. Dataset Tasks

According to the dimension of bounding boxes and masks, datasets that incorporate radar and camera modalities in object detection and semantic segmentation are categorized into four groups:

- *2D object detection:* SeeingThroughFog [53], CAR-RADA [33], Zendar [167], RADIATE [168], AIO-Drive [169], CRUW [170], RaDICaL [171], RADDet [97], FloW [172], RADlal [105], VoD [34], Boreas [173] and WaterScenes [174];
- *3D object detection:* nuScenes [5], Astyx [175], SeeingThroughFog [53], AIODrive [169], VoD [34], TJ4DRadSet [176], K-Radar [177] and aiMotive [178];
- *2D semantic segmentation:* CARRADA [33], RadarScenes [179] and RADlal [105];
- *3D semantic segmentation:* HawkEye [180].

Regarding dataset tasks, most datasets are oriented toward object detection, whereas comparatively fewer datasets are employed for semantic segmentation tasks. Notably, CAR-RADA [33], RadarScenes [179] and RADlal [105] can be applied to both object detection and semantic segmentation tasks. As to those datasets that contribute to multiple tasks, nuScenes [5] is the most widely used dataset in radar-camera fusion algorithms, supporting tasks of detection, tracking, prediction and localization. In addition to object detection, RADIATE [168] involves object tracking, scene understanding and SLAM tasks. Moreover, datasets like Zendar [167] and Boreas [173] are available for localization and odometry.

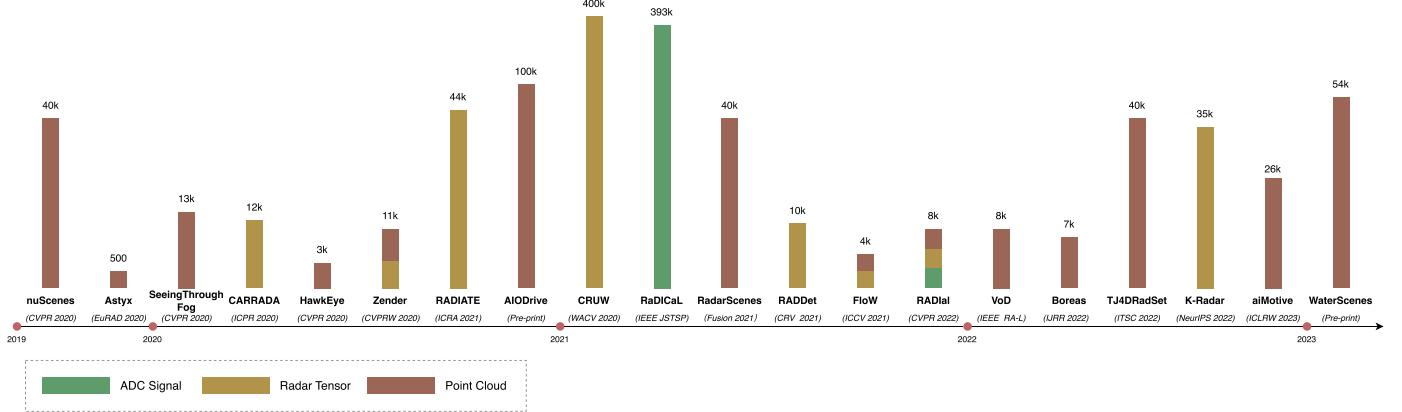


Fig. 5. Overview of radar-camera fusion datasets. Color of the bar denotes different radar representations, and height of the bar represents the frame of each dataset.

### B. Sensing Modalities

For radar-camera fusion datasets in object detection and semantic segmentation tasks, the data produced by the camera sensor is either a single image or a video over a while, both of which are essentially 2D images. In comparison, data produced by the radar sensor is rich in representations, which can be grouped into ADC signal, radar tensor, and point cloud according to the stages of data processing.

1) *ADC Signal*: As the raw data produced by radar sensors, ADC signals retain all semantic information and can be highly valuable in deep learning applications. Up to now, only two radar-camera fusion datasets provide raw ADC signal data: RaDICal [171] and RADIal [105]. RaDICal [171] is the first dataset providing raw ADC signal data, specialized for object detection tasks involving pedestrians and vehicles. The authors encouraged researchers to further design their own processing methods by providing the raw radar measurements. RADIal [105] is the richest dataset regarding radar data representations, offering not only ADC signals, but also processed data after ADC signals, including radar tensors and point clouds.

2) *Radar Tensor*: After multiple FFTs, radar tensors are obtained from ADC signals. They can be classified into three categories: 2D tensors (e.g., RADIATE [168], CRUW [170], FloW [172]), 3D tensors (e.g., CARRADA [33], Zender [167], RADDet [97], RADIal [105]) and 4D tensors (e.g., K-Radar [177]). Both RADIATE [168] and CRUW [170] are in range-azimuth coordinates, presenting the BEV position of objects, while the FloW [172] dataset is in range-Doppler coordinates, illustrating the relationship between range and Doppler velocity of each object. CARRADA [33] is the first dataset that combines synchronized stereo RGB images and 3D radar RAD tensors in autonomous driving. As far as we are aware, K-Radar [177] is the only dataset containing 4D radar tensors, with full information on range, Doppler, azimuth and elevation.

3) *Point Cloud*: Compared to radar tensors, point clouds serve as a lighter and more intuitive representation of objects. They are also the format of data output from commercial radars. Conventional 3D radars produce sparse point clouds, such as data in nuScenes [5], Zender [167], SeeingThroughFog [53],

HawkEye [180], AIODrive [169], RADIal [105], FloW [172], RadarScenes [179] and aiMotive [178] dataset. In recent years, the radar sensor has advanced from 3D to 4D with improvements in resolution and elevation measurement capabilities. Consequently, public 4D radar-camera fusion datasets are emerging, with examples such as Astyx [175], VoD [34], TJ4DRadSet [176] and WaterScenes [174]. Although Astyx [175] is the first 4D point cloud dataset, it is limited by the data size, containing only 500 frames. VoD [34] and TJ4DRadSet [176] datasets are improved in terms of data categories and data size, with the former consisting of 13 types and 8,693 frames, and the latter containing eight types and 40 k frames. Meanwhile, these two datasets also contain simultaneous LiDAR data, facilitating comparison between the 4D radar point clouds and LiDAR point clouds.

### C. Dataset Categories

For autonomous driving, it is critical to identify Vulnerable Road Users (VRU) on roads. Therefore, the most common categories in these datasets are pedestrians, bicycles, and cars [33], [34], [168], [169], [170], [175], [179]. Datasets such as nuScenes [5], AIODrive [169], RadarScenes [179], VoD [34] and aiMotive [178] have studied more than ten categories. nuScenes [5] provides precision in its classifications, with 23 object categories refining certain ambiguous categories. For instance, the pedestrian category is sub-categorized into groups such as adult and child, while the vehicle category is subclassed into the car, ambulance, police, motorcycle, trailer and truck. Except for the category of pedestrian, RADIATE [168] and RadarScenes [179] include a class called pedestrian group. AIODrive [169] and VoD [34] classify stationary objects on the roadside, such as building, road, wall, traffic sign, unused bicycle and bicycle rack. Furthermore, apart from objects on road surfaces, FloW [172] is a floating waste dataset containing the category of bottle that can be utilized for Unmanned Surface Vehicles (USVs) on water surfaces. WaterScenes [174] contains more objects of interest on water surfaces, including static objects such as piers and buoys, and dynamic objects such as ships, boats, vessels, kayaks, and sailors aboard these surface vehicles.

In addition to the primary object categories, some specific attributes of the object are also labeled in some datasets. Examples can be found in nuScenes [5], in which vehicles are labeled as moving, stopped or parked, while pedestrians are marked as moving or standing. Besides, in VoD [34] dataset, two types of occlusions (“spatial” and “lighting”) and attributes related to an object’s activity (“stopped”, “moving”, “parked”, “pushed” and “sitting”) are also annotated. All these specific attributes are essential for scene understanding.

#### D. Dataset Size

The reviewed datasets differ significantly in size, ranging from 500 to 1.4 million frames. nuScenes [5] is the largest dataset with 1.4 million images, radar frames and object bounding boxes in 40 k keyframes. These data frames are split from 15 hours and 242 kilometers of driving data. On the other hand, Astyx [175] provides only 500 frames, containing around 3 k labeled 3D object annotations. Others like CRUW [170], CARRADA [33], RADIATE [168], AIODrive [169], SeeingThroughFog [53], CRUW [170], RADDet [97] and RADlal [105] all contribute hundreds of thousands of frames.

In addition to data in frames, some datasets deliver videos for researchers to split keyframes and conduct further research on videos. For example, RadarScenes [179] offers 158 individual sequences with a total length of over four hours. Similarly, CRUW [170] and CARRADA [33] datasets provide videos of 3.5 hours and 21.2 minutes, respectively.

#### E. Recording Scenarios

A rich data collection environment is crucial for training robust models in autonomous driving. Generally, datasets for autonomous driving are collected in road environments like urban streets, country roads, highways and parking lots, which are all represented in datasets like CARRADA [33], RADIATE [168], RadarScenes [179], RADlal [105], and K-Radar [177]. However, it is not enough to collect data in common areas. nuScenes [5], Zendar [167], SeeingThroughFog [53], AIODrive [169] and CRUW [170] involve dense traffic and challenging driving situations, including urban roads, residential areas and industrial areas. Moreover, RadarScenes [179] offers data for selected particular scenarios, such as T-junctions, commercial areas and road works. All the datasets mentioned above are from outdoor environments, and as to indoor scenarios, HawkEye [180] and RaDICaL [171] are collected in indoor parking garages. These indoor environments present unique challenges and can help advance research in indoor autonomous vehicle navigation.

In terms of weather and lighting conditions, related data can be found in nuScenes [5], SeeingThroughFog [53], CARRADA [33], RADIATE [168], K-Radar [177], AIODrive [169], CRUW [170], TJ4DRadSet [176], aiMotive [178] and WaterScenes [174]. In particular, SeeingThroughFog [53] focuses on extreme weather conditions, like fog, snow and rain. This dataset highlights the importance of data fusion and the redundancy of multiple sensors in adverse weather environments. In addition to adverse weather conditions, RADIATE [168], K-Radar [177]

and aiMotive [178] involve conditions for the night. AIODrive [169], CRUW [170] and TJ4DRadSet [176] supply data in specific vision-fail scenarios, such as darkness, bright light and blur, where the images are pretty bad in quality. These datasets provide valuable information about how autonomous driving technology operates in low visibility and low light scenarios. Boreas [173] is a dataset involving data taken from specific routes through one-year repeat collection. It can be leveraged to study the effects of seasonal variation on self-localization and object detection.

Since it is time-cost and resource-cost for data collection on roads, some researchers have adopted simulated data to generate datasets. In HawkEye [180], raw low-resolution heatmaps are transformed into high-frequency shapes. Additionally, the authors developed a data synthesizer to simulate radar signals from the created 3D point reflector models of cars. Meanwhile, Weng et al. [169] used Carla [181] simulator to create different driving scenarios with various sensors. With the annotation data generated by combining and post-processing Carla outputs, they presented AIODrive [169], a large-scale synthetic dataset for all mainstream perception tasks.

## IV. FUSION METHODOLOGIES

In this section, we delve into the methodologies of radar-camera fusion related to object detection and semantic segmentation tasks, starting with “why to fuse”, that is, the purpose and advantages of fusion. Subsequently, we analyze “what to fuse”, covering diverse representations of both radar and camera modalities implicated in fusion. Next, we investigate “where to fuse”, describing the coordinate relations between the two modalities before fusion. In the section on “when to fuse”, we categorize the fusion levels and illustrate their differences. After that, we explore the specifics of “how to fuse”, including temporal-spatial synchronization and fusion operations. Regarding network architectures for fusion, we categorize them into two architectures: point-based and tensor-based, followed by a more detailed classification of each category and the main ideas in these architectures. Finally, in the model evaluations section, we review various evaluation metrics and assess the performance of popular methods. An overview of radar-camera fusion methodologies containing the questions and answers is demonstrated in Fig. 6.

### A. Why to Fuse

The integration of radar and camera sensors for object detection and semantic segmentation is intended to enhance the perception outcomes by capitalizing on the advantages of both sensing modalities. As illustrated in Fig. 4 in Section II-C, the combination of a radar sensor and a camera sensor enables the measurement of rich object attributes such as color, shape, range, and velocity. In addition, with the ability to perceive in darkness and adverse weather conditions, the fusion of radar and camera can work all day for autonomous driving vehicles. For autonomous driving tasks, object detection and segmentation results obtained from radar-camera fusion can also assist in object tracking [182], [183], providing accurate environment

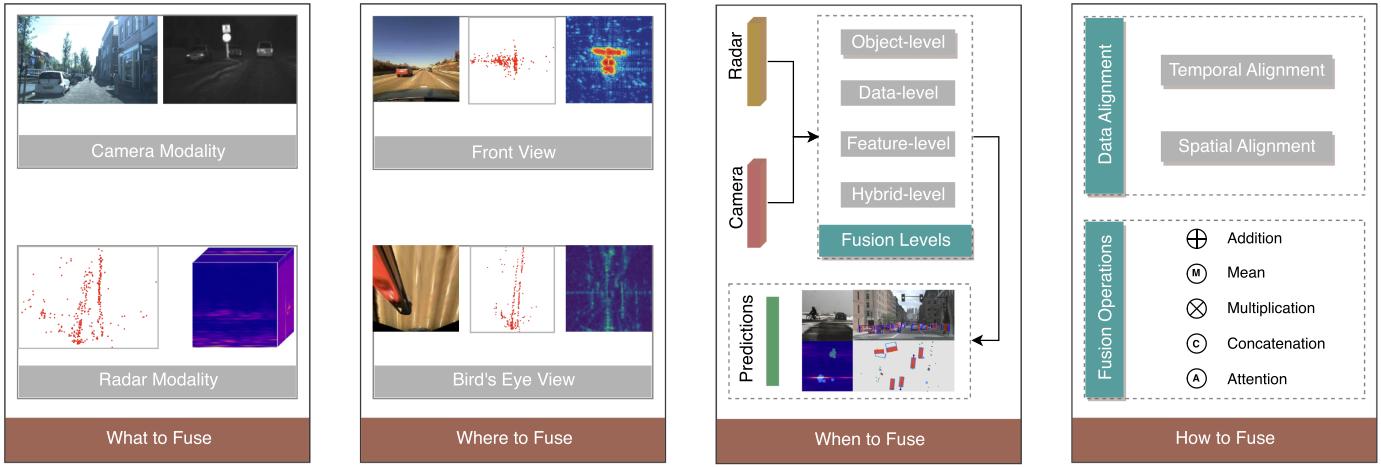


Fig. 6. Methodologies of radar-camera fusion. (a) “What to fuse”: the input modalities for radar-camera fusion, such as RGB images, near-infrared images, point clouds, and radar tensors. (b) “Where to fuse”: the position transformation for radar-camera fusion, including two perspectives: front view and bird’s eye view. (c) “When to fuse”: at which stage the two modalities are fused, encompassing the fusion of final objects, raw data, extracted features and hybrid representations. (d) “How to fuse”: the data alignment and fusion operations, comprising of two types of alignment, namely temporal alignment and spatial alignment, and five types of fusion operations, namely addition, mean, multiplication, concatenation and attention.

perception information for the decision-making and control systems. For other downstream tasks, such as trajectory prediction [184] and vehicle navigation [185], radar-camera fusion has successfully demonstrated excellent driving performance in both unseen urban and heavy traffic scenarios.

Numerous studies have also demonstrated that radar-camera fusion improves the accuracy and robustness of the network. As it is difficult for image-based detectors to detect distant objects, Chadwick et al. [186] combined a radar sensor and sets of camera sensors in their experiments. Results exceed the performance of the camera detector, as the radar sensor persists in delivering a potent indication of motion for faraway objects. Major et al. [16] also proved that the velocity dimension derived from the radar sensor could be leveraged to increase detection performance. Additionally, Nabati and Qi [18] utilized radar features (e.g., depth, rotation, velocity) to complement the image features, resulting in an improvement of the overall nuScenes Detection Score (NDS) by more than 12% compared to the SOTA camera-based algorithm including OFT [187], MonoDIS [188] and CenterNet [189]. In noisy circumstances, Yadav et al. [190] discovered that radar data exhibit robustness in detection, and integration of radar data could enhance performance in these challenging scenarios.

### B. What to Fuse

The objective of radar-camera fusion is the output data from the radar sensor and camera sensor, which are presented in different modalities at various fusion levels and via different fusion techniques. For the camera sensor, the output data is typically presented as 2D images. In radar-camera fusion, there are mainly two kinds of images. One type is the RGB image with rich color information, such as images in nuScenes [5] dataset. The other is the infrared image captured with infrared cameras (including Far Infrared (FIR) and Near Infrared (NIR)), as illustrated in the images from SeeingThroughFog [53] dataset.

Though in lower resolutions, these images contain specific advantages in temperature differences and night visibility. The data structure of an image is relatively simple, with low data dimensionality and high correlations between neighboring pixels. The simplicity of this structure allows deep neural networks to learn the fundamental representations of images, thus enabling them to detect objects within the images [177].

As mentioned in Section II-A3, radar data can be classified into different representations depending on the level of processing. ADC Signals, the underlying digital signals of the radar, cannot be marked with the location information of an object. MDS, a Time-Frequency representation, consists of consecutive radar frames and does not correspond to a single image frame. As a result, ADC Signals and MDS are commonly used for identifying the presence of objects and discriminating between different objects [46], [191], [192], [193]. With the ability to describe the shape of an object, radar tensors and point clouds are commonly leveraged for object detection and semantic segmentation tasks.

### C. Where to Fuse

1) *Front View*: Fusion at the Front View (FV) involves projecting the radar data onto an image plane, where the radar data can be 3D point clouds, partial point cloud information or radar tensors. Around the projected area, proposals that indicate potential objects are generated [9], [190], [194], [195], [196]. In this way, a large number of non-object regions are excluded, thus reducing the computational burden and increasing recognition speed. Radar data mapped to the image plane can also be utilized to create feature maps for complementing the image-based features [18], [186], [197], [198]. These methods improve the detection accuracy by leveraging the additional input, including ranges, velocities and RCS values. Moreover, some researchers project radar point clouds onto the image plane to form a radar pseudo-image [28], [199], [200], [201]. For example, In RVNet [28] and SO-Net [199], a pseudo-image named “Sparse

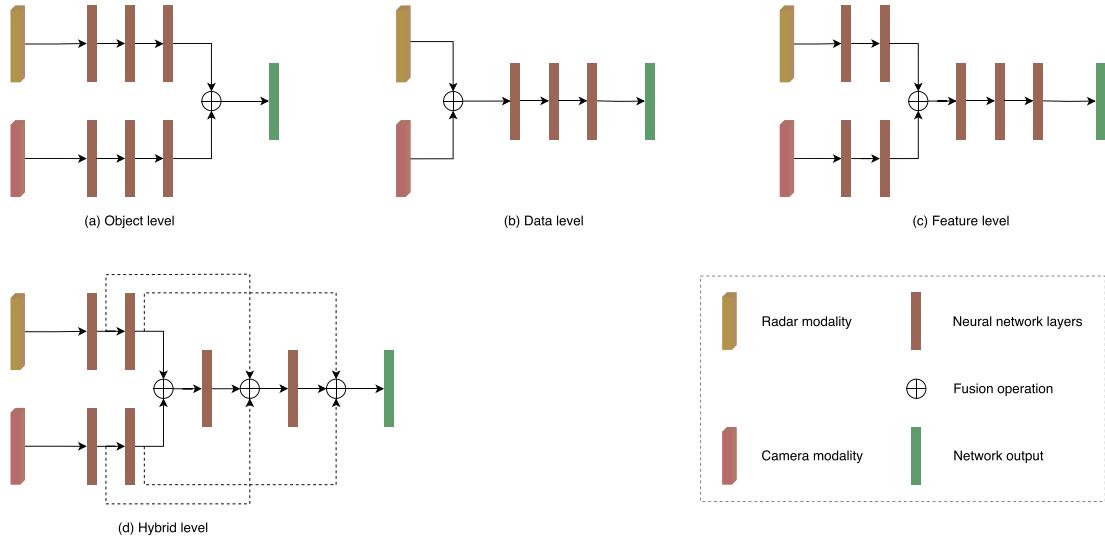


Fig. 7. Taxonomy of radar-camera fusion levels. (a) Object-level fusion. (b) Data-level fusion. (c) Feature-level fusion. (d) Hybrid-level fusion.

Radar Image” is generated from radar data, containing information regarding depth, lateral velocity and longitudinal velocity. Besides, Dong et al. [200] projected both radar point clouds and 2D bounding boxes onto the image plane, forming new pseudo-images from camera RGB images. MS-YOLO [201] generates radar mask maps by a mapping transformation neural network. In each mask map, the boxed area represents the presence of an object, and the gray value of each box indicates the velocity information of that object.

Projecting radar data onto the image plane assists in providing proposals and features. However, due to the low resolution in the azimuth angle provided by the radar as well as camera calibration errors, projected radar point clouds may deviate from the object. While increasing the Region of Interest (RoI) could potentially address the issue, it results in multiple objects within the same region, and consequently being detected repeatedly, causing confusion in object matching. Moreover, due to the occlusion of objects, the projection of radar data onto the image perspective may be limited.

2) *Bird’s Eye View*: Another fusion position is to convert radar data or camera images into BEV coordinates. For example, radar point clouds from each frame generate a BEV image of six height maps and one density map in [175]. Besides, Cui et al. [195] projected radar point clouds to both FV and BEV, and proposed a 3D region proposal network to generate proposals from both camera images and radar BEV images. Compared with generating proposals directly from point clouds, the CNN-based proposal generation approach increases the quality of proposals by leveraging the network’s ability to extract deeper and richer information. Problems come that BEV images discretize the sensing space into grids, which may lead to the loss of valuable information necessary to refine bounding boxes. To address this issue, Bansal et al. [202] added additional point-based features (e.g., velocities, RCS values) to the BEV map. Simple-BEV [203] converts all radar point clouds from multiple radar sensors into BEV coordinates to yield high-dimensional BEV feature maps.

Apart from projecting radar data into BEV, Inverse Projection Mapping (IPM) [204], [205] method can be utilized to convert camera images from FV to BEV with a homography matrix. For instance, Lim et al. [206] transformed the camera images into Cartesian coordinates using IPM and then combined them with 2D radar RA tensors. In addition, both radar point clouds and camera images are projected onto BEV in [207], where the independent feature extractors learn shared features. Consequently, projecting data on BEV offers several advantages over FV, particularly in the case of occlusion [208]. Nonetheless, since IPM is based on an assumption of flat road surfaces, it often produces distortions of dynamic objects when applied to real-world scenarios [207].

#### D. When to Fuse

When to fuse refers to at which stage the radar and camera data are fused in the network. Based on the occasion of the fusion process, we classify radar-camera fusion levels into object-level, data-level, feature-level and hybrid-level. Fig. 7 illustrates the overview and difference between the four fusion levels.

1) *Object-Level Fusion*: For object-level fusion (also known as decision-level fusion or late-level fusion), the independent objects acquired from the radar and camera sensors are fused at a later stage of the network to obtain the final integrated results, as demonstrated in Fig. 7(a). In object-level fusion, how to match the results from the two different modalities is worth considering. One way is to calculate the similarity (e.g., location, size, category) and then employ methods such as the Kalman filter, Bayesian theory, Hungarian algorithm and Bipartite Matching to match the outputs. Another approach involves utilizing the transformation matrix between the radar and camera to determine the position relationships between the two modalities. For example, Jha et al. [209] projected the radar detections onto the image plane using the transformation matrix, and then aligned independent detection objects from the two sensors. Moreover, after completing the association of radar

point clouds with camera images, Dong et al. [200] proposed AssociationNet for learning the semantic representation information from the two sensors. This network improves the accuracy of association by calculating and minimizing Euclidean distance between the representations from the pair of radar point clouds and image bounding boxes.

Object-level fusion is commonly used in conventional radars and cameras, which offers high flexibility and modularity [210]. However, it also relies heavily on the accuracy of outputs from individual modules. For example, in scenarios where the camera sensor is obstructed, object-level fusion exclusively depends on the final objects detected by the radar sensor. Besides, rich intermediate features are discarded due to the sensing modality's weaknesses or errors in the sensors. As a result, object-level fusion methods can only utilize limited information obtained from the detection results.

2) *Data-Level Fusion*: For data-level fusion (also referred to as low-level fusion or early-level fusion), the raw data or pre-processing data from radar and camera sensors are fused at the early stage of deep learning models, illustrated in Fig. 7(b). Nobis et al. [211] fed the concatenated camera and radar point clouds into the network and then employed VGG [212] to extract features from the combined data. Moreover, Bansal et al. [202] created a Semantic-Point-Grid (SPG) representation from camera semantic maps, radar point clouds and radar BEV grid maps. In their method, the SPG representation is then fed into SPG encoding to extract semantic information from cameras, aiding in the identification of radar points associated with objects of interest. Instead of fusing radar point clouds with camera images, Nabati and Qi [9] proposed an RRPN, which generates object proposals to narrow the scope of detection on the camera images. However, if there is no radar point on an object, this object will be ignored. To solve the difficulty of associating radar point clouds with image pixels, Long et al. [213] presented Radar-Camera Pixel Depth Association (RC-PDA), a learned method that associates radar point clouds with nearby image pixels to enhance and densify the radar image.

With the input of raw data, it is possible to exploit complete characteristics and learn a joint representation from these two modalities. However, data-level fusion methods tend to be sensitive to temporal or spatial misalignment within the data. Precise external calibration of the two sensors is essential for data-level fusion. Besides, as radar data representations are not consistent with the object's shape, it is difficult to match the radar tensors or radar point clouds with the image pixels.

3) *Feature-Level Fusion*: In feature-level fusion (also called middle-level fusion), features extracted from separate radar data and camera images are combined at an intermediate stage in deep learning-based fusion networks, as shown in Fig. 7(c). In [186], features from both radar and camera branches are generated by ResNet [99] blocks and then fused by concatenation and addition operations. CenterFusion [18] detects objects by locating their center points in the image using CenterNet [189]. After that, it utilizes a frustum-based association strategy to accurately match radar detections with objects in the image, generating radar-based feature maps to augment the image features. SAF-FCOS [214] introduces the attention mechanism for weighting

different positions of the feature maps. Specifically, it utilizes a Spatial Attention Fusion (SAF) block to merge the feature maps from radar and camera. In the SAF block, the radar image's feature maps are encoded as a spatial attention weight matrix, which is then applied to all channels to re-weight the feature maps extracted by the camera sensor. BIRANet [190] uses Concurrent Spatial and Channel Squeeze & Excitation (scSE) blocks [216] to highlight important spatial features and significant channels. The scSE block acts as attention and adaptively boosts activation of areas where radar point clouds are present while suppressing activation at other locations. This boosted feature map is then fused with the image feature map to improve the performance of the detection network. Considering feature maps that provide enhanced spatial and channel-wise information, BIRANet exhibits the capability to detect small objects that SAF fails to identify, as illustrated by the detector heatmaps shown in Fig. 8. In fact, attention maps can be generated from various sensors. Bijelic et al. [53] extended the sensors to RGB camera, gated camera, LiDAR and radar by transforming all sensor data into uniform image coordinates. The feature maps of different sensors are then superimposed together by concatenation and multiplied with the sigmoid-processed entropy map for the final feature output.

For feature-level fusion, it is possible to design appropriate feature extractors for each modality according to its specific characteristics. Neural networks can also learn features jointly across modalities, making them complementary to each other. However, it is worth noting that feature extraction and feature fusion do not address scenarios where camera sensors become unreliable [210].

4) *Hybrid-Level Fusion*: Apart from equally fusing final objects, raw data or features from two modalities, some fusion methods combine different stages of data, which we define as hybrid-level fusion, shown in Fig. 7(d). In [194], radar proposals are first generated from radar point clouds and 3D anchors derived from camera images. Then a Radar Proposal Refinement (RPR) network is proposed to fuse the radar proposals with camera image features, which enables the adjustment of the size and location of the radar proposals in the image. Besides, the RPR network also estimates an objectness score for each radar proposal, as some radar point clouds are caused by background noise. Similarly, Cui et al. [195] generated proposals based on camera images and radar BEV point clouds, followed by projecting the proposals onto three feature maps from camera images, radar BEV point clouds and radar FV point clouds. A Self-Supervised Model Adaptation (SSMA) block [217] is utilized to fuse the proposals with features, which leverages an attention scheme for better correlation. Furthermore, HRFuser [218] introduces ideas from HRNet [134] and HRFormer [219], adopting an asymmetric Multi-Window Cross-Attention (MWCA) to fuse the features captured by the RGB camera, LiDAR, radar and gated camera.

Compared with data-level and feature-level fusion, fusion from both proposals and features leads to more accurate proposals, producing better features for the two-stage network [194], [195]. Generally, different modalities have different contributions to radar-camera fusion. One modality dominates, while

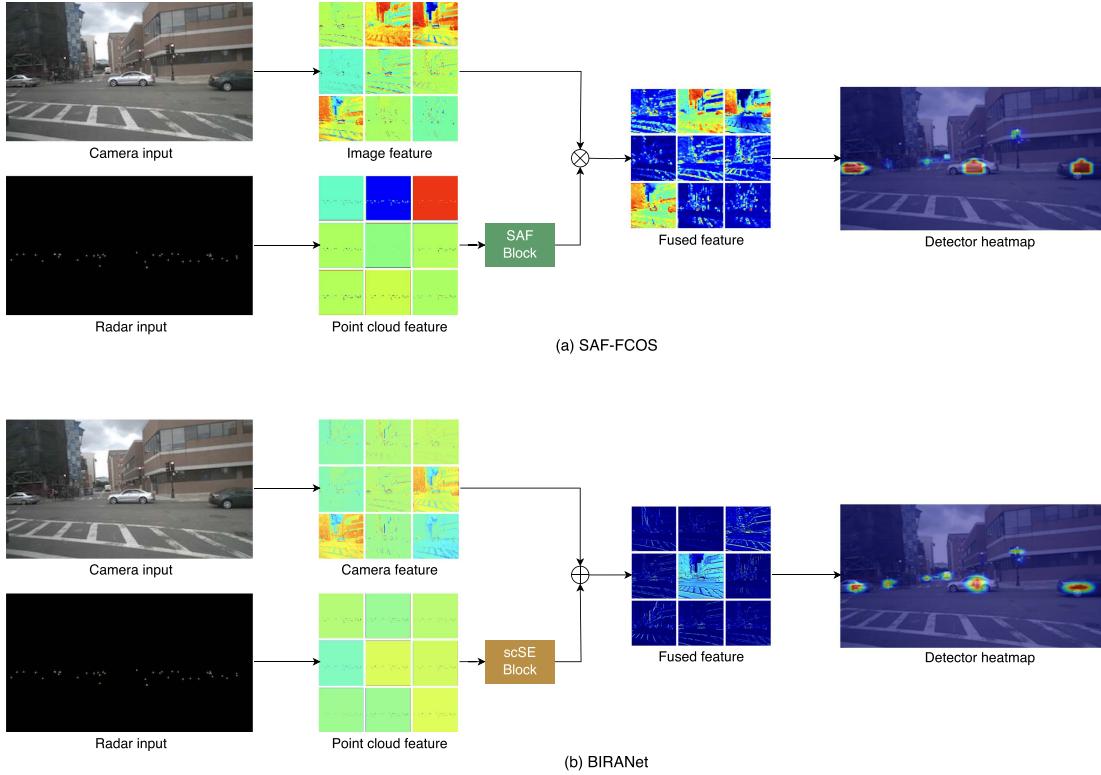


Fig. 8. Feature heatmaps of two feature-level fusion networks, namely (a) SAF-FCOS [214] and (b) BIRANet [190]. Both networks take inputs from camera and radar branches, and extract image features and point cloud features using respective backbones. In the feature fusion stage, SAF-FCOS employs a multiplication operation, while BIRANet performs an element-wise addition operation. Detector heatmaps generated by the Grad-CAM [215] are utilized for the result analysis of each method. To clearly show feature maps, only nine channels are selected for visualization.

the other provides supplementary information to refine the features. Thus, hybrid-level fusion takes advantage of different data levels and effectively preserves information at various stages. However, hybrid-level fusion should consider the importance of different modalities, which also pose implementation challenges. Since most implementations of hybrid-level fusion are based on experience and lack explainability to some extent, conducting numerous ablation experiments is needed to validate the efficiency of hybrid-level fusion. Moreover, models based on hybrid-level fusion typically have more branches in neural networks, dramatically slowing down the inference time.

#### E. How to Fuse

In this section, we present how to fuse radar data with camera images. First of all, the primary consideration is the temporal and spatial alignment between the two sensors. Then, in fusion operations, we compare five operations and analyze their advantages and disadvantages.

##### 1) Data Alignment:

a) *Temporal Alignment*: Temporal alignment in sensor fusion refers to synchronizing the temporal sequences of data from different sensors. To obtain high-quality fusion results, the data collected by each sensor must be synchronized with the same time dimension. However, there may be time offsets between these sensors due to the differences in set-up time, crystal oscillator frequencies and measurement latency. Depending on the

object of the temporal alignment methods, we categorize them into two types: estimating temporal latency between sensors and estimating temporal offset within the same frame.

*Estimating Temporal Latency*: Generally, temporal latency consists of the measurement latency between sensors and the drift between different frames. Measurement latency mainly stems from computer scheduling, measurement acquisition, pre-processing, and communication transfer time. In aligning cycle time, drift is caused by the offset between the internal clock and the Coordinated Universal Time (UTC).

A software-based technique for reducing temporal error is periodically estimating the maximum measurement latency and drift time [220], [221], [222]. Another alternative approach is to predict the future latency between sensors using Kalman filters [223] or Bayesian estimation [224] based on prior knowledge of the sensors' latency. These methods improve the synchronization result and are suitable for most applications. Since trigger signals for sensors are not initiated simultaneously, there inevitably remains some degree of unknown latency, which can cause variations in acquisition times during data fusion. Thus, some researchers proposed solutions by combining a hardware controller trigger with the software strategy to reduce the execution time of activation threads in software [225], [226], [227]. These approaches communicate with hardware synchronization components at a low level to eliminate the data acquisition latency. However, standard commercial hardware often lacks hardware synchronization interfaces [228]. When using such

methods, the complexity and portability of the system design should be considered.

*Estimating Temporal Offset:* As the temporal offset between sensors directly affects the fusion quality, some studies proposed temporal calibration strategies based on aligning the same objects from camera and radar sensors to extract timestamp offset. For example, Du et al. [229] aligned the frames that a vehicle passes the detection line and then estimated the temporal offset between these two frames. Moreover, some researchers [211], [230], [231] suggested employing real-time pre-processing buffers that leverage algorithms like YOLOv3 [77] and DBSCAN [232] to reorganize the same frame.

*b) Spatial Alignment:* Spatial alignment between radar and camera sensors involves transformation operations that map 3D or 2D radar point clouds to camera image pixels. As the spatial calibration between radar and camera is a fundamental task for information fusion, several methods of joint calibration have been proposed. Among these approaches, whether a specially designed calibration target is needed in the calibration process is an important indicator, leading to different calibration design strategies.

*Target-based Approaches:* For target-based calibration approaches, specific calibration targets are utilized so that sensors can get precise locations of the target. These locations estimate the rigid transformation relations between the radar and camera sensors. The triangular corner reflector is the most common choice for radar calibration, reflecting specific RCS values for the positional information. Moreover, to obtain positions of the calibrated targets from both the radar sensor and camera sensor, some novel designed calibration boards are proposed. For example, a corner reflector and a styrofoam board are combined as the calibration target in [233], [234]. The styrofoam board is applied for visual recognition in camera sensors without affecting the radar signals. Wang et al. [233] designed a calibration board consisting of a styrofoam board with four holes and a corner reflector in the center of these holes. The Perspective-n-Point (PnP) [235] algorithm is then used to extract the 3D location of holes and estimate the location of the corner reflector. Moreover, Peršić et al. [234] introduced a calibration board consisting of a corner reflector and a styrofoam triangle adorned with a checkerboard, from which both radar and camera sensors can obtain accurate target positional readings. Based on the paired set of image pixels and radar points for the same targets in different locations, the transformation matrix between the radar coordinates and camera coordinates is calculated.

*Target-less Approaches:* On the other hand, target-less calibration approaches do not rely on specific checkerboards, thus improving the portability of calibration. However, the uncertainty of environmental factors when extracting the same features from multiple sensors is a common drawback in target-less calibration methods. Some researchers utilize precise radar velocity measurements based on the moving objects and the camera pose to implement radar-to-camera extrinsic calibration algorithms [234], [236], [237], [238], [239]. Besides, machine learning algorithms are also utilized in [233], [240], [241], [242] to predict calibration parameters based on improving the consistency of radar point clouds and camera images.

*2) Fusion Operations:* In radar-camera fusion, different fusion operations are used to fuse data from the two modalities. Specifically, for object-level and data-level fusion networks, a transformation matrix is commonly used to align final objects or raw data [209], [243]. In contrast, feature-level and hybrid-level fusion networks tend to utilize addition and concatenation operations. In the **addition** operation, element-wise features in the feature maps are added. Thus, each channel in the feature map contains more feature information, making the classifier comprehend the feature details. Similarly to the addition operation, the **mean** and **multiplication** operations calculate the average mean and multiplication of the element-wise feature maps, respectively. In the **concatenation** operation, the feature maps are flattened into vectors and then concatenated along the rows. The primary objective of the concatenation operation is to enrich feature diversity, enabling the classifier to recognize objects with higher accuracy.

Given that the features of radars and cameras are heterogeneous, and the above fusion operations are sensitive to changes in the input data, the effectiveness of the modalities in particular scenarios is ignored. For example, the performance of the camera sensor tends to reduce in adverse weather conditions, while the radar sensor continues to work properly. Thus, the **attention** operation is proposed to re-calculate the weights of the feature maps from two modalities. One example of such an approach is Spatial Attention Fusion (SAF), proposed by Chang et al. [214]. SAF extracts the spatial attention matrix from the radar images and then employs it to re-weight the feature maps from the image branch. Other approaches leverage the Mixture of Expert (MoE) [244], [245] to extract feature maps from respective expert networks and calculate the attention weights by a gating network. After that, based on these weights, the feature maps are re-assigned to optimize fusion performance.

## F. Network Architectures

Generally, networks for radar-camera fusion are structured with dual input branches, where the data from radar and camera is input separately. Depending on the desired fusion stage, raw data, feature maps or final objects are fused in the designed network for fusion results. Based on the representations of radar data, we classify the radar-camera fusion networks in object detection and semantic segmentation tasks into point-based and tensor-based networks. We also provide a chronological overview of radar-camera fusion algorithms in Fig. 9, and summarize the comparable contents in Table III.

*1) Point-Based Networks:* Point-based Networks take radar point clouds as input. According to the different radar point cloud processing methods, we subdivide the point-based methods into projection-based, pseudo-image-based, voxel-based and BEV-based methods.

*Projection-based Methods:* In point-based radar-camera fusion networks, radar point clouds are mostly projected onto the 2D image plane to provide proposals or features. Then networks such as VGG [212], ResNet [99], U-Net [133], YOLOv3 [77] and YOLOv4 [78] are used for feature extraction. Chadwick et al. [186] projected the radar point clouds onto

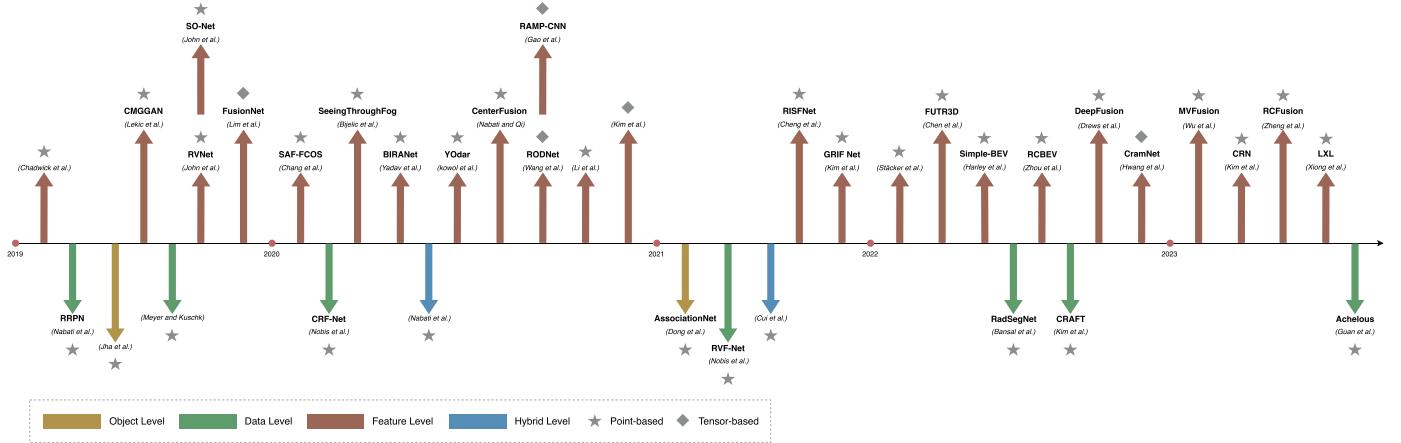


Fig. 9. Chronological overview of radar-camera fusion algorithms. Color of the arrow represents different fusion levels, and shape of the icon represents different network architectures.

the camera plane and generated two kinds of radar images: range image and range-rate image. Then they integrated an additional radar input branch upon the SSD [83] network, and used both concatenation and element-wise addition operations to fuse the radar features after the image block. The branch structure exhibits potential flexibility in re-calculating weights between the camera image and radar representations. Besides, Meyer and Kuschk [175] generated a BEV image with six height maps and one density map from the point clouds of each frame. The authors also proposed a 3D region proposal network based on VGG [212] to predict the position of boxes and the front angle of the detected object. RVNet [28], a one-stage object detection network based on the YOLOv3 [77], contains two input branches for radar and camera, and two output branches for small obstacles and big obstacles. Specifically, radar point clouds are transformed into sparse radar images in the image coordinate system via the intrinsic matrix from the camera sensor. Each sparse radar image consists of three channels, namely depth, lateral velocity and longitudinal velocity. Based on RVNet [28], SO-Net [199] is presented, focusing on multi-task learning within a single network. In RVNet, the two output branches are modified for vehicle detection and free-space segmentation. CRF-Net [211] projects the radar point clouds onto the image plane and feeds the concatenated camera and radar data into a designed VGG-based network. This network enables learning which layer fusion would yield the best benefits by adjusting the weights to radar features on different layers. In fact, particular objects in camera images tend to remain undetected in night-time scenarios, even when using standard object detection frameworks like YOLOv3 [77]. YOdar [197] involves lowering the score threshold and assigning radar point clouds to image slices, which are then combined through an aggregated output. Finally, a gradient-boosting classifier is employed to minimize the number of false positive predictions, improving the detection accuracy at night conditions.

In conclusion, projection-based methods in point-based processing leverage radar-to-image projection techniques and various deep learning networks for feature extraction. By leveraging techniques such as radar image generation, multi-stage fusion,

and network adaptation, these methods enable robust perception and scene understanding in complex environments, thereby advancing the field of radar-camera fusion networks.

**Pseudo-Image-based Methods:** Since image-based CNN networks cannot directly learn original radar point clouds, some studies convert radar point clouds into radar pseudo-images and then utilize image-based methods to extract features. Based on the distant object detection method in [186], Chang et al. [214] proposed a radar pseudo-image generation model. Apart from transforming radar point clouds from 3D coordinates into 2D camera coordinates, they also converted the depth, longitudinal velocity and lateral velocity to a real pixel value in RGB channels. Then they introduced a Spatial Attention Fusion-based Fully Convolutional One-Stage (SAF-FCOS) network using a SAF block to merge feature maps derived from radar and camera sensors. In the SAF block, features of radar images are encoded as a spatial attention weight matrix, which is employed to re-weight the feature maps from the image branch. SeeingThroughFog [53] introduces a measurement entropy to fuse features from multiple sensors adaptively. Specifically, it applies convolution and sigmoid to the input entropy for a multiplication matrix. The matrix is then utilized to scale the concatenated features from different sensors. This approach adaptively fuses features in the feature extraction stack with the most accurate information. In CenterFusion [18], a novel frustum-based radar association method is proposed to correlate radar detections with preliminary image results. Notably, the authors generated a heat map using depth and radial velocity channels to produce complementary features for the image. After that, they fed the concatenated features into the regression heads to refine the preliminary detection by re-calculating the object's depth, rotation, velocity and attributes. Finally, the results from the regression heads are decoded into 3D bounding boxes.

Overall, pseudo-image-based methods in point-based radar point cloud processing involve the transformation of radar point clouds into radar pseudo-images, which are then processed using image-based techniques. These methods utilize innovative approaches such as spatial attention mechanisms, adaptive feature fusion based on measurement entropy, and frustum-based radar

association to enhance the accuracy and robustness of detection results.

*Voxel-based Methods:* Apart from projecting radar point clouds onto the camera plane and transforming them into pseudo-images, some researchers extract features directly from 3D radar point clouds to complement the image features. This approach exploits the rich information from the radar point clouds, but requires more sophisticated processing techniques to handle the high-dimensional and unstructured nature of the data. In GRIF Net [245], an FPN [74] and a Sparse Block Network (SBNet) [246] are used as radar backbones to achieve superior performance with low computational resources. Specifically, in point cloud processing, GRIF Net converts point clouds into voxels. As the point clouds are sparse and most voxels are empty, it leverages SBNet to convolve only on masked areas, avoiding ineffective blank areas. In the fusion module, RoI features from image and radar feature maps are combined by convolutional MoE, demonstrating the effectiveness of radar sensors in detecting vehicles at longer distances than cameras. In LXL [247], the 4D radar branch produces 3D radar occupancy grids that indicate the occupancy status of radar point clouds. These 3D radar occupancy grids are leveraged together with predicted image depth maps to assist in the transformation of image perspective features to the BEV domain. This integration method effectively aligns image features with radar BEV representations, enabling effective fusion with radar features.

Above all, voxel-based methods in point-based radar point cloud processing extract features directly from 3D radar point clouds. These methods utilize techniques such as voxelization, sparse convolution, and occupancy grids to handle the high-dimensional and unstructured nature of radar data. By integrating radar features with image features, these methods demonstrate improved performance in detecting vehicles, especially at longer distances and in scenarios where camera data may be limited. The voxel-based approach allows for effective fusion and alignment of information between radar and camera modalities.

*BEV-based Methods:* Recently, architectures utilizing BEV representations and transformer networks exhibited impressive performance. Simple-BEV [203] focuses on BEV maps from multiple cameras and radars. This method generates a 3D volume with features by projecting 3D coordinates around the ego-vehicle camera images and bilinearly sampling features at projected locations. Later, a BEV feature map is produced by concatenating the 3D features with a rasterized radar image. CRAFT [248] refines image proposals by radar point clouds via a Spatio-Contextual Fusion Transformer (SCFT). The SCFT aims to leverage cross-attention layers to exchange spatial and contextual information in BEV, enabling the fusion network to learn where and what information should be extracted from camera and radar modalities. MVFusion [249] employs multi-view camera images to obtain semantic-aligned radar features, and subsequently integrates these features in a robust fusion transformer to optimize the cross-modal information interaction. CRN [250] introduces multi-modal deformable attention to tackle the spatial misalignment between radar and camera feature maps. With its aggregated semantic features and accurate BEV representations,

CRN [250] currently ranks first among all radar-camera fusion detectors in the nuScenes [5] dataset, being the best approach in 3D radar-camera fusion. RCFusion [251] achieves multi-modal feature fusion under a unified BEV perspective with the input of 4D radar and camera. The Interactive Attention Module (IAM), a key component of RCFusion, is utilized to weight the features of each modality, thus fully exploiting the advantages of both modalities.

In summary, BEV-based methods in point-based radar point cloud processing leverage BEV representations and transformer networks to achieve impressive performance in radar-camera fusion. These methods incorporate techniques such as refining proposals, cross-modal information interaction, semantic alignment, and attention mechanisms to optimize feature extraction and fusion between radar and camera data. With the ability to handle spatial misalignment and exploit the advantages of top-down perspective, BEV-based methods demonstrate high performance in 3D radar-camera fusion tasks.

*2) Tensor-Based Networks:* Due to the potential loss of crucial information concerning objects or the surrounding environment during the processing of radar point clouds after CFAR detection, several researchers have put forward a fusion scheme that involves the fusion of radar tensors with camera images. We categorize these tensor-based networks into cross-supervised-based methods and projection-based methods.

*Cross-Supervised-based Methods:* For radar data in the format of tensors, it is challenging to label the radar data as they are not spatially consistent compared to image data. Thus, some researchers propose cross-modal supervision methods to generate radar labels with the supervision of camera images. Specifically, RODNet [17], [170] is a radar object detection network using a camera-radar fusion strategy to cross-supervised 3D localization of detected objects during the training stage. It takes sequences of RA tensors as input and uses a neural network-based approach to extract the Doppler information. Specifically, to handle multi-chirp merging information and dynamic object motion, RODNet introduces two customized modules, namely M-Net and Temporal Deformable Convolution (TDC). Moreover, Gao et al. [104] fed sequences of RD tensor, RA tensor and AD tensor into convolutional autoencoders. They proposed a Radar Multiple-Perspectives Convolutional Neural Network (RAMP-CNN) that utilizes the temporal information in the chirps within a single frame, along with the change in spatial information between frames. In RAMP-CNN, features of these three tensors are then fused in a fusion module to generate new range-azimuth features. Compared to RODNet [17], [170], RAMP-CNN achieves significant performance and maintains the same detection accuracy in night scenes as in the daytime. Recently, Jin et al. [252] utilized the segmented camera image with radar customized adaption as the ground truth for training deep neural networks to perform panoptic segmentation on radar data. The proposed network utilizes panoptic segmentation to achieve radar-tailored sensing, including free-space segmentation and object detection, with only radar RA tensor in urban, rural, and highway scenarios.

In conclusion, cross-supervised-based methods enable radar data to benefit from the rich spatial information available in

camera data. Techniques like RODNet, RAMP-CNN, and cross-supervised panoptic segmentation have demonstrated the effectiveness of incorporating multi-modal supervision techniques to enhance object detection and segmentation tasks, thus showing performance improvements in handling the unique characteristics and challenges of radar data.

*Projection-based Methods:* FusionNet [206] converts both radar RA tensors and camera images into Cartesian coordinates, followed by projecting camera images onto the radar plane using a homography transformation. Upon passing through the independent feature extractor branches, features of the two modalities are passed through the additional fusion layers to form a unified feature map. As it is challenging to fuse radar tensors and camera images in 3D coordinates, Hwang et al. [253] proposed a radar-camera matching network named CramNet. CramNet overcomes the uncertainties in the geometric correspondences between the camera and radar through a ray-constrained cross-attention mechanism. Specifically, since a peak in the radar returns usually accompanies the optimal 3D position corresponding to the foreground pixel of an image, CramNet projects radar features along the pixel rays to estimate the depth and refine the 3D locations of camera pixels. Experiments on the RADIADE [168] dataset demonstrate that the CramNet outperforms the baseline results from the Faster R-CNN [73] detector. Additionally, by conducting experiments on the filtering of RA tensors via varying intensity thresholds, radar RA tensors prove to contain more meaningful information for 3D object detection than sparse point clouds.

To sum up, projection-based fusion methods, such as FusionNet and CramNet, offer practical solutions for integrating radar and camera data by leveraging geometric transformations and novel attention mechanisms. These methods contribute to advancing the integration of multi-modal information and demonstrate promising results in object detection tasks, highlighting the significance of leveraging radar tensors in perception systems.

#### G. Model Evaluations

1) *Evaluation Metrics:* Various evaluation metrics are adopted or newly proposed to evaluate the performance of radar-camera fusion models, as summarized in Table IV. Similar to image-based object detection and semantic segmentation tasks, in radar-camera fusion, commonly used evaluation metrics are precision, recall, Average Precision (AP), Average Recall (AR), mean Average Precision (mAP) and mean Intersection over Union (mIoU). However, these metrics only calculate the prediction accuracy on a given test dataset. Attributes in multi-modal datasets, such as velocity, range, size and orientation, are ignored. Besides, for multi-modal networks, the IoU thresholds should depend on object distance and occlusion, as well as the type of sensors [3].

To overcome these drawbacks, the nuScenes [5] dataset introduces mATE, mASE, mAOE, mAVE and mAAE, which stand for mean average translation, scale, orientation, velocity and attribute errors, respectively. Furthermore, they presented nuScenes Detection Score (NDS), half based on the mAP, half

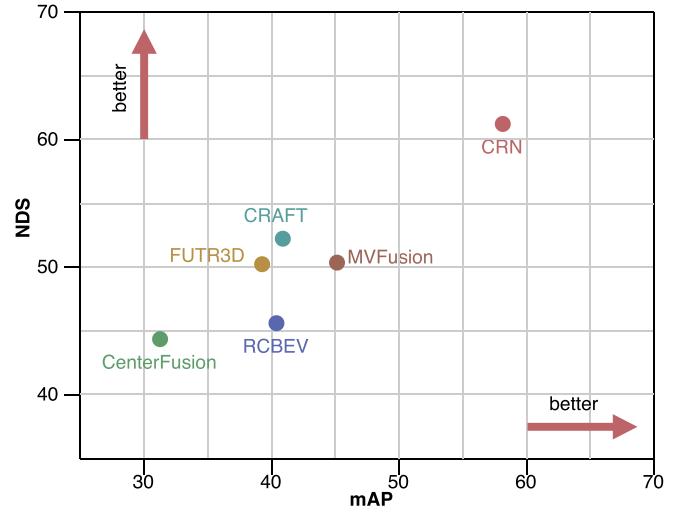


Fig. 10. Performance of radar-camera fusion methods on the nuScenes *test* set. The horizontal and vertical axes are mAP and NDS, respectively, and the larger their values, the better the performance. mAP and NDS are the metrics for evaluating the horizontal and vertical axes, respectively, where greater values indicate better performance.

quantifying the previous five metrics. To evaluate how well a detection result matches the ground truth, Wang et al. [17] defined Object Location Similarity (OLS) that quantifies the correlation between two detections concerning their distance, classes and scale information. Additionally, some metrics designed for LiDAR object detection are also adopted in radar point clouds. For example, Cui et al. [195] utilized Average Heading Similarity (AHS) to calculate the accuracy, which is formulated initially to calculate the average orientation angle in 3D LiDAR IoU defined in AVOD [7].

2) *Performance Evaluation:* Given that the majority of researchers have employed the nuScenes [5], VoD [34] and TJ4DRadSet [176] datasets to evaluate the performance of their algorithms, we provide a comprehensive summary of the evaluation metrics and performance outcomes on these dataset in Table V. We also provide Fig. 10 to clearly show the performance comparison of radar-camera fusion methods using the complete nuScenes dataset. CenterFusion [18], the first radar-camera fusion algorithm to operate on the complete nuScenes dataset, achieves a performance outcome of 32.6% on the mAP and 44.9% on the NDS. CenterFusion solved the critical data association problem in radar-camera fusion by proposing a novel frustum-based radar association method, which generates a RoI frustum around objects in 3D space using preliminary detection results, and maps the radar detection to the center of objects on the image. In comparison to CenterNet [254], which solely relies on image input, CenterFusion delivers a relative increase of 38.1% and 62.1% on the NDS and velocity error metrics, respectively, demonstrating the effectiveness of using radar features and robustness of radar-camera fusion in challenging environments.

After that, numerous radar-camera fusion algorithms employ CenterFusion as the baseline. For example, RCBEV [207], a feature-level fusion approach, extracts radar features using a

temporal-spatial encoder and transforms image features into BEV representations. Experimental results demonstrate superior feature representation and more accurate 3D object detection outcomes, receiving the mAP and NDS of 40.6% and 48.6%, respectively. CRAFT [248] achieves 41.1% mAP and 52.3% NDS on the nuScenes, where most of the gain in performance originates from the improved localization and velocity estimation with the assistance of the spatio-contextual fusion transformer. This transformer exploits both spatial and contextual properties of camera and radar data to detect objects in 3D space more accurately. CRN [250] currently emerges as the top-performing detector among all radar-camera fusion algorithms on the nuScenes dataset with 57.5% mAP and 62.4% NDS, being the best approach in 3D radar-camera fusion. The performance gain of the proposed CRN framework comes from its Radar-assisted View Transformation (RVT), which overcomes the lack of spatial information in an image and transforms perspective view image features to BEV with the help of sparse but accurate radar points. The transformed image features in BEV are then used in the Multi-modal Feature Aggregation (MFA) layers to generate a semantically rich and spatially accurate BEV representation. For 4D radar-camera fusion, LXL [247] has 56.31% mAP and 36.32% mAP, ranks first in the VoD and TJ4DRadSet dataset, respectively, being the best approach for 4D radar-camera fusion.

In general, substantial progress has been made by various algorithms operating on radar-camera fusion datasets. The mAP has improved by 24.9%, and the NDS has increased by 17.5% on the complete nuScenes dataset. On the other hand, the mAP has improved by 18.31% on the VoD dataset. The incorporation of transformer architectures, attention mechanisms, and BEV features are crucial factors that have significantly contributed to enhancing performance outcomes.

#### H. Summary

Above all, we present the methodologies of radar-camera fusion related to object detection and semantic segmentation tasks. Through in-depth analysis of the five questions revolving around “why to fuse”, “what to fuse”, “where to fuse”, “when to fuse” and “how to fuse”, we gain insight into the positive benefits of radar-camera fusion when applied on autonomous driving vehicles, including improved accuracy, robustness and redundancy. As indicated by the summary information in Table III, the number of methods designed for semantic segmentation tasks is fewer than that for the object detection task. On the one hand, as enumerated in Section III-A and Table II, there are fewer public datasets with segmentation annotations than those with detection annotations available for radar-camera fusion. On the other hand, the intrinsic characteristics of radar data render it more suitable for detection tasks. For example, the capacity for long-distance detection and velocity measurement confers a distinct advantage upon radar data for the effective detection of moving obstacles [186]. Conversely, the sparse and noisy point cloud structure presents significant limitations for semantic segmentation tasks [38].

It is important to note that although radar is not conventionally used for semantic segmentation tasks, it can effectively improve the accuracy and reliability of the fused data [202]. By providing complementary information, such as depth information, radar data can enhance the performance of semantic segmentation algorithms. Also, radar data can be employed to localize objects in 3D space, which can validate the 2D output generated by semantic segmentation algorithms. However, simple fusion operations of image and radar (e.g., concatenation) for semantic segmentation can distort the semantic structure of the image, essentially adding noise to the image. As a result, this can detrimentally impact convergence rates, causing slow and suboptimal learning of the segmentation model.

## V. CHALLENGES AND RESEARCH DIRECTIONS

It is a challenging problem to balance the performance of different modalities so that all of them can perform at their best level and thus improve the overall performance. As described in Section I, radar-camera fusion faces numerous challenges. If these challenges are not properly addressed in sensor perception, they may also affect the subsequent tasks such as localization, prediction, planning and control. In this section, we focus on improving the accuracy and robustness of radar-camera fusion, with discussions on the critical challenges and possible research directions from two aspects: multi-modal data and multi-modal fusion. An overview of these challenges and research directions is summarized in Table I.

### A. Multi-Modal Data

*1) Data Quality:* Unlike uni-modal data, multi-modal data requires consideration of each modality’s native characteristics. The information within an image is structured and regular, with partial information being associated with the whole image. In contrast, the spatial information embodied in radar point clouds tends to be disordered. As a result, handling radar data poses a more significant challenge in the context of radar-camera fusion. We categorize these challenges related to data quality into three directions: sparsity, inaccuracy, and noise.

*a) Sparsity:* The sparsity of radar point clouds poses a challenge for neural networks to learn features effectively. Besides, as these point clouds do not comprehensively represent an object’s shape, a fixed-size bounding box approach would be impractical. To address the issue of sparsity, researchers usually combine multiple frames (from 0.25 seconds to 1 s) of radar data to get denser point clouds, making it conducive to improving accuracy [18], [34], [38], [203], [211], [213], [245]. However, the time-dependent approach also causes system delays. Nowadays, the 4D radar sensor is a potential research direction as it produces denser point clouds, reaching hundreds of points on a car. The spatial distribution of objects is effectively represented in 4D radar datasets, including Astyx [175], VoD [34] and TJ4DRadSet [176]. Experiments also indicate that the 4D radar is helpful in radar detection. For example, Zheng et al. [176] proved that 4D radar has potential in 3D perception as the points get dense. Palffy et al. [34] pointed out that the additional

TABLE I  
OVERVIEW OF CHALLENGES AND RESEARCH DIRECTIONS

Topic	Sub Topic	Challenges	Research Directions
Multi-modal Data	Data Quality	<ul style="list-style-type: none"> <li>• Sparsity</li> <li>• Inaccuracy</li> <li>• Noise</li> </ul>	<ul style="list-style-type: none"> <li>• Leveraging multiple radar frames to enhance the density</li> <li>• Applying 4D radar sensors with higher resolution</li> <li>• Studying distribution of point clouds (e.g., Gaussian distribution)</li> <li>• Denoising based on physical characteristics of radar data</li> </ul>
	Data Diversity	<ul style="list-style-type: none"> <li>• Small Size</li> <li>• Insufficient Conditions</li> </ul>	<ul style="list-style-type: none"> <li>• Collecting data from adverse conditions</li> <li>• Integrating synthetic data with real-world data</li> <li>• Domain adaptation for model generalization with limited data</li> </ul>
	Data Synchronization	<ul style="list-style-type: none"> <li>• High Calibration Requirements</li> <li>• Difficulties in Labeling</li> </ul>	<ul style="list-style-type: none"> <li>• 4D radar-camera calibration</li> <li>• Real-time online calibration and correction</li> <li>• Auto-labeling to reduce manual labeling</li> <li>• Improving labeling efficiency via active learning, domain adaptation, transfer learning, semi-supervised learning</li> </ul>
Multi-modal Fusion	Feature Extraction	<ul style="list-style-type: none"> <li>• Less effective using LiDAR-based or image-based algorithms</li> </ul>	<ul style="list-style-type: none"> <li>• Introducing attention mechanisms</li> <li>• Using Graph Neural Networks to dig deeper into the relationship between sparse point clouds</li> <li>• Applying neural networks to extract radar information instead of traditional FFT operations</li> </ul>
	Data Association	<ul style="list-style-type: none"> <li>• Ambiguity in associating radar data with image data</li> <li>• Poor association using calibration matrix</li> </ul>	<ul style="list-style-type: none"> <li>• Attention-based association with adaptive thresholds</li> <li>• Joint state estimation</li> <li>• Uncertainty estimation of object tracks</li> </ul>
	Data Augmentation	<ul style="list-style-type: none"> <li>• Correlation and interdependence between radar and camera modalities</li> </ul>	<ul style="list-style-type: none"> <li>• Joint data augmentation rather than augmenting each modality separately</li> </ul>
	Training Strategies	<ul style="list-style-type: none"> <li>• Difficulties in training</li> <li>• Overfitting for multi-modal model</li> </ul>	<ul style="list-style-type: none"> <li>• Weighting operations on loss functions</li> <li>• Weights freezing strategies</li> <li>• Knowledge distillation on uni-modal features for the multi-modal networks</li> </ul>
	Model Robustness	<ul style="list-style-type: none"> <li>• Sensor degradation or failure in adverse conditions</li> <li>• Unseen driving scenarios</li> </ul>	<ul style="list-style-type: none"> <li>• Attention mechanisms</li> <li>• Uncertainty estimation</li> <li>• Generative models for sensor defects or new scenarios</li> </ul>
	Model Evaluations	<ul style="list-style-type: none"> <li>• Different selected sub-dataset</li> <li>• Unknown objects in the open world</li> </ul>	<ul style="list-style-type: none"> <li>• Metrics related to uncertainties</li> <li>• Developing visual toolkits for analyzing and optimizing fusion networks</li> </ul>
	Model Deployment	<ul style="list-style-type: none"> <li>• Edge devices with limited computational resources</li> <li>• Balancing the importance of different tasks</li> </ul>	<ul style="list-style-type: none"> <li>• Lightweight models and acceleration schemes (e.g., pruning and quantization)</li> <li>• Fusion-based multi-task perception and panoptic perception</li> </ul>

elevation data increases object detection performance (from 31.9% to 38.0% in mAP) in their VoD [34] dataset.

*b) Inaccuracy:* Aside from the sparsity of radar point clouds, the points may not be located at the object's center, but may be at any corner of the object or even outside of it [175]. To make the radar points located on or close to the object, Chadwick et al. [186] marked each radar point on the image as a small circle instead of a single pixel. Nabati and Qi [9] proposed RRPN to generate several anchors with different sizes and aspect ratios centered at the points of interest. These translated anchors are employed to achieve more precise bounding boxes where the points of interest are mapped to the object's right, left, or bottom.

Researchers have employed column and pillar expansion techniques to improve the accuracy of radar point clouds in the vertical dimension. For example, Nobis et al. [211] assumed a height extension of three meters on each radar detection to associate camera pixels with radar data. Nabati and Qi [18] used pillar expansion to expand each radar point to a fixed-size pillar. In their experiment, the size of a pillar is set to [0.2, 0.2, 1.5] meters along the  $[x, y, z]$  directions. Notably, the column size should be different for different types of objects. In [196], the authors utilized a clustering approach to group ground truth bounding

boxes of vehicles into three distinct height categories. Following this, radar points are assigned a scale based on boundary edge values in each category.

In our opinion, column or pillar expansion is effective but still hardly convincing. The distribution of point clouds is a direction worth investigating. For example, Stäcker et al. [255] assumed a Gaussian distribution to measure the azimuth angle. According to the resulting Gaussian density curve, they generated denser radar point clouds, horizontally distributed over multiple pixels.

*c) Noise:* Actually, the radar sensor returns noisy data from irrelevant objects, including ghost objects, ground detections and even multi-radar mutual interference [256]. The noise from radars could cause the detection of fake targets, thereby limiting the accuracy of radar-based detection or segmentation.

Conventional methods [257], [258], [259], [260] for automotive radar denoising are typically based on CFAR and peak detection algorithms, which exhibit poor generalization capabilities. Recently, deep learning methods have provided a key solution to the challenges associated with automotive radar data denoising. In [261], a deep neural network is proposed to enhance the target peaks on RA tensors. Rock et al. [262] analyzed the quantization of CNN-based denoising autoencoder for radar interference mitigation on radar RD tensors to guarantee real-time

inference on low-performance equipment. Dubey et al. [263] realized the multi-radar mutual interference and object detection on RD tensors simultaneously with one one-stage CNN-based neural network. Moreover, in [264], [265], [266], [267], [268], fully-convolution networks are widely proposed and applied to interference mitigation on RD tensors.

However, conventional radars generally do not provide access to the radar tensor, highlighting the importance of noise mitigation techniques at the level of radar point clouds. For example, Nobis et al. [211] designed a ground-truth filter to remove radar detections outside of the 3D ground truth bounding boxes. Cheng et al. [269] proposed a cross-modal radar point detector through the assistance of LiDAR, which could also remove the noisy points. Essentially, noise mitigation at the point cloud level is semantic segmentation of the point cloud, whereby a semantic segmentation model is leveraged to assign a category to each point. Numerous studies focus on point cloud segmentation, including PointMLP [270], PointNeXt [271], Point Transformer [272] and Point Cloud Transformer [273]. Notably, Point Transformer and Point Cloud Transformer introduce the self-attention mechanism within their point cloud processing networks to capture contextual features. However, it is worth noting that these approaches still rely on the advantages provided by two essential modules derived from PointNet++: Set Abstraction (SA) and Multi-Scale Grouping (MSG). Recently, a revolutionary non-parametric point cloud segmentation model called Point-NN has been proposed. Point-NN [274] elegantly assembles farthest point sampling, K nearest neighbor, pooling, trigonometric position encodings and similarity measurement, thereby achieving SOTA performances on several benchmarks with superior performance to any other point cloud processing models.

In all, regardless of the stage of denoising in radar tensors or point clouds, a dataset with high-quality annotations is necessary. For the noise removal on radar tensors, researchers may adopt image denoising and restoration principles. For removing noisy point clouds, constructing features based on the physical characteristics of point clouds to guide models separate targets from clutters makes sense. As radar point clouds are sparse and inaccurate, modeling the inherent uncertainty is an open question, which can aid in effectively distinguishing targets from noise.

## 2) Data Diversity:

a) *Small Size*: Deep learning models rely on large amounts of training data to achieve high levels of accuracy. However, multi-modal datasets consisting of both radar and camera data are much smaller than uni-modal image data. For instance, compared to the ImageNet [275], [276] dataset with over 14 million images and over 20 k classes, the largest radar-camera fusion dataset to date named CRUW [170] has only 400 k frames and 260 k objects. Furthermore, regarding category distribution, most labels are vehicles, while pedestrians and bicycles are far less prevalent. The imbalance in these categories' distribution may result in overfitting designed deep learning networks [158].

b) *Insufficient Conditions*: In real scenarios, 360-degree perception of the surrounding environment is critical in autonomous driving, requiring multiple cameras and radars to

work together. Besides, the multi-modal dataset also needs to consider complex weather conditions (e.g., rain, fog, snow) and complex road conditions (e.g., blocked roads, rural paths, intersections), all of which are time-consuming and labor-consuming tasks.

Some studies [169], [180] generate synthetic data via simulation tools (e.g., Carla [181]). Researchers can freely match different sensors and generate different driving conditions with these tools, especially in complex and dangerous scenarios. However, it is also worth noting that although simulators can generate a variety of virtual datasets, the simulated data cannot completely replace the data from real scenarios. Moreover, exploring the appropriate methodology for integrating synthetic data with real-world data is a critical area of inquiry warranting further investigation [277].

Domain adaptation is also a valuable research direction that aims to leverage knowledge learned from a related domain with adequate labeled data. Although domain adaptation has been applied in radar data, including radar tensor reconstruction [278], human sensing [279], human activity recognition [280] and gesture recognition [281], it has not been employed in radar-camera fusion till now.

## 3) Data Synchronization:

a) *High Calibration Requirements*: For radar-camera fusion systems, well-calibrated sensors are the prerequisite. In multi-sensor calibration, LiDAR sensors are typically employed as an essential intermediary component. The LiDAR sensor is calibrated separately from the camera sensor and radar sensor, and then a transformation matrix between the radar and camera can be calculated [173], [176]. Although numerous approaches (e.g., [195], [282], [283]) are proposed for calibration between radars and the cameras, the accuracy of the calibration remains a challenge due to the inaccurate and vulnerable radar returns. Besides, as far as we know, [195] and [177] are the only methods for 4D radar and camera calibration. As 4D radar technologies are developing rapidly, we believe that 4D radar calibration is a potential direction, and more finds will be proposed in the future.

In real scenarios, extrinsic calibration parameters between radar and camera sensors may change from vehicle vibration. Besides, different sampling frequencies of the radar and camera may produce a particular temporal difference between the data from each sensor. The temporal difference would cause data inconsistency, especially when the ego-car or targets move at high speed. Therefore, real-time online calibration and correction are essential research directions in the future.

b) *Difficulties in Labeling*: The process of labeling data is labor-intensive and time-consuming, especially when dealing with multi-modal data. This is particularly true for radar-camera fusion, where the physical shapes of objects cannot be discerned directly from the radar data representation. Auto-labeling radar data is a potential research direction to address the challenge of laborious data labeling. Actually, labels for radar data can be calculated based on the corresponding ground truth from camera images and the extrinsic matrix between the radar sensor and the camera sensor. But the problem is that applying this labeling approach for radar data is not perfect, as radar targets may not always be located in the ground truth from

images. Sengupta et al. [284] proposed a camera-aided method for automatically labeling radar point clouds, leveraging a pre-trained YOLOv3 [77] network and the Hungarian algorithm for enhanced accuracy and efficiency. However, despite the potential advantages of auto-labeling radar data, filtering out noisy data around the object of interest is still challenging.

For camera image labeling, it is worth considering how to select appropriate labeling data for reducing labor costs. Active learning is a supervised learning method that aims to select the smallest possible training set to achieve the desired data efficiency [285], [286]. The active learning network iteratively queries the most informative samples from the human labelers in an unlabeled data pool and then updates the weights for the network. This approach leads to equivalent performance with less labeled training data, reducing human labeling efforts. Experiments from [287] indicate that using only about 40% of the data in the training set leads to the same classification results as the completely supervised reference experiment. Furthermore, many other methods would also be used to reduce the burden of data labeling, such as domain adaptation [278], [288], [289], transfer learning [290], semi-supervised learning [291] and lifelong learning [292].

## B. Multi-Modal Fusion

1) *Feature Extraction*: Applying LiDAR-based feature extraction algorithms to radar modality is less effective due to the inherent sparsity of radar point clouds. As an example, PointPillars [293] algorithm converts LiDAR point clouds into pillars and then extracts features from each pillar. When this algorithm is adapted to radar point clouds, there may be few or even no points in a pillar, which makes it hard to extract features. In fact, results in [108] and [294] also indicate that the average precision of radar point clouds using PointPillars [293] is much lower than using SSD [83] and YOLOv3 [77] detectors.

As PointPillars [293] focus on local features, the attention mechanism is a potential research direction to extract global features to improve accuracy. For example, Radar Transformer [295] incorporates both vector attention and scalar attention mechanisms to effectively leverage spatial information, Doppler information, and reflection intensity information from radar point clouds. By integrating local attention features and global attention features, Radar Transformer achieves deep integration of radar information. RPFA-Net [296] leverages the self-attention mechanism to extract global features (e.g., orientation) from point clouds. These global features enable the network to perform more efficient and effective regression of key object parameters (e.g., heading angle), thereby enhancing the accuracy and reliability of object detection. In addition, Gaussian Radar Transformer [161] employs attentive upsampling and downsampling modules to enlarge the receptive field and capture distinctive spatial correlations, effectively addressing the challenge of capturing long-range dependencies in radar data. The attention-based techniques and multi-task learning used in HARadNet [109] also lead to a significant performance improvement in the classification and detection.

To dig deeper into the relationship between sparse point clouds, GNN [297] is a promising research direction in which each point is considered as a node, and edges are the relationship between the points. In Radar-PointGNN [98], GNN adopted for feature extraction of radar point clouds demonstrates that the graph representation produces more effective object proposals than other point cloud encoders by mapping radar point clouds to contextual representations. RadarGNN [39] indicates that GNNs can operate on unstructured and unordered data, obtaining both point features and point-pair features embedded in the edges of the graph. Thus, compared to voxelization operations, GNN eliminates the information loss from the sparse radar point clouds. GNN also shows its advantages in detection from RA tensors. The Graph Tensor Radar Network (GTR-Net) [298] architecture utilizes graph convolutional operations to aggregate information across the point cloud nodes. The process involves weighting the features of connected nodes based on their respective edge weights. In this way, it improves the defective sparse points by aggregating relevant information and thus leads to better performance.

Another potential research opportunity is using neural networks to extract radar information instead of traditional FFT operations, which can reduce the computational requirements that consume most of the operations and simplify the data flow in the embedded implementation. For example, in RODNet [17], FFT operations are only performed in sample and antenna dimensions, while the chirp dimension remains to get the range-azimuth-chirp tensor. Then a neural network is employed to process the chirp dimension for extracting Doppler features, enabling end-to-end training of radar features in-depth within the deep learning framework.

2) *Data Association*: Another significant challenge is the ambiguity in associating radar data with image data, as they are heterogeneous. The typical way is to project radar data onto the image plane and then bind the data in the same position through a calibration matrix [299]. However, direct projection results in poor association with the objects' centers. As aforementioned, radar data is sparse, inaccurate and noisy, making poor association at either the object-level or data-level fusion.

Thus, associating image data with radar data is an open question. Nabati and Qi [194] proposed a Radar Proposal Refinement (RPR) network to match proposals from radars and cameras. Later, they integrated the detection boxes and pillar expansion through frustum association in CenterFusion [18], allowing for the mapping of radar detections to the centers of objects and mitigating the issue of overlapping. Dong et al. [200] used AssociationNet to learn the semantic representation information and associate radar point clouds and image bounding boxes by Euclidean distance. For associating the semantics to radar point clouds, Bansal et al. [202] proposed a representation named Semantic-Point-Grid (SPG), which encodes semantic information from camera images into radar point clouds to identify camera pixel correspondences for each radar point.

In our opinion, a potential approach to associate radar data with image data is the attention-based association with adaptive thresholds. For example, Radar-Camera Pixel Depth Association (RC-PDA) is proposed to filter out occluded radar returns

and enhance the projected radar depth map by generating a confidence measure for these associations in [213]. Soft Polar Association (SPA) is proposed to associate radar point clouds around the image proposals in polar coordinates [248]. In order to overcome background clutter, it utilized consecutive cross attention-based encoder layers to integrate image proposal features and radar point features.

**3) Data Augmentation:** Numerous data augmentation methods have been proposed to increase the quantity and diversity of data samples, thus preventing network overfitting and enhancing model generalization. For radar data in the form of point clouds, data augmentations such as random rotation, scaling, and flipping shifting are applied to enrich the diversity of samples in [18], [176], [207]. In addition, since radar tensors can be treated as images, existing image-based data augmentation algorithms (e.g., horizontal flipping, translating in range, interpolating, mixing) are tested in experiments and proved to be effective [104].

However, all these data augmentation methods above are based only on the radar modality. In radar-camera fusion perception, designing effective data augmentation methods needs to consider the correlation and interdependence between radar and camera modalities, which means joint data augmentation methods are necessary rather than augmenting each modality separately. Otherwise, the model will learn from incorrect data, whose physical properties are unreliable. For example, when the coordinates of radar data and image are aligned, applying Cutmix [300] on the image and radar feature maps will undoubtedly destroy the target features (e.g., azimuth and elevation) obtained by the radar sensor, leading to incorrect model inferences. Therefore, designing joint data augmentation algorithms for the unique radar representations combined with image modality remains a significant challenge.

**4) Training Strategies:** Since a multi-modal network has additional input information, it should match or outperform the uni-modal network. However, this is not always the case. A multi-modal network is often prone to overfitting and tends to learn to ignore one branch if the hyper-parameters set for training are more suitable for the other branch. Wang et al. [301] argued that the rates of overfitting and generalization vary across different modalities, and training a multi-modal network using the uni-modal training strategy may not be optimal for the overall network.

A feasible approach to balance the performance is to add loss functions for each modality. In this way, after one modality converges, the remaining modality can still be generalized. Besides, weighting operations on loss functions could be more beneficial to adapt to the learning rates of each modality. In recent studies, Wang et al. [301] proposed Gradient-Blending, which computes an optimal blending of modalities based on their overfitting behaviors. Although this method achieves SOTA accuracy on audio and vision benchmarks, the idea has yet to be applied in radar and camera modalities. Moreover, dropout operation helps overcome overfitting. Nobis et al. [211] introduced BlackIn by deactivating camera image data. The lack of camera input data forces the network to rely more on sparse radar data for specific potential values.

Fine-tuning a multi-modal network over pre-trained uni-modal encoders can also outperform fusion from scratch. Lim et al. [206] utilized the weights freezing strategy to train a single branch network using the optimal training hyper-parameters. These weights were subsequently loaded into the corresponding branches to train the fusion network. Experimental results indicate that the best strategy is to train the camera branch in advance and then train the entire network with the gradient propagation disabled through the camera branch. Recently, knowledge distillation has shown its performance in multi-modal networks by distilling the pre-trained uni-modal features to the multi-modal networks [302], [303]. It could also be a potential research direction in radar-camera fusion.

**5) Model Robustness:** Another challenge is how to guarantee the model's robustness when the sensors are degraded, or the autonomous driving vehicles enter into adverse or unseen driving scenarios. Most reviewed methods focus on the accuracy of public datasets, while only a few consider sensor failure with only one modality as the input data. In RadSegNet [202], the SPG encoding independently extracts information from cameras and radar, as well as encodes semantic information from camera images into radar point clouds. Thus in scenarios where the camera input becomes unreliable, the SPG encoding method maintains reliable operation using radar data alone. Bijelic et al. [53] introduced an entropy channel for each sensor stream and a feature fusion architecture to exchange features, which still work in unseen weather conditions and sensor failures. Moreover, the attention mechanism is also an effective choice for guiding mixed information from different sensors to fuse features of multiple modalities, as well as handle original features from a single modality. For example, attention maps leverage features learned from different sensors to predict the importance of specific parameters in [304].

It is essential to focus not only on the accuracy of the predicted outcomes, but also on the degree of certainty the model has about them. Uncertainty is a potential direction that can be used to handle unseen driving scenarios. Specifically, a multi-modal network should present higher uncertainty against unseen objects. The Bayesian neural network is a valuable choice for calculating uncertainty. It utilizes a prior distribution of network weights to infer the posterior distribution, thereby estimating the probability associated with a given prediction [3], [305]. In radar-camera fusion, YOdar [197] is an uncertainty-based method in which uncertainty combines outputs of radar and camera networks with a gradient-boosting classifier. Experimental results show that YOdar increases performance significantly at night scenes.

Another way that may be useful to increase the networks' robustness is generative models. They can detect sensor defects or new scenarios an autonomous vehicle has never entered. Wheeler et al. [306] described a methodology for constructing stochastic automotive radar models based on deep learning with adversarial loss connected to real-world data. The resulting model exhibits fundamental radar effects while maintaining real-time capability. Lekic and Babic et al. [307] introduced a Conditional Multi-Generator Generative Adversarial Network (CMGGAN) to generate pseudo-images containing all the

surrounding objects detected by the radar sensor. In our opinion, designing specific deep generative models for radar-camera fusion is an interesting open question.

*6) Model Evaluations:* Most researchers utilize the nuScenes [5] dataset to evaluate the performance of their algorithms. However, the selected sub-dataset and the evaluation metrics are different, leading to a lack of direct comparisons. As summarized in Table V, some methods [9], [197], [199], [214], [243] use portions of the nuScenes dataset for training, validating and testing, while some others [28], [194], [245], [255] exploit data collected by part of sensors within the nuScenes dataset. In addition, some researchers [190], [196], [211] do not clarify which section of the nuScenes dataset they utilized in their experiments.

In terms of evaluation metrics, even though some studies provide results using AP and mAP metrics, the type and value of the threshold are different. Besides, only a few works provide information on the inference time, which is also calculated by authors on their own devices and lacks uniform hardware measurements. In our opinion, since the nuScenes [5] dataset has been used to evaluate the performance of major algorithms, researchers should validate the performance of their algorithms on the same IoU, metrics and sub-dataset. This would enable a more meaningful and direct comparison of the results obtained from various studies.

Moreover, standard evaluation metrics are not specifically designed for situations where sensors are defective. Metrics related to uncertainties, such as Probability-based Detection Quality (PDQ) [308], may be helpful in radar-camera fusion to compare the robustness of different algorithms. Radar-camera fusion also faces the challenges of unknown objects in the open world. In such scenarios, evaluation metrics proposed in [309], [310] can be utilized for open-set objects in radar-camera fusion.

Furthermore, visualization evaluation techniques are a potential research direction for analyzing and optimizing radar-camera fusion networks. Several methods [311], [312] have been proposed for interpreting and understanding deep neural networks. However, to the best of our knowledge, there has yet to be an investigation of visual analytics in radar-camera fusion. How to design the visual toolkits for radar-camera fusion networks is still an open and challenging question.

*7) Model Deployment:* Radar-camera fusion holds significant potential in practical autonomous driving vehicles, where models of radar-camera fusion are deployed on edge devices. Compared with high computational servers, edge devices are often equipped with limited computational resources in memory, bandwidth, Graphics Processing Unit (GPU) and Central Processing Unit (CPU). Nevertheless, they still need to meet the low-latency and high-performance requirements. Currently, [313] is the only work that reports on fusion output speed, reaching 11 Hz on an NVIDIA Jetson AGX TX2. The results of fusion algorithms on edge devices are an open question, and how to improve the computational efficiency is worth considering. Some network acceleration schemes (e.g., pruning and quantization [314], [315]) are good choices to be applied to radar-camera fusion models.

It is valuable to implement multiple tasks in a uni-model for real applications. In multi-task learning, the knowledge learned during training for one task can be shared and used to improve performance on the other tasks [4]. Besides, by sharing model features between multiple tasks, the overall number of parameters and computations can be reduced, making it more efficient in real-time autonomous driving applications [194], [316]. Multi-task in radar-camera fusion is still in the preliminary stage, and we believe the multi-task approach in radar-camera fusion is a potential research direction. Nonetheless, combining multiple tasks into a unified optimization objective results in a complex optimization problem, especially when the tasks are related but have different performance metrics. Finding a set of hyper-parameters that can effectively balance the importance of different tasks is challenging.

## VI. CONCLUSION

With the rapid development of autonomous driving, radar-camera fusion, a multi-modal and all-weather solution, is gaining more attention in both academic research and industrial applications. This review investigates and discusses radar-camera fusion studies on object detection and semantic segmentation tasks. Starting with the working principles of radar and camera sensors, we gradually introduce the importance of radar-camera fusion in autonomous driving perception. Through the analysis of radar signal processing, we gain a deep understanding of radar representations, which also provides fundamental support for the radar-camera fusion datasets. As to fusion methodologies, we delve into various fusion methods and explore questions about “why to fuse”, “what to fuse”, “where to fuse”, “when to fuse” and “how to fuse”.

Based on the current radar-camera fusion datasets and methods, we discuss the critical challenges and raise possible research directions involving multi-modal data and multi-modal fusion. In general, radar-camera fusion is moving towards data representations containing rich information. On the one hand, representations such as ADC signals and radar tensors provide more potential information, which is valuable for multi-modal fusion. On the other hand, the new 4D radar sensors provide denser point clouds and higher resolutions, which will become a new trend in autonomous driving. Fusion approaches are evolving towards customizing radar algorithms based on particular radar characteristics. Additionally, methods on multi-frames and multi-tasks in radar-camera fusion are expected in future works. Above all, we hope that our survey serves as a comprehensive reference for researchers and practitioners in developing robust perception in radar-camera fusion.

## APPENDIX

Tables II–V provide overviews and summaries of radar-camera fusion datasets, methods, evaluation metrics, and performance of various methods.

**TABLE II**  
**OVERVIEW OF RADAR-CAMERA FUSION DATASETS**

Name	Year	Tasks	Annotations	Data Representations	Categories	Size	Link
nuScenes [5]	2019	Object Detection, Object Tracking	3D box-level	Camera: RGB image; Radar: point cloud	23 classes (Vehicle, Pedestrian, Bicycle, Movable Object, Static Object, etc.)	1000 scenes, 1.4M boxes in 40k frames, 5.5 hours	<a href="https://www.nuscenes.org/nuscenes">https://www.nuscenes.org/nuscenes</a>
Astyx [317]	2019	Object Detection	3D box-level	Camera: RGB image; Radar: point cloud	7 classes (Bus, Car, Cyclist, Motorcyclist, Person, Trailer, Truck)	500 frames, around 3000 labeled objects	<a href="http://www.astyx.net">http://www.astyx.net</a>
SeeingThroughFog [53]	2020	Object Detection	2D box-level, 3D box-level	Camera: RGB image, gated image, thermal image; Radar: point cloud	4 classes (Passenger Car, Large Vehicle, Pedestrian, Ridable Vehicle)	12k samples in real-world driving scenes and 1.5k samples in controlled weather conditions within a fog chamber, 100k objects	<a href="https://www.uni-ulm.de/en/in-driveu/projects/dense-datasets">https://www.uni-ulm.de/en/in-driveu/projects/dense-datasets</a>
CARRADA [33]	2020	Object Detection, Semantic Segmentation, Object Tracking, Scene Understanding	2D box-level, 2D pixel-level	Camera: RGB image; Radar: range-azimuth tensor, range-Doppler tensor	3 classes (Pedestrian, Car, Cyclist)	12,666 frames, 78 instances, 7,139 annotated frames with instances, 23GB synchronized camera and radar views	<a href="https://arthrouaknine.github.io/codeanddata/carrada">https://arthrouaknine.github.io/codeanddata/carrada</a>
HawkEye [180]	2020	Semantic Segmentation	3D point-level	Camera: RGB image; Radar: point cloud	9 classes of cars (Sub-compact, Compact, Mid-sized, Full-sized, Sport, SUV, Jeep, Van, Truck)	3k images, 4k scenes, 120 car models	<a href="https://jguan.page/HawkEye">https://jguan.page/HawkEye</a>
Zendar [167]	2020	Object Detection, Mapping, Localization	2D box-level	Camera: RGB image; Radar: range-Doppler tensor, range-azimuth tensor, point cloud	1 class (Car)	Over 11k moving cars labeled in 27 diverse scenes with over 40k automatically generated labels	<a href="http://zendar.io/dataset">http://zendar.io/dataset</a>
RADIATE [168]	2020	Object Detection	2D box-level	Camera: RGB image; Radar: range-azimuth tensor	8 classes (Car, Van, Bus, Truck, Motorbike, Bicycle, Pedestrian, A group of pedestrians)	200k bounding boxes over 44k radar frames	<a href="http://pro.hw.ac.uk/radiate">http://pro.hw.ac.uk/radiate</a>
AIODrive [169]	2020	Object Detection, Semantic Segmentation, Object Tracking, Trajectory Prediction, Depth Estimation	2D box-level, 3D box-level	Camera: RGB image; Radar: point cloud	11 classes (Vehicle, Pedestrian, Vegetation, Building, Road, Sidewalk, Wall, Traffic Sign, Pole and Fence)	500k annotated images for five camera viewpoints, 100k annotated frames for radar sensor	<a href="http://www.aiodrive.org">http://www.aiodrive.org</a>
CRUW [122]	2021	Object Detection	2D box-level	Camera: RGB image; Radar: range-azimuth tensor	3 classes (Pedestrian, Cyclist, Car)	400k frames, 260k objects, 3.5 hours	<a href="https://www.cruwdataset.org/">https://www.cruwdataset.org/</a>
RaDiCaL [171]	2021	Object Detection	2D box-level	Camera: RGB image, RGB-D image; Radar: ADC signal	2 classes (Car, Pedestrian)	393k frames	<a href="https://publish.illinois.edu/radicldata">https://publish.illinois.edu/radicldata</a>
RadarScenes [179]	2021	Semantic Segmentation, Object Tracking	2D point-level	Camera: RGB image; Radar: point cloud	11 classes (Car, Large Vehicle, Truck, Bus, Train, Bicycle, Motorized Two-wheeler, Pedestrian, Pedestrian Group, Animal, Other)	40,208 frames, 158 individual sequences, 118.9M radar points	<a href="https://radar-scenes.com">https://radar-scenes.com</a>
RADDet [97]	2021	Object Detection	2D box-level, 3D box-level	Camera: RGB image; Radar: range-azimuth-Doppler tensor	6 classes (Person, Bicycle, Car, Motorcycle, Bus, Truck)	10,158 frames	<a href="https://github.com/ZhangAoCana/RADDet">https://github.com/ZhangAoCana/RADDet</a>
FloW [172]	2021	Object Detection	2D box-level	Camera: RGB image; Radar: range-Doppler tensor, point cloud	1 class (Bottle)	4k frames	<a href="https://github.com/ORCA-Ubot/FloW-Dataset">https://github.com/ORCA-Ubot/FloW-Dataset</a>
RADial [105]	2021	Object Detection, Semantic Segmentation	2D box-level	Camera: RGB image; Radar: ADC signal, range-azimuth-Doppler tensor, range-azimuth tensor, range-Doppler tensor, point cloud	1 class (Vehicle)	8,252 frames are labelled with 9,550 vehicle	<a href="https://github.com/valeoai/RADial">https://github.com/valeoai/RADial</a>
VoD [34]	2022	Object Detection	2D box-level, 3D box-level	Camera: RGB image; Radar: point cloud	13 classes (Car, Pedestrian, Cyclist, Rider, Unused Bicycle, Bicycle Rack, Human Depiction, Moped or Scooter, Motor, Ride, Other, Vehicle, Other, Truck, Ride Uncertain)	8693 frames, 123,106 annotations of both moving and static objects, including 26,587 pedestrian, 10,800 cyclist and 26,949 car labels	<a href="https://tudelft-iv.github.io/view-of-delft-dataset">https://tudelft-iv.github.io/view-of-delft-dataset</a>
Boreas [173]	2022	Object Detection, Localization, Odometry	2D box-level	Camera: RGB image; Radar: range-azimuth tensor	4 classes (Car, Pedestrian, Cyclist, Misc)	7.1k frames for detection, over 350km of driving data, 326,180 unique 3D box annotations	<a href="https://www.boreas.utias.utoronto.ca">https://www.boreas.utias.utoronto.ca</a>
TJ4DRadSet [176]	2022	Object Detection, Object Tracking	3D box-level	Camera: RGB image; Radar: point cloud	8 classes (Car, Pedestrian, Cyclist, Bus, Motorcyclist, Truck, Engineering Vehicle, Tricycle, Tricyclist)	40k frames in total, 7757 frames within 44 consecutive sequences	<a href="https://github.com/TJRadarLab/TJ4DRadSet">https://github.com/TJRadarLab/TJ4DRadSet</a>
K-Radar [177]	2022	Object Detection, Object Tracking, SLAM	3D box-level	Camera: RGB image; Radar: range-azimuth-Doppler tensor	5 classes (Pedestrian, Motorbike, Bicycle, Sedan, Bus or Truck)	35k frames of 4D radar tensor	<a href="https://github.com/kaist-avelab/k-radar">https://github.com/kaist-avelab/k-radar</a>
aiMotive [178]	2022	Object Detection	3D box-level	Camera: RGB image; Radar: point cloud	14 classes (Pedestrian, Car, Bus, Truck, Van, Motorcycle, Pickup, Rider, Bicycle, Trailer, Train, Shopping Cart, Other Object)	26,583 frames, 425k objects	<a href="https://github.com/aimotive/aimotive_dataset">https://github.com/aimotive/aimotive_dataset</a>
WaterScenes [174]	2023	Object Detection, Instance Segmentation, Semantic Segmentation, Free-space Segmentation, Waterline Segmentation, Panoptic Perception	2D box-level	Camera: RGB image; Radar: point cloud	7 classes (Pier, Buoy, Sailor, Ship, Boat, Vessel, Kayak)	54,120 frames, 202k objects	<a href="https://waterscenes.github.io">https://waterscenes.github.io</a>

**TABLE III**  
**SUMMARY OF RADAR-CAMERA FUSION METHODS**

Reference	Year	Task	Annotations	Categories	Modalities	Network Architecture	Fusion Level	Fusion Operation	Dataset	Source Code
Chadwick <i>et al.</i> [186]	2019	Object Detection	2D box-level	Vehicle	Camera: RGB image; Radar: point cloud	One-stage network based on ResNet [99]	Feature-level	Addition; Concatenation	Self-recorded	-
RRPN [9]	2019	Object Detection	2D box-level	Car, Truck, Person, Motorcycle, Bicycle, Bus	Camera: RGB image; Radar: point cloud	RRPN [9]	Data-level	Transformation matrix	nuScenes [5]	<a href="https://github.com/mrnabati/RRPN">https://github.com/mrnabati/RRPN</a>
Jha <i>et al.</i> [209]	2019	Object Detection	2D box-level	Pedestrian, Lamp Post	Camera: RGB image; Radar: point cloud	YOLOv3 [77]	Object-level	Transformation matrix	Self-recorded	-
CMGGAN [307]	2019	Semantic Segmentation	2D point-level	Free Space	Camera: RGB image; Radar: grid map	CMGGAN [307]	Feature-level	Addition	Self-recorded	-
Meyer and Kuschk [317]	2019	Object Detection	3D box-level	Car	Camera: RGB image; Radar: point cloud	A 3D region proposal network based on VGG [212]	Data-Level	Transformation matrix	Astyx [317]	-
RVNet [28]	2019	Object Detection	2D box-level	Vehicle, Pedestrian, Two-wheelers, Objects (movable objects and debris)	Camera: RGB image; Radar: point cloud	RVNet [28] based on YOLOv3 [77]	Feature-level	Concatenation	nuScenes [5]	-
FusionNet [206]	2019	Object Detection, Object Classification	2D box-level	Vehicle	Camera: RGB image; Radar: range-azimuth tensor	FusionNet [206] inspired by SSD [83]	Feature-level	Concatenation	nuScenes [5]	-
SO-Net [199]	2020	Object Detection, Semantic Segmentation	2D box-level, 2D pixel-level	Vehicle, Free Space	Camera: RGB image; Radar: point cloud	SO-Net [199] based on the RVNet [28]	Feature-level	Concatenation	nuScenes [5]	-
SAF-FCOS [214]	2020	Object Detection	2D box-level	Bicycle, Car, Motorcycle, Bus, Train, Truck	Camera: RGB image; Radar: point cloud	SAF-FCOS [214] based on FCOS [318]	Feature-level	Addition; Multiplication	nuScenes [5]	<a href="https://github.com/Singingkettle/SAF-FCOS">https://github.com/Singingkettle/SAF-FCOS</a>
CRF-Net [211]	2020	Object Detection	2D box-level	Car, Bus, Motorcycle, Truck, Trailer, Bicycle, Human	Camera: RGB image; Radar: point cloud	CRF-Net [211] based on RetinaNet [84]	Data-level	Concatenation	nuScenes [5], Self-recorded	<a href="https://github.com/TUMFTM/CameraRadarFusionNet">https://github.com/TUMFTM/CameraRadarFusionNet</a>
Bijelic <i>et al.</i> [53]	2020	Object Detection	2D box-level	Vehicle	Camera: RGB image; Radar: point cloud	A modified VGG [212] backbone and SSD [83] blocks	Feature-level	Concatenation; Attention	DENSE [53]	<a href="https://github.com/princeton-computational-imaging/SeeingThroughFog">https://github.com/princeton-computational-imaging/SeeingThroughFog</a>
BIRANet [190]	2020	Object Detection	2D box-level	Car, Truck, Person, Motorcycle, Bicycle, Bus	Camera: RGB image; Radar: point cloud	RANet and BIRANet [190] based on ResNet [99]	Feature-level	Addition	nuScenes [5]	<a href="https://github.com/RituYadav92/Radar-RGB-Attentive-Multimodal-Object-Detection">https://github.com/RituYadav92/Radar-RGB-Attentive-Multimodal-Object-Detection</a>
Nabati and Qi [194]	2020	Object Detection, Depth Estimation	2D box-level	Car, Truck, Person, Motorcycle, Bicycle, Bus	Camera: RGB image; Radar: point cloud	FPN [74] with ResNet [99] as backbone, and RPN in Faster R-CNN [73]	Hybrid-level	-	nuScenes [5]	-
YODar [197]	2020	Object Detection	2D box-level	Vehicle	Camera: RGB image; Radar: point cloud	YODar [197] based on YOLOv3 [77]	Feature-level	Concatenation	nuScenes [5]	-
CenterFusion [18]	2020	Object Detection	3D box-level	Car, Truck, Bus, Trailer, Pedestrian, Barrier, Motorcycle, Bicycle, Traffic Cone	Camera: RGB image; Radar: point cloud	CenterNet [189] with the DLA [319] backbone	Feature-level	Concatenation	nuScenes [5]	<a href="https://github.com/mrnabati/CenterFusion">https://github.com/mrnabati/CenterFusion</a>
RODNet [17]	2021	Object Detection	2D box-level	Pedestrian, Cyclist, Car	Camera: RGB image; Radar: range-azimuth tensor	RODNet [17]	Feature-level	-	CRUW [170]	<a href="https://github.com/yizhou-wang/RODNet">https://github.com/yizhou-wang/RODNet</a>
RAMP-CNN [104]	2021	Object Detection	2D box-level	Pedestrian, Cyclist, Car	Camera: RGB image; Radar: range-azimuth-Doppler tensor	RAMP-CNN [104]	Feature-level	Concatenation	CRUW [170]	-
Li and Xie [196]	2021	Object Detection	3D box-level	Vehicle	Camera: RGB image; Radar: point cloud	A network based on YOLOv3 [77]	Feature-level	Concatenation; Multiplication	nuScenes [5]	-
Kim <i>et al.</i> [210]	2021	Object Detection	3D box-level	Vehicle	Camera: RGB image; Radar: range-azimuth tensor	A network based on VGG [212] and FPN [74]	Feature-level	Concatenation	Self-recorded	-
AssociationNet [200]	2021	Object Detection	2D box-level	Vehicle	Camera: RGB image; Radar: point cloud	AssociationNet [200]	Object-level	Transformation matrix; Concatenation	Self-recorded	-
RVF-Net [243]	2021	Object Detection	3D box-level	Car	Camera: RGB image; Radar: point cloud	RVF-Net [243] based on VoxelNet [117]	Data-level	Concatenation	nuScenes [5]	<a href="https://github.com/TUMFTM/RadarVoxelFusionNet">https://github.com/TUMFTM/RadarVoxelFusionNet</a>

TABLE III  
(CONTINUED).

Reference	Year	Task	Annotations	Categories	Modalities	Network Architecture	Fusion Level	Fusion Operation	Dataset	Source Code
Cui <i>et al.</i> [195]	2021	Object Detection	3D box-level	Car	Camera: RGB image; Radar: point cloud	CNN with [217] block	SSMA	Hybrid-level	Concatenation	Astyx [317]
RISFNet [198]	2021	Object Detection	2D box-level	Bottle	Camera: RGB image; Radar: point cloud	RISFNet [198] based on CSPdarknet53 [78] and VGG [212]	Feature-level	Concatenation; Addition; Multiplication	FloW [172]	-
GRIF Net [245]	2021	Object Detection	3D box-level	Vehicle	Camera: RGB image; Radar: point cloud	GRIF Net [245] based on FPN [74] and SBN-Net [246]	Feature level	Attention	nuScenes [5]	-
Stäcker <i>et al.</i> [255]	2022	Object Detection	2D box-level	Car, Person, Truck, Bicycle, Motorcycle	Camera: RGB image; Radar: point cloud	A network based on RetinaNet [84] architecture with a ResNet [99] backbone	Feature level	Addition, Concatenation	nuScenes [5]	-
FUTR3D [320]	2022	Object Detection	3D box-level	Car, Truck, Bus, Pedestrian, Barrier, Trailer, Construction Vehicle, Motorcycle, Bicycle, Traffic cone	Camera: RGB image; Radar: point cloud	FUTR3D [207] with ResNet-101 [99] as backbone and FPN [74] as neck	Feature-level	Concatenation	nuScenes [5]	<a href="https://github.com/Tsinghua-MAR-S-Lab/futr3d">https://github.com/Tsinghua-MAR-S-Lab/futr3d</a>
Simple-BEV [203]	2022	Semantic Segmentation	2D pixel-level	Vehicle, Background	Camera: RGB image; Radar: point cloud	A network with a ResNet [99] backbone	Feature-level	Concatenation	nuScenes [5]	-
RadSegNet [202]	2022	Object Detection, Semantic Segmentation	2D box-level, 2D pixel-level	Car, Truck	Camera: RGB image; Radar: point cloud, range-azimuth tensor	RadSegNet [202]	Data-level	Concatenation	Astyx [317], RADIATE [168]	-
RCBEV [207]	2022	Object Detection	3D box-level	Car, Truck, Bus, Pedestrian, Barrier, Trailer, Construction Vehicle, Motorcycle, Bicycle, Traffic cone	Camera: RGB image; Radar: point cloud	RCBEV [207] with Swin Transformer [89] as backbone and FPN [74] as neck	Feature-level	Concatenation	nuScenes [5]	-
CRAFT [248]	2022	Object Detection	3D box-level	Car, Truck, Bus, Pedestrian, Barrier, Trailer, Construction Vehicle, Motorcycle, Bicycle, Traffic cone	Camera: RGB image; Radar: point cloud	CRAFT [248] based on DLA [319]	Data-level	Concatenation	nuScenes [5]	-
DeepFusion [321]	2022	Object Detection	3D box-level	Car, Truck, Bus, Pedestrian, Barrier, Trailer, Construction Vehicle, Motorcycle, Bicycle, Traffic cone	Camera: RGB image; Radar: point cloud	DeepFusion [321]	Feature-level	Concatenation	Self-recorded, nuScenes [5]	-
CramNet [253]	2022	Object Detection	3D box-level	Car, Van, Truck, Bus, Motorbike, Bicycle	Camera: RGB image; Radar: range-azimuth tensor	CramNet [253]	Feature-level	Attention	RADIATE [168]	-
MVFusion [249]	2023	Object Detection	3D box-level	Car, Truck, Bus, Pedestrian, Barrier, Trailer, Construction Vehicle, Motorcycle, Bicycle, Traffic cone	Camera: RGB image; Radar: point cloud	MVFusion [249]	Feature-level	Addition; Concatenation	nuScenes [5]	-
CRN [250]	2023	Object Detection	3D box-level	Car, Truck, Bus, Pedestrian, Barrier, Trailer, Construction Vehicle, Motorcycle, Bicycle, Traffic cone	Camera: RGB image; Radar: point cloud	CRN [250] based on ResNet [99], ConvNeXt [322] and FPN [74]	Feature-level	Concatenation	nuScenes [5]	-
RCFusion [251]	2023	Object Detection	3D box-level	Car, Truck, Pedestrian, Cyclist	Camera: RGB image; Radar: point cloud	RCFusion [251]	Feature-level	Concatenation; Multiplication; Attention	VoD [34], TJ4DRadSet [176]	-
LXL [247]	2023	Object Detection	3D box-level	Car, Pedestrian, Cyclist	Camera: RGB image; Radar: point cloud	LXL [247]	Feature-level	Concatenation; Multiplication	VoD [34], TJ4DRadSet [176]	-
Achelous [323]	2023	Object Detection, Semantic Segmentation, Free-space Segmentation, Waterline Segmentation, Point Cloud Segmentation	2D box-level, 2D pixel-level, 3D point-level	Pier, Buoy, Sailor, Ship, Boat, Vessel, Kayak	Camera: RGB image; Radar: point cloud	Achelous [323]	Data-level	Concatenation	WaterScenes [174]	<a href="https://github.com/GuanRunwei/Achelous">https://github.com/GuanRunwei/Achelous</a>

TABLE IV  
SUMMARY OF RADAR-CAMERA FUSION EVALUATION METRICS

Metric	Formula	Definition
Accuracy	$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$ (1)	Accuracy is the number of correct predictions over all predictions.
Precision	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$ (2)	Precision is the fraction of true positive among total predicted positive.
Recall	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$ (3)	Recall is the fraction of true positive over all actual positive in the dataset.
F1-Score	$\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$ (4)	F1-Score is the harmonic mean of precision and recall, describing a balance between precision and recall.
Average Precision (AP)	$\text{AP} = \int_0^1 \text{precision}(r)dr$ (5)	AP is the precision averaged over all recall values between 0 and 1 for a single class. It is the area under the Precision-Recall curve. <ul style="list-style-type: none"> <li>• <math>r</math>: recall value</li> <li>• <math>\text{precision}(r)</math>: the precision at recall value of <math>r</math></li> </ul>
Average Recall (AR)	$\text{AR} = 2 \int_{0.5}^1 \text{recall}(\alpha)d\alpha$ (6)	AR is the average of all recalls at IoU thresholds from 0.5 to 1.0. It is twice the area under the Recall-IoU curve. <ul style="list-style-type: none"> <li>• <math>\alpha</math>: the IoU overlap</li> <li>• <math>\text{recall}(\alpha)</math>: the recall at IoU value of <math>\alpha</math></li> </ul>
Frame Per Second (FPS)	$\text{FPS} = \frac{m}{s}$ (7)	FPS is a measure of how many images the model processes per second. <ul style="list-style-type: none"> <li>• <math>m</math>: the number of images</li> <li>• <math>s</math>: total seconds consumed</li> </ul>
Mean Average Precision (mAP)	$\text{mAP} = \frac{1}{N} \sum_{i=1}^N \text{AP}_i$ (8)	mAP is the average value of AP, that is, the average of the area under the Precision-Recall curve of each category. <ul style="list-style-type: none"> <li>• <math>N</math>: the number of classes</li> <li>• <math>\text{AP}_i</math>: AP value of the <math>i</math>th class</li> </ul>
Mean Intersection over Union (mIoU)	$\text{mIoU} = \frac{1}{N+1} \sum_{i=0}^N \frac{\text{TP}}{\text{FN} + \text{FP} + \text{TP}}$ (9)	IoU is the overlap between the predicted value and the ground truth divided by the area of union. Then, mIoU is the average value of IoU over all classes.
nuScenes Detection Score (NDS) [5]	$\text{NDS} = \frac{1}{10} \left[ 5\text{mAP} + \sum_{\text{mTP} \in \text{TP}} (1 - \min(1, \text{mTP})) \right]$ (10)	NDS is a weighted sum of mAP and five TP metrics. <ul style="list-style-type: none"> <li>• mAP: mean Average Precision over all classes</li> <li>• TP: the set of the five mean True Positive metrics, including box location, size, orientation, attributes, and velocity</li> <li>• mTP: the mean True Positive over all classes</li> <li>• C: the set of classes</li> </ul>
Object Location Similarity (OLS) [17]	$\text{OLS} = \exp \left\{ \frac{-d^2}{2(s\kappa_{cls})^2} \right\}$ (12)	OLS describes the correlation between two detections related to distance, classes and scale information. <ul style="list-style-type: none"> <li>• <math>d</math>: distance between two points in an RA tensor</li> <li>• <math>s</math>: the distance between the object and the radar sensor, indicating object scale information</li> <li>• <math>\kappa_{cls}</math>: a constant value that denotes the error tolerance for each class <math>cls</math>, which can be calculated based on the average object size of that class</li> </ul>
Average Heading Similarity (AHS) [7], [163]	$\text{AHS} = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} \max_{\tilde{r}: \tilde{r} \geq r} s(\tilde{r})$ (13)	AHS is the average orientation accuracy in 3D IOU and global orientation angle. <ul style="list-style-type: none"> <li>• <math>r</math>: recall value</li> <li>• <math>\mathcal{D}(r)</math>: all object detections at recall rate <math>r</math></li> <li>• <math>\Delta_\theta^{(i)}</math>: difference in global orientation of detection <math>i</math> as determined by the estimated and ground truth orientation</li> <li>• <math>\delta_i</math>: whether detection <math>i</math> is assigned to a ground truth bounding box</li> </ul>
	$s(r) = \frac{1}{ \mathcal{D}(r) } \sum_{i \in \mathcal{D}(r)} \frac{1 + \cos \Delta_\theta^{(i)}}{2} - \delta_i$ (14)	

**TABLE V**  
**PERFORMANCE OVERVIEW OF RADAR-CAMERA METHODS**

Method	Dataset	Metrics				Threshold	Sub-dataset	Hardware
		AP/AR	mAP	Others	Inference Time			
RRPN [9]	nuScenes	AP(NS-F): 43.0 AP <sup>&gt;0</sup> (NS-F): 64.9 AP <sup>&gt;75</sup> (NS-F): 48.5 AP(NS-FB): 35.5 AP <sup>&gt;0</sup> (NS-FB): 59.0 AP <sup>&gt;75</sup> (NS-FB): 37.0 AR(NS-F): 48.6 AR(NS-FB): 42.1	-	-	-	IoU={0.5, 0.75}	a) NS-F sub-dataset: from front camera and front radar only, with 23k samples b) NS-FB sub-dataset: from the rear camera and two rear radars, with 45k samples	-
RVNet [28]	nuScenes	AP(Cycle): 20.0 AP(Pedestrian): 14.0 AP(Vehicle): 59.0 AP(Obstacle): 26.0	25.0	-	17 ms	IoU={0.5}	Samples from front camera and front radar	One NVIDIA GeForce 1080 GPU
SO-Net [199]	nuScenes	AP(detection): 42.3 AP(segmentation): 99.1	-	-	25 ms	IoU={0.5}	Samples under rainy and nighttime conditions (308 pairs for training and 114 pairs for testing)	One NVIDIA GeForce 1080 GPU
SAF-FCOS [214]	nuScenes	AP: 72.4 AP <sup>&gt;0</sup> : 90.0 AP <sup>&gt;75</sup> : 79.3 AR: 79.0	-	-	-	IoU={0.5, 0.75, 0.5-0.95}	A total of 34,149 radar-camera pairs	Eight NVIDIA GeForce GTX 1080Ti GPUs
CRF-Net [211]	nuScenes	-	55.23	-	43 ms	-	Merge the original 23 object classes into seven classes	One NVIDIA Titan XP GPU
BIRANet [190]	nuScenes	AP: 72.3 AP <sup>&gt;0</sup> : 88.9 AP <sup>&gt;75</sup> : 84.3 AP <sup>&gt;85</sup> : 65.7 AR: 75.3	-	-	-	IoU={0.5, 0.75, 0.85}	Merged relevant classes into six classes	One NVIDIA Titan Pascal GPU
Nabati and Qi [194]	nuScenes	AP: 35.6 AP <sup>&gt;0</sup> : 60.53 AP <sup>&gt;75</sup> : 37.38 AR: 42.1	-	MAE: 2.65	-	IoU={0.5, 0.75}	Samples from front and rear cameras together with all radars	Two NVIDIA Quadro P6000 GPUs
YODar [197]	nuScenes	AP: 43.1	39.4	-	-	-	Samples from nuScenes frames for training, 3,289 frames for validation and 1,006 frames for testing)	One NVIDIA Quadro P6000 GPU
CenterFusion [18]	nuScenes	-	32.6	NDS: 44.9 mATE: 63.1 mASE: 26.1 mAOE: 51.6 mAVE: 61.4 mAAE: 11.5	-	Distance={0.5, 1, 2, 4}	Complete nuScenes	Two NVIDIA P5000 GPUs
Li and Xie [196]	nuScenes	AP: 24.3 AP <sup>&gt;0</sup> : 48.4 AP <sup>&gt;75</sup> : 22.3 AR: 33.7	48.4	-	-	IoU={0.5, 0.75}	Dataset is randomly divided into a training set, a validation set, and a testing set according to the ratio of 6:2:2	One NVIDIA GeForce GTX 1080Ti GPU
RVF-Net [243]	nuScenes	AP: 54.86	-	-	44 ms	Distance={0.5}	Samples under rainy and nighttime conditions	One NVIDIA Titan XP GPU
GRIF Net [245]	nuScenes	AP <sup>0.5m</sup> : 44.1 AP <sup>1m</sup> : 66.5 AP <sup>2m</sup> : 71.9 AP <sup>4m</sup> : 74.9	-	-	-	Distance={0.5, 1, 2, 4}	One front camera and 3 front radars	One NVIDIA GeForce GTX 1080Ti GPU
Stäcker <i>et al.</i> [255]	nuScenes	-	36.78	-	36.7 ms	IoU={0.5}	Sample from front camera and radar	One NVIDIA GeForce RTX 2080 GPU
FUTR3D [320]	nuScenes	-	39.9	NDS: 50.8	-	Distance={0.5, 1, 2, 4}	Complete nuScenes	-
RCBEV [207]	nuScenes	-	40.6	NDS: 45.6 mATE: 48.4 mASE: 25.7 mAOE: 58.7 mAVE: 70.2 mAAE: 14.0	-	Distance={0.5, 1, 2, 4}	Complete nuScenes	Four NVIDIA GeForce GTX 3090 GPUs
CRAFT [248]	nuScenes	-	41.1	NDS: 52.3 mATE: 46.7 mASE: 26.8 mAOE: 45.3 mAVE: 51.9 mAAE: 11.4	4.1 FPS	Distance={0.5, 1, 2, 4}	Complete nuScenes	Training: four NVIDIA GeForce RTX 3090 GPUs; Testing: one RTX 3090 GPU
MVFusion [249]	nuScenes	-	45.3	NDS: 51.7 mATE: 56.9 mASE: 24.6 mAOE: 37.9 mAVE: 78.1 mAAE: 12.8	-	Distance={0.5, 1, 2, 4}	Complete nuScenes	Eight NVIDIA RTX A6000 GPUs
CRN [250]	nuScenes	-	57.5	NDS: 62.4 mATE: 46.0 mASE: 27.3 mAOE: 44.3 mAVE: 35.2 mAAE: 18.0	7.2 FPS	Distance={0.5, 1, 2, 4}	Complete nuScenes	-
RCFusion [251]	VoD	-	49.65	-	-	IoU={0.25, 0.5}	Complete VoD	NVIDIA GeForce RTX 3090 GPUs
	TJ4DRadSet	-	33.85	BEV mAP: 39.76	10.8 FPS	IoU={0.25, 0.5}	Complete TJ4DRadSet	
LXL [247]	VoD	-	56.31	-	-	IoU={0.25, 0.5}	Complete VoD	-
	TJ4DRadSet	-	36.32	BEV mAP: 41.20	-	IoU={0.25, 0.5}	Complete TJ4DRadSet	

## REFERENCES

- [1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016.
- [2] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multi-modal end-to-end autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 537–547, Jan. 2022.
- [3] D. Feng et al., "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.
- [4] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-task multi-sensor fusion for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7345–7353.
- [5] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11621–11631.
- [6] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1907–1915.
- [7] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–8.
- [8] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D lidar using fully convolutional network," 2016, *arXiv:1608.07916*.
- [9] R. Nabati and H. Qi, "RRPN: Radar region proposal network for object detection in autonomous vehicles," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 3093–3097.
- [10] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhori, "Sensor fusion for semantic segmentation of urban scenes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1850–1857.
- [11] R. Yin, Y. Cheng, H. Wu, Y. Song, B. Yu, and R. Niu, "Fusionlane: Multi-sensor fusion for lane marking semantic segmentation using deep neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1543–1553, Feb. 2022.
- [12] A. Asvadi, P. Girao, P. Peixoto, and U. Nunes, "3D object tracking using RGB and LIDAR data," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst.*, 2016, pp. 1255–1260.
- [13] Y. Fang, H. Zhao, H. Zha, X. Zhao, and W. Yao, "Camera and LiDAR fusion for on-road vehicle tracking with reinforcement learning," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 1723–1730.
- [14] K. Yoneda, N. Suganuma, R. Yanase, and M. Aldibaja, "Automated driving recognition technologies for adverse weather conditions," *IATSS Res.*, vol. 43, no. 4, pp. 253–262, 2019.
- [15] P. Li, P. Wang, K. Berntorp, and H. Liu, "Exploiting temporal relations on radar perception for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17071–17080.
- [16] B. Major et al., "Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 924–932.
- [17] Y. Wang, Z. Jiang, Y. Li, J.-N. Hwang, G. Xing, and H. Liu, "RODNet: A real-time radar object detection network cross-supervised by camera-radar fused object 3D localization," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 4, pp. 954–967, Jun. 2021.
- [18] R. Nabati and H. Qi, "CenterFusion: Center-based radar and camera fusion for 3D object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1527–1536.
- [19] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.
- [20] Y. Cui et al., "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 722–739, Feb. 2022.
- [21] K. Huang, B. Shi, X. Li, X. Li, S. Huang, and Y. Li, "Multi-modal sensor fusion for auto driving perception: A survey," 2022, *arXiv:2202.02703*.
- [22] S. Jusoh and S. Almajali, "A systematic review on fusion techniques and approaches used in applications," *IEEE Access*, vol. 8, pp. 14424–14439, 2020.
- [23] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *IEEE Access*, vol. 8, pp. 2847–2868, 2020.
- [24] Z. Wei, F. Zhang, S. Chang, Y. Liu, H. Wu, and Z. Feng, "mmWave radar and vision fusion for object detection in autonomous driving: A review," *Sensors*, vol. 22, no. 7, 2022, Art. no. 2542.
- [25] C. Iovescu and S. Rao, "The fundamentals of millimeter wave sensors," *Texas Instrum.*, pp. 1–8, 2017.
- [26] R. Appleby and R. N. Anderton, "Millimeter-wave and submillimeter-wave imaging for security and surveillance," *Proc. IEEE*, vol. 95, no. 8, pp. 1683–1690, Aug. 2007.
- [27] P. Fritzsche, S. Kueppers, G. Briese, and B. Wagner, "Radar and LiDAR sensorfusion in low visibility environments," in *Proc. 13th Int. Conf. Inform. Control, Automat. Robot.*, 2016, pp. 30–36.
- [28] V. John and S. Mita, "RVNet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments," in *Proc. Image Video Technol.: 9th Pacific-Rim Symp.*, 2019, pp. 351–364.
- [29] Y. Long, D. Morris, X. Liu, M. Castro, P. Chakravarty, and P. Narayanan, "Full-velocity radar returns by radar-camera fusion," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16198–16207.
- [30] Y. Li, J. Deng, Y. Zhang, J. Ji, H. Li, and Y. Zhang, "EZFusion: A close look at the integration of LiDAR, millimeter-wave radar, and camera for accurate 3D object detection and tracking," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 11182–11189, Oct. 2022.
- [31] H. Griffiths et al., "Radar spectrum engineering and management: Technical and regulatory issues," *Proc. IEEE*, vol. 103, no. 1, pp. 85–102, Jan. 2015.
- [32] J. Kopp, D. Kellner, A. Piroli, and K. Dietmayer, "Tackling clutter in radar data-label generation and detection using pointnet," 2023, *arXiv:2303.09530*.
- [33] A. Ouaknine, A. Newson, J. Rebut, F. Tupin, and P. Pérez, "Cararda dataset: Camera and automotive radar with range-angle-doppler annotations," in *Proc. IEEE 25th Int. Conf. Pattern Recognit.*, 2021, pp. 5068–5075.
- [34] A. Palffy, E. Pool, S. Baratam, J. F. P. Kooij, and D. M. Gavrila, "Multi-class road user detection with 3+1D radar in the view-of-delft dataset," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4961–4968, Apr. 2022.
- [35] D. Gusland, J. M. Christiansen, B. Torvik, F. Fioranelli, S. Z. Gurbuz, and M. Ritchie, "Open radar initiative: Large scale dataset for benchmarking of micro-doppler recognition algorithms," in *Proc. IEEE Radar Conf.*, 2021, pp. 1–6.
- [36] F. E. Nowruzi et al., "Deep open space segmentation using automotive radar," in *Proc. IEEE MTT-S Int. Conf. Microw. Intell. Mobility*, 2020, pp. 1–4.
- [37] B. Yang, I. Khatri, M. Happold, and C. Chen, "ADCNet: End-to-end perception with raw radar ADC data," 2023, *arXiv:2303.11420*.
- [38] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Semantic segmentation on radar point clouds," in *Proc. IEEE 21st Int. Conf. Inf. Fusion*, 2018, pp. 2179–2186.
- [39] F. Fent, P. Bauerschmidt, and M. Lienkamp, "RadarGNN: Transformation invariant graph neural network for radar-based perception," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 182–191.
- [40] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 4, pp. 608–621, Jul. 1983.
- [41] P. P. Gandhi and S. A. Kassam, "Analysis of CFAR processors in nonhomogeneous background," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 24, no. 4, pp. 427–445, Jul. 1988.
- [42] K. Werber et al., "Automotive radar gridmap representations," in *Proc. IEEE MTT-S Int. Conf. Microw. Intell. Mobility*, 2015, pp. 1–4.
- [43] L. Sless, B. El Shlomo, G. Cohen, and S. Oron, "Road scene understanding by occupancy grid learning from sparse radar clusters using semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 867–875.
- [44] O. Schumann, J. Lombacher, M. Hahn, C. Wöhler, and J. Dickmann, "Scene understanding with automotive radar," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 188–203, Jun. 2020.
- [45] J. Lombacher, M. Hahn, J. Dickmann, and C. Wöhler, "Detection of arbitrarily rotated parked cars based on radar sensors," in *Proc. IEEE 16th Int. Radar Symp.*, 2015, pp. 180–185.
- [46] S. Chen, W. He, J. Ren, and X. Jiang, "Attention-based dual-stream vision transformer for radar gait recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 3668–3672.
- [47] Wikipedia contributors, "Image sensor – Wikipedia, the free encyclopedia," Mar. 24, 2023. Accessed: Apr. 10, 2023. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Image\\_sensor&oldid=1146373856](https://en.wikipedia.org/w/index.php?title=Image_sensor&oldid=1146373856)
- [48] C. Michaelis et al., "Benchmarking robustness in object detection: Autonomous driving when winter is coming," 2019, *arXiv:1907.07484*.

- [49] X. Xu, Y. Ma, and W. Sun, "Towards real scene super-resolution with raw images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1723–1731.
- [50] X. Liu, K. Shi, Z. Wang, and J. Chen, "Exploit camera raw data for video super-resolution via hidden Markov model inference," *IEEE Trans. Image Process.*, vol. 30, pp. 2127–2140, 2021.
- [51] H. Song, W. Choi, and H. Kim, "Robust vision-based relative-localization approach using an RGB-depth camera and lidar sensor fusion," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3725–3736, Jun. 2016.
- [52] S. Inbar and O. David, "Laser gated camera imaging system and method," U.S Patent 7,379,164, May 27, 2008.
- [53] M. Bijelic et al., "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11682–11692.
- [54] G. Gallego et al., "Event-based vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.
- [55] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, 2019, Art. no. 648.
- [56] S. Saponara and B. Neri, "Radar sensor signal acquisition and multi-dimensional FFT processing for surveillance applications in transport systems," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 4, pp. 604–615, Apr. 2017.
- [57] S. Sun, A. P. Petropulu, and H. V. Poor, "MIMO radar for advanced driver-assistance systems and autonomous driving: Advantages and challenges," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 98–117, Jul. 2020.
- [58] B. Li, "3D fully convolutional network for vehicle detection in point cloud," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1513–1518.
- [59] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2147–2156.
- [60] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Towards reaching human performance in pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 973–986, Apr. 2018.
- [61] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster R-CNN doing well for pedestrian detection?," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 443–457.
- [62] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals using stereo imagery for accurate object class detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1259–1272, May 2018.
- [63] M. Weber, P. Wolf, and J. M. Zöllner, "DeepTLR: A single deep convolutional network for detection and classification of traffic lights," in *Proc. IEEE Intell. Veh. Symp.*, 2016, pp. 342–348.
- [64] J. Müller and K. Dietmayer, "Detecting traffic lights by single shot detection," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 266–273.
- [65] M. Bach, S. Reuter, and K. Dietmayer, "Multi-camera traffic light recognition using a classifying labeled multi-bernoulli filter," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1045–1051.
- [66] K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1370–1377.
- [67] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2110–2118.
- [68] H. S. Lee and K. Kim, "Simultaneous traffic sign detection and boundary estimation using convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1652–1663, May 2018.
- [69] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan, "Traffic sign recognition using a multi-task convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1100–1111, Apr. 2018.
- [70] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
- [71] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
- [72] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
- [73] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. 28th Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [74] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.
- [75] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
- [76] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.
- [77] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [78] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [79] G. Jocher, "YOLOv5 by ultralytics," 2020. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [80] C. Li et al., "YOLOV6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.
- [81] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 7464–7475.
- [82] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by ultralytics," Jan. 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [83] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
- [84] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.
- [85] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.
- [86] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.
- [87] W. Lv et al., "Detrs beat yolos on real-time object detection," 2023, *arXiv:2304.08069*.
- [88] F. Liu, H. Wei, W. Zhao, G. Li, J. Peng, and Z. Li, "WB-DETR: Transformer-based detector without backbone," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 2979–2987.
- [89] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
- [90] Y. Fang et al., "You only look at one sequence: Rethinking transformer in vision through object detection," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021, pp. 26183–26197.
- [91] Z. Peng et al., "Conformer: Local features coupling global representations for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 367–376.
- [92] J. Pan et al., "EdgeViTs: Competing light-weight CNNs on mobile devices with vision transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 294–311.
- [93] S. Mehta and M. Rastegari, "Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.
- [94] Y. Xu, Q. Zhang, J. Zhang, and D. Tao, "Vitae: Vision transformer advanced by exploring intrinsic inductive bias," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 28522–28535.
- [95] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, "Visformer: The vision-friendly transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 589–598.
- [96] A. Palffy, J. Dong, J. F. P. Kooij, and D. M. Gavrila, "CNN based road user detection using the 3D radar cube," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1263–1270, Apr. 2020.
- [97] A. Zhang, F. E. Nowruzi, and R. Laganiere, "Raddet: Range-azimuth-doppler based radar object detection for dynamic road users," in *Proc. IEEE 18th Conf. Robots Vis.*, 2021, pp. 95–102.
- [98] P. Svenssonsson, F. Fioranelli, and A. Yarovoy, "Radar-PointGNN: Graph based object recognition for unstructured radar point-cloud data," in *Proc. IEEE Radar Conf.*, 2021, pp. 1–6.
- [99] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

- [100] X. Gao, G. Xing, S. Roy, and H. Liu, "Experiments with mmWave automotive radar test-bed," in *Proc. IEEE 53rd Asilomar Conf. Signals, Syst., Comput.*, 2019, pp. 1–6.
- [101] X. Dong, P. Wang, P. Zhang, and L. Liu, "Probabilistic oriented object detection in automotive radar," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 102–103.
- [102] W. Ng, G. Wang, Siddhartha, Z. Lin, and B. J. Dutta, "Range-Doppler detection in automotive radar with deep learning," in *Proc. 20th Int. Joint Conf. Neural Netw.*, Glasgow, UK, 2020, pp. 1–8, doi: [10.1109/IJCNN48605.2020.9207080](https://doi.org/10.1109/IJCNN48605.2020.9207080).
- [103] C. Decourt, R. VanRullen, D. Salle, and T. Oberlin, "Darod: A deep automotive radar object detector on range-doppler maps," in *Proc. IEEE Intell. Veh. Symp.*, 2022, pp. 112–118.
- [104] X. Gao, G. Xing, S. Roy, and H. Liu, "RAMP-CNN: A novel neural network for enhanced automotive radar object recognition," *IEEE Sensors J.*, vol. 21, no. 4, pp. 5119–5132, Feb. 2021.
- [105] J. Rebut, A. Ouaknine, W. Malik, and P. Pérez, "Raw high-definition radar for multi-task learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17021–17030.
- [106] A. Danzer, T. Griebel, M. Bach, and K. Dietmayer, "2D car detection in radar data with pointnets," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 61–66.
- [107] J. F. Tilly et al., "Detection and tracking on automotive radar data with deep learning," in *Proc. IEEE 23rd Int. Conf. Inf. Fusion*, 2020, pp. 1–7.
- [108] N. Scheiner et al., "Seeing around street corners: Non-line-of-sight detection and tracking in-the-wild using Doppler radar," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2068–2077.
- [109] A. Dubey, A. Santra, J. Fuchs, M. Lübke, R. Weigel, and F. Lurz, "HaradNet: Anchor-free target detection for radar point clouds using hierarchical attention and multi-task learning," *Mach. Learn. Appl.*, vol. 8, 2022, Art. no. 100275.
- [110] B. Tan et al., "3D object detection for multi-frame 4D automotive millimeter-wave radar point cloud," *IEEE Sensors J.*, vol. 23, no. 11, pp. 11125–11138, Jun. 2023.
- [111] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
- [112] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.
- [113] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.
- [114] M. Dreher, E. Erçelik, T. Bänziger, and A. Knoll, "Radar-based 2D car detection using deep neural networks," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–8.
- [115] D. Köhler, M. Quach, M. Ulrich, F. Meinl, B. Bischoff, and H. Blume, "Improved multi-scale grid rendering of point clouds for radar object detection networks," 2023, [arXiv:2305.15836](https://arxiv.org/abs/2305.15836).
- [116] J. Liu, Q. Zhao, W. Xiong, T. Huang, Q.-L. Han, and B. Zhu, "SMURF: Spatial multi-representation fusion for 3D object detection with 4D imaging radar," 2023, [arXiv:2307.10784](https://arxiv.org/abs/2307.10784).
- [117] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.
- [118] S. Tsutsui, T. Kerola, and S. Saito, "Distantly supervised road segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 174–181.
- [119] S. Tsutsui, T. Kerola, S. Saito, and D. J. Crandall, "Minimizing supervision for free-space segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 988–997.
- [120] Y.-C. Chan, Y.-C. Lin, and P.-C. Chen, "Lane mark and drivable area detection using a novel instance segmentation scheme," in *Proc. IEEE/SICE Int. Symp. Integr.*, 2019, pp. 502–506.
- [121] Y. Qian, J. M. Dolan, and M. Yang, "DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4670–4679, Nov. 2020.
- [122] H. Wang, R. Fan, P. Cai, and M. Liu, "SNE-RoadSeg : Rethinking depth-normal translation and deep supervision for freespace detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1140–1145.
- [123] M.-E. Shao, M. A. Haq, D.-Q. Gao, P. Chondro, and S.-J. Ruan, "Semantic segmentation for free space and lane based on grid-based interest point detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8498–8512, Jul. 2022.
- [124] J. Kim and C. Park, "End-to-end ego lane estimation based on sequential transfer learning for self-driving cars," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 30–38.
- [125] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 286–291.
- [126] Z. Wang, W. Ren, and Q. Qiu, "LaneNet: Real-time lane detection networks for autonomous driving," 2018, [arXiv:1807.01726](https://arxiv.org/abs/1807.01726).
- [127] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Jan. 2020.
- [128] X. Chen, Y. Liu, and K. Achuthan, "WODIS: Water obstacle detection network based on image segmentation for autonomous surface vehicles in maritime environments," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 7503213.
- [129] B. Bovcon and M. Kristan, "WaSR—A water segmentation and refinement maritime obstacle detection network," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 12661–12674, Dec. 2022.
- [130] B. Bovcon, J. Muhovič, D. Vranac, D. Mozetič, J. Perš, and M. Kristan, "Mods—A USV-oriented object detection and obstacle segmentation benchmark," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13403–13418, Aug. 2022.
- [131] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
- [132] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
- [133] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
- [134] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 173–190.
- [135] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, [arXiv:1412.7062](https://arxiv.org/abs/1412.7062).
- [136] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
- [137] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, [arXiv:1706.05587](https://arxiv.org/abs/1706.05587).
- [138] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.
- [139] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, [arXiv:1606.02147](https://arxiv.org/abs/1606.02147).
- [140] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.
- [141] P. Wang et al., "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1451–1460.
- [142] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3684–3692.
- [143] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6881–6890.
- [144] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7262–7272.
- [145] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021, pp. 12077–12090.
- [146] H. Yan, C. Zhang, and M. Wu, "Lawin transformer: Improving semantic segmentation transformer with multi-scale representations via large window attention," 2022, [arXiv:2201.01615](https://arxiv.org/abs/2201.01615).
- [147] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021, pp. 17864–17875.

- [148] A. Ouaknine, A. Newson, P. Pérez, F. Tupin, and J. Rebut, "Multi-view radar semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15671–15680.
- [149] S. T. Isele, F. Klein, M. Brosowsky, and J. M. Zöllner, "Learning semantics on radar point-clouds," in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 810–817.
- [150] F. E. Nowruzi et al., "PolarNet: Accelerated deep open space segmentation using automotive radar in polar domain," 2021, *arXiv:2103.03387*.
- [151] J. Lombacher, K. Laudt, M. Hahn, J. Dickmann, and C. Wöhler, "Semantic radar grids," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1170–1175.
- [152] R. Prophet, G. Li, C. Sturm, and M. Vossiek, "Semantic segmentation on automotive radar maps," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 756–763.
- [153] P. Kaul, D. De Martini, M. Gadd, and P. Newman, "RSS-Net: Weakly-supervised multi-class semantic segmentation with FMCW radar," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 431–436.
- [154] I. Orr, M. Cohen, and Z. Zalevsky, "High-resolution radar road segmentation using weakly supervised learning," *Nature Mach. Intell.*, vol. 3, no. 3, pp. 239–246, 2021.
- [155] L. Zhang et al., "PeakConv: Learning peak receptive field for radar semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17577–17586.
- [156] Z. Feng, S. Zhang, M. Kunert, and W. Wiesbeck, "Point cloud segmentation with a high-resolution automotive radar," in *Proc. IEEE Automot. Meets Electron.; 10th GMM-Symp. VDE*, 2019, pp. 1–5.
- [157] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Supervised clustering for radar applications: On the way to radar instance segmentation," in *Proc. IEEE MTT-S Int. Conf. Microw. Intell. Mobility*, 2018, pp. 1–4.
- [158] F. Nobis, F. Fent, J. Betz, and M. Lienkamp, "Kernel point convolution LSTM networks for radar point cloud segmentation," *Appl. Sci.*, vol. 11, no. 6, 2021, Art. no. 2599.
- [159] J. Liu et al., "Deep instance segmentation with automotive radar detection points," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 84–94, Jan. 2023.
- [160] R. Prophet, A. Deligiannis, J.-C. Fuentes-Michel, I. Weber, and M. Vossiek, "Semantic segmentation on 3D occupancy grids for automotive radar," *IEEE Access*, vol. 8, pp. 197917–197930, 2020.
- [161] M. Zeller, J. Behley, M. Heidingsfeld, and C. Stachniss, "Gaussian radar transformer for semantic segmentation in noisy radar data," *IEEE Robot. Automat. Lett.*, vol. 8, no. 1, pp. 344–351, Jan. 2023.
- [162] W. Xiong, J. Liu, Y. Xia, T. Huang, B. Zhu, and W. Xiang, "Contrastive learning for automotive mmWave radar detection points based instance segmentation," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst.*, 2022, pp. 1255–1261.
- [163] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
- [164] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.
- [165] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.
- [166] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2446–2454.
- [167] M. Mostajabi, C. M. Wang, D. Ranjan, and G. Hsyu, "High-resolution radar dataset for semi-supervised learning of dynamic objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 100–101.
- [168] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "RADIATE: A radar dataset for automotive perception in bad weather," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 1–7.
- [169] X. Weng, Y. Man, J. Park, Y. Yuan, M. O'Toole, and K. M. Kitani, "All-in-one drive: A comprehensive perception dataset with high-density long-range point clouds," Jun. 8, 2021. [Online]. Available: <https://openreview.net/forum?id=yI9aThYT9W>
- [170] Y. Wang, G. Wang, H.-M. Hsu, H. Liu, and J.-N. Hwang, "Rethinking of radar's role: A camera-radar dataset and systematic annotator via coordinate alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2815–2824.
- [171] T.-Y. Lim, S. A. Markowitz, and M. N. Do, "Radical: A synchronized FMCW radar, depth, IMU and RGB camera data dataset with low-level FMCW radar signals," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 4, pp. 941–953, Jun. 2021.
- [172] Y. Cheng et al., "Flow: A dataset and benchmark for floating waste detection in inland waters," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10953–10962.
- [173] K. Burnett et al., "Boreas: A multi-season autonomous driving dataset," *Int. J. Robot. Res.*, vol. 42, no. 1/2, pp. 33–42, 2023.
- [174] S. Yao et al., "Waterscenes: A multi-task 4D radar-camera fusion dataset and benchmark for autonomous driving on water surfaces," 2023, *arXiv:2307.06505*.
- [175] M. Meyer and G. Kuschk, "Deep learning based 3D object detection for automotive radar and camera," in *Proc. IEEE 16th Eur. Radar Conf.*, 2019, pp. 133–136.
- [176] L. Zheng et al., "Tj4dradset: A 4D radar dataset for autonomous driving," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst.*, 2022, pp. 493–498.
- [177] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, "K-radar: 4D radar object detection dataset and benchmark for autonomous driving in various weather conditions," 2022, *arXiv:2206.08171*.
- [178] T. Matuszka et al., "Aimotive dataset: A multimodal dataset for robust autonomous driving with long-range perception," 2022, *arXiv:2211.09445*.
- [179] O. Schumann et al., "RadarScenes: A real-world radar point cloud data set for automotive applications," in *Proc. IEEE 24th Int. Conf. Inf. Fusion*, 2021, pp. 1–8.
- [180] J. Guan, S. Madani, S. Jog, S. Gupta, and H. Hassanieh, "Through fog high-resolution imaging using millimeter wave radar," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11464–11473.
- [181] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.
- [182] Z. Liu et al., "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6640–6653, Jul. 2022.
- [183] R. Nabati, L. Harris, and H. Qi, "CFTrack: Center-based radar and camera fusion for 3D multi-object tracking," in *Proc. IEEE Intell. Veh. Symp. Workshops*, 2021, pp. 243–248.
- [184] A. Benterki, M. Boukhnifer, V. Judalet, and C. Maaoui, "Artificial intelligence for vehicle behavior anticipation: Hybrid approach based on maneuver classification and trajectory prediction," *IEEE Access*, vol. 8, pp. 56992–57002, 2020.
- [185] P. Cai, S. Wang, Y. Sun, and M. Liu, "Probabilistic end-to-end vehicle navigation in complex dynamic environments with multimodal sensor fusion," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4218–4224, Jul. 2020.
- [186] S. Chadwick, W. Maddern, and P. Newman, "Distant vehicle detection using radar and vision," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 8311–8317.
- [187] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3D object detection," 2018, *arXiv:1811.08188*.
- [188] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kortschieder, "Disentangling monocular 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1991–1999.
- [189] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.
- [190] R. Yadav, A. Vierling, and K. Berns, "Radar RGB fusion for robust object detection in autonomous vehicle," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 1986–1990.
- [191] P. Zhao, C. X. Lu, B. Wang, N. Trigoni, and A. Markham, "CubeLearn: End-to-end learning for human motion recognition from raw mmWave radar signals," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10236–10249, Jun. 2023.
- [192] T. Stadelmayer, A. Santra, R. Weigel, and F. Lurz, "Data-driven radar processing using a parametric convolutional neural network for human activity classification," *IEEE Sensors J.*, vol. 21, no. 17, pp. 19529–19540, Sep. 2021.
- [193] Y. Kim and T. Moon, "Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 8–12, Jan. 2016.
- [194] R. Nabati and H. Qi, "Radar-camera sensor fusion for joint object detection and distance estimation in autonomous vehicles," 2020, *arXiv:2009.08428*.
- [195] H. Cui, J. Wu, J. Zhang, G. Chowdhary, and W. R. Norris, "3D detection and tracking for on-road vehicles with a monovision camera and dual low-cost 4D mmWave radars," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 2931–2937.

- [196] L.-Q. Li and Y.-L. Xie, "A feature pyramid fusion detection algorithm based on radar and camera sensor," in *Proc. IEEE 15th Int. Conf. Signal Process.*, 2020, pp. 366–370.
- [197] K. Kowol, M. Rottmann, S. Bracke, and H. Gottschalk, "Yodar: Uncertainty-based sensor fusion for vehicle detection with camera and radar sensors," 2020, *arXiv:2010.03320*.
- [198] Y. Cheng, H. Xu, and Y. Liu, "Robust small object detection on the water surface through fusion of camera and millimeter wave radar," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15263–15272.
- [199] V. John, M. Nithilan, S. Mita, H. Tehrani, R. Sudheesh, and P. Lal, "SO-Net: Joint semantic segmentation and obstacle detection using deep fusion of monocular camera and radar," in *Proc. Pacific-Rim Symp. Image Video Technol.*, 2020, pp. 138–148.
- [200] X. Dong, B. Zhuang, Y. Mao, and L. Liu, "Radar camera fusion via representation learning in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1672–1681.
- [201] Y. Song, Z. Xie, X. Wang, and Y. Zou, "MS-YOLO: Object detection based on YOLOv5 optimized fusion millimeter-wave radar and machine vision," *IEEE Sensors J.*, vol. 22, no. 15, pp. 15435–15447, Aug. 2022.
- [202] K. Bansal, K. Rungta, and D. Bharadia, "Radsegnet: A reliable approach to radar camera fusion," 2022, *arXiv:2208.03849*.
- [203] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-BEV: What really matters for multi-sensor BEV perception?", in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 2759–2765.
- [204] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo inverse perspective mapping: Theory and applications," *Image Vis. Comput.*, vol. 16, no. 8, pp. 585–590, 1998.
- [205] M. Oliveira, V. Santos, and A. D. Sappa, "Multimodal inverse perspective mapping," *Inf. Fusion*, vol. 24, pp. 108–121, 2015.
- [206] T.-Y. Lim et al., "Radar and camera early fusion for vehicle detection in advanced driver assistance systems," in *Proc. Mach. Learn. Auton. Driving Workshop, 33rd Conf. Neural Inf. Process. Syst.*, 2019, vol. 2, no. 7.
- [207] T. Zhou, J. Chen, Y. Shi, K. Jiang, M. Yang, and D. Yang, "Bridging the view disparity between radar and camera features for multi-modal fusion 3D object detection," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1523–1535, Feb. 2023.
- [208] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8445–8453.
- [209] H. Jha, V. Lodhi, and D. Chakravarty, "Object detection and identification using vision and radar data fusion system for ground-based navigation," in *Proc. IEEE 6th Int. Conf. Signal Process. Integr. Netw.*, 2019, pp. 590–593.
- [210] J. Kim, Y. Kim, and D. Kum, "Low-level sensor fusion network for 3D vehicle detection using radar range-azimuth heatmap and monocular image," in *Proc. Asian Conf. Comput. Vis.*, 2020.
- [211] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, "A deep learning-based radar and camera sensor fusion architecture for object detection," in *Proc. IEEE Sensor Data Fusion: Trends, Solutions, Appl.*, 2019, pp. 1–7.
- [212] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [213] Y. Long, D. Morris, X. Liu, M. Castro, P. Chakravarty, and P. Narayanan, "Radar-camera pixel depth association for depth completion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12507–12516.
- [214] S. Chang et al., "Spatial attention fusion for obstacle detection using mmWave radar and vision sensor," *Sensors*, vol. 20, no. 4, 2020, Art. no. 956.
- [215] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
- [216] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks," in *Proc. 21st Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2018, pp. 421–429.
- [217] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *Int. J. Comput. Vis.*, vol. 128, no. 5, pp. 1239–1285, 2020.
- [218] T. Broedermann, C. Sakaridis, D. Dai, and L. Van Gool, "HRFuser: A multi-resolution sensor fusion architecture for 2D object detection," 2022, *arXiv:2206.15157*.
- [219] Y. Yuan et al., "HRFormer: High-resolution transformer for dense prediction," 2021, *arXiv:2110.09408*.
- [220] E. Olson, "A passive solution to the sensor synchronization problem," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1059–1064.
- [221] A. Westenberger, T. Huck, M. Fritzsche, T. Schwarz, and K. Dietmayer, "Temporal synchronization in multi-sensor fusion for future driver assistance systems," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control, Commun.*, 2011, pp. 93–98.
- [222] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1280–1286.
- [223] T. Huck, A. Westenberger, M. Fritzsche, T. Schwarz, and K. Dietmayer, "Precise timestamping and temporal synchronization in multi-sensor fusion," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 242–247.
- [224] J. E. Guivant, S. Marden, and K. Pereira, "Distributed multi sensor data fusion for autonomous 3D mapping," in *Proc. IEEE Int. Conf. Indoor Positioning Indoor Navigation*, 2012, pp. 1–11.
- [225] J. Steinbäck, C. Steger, E. Brenner, and N. Druml, "A hybrid timestamping approach for multi-sensor perception systems," in *Proc. IEEE 23rd Euromicro Conf. Digit. Syst. Des.*, 2020, pp. 447–454.
- [226] A. English, P. Ross, D. Ball, B. Upcroft, and P. Corke, "Triggersync: A time synchronisation tool," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 6220–6226.
- [227] L. Lu, C. Zhang, Y. Liu, W. Zhang, and Y. Xia, "IEEE 1588-based general and precise time synchronization method for multiple sensors," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2019, pp. 2427–2432.
- [228] M. Huber, M. Schlegel, and G. Klinker, "Temporal calibration in multi-sensor tracking setups," in *Proc. IEEE 8th Int. Symp. Mixed Augmented Reality*, 2009, pp. 195–196.
- [229] Y. Du, B. Qin, C. Zhao, Y. Zhu, J. Cao, and Y. Ji, "A novel spatio-temporal synchronization method of roadside asynchronous MMW radar-camera for sensor fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22278–22289, Nov. 2022.
- [230] Y. Fu et al., "A camera–radar fusion method based on edge computing," in *Proc. IEEE Int. Conf. Edge Comput.*, 2020, pp. 9–14.
- [231] F. Liu, J. Sparber, and C. Stiller, "IMMPDA vehicle tracking system using asynchronous sensor fusion of radar and vision," in *Proc. IEEE Intell. Veh. Symp.*, 2008, pp. 168–173.
- [232] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 226–231.
- [233] L. Wang, Z. Zhang, X. Di, and J. Tian, "A roadside camera-radar sensing fusion system for intelligent transportation," in *Proc. IEEE 17th Eur. Radar Conf.*, 2021, pp. 282–285.
- [234] J. Peršić, L. Petrović, I. Marković, and I. Petrović, "Spatiotemporal multisensor calibration via Gaussian processes moving target tracking," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1401–1415, Oct. 2021.
- [235] S. Li, C. Xu, and M. Xie, "A robust O(n) solution to the perspective-n-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1444–1450, Jul. 2012.
- [236] E. Wise, J. Peršić, C. Grebe, I. Petrović, and J. Kelly, "A continuous-time approach for 3D radar-to-camera extrinsic calibration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13164–13170.
- [237] J. Peršić, L. Petrović, I. Marković, and I. Petrović, "Online multi-sensor calibration based on moving object tracking," *Adv. Robot.*, vol. 35, no. 3/4, pp. 130–140, 2021.
- [238] K. Qiu, T. Qin, J. Pan, S. Liu, and S. Shen, "Real-time temporal and rotational calibration of heterogeneous sensors using motion correlation analysis," *IEEE Trans. Robot.*, vol. 37, no. 2, pp. 587–602, Apr. 2021.
- [239] X.-P. Guo, J.-S. Du, J. Gao, and W. Wang, "Pedestrian detection based on fusion of millimeter wave radar and vision," in *Proc. Int. Conf. Artif. Intell. Pattern Recognit.*, 2018, pp. 38–42.
- [240] G. Iyer, R. K. Ram, J. K. Murthy, and K. M. Krishna, "CalibNet: Geometrically supervised extrinsic calibration using 3D spatial transformer networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1110–1117.
- [241] C. Schöller et al., "Targetless rotational auto-calibration of radar and camera for intelligent transportation systems," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 3934–3941.
- [242] G. Zhao, J. Hu, S. You, and C.-C. J. Kuo, "CalibDNN: Multimodal sensor calibration for perception using deep neural networks," *Proc. SPIE*, vol. 11756, pp. 324–335, 2021.

- [243] F. Nobis, E. Shafiei, P. Karle, J. Betz, and M. Lienkamp, "Radar voxel fusion for 3D object detection," *Appl. Sci.*, vol. 11, no. 12, 2021, Art. no. 5598.
- [244] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, 1991.
- [245] Y. Kim, J. W. Choi, and D. Kum, "Grif Net: Gated region of interest fusion network for robust 3D object detection from radar point cloud and monocular image," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 10857–10864.
- [246] M. Ren, A. Pokrovsky, B. Yang, and R. Urtasun, "SBNet: Sparse blocks network for fast inference," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8711–8720.
- [247] W. Xiong, J. Liu, T. Huang, Q.-L. Han, Y. Xia, and B. Zhu, "LXL: Lidar exclusive lean 3D object detection with 4D imaging radar and camera fusion," 2023, *arXiv:2307.00724*.
- [248] Y. Kim, S. Kim, J. W. Choi, and D. Kum, "Craft: Camera-radar 3D object detection with spatio-contextual fusion transformer," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 1160–1168.
- [249] Z. Wu, G. Chen, Y. Gan, L. Wang, and J. Pu, "MVFusion: Multi-view 3D object detection with semantic-aligned radar and camera fusion," 2023, *arXiv:2302.10511*.
- [250] Y. Kim, S. Kim, J. Shin, J. W. Choi, and D. Kum, "CRN: Camera radar net for accurate, robust, efficient 3D perception," 2023, *arXiv:2304.00670*.
- [251] L. Zheng et al., "RCFusion: Fusing 4D radar and camera with bird's-eye view features for 3D object detection," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 8503814.
- [252] Y. Jin, A. Deligiannis, J.-C. Fuentes-Michel, and M. Vossiek, "Cross-modal supervision-based multitask learning with automotive radar raw data," *IEEE Trans. Intell. Veh.*, vol. 8, no. 4, pp. 3012–3025, Apr. 2023.
- [253] J.-J. Hwang et al., "CramNet: Camera-radar fusion with ray-constrained cross-attention for robust 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 388–405.
- [254] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6569–6578.
- [255] L. Stäcker, P. Heidenreich, J. Rambach, and D. Stricker, "Fusion point pruning for optimized 2D object detection with radar-camera fusion," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 3087–3094.
- [256] C. X. Lu et al., "See through smoke: Robust indoor mapping with low-cost mmWave radar," in *Proc. 18th Int. Conf. Mobile Syst., Appl., Serv.*, 2020, pp. 14–27.
- [257] J. Wang, "CFAR-based interference mitigation for FMCW automotive radar systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12229–12238, Aug. 2022.
- [258] R. Zhang and S. Cao, "Support vector machines for classification of automotive radar interference," in *Proc. IEEE Radar Conf.*, 2018, pp. 0366–0371.
- [259] F. Jin and S. Cao, "Automotive radar interference mitigation using adaptive noise canceller," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3747–3754, Apr. 2019.
- [260] M. Alhumaidi and M. Wintermantel, "Interference avoidance and mitigation in automotive radar," in *Proc. IEEE 17th Eur. Radar Conf.*, 2021, pp. 172–175.
- [261] C. Schüßler, M. Hoffmann, I. Ullmann, R. Ebelt, and M. Vossiek, "Deep learning based image enhancement for automotive radar trained with an advanced virtual sensor," *IEEE Access*, vol. 10, pp. 40419–40431, 2022.
- [262] J. Rock, W. Roth, M. Toth, P. Meissner, and F. Pernkopf, "Resource-efficient deep neural networks for automotive radar interference mitigation," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 4, pp. 927–940, Jun. 2021.
- [263] A. Dubey, J. Fuchs, V. Madhavan, M. Lübke, R. Weigel, and F. Lurz, "Region based single-stage interference mitigation and target detection," in *Proc. IEEE Radar Conf.*, 2020, pp. 1–5.
- [264] N.-C. Ristea, A. Anghel, and R. T. Ionescu, "Fully convolutional neural networks for automotive radar interference mitigation," in *Proc. IEEE 92nd Veh. Technol. Conf.*, 2020, pp. 1–5.
- [265] J. Fuchs, A. Dubey, M. Lübke, R. Weigel, and F. Lurz, "Automotive radar interference mitigation using a convolutional autoencoder," in *Proc. IEEE Int. Radar Conf.*, 2020, pp. 315–320.
- [266] S. Chen, J. Taghia, T. Fei, U. Kühnau, N. Pohl, and R. Martin, "A DNN autoencoder for automotive radar interference mitigation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 4065–4069.
- [267] S. Chen, J. Taghia, U. Kühnau, N. Pohl, and R. Martin, "A two-stage DNN model with mask-gated convolution for automotive radar interference detection and mitigation," *IEEE Sensors J.*, vol. 22, no. 12, pp. 12017–12027, Jun. 2022.
- [268] M. L. L. de Oliveira and M. J. Bekooij, "Deep convolutional autoencoder applied for noise reduction in range-doppler maps of FMCW radars," in *Proc. IEEE Int. Radar Conf.*, 2020, pp. 630–635.
- [269] Y. Cheng, J. Su, H. Chen, and Y. Liu, "A new automotive radar 4D point clouds detector by using deep learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 8398–8402.
- [270] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," 2022, *arXiv:2202.07123*.
- [271] G. Qian et al., "PointNext: Revisiting pointnet with improved training and scaling strategies," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2022, pp. 23192–23204.
- [272] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 16259–16268.
- [273] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, pp. 187–199, 2021.
- [274] R. Zhang, L. Wang, Y. Wang, P. Gao, H. Li, and J. Shi, "Parameter is not all you need: Starting from non-parametric networks for 3D point cloud analysis," 2023, *arXiv:2303.08134*.
- [275] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
- [276] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.
- [277] F. E. Nowruzi, P. Kapoor, D. Kolhatkar, F. A. Hassamat, R. Laganiere, and J. Rebut, "How much real data do we actually need: Analyzing object detection performance using synthetic and real data," 2019, *arXiv:1907.07061*.
- [278] M. Stephan, T. Stadelmayer, A. Santra, G. Fischer, R. Weigel, and F. Lurz, "Radar image reconstruction from raw ADC data using parametric variational autoencoder with domain adaptation," in *Proc. IEEE 25th Int. Conf. Pattern Recognit.*, 2021, pp. 9529–9536.
- [279] T. Li, L. Fan, Y. Yuan, and D. Katabi, "Unsupervised learning for human sensing using radio signals," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 3288–3297.
- [280] X. Li, Y. He, F. Fioranelli, and X. Jing, "Semisupervised human activity recognition with radar micro-doppler signatures," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5103112.
- [281] Y. Li et al., "Towards domain-independent and real-time gesture recognition using mmwave signal," *IEEE Trans. Mobile Comput.*, early access, Sep. 19, 2022, doi: [10.1109/TMC.2022.3207570](https://doi.org/10.1109/TMC.2022.3207570).
- [282] J. Domhof, J. F. P. Kooij, and D. M. Gavrila, "A joint extrinsic calibration tool for radar, camera and lidar," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 571–582, Sep. 2021.
- [283] J. Zhang, S. Zhang, G. Peng, H. Zhang, and D. Wang, "3DRadar2ThermalCalib: Accurate extrinsic calibration between a 3D mmWave radar and a thermal camera using a spherical-triangular," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst.*, 2022, pp. 2744–2749.
- [284] A. Sengupta, A. Yoshizawa, and S. Cao, "Automatic radar-camera dataset generation for sensor-fusion applications," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2875–2882, Apr. 2022.
- [285] P. Ren et al., "A survey of deep active learning," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–40, 2021.
- [286] X. Zhan, Q. Wang, K.-H. Huang, H. Xiong, D. Dou, and A. B. Chan, "A comparative survey of deep active learning," 2022, *arXiv:2203.13450*.
- [287] T. Winterling, J. Lombacher, M. Hahn, J. Dickmann, and C. Wöhler, "Optimizing labelling on radar-based grid maps using active learning," in *Proc. IEEE 18th Int. Radar Symp.*, 2017, pp. 1–6.
- [288] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 53–69, May 2015.
- [289] Y. Chen, W. Li, X. Chen, and L. V. Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1841–1850.
- [290] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [291] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. 27th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2014, Art. no. 3581–3589.

- [292] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, 2019.
- [293] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12697–12705.
- [294] N. Scheiner, F. Kraus, N. Appenrodt, J. Dickmann, and B. Sick, "Object detection for automotive radar point clouds—A comparison," *AI Perspectives*, vol. 3, no. 1, pp. 1–23, 2021.
- [295] J. Bai, L. Zheng, S. Li, B. Tan, S. Chen, and L. Huang, "Radar transformer: An object classification network based on 4D MMW imaging radar," *Sensors*, vol. 21, no. 11, 2021, Art. no. 3854.
- [296] B. Xu et al., "RPFA-Net: A 4D radar pillar feature attention network for 3D object detection," in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 3061–3066.
- [297] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
- [298] M. Meyer, G. Kuschk, and S. Tomforde, "Graph convolutional networks for 3D object detection on radar data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3060–3069.
- [299] J. Bai, S. Li, L. Huang, and H. Chen, "Robust detection and tracking method for moving object based on radar and camera data fusion," *IEEE Sensors J.*, vol. 21, no. 9, pp. 10761–10774, May 2021.
- [300] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6023–6032.
- [301] W. Wang, D. Tran, and M. Feiszli, "What makes training multi-modal classification networks hard?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12695–12705.
- [302] Q. Dou, Q. Liu, P. A. Heng, and B. Glocker, "Unpaired multi-modal segmentation via knowledge distillation," *IEEE Trans. Med. Imag.*, vol. 39, no. 7, pp. 2415–2425, Jul. 2020.
- [303] C. Du et al., "Improving multi-modal learning with uni-modal teachers," 2021, *arXiv:2106.11059*.
- [304] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2365–2374.
- [305] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Comput.*, vol. 4, no. 3, pp. 448–472, 1992.
- [306] T. A. Wheeler, M. Holder, H. Winner, and M. J. Kochenderfer, "Deep stochastic radar models," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 47–53.
- [307] V. Lekic and Z. Babic, "Automotive radar and camera fusion using generative adversarial networks," *Comput. Vis. Image Understanding*, vol. 184, pp. 1–8, 2019.
- [308] D. Hall et al., "Probabilistic object detection: Definition and evaluation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1031–1040.
- [309] A. Dhamija, M. Gunther, J. Ventura, and T. Boult, "The overlooked elephant of object detection: Open set," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 1021–1030.
- [310] K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5830–5840.
- [311] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digit. Signal Process.*, vol. 73, pp. 1–15, 2018.
- [312] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 8, pp. 2674–2693, Aug. 2019.
- [313] H. Yu, F. Zhang, P. Huang, C. Wang, and L. Yuanhao, "Autonomous obstacle avoidance for UAV based on fusion of radar and monocular camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5954–5961.
- [314] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," 2015, *arXiv:1510.00149*.
- [315] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*.
- [316] Z. Liu et al., "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 2774–2781.
- [317] M. Meyer and G. Kuschk, "Automotive radar dataset for deep learning based 3D object detection," in *Proc. IEEE 16th Eur. Radar Conf.*, 2019, pp. 129–132.
- [318] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.
- [319] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2403–2412.
- [320] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "Futr3D: A unified sensor fusion framework for 3D detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 172–181.
- [321] F. Drews, D. Feng, F. Faion, L. Rosenbaum, M. Ulrich, and C. Gläser, "DeepFusion: A robust and modular 3D object detector for lidars, cameras and radars," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 560–567.
- [322] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11976–11986.
- [323] R. Guan et al., "Achelous: A fast unified water-surface panoptic perception framework based on fusion of monocular camera and 4D mmWave radar," 2023, *arXiv:2307.07102*.



**Shanliang Yao** (Student Member, IEEE) received the B.E. degree from the School of Computer Science and Technology, Soochow University, Suzhou, China, in 2016, and the M.S. degree from the Faculty of Science and Engineering, University of Liverpool, Liverpool, U.K., in 2021. He is currently a joint Ph.D. student of the University of Liverpool, Xi'an Jiaotong-Liverpool University, Institute of Deep Perception Technology, and Jiangsu Industrial Technology Research Institute. His research interests include multi-modal perception using deep learning approach for autonomous driving, robotics, autonomous vehicles, and intelligent transportation systems.



**Runwei Guan** (Student Member, IEEE) received the M.S. degree in data science from the University of Southampton, Southampton, U.K., in 2021. He is currently a joint Ph.D. student of the University of Liverpool, Liverpool, U.K., Xi'an Jiaotong-Liverpool University, Institute of Deep Perception Technology, Jiangsu Industrial Technology Research Institute. His research interests include visual grounding, panoptic perception based on the fusion of radar and camera, lightweight neural network, multi-task learning and statistical machine learning. He is the peer Reviewer of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, Engineering Applications of Artificial Intelligence, Journal of Supercomputing, and International Joint Conference on Neural Networks.



**Xiaoyu Huang** (Student Member, IEEE) received the B.S. degree in electronic science and technology from the Nanjing University of Information Science and Technology, Nanjing, China, in 2019, and the M.S. degree in 2022 from the University of Liverpool and Xi'an Jiaotong-Liverpool University joint program, where he is currently working toward the Ph.D. degree with the School of Advanced Technology. His research interests include computer vision, pattern recognition, deep learning, and image processing.



**Zhuoxiao Li** (Student Member, IEEE) received the M.Sc. degree in information systems from the Information School, University of Sheffield, Sheffield, U.K., in 2020. He is currently working toward the Ph.D. degree with the Department of Computer Science, University of Liverpool, Liverpool, U.K. He joined the VR+ Culture Lab, School of Information Management, Sun Yat-sen University, Guangzhou, China, as a Research Assistant. His main research interests include the combination of unmanned surface vehicle and virtual reality/augmented reality, deep learning applications in geographic information systems, and remote sensing.



**Hyunjoon Seo** (Member, IEEE) received the bachelor's degree in civil engineering and the Ph.D. degree in geotechnical engineering from Korea University, Seoul, South Korea, in 2007, and 2013, respectively. In 2013, he was a Research Professor with Korea University. He was a Visiting Scholar with the University of Cambridge, Cambridge, U.K. From 2014 to 2016, he was a Research Associate with Engineering Department, University of Cambridge. In August 2016, he got an Assistant Professor position with the Department of Civil Engineering, Xi'an Jiaotong Liverpool University, Suzhou, China. Since 2020, he has been an Assistant Professor with the University of Liverpool, Liverpool, U.K. He has authored or coauthored more than 50 scientific papers including a book on Geotechnical Engineering and SMART monitoring. His research interests include monitoring using artificial intelligence and SMART monitoring system for infrastructure, soil-structure interaction (tunneling, slope stability, pile), Antarctic survey and freezing ground. He is the Director of the Centre for SMART Monitoring Infrastructure (CSMI). CSMI is collaborating with University of Cambridge, University of Oxford, University of Bath, UC Berkeley University, Nanjing University, and Tongji University on SMART monitoring. He presented keynote speech at the 15th European Conference on Soil Mechanics and Geotechnical Engineering in 2015. He is currently an appointed Editor of the *CivilEng* journal and organized two international conferences.



**Xiangyu Sha** (Student Member, IEEE) is currently working toward the B.S. degree in computer science from the University of Liverpool, Liverpool, U.K. She will complete her undergraduate study in 2024. Her research interests include computer vision, virtual reality, extended reality, robotics simulation and sensor fusion in autonomous driving vehicles. She has won full scholarships for two consecutive years and participated the Programme and Poster Competition as a Summer Undergraduate Research Fellow in 2022.



**Yong Yue** received the B.Eng. degree in mechanical engineering from Northeastern University, Shenyang, China, in 1982, and the Ph.D. degree in computer aided design from Heriot-Watt University, Edinburgh, U.K., in 1994. He was with the industry for eight years and followed experience in academia with the University of Nottingham, Nottingham, U.K., Cardiff University, Cardiff, U.K., and the University of Bedfordshire, Luton, U.K. He is currently a Professor and the Director with the Virtual Engineering Centre, Xi'an Jiaotong-Liverpool University, Suzhou, China.

He is a Fellow of Institution of Engineering and Technology. His research interests include computer graphics, virtual reality, and robot navigation.



**Ka Lok Man** (Member, IEEE), received the Dr.Eng. degree in electronic engineering from the Politecnico di Torino, Turin, Italy, in 1998, and the Ph.D. degree in computer science from Technische Universiteit Eindhoven, Eindhoven, The Netherlands, in 2006. He is currently a Professor of computer science and software engineering with Xi'an Jiaotong-Liverpool University, Suzhou, China. His research interests include formal methods and process algebras, embedded system design and testing, and photovoltaics.



**Xiaohui Zhu** (Member, IEEE) received the Ph.D. degree from the University of Liverpool, Liverpool, U.K., in 2019. He is currently an Assistant Professor, Ph.D. Supervisor, and the Programme Director with the Department of Computing, School of Advanced Technology, Xi'an Jiaotong-Liverpool University Suzhou, China. He focuses on advanced techniques related to autonomous driving, including sensor-fusion perception, fast path planning, autonomous navigation and multi-vehicle collaborative scheduling.



**Eng Gee Lim** (Senior Member, IEEE) received the B.Eng. (Hons.) and Ph.D. degrees in electrical and electronic engineering (EEE) from Northumbria University, Newcastle, U.K., in 1998 and 2002, respectively. From 2002 to 2007, he was with Andrew Ltd., Coventry, U.K., a leading communications systems company. Since 2007, he has been with Xi'an Jiaotong-Liverpool University, Suzhou, China, where he was the Head of the EEE Department, and the University Dean of research and graduate studies. He is currently the School Dean of Advanced Technology, Director of the AI University Research Centre, and a Professor with the Department of EEE. He has authored or coauthored more than 100 refereed international journals and conference papers. His research interests include artificial intelligence, robotics, AI+ health care, international standard (ISO/IEC) in robotics, antennas, RF/microwave engineering, EM measurements/simulations, energy harvesting, power/energy transfer, smart-grid communication, and wireless communication networks for smart and green cities. He is a Chartered Engineer and a Fellow of The Institution of Engineering and Technology and Engineers Australia. He is also a Senior Fellow of Higher Education Academy.



**Yutao Yue** (Member, IEEE) was born in Qingzhou, Shandong, China, in 1982. He received the B.S. degree in applied physics from the University of Science and Technology of China, Hefei, China, in 2004, and the M.S. and Ph.D. degrees in computational physics from Purdue University, West Lafayette, IN, USA, in 2006 and 2010, respectively. From 2011 to 2017, he was a Senior Scientist with the Shenzhen Kuang-Chi Institute of Advanced Technology and a Team Leader of the Guangdong "Zhujiang Plan" 3rd Introduced Innovation Scientific Research Team. From 2017 to 2018, he was a Research Associate Professor with the Southern University of Science and Technology, Shenzhen, China. Since 2018, he has been the Founder and the Director of the Institute of Deep Perception Technology, JITRI, Jiangsu, China. Since 2020, he has been an Honorary Recognized Ph.D. Advisor of the University of Liverpool, U.K., and Xi'an Jiaotong-Liverpool University, Suzhou, China. He is the co-inventor of more than 300 granted patents of China, USA, and Europe. He is also the author of more than 20 journals and conference papers. His research interests include computational modeling, radar vision fusion, perception and cognition cooperation, artificial intelligence theory, and electromagnetic field modulation. Dr. Yue was the recipient of the Wu WenJun Artificial Intelligence Science and Technology Award in 2020.