# mmBox: Harnessing Millimeter-Wave Signals for Reliable Vehicle and Pedestrians Detection

ZHUANGZHUANG GU, Computer Science and Engineering, University of South Carolina, Columbia, United States
HEM REGMI, Computer Science and Engineering, University of South Carolina, Columbia, United States
SANJIB SUR, Computer Science and Engineering, University of South Carolina, Columbia, United States

Object detection plays a pivotal role in various fields, for example, a smart traffic system relies on the detected results for decision-making. However, existing studies predominately utilize optical camera and LiDAR, which exhibit limitations in adverse outdoor environments, such as foggy weather. To address these challenges, millimeter-waves (mmWaves) attract researchers' attention to detect objects in severe conditions since they can work effectively in low-visibility conditions and overcome small obstacles. Yet, previous mmWave-based works have shown limited performance, such as no shape information for objects. Therefore, we design and implement a two-stage system, *mmBox*, to accurately predict bounding boxes with depth for vehicles and pedestrians, which first generates heatmaps in different dimensions and then leverages a deep learning model to extract features for predictions. To evaluate the performance of *mmBox*, we collected real-world mmWave reflections from urban traffic intersections and dense-fog environments. The extensive evaluation metrics show remarkable accuracy and the low latency of our model.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → **Computer vision tasks**; • **Hardware** → **Sensor applications and deployments;**

Additional Key Words and Phrases: Millimeter-wave, deep learning, object detection, wireless sensing

**ACM Reference Format:**
Zhuangzhuang Gu, Hem Regmi, and Sanjib Sur. 2024. mmBox: Harnessing Millimeter-Wave Signals for Reliable Vehicle and Pedestrians Detection. *ACM Trans. Internet Things* 5, 4, Article 22 (October 2024), 30 pages. https://doi.org/10.1145/3695883

## 1 Introduction

Improving pedestrian safety at traffic intersections is a paramount concern, especially when considering the grim reality that pedestrian accidents at these junctions constitute the leading cause of fatal injuries on the road, thereby significantly compromising overall road safety [1–3]. While many existing works aimed to mitigate these risks, including smart traffic symbols [4], traffic

This work is partially supported by the National Science Foundation under grant nos. CAREER-2144505, NeTS2342833, and MRI-2018966.
Authors' Contact Information: Zhuangzhuang Gu, Computer Science and Engineering, University of South Carolina, Columbia, South Carolina, United States; e-mail: ZG5@email.sc.edu; Hem Regmi, Computer Science and Engineering, University of South Carolina, Columbia, South Carolina, United States; e-mail: hregmi@email.sc.edu; Sanjib Sur, Computer Science and Engineering, University of South Carolina, Columbia, South Carolina, United States; e-mail: sur@cse.sc.edu.

ACM 2577-6207/2024/10-ART22
https://doi.org/10.1145/3695883

education programs [5], safety software in the vehicle [6], and vehicle hardware enhancements [7–9], pedestrian collision still persists during inclement weather conditions such as heavy rain, fog, snow, and reduced visibility [10–12]. These adverse weather scenarios demand a more proactive and dynamic approach to detecting pedestrians and vehicles near traffic intersections. We can actively monitor pedestrians with traffic cameras [13], but cameras fail to work during low light and harsh weather conditions [14], the time when system is required to work effectively. LiDAR-based object detection works well with low light, but fails to work during foggy conditions [15]. Furthermore, LiDARs are expensive and bulky [16], making them infeasible for mass deployment across traffic intersections.

Recently, **millimeter-wave (mmWave)**, operating at a very high frequency between 30 GHz and 300 GHz [17], has garnered significant attention from researchers owing to its distinctive advantages. Compared with other wireless signals, for example, WiFi [18, 19] and ultrasound [20, 21], mmWave exhibits the ability to penetrate various small particles, such as rain droplets, snow, and fog, offering high-range resolution at the centimeter level [22], which enables it to be applied in extreme conditions. Moreover, due to the small wavelength of mmWave, antennas to receive it are tiny, which allows the packing of many antennas into palm-sized mmWave radar devices. Having a large number of antennas facilitates **Multi-Input Multi-Output (MIMO)** by providing a large number of virtual antennas to increase the angle resolution. MIMO creates the virtual antenna arrays of $M \times N$ receiver antennas, with a single transmitter with $M$ transmitters and $N$ receivers. The utilization of a significant number of antennas in mmWave radar systems results in better resolution for both angles compared with traditional radar-ranging applications [23].

Therefore, numerous studies [24–26] have leveraged mmWave devices to detect vehicles and pedestrians on the road. RODNet [25] and DANet [27] process mmWave signals to range-azimuth heatmaps and then employ a deep learning network to detect vehicles, pedestrians, and cyclists. However, they only predict the object's probability of cluster without detailed shape information. Another notable approach is Radatron [24], which utilizes the cascaded mmWave device with a large number of antennas to predict the 2D bounding box of vehicles in **Bird's-Eye-View (BEV)**. While successful in providing the width and length of vehicles, Radatron does not provide detection for pedestrians, which is a crucial aspect in road safety. Due to the specularity of mmWave and small reflective surfaces, the reflected signals from pedestrians with weak reflectivity are challenging to capture. Furthermore, predictions in the front view offer advantages for users in understanding the scene, as the front view closely resembles human observation and camera images. Vision-based works in object detection achieve high performance in clear weather with various state-of-the-art models [28–30], contributing to the widespread use of cameras in most smart devices. Results of mmWave systems obtained in the front view can be seamlessly integrated with other vision-based works [31] to enhance decision-making. Conversely, papers focusing on object detection in BEV, which are not feasible to convert to the front view, miss the opportunity to collaborate with prevalent vision-based works.

In this article, we propose a novel two-stage system, *mmBox*, designed to generate heatmaps from mmWave signals and subsequently extract valuable features from these heatmaps, enabling the precise predictions of bounding boxes with depth values for both vehicles and pedestrians in the front view (Figures 1(a) and 1(b)). Instead of merging multiple frames of mmWave signals to increase the density of detected points [32], *mmBox* utilizes a single frame of mmWave signals, ensuring a low latency system. The results detected in the front view allow users to easily understand the positions of vehicles and pedestrians even when the camera, LiDAR and other sensors cannot work under the extreme weather conditions. This radar-based detection can be seamlessly integrated with existing vision-based models to enhance decision-making, particularly in light rain or foggy weather.

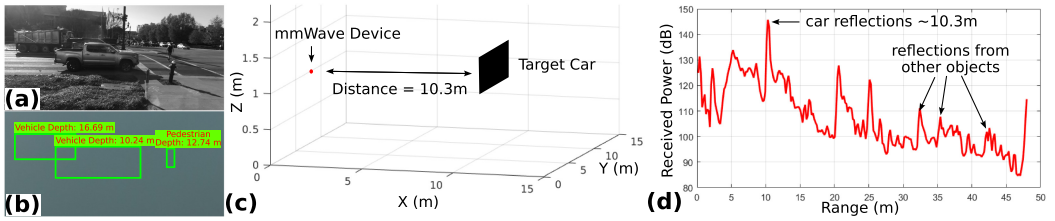


Fig. 1. (a) Example of outdoor street scene with bounding boxes for pedestrians and vehicles. The various vehicles, pedestrians, fire hydrant, grass, and street lamp make the mmWave sensing difficult to detect targets. (b) This figure illustrates the corresponding bounding boxes with depth value for vehicles and pedestrians under dense fog conditions. (c) An example of mmWave measurement in outdoor street scene. The target car is placed at 10.3 m from the mmWave transceiver. (d) The received power shows strong reflections of the car at 10.3 m, and there are other reflection peaks from surrounding objects.

To achieve these desirable features, detecting both vehicles and pedestrians in the front view with high precision and low latency, *mmBox* needs to address two primary challenges. The first challenge is associated with the hardware limitation of mmWave devices. Given the high resolution of the cascaded mmWave device (even 5 cm in range and 1.2° in azimuth [24]), recognizing objects with length and width in BEV can be solved with high performance. However, the resolution in elevation is very low, as the mmWave device has limited virtual channels in this dimension. The height information in a range-azimuth heatmap is restricted, and the coordinate differences between the range-azimuth of inputs and elevation-azimuth of outputs will make the predicted bounding box shift. However, generating 2D bounding boxes in the image plane requires accurate height measurement of objects, which challenges the performance of our system. To overcome this challenge, we design an algorithm using a Gaussian filter to project all detected points with energy distribution into the elevation-azimuth plane, which provides more height information and accurate locations of objects. In a deep learning network, we leverage the same feature extractor to capture valuable information and then fuse it with range-azimuth heatmaps at different feature levels. We also applied an anchor-based method to extract predefined knowledge by K-means on ground-truth shape distributions of various targets. *Hence, our anchor-based model extracting features from the elevation-azimuth heatmap can accurately predict the height and location of detected objects in the front view.*

Secondly, our work involves an outdoor environment featuring diverse objects (Figure 1(a)), such as vegetation, display boards, traffic lights, and fire hydrants. The received mmWave signals are from objects of interest as well as other surrounding objects that introduce noise in the reflections. Additionally, outdoor scenarios do not have controlled environments, and thermal noises also deteriorate the mmWave signals [33]. Compared with vehicles, the reflected mmWave signals from pedestrians featuring with small reflective surfaces are difficult to receive in outdoor surroundings. The single-frame input instead of combining multiple frames increases the possibility of missing objects. To address this problem, we distinguish between moving and stationary detected objects in dynamic and static range-azimuth heatmaps. This separation eliminates noise from the surroundings and captures more features, enhancing the detection accuracy compared with using a single heatmap. Since most of the moving objects across all of the traffic intersections are pedestrians and vehicles, having separate dynamic range-azimuth heatmaps makes *mmBox* robust and generalizable. In addition, we design a multiple-feature fusion strategy to effectively extract different features from both heatmaps and fuse them into three levels. This design enables the deep fusion of object representations from various inputs, significantly improving the performance of *mmBox*. We also implement our cascade mmWave device and collect data with 32 chirps

per frame, providing more information from reflected signals of surroundings. *Therefore, the separate dynamic and static heatmaps, feature-fusion method, and multiple chirps of a single frame make our system robust and reliable to detect vehicles and pedestrians in challenging outdoor conditions.*

Generally, *mmBox* consists of two stages. In the first stage, its *Heatmap Generation* module processes raw mmWave signals to produce heatmaps in multiple dimensions. Given that the mmWave device has high resolution in range and azimuth dimensions, we first generate separate static and dynamic range-azimuth heatmaps for both stationary and moving objects. To compensate for the height information and provide accurate object locations, we produce an elevation-azimuth heatmap. In the second stage, we employ a *Multi-Scale Bounding Box Generator* model to separately extract features from those heatmaps and correctly fuse them at different levels to generate predictions. To facilitate convergence during the training process, our model applies $3 \times 3$ predefined anchors produced by K-means on ground-truth data distribution rather than predicting actual value. The anchor-based method maps objects in different sizes to three-level feature maps,; for every level, we leverage three anchors with different height to width ratios to match with various objects, which significantly improves the performance of *mmBox* compared with the anchor-free approach. During inference, we filter out low confident bounding boxes and then apply a non-maximum-suppression algorithm to eliminate overlapped high-confidence 2D bounding boxes that correspond to the same object, ensuring that the final predictions are accurate and not redundant.

We prototype an off-the-shelf mmWave device and camera to collect experimental data in a real traffic intersection scene. In total, we collected 29,110 samples with reflected mmWave signals, depth images, and gray-scale images. Even though manually labeling 2D bounding boxes as ground truth is accurate, it is time-consuming. Given that YOLO [28] is the **state-of-the-art (SOTA)** vision-based approach for 2D bounding box detection, we develop a self-labeling program called *BoxLabeling* based on pre-trained YOLO to automatically detect vehicles and pedestrians using gray-scale images and find the related distance from depth image. We use 23,288 frames of data to train the model and 5,822 for validation, respectively. The results of *mmBox* combining range-azimuth heatmaps and elevation-azimuth heatmaps show that our system achieves 0.659 in $mAP_{50}$ and 0.320 m in mean $50^{th}$% *Depth Error*. To evaluate the performance under adverse outdoor environments, we prototype a plexiglass cuboid box to simulate dense fog. In total, we collect 2,392 samples to fine-tune and validate the pre-trained model. The $mAP_{50}$ of fog data is still high at 0.688, which indicates that *mmBox* can be applied in extreme outdoor environments. Moreover, our proposed deep learning model exhibits low latency, taking only 27.2 ms to generate predictions for a single frame of data, and the **frames per second (FPS)** is 36.8.

In summary, we have the following contributions. (1) We propose an algorithm to separately generate static and dynamic range-azimuth heatmaps from mmWave reflections capturing stationary and dynamic objects, which denoise the signals from complex outdoor environments and enrich the features for accurate detection of vehicles and pedestrians. (2) We further generate the elevation-azimuth heatmap to compensate the height information and provide accurate locations of objects. (3) We present a multi-scale object detection model that correctly fuses features from multiple inputs to detect pedestrians and vehicles across traffic intersections. (4) We collected the data samples across the traffic intersections with foggy conditions and evaluated our network under harsh weather conditions. To catalyze the research, we will open-source our dataset and codebase.

## 2 Background and Fundamental Challenges

### 2.1 Millimeter Wave Sensing

In mmWave imaging, a device periodically transmits **Frequency Modulated Continuous Wave (FMCW)** signals, which linearly increase with time in a certain bandwidth, and receives a

mixture of reflections from objects in the surroundings and the transmitted signals. By measuring the frequency of the received signals, the distance of the object can be calculated. Specifically, for one object, the transmitted FMCW will be reflected by the surface of the object, and the received FMCW contains a constant frequency tone representing the object's distance. In the multiple objects case, the mixture of different frequency tones can be distinguished by applying **Fast Fourier Transforms (FFTs)**. The antenna array in vertical and horizontal directions, such as $2 \times 4$, can identify the relative azimuth and elevation angles since receiving antennas get reflection signals with different phases generated by the same object. Thus, the distance, azimuth angles, and elevation angles from different objects can be mapped into a heatmap with exact locations. The specific short wavelength and wide bandwidth enable mmWaves to achieve higher resolution than other low-frequency signals, such as WiFi. It can also work well under low-light or no-light environments, and its ability to penetrate small obstacles enables mmWaves to be resilient for adverse weather, such as heavy fog [34], whereas an optical camera sensor captures extremely limited information in such environments. Furthermore, mmWave can be applied in non-destructive and privacy-sensitive situations, for example, monitoring human activities [35]. Therefore, the mmWave sensor, featuring high resolution in light-free conditions, resilience in extreme conditions, and privacy protection, has significant potential for outdoor object detection.

### 2.2 Challenges in Outdoor mmWave-Based Object Detection

Our objective is to develop a 2D object detection system in the front view using mmWave signals, aiming for precise prediction of bounding boxes with depth values for vehicles and pedestrians. However, there exist two primary challenges. (1) **Difficulty in Obtaining Height Information.** The mmWave device achieves high resolution in the range and azimuth dimensions, allowing for accurate generation of bounding box detections with width and length in the BEV, as demonstrated by past mmWave-based works such as Radatron. However, the resolution in the elevation dimension is notably low due to the limited virtual channels in the vertical direction. While some researchers attempt to generate point clouds from reflected mmWave signals to obtain 3D information, including elevation, the visualized distribution of generated **point cloud data (PCD)** reveals limited resolution in elevation angle [36]. Additionally, noise-filtered points lead to significant sparsity, posing a challenge for deep learning models to extract sufficient features. Moreover, the commonly used range-azimuth heatmap provides restricted height information, as differences between reflective points of the same target can be ignored when the distance dramatically increases in outdoor scenarios. For example, when the distance between the mmWave device and target pedestrian is high, the projected points of the head and feet may not be discerned in the range-azimuth plane. (2) **Complex Outdoor Environment.** Our dataset is collected from real outdoor traffic intersection scenes. Compared with simple indoor applications, outdoor environments encompass more complicated objects and surroundings, significantly elevating the difficulty of extracting valuable features from sparse and noisy mmWave reflections. For example, Figure 1(a) demonstrates one sample from our dataset in the busy traffic intersection, featuring diverse vehicles and pedestrians alongside various other reflectors, such as the fire hydrant, grass, and street lamp. As depicted in Figures 1(c) and 1(d), the received mmWave signals include various reflections from objects surrounding the target car. Different materials have distinct reflectivity, for example, the metallic objects reflect signals strongly while the non-metallic objects have weaker reflectivity. Compared with vehicle detection, our system faces challenges in accurately generating bounding boxes for non-metallic entities such as pedestrians with small reflective surfaces in the distance.

## 3 System Design

### 3.1 Overview

*mmBox* aims to develop a front-view object detection system capable of accurately detecting vehicles and pedestrians using bounding boxes in challenging outdoor scenarios. The basic idea of *mmBox* is to leverage deep learning models to extract valuable features from mmWave signals and subsequently generate bounding boxes with depth information for both vehicles and pedestrians. However, the complex mmWave reflection signals make direct utilization for deep learning models difficult. Moreover, previous solutions, such as combining multiple frames as one input to improve density of valuable information, are impractical for moving objects due to the time-consuming nature of capturing and processing multiple frames. To address these limitations, *mmBox* utilizes a heatmap produced by a single frame, which provides a visual representation of mmWave signals in the form of image-like data and describes the energy distribution of reflections. Specifically, we generate two **Range-Azimuth Heatmaps (RAMaps)** to separate reflections of stationary and moving objects, capturing more features compared with a single heatmap. Separating static and dynamic RAMaps makes *mmBox* robust and generalizable across diverse traffic intersections. Recognizing that the mmWave device offers numerous virtual channels on range and azimuth dimensions, and that most movements of vehicles and pedestrians occur in this coordinate space, we initially utilize RAMaps to train the deep learning model and generate predictions. However, even though the RAMaps provide detailed information in the BEV, accurately extracting height features of objects remains challenging due to the restricted height information. Additionally, the convolution filter, which extracts features from a local region of inputs in the range-azimuth coordinate, can lead to a significant shift between predicted and ground truth object location in the elevation-azimuth coordinate. Therefore, we redesign this system to incorporate an **Elevation-Azimuth Heatmap (EAMap)** with static and dynamic RAMaps for higher accuracy. Figure 2 depicts an overview of our methodology, consisting of two key modules: *Heatmap Generation*, responsible for transforming raw mmWave signals into static and dynamic RAMaps to capture stationary and moving objects in the BEV and an EAMap to map reflected points from objects into front view with energy distribution; and a *Multi-Scale Bounding Box Generator*, tasked with extracting features from heatmaps and producing accurate bounding boxes with distance values for vehicles and pedestrians across various scales. To automatically generate ground truth, we develop a *BoxLabeling* program capable of producing bounding boxes and depth values from captured grayscale images and depth images.

### 3.2 RAMaps Only

In this section, we only consider static and dynamic RAMaps for *Heatmap Generation*, which utilizes *Doppler FFT* on mmWaves to distinguish moving and stationary objects and subsequently visualizes them with energy distribution. *BoxLabeling* is applied to automatically produce ground truth bounding boxes with depth for training and evaluating the deep learning model. To correctly predict bounding boxes, *mmBox* employs a *Multi-Feature Fusion Extractor* to fuse features from both RAMaps in three different sizes. As the height and width of inputs reduce, convolutional neural units compute global information from inputs, but the small objects might be ignored in deep layers. To address this issue, *mmBox* designs a *Three-Level Bounding Box Predictor* to adapt different sizes of objects. Since convolution filters in deep layers can capture whole shape features of large objects while preserving more valuable features of small objects in large feature maps, we use a strategy of three-level predictions to separately cover various scale target objects. Instead of using a classifier module to build mapping between features and ground truth values, *mmBox* directly utilizes convolution layers to produce results, which decreases computing costs and speeds up the

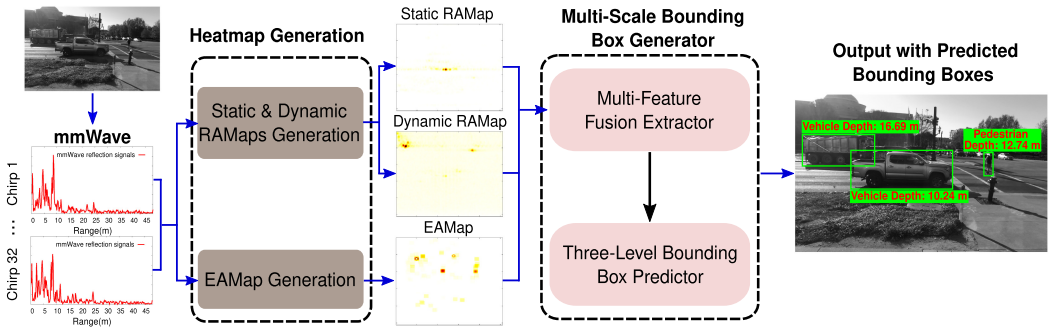


Fig. 2. The two-stage system pipeline of *mmBox*. In the first stage, the 32 chirps of mmWave signals captured from an outdoor street scene are processed by a Heatmap Generation module, which eliminates noise and generates both static and dynamic RAMaps and an EAMap. In the second stage, a Multi-Scale Bounding Box Generator fuses features from heatmaps at three levels and accurately predicts bounding boxes with depth values for vehicles and pedestrians.

model. Furthermore, *mmBox* applies K-means on the ground-truth data distribution to produce predefined anchors, which contain 3 different ratios of height to width at 3 levels. Thus, the *Three-Level Bounding Box Predictor* module only needs to generate the correct offsets based on anchors, accelerating the process of convergence during training. Due to the sparsity of valid bounding boxes in the predictions, fewer positive samples might lead to the gradient disappearance problem in loss calculation. To increase the number of positive samples, we leverage a nearest points strategy to map one target object to multiple closed grids. For loss computation, *mmBox* takes EIoU to compute bounding box loss and Binary Cross Entropy to quantify classification, confidence, and depth value errors. To generate accurate predictions for vehicles and pedestrians, a confidence filter and Non-maximum Suppression are applied to remove low confident and overlapping bounding boxes.

*3.2.1 Heatmap Generation.* *mmBox* uses the received mmWave signals from the objects to predict the 2D bounding boxes of vehicles and pedestrians. Rather than using the raw mmWave signals for object detection, *mmBox* generates the heatmaps in the spatial domains by processing the raw reflected signals from multiple receiver antennas of the mmWave device. Reflections from objects at the mmWave device are complex time domain signals with I and Q channels corresponding to the amplitude and phase. Multiple transmitters and receivers of mmWave devices with fixed spacing enable many virtual antennas in azimuth directions, separated by $\lambda/2$, enabling wider aperture and better azimuth resolution. With known spacing between the virtual antennas, we can look at the tiny phase difference caused by antenna separation to determine the Angle of Arrival (Azimuth Angle), and the **Time of Flight (TOF)** of the mmWave signal determines the range. Adding the reflected energy from the objects with the same azimuth angle and range generates the 2D heatmap. Strong reflective or nearby objects have higher reflected energy and, hence, have strong peaks in the heatmap compared with other noisy reflections. A heatmap represents the object's location in the range and azimuth direction, ignoring minute displacement in the phase caused by noise. Heatmap generation is a two-stage process. First, we generate 2D heatmaps from 1D complex signals and then align the **Field-of-View (FoV)** of mmWave heatmaps with ground-truth ZED camera images.

**Static and Dynamic RAMap Generation:** We use FMCW to transmit the mmWave signal towards the object with a custom-designed frame. In a single frame, it has fast-time samples (single chirp), which capture the reflections to generate the RAMap of the objects. However, we design multiple chirps in a single frame, which are slow-time samples and capture the movement of the

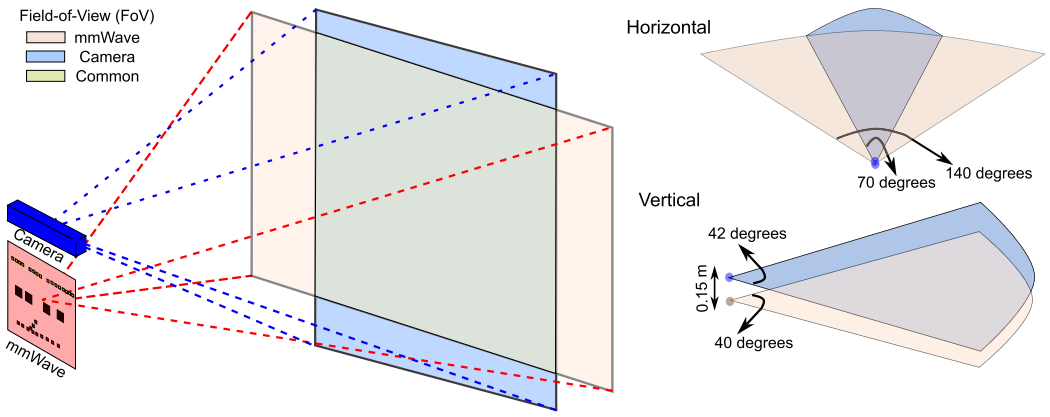


Fig. 3. Field-of-View (FoV) representation of mmWave device and stereo camera.

objects in the environment. Since frame duration is ~ 100 ms, the mmWave signals from static objects in all the chirps of a single frame will be similar, as opposed to dynamic objects. Reflected signals from dynamic objects will show minute changes on the chirps. We use the slow-time samples of the single frame to separate the static and dynamic objects.

Separating mmWave heatmaps into static and dynamic heatmaps maps the surrounding objects into two sets. All the static objects, such as traffic light poles, vegetation and roads, map to the static heatmap. All other moving objects, such as vehicles and pedestrians, map to the dynamic heatmap. Since most of the moving objects across all traffic intersections are pedestrians and vehicles, having a separate dynamic RAMap makes *mmBox* robust and generalizable. Compared with sparse PCD produced by mmWave signals, extracting features from heatmaps is easier and more reliable. Mapping objects into dynamic and static RAMaps can benefit the deep learning model to better distinguish overlapped moving and static objects when the number of target objects in one frame increases. To eliminate noise and generate high-quality heatmaps, *Static and Dynamic RAMap Generation* leverages 4 steps to process raw mmWave reflection signals. *Range FFT* is first applied on raw mmWave reflection to get distance information by changing time domain signals to frequency domain. Due to the moving objects, reflected signals in multiple chirps vary with time. *Doppler FFT* maps objects with different speeds in the different Doppler bins. Then, *Angle FFT* extracts the azimuth angle from Range-Doppler data by applying FFT across unique virtual antenna channels. Finally, we generate a static RAMap by consolidating the middle three Doppler bins and using the remaining bins to generate dynamic RAMap. The camera sensor has a smaller FoV than the mmWave device; thus, we prune the RAMap to match with the ground truth FoV.

**Field-of-View Alignment:** To ensure that both devices look at the same portion of the environment, we need to fix the FoV of the devices and translate them to the same coordinate system (as Figure 3 shows). *First,* with many virtual antennas available in the azimuth dimension, the mmWave device has a wider FoV compared with stereo cameras. An mmWave device has a total of 140 degrees FoV, ranging from -70 to 70 degrees [37], while the stereo camera only has 70 degrees FoV [38]. Similarly, for the vertical FoV, we remove the portion of the camera image that is not covered by mmWave samples since the mmWave vertical FoV is smaller than the camera FoV. Therefore, we ignore the region of mmWave samples outside the FoV of the stereo camera. *Second,* despite placing the mmWave device and stereo camera as close as possible, the center points of the devices are 15 cm apart in the vertical direction and less than 1 cm offset in the horizontal direction. We apply positional Y-offset in the mmWave samples to generate FoV-aligned mmWave and camera data pairs.

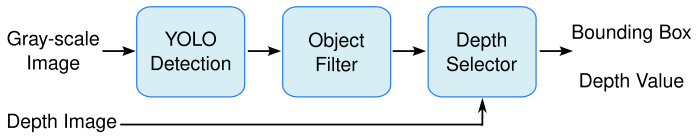


Fig. 4. The flow of *BoxLabeling* for producing a ground truth bounding box and depth value.

*3.2.2 Ground Truth Bounding Box Generation.* The ground truth data for *mmBox* is based on camera images. Manually labeling the dataset is time-consuming as it involves marking bounding boxes, labeling classes, and measuring distances. To streamline this process, we prototyped a self-labeling program, *BoxLabeling*, as shown in Figure 4. *First*, we use the mmWave device and stereo camera to collect mmWave reflections, gray-scale images, and depth images. *Then*, *BoxLabeling* applies a YOLO Detection model [28] to generate 2D bounding boxes from gray-scale images containing different objects with various confidence, an *Object Filter* to choose pedestrian and vehicle objects from different labels and set a threshold of 50% to remove low confident bounding boxes, and a *Depth Selector* to combine depth image to get the correct depth value. Since the valid ranges of depth image and RAMaps are limited to 20 m, the Depth Selector module further filters out bounding boxes more than 20 m in depth. *Finally*, *BoxLabeling* outputs the top-left and bottom-right coordinates of bounding boxes with depth values as ground truth.

*3.2.3 Multi-Scale Bounding Box Generator.* The basic idea of this module is to extract features from static and dynamic RAMaps independently and then precisely predict the center point, width, height, and depth value of bounding boxes for both vehicles and pedestrians. Since the static and dynamic RAMaps capture stationary and moving objects separately, the distinction makes it easier for deep learning models to learn object features from heatmaps, respectively. However, correctly fusing features from different inputs remains a challenge for deep learning models [39]. To address this, *mmBox* designs a *Multi-Feature Fusion Extractor* module to separately extract features from both RAMaps and fuse them in three different scales to improve performance. Considering the significant variability in bounding box dimensions in the front view due to distance and object category variations, *mmBox* employs a *Three-Level Bounding Box Predictor* to match different size objects to three level predictions. To ensure that the features produced by *the Multi-Feature Fusion Extractor* get enough local and global information, *the Three-Level Bounding Box Predictor* uses reverse convolution to upsample global features and concatenate with other layers containing more local features to enrich valuable information. Additionally, directly predicting width and height of the bounding box makes it difficult for a deep learning model to converge, as the width and height vary dramatically. To facilitate this process, *mmBox* defines $3 \times 3$ anchors with different sizes, produced by K-means clustering on ground truth bounding boxes. As a result, the model only predicts offsets of the width and height based on predefined anchors. Figure 5 illustrates the whole pipeline of the *Multi-Scale Bounding Box Generator*.

**Multi-Feature Fusion Extractor:** This module is designed to extract features from two separate RAMaps that capture stationary and moving objects. However, correctly fusing multiple features from different inputs is a challenge for deep learning models. Instead of fusing them in a specific layer, such as at the beginning, the *Multi-Feature Fusion Extractor* concatenates features from both static and dynamic RAMaps in three different feature sizes, which can fuse features more deeply and improve model performance. As depicted in Figure 5(a), the Focus layer first slices the initial heatmap into 4 low-resolution images and expands the channels to 64. To further extract global features, we continuously reduce the height and width of the feature map and utilize Dark Block and **Spatial Pyramid Pooling (SPP)** [40] to extract detailed object information. The core idea of Dark Block (Figure 6(a)) is using multiple shortcut structure, similar to ResNet [41],

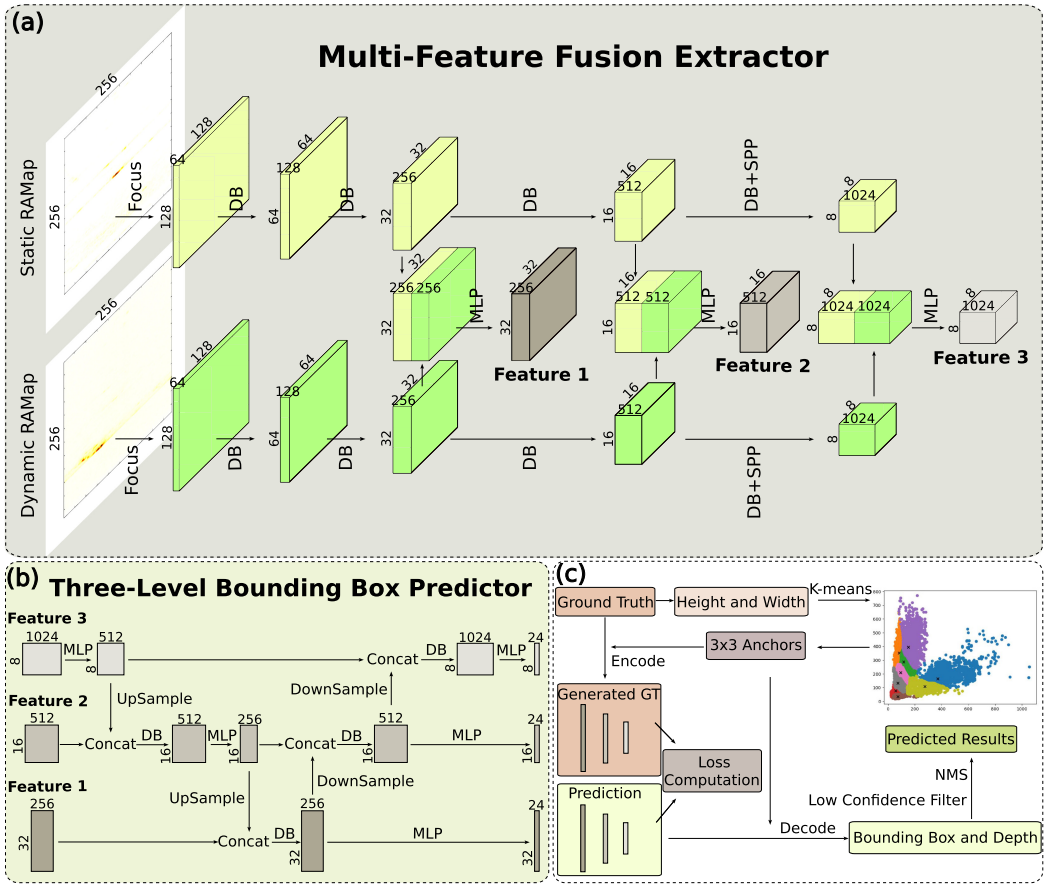


Fig. 5. (a) The architecture of the Multi-Feature Fusion Extractor, which takes static and dynamic RAMaps as inputs and fuses features in multiple scales to output three-level feature maps. DB represents the Dark Block. (b) The architecture of the Three-Level Bounding Box Predictor, which further fuses feature maps and generates bounding box predictions in three levels. (c) The process of generating predefined anchors, computing loss, and producing the actual predictions.

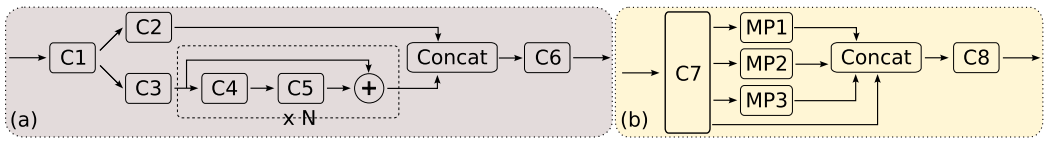


Fig. 6. (a) The architecture of Dark Block. C represents the combination of convolution layers, batch normalization, and LiSU activation function. N indicates the number of times that this residual structure is repeated. (b) The architecture of SPP, where MP represents a MaxPooling2D layer.

to preserve more local features while extracting global object information. To enrich features, SPP (Figure 6(b)) takes 3 MaxPooling2D layers with various kernel sizes to refine features further. Table 1 demonstrates the exact parameters of every layer mentioned in Figure 6. During the process of producing feature maps, we fuse two features from different inputs in equal channels and use **Multiple Layer Perception (MLP)** to generate feature maps in 3 levels.

Table 1. Parameters for Dark Block and SPP

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | MP1 | MP2 | MP3 | C8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Filter Size | 1x1 | 1x1 | 1x1 | 1x1 | 3x3 | 1x1 | 1x1 | 5x5 | 9x9 | 13x13 | 1x1 |
| Stride | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Padding | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 4 | 6 | 0 |
| Channel # (in, out) | (n,2n) | (2n,n) | (2n,n) | (n,n) | (n,n) | (2n,2n) | (2n,n) | (n,n) | (n,n) | (n,n) | (4n,n) |

**ALGORITHM 1:** Process of Generating Predicted Bounding bBox Based on Predefined Anchors

**Input**: outputs of 3-level Bounding Box Predictor $O$; $3 \times 3$ predefined anchors $A$.
**Output**: translated bounding boxes $Results$.
Initialize $Results$ with the same shape of $O$.
**for** every level output $o$ **in** $O$ **do**
  Normalize to (0,1), $o' = Sigmoid(o)$.
  Build grid matrix with the same shape of $o'$.
  **for** each pixel $p$ **in** $o'$ **do**
    Get center points offset $(x, y)$ and height and width offset $(h, w)$ from $p$.
    Find $(grid_x, grid_y)$ according to the position of $p$.
    Calculate real center point, $x' = (x * 2) - 0.5 + grid_x, y' = (y * 2) - 0.5 + grid_y$.
    Get anchors $(anchor_h, anchor_w)$ matched with this level from $A$.
    Calculate real height and width, $h' = (h * 2)^2 * anchor_h, w' = (w * 2)^2 * anchor_w$.
    Scale $(x', y', h', w')$ to real size and save into $Results$.
  **end for**
**end for**
**return** $Results$

**Three-Level Bounding Box Predictor:** The bounding box size of objects can vary significantly based on category and distance. For example, the bounding box of a distant car is notably smaller than one nearby, and the bounding box of a bus is distinctly different from a car. Due to the various scales of bounding boxes, small objects in a feature map might be ignored as the size reduces, whereas large objects pretend to be easily observed by convolution filters in reduced a feature map. Therefore, we implement the *Three-Level Bounding Box Predictor* to predict objects in different scales. As demonstrated in Figure 5(b), the predictor utilizes three feature maps from the *Multi-Feature Fusion Extractor* and finally outputs three-level predictions. The small size of predictions mainly focuses on the large bounding boxes, such as small objects at a short distance, while large-scale predictions consider larger bounding boxes more. Specifically, to further improve performance, this module first upsamples global object features and concatenates them with detailed local features. Then, the combination of all feature maps is downsampled and concatenated with every layer. Finally, MLP is applied to refine the features and accurately predict bounding boxes for vehicles and pedestrians.

**Predefined Anchors:** Due to the variety of bounding boxes for different objects, directly predicting the exact shape of vehicles and pedestrians will take a long time to converge. To speed up the training process, *mmBox* proposes a predefined anchor-based prediction method. Figure 5(c) illustrates the anchor-based approach. K-means is applied to find $3 \times 3$ center points, as well as predefined anchors, from the height and width of ground truth. These $3 \times 3$ anchors are matched with three-level predictions in 3 different shapes. Based on the predefined anchors, the process of generating predicted boxes is defined as Algorithm 1.

**Prediction:** During the training process, *mmBox* directly compares the outputs of the *Three-Level Bounding Box Predictor* with a generated target from ground truth for loss computation and

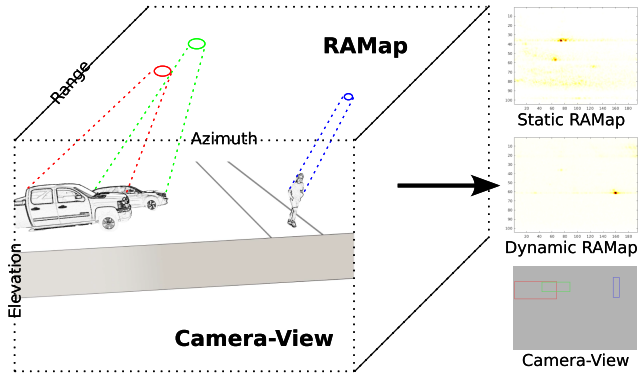


Fig. 7. The limitations of RAMaps. The vehicles and pedestrians are captured in the range-azimuth coordinate by the RAMap, whereas the camera view depicts actual bounding boxes in elevation-azimuth space. Different objects are marked with various colors.

back-propagation, but such predictions cannot be directly used for evaluation and testing. To correctly generate bounding boxes and eliminate redundant results, *mmBox* uses a confidence filter to remove boxes with low confidence and **Non-Maximum Suppression (NMS)** to filter out boxes of the same object (Figure 5(c)). NMS works based on the **Intersection over Union (IoU)** of predicted bounding boxes with the same label. If the IoU of overlapping boxes exceeds a threshold, which is set 0.3 for *mmBox*, NMS will keep the box with the highest confidence and discard other intersecting boxes.

### 3.3 RAMaps + EAMap

As Figure 2 shows, the generated bounding boxes for vehicles and pedestrians are predicted on the elevation-azimuth coordinate (camera view), which includes height information and facilitates humans to recognize the shape of objects. However, the inputs are in range-azimuth space, leading to several problems. (1) Predicting object shapes is challenging because the height information in RAMaps is restricted. When collecting mmWave reflection signals, distances between the measuring point and each part of object are different. For example, if the mmWave device is at the same height as a pedestrian's head, the measured distance of the feet will be greater than the distance of the head. As a result, the height features will be compressed in RAMaps, making it difficult for the deep learning model to extract and predict object shapes accurately. Especially in the outdoor scenes, differences between reflective points of the same target are ignored owing to the increased distance. Figure 7 demonstrates this process, showing that vehicles and pedestrians are mapped to range-azimuth coordinates as RAMaps, but the target spots do not directly provide the shape features of objects. (2) The predicted position of a bounding box might be shifted. Our deep learning model applies the convolution filter to extract features from RAMaps. With the reduction of height and width, the output layers of the *Multi-Feature Fusion Extractor* generate three feature maps for different scale objects. The features in each pixel originate from the limited regions of elevation-azimuth space. However, the positions of objects in RAMaps might shift in the front view. For instance, in Figure 7, the pedestrian is projected to range-azimuth space. Due to the short distance, the spot related to the pedestrian is closer to the bottom of RAMap, but in elevation-azimuth coordinate, the center point of pedestrian appears near the top (see blue box in camera view). Hence, relying solely on RAmaps challenges our system in accurately predicting the actual position of bounding boxes. To address these problems, we redesign *mmBox* to further incorporate the EAMap to improve the accuracy. The EAMap captures shape features of objects

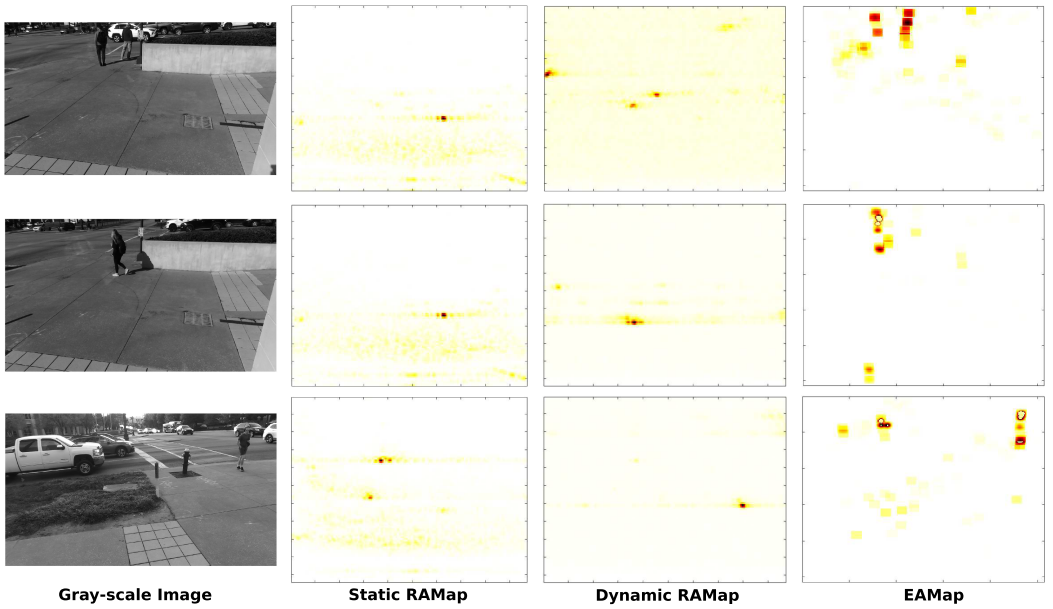


Fig. 8. Examples from a generated EAMap. From left to right, the columns represent the gray-scale image, static RAMap, dynamic RAMap, and EAMap, respectively.

in the same view as prediction; thus, the produced features directly relate to the actual position of the targets. Additionally, the height information is clearly demonstrated on the EAMap, even though it provides limited features due to low elevation resolution of the mmWave device. To fuse the compensated height information from the EAMap, the *Heatmap Generation* module employs *EAMap Generation* to project detected points into the elevation-azimuth space by a Gaussian filter, and the *Multi-Scale Bounding Box Generator* is extended to a three-input model.

*3.3.1 EAMap Generation.* To generate the EAMap, we first process raw mmWave reflection signals to detect points. Then, the detected points are projected to the elevation-azimuth coordinate. Algorithm 2 (Appendix A) details all the steps. To eliminate noise and accurately detect valid points, Constant False Alarm Rate (CFAR) is applied to find overlapped detection in Doppler and Range. Angle FFT can get angles in elevation and azimuth dimensions. Since the mmWave transceiver and camera are in different positions (see Figure 10(a)), we further use offsets in 3D space to calibrate all detected points. Finally, the Gaussian filter takes SNR of calibrated points to calculate energy distribution and map to the EAMap in correct positions, which are decided by the predefined window size of the Gaussian filter and closest positions from projected points in elevation-azimuth space. Figure 8 demonstrates some examples of the generated EAMap, which directly provides the height information of objects and the location of the target related to the front view. For example, in the first and second rows, we can identify pedestrians on the road from gray-scale images. Dynamic RAMaps correctly capture those targets even though some strong reflective objects exist, and EAMaps provide height features in the front view. The third row, including more vehicles and pedestrians, demonstrates RAMaps' effectiveness in capturing valuable information in heatmaps.

*3.3.2 Multi-Scale Bounding Box Generator.* The *Multi-Scale Bounding Box Generator* in the RAMaps-only model mainly focuses on extracting valuable features from dynamic and static RAMaps. However, with the introduction of the EAMap as an input to provide height information

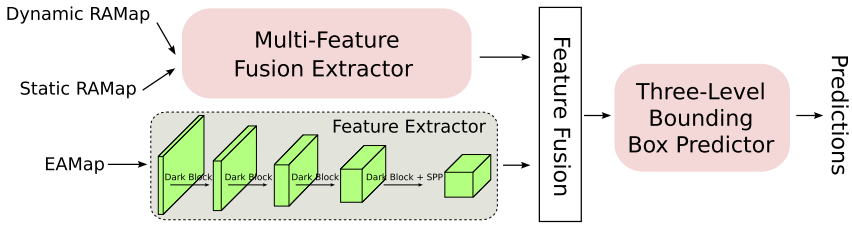


Fig. 9. The architecture of a redesigned *Multi-Scale Bounding Box Generator* is illustrated. It involves to extracting features from an EAMap and fusing with the outputs of a *Multi-Feature Fusion Extractor* for subsequent use in a *Three-Level Bounding Box Predictor*.

and directly related positions, it becomes necessary to redesign the architecture of this model to adapt three inputs (Figure 9). While this new architecture retains most components of the previous model, it builds an additional *Feature Extractor* to capture height and position features. The *Feature Fusion* concatenates features from both parts and employs a multi-layer perceptron (MLP) to generate three-level feature maps for the *Three-Level Bounding Box Predictor*. Specifically, in the *Feature Extractor*, Dark Block and SPP are applied to reduce the height and width of feature maps and to increase channels.

### 3.4 Loss Computation

The final outputs of the *Three-Level Bounding Box Predictor* consist of predictions in three different scales, making loss computation challenging since we cannot directly compare the ground truth bounding box with predicted results. To solve this problem, *mmBox* maps ground truth boxes into three levels, similar to predicted results. The process is detailed in Algorithm 3. First, we initialize three-level grid matrices, the same as predictions. Then, according to the center point coordinates of ground truth bounding boxes, we place these boxes with thresholded shapes based on the ratio of predefined anchors into grid matrices. However, due to the limited number of target objects in a single frame, the valid grids of generated targets are very sparse, which leads to serious convergence problems during loss computation. To address this, we employ a nearest points strategy to map the ground truth bounding box into the three closest positions, thereby increasing the density of targets.

The loss computation is composed of 4 components: bounding box loss, confidence loss, classification loss, and depth loss. Many popular algorithms exist to calculate bounding box loss, such as IoU, defined as

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{1}$$

However, IoU cannot quantify the closeness if predicted and ground truth boxes don not have an overlapped area and, if the prediction covers the entire ground truth box, the IoU will always remain the same regardless of the position changes. GIoU [42] is proposed to include the non-overlapping area as a part of the loss function, solving the no-overlap issue. DIoU [43] uses the center point distance instead of the no-overlapping area, accelerating the convergence process and solving the complete overlap problem. CIoU [43] considers the ratio of height and width to reflect the difference on bounding box shape. Our work uses EIoU [44], defined as follows:

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{(w^c)^2 + (h^c)^2} + \frac{\rho^2(w, w^{gt})}{(w^c)^2} + \frac{\rho^2(h, h^{gt})}{(h^c)^2}, \tag{2}$$

where $b$ and $b^{gt}$ denote the center points of the predicted and ground truth boxes, $w^c$ and $h^c$ represent the width and height of the smallest enclosing box covering the two boxes, and $w$, $h$,

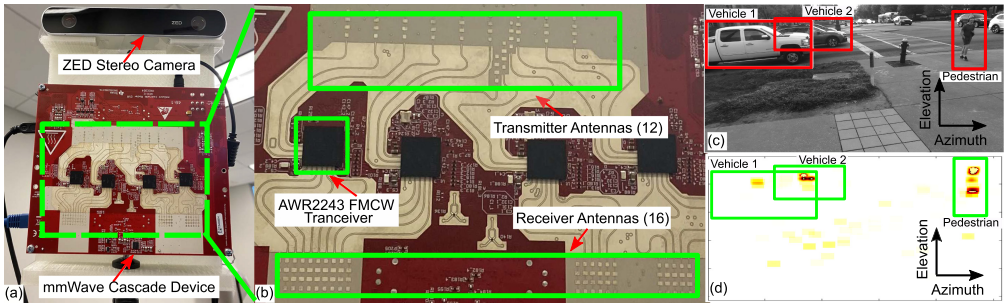


Fig. 10. (a) mmWave cascade device with ZED stereo camera. (b) Zoom-in view of cascaded device with transmit and receive antennas. (c) An example of left image from ZED stereo camera of the environment. (d) EAMap of mmWave signals captured by cascaded device corresponding to the camera image of 10(c).

$w^{gt}$, and $h^{gt}$ are the width and height of the predicted and ground truth boxes. Compared with other IoU-based methods, EIoU considers the IoU loss, center points distance, and difference of height and width, which is faster than CIoU. The loss of classification, confidence, and depth use **Binary Cross-Entropy Loss (BCE Loss)**:

$$L_{BCE}(x, y) = -(y \log(x) + (1 - y) \log(1 - x)), \tag{3}$$

where x is prediction and y is ground truth. While the ground truth does not provide confidence value for boxes, we treat the result of EIoU as the target confidence. In general, the final loss function of *mmBox* is generated as follows:

$$L_{mmBox} = W_a * L_{BCE}(Confidence) * (W_b * L_{EIoU} + W_c * L_{BCE}(Label) + W_d * L_{BCE}(Depth)), \tag{4}$$

where $W$ represents the weight of each component.

## 4 Implementation

### 4.1 Hardware Platform

Due to the lack of open-source data samples of traffic intersections collected by mmWave devices, we built a custom hardware setup with a 3D printed structure to include mmWave cascaded radar and a stereo camera. Figure 10(a) shows the 3D printed structure holding cascaded mmWave radar, TI MMWCAS-RF-EVM and MMWCASDSP-EVM [37, 45], to collect mmWave reflections from surrounding vehicles and pedestrians, and a ZED stereo camera [38] to collect the ground-truth camera and depth images. A **Commercial-Off-The-Shelf (COTS)** mmWave cascaded radar and ZED stereo camera are aligned and kept as close as possible with the 3D printed structure to maximize the overlapped FoV. The mmWave cascaded device combines four chipsets [46] to increase the aperture and achieve high angular resolution (see Figure 10(b)). Each mmWave device chip consists of 3 transmitters and 4 receiver antennas. All the antennas from chipsets are placed to obtain $\lambda/2$ spacing to overcome aliasing. In total, the mmWave cascaded device has 12 transmitters and 16 receivers, providing 192 virtual channels. Among 12 transmitters, 9 are placed in the azimuth dimension with $2\times\lambda$ spacing to obtain 144 virtual antennas in the azimuth direction. Out of 144 virtual antennas, 86 are non-overlapping and are separated by $\lambda/2$, providing an aperture of $43\lambda$ and $1.4^\circ$ angle resolution. The cascade device uses the following data collection parameters: Start frequency, 77 GHz; frequency ramp slope, 25 MHz/$\mu$S; number of complex **Analog-to-Digital Converter (ADC)** samples, 256; ADC sampling rate, 8 MHz/s; chirp duration, 40 $\mu$s; frame interval, 100 ms; and maximum receive antenna gain, 48 dB. The device is capable of collecting data from a maximum range of up to $\sim$ 48 m with a range resolution of 0.19 m and has a total bandwidth of 800 MHz. Our stereo camera collects the left, right, and depth images with a resolution of 1 mm.

Table 2. Number of Different Objects Detected by ZED Stereo Camera on ~ 29K Data Samples at the Traffic Intersection

| Environment | Total Data Samples | Number of Vehicles | Number of Pedestrians |
|---|---|---|---|
| Traffic Intersection | 29110 | 94730 | 25833 |

Since it is an outdoor environment, the depth images are noisy beyond 20 m. Hence, we keep a 20-m working range for the mmWave device and camera.

### 4.2 Real Data Collection

We collect real datasets by placing our custom setup in the camera tripod at a height of 6 ft and facing towards the traffic intersection (see Figure 10(c) for an example). We connect an mmWave device and stereo camera to the host PC with the Windows operating system. We begin the data collection process by running a MATLAB program that initiates the mmWave studio and loads the hardware configuration parameters, such as chirp profile and frame configuration. We also initialize the ZED stereo camera by opening the camera port and configuring it to collect HD1080 images. Since the mmWave device takes longer to configure than the ZED stereo camera, the camera waits for acknowledgment from the mmWave studio before it starts collecting images. In a single data collection, we collect approximately 30 secs, resulting in 300 data samples, each separated by 100 ms. The mmWave device collects the raw ADC data files which are processed to generate a static heatmap, dynamic heatmap for the range-azimuth domain, and a single EAMap similar to the camera viewpoint (see Figure 10(d)). Similarly, the ZED camera provides the left and right RGB and depth images. We use the left image to get the 2D bounding boxes with the pre-trained YOLO model [28] and localize them combined with depth images to get the median depth of the different objects. Since both are different types of COTS hardware, tight synchronization between the mmWave device and the camera is impossible. We solve this problem with software synchronization by finding the closest samples. Even though we try to collect mmWave samples and camera images at 100 ms with software synchronization, we verify the samples by taking **Structural Similarity Index Measure (SSIMs)** [47] between camera images, static RAMaps, and dynamic RAMaps. We observe SSIM between camera images tracks the SSIM between dynamic RAMaps, and the timestamp difference between mmWave heatmaps and camera images is less than 10 ms. Figure 8 shows the synchronized static RAMaps, dynamic RAMaps, EAMaps, and camera images for traffic intersections.

We place our setup facing a traffic intersection so that it can capture various scenarios, such as pedestrians crossing both directions in the road and vehicles passing through the traffic light at different office hours. Table 2 shows the summary of objects presented in our dataset. We consider two object categories in *mmBox*: (1) *Vehicle* for all cars, buses, and trucks; and (2) *Pedestrian* for pedestrians and cyclists. Recall that the traffic intersection is around the campus. Therefore, we have a large number of pedestrians crossing the road. Also, the road has 6 lanes total, with a speed limit of up to 35 mph; therefore, we also observe many vehicles. In total, we collected more than 30,000 data samples over 6 months, resulting in nearly 1 TB of data. After removing data samples without objects within 20 m, we have 29,110 data samples, out of which we use 23,288 samples in training and the rest of the samples for model benchmarking and testing. To validate the robustness of our system, we further collect data from 2 different traffic intersections, such as single-lane roads and intersections with stop signs. We use 80% of 1,355 processed samples to fine-tune the deep learning model and 20% to evaluate the performance.

### 4.3 Foggy Data Collection

To simulate the extreme outdoor environment, we implement a foggy box to collect data in dense fog, as shown in Figure 11(a). The foggy box is made of clear plexiglass, which allows mmWave

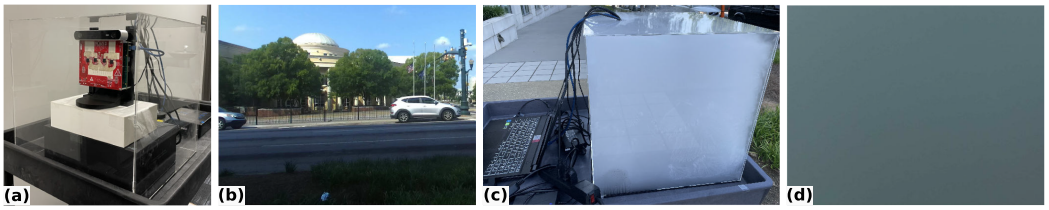


Fig. 11. (a) Foggy box implementation. The outside box is made of clear plexiglass. The mmWave device and camera are put on the 3D printed structure, and the fog machine is placed behind them. (b) RGB image collected by the ZED camera inside the foggy box without fog. (c) The box filled with dense fog during data collection. (d) RGB image collected by camera inside the foggy box with dense fog.

to pass through while maintaining dense fog inside. The cascaded mmWave radar and ZED stereo camera, held by a 3D printed structure, are placed inside the box. To generate dense fog, we put a Donner DMF-400S fog machine behind the radar and camera to simulate dense fog. To ensure that this clear plexiglass box does not influence the quality of the collected images, Figure 11(b) shows an RGB image example captured by the camera inside the box without fog. The objects and surroundings are clearly visible through the foggy box. Before starting data collection, the box is filled with dense fog (Figure 11(c)). This extreme foggy weather simulation makes a commonly used camera ineffective, capturing nothing (see Figure 11(d)). Therefore, we place the ZED camera outside the foggy box to collect ground truth data. The radar setup for data collection remains the same as previously described. In total, we captured 2,392 samples in the traffic intersection scene. Due to the difference of position and orientation, we use 1,595 frames to fine-tune the pre-trained model and the remaining 797 samples to validate the performance of *mmBox*.

### 4.4 Network Training

To train our network, we initialize the weights of *mmBox* from a normal distribution with a mean of 0 and a standard deviation of 0.02. The learning rate starts from 0.0001 and automatically decays by a multiplicative factor of 0.5 after 10 epochs. To minimize the loss between prediction and ground truth, we apply Adam [48] as optimizer to update the parameters of the model, as it can effectively learn the mapping between inputs and outputs. Since our model is anchor based, we use K-means clustering to calculate 9 center points from the width and height of ground truth boxes and distribute them across 3 layers with 3 different shapes. Our networks converge within 200 epochs. To fine-tune *mmBox* with foggy data, we load the pre-trained model and optimize it with an initial learning rate of 0.00001 by 200 epochs. Our models are implemented using PyTorch with Python 3.7 and trained on a Tesla v100 **graphics processing unit (GPU)** server.

## 5 Experimental Results

We now evaluate the performance of *mmBox* and analyze the model design. We validate our model with the commonly used metric *Average Precision* as the primary evaluation method. Additionally, we quantify the shape difference using the *Height and Width Error*, evaluate the precision of the predicted object locations using the *Center Point Distance*, and analyze *Depth Error* as *mmBox* produces depth values for distances between the objects and the device. We also consider *Missed Objects per Frame* and *Miss Rate* for evaluating the effectiveness performance. *mmBox* features the robustness of working under extreme environments; thus, we explore the performance under heavy fog conditions. To evaluate the robustness and reliability of our system, we collected more data from different surroundings to fine-tune and test our model. Notably, there is no similar model available for comparison because the previous works, for example, Radatron and RODNet,

provide limited information and detect objects in the BEV as compared with our front-view work. In addition, we try to analyze the effect of major components and the complexity of this system to get better understanding of *mmBox*.

**Intersection over Union (IoU):** IoU measures the ratio of the overlapping area between two bounding boxes to the region of union, which is a commonly used method to measure the similarity of the predicted bounding box and ground truth. In our evaluation methodology, IoU is used as a threshold for the confusion matrix, which decides whether the object prediction is correctly matched.

**Average Precision (AP) and mean Average Precision (mAP):** Precision represents the proportion of true-positive predictions in all generated bounding boxes, whereas Recall measures the ratio of true-positive predictions to all ground truth targets. By setting the IoU threshold with a specific value, we can separately draw a **Precision Recall (PR)** curve for vehicles and pedestrians. *AP* and *mAP* calculated from a PR curve, which adapt the all-interpolated method [49] outperforming the 11-interpolated from [50], are important metrics for evaluating model performance in bounding box detection.

**Height and Width Error:** The shape of the bounding box can be directly reflected by its height and width. Therefore, to evaluate the accuracy of the predicted shape, we quantify the *Height Error* as the difference between the ratio of generated height to ground truth height and 1. Similarly, the *Width Error* is calculated in the same manner.

**Center Point Distance and Depth Error:** The center point, a crucial element that influences the quality of predictions, is derived from the predicted values and the grid coordinates of the *Three-Level Bounding Box Predictor* outputs. It represents the exact location of the vehicle or pedestrian. The location error of prediction can be quantified by the distance between the predicted center point and ground truth, which is represented in pixels as the object distance on a 2D image cannot be measured in meters of Euclidean geometry. Additionally, we further generate the depth value to accurately measure object distance. The *Depth Error* is calculated as the difference between depth of prediction and ground truth.

**Missed Objects per Frame and Miss Rate:** To evaluate the effectiveness performance of *mmBox*, we introduce *Missed Objects per Frame* and *Miss Rate*. *Missed Objects per Frame* represents the ratio of the missed objects to total number of frames, whereas *Miss Rate* is calculated by the proportion of the missed objects in total objects.

**Evaluation Summary:** (1) The RAMaps+EAMap model outperforms the RAMaps-only model by achieving $mAP_{30}$, $mAP_{50}$, and $mAP_{75}$ scores of 0.716, 0.659, and 0.508, respectively, compared with 0.424, 0.345, and 0.225. This model yields a 125.8% improvement on $mAP_{75}$. The results highlight the significant effectiveness of the compensatory information provided by EAMap in enhancing shape, location, and depth accuracy of predictions. (2) By simulating the dense foggy weather using a prototype foggy box, we demonstrate the robustness of *mmBox* in generating bounding boxes in extreme outdoor environments. It achieves the $mAP_{50}$ score of 0.688 even when the camera could not capture anything for object detection. (3) The use of predefined anchors and a three-level predictor design is more effective than the anchor-free method. Additionally, in the comparison of different feature fusion approaches, *mmBox* exhibits higher performance compared with multiple crossed fusion work. (4) The low latency of 27.2 ms of the deep learning model underscores the potential feasibility of implementing our work in real-time systems.

## 5.1 Performance Analysis

*5.1.1 RAMaps-only.* Figure 12(a) illustrates the performance of *mmBox* on $AP_{30}$, $AP_{50}$, and $AP_{75}$ with IoU thresholds of 0.3, 0.5, and 0.75, respectively, resulting in $mAP_{30}$, $mAP_{50}$, and $mAP_{75}$ values of 0.424, 0.345, and 0.225, respectively. To explore more metrics, Figures 12(b) to 12(d) present

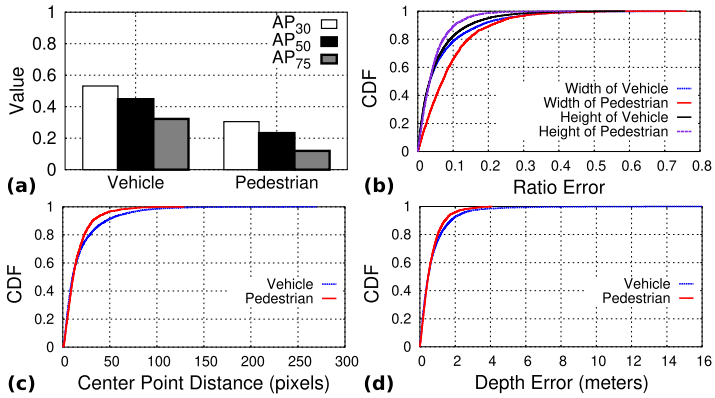


Fig. 12. (a) $AP_{30}$, $AP_{50}$, and $AP_{75}$ for vehicles and pedestrians. (b) CDF of Width and Height Error for vehicles and pedestrians. (c) CDF of Center Point Distance in pixels between prediction and ground truth. (d) CDF of Depth Error in meters between prediction and ground truth.


Fig. 13. (a) mAP comparison between RAMaps-only and RAMaps+EAMap model under the IoU threshold of 0.30, 0.50, and 0.75. (b) Predictions of RAMaps-only model. (c) Predicted bounding boxes of RAMaps+EAMap model.

the **Cumulative Distribution Function (CDF)** of the predicted *Height and Width Error*, *Center Point Distance* in pixels, and *Depth Error* under the IoU threshold of 0.5. The $50^{th}$ percentiles of *Height and Width Error* of vehicles are 0.033, whereas the pedestrian achieves 0.063 and 0.035 in *Height and Width Error*. This small difference indicates that the predictions of our model are very similar to the ground truth on shape. The $50^{th}$ percentile of *Center Point Distance* for vehicles is 10.25 pixels, and pedestrians achieve 11.59 pixels. Given that the generated bounding boxes are designed for HD1080 images, such small center point offsets still accurately describe the exact positions of objects. The *Depth Error* of vehicles and pedestrians in the $50^{th}$ percentile are 0.504 m and 0.478 m. With an IoU threshold of 0.5 to match predicted bounding boxes and ground truth, the *Missed Vehicles and Pedestrians per Frame* are 1.54 and 0.56, and the *Miss Rates* for vehicles and pedestrians are 47% and 63%, respectively. *In summary, the results of various metrics show great potential to predict bounding boxes for vehicles and pedestrians, but the accuracy needs to be improved.*

*5.1.2 RAMaps + EAMap.* While the static and dynamic RAMaps provide detailed information in the BEV (range-azimuth plane), extracting the height features of objects is challenging. Additionally, the actual position of an object in the front view differs from that in both RAMaps. Therefore, we further use EAMap, projected by the mmWave point cloud, as an additional input to improve the accuracy of predicted bounding boxes. As Figure 13(a) demonstrates, combining the static and dynamic RAMaps with EAMap significantly improves the *mAP* performance. Compared with the RAMaps-only model, this model achieves improvements of 68.9%, 90.9%, and 125.8% in $mAP_{30}$, $mAP_{50}$, and $AP_{75}$, respectively. To examine the similarity on shape, Table 3 compares the *Width and*

Table 3. Performance Comparison between RAMaps-only and RAMaps+EAMaps Model under Various Metrics for Vehicles and Pedestrians

| | $50^{th}$% WE | $50^{th}$% HE | $50^{th}$% CPD | $50^{th}$% DE | MOF | Miss Rate |
|---|---|---|---|---|---|---|
| $Vehicle_{RAMaps-only}$ | 0.033 | 0.033 | 10.25 pixels | 0.504 m | 1.54 | 47% |
| $Vehicle_{RAMaps+EAMap}$ | 0.017 | 0.018 | 4.81 pixels | 0.272 m | 1.02 | 32% |
| $Pedestrian_{RAMaps-only}$ | 0.063 | 0.035 | 11.59 pixels | 0.478 m | 0.56 | 63% |
| $Pedestrian_{RAMaps+EAMap}$ | 0.043 | 0.023 | 6.21 pixels | 0.367 m | 0.23 | 26% |

Metrics: $50^{th}$% Width Error (WE), $50^{th}$% Height Error (HE), $50^{th}$% Center Point Distance (CPD) in Pixels, $50^{th}$% Depth Error (DE) in Meters, Missed Objects Per Frame (MOF), and Miss Rate.
We use the IoU threshold of 0.5 for all evaluations.

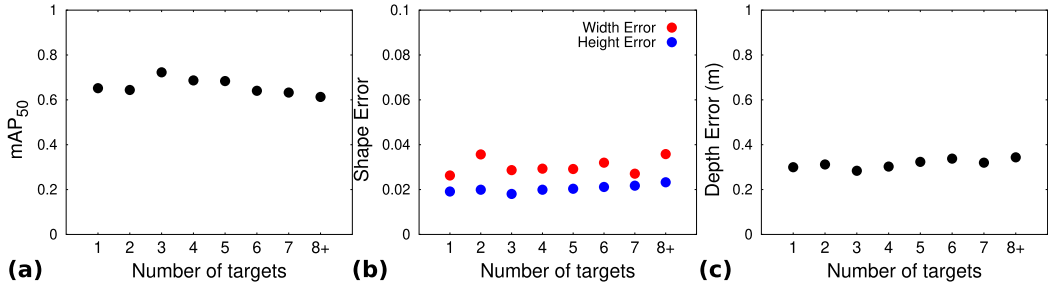


Fig. 14. Performance analysis on $mAP_{50}$ (a), Shape Error (b), and Depth Error (c) with different targets in one frame. The samples, including 8 and more than 8 targets, are counted in 8+. The evaluations of Shape Error and Depth Error set the IoU threshold as 0.5.

*Height Error* and *Center Point Distances* in pixels between the two models, using an IoU threshold of 0.5. The results indicate the reduced differences in shape and position. The lower height error demonstrates the effectiveness of our design to extract height information from low-elevation resolutions of mmWave signals. Figures 13(b) and 13(c) display the prediction comparison for the same sample. The RAMaps-only model predicts bounding boxes for pedestrians with some offsets, because the RAMaps cannot provide corresponding positions in the front view. In contrast, the RAMaps+EAMap model correctly performs the object position and further improves accuracy on the height of bounding boxes due to the detailed height information from EAMap. The $50^{th}$% of *Depth Error* is improved by 0.232 m and 0.111 m for vehicles and pedestrians, respectively. For effectiveness, this model also performs better in terms of *Missed Objects per Frame* and *Miss Rate*, particularly reducing the *Miss Rate* for pedestrians to 26%. To analyze the performance changes of different targets in a single sample, we further evaluate the $mAP_{50}$, *Width Error*, *Height Error*, and *Depth Error* in 8 cases (Figure 14). *mmBox* maintains high accuracy across all cases, though $mAP_{50}$ for cases with more targets is slightly lower compared with those with less targets. Specifically, when the number of targets reaches 8 or more, the $mAP_{50}$ degrades to 0.613, which only drops about 7% compared with the overall performance of 0.659. The predicted shapes of bounding boxes exhibit small differences with ground truth, despite the *Width Error* fluctuating more than *Height Error*. Our system shows similar performance on generated depth between targets and the mmWave device. *In general, the RAMaps+EAMap model outperforms the RAMaps-only model in all evaluation metrics. Our system is robust enough to accurately predict bounding boxes for vehicles and pedestrians with low error across various numbers of targets.*

*5.1.3 Performance under Foggy Conditions.* Compared with other sensors, such as LiDAR and cameras, mmWave is resilient in adverse outdoor environments. For example, mmWave signals

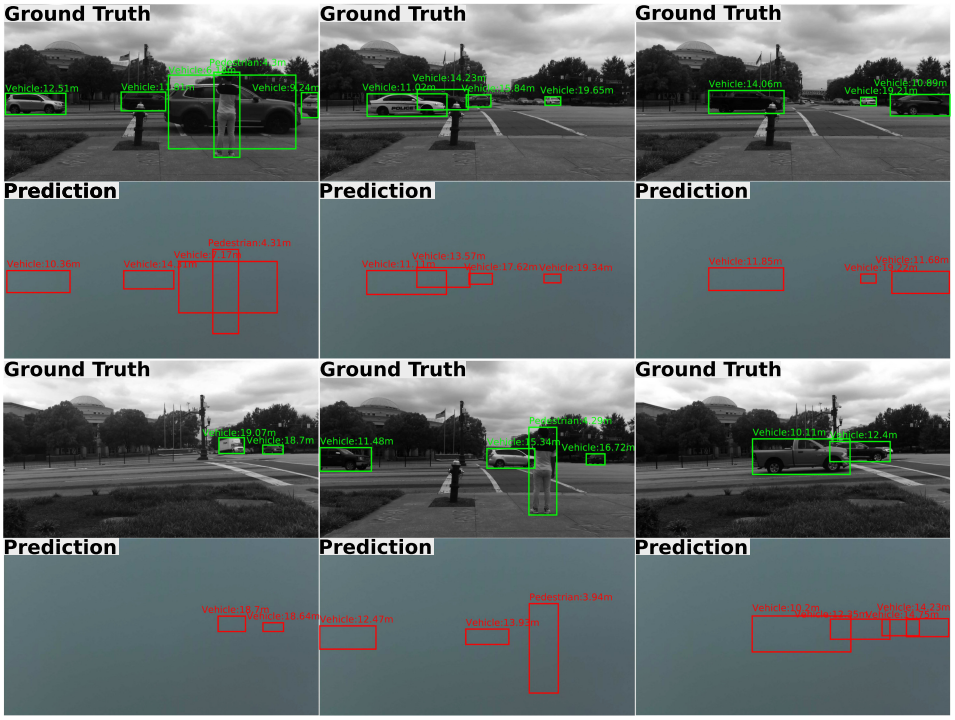


Fig. 15. Examples of *mmBox* performance under a foggy environment. The pictures labeled with ground truth show the actual bounding boxes on real images outside foggy box, and the pictures labeled with prediction show the predicted bounding boxes on foggy images captured by a camera. The text over the bounding boxes indicates the class and depth of the objects.

can penetrate fog to reach the surface of an object and get reflections, whereas LiDAR and cameras will be impeded by the dense fog particles. Therefore, to evaluate *mmBox* performance under extreme weather conditions, a foggy box is applied to collect data under dense fog in real traffic intersections. Since the position and orientation of collecting foggy data are different from previous data collection, we further fine-tune the RAMaps+EAMap model and evaluate under the same metrics to compare the performance. Figure 15 demonstrates some examples of *mmBox* predictions. The ground truth shows the actual bounding boxes in the front-view image, whereas the prediction in the same column represents predicted bounding boxes on a foggy image captured by a camera. Even in the dense fog, when the camera cannot capture any surrounding information, our system *mmBox* can still accurately predict bounding boxes and depth values of objects, which enables many potential applications for road safety. The $mAP_{50}$ on foggy data is still high at 0.688, indicating that our system can actually work under severe outdoor weather conditions. *As shown in Table 4,* mmBox *demonstrates robust and consistent performance across various metrics, underscoring its effectiveness in adverse weather conditions.*

*5.1.4 Robustness to Environmental Dynamics.* To evaluate the robustness of *mmBox* in different environments, we used the same setup to collect data in more road conditions, such as intersections with stop signs (Figure 16(a)) and single-lane roads (Figure 16(b)). In total, we processed 1,355 samples from new environments to fine-tune and evaluate our system. Figure 16(c) presents the testing results on the new dataset. Although the score of the new dataset is slightly lower than

Table 4. Performance Analysis on Foggy Data under Various Metrics for Vehicles and Pedestrians

| | $mAP_{50}$ | $50^{th}$% WE | $50^{th}$% HE | $50^{th}$% CPD | $50^{th}$% DE | MOF | Miss Rate |
|---|---|---|---|---|---|---|---|
| Under fog | 0.688 | 0.045 | 0.026 | 6.82 pixels | 0.391 m | 0.68 | 26% |
| Without fog | 0.659 | 0.030 | 0.021 | 5.51 pixels | 0.320 m | 0.63 | 29% |

Metrics: $mAP_{50}$, $50^{th}$% Width Error (WE), $50^{th}$% Height Error (HE), $50^{th}$% Center Point Distance (CPD) in Pixels, $50^{th}$% Depth Error (DE) in Meters, Missed Objects Per Frame (MOF), and Miss Rate.
We use the IoU threshold of 0.5 for all evaluations.


Fig. 16. (a) An example from a new collected dataset. The car is parking in front of stop sign. (b) An example of a single-lane road from a new collected dataset. (c) The performance comparison on $mAP_{50}$ for the original dataset and new dataset from different environments.

the original dataset, *mmBox* still achieves high performance on $mAP_{50}$, 0.585, demonstrating the robustness and reliability of our system across various conditions.

### 5.2 Ablation Study

To comprehensively understand the *Multi-Scale Bounding Box Generator*, we now analyze its major designs. Due to the variety of ground truth bounding boxes, we apply predefined anchors generated from K-means as the basic knowledge to calibrate the predictions, which facilitates the training process. These anchors map different sizes of bounding boxes into three level targets, limiting the shape range of objects in every layer. To verify the effects of our anchor-based method, we train a model that directly predicts actual shape and generates various bounding boxes in one layer prediction instead of the three-level design. Fusing features in deep learning models is challenging for multiple inputs. This work fuses both RAMaps and EAMap in three different feature sizes. We further compare the functionality of the *mmBox* design with other fusion approaches.

*5.2.1 Effect of Predefined Anchors and Three-Level Predictor.* This work aims to extract features from mmWave and generate accurate bounding boxes for vehicles and pedestrians in the front view. However, the shapes and sizes of target boxes vary significantly due to the different distances and categories. For example, a car parked in a close area appears larger than one at a farther distance in a camera image. This variety makes it challenging for deep learning models to converge and predict accurately. Hence, we take predefined anchors as basic knowledge to map with different shapes and sizes, accelerating the training process and increasing the accuracy. To explore the effect of this anchor-based method, we remove the anchors design to directly predict values for width and height of bounding boxes. In addition, *mmBox* uses three-level predictions to distinguish different scale bounding boxes and, for every layer, three anchors with different shapes are applied to match various objects. Since the predefined anchors map ground truth boxes into 3 levels using the threshold on ratio of width and height, we validate the functionality of the three-level predictor design simultaneously. In the *Three-Level Bounding Box Predictor*, we keep most layers the same and only consider the first prediction of original outputs as the final result.

Table 5. Performance Comparison between Anchor-Based and Anchor-Free Design under Various Metrics

| | $mAP_{50}$ | $50^{th}$% WE | $50^{th}$% HE | $50^{th}$% CPD | $50^{th}$% DE | MOF | Miss Rate |
|---|---|---|---|---|---|---|---|
| Anchor-based | 0.659 | 0.030 | 0.021 | 5.51 pixels | 0.320 m | 0.63 | 29% |
| Anchor-free | 0.457 | 0.032 | 0.023 | 8.36 pixels | 0.304 m | 0.99 | 49% |

Metrics: $mAP_{50}$, $50^{th}$% Width Error (WE), $50^{th}$% Height Error (HE), $50^{th}$% Center Point Distance (CPD) in Pixels, $50^{th}$% Depth Error (DE) in Meters, Missed Objects Per Frame (MOF), and Miss Rate.
We use the IoU threshold of 0.5 for all evaluations.

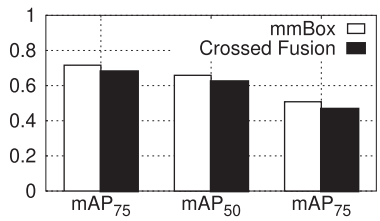


Fig. 17. mAP comparison of the fusion design under IoU thresholds 0.3, 0.5, and 0.75.

Additionally, we adapt the loss computation for one-layer anchor-free prediction to optimize the model. After training, we analyze performance using the same metrics. Table 5 provides a detailed comparison of anchor-free and anchor-based design. To better understand the gap, we use the mean value of vehicles and pedestrians on all metrics. We observe that *the anchor-based design achieves better scores on every evaluation aspect, aligning with our expectations.*

*5.2.2 Effect of Feature Fusion Method.* Our mmWave device has more virtual antennas to achieve better resolution in range-azimuth coordinates; the motion of vehicles and pedestrians primarily occurs in this space. Therefore, we first generate RAMaps from mmWave reflection signals. Compared with projecting point cloud data from mmWave signals to one heatmap, dynamic and static RAMaps can capture more valuable features of moving and stationary objects in complex outdoor environments. Our work, *mmBox*, fuses different inputs at the feature level, which allows us to deeply extract and fuse features. While other works [51, 52] propose an extreme fusion method to combine features, our work only fuses features in output layers of the feature extractor. To validate the importance of our design, we implement the crossed fusion architecture based on our model. Since the features from static and dynamic RAMaps are in the same plane, we fuse them in more layers while keeping other parts of the model the same to examine the performance. Figure 17 shows the $mAP_{30}$, $mAP_{50}$, and $mAP_{75}$ as a main comparison between the two fusion methods. Even though the performance of crossed fusion design is very close to *mmBox*, it is still lower on mAP evaluations. The $mAP_{50}$ of the crossed fusion model is degraded by 0.036 compared with 0.659, that of *mmBox*. Table 6 demonstrates that the crossed fusion model can predict bounding boxes for target objects, as indicated by *Missed Objects per Frame* and *Miss Rate*. However, the shapes and center points of predicted boxes show more difference than *mmBox*. Overall, the results confirm that our design decision to fuse features only in three output layers of the feature extractor is effective. *While the crossed fusion design has comparable performance, it shows slightly lower mAP scores and greater differences in the shapes and center points of predicted boxes. Thus, our approach,* mmBox*, provides a better performance with lower computational complexity, resulting in more accurate predictions.*

Table 6. Performance Comparison between *mmBox* and Crossed Fusion Design under Various Metrics for Vehicles and Pedestrians

| | $50^{th}$% WE | $50^{th}$% HE | $50^{th}$% CPD | $50^{th}$% DE | MOF | Miss Rate |
|---|---|---|---|---|---|---|
| *mmBox* | 0.030 | 0.021 | 5.51 pixels | 0.320 m | 0.63 | 29% |
| Crossed Fusion | 0.060 | 0.038 | 11.84 pixels | 0.600 m | 0.67 | 32% |

Metrics: $50^{th}$% Width Error (WE), $50^{th}$% Height Error (HE), $50^{th}$% Center Point Distance (CPD) in Pixels, $50^{th}$% Depth Error (DE) in Meters, Missed Objects Per Frame (MOF), and Miss Rate.
We use the IoU threshold of 0.5 for all evaluations.

Table 7. Parameters, FLOPs, Latency, and FPS of Deep Learning Model

| Parameters | FLOPs | Latency | FPS |
|---|---|---|---|
| 137.94 M | 53.0 G | 27.2 ms | 36.8 |

### 5.3 Complexity of the Model

For a low-latency system, the costs of generating accurate results significantly affect the user's experience. The object detection on road requires low response time to detect vehicles and pedestrians in emergencies. To evaluate the complexity of *mmBox*, we run this deep learning model on a GPU server with Tesla v100 and analyze the number of parameters, **Floating Point Operations Per Second (FLOPs)**, Latency, and FPS (Table 7). While processing one frame of mmWave reflections currently takes 700 ms, this can be reduced to ~ 230 ms by parallelizing the loading and calculating processes. *The results of the deep learning model show a low latency of 27.2 ms to correctly generate bounding boxes for the real street scene, which enables the potential feasibility of deploying* mmBox *in a real-time system.*

## 6 Related Work

### 6.1 Indoor and Outdoor Millimeter-Wave Sensing

Millimeter wave's attributes, such as working under low light, penetrating small objects, and low cost, attract researchers to apply it to indoor and outdoor sensing systems. Traditional works rely on improving hardware devices and imaging algorithms to get better resolution. UWBMap [53] utilizes SAR, which can compose a virtual antenna array to emulate a large-aperture radar by moving the device along with a specific trajectory and then building the indoor floor plan through smoke. UWBMap develops the prototype with the help of a commercial mobile robot to control moving speed and direction, which heavily limits its applications. To describe environment information, point cloud data is an efficient way compared with Meshes and Voxels. MilliPCD [32] builds a two-stage model to generate high-resolution point cloud data for indoor environments, which shows an example of combining traditional mmWave processing algorithms with deep learning models to further improve mmWave sensing performance. However, indoor environments, which only need to consider walls, ceilings, floors, and the like, is much simpler than outdoor surroundings, which include plenty of objects with variable reflectivity. MilliPoint [54] extends SAR imaging into the outdoors with consumer-grade vehicle radar. Radar self-tracking enables SAR to work along linear motion of vehicles by modeling the relationship between signal variation and radar movement at millimeter-level precision. However, MilliPoint also inherits the disadvantages of SAR. It requires that the object and radar device cannot be quasistatic and keep linear motion. The authors of [26] propose a mmWave radar and RGB-D fusion system to detect obstacles outdoors. The RGB-D camera provides detailed features as input for deep learning models to classify and get

positions, whereas mmWave generates velocity and 2D coordinate location. However, the camera performance will be heavily degraded under extreme outdoor conditions, such as the evening with insufficient light and foggy weather. In contrast, *mmBox* applies mmWave sensing to real outdoor street scenes, which is much complex than the indoors. To prove the reliability of our system, we further simulate dense foggy conditions to evaluate the performance under adverse weather. *The results show that* mmBox *is resilient to accurately detect vehicles and pedestrians in extreme outdoor environments.*

### 6.2 Object Detection Based on Millimeter-Wave

Object detection is one of fundamental computer vision problems in attempting to provide valuable information for environment semantic understanding. In the past works, generating bounding boxes for object detection can be briefly categorized into 2 classes by data source. (1) Camera-Based Works. The optical camera sensor is the most common tool to generate images, which captures ambient features from surroundings. YOLO [55] utilizes a one-stage model, which frames object detection as a regression problem, to predict separated bounding boxes and classification probability based on image data. However, camera sensors cannot work in low or no-light environments, which is common in outdoor scenes, even though there are some works concentrating on enhancing performance under low light. For example, MSR-net [56] showcases a supervised machine learning model designed to capture the end-to-end mapping between dark and bright images, enabling the reconstruction of intricate details. However, its effectiveness is constrained in low-light or completely dark environments. (2) LiDAR-Based Works. LiDAR can proactively emit pulsed light waves, which enables LiDAR to work without light. High-resolution LiDAR sensors can scan the surrounding environment and generate dense PCD. YOLO3D [57] employs PCD generated from LiDAR sensors to capture 3D space information and extends YOLOv2 to perform 3D bounding boxes for object detection. However, even though with the increased practical application of LiDAR the cost has been lower, LiDAR is still expensive compared with other sensors [58], such as the millimeter wave sensor. Besides, in harsh outdoor environments, LiDAR's performance will be heavily degraded, such as in foggy weather, because the pulsed light wave will be impeded by small obstacles.

Since the mmWave sensor can work under low visibility or no-light environments and harsh weather, some articles apply mmWave on outdoor object detection. The authors of [59] present a PointNet-based model that uses PCD generated from high-resolution radar to estimate 2D bounding boxes. The results show a great potential for outdoor mmWave application on object detection, but the environment of collecting data is strictly controlled, which only includes one testing car. RODNet [25] takes the generated radiofrequency images from mmWave and predicts the likehood of objects in the radar's FoV, from which it is hard to recognize the shape of objects. Radatron [24] uses high-resolution cascaded MIMO radar, which achieves in 5-cm range resolution and 1.2° in angular resolution, to collect a dataset. This proposed model can generate accurate bounding boxes from RAMap, but Radatron only considers vehicles on the road, and the prediction in the BEV makes it difficult for humans to understand compared with the detected result in the front view. Considering the attractive attributes of mmWave, some sensor fusion works [60–62] explore the object detection methods that combine LiDAR and radar to overcome the measurement ambiguities and sparsity of mmWave. MVDNet [63] and ST-MVDNet [64] implement the fusion systems that operate in foggy weather to accurately detect vehicles. However, under extreme foggy conditions, LiDAR's ability to gather surrounding information is significantly hindered, leaving only radar working, which greatly degrades the performance of sensor fusion systems. *Therefore, our mmWave-only system,* mmBox*, effectively extracts features from heatmaps and then fuses them in different levels to generate bounding boxes for vehicles and pedestrians in the front view, which is*

*intuitive and allows humans to easily discern the shape of objects. Importantly, the generated EAMap compensates the height information and improves the location accuracy, and the separate dynamic and static RAMaps make* mmBox *robust to adapt to different environments.*

## 7 Discussion and Future Works

**Time Complexity of Data Processing:** We collect raw mmWave reflections from outdoor street scenes and then use the *Heatmap Generation* module to generate both static and dynamic RAMaps and EAMap. Even though the deep learning model, *Multi-Scale Bounding Box Generator*, can generate results in a low latency of 27.2 ms, the processing time of generating heatmaps, consuming 700 ms, heavily degrades the capability of real time. To solve this problem, we can parallel the process to generate RAMaps and EAMaps with different processing threads. Parallelization allows us to load the raw ADC files and use different threads to generate mmWave heatmaps. It will help to reduce the processing time to ∼ 230 ms from 700 ms. Since we use a general-purpose **central processing unit (CPU)** to process data, dedicated hardware [65] can reduce process time and bring data processing time into real time.

**Improving Performance:** *mmBox* has achieved impressive results across various metrics, yet there is room to improve the accuracy. For instance, in the six group examples of Figure 15, the first prediction missed one car compared with the ground truth, and the last prediction generated more bounding boxes than expected. There are two possible approaches to enhance performance. (1) Collecting a larger dataset. The total number of samples used by *mmBox* is 29,110 frames, which is significantly smaller than the benchmark dataset in the computer vision field, such as ImageNet [66]. Since the ability of generalization is strongly correlated with the size of the dataset, collecting more samples is expected to yield higher scores. (2) Improving the quality of ground truth. This work developed a self-labeling program, *BoxLabeling*, to detect objects using YOLO on gray-scale images and extract corresponding distance values from depth images. Although we have considered most situations affecting quality of labeled ground truth boxes, such as discarding samples without depth values, there still exist some issues due to the inaccuracy of YOLO detection and depth error caused by the ZED camera. Using more reliable equipment to detect depth and manually checking the dataset might resolve these problems.

**Extending to 3D Bounding Box:** Our work, which involves predicting 2D bounding boxes for vehicles and pedestrians in outdoor environments, can be extended to 3D bounding box generation. However, 3D object detection is challenging, since the mmWave signals reflected by objects primarily come from the side facing the transceiver, which is insufficient to construct the 3D shape of objects. Consequently, extending this work necessitates a more robust and generalized design for the deep learning model. Moreover, aside from predicting the length, width, and height of object, the generated 3D bounding box also needs to account for the rotation angle on all three axes. For example, when the car is turning left, the rotation angle on the vertical axis must be adjusted. We plan to address these challenges in future work.

## 8 Conclusion

In this work, we design a two-stage system, *mmBox*, which leverages a deep learning model, *Multi-Scale Bounding Box Generator*, to effectively extract features of RAMaps and EAMap produced from a single-frame mmWave signal, correctly fuse features, and accurately generate three-level predictions of bounding boxes and depth values for vehicles and pedestrians. We evaluate the performance of *mmBox* under dense foggy conditions to simulate adverse outdoor weather. The comprehensive metrics of performance analysis prove the effectiveness and robustness of *mmBox* to detect objects with depth, and the low latency of the deep learning model underscores its potential for real-time applications.

## A Appendix

**ALGORITHM 2:** The Process of Generating EAMap

**Input**: raw mmWave reflection signals $S$.
**Output**: generated Elevation-Azimuth Heatmap *EAMap*.
Apply Range FFT and Doppler FFT on $S$ to get *Range_Doppler*.
Use CFAR to detect peaks on Range dimension and find overlapped *Detections* with Doppler dimension.
Apply Angle FFT on *Detections* to detect objects and calculate angles in Azimuth and Elevation.

Fetch *points* and *SNR* from previous results.
Apply offset to calibrate *points*.
Initialize Elevation-Azimuth Heatmap *EAMap*.
**for** every point $p$ **in** *points* **do**
  Find closest position *pos* for $p$ in *EAMap*.
  Apply Gaussian filter on corresponding *SNR* to calculate the energy distribution *value*.
  Map *value* in *pos* with predefined window size.
**end for**
**return** *EAMap*

**ALGORITHM 3:** The Process of Producing Targets from Ground Truth Bounding Boxes

**Input**: ground truth bounding boxes $GT$; 3x3 predefined anchors $A$; threshold $\tau$.
**Output**: generated 3-level target from ground truth $T$.
Initialize $T$ with the same shape of prediction.
**for** every level target $t$ **in** $T$ **do**
  **for** each bounding box $gt$ **in** $GT$ **do**
    Get center point $(x, y)$, height and width $(h, w)$, and depth value $d$ from $gt$.
    Fetch 3 anchors $a$ corresponding to the level and scale it.
    Calculate the ratios $R$ of $(h, w)$ and each anchor of $a$.
    **for** each ratio $r$ **in** $R$ **do**
      **if** $r < \tau$ **then**
        Find nearest positions by $(x, y)$ in $t$ and save scaled $gt$ into these positions of this anchor.
      **end if**
    **end for**
  **end for**
**end for**
**return** $T$

## References

[1] Emre Kuşkapan, Mohammad Ali Sahraei, Merve Kayaci Çodur, and Muhammed Yasin Çodur. 2022. Pedestrian safety at signalized intersections: Spatial and machine learning approaches. *Journal of Transport & Health* 24 (2022).
[2] Dania Ammar, Yueru Xu, Bochen Jia, and Shan Bao. 2022. Examination of recent pedestrian safety patterns at intersections through crash data analysis. *Transportation Research Record* 2676, 12 (2022).
[3] Ali Karami, Hossein Karami, Amir Reza Mamdoohi, and Mahdieh Allahviranloo. 2022. Pedestrian safety at urban intersections: Lighting conditions is the question. *International Journal of Crashworthiness* (2022).

[4] Ahadul Islam, Michelle Mekker, and Patrick A. Singleton. 2022. Examining pedestrian crash frequency, severity, and safety in numbers using pedestrian exposure from Utah traffic signal data. *Journal of Transportation Engineering, Part A: Systems* 148, 10 (2022).
[5] Forouzan Rezapur-Shahkolai, Maryam Afshari, Amin Doosti-Irani, Saeed Bashirian, and Shahnaz Maleki. 2022. Interventions to prevent road traffic injuries among pedestrians: A systematic review. *International Journal of Injury Control and Safety Promotion* 29, 4 (2022).
[6] Tong Zhao, Ekim Yurtsever, Joel A. Paulson, and Giorgio Rizzoni. 2022. Formal certification methods for automated vehicle safety assessment. *IEEE Transactions on Intelligent Vehicles* 8, 1 (2022).
[7] Zoltan Ferenc Magosi, Hexuan Li, Philipp Rosenberger, Li Wan, and Arno Eichberger. 2022. A survey on modelling of automotive radar sensors for virtual test and validation of automated driving. *Sensors* 22, 15 (2022).
[8] Honghong Chen, Xinyu Han, Zhanjun Hao, Hao Yan, and Jie Yang. 2023. Non-contact monitoring of fatigue driving using FMCW millimeter wave radar. *ACM Transactions on Internet of Things* 5, 1 (2023).
[9] Zhu Juncen, Jiannong Cao, Yanni Yang, Wei Ren, and Huizi Han. 2023. mmDrive: Fine-grained fatigue driving detection using mmWave radar. *ACM Transactions on Internet of Things* 4, 4 (2023).
[10] Xiaoqi Zhai, Helai Huang, N. N. Sze, Ziqi Song, and Kai Kwong Hon. 2019. Diagnostic analysis of the effects of weather condition on pedestrian crash severity. *Accident Analysis & Prevention* 122 (2019).
[11] Duo Li, Prakash Ranjitkar, Yifei Zhao, Hui Yi, and Soroush Rashidi. 2017. Analyzing pedestrian crash injury severity under different weather conditions. *Traffic Injury Prevention* 18 (2017).
[12] Paulius Tumas, Adam Nowosielski, and Arturas Serackis. 2020. Pedestrian detection in severe weather conditions. *IEEE Access* 8 (2020).
[13] Willy Carlos Tchuitcheu, Christophe Bobda, and Md Jubaer Hossain Pantho. 2020. Internet of smart-cameras for traffic lights optimization in smart cities. *Internet of Things* 11 (2020).
[14] Qiuyu Jiang, Lijun Zhang, and Dejian Meng. 2019. Target detection algorithm based on MMW radar and camera fusion. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC'19)*.
[15] Matti Kutila, Pasi Pyykönen, Hanno Holzhüter, Michele Colomb, and Pierre Duthon. 2018. Automotive LiDAR performance verification in fog and rain. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC'18)*.
[16] I. J. Dowman. 2004. Integration of LIDAR and IFSAR for mapping. *International Archives of Photogrammetry and Remote Sensing* 35 (2004).
[17] Aida Vera Lopez, Andrey Chervyakov, Greg Chance, Sumit Verma, and Yang Tang. 2019. Opportunities and challenges of mmWave NR. *IEEE Wireless Communications* 26 (2019).
[18] Ying He, Yan Chen, Yang Hu, and Bing Zeng. 2020. WiFi vision: Sensing, recognition, and detection with commodity MIMO-OFDM WiFi. *IEEE Internet of Things Journal* 7, 9 (2020).
[19] Bao Zhou, Zuona Chen, Ziyuan Gong, and Rui Zhou. 2020. Detection of suspicious objects concealed by walking pedestrians using WiFi. In *2020 IEEE Wireless Communications and Networking Conference (WCNC'20)*.
[20] Jawad N. Yasin, Sherif A. S. Mohamed, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. 2021. Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. *International Journal of Information Technology* 13 (2021).
[21] Jihwan Youn, Martin Lind Ommen, Matthias Bo Stuart, Erik Vilain Thomsen, Niels Bent Larsen, and Jørgen Arendt Jensen. 2020. Detection and localization of ultrasound scatterers using convolutional neural networks. *IEEE Transactions on Medical Imaging* 39, 12 (2020).
[22] Sandeep Rao. 2017. Introduction to mmWave sensing: FMCW radars. *Texas Instruments (TI) mmWave Training Series* (2017).
[23] Yao Yu, Athina P. Petropulu, and H. Vincent Poor. 2009. Compressive sensing for MIMO radar. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*.
[24] Sohrab Madani, Jayden Guan, Waleed Ahmed, Saurabh Gupta, and Haitham Hassanieh. 2022. Radatron: Accurate detection using multi-resolution cascaded MIMO radar. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*.
[25] Yizhou Wang, Zhongyu Jiang, Yudong Li, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. 2021. RODNet: A real-time radar object detection network cross-supervised by camera-radar fused object 3D localization. *IEEE Journal of Selected Topics in Signal Processing* 15, 4 (2021).
[26] Keegan Garcia, Mingjian Yan, and Alek Purkovic. 2018. Robust traffic and intersection monitoring using millimeter wave sensors. *Texas Instruments* (2018).
[27] Bo Ju, Wei Yang, Jinrang Jia, Xiaoqing Ye, Qu Chen, Xiao Tan, Hao Sun, Yifeng Shi, and Errui Ding. 2021. DANet: Dimension apart network for radar object detection. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*.
[28] Guanhao Yang, Wei Feng, Jintao Jin, Qujiang Lei, Xiuhao Li, Guangchao Gui, and Weijun Wang. 2020. Face mask recognition system with YOLOV5 based on image recognition. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC'20)*.

[29] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*.
[30] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[31] Xian Shuai, Yulin Shen, Yi Tang, Shuyao Shi, Luping Ji, and Guoliang Xing. 2021. milliEye: A lightweight mmWave radar and camera fusion system for robust object detection. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*.
[32] Pingping Cai and Sanjib Sur. 2023. MilliPCD: Beyond traditional vision indoor point cloud generation via handheld millimeter-wave devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2023).
[33] Lydia Chioukh, Halim Boutayeb, Dominic Deslandes, and Ke Wu. 2014. Noise and sensitivity of harmonic radar architecture for remote sensing and detection of vital signs. *IEEE Transactions on Microwave Theory and Techniques* 62, 9 (2014).
[34] Junfeng Guan, Sohrab Madani, Suraj Jog, Saurabh Gupta, and Haitham Hassanieh. 2020. Through fog high-resolution imaging using millimeter wave radar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[35] Aakriti Adhikari and Sanjib Sur. 2023. Argosleep: Monitoring sleep posture from commodity millimeter-wave devices. In *Proceedings IEEE INFOCOM*.
[36] Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. 2021. mmMesh: Towards 3D real-time dynamic human mesh construction using millimeter-wave. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*.
[37] Texas Instrument. 2023. MMWCAS-RF-EVM. (2023). Retrieved from https://www.ti.com/tool/MMWCAS-RF-EVM
[38] Stereo Labs. 2023. ZED 2. (2023). Retrieved from https://www.stereolabs.com/zed-2/
[39] Vijay John and Seiichi Mita. 2019. RVNet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments. In *Image and Video Technology: 9th Pacific-Rim Symposium, PSIVT 2019, Sydney, NSW, Australia, November 18–22, 2019, Proceedings 9*.
[40] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).
[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[42] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[43] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. 2020. Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
[44] Yi-Fan Zhang, Weiqiang Ren, Zhang Zhang, Zhen Jia, Liang Wang, and Tieniu Tan. 2022. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* 506 (2022).
[45] Texas Instruments. 2023. MMWCAS-DSP-EVM. (2023). Retrieved from https://www.ti.com/tool/MMWCAS-DSP-EVM
[46] Texas Instruments. 2023. AWR2243 Dataset. (2023). Retrieved from https://www.alldatasheet.com/datasheet-pdf/pdf/1245311/TI/AWR2243.html
[47] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004).
[48] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[49] Mark Everingham and John Winn. 2012. The PASCAL Visual Object Classes challenge 2012 (VOC2012) development kit. *Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep* 2007, 1-45 (2012).
[50] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. ([n. d.]). Retrieved from http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
[51] Felix Nobis, Maximilian Geisslinger, Markus Weber, Johannes Betz, and Markus Lienkamp. 2019. A deep learning-based radar and camera sensor fusion architecture for object detection. In *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF'19)*.
[52] Vijay John, . K. Nithilan, Seiichi Mita, Hossein Tehrani, R. S. Sudheesh, and P. P. Lalu. 2020. SO-Net: Joint semantic segmentation and obstacle detection using deep fusion of monocular camera and radar. In *Image and Video Technology: PSIVT 2019 International Workshops, Sydney, NSW, Australia, November 18–22, 2019, Revised Selected Papers 9*.
[53] Weiyan Chen, Fusang Zhang, Tao Gu, Kexing Zhou, Zixuan Huo, and Daqing Zhang. 2021. Constructing floor plan through smoke using ultra wideband radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (2021).

[54] Kun Qian, Zhaoyuan He, and Xinyu Zhang. 2020. 3D point cloud generation with millimeter-wave radar. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020).
[55] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[56] Yuen Peng Loh and Chee Seng Chan. 2019. Getting to know low-light images with the exclusively dark dataset. *Computer Vision and Image Understanding* 178 (2019).
[57] Waleed Ali, Sherif Abdelkarim, Mahmoud Zidan, Mohamed Zahran, and Ahmad El Sallab. 2018. YOLO3D: End-to-end real-time 3D oriented object bounding box detection from LiDAR point cloud. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.
[58] Dinh Van Nam and Kim Gon-Woo. 2021. Solid-state LiDAR based-SLAM: A concise review and application. In *2021 IEEE International Conference on Big Data and Smart Computing (BigComp'21)*.
[59] Andreas Danzer, Thomas Griebel, Martin Bach, and Klaus Dietmayer. 2019. 2D car detection in radar data with PointNets. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC'19)*.
[60] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. 2020. RadarNet: Exploiting radar for robust perception of dynamic objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*.
[61] Meet Shah, Zhiling Huang, Ankit Laddha, Matthew Langford, Blake Barber, Sidney Zhang, Carlos Vallespi-Gonzalez, and Raquel Urtasun. 2020. LiRaNet: End-to-end trajectory prediction using spatio-temporal radar fusion. *arXiv preprint arXiv:2010.00731* (2020).
[62] Yingjie Wang, Jiajun Deng, Yao Li, Jinshui Hu, Cong Liu, Yu Zhang, Jianmin Ji, Wanli Ouyang, and Yanyong Zhang. 2023. Bi-LRFusion: Bi-directional LiDAR-radar fusion for 3D dynamic object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[63] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. 2021. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[64] Yu-Jhe Li, Jinhyung Park, Matthew O'Toole, and Kris Kitani. 2022. Modality-agnostic learning for radar-lidar fusion in vehicle detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[65] Sai Deepika Regani, Chenshu Wu, Beibei Wang, Min Wu, and K. J. Ray Liu. 2021. mmWrite: Passive handwriting tracking using a single millimeter-wave radio. *IEEE Internet of Things Journal* 8 (2021).
[66] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

Received 30 January 2024; revised 22 July 2024; accepted 17 August 2024