CoSense: Deep Learning Augmented Sensing for Coexistence with Networking in Millimeter-Wave Picocells

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

We present *CoSense*, a system that enables coexistence of networking and sensing on next-generation millimeter-wave (mmWave) picocells for traffic monitoring and pedestrian safety at intersections in all weather conditions. Although existing wireless signal-based object detection systems are available, they suffer from limited resolution and their outputs may not provide sufficient discriminatory information in complex scenes, such as traffic intersections. *CoSense* proposes using 5G picocells, which operate at mmWave frequency bands and provide higher data rates and higher sensing resolution than traditional wireless technology. However, it is difficult to run sensing applications and data transfer simultaneously on mmWave devices due to potential interference, and using special-purpose sensing hardware can prohibit deployment of sensing applications to a large number of existing and future inexpensive mmWave devices. Additionally, mmWave devices are vulnerable to weak reflectivity and specularity challenges, which may result in loss of information about objects and pedestrians. To overcome these challenges, *CoSense* design customized deep learning models that not only can recover missing information about the target scene but also enable coexistence of networking and sensing. We evaluate *CoSense* on diverse data samples captured at traffic intersections and demonstrate that it can detect and locate pedestrians and vehicles, both qualitatively and quantitatively, without significantly affecting the networking throughput.

CCS Concepts: • **Human-centered computing** → **Ubiquitous computing**; • **Networks** → **Network simulations;**

Additional Key Words and Phrases: Millimeter-wave, picocells, conditional generative adversarial networks, convolutional neural network, joint networking and sensing

ACM Reference Format:
Hem Regmi and Sanjib Sur. 2024. CoSense: Deep Learning Augmented Sensing for Coexistence with Networking in Millimeter-Wave Picocells. *ACM Trans. Internet Things* 5, 3, Article 17 (August 2024), 35 pages. https://doi.org/10.1145/3670415

1 Introduction

In 2021, a staggering 7,500 pedestrian fatalities were reported in the United States as a result of vehicular collisions [1]. According to the US Department of Transportation, over 50% of fatal or injurious road accidents occur at or in close proximity to traffic intersections [2]. Most, if not

This work is partially supported by the National Science Foundation (grant nos. CNS-1910853, CAREER-2144505, NeTS-2342833, and MRI-2018966).
Authors' Contact Information: Hem Regmi, Storey Innovation Center, Rm 1207 550 Assembly Street, Columbia, SC 29201, United States; e-mail: hregmi@email.sc.edu; Sanjib Sur, Storey Innovation Center, Rm 1207 550 Assembly Street, Columbia, SC 29201, United States; e-mail: sur@cse.sc.edu.

ACM 2577-6207/2024/08-ART17
https://doi.org/10.1145/3670415

all, of these deaths and injuries can be prevented by proactively warning the drivers, vehicles, and pedestrians [3], for example, by notifying the pedestrian of oncoming vehicles at a cross walk or by enabling smarter speed control for vehicles near traffic interactions. While the advent of full driving automation (i.e., Level 5 autonomy [4]) holds promise for a future without such tragedies, there is a pressing need for an interim solution at intersections to reduce the frequency of these incidents. Such a system can also collect important statistics and telemetry information, such as real-time pedestrian and vehicular traffic at intersections, their speeds, vehicle proximity to intersection stop bars, occupied lanes, and vehicle types, which can enable a variety of applications related to traffic monitoring and management. Existing vision-based sensors, such as cameras and LiDARs, provide powerful tools to not only measure such traffic behavior at intersections but also improve pedestrian safety. However, the performance of the vision-based sensors are often significantly impaired by the scene conditions, such as no ambient lights or poor visibility during nighttime, heavy rain, or dense fog.

Wireless signal–based object detection systems can alleviate such a problem. A wireless device can illuminate the target scene by transmitting wireless signals and receiving them bouncing off of different objects. Based on the time-of-flight and angle of reflections, this device can map the entire environment and "see" the static and dynamic objects within it, even under low visibility and poor weather conditions. Traditionally, these systems rely on Wi-Fi/LTE devices or special-purpose radars to transmit and receive low-frequency signals and capture information about objects and activities [5–8]. However, the information provided by these systems is limited in resolution due to the long wavelength and narrow bandwidth operations. The outputs from these systems may also lack meaningful discriminatory information on par with the vision-based systems, e.g., RGB or depth cameras [9–11]. This is particularly true in complex scenes, such as traffic intersections, where the outputs from these low-frequency wireless systems may not provide sufficient discriminatory information to distinguish objects, humans, and their characteristics, such as location, walking or driving direction, and speed.

Fortunately, next-generation wireless networking devices operating at higher frequency, such as 5G picocells [12], offers a solution to this issue. These networking devices have built-in **millimeter-wave (mmWave)** technology, which offers a substantially higher data rate than traditional wireless technology and can host multiple, palm-sized antenna arrays to create hundreds of beams for serving mobile users. Due to the short wavelength and wide bandwidth operation of mmWave signals, each picocell can also function as a high-precision environment sensor. With a wider contiguous bandwidth and multiple antennas, mmWave sensing can detect objects in harsh weather conditions with more detail, such as their shape and bounding box. Thus, these devices can be incorporated into roadside infrastructures, particularly at traffic intersections, to provide high-resolution monitoring of vehicles and pedestrians. Existing research works have also demonstrated the potential of mmWave for a range of applications, such as identifying human postures for exercise monitoring [13, 14], detecting small objects [15, 16], sensing soil characteristics [17, 18], and detecting vehicle occupancy [19, 20]. Furthermore, the shorter wavelength and wider bandwidth of mmWave signals, compared with traditional Wi-Fi or LTE signals, theoretically allows for higher-resolution capture of the target scene. Additionally, mmWave devices provide an advantage over camera-based systems during poor weather and low visibility conditions, as wireless signals can penetrate through some obstructions such as dense fog, while lights cannot. Thus, *the ubiquity of mmWave technology in 5G-and-beyond devices, such as the picocells in roadside infrastructure, enables the opportunity to bring traffic monitoring and pedestrian safety at intersections in all weather conditions.* However, the design of mmWave sensing on networking devices presents two challenges.

First, although mmWave devices are good environmental sensors, it is difficult to simultaneously run sensing applications and data transfer. For instance, if a pedestrian walks in front of an mmWave picocell while it is streaming data, it can disrupt the **Line-of-Sight (LOS)** communication path. While its beam can be steered towards the **Non-Line-of-Sight (NLOS)** path or networking and sensing operations can be time-multiplexed to reduce interference, these can negatively impact both pedestrian detection accuracy and network performance by reducing throughput, increasing latency, and disrupting the delivery of packets to critical applications. A strawman approach for networking-sensing coexistence is to augment devices with special-purpose sensing hardware to use different parts of the mmWave spectrum and avoid interference. However, this will prohibit deployment of the sensing applications to a large number of existing and future inexpensive mmWave devices.

Second, mmWave devices are vulnerable to more specular and variable reflectivity challenges (compared with Wi-Fi or LTE) due to their high-frequency operations. Thus, depending on the location, orientation, and absorption properties of objects and pedestrians on the road, the signals transmitted may not reach back to the device [21–23]. This can result in a loss of information about objects and pedestrians as well as difficulties in accurately capturing their properties.

To address these challenges, *we present CoSense*, *which seamlessly integrates networking and sensing on picocells without compromising performance in all weather conditions.* Existing approaches have used mmWave signals for pedestrian and vehicle detection at traffic intersections using dedicated sensing hardware, such as radar, with high data acquisition rates [24, 25]. However, as *CoSense* designs sensing applications on networking devices, high acquisition is infeasible due to the need to share time between networking and sensing. Thus, we find opportunistic idle times within the data transfer for sensing, allowing only partial temporal observation. Additionally, each sensing sample only allows partial spatial observation due to specularity and weak reflectivity. To this end, *CoSense* designs deep learning augmented models to recover the missing information in space and time.

The key idea is to learn the representation of mmWave reflections to the pedestrian and object properties from visual data in clear weather conditions by identifying patterns from several examples. However, instead of trying to learn thousands of pixels in high-resolution visual images from only a few space and time samples in the mmWave reflected signals, which could lead to a network divergence during learning, *CoSense* divides the learning task into two networks. *First*, it designs customized **conditional Generative Adversarial Networks (cGAN)** for object detection model [26] to learn the pedestrians and objects, assuming that the sensing samples are continuously available. *Then*, it augments the residual networks with a custom-designed deconvolution layer that identifies the missing sensing samples in time with dynamic frame prediction from past frame and velocity information and recovers information about the scene. To assist the deep learning model in comprehending visual data and distinguishing static and dynamic scenes, *CoSense* produces image-like outputs from the mmWave reflected signals by generating heatmaps for static and dynamic objects (see Figure 1). The *CoSense* output includes depth images of pedestrians and vehicles, which contain bounding boxes of objects and their average depth from the device, enabling precise location of objects in the environment. Furthermore, consecutive frames of depth images enable the estimation of object velocities, an essential metric for anticipating and avoiding pedestrian collisions.

Due to the lack of open-source mmWave datasets at the traffic intersection, we implement and evaluate *CoSense* by collecting datasets from a custom-designed experimental setup. The setup consists of a **Commercial-Off-The-Shelf (COTS)** mmWave cascade device [27, 28] and a ZED

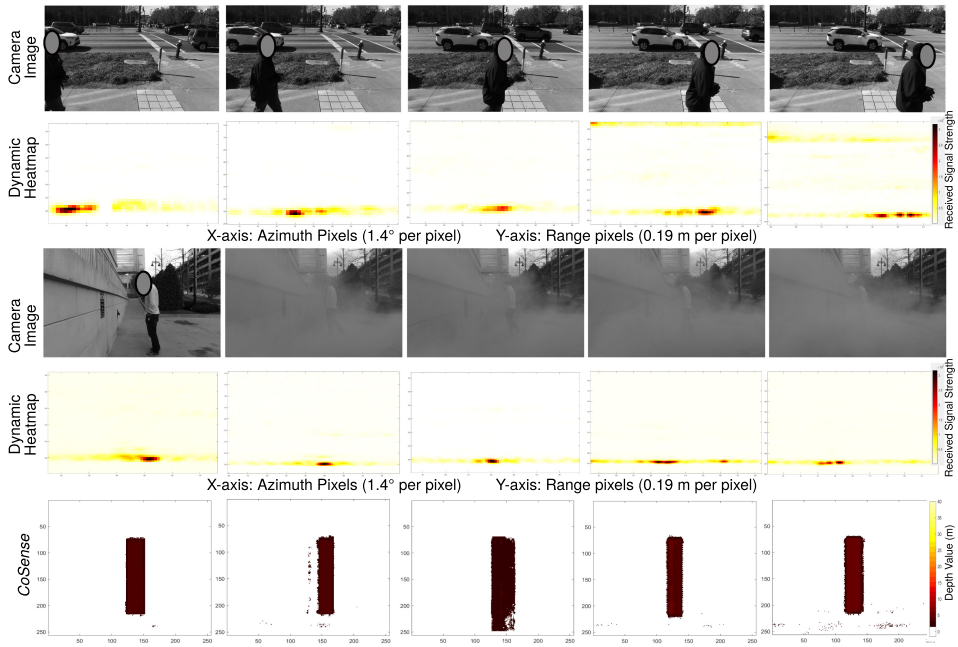

Fig. 1. *CoSense* predicts the bounding box of pedestrians irrespective of the environmental conditions.

stereo camera [29] to collect the mmWave and ground-truth data samples. As there is currently no open-source mmWave platform that supports 5G communication, we simulate a Ray-Tracing-based 5G communication protocol [30] with the same hardware parameters as our experimental setup to evaluate the joint networking and sensing tasks. For training, benchmarking, and testing the design, we have collected ∼1.67 TB of data samples over a 6-month period. To evaluate the performance of *CoSense*, we use well-established metrics such as the **Intersection-over-Union (IoU)** [31], **Multi-Scale Structural Similarity Index Measure (MS-SSIM)** [32], and pixel-to-pixel errors [33] to identify pedestrians and vehicles. Our results show that *CoSense* can identify pedestrians and vehicles with a median IoU of 0.55 and 0.63 with regard to the ground truth visual images for pedestrian and vehicle detection, respectively, indicating a good match. Additionally, the mean depth error for both pedestrians and vehicles is less than 0.66 m on 90^{th} percentile data. Our context-aware model reduces sensing overhead by 70% while maintaining a good detection performance, with only a 27% drop in median IoU, a 3.5% drop in MS-SSIM, and only a ∼15% drop in the data throughput. The mean depth error for 90^{th} percentile data increases from 0.16 m to 0.35 m, which is tolerable for outdoor applications. Finally, our system demonstrates accurate predictions of pedestrian bounding boxes and mean depth in foggy weather conditions, indicating that *CoSense* is effective in challenging environments such as traffic intersections with poor visibility.

In summary, we have the following contributions: (1) We propose a custom cGAN-based object detection network to detect and locate pedestrians and vehicles, and evaluate it with experimental datasets from a COTS mmWave device. (2) We recover the unobserved heatmaps from previous heatmaps using a residual network to enable joint networking and sensing with a single mmWave device. To accelerate the research on joint networking and sensing at mmWave, we will open-source our measured dataset, 5G simulator, and deep learning codebase.

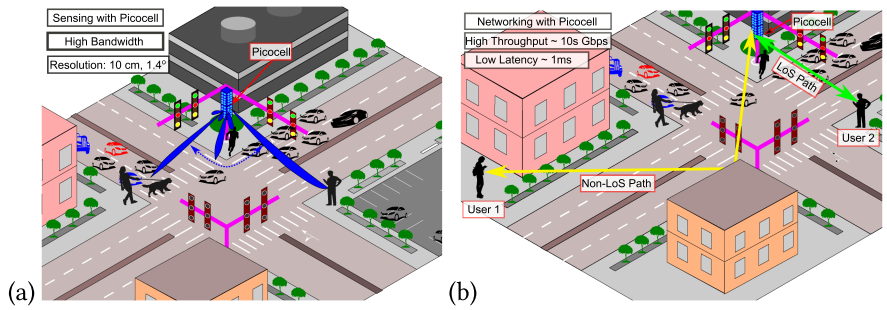

Fig. 2. (a) An illustration of sensing with mmWave picocell at a traffic intersection. (b) Picocell communication to the user through Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) paths.

2 Background & Fundamentals

2.1 Millimeter-Wave Picocells, Networking, and Sensing

2.1.1 Picocell Fundamentals. Picocell technology is a next-generation cellular network solution that offers wireless coverage in a small geographic area, such as a residential or commercial space. Unlike traditional macrocell networks, picocell networks are characterized by their small size, fast deployment, and lower infrastructure costs. They are widely used today to provide 5G/mmWave connectivity in challenging environments such as areas with high-rise buildings or rural locations, as well as to enhance the existing macrocell network capacity in areas experiencing high user density. Picocell devices use electronically steerable beams and communicate on very high frequency and wide bandwidth to achieve substantially higher data rates than traditional wireless networks at a short distance. Figure 2 shows a picocell installed at a traffic intersection on a pole [34–36].

2.1.2 5G/Millimeter-Wave Networking. With the capability to transmit data at 10 s of **gigabits per second (Gbps)**, 5G networks promise to support next-generation applications in streaming; virtual, augmented, and extended reality; telepresence; and network slicing that enables providers to divide the network into multiple virtual stacks with distinct service requirements. 5G **New Radio (NR)** represents a significant improvement over 4G/LTE, enabling enhanced mobile broadband with high data rates, supporting a large number of **Internet-of-Things (IoT)** devices with variable bandwidth needs, and providing ultra-reliable and low-latency communication (up to 1 ms) [37]. NR devices leverage mmWave as its core technology, with a center frequency of up to 90 GHz and a combined spectrum over 10 GHz, and use beam-forming and spatial multiplexing techniques to achieve high data rates for multiple users [37, 38]. In addition, they use high **subcarrier spacing (SCS)** up to 240 KHz and can configure multiple bandwidth configurations within a single device [37, 39]. Typically, 5G frames are 10 ms long, with 10 subframes and with 15 KHz SCS. Each slot length is 1 ms, supporting up to 14 OFDM symbols [39]. Increasing the SCS to 240 KHz reduces the slot duration to 0.0625 ms, which improves latency and increases transmission efficiency [37].

2.1.3 Millimeter-Wave Sensing. Recently, mmWave wireless sensing technology has been gaining widespread use across various domains, including security and surveillance, industry, and automotive [40–43]. For example, in security and surveillance applications, mmWave sensing can be used for high-resolution imaging of hidden, contra-band, and hazardous materials; target tracking; situational awareness; detecting intruders; and tracking people and vehicles. In industrial applications, it can be used to detect objects and their properties non-destructively, measure temperature,

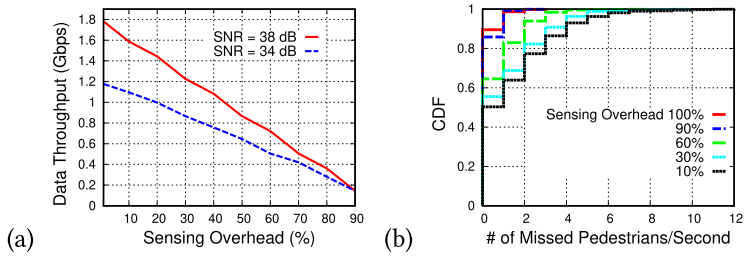

Fig. 3. (a) Effect on data throughput to the user with different sensing overheads. (b) Number of pedestrians missed per second with joint networking and sensing tasks.

and detect motion. In automotive applications, it can be used for detecting pedestrians, controlling cruising speed, providing collision warnings, and assisting in parking. The main motivation for using mmWave sensing for object detection is that it works in harsh weather conditions and provides more detail than traditional radars, which provide only the distance and velocity of an object without its category and semantic features. The shorter wavelength and wider bandwidth of mmWave signals allow a better perception of the environment than the traditional low-frequency signals (used by existing Wi-Fi or LTE devices). Such perception also works under low-light or harsh weather conditions, since the wireless signal is (almost) unaffected by such environmental conditions at a short distance. Since 5G picocells, installed at traffic light poles, are equipped with such mmWave technologies, they can be repurposed for sensing applications. The use of a large number of transmitter and receiver antennas and GHz-wide bandwidth in 5G picocells can also provide better angle and depth resolutions and more accurate perception of objects in the target scene than traditional Wi-Fi or LTE. However, it should be noted that not all reflected signals may reach the mmWave receiver due to specularity and weak reflectivity, leading to the loss of some information about the scene.

2.2 Challenges in Joint Networking and Sensing

The integration of networking and sensing capabilities within a single mmWave picocell poses significant challenges due to the potential for one function to adversely affect the performance of the other. One possible approach to addressing this challenge is through spectrum sharing, whereby a portion of the available spectrum is dedicated to networking and another portion is dedicated to sensing. However, this approach can lead to reduced effective bandwidth for both tasks, which in turn can negatively impact throughput performance and sensing resolution. Alternatively, spatial multiplexing can be used to enable separate transceiver pairs to direct their beams toward data users and the environment for networking and sensing, respectively. However, this may result in a wider beam and reduced power due to the direct correlation between transmitted power and the number of transceiver pairs used for beamforming. Although time-sharing could be a potential solution whereby dedicated time slots are used for networking and sensing, allowing for full bandwidth and transmitted power to be leveraged for each task, it can still impact the performance of both networking and sensing.

Impact on Networking: To understand the impact of allocating dedicated time slots for mmWave sensing on network performance, we conducted simulations of a 5G network operating at a traffic intersection (following [30]), in which we employed a time-sharing approach to allocate dedicated time slots for networking and sensing. Figure 3(a) demonstrates a linear decrease in throughput to the users with increasing sensing overheads for two distinct signal-to-noise ratios. Thus, minimizing the sensing overhead is crucial to maximizing the data throughput for the users.

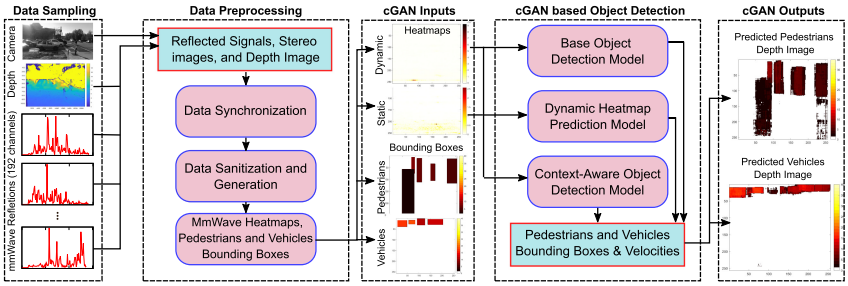

Fig. 4. System overview of *CoSense*.

Impact on Sensing: Reducing the sensing duration can result in significant inaccuracies in traffic monitoring, such as an increased number of pedestrians missed by the device. Figure 3(b) shows that reducing the sensing frequency can lead to a greater average number of pedestrians missed per second. For instance, when sensing frequency is reduced to 30% of the time, more than three pedestrians are missed on 20% of the occasions. As traffic intersections tend to experience high pedestrian activity, it is critical to detect all pedestrians accurately to ensure their safety. The absence of data samples at all timestamps mostly impacts the precision of object detection and emphasizes the importance of balancing the networking and sensing requirements.

3 *CoSense* Design

3.1 System Overview

The *CoSense* system provides a solution to the aforementioned challenges by enabling an existing mmWave device to perform joint communication and sensing tasks. *CoSense* follows the time-sharing approach for the coexistence of networking and sensing. This system can be deployed in various picocells to enhance pedestrian safety at traffic intersections. To train the model, the *CoSense* system uses an mmWave cascade device and a ZED stereo camera pair to collect a large dataset of samples from traffic intersections. The mmWave samples are processed using **one-dimensional (1D) Fast Fourier Transform (FFT)** to produce dynamic and static heatmaps, while an object detection algorithm [44] is applied to the stereo images to obtain the ground truth bounding box locations of pedestrians and vehicles. The heatmaps and depth images are synchronized and fed into a deep learning object detection model built using multiple convolution layers based on cGANs [15, 23]. The model is trained using thousands of samples to establish a mapping between the heatmaps and ground truth data. Once trained, the model can be deployed in mmWave communication devices, where it takes mmWave wireless signals reflected from the environment as inputs and generates bounding boxes and velocities of pedestrians and vehicles as outputs. Figure 4 provides an overview of the system.

As there is currently no open-source mmWave platform that supports 5G communication, we developed a Ray-Tracing-based simulator with the same hardware parameters used for pedestrian detection to design and evaluate joint communication and sensing tasks. To this end, we used the Ray-Tracing and terrain buildings from an open street map [45] to estimate the mmWave channel of the environment and then modified the physical and MAC layers' parameters and "slot sequence" to control data transmission. To evaluate data throughput, we sent a large amount of data from the picocell to randomly placed users, decoded the received signal, and recorded the "slot sequence." During the "slots" when sensing data was unavailable, we recovered the heatmaps through a dynamic heatmap prediction network and predicted pedestrians and vehicles using the context-aware object detection model. In the following sections, we elaborate on our design component in detail.

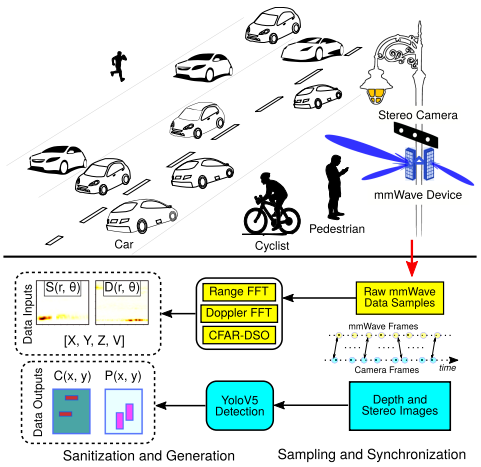

Fig. 5. *CoSense*'s data preprocessing framework.

3.2 Data Preprocessing

Prior to designing the deep learning model and learning only the necessary features from mmWave reflections and ground truth data, *CoSense* first preprocesses the datasets to eliminate spurious information. Our custom-built data collection platform collects and saves the data on a host PC with their corresponding timestamps. The platform uses separate devices to collect the ground truth samples and mmWave reflection samples, requiring tight synchronization, noise filtering, and pruning to extract meaningful information for the deep learning model. The vision sensor used for ground truth images has a smaller **field of view (FoV)** than the mmWave device, necessitating the pruning of mmWave reflections to match the ground truth FoV. To this end, the data preprocessing consists of two steps: (1) sampling and synchronizing the mmWave data with ground truth vision data and (2) sanitizing the datasets to remove noise and generating input-output data pairs.

3.2.1 Sampling and Synchronization. We acquire 10 data frames per second to obtain both the mmWave samples and stereo images, with samples collected every 100 ms due to limitations of the camera hardware. Despite our efforts to trigger both devices almost simultaneously from the same host PC, there is still tens of milliseconds of hardware latency that impedes a tight synchronization. To address this issue, we store the real-time timestamps of the host PC at the moment each device records the data sample and match the mmWave samples with images by examining their timestamps and interpolating any missing samples. If we observe a significant discrepancy between the timestamps of the first frames captured by the mmWave device and camera, we discard the entire batch of data samples, as they do not represent the same time instance and could negatively impact the learning network. Figure 5 illustrates the data acquisition, sampling, and synchronization processes, in which a co-located mmWave device and stereo camera capture the same region of the environment.

3.2.2 Sanitization and Generation. After synchronizing the mmWave samples and stereo images to ensure that they correspond to the same timestamp, we apply a series of FFTs to the raw and unprocessed mmWave reflections to generate the range-azimuth heatmaps of dynamic and static objects, represented as $D(r, \theta)$ and $S(r, \theta)$, respectively (see Figure 5). The heatmap representations are preferable since it is relatively easier to learn correlations between image-like ground truth output from image-like inputs [46, 47]. A static heatmap captures the details about the stationary objects in the scene, such as traffic poles, fire hydrants, and parked cars. In contrast, dynamic

heatmaps capture the objects that are in motion. Separating the static and dynamic heatmaps allows the model to learn the association of the static and dynamic objects to the ground truth data separately. Having separate static and dynamic heatmaps helps to make the *CoSense* robust in different traffic scenarios. Since mmWave reflections are mostly unaffected by harsh weather conditions, such as low light [48], fog [23], and snow [49], except slight attenuation [50], the mmWave heatmaps are also immune to such harsh weather conditions. Furthermore, attenuation happens for object and background noises without significantly changing the heatmap's nature. Still, the raw pixel values from the stereo and depth images cannot be directly used as ground truth for object classification and localization because they lack discriminating information about the object. To this end, we employ a popular open-source object detection algorithm, YOLOv5 [44], to predict the locations of objects such as pedestrians, cars, bicycles, and buses, that may appear in the traffic intersection from the stereo images and depth images. "You Only Look Once (YOLO)" is a famous object detection algorithm that provides the accurate **two–dimensional (2D)** bounding box of objects in clear weather [44]. It comprises 80 object classes, including Pedestrians, Cars, Buses, and Cyclists. In *CoSense*, we focus our system on detecting the 2D bounding box of objects related to pedestrian safety. YOLOv5 is the latest model released and provides accurate 2D bounding box results. We acknowledge that the 2D bounding box from YOLOv5 will be less accurate than manual labeling of the 2D bounding box. However, our empirical result shows that it can detect up to 23 objects in a single image. Furthermore, YOLOv5 may fail to detect vehicles that are far away and appear as small blobs, but the reflected signals from far-away objects to the mmWave devices are also weak and close to the noise floor. Hence, considering the high accuracy of YOLOv5 and ability to streamline the ground truth generation process, we select the YOLOv5 model for generating a ground-truth 2D bounding box.

To prepare the ground truth data for the deep learning model, we first separate the detected objects into two categories: pedestrians and vehicles, and generate two corresponding depth images, $P(x, y)$ and $C(x, y)$, respectively. Since pedestrians are comparable in size and reflective surface areas, we keep all pedestrians in a single-depth image, $P(x, y)$. In contrast, vehicles spread on both azimuth and elevation, and are kept in $C(x, y)$ for a scene. This approach enables the deep learning network to generate distinctive depth images for pedestrians and vehicles by allowing us to map each peak on the static and dynamic heatmaps to corresponding objects. The pedestrian depth image is set to the median depth of pedestrian-like objects, such as pedestrians, bicycles, and motorbikes, for their 2D locations and 0 for all other locations. Similarly, the vehicle depth image covers the locations of cars, buses, and trucks. We only consider objects with confidence scores of 50% or higher to ensure that our ground truth depth images are noise-free. In cases of overlapping objects, such as two pedestrians, we assign the depth value of the closest pedestrian to the overlapped region, as that is the portion of the FoV that the mmWave device sees. Finally, we truncate the dynamic and static heatmaps to match the FoV of the vision camera. In summary, our approach generates static and dynamic heatmaps, along with **three-dimensional (3D)** points with velocity, as inputs to the network, and produces a pedestrian depth image and a vehicle depth image as the outputs, providing information on the location and category of objects.

3.3 Deep Learning Augmented Object Detection with mmWave Device

3.3.1 Challenges with Existing Object Detection Methods. Object detection is a critical task for many applications, including industrial automation [51], autonomous driving [52], and monitoring and surveillance [53, 54]. Most of the applications rely on the acquisition of visual images to first extract the useful features and then use those particular features for robust object detection and segmentation. *CoSense*, however, relies on object and pedestrian detection from incomplete mmWave wireless signals from picocells. Even though the existing Mask RCNN approach [55] is

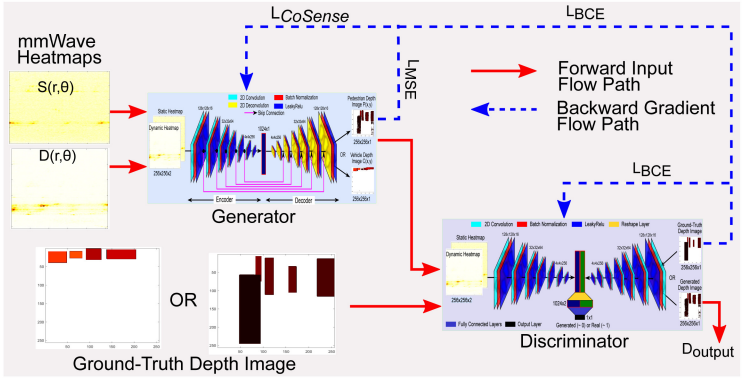

Fig. 6. *CoSense*'s cGAN network and its dataflow paths.

effective for vision images, which isolates **Regions-of-Interest (ROI)** and extracts features for classification, we cannot use the Mask RCNN network architecture directly because of the following two challenges with mmWave wireless signals: (1) the limited object details due to specularity and weak reflectivity of signals, where reflections of not all transmitted signals reach the receiver; and (2) the intermittent capture of target scene information due to joint networking and sensing. To address these issues, *CoSense* uses cGANs because these generative networks are suitable methods for generating realistic images from latent noise in the given domain [26]. The cGAN helps to recover missing regions of the mmWave heatmaps by learning from the corresponding vision images of the clear weather during the training phase. The *CoSense* cGAN network has multiple 2D convolution layers, 2D deconvolution layers, and skip connections [56] on its Generator network to produce 2D bounding box depth images from mmWave heatmaps. Next, we outline the steps involved in the base model for object detection using all data samples, dynamic heatmap recovery from missing data samples, and final object detection with recovered frames.

GAN Fundamentals: Generative networks are similar to the spirit of auto-encoder [57], which uses a few random samples to learn the data distribution during the training. After training, the generator network can generate new samples that never existed using the random noise [58]. Generative modeling is popular in synthetic data generation, in which thousands of new samples are generated from a few observations. A GAN uses two sub-models during training: (1) *Generator* **G**, which tries to generate samples close to real samples; and (2) *Discriminator* **D**, which predicts whether the data sample generated (by **G**) is real or not. Output is the probability of the sample being real, ~ 1 indicates real, and ~ 0 indicates generated. During the training, it is formulated as an adversarial game [59] until **G** completely fools **D**, which indicates that **D** now thinks of generated samples as real samples. However, providing only random noise to the generative networks can produce any output category, and output data distribution is not controlled. Therefore, we use mmWave heatmaps as a "condition" to the GAN and use the cGAN network [26] because ground truth depth images are in the same FOV and timestamp as mmWave heatmaps.

3.3.2 Base Model Object Detection with cGAN. The base model for object detection assumes that the device continuously captures information about the target scene. We will then augment the model for intermittently captured data samples. Figure 6 shows the learning framework of *CoSense*. The base model includes the Generator (**G**) and Discriminator (**D**). We create two instances of the base model with the same network architecture for pedestrians and vehicles. During training, we update the network parameters of each model instance with mmWave heatmaps and corresponding ground truth depth images of pedestrians and vehicles, respectively.

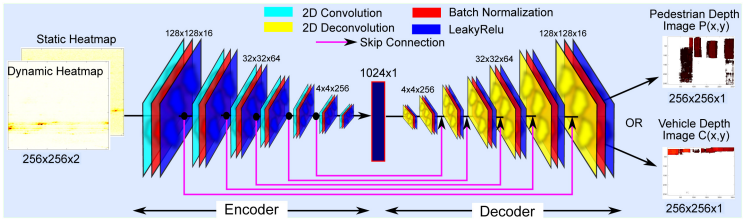

Fig. 7. *CoSense*'s Generator network architecture.

Table 1. Generator Network Parameters

	2DC1	2DC2	2DC3	2DC4	2DC5	2DC6	2DC7	2DDC1	2DDC2	2DDC3	2DDC4	2DDC5	2DDC6	2DDC7	Output
Filter #	16	32	64	128	256	512	1024	1024	512	256	128	64	32	16	
Filter Size	3×3	3×3	3×3	3×3	3×3	3×3	3×3	2×2	2×2	2×2	2×2	2×2	2×2	2×2	
Dilation	2×2	2×2	2×2	2×2	2×2	2×2	2×2	1×1	2×2	2×2	2×2	2×2	2×2	2×2	
Act. Fcn	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LRelu	Linear

2DC: 2D Convolution (with Batch Normalization); 2DDC: 2D DeConvolution (with Batch Normalization); Act. Fcn: Activation Function; LReLU: LeakyReLU Activation Function;.There are 5 skip connections between 2DC and 2DDC layers; Output layer uses linear activation.

Generator: Figure 7 show the Generator, in which we design a deep learning network with an encoder and decoder that converts the mmWave heatmaps into 2D bounding boxes with depth values. After static and dynamic heatmaps of size $256 \times 256 \times 1$ are merged in channel dimension at the input layer to create a single input of size $256 \times 256 \times 2$, multiple 2D convolution layers of **G**'s encoder extract the local and global features from mmWave heatmaps on successive layers and locate all the objects. The Encoder network uses batch normalization and LeakyReLU activation after each 2D convolution layer to make training faster and more stable. Batch normalization scales the input between 0 and 1 for the given batch of data, making it immune to the mmWave reflection attenuation. Since the nature of the heatmap input is not changed due to slight attenuation, batch normalization reduces the effect of the particular pixel value of the heatmap on cGAN output and focuses on the relationship among neighboring pixels to extract useful features. In addition, skip connections [23, 56] between successive layers of encoder and decoder preserves the details present in mmWave heatmaps and passes it to the generated depth images. Once the encoder network generates a 1D abstract feature vector, we use the decoder to convert the abstract feature vector into a 2D depth image by expanding its spatial dimension. The decoder network comprises 2D deconvolution layers with batch normalization and LeakyReLU activation, similar to an up-sampling process in which the network continuously increases its spatial dimensions until the desired output shape is reached. Table 1 provides details of the convolution layers and deconvolution layers, including the number of filters, filter size, activation on each layer, and spatial dimension dilation.

Discriminator: The Discriminator's primary goal is to guide **G** during the training process. The Generator tries to use the mmWave heatmaps and learns to generate output close to ground truth depth images. Figure 8 shows the discriminator network architecture. **D** has two encoders, *Encoder A* and *Encoder B*, to extract features from mmWave heatmaps and ground truth depth images, respectively. Both encoder network architectures are similar to the Generator's encoder network architecture. *Encoder A* converts the mmWave heatmaps of size $256 \times 256 \times 2$ to abstract a feature vector of size 1024×1, following multiple 2D convolution, batch normalization, and LeakyReLU activation. *Encoder B* similarly converts depth image of size $256 \times 256 \times 1$ to 1D a feature vector of size 1024×1. Finally, **D** combines 1D abstract features from mmWave heatmaps and ground truth depth images and reshapes them to a long 1D vector of size 2048×1, and then passes through 2

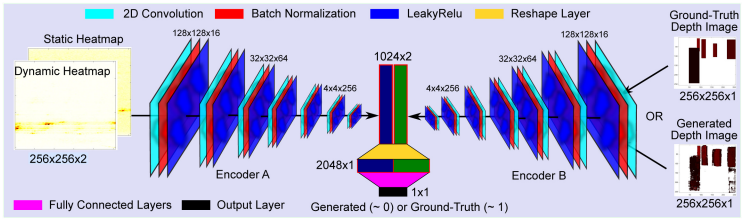

Fig. 8. *CoSense*'s Discriminator network architecture.

Table 2. Discriminator Network Parameters

	2DC1	2DC2	2DC3	2DC4	2DC5	2DC6	2DC7	FC1	FC2	Output
Filter #	16	32	64	128	256	512	1024			
Filter Size	3×3	3×3	3×3	3×3	3×3	3×3	3×3			
Dilation	2×2	2×2	2×2	2×2	2×2	2×2	2×2			
Act. Fcn	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLU	LReLUu	LReLU	Sigmoid

2DC: 2D Convolution (with Batch Normalization); Act. Fcn: Activation Function; LReLU: LeakyReLU Activation Function. Output layer uses sigmoid activation.

fully connected layers to generate the output probability of a value between 0 and 1 with a sigmoid activation function on its output layer. The output probability indicates the closeness of the input depth image to the ground truth depth image. We end training when **D** continuously outputs a probability close to 0.5 for all the samples, which suggests that **D** can no longer distinguish between ground truth and generated samples. Once this stage of training is reached, **G** generates depth images with the same data distribution as ground truth depth images. Table 2 summarizes the network parameters of the Discriminator.

3.3.3 Context-Aware Object Detection with cGAN. While the base model assumes that sensing samples are continuously available, in practice, simultaneous networking and sensing on a picocell can result in intermittent availability of the sensing samples. To recover the missing sensing samples, we propose context-aware object detection with a cGAN. To achieve communication and sensing tasks, we use the "slot sequence" ($S_{timestamps}$) from the 5G network protocol (see Algorithm 1). During the learning phase, we can drop the mmWave sensing samples to emulate the networking slots and push our system closer to the actual hardware that performs both communication and sensing tasks. With fewer mmWave samples available, *CoSense* has a limited ability to detect and locate pedestrians and vehicles. Furthermore, we assign most of the time slots to network communication since the picocell's primary function is to support required data throughput. We improve our heatmap prediction process by estimating the static and dynamic heatmaps based on the past few observed heatmaps. We expect the static heatmap to be primarily stationary and exhibit minor changes, whereas the dynamic heatmap with moving pedestrians and vehicles could change significantly, such as when new objects enter or leave the FoV. Figure 9 illustrates the process of removing and recovering static and dynamic heatmaps based on different slot sequence configurations.

Figure 10 illustrates the deep learning architecture for predicting dynamic heatmaps, which utilizes the previous dynamic heatmaps and the dominant reflecting points along with their corresponding velocities to forecast the movements of objects in future dynamic heatmaps. The underlying idea is that passing the prior dynamic heatmap alongside the group of points with velocity property will facilitate the generation of the dynamic heatmap based on the direction and speed of objects in the heatmap. To select the network for dynamic heatmap prediction, we try multiple vision models for feature extraction to predict the next dynamic heatmap from the current dynamic

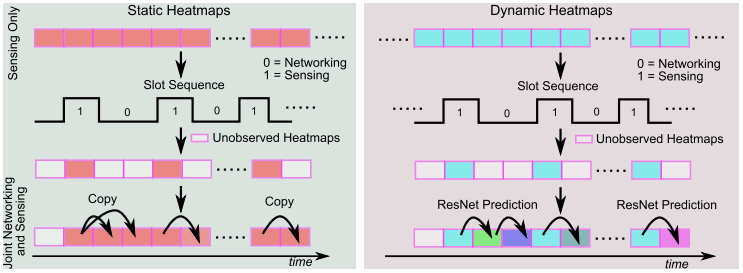

Fig. 9. *CoSense*'s static and dynamic heatmap recovery processes.

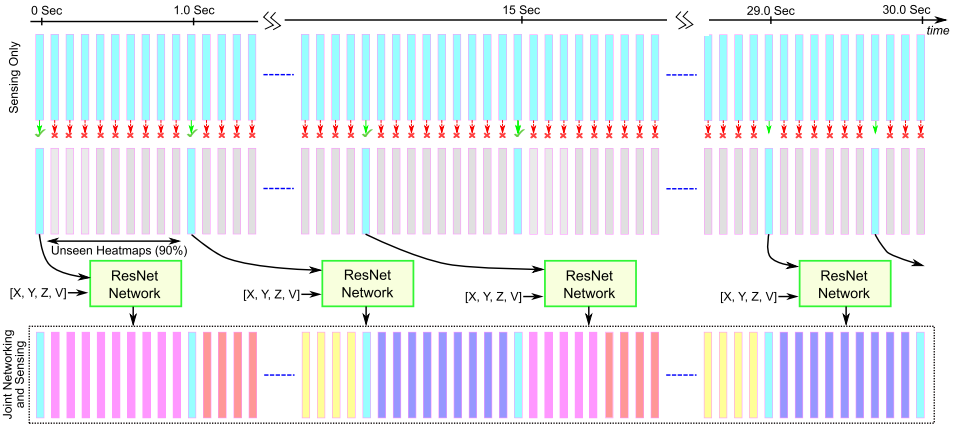

Fig. 10. An example of the dynamic heatmap prediction process with residual network (ResNet18) for a complete batch of data (with 90% unseen sensing samples).

heatmap, such as VGG16 and InceptionV3 [60]; however, Residual Network [61] performed best in our test dataset. Following multiple convolution layers of various filter sizes and a series of activation functions of Residual Networks [61], we obtain an abstract feature vector D_F of size $1,000 \times 1$. Similarly, we encode the velocity of high **signal-to-noise ratio (SNR)** points (i.e., strong reflecting objects) and pass them through a series of 1D convolution layers to obtain a feature vector V_F of size 50×1. Finally, we concatenate V_F and D_F and pass through a series of deconvolution layers to predict $D'(r, \theta)_{t+1}$ at the output layer of size 256×256. Mathematically, we can approximate this as $D'(r, \theta)_{t+1} = DHP_\beta([D(r, \theta)_t, V_t])$, where DHP_β represents the parameterized dynamic heatmap prediction network and $[D(r, \theta)_t, V_t]$ is the dynamic heatmap and velocity at time t.

By leveraging the recovered dynamic heatmaps, the cGAN based object detection model can access past contextual information about the environment that would otherwise be unattainable due to the networking and sensing obligations of the mmWave device.

3.3.4 Network Loss Functions. The loss function is a critical component of deep learning models that control the optimal convergence of the network. *CoSense* employs a combination of **Mean Squared Error (MSE)** [33] and **Binary Cross Entropy (BCE)** [32] for its cGAN learning framework. BCE measures the entropy loss of the Discriminator's output and helps to guide both Generator and Discriminator for the optimal value of parameters in their networks. MSE loss is used to enforce pixel-to-pixel mapping in the reconstructed depth images. For the network training of pedestrians and vehicles, we use a combined loss function to train the cGAN with two different

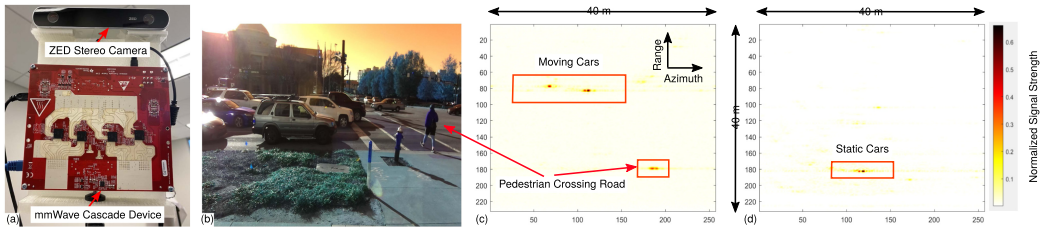

Fig. 11. (a) MmWave cascade device with ZED stereo camera. (b) Left stereo image collected from a ZED stereo camera. (c–d) Static and dynamic heatmaps of objects in the scene after FFT on mmWave samples.

model instances of the same network architecture. The combined loss function is given by the following equation:

$$L_{CoSense} = \lambda_{MSE} \times L_{MSE} + \lambda_{BCE} \times L_{BCE}, \tag{1}$$

where $L_{MSE} = \text{MSE}(G(x, y), M(x, y))$ and $L_{BCE} = \text{BCE}([\mathbf{D}(S(r, \theta), D(r, \theta)), G(x, y) \text{ or } M(x, y)], 1 \text{ or } 0)$. $G(x, y)$ and $M(x, y)$ are generated depth images and ground truth depth images for pedestrians and vehicles, respectively. λ_{MSE} and λ_{BCE} are the hyper-parameters that control the predicted depth values of the image and bounding-box similarity, respectively, and are calculated based on the validation dataset. Finding the optimal values for hyper-parameters is tricky and requires heuristics. We expect our networks to focus on learning the accurate bounding boxes and correct depth values of the objects rather than on the generated image's quality. Thus, intuitively, we can assign a higher weight to λ_{MSE} than λ_{BCE}. We discuss the choices of the hyper-parameters in detail in Section 4. For the dynamic heatmap prediction model, we use the MSE between the predicted and ground truth dynamic heatmap as the loss function to train the network.

In summary, CoSense detects pedestrians and vehicles from mmWave heatmaps and residual network with already observed heatmaps to predict unobserved heatmaps to enable joint networking and sensing.

4 Implementation

4.1 Hardware, Data, and Training

4.1.1 Hardware Platform. Due to the unavailability of open-source 5G/mmWave devices for joint communication and sensing tasks, we build a custom hardware setup for real-time data collection and post-process them offline. Figure 11(a) shows our hardware setup, which consists of a COTS mmWave cascade device, TI MMWCAS-RF-EVM and MMWCAS-DSP-EVM [27, 28] for mmWave data collection, and a ZED stereo camera [29] for RGB and depth image capture. A 3D-printed structure holds the co-located devices in place so that the mmWave and visual sensor data are spatially aligned. The mmWave cascade device combines 4 separate mmWave chipsets [62], each controlling 3 transmit and 4 receive antennas. This results in a system with total 12 transmit antennas, 16 receive antennas, and 192 virtual channels (i.e., $12 \times 16 = 192$). Eighty-six are placed in the azimuth direction, which effectively provides 1.4° azimuth angle resolution. The cascade device uses the following data collection parameters: Start frequency, 77 GHz; frequency ramp slope, 25 MHz/μS; number of complex ADC samples, 256; ADC sampling rate, 8 MHz/s; sweep duration, 40 μS; 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 ~48 m with a range resolution of 0.19 m and has a total bandwidth of 800 MHz. Since the cascade device collects reflection signals from four separate chipsets, each introducing a slightly different but fixed offset in time and phase, the device needs to be calibrated offline once. Appendix A describes the calibration process in detail. The ZED camera is used to collect the ground truth data, capturing stereo RGB and depth images

Table 3. Description of the Environment Near Traffic Intersection and Garage

Environment	Number of Samples	Static Objects	Dynamic Objects
Traffic Intersection	40,000	Traffic lights, fire hydrant	Pedestrian, Vehicles
Near Garage	7,500	Parked cars, vegetation, garage in background	Pedestrian
Near Garage + Fog	2,500	Parked cars, vegetation, garage in background	Pedestrian

with millimeter resolution and 10 frames per second. To eliminate spurious reflections far away from a traffic intersection, we limit the maximum depth range of both the mmWave device and ZED camera to 40 m.

4.1.2 Real Data Collection. We collect real datasets from the custom-built setup across different traffic intersections around our office building (see Figure 11(b) for an example). We place our setup with different orientations in the traffic intersection to capture the multiple scenarios of pedestrians and vehicles (camera images of Figure 11(b) and Figures 12(a)–(e)). Since the mmWave data and stereo images are from different COTS devices, a tight synchronization is unavailable in hardware. Therefore, we use software synchronization based on the timestamps of data samples. A MATLAB program, running on a host PC, initiates the configuration of the mmWave device using the mmWave studio [63], which takes a few minutes to complete. Subsequently, the MATLAB program triggers a Python script that is programmed to collect stereo images and depth images from the ZED camera. As the configuration of the ZED camera is typically faster than that of the mmWave device, the ZED camera is programmed to wait for further instructions from MATLAB before collecting data. Once the mmWave device is configured, both the ZED camera and mmWave device are triggered to capture samples in real time. Each data collection lasts approximately 30 seconds, yielding approximately 300 data samples. A single data sample provides a static heatmap, dynamic heatmap, object points with their corresponding velocities, and stereo RGB and depth images. We first verify the timestamps of data collection between MATLAB and Python by placing nothing in front of the setup and suddenly appearing in the FoV. We compute the SSIM between consecutive stereo images, dynamic frames, and static frames to verify that timestamps have ∼10 ms accuracy between MATLAB and Python. To verify the accuracy of the timestamps between MATLAB and Python, the SSIM was calculated between consecutive stereo images, dynamic frames, and static frames in a controlled setup, revealing an accuracy of approximately 10 ms. Figures 11(b)–(d) present examples of the RGB image captured by the ZED camera and dynamic and static heatmaps produced by the mmWave device at a sample traffic intersection during pedestrian crossing.

Table 3 provides a detailed description of the data samples collected to evaluate the performance of *CoSense* in two distinct environments. At a traffic intersection, our setup was positioned to face the intersection and data samples were collected to capture various scenarios, including pedestrians waiting to cross the road, vehicles traversing the intersection, pedestrians crossing the road in both directions, vehicles passing through the intersection without any pedestrians in view, and many other real-life situations. Table 4 summarizes the distribution of the collected data samples, which include a large number of pedestrians, cars, and trucks appearing in the majority of samples. Additionally, we collected data samples using an artificial fog generator [64] with medium and dense liquid density, placed in front of the mmWave device with pedestrians nearby (see Figure 24). In total, we gathered 50,000 data samples over 6 months, equivalent to approximately 1.67 TB of data. We used 40,000 samples for training and reserved the remaining samples for testing and benchmarking. *The collected data from a real-world traffic intersection under uncontrolled conditions allow us to evaluate CoSense's robustness in detecting both pedestrians and vehicles in diverse and challenging environments.*

Table 4. Number of Different Objects Detected by ZED Stereo Camera on 40,000 Data Samples at the Traffic Intersection

Environment	Total Data Samples	# Pedestrians	# Bicycles	# Motorcycles	# Cars	# Buses	# Trucks
Traffic Intersection	40,000	65,000	507	284	150,000	616	16,000

4.1.3 Network Training. We conduct an empirical analysis on multiple training configurations to determine the optimal model and parameters for *CoSense*. To train the networks, we use the learning rate of 0.0005 for successive epochs. We keep the learning rate low to make sure the loss of network does not diverge as training progresses. For the optimizer, we use RMSProp, which is a gradient-based optimization technique with second momentum and decay to update the network parameters [65]. To identify the best values of the network hyper-parameters, we explored different combinations of λ_{MSE} and λ_{BCE}, and find that a cGAN performs better when the ratio between λ_{MSE} and λ_{BCE} is ~ 10. Therefore, $(\lambda_{MSE}, \lambda_{BCE}) = (1, 0.1)$ performs optimally for our neural network models because it aims to find maximum overlap and correct mean depth of the object. *CoSense* achieves that by keeping the contribution of λ_{MSE} higher than λ_{BCE}. We train all our networks for 1,000 epochs but terminate the training if there was no improvement on the validation dataset for 30 consecutive epochs. All of our networks are trained on Python 3.10 [66] using Tensorflow and PyTorch APIs [67, 68] on a host server. A single network training takes ~12 hours with 2 RTX A6000 NVIDIA **graphics processing unit (GPU)** cores [69]. However, training can be further improved by uploading datasets to the cloud server and using **tensor processing unit (TPU)** devices [70].

4.2 5G Network Simulation

Since our custom-made hardware does not support a real-time evaluation of the 5G/mmWave joint networking and sensing applications, we evaluate the effectiveness of *CoSense* by simulating the 5G protocol based on an open-source, realistic Ray-Tracing method [30]. Conventional simulations using Friis path loss [71] are insufficient in capturing the intricacies of channel behavior at high frequencies, whereas the Ray-Tracing takes into account the environmental layout and is capable of providing more accurate channel estimation [72–74]. By implementing the Ray-Tracing method [30], we are able to estimate the channel and then modify the 5G **Medium Access Control (MAC)** layer to enable data scheduling to the user and sensing tasks for vehicles and pedestrians at opportunistic time slots. This allows us to accurately quantify the data throughput and sensing performance in various scenarios.

4.2.1 Channel Estimation. To accurately simulate an environment and estimate realistic channel conditions, we use the open-street map [45] of our traffic intersection. The map provides detailed information on the building structure, lamp posts, and terrain of the intersection (refer to Figure 28(a) in Appendix A.2). The Ray-Tracing method [30] is employed to estimate the propagation paths by sending electromagnetic waves from the transmitter to the receiver and utilizing the **shooting and bouncing rays (SBR)** approach. When a signal encounters a flat surface, it is reflected; however, when it encounters an edge, it undergoes diffraction. Given that the majority of building surfaces are flat, we assume the presence of a flat surface for reflection purposes. In *CoSense*, we only consider the 1^{st} order of reflection apart from the LOS communication path in the simulation since the other reflections are very weak and close to the noise floor at mmWave frequencies [75–77]. We use the following parameters in the Ray-Tracing simulation: picocell antenna height, 4 m; size of picocell antenna, $[8 \times 8]$; user antenna height, 1 m; and size of user antenna, $[2 \times 2]$. Given a large number of transmit and receive antennas for the picocell, a narrow directional beam can be employed. The user, with a smaller number of antennas, uses an

omnidirectional beam pattern. We run the channel measurements for each **User Equipment (UE)** and calculate the path delays, average path gains, angles of arrival, and angle of departure to assign the channel properties. These, in turn, are used to calculate the resultant throughput for a user.

4.2.2 Throughput Calculation for Joint Networking and Sensing. The calculation of throughput for joint networking and sensing involves several steps. *First*, we determine the channel properties and the physical parameters in an environment following Section 4.2.1. *Second*, we modify the 5G MAC layer to control the data transmission scheduling, where the parameters for modulation (e.g., 16 QAM), code rate (e.g., 490/1024), number of **hybrid automatic repeat request (HARQ)** processes (e.g., 16), number of resource blocks (e.g., 275), and sub-carrier spacing (e.g., 120 kHz) are set. The physical downlink parameters are then configured, and the total number of slots for simulation is calculated to schedule the different types of data transmission to the UE via the downlink channel. The transport block sizes are then calculated based on the total number of slots for each transmission and fed into each HARQ process. Note that only the slot designated as "N" is used for data transmission, while the special slot "S" is reserved for sensing and places the channel in idle mode. The transport block is encoded, resource blocks are created, modulation and precoding are applied, and the data is transmitted through the channel using the transmit waveform. The channel is then measured to calculate the received waveforms, which are added with noise to simulate the real-world channel as closely as possible. The noise-added received waveform is demodulated to recover the transmitted data across multiple resource grids. Once the simulation is complete, it provides the effective throughput achieved and the slot sequence used for data transmission. This process is then repeated with multiple slot sequences, and the effective throughput is recorded to evaluate the networking performance. The slot sequence is then used by the object detection model to drop data frames collected at timeslots designated for networking purposes to evaluate the sensing performance. Algorithm 1 in Appendix A.2 shows the process of calculating the downlink's data throughput and its *slot sequence* during transmission.

5 Performance Evaluation

5.1 Evaluation Metrics and Summary

We now evaluate the performance of *CoSense* in two distinct environments: a traffic intersection with diverse pedestrian and vehicle movements and a garage area in foggy conditions. We compare our models with different standard metrics commonly used on bounding box detection problems for the objects and pedestrians plus number of pedestrians and vehicles missed per second.

- **Intersection-over-Union (IoU):** It measures the region overlap between the bounding boxes of the predicted and ground truth objects and pedestrians. The value range is between 0 and 1.
- **Multi-Scale Structural Similarity Index Measure (MS-SSIM):** It measures the similarity between the generated mask image and the ground truth image for pedestrians and vehicles. The value is between 0 and 1.
- **Mean Absolute Error (MAE):** It measures the absolute difference between the predicted metric and the ground truth metrics for pedestrians and vehicles.
- **Miss Rate:** It measures the average number of pedestrians and vehicles missed per second by *CoSense* compared with the ground truth vision-based camera in clear weather conditions.

Evaluation Summary: Our evaluation of *CoSense*'s performance reveals the following key findings. (1) The system accurately predicts pedestrians and vehicles, achieving a median IoU of 0.55 and 0.63, respectively. Furthermore, the mean depth error is less than 0.66 m on 90^{th} percentile data for both pedestrians and vehicles. (2) *CoSense*'s context-aware model reduces sensing overhead by

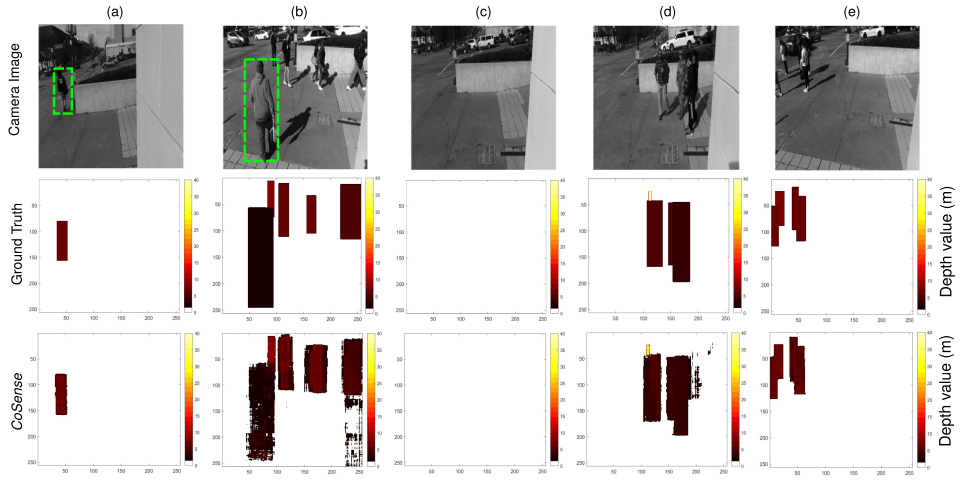

Fig. 12. Example results for pedestrian detection at a traffic intersection. Some pedestrians are marked in camera images.

70%, while only dropping median IoU and MS-SSIM by 27% and 3.5%, respectively. The mean depth error increases from 0.16 m to 0.35 m for the 90th percentile of data, which is deemed tolerable for outdoor applications. (3) Our system accurately predicts the bounding boxes and mean depth for pedestrians even under foggy weather conditions, demonstrating that *CoSense* can effectively operate at traffic intersections and in poor visibility.

5.2 Object Detection with Joint Networking and Sensing

5.2.1 Base Object Detection Model. In this section, we evaluate the performance of the *CoSense* model for detecting pedestrians and vehicles in traffic intersections during office working hours. We collect 50K data samples and preprocess them to produce mmWave heatmaps and ground truth depth images (following Section 3.2). Of the 50K samples, 40K samples are used for training *CoSense*'s base object detection model (Section 3.3). After training, we use the remaining 10K samples to predict the bounding boxes for pedestrians and vehicles from mmWave signals. As an illustration, Figures 12(a)–(e) show the generated bounding boxes for multiple pedestrians in a sample test case. In Column (a) of Figure 12, there is a single pedestrian waiting to cross the road while vehicles are moving; *CoSense* accurately predicts the pedestrian's bounding box. *CoSense* also performs well in generating accurate bounding boxes for other static and dynamic pedestrians (see Columns [b],[d], and [e] of Figure 12). Figure 12(c) represents a scenario with no pedestrians; the system predicts that accurately as well. Also, Figures 13(a)–(e) depict the bounding box generation for single and multiple vehicles on the road, including those that are crossing the street or waiting for a traffic signal. *CoSense* accurately generates depth images for all vehicles. While it can occasionally output spurious blobs on the bounding boxes, we can easily discard them since they are small in size and irregular in shape.

Figures 14(a)–(c) show the IoU, MS-SSIM, and MAE between the ground truth and generated bounding boxes across all the test samples for pedestrians and vehicles. For pedestrians, *CoSense* achieves a median IoU of 0.55 and 90th percentile IoU of 0.76, indicating a good match across most of the samples. For vehicles, *CoSense* achieves a median IoU of 0.62 and 90th percentile IoU of 0.83. The detection performance for vehicles is better compared with pedestrians; this is intuitively correct because vehicles have a larger and smoother surface area compared with pedestrians and can reflect strong mmWave signals. Figure 14(b) shows the median MS-SSIM of *CoSense* generated

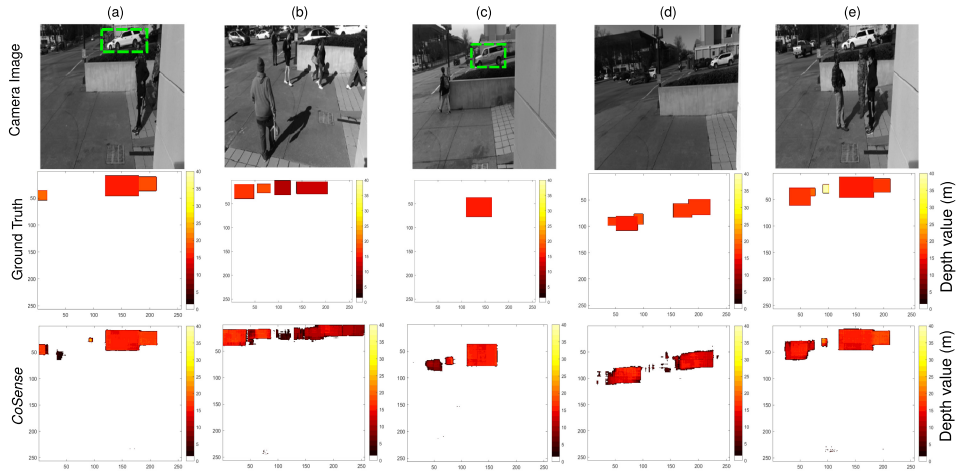

Fig. 13. Example results for vehicle detection at a traffic intersection. Some vehicles are marked in camera images.

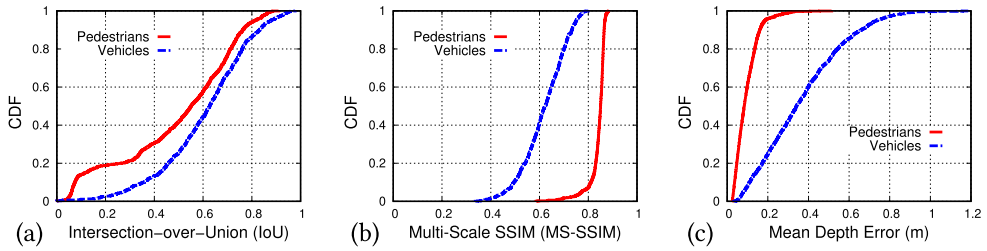

Fig. 14. Results for different metrics from the base model for pedestrian and vehicle detection. (a) Intersection-over-Union (IoU). (b) MS-SSIM. (c) Mean depth error.

bounding boxes are 0.85 and 0.62 for pedestrians and vehicles, respectively. This result indicates that *CoSense* accurately generates the bounding box for pedestrians and vehicles. Furthermore, Figure 14(c) shows the *CoSense*'s performance in identifying the depth of the vehicles and pedestrians from the mmWave device. *CoSense* achieves a median depth error of 0.08 m and 0.34 m for pedestrians and vehicles, respectively. Vehicles have higher depth error than pedestrians because the object is larger and has more range variation from a mmWave device. Still, the 90^{th} percentile depth error does not exceed more than 0.66 m, indicating high accuracy in ranging for both pedestrians and vehicles. *The high accuracy result on pedestrians and vehicles indicates that the azimuth angular resolution of mmWave heatmaps and context-aware learning network enables such high similarity between generated and ground truth depth images.*

5.2.2 Effect of Sensing on 5G Networking. To evaluate the impact of sensing on networking performance, we modify the data transmission schedules of the 5G protocol as described in Section 4.2 and Appendix A.2 to run the sensing applications on top of networking. We simulate the picocell and user device in an open-street map near the traffic intersection, with the picocell and user device placed 40 m apart at the height of 4 m and 1.5 m, respectively (see Figure 28 for an illustration). We control the data scheduling pattern by sending packets to the user device through the emulated PHY and MAC layers when the slot symbol is 'N' and placing the picocell node in an idle state when the slot symbol is 'S'.

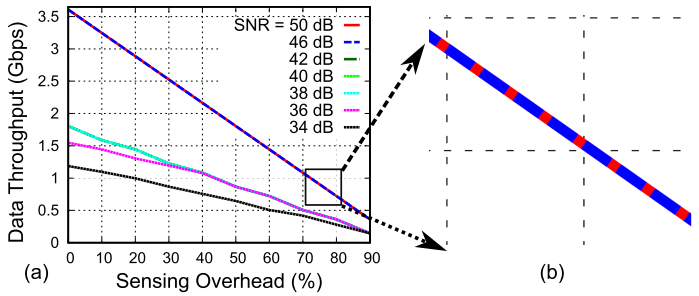

Fig. 15. (a) Throughput results under various sensing overheads and SNR conditions. (b) Zoom-in view of percentage of Sensing Overhead for 50 dB, 46 dB, and 42 dB SNR.

Figure 15 shows the data throughput from the picocell to the user in the downlink channel at various SNRs. We also calculate additive white Gaussian noise (AWGN) with the given SNR and add it to the received waveform through the channel. When the channel has a high SNR, i.e., the SNR is greater than 46 dB, the picocell transfers data with 3.25 Gbps with 10% sensing overheads. However, as the sensing overhead increases, the data throughput drops almost linearly. When the SNR drops below 40 dB, the picocell node may have to retransmit some packets due to potential bit errors on the data transport block. In this scenario, the data throughput is 1.5 Gbps for 10% sensing overhead and only drops to 1.25 Gbps for 30% sensing overhead, which can still enable detection of objects and pedestrians in the environment. *This result suggests that we can piggyback mmWave sensing on top of networking without a significant drop in the user data throughput.*

5.2.3 Effect of Networking on Sensing. This section examines the effect of different percentages of data frame drop on sensing accuracy. In the base model, we evaluate the object detection accuracy of *CoSense* using mmWave devices for sensing tasks only, i.e., 100% sensing. Since mmWave devices can also transfer data for networking tasks, sensing application is executed only opportunistically, thus, the need to allocate time between networking and sensing to maximize the use of the full bandwidth spectrum. To evaluate under this scenario, we intentionally drop a percentage of the mmWave heatmaps in *CoSense*, ranging from 90% to 10%, to downsample them, resulting in sampling frequencies varying from 9 fps to 1 fps. For each sample, we conduct our experiments on a total of ~83 minutes of samples, with the first 65 minutes allocated for training and the rest for testing. Within the training samples, we formed batches of 10 consecutive mmWave heatmaps and dropped the heatmaps from each batch based on the sensing overhead. For example, if the mmWave device was at 90% sensing capacity, we randomly dropped 90% of the heatmaps. We followed a similar process for sequential test samples.

We first train the dynamic heatmap prediction model by passing the previous dynamic heatmaps and corresponding velocity points. We use the dynamic heatmap prediction model post-training to predict future dynamic heatmaps from past observations. Figures 16(a)–(e) show predicted dynamic heatmaps and corresponding ground truth heatmaps. In Figures 16(b)–(c), *CoSense* accurately predicts the location of a moving object(s) in the dynamic heatmap. However, the prediction is not always accurate due to noise on the previous dynamic heatmap. For example, Figure 16 shows that noise is low in the actual heatmap, but the deep learning network couldnot suppress it. Even though we still observe the peaks at the correct location, they are buried by noise on heatmaps. We can eliminate this by augmenting data for different scenarios and noise cases. In a couple of cases, we also observe that two blobs are merged into one. This particularly happens when two objects are moving towards each other and, hence, the deep learning model thinks they are close in the upcoming frames. Overall, the *CoSense* dynamic heatmap prediction model preserves the peaks of

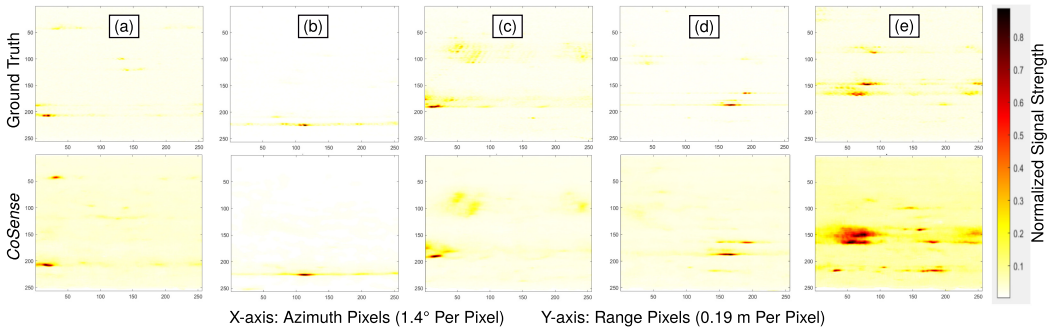

Fig. 16. Sample outputs from dynamic heatmap prediction from *CoSense*.

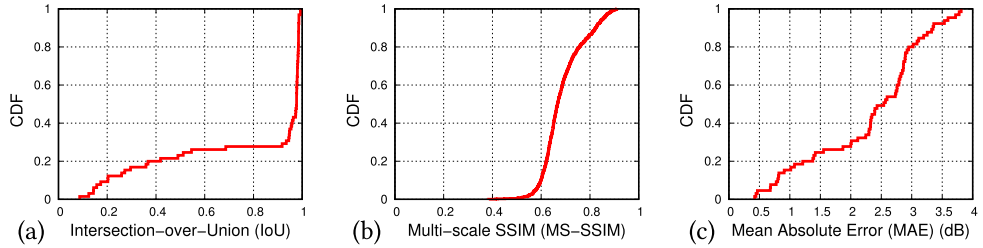

Fig. 17. Results for different metrics from the dynamic heatmap prediction framework: (a) IoU. (b) MS-SSIM. (c) Mean absolute signal strength error (pixel-to-pixel).

the next heatmap based on the previously observed heatmap and velocity points. Figures 17(a)–(c) show the IoU, MS-SSIM, and mean absolute pixel-to-pixel error across test samples. We get a median pixel-to-pixel error of 2.54 with not more than 3.35 for 90% of test samples. However, we get very high IoU and low MS-SSIM because dynamic heatmaps are sparse blobs. Predicting a blob at the right location in the heatmap provides high IoU. However, the image similarity of the predicted heatmap and ground truth heatmap could still be low because of the noise on the predicted dynamic heatmap.

Once the dynamic heatmap network training is complete, we use it to recover dynamic frames before feeding them into the object detection network to get the depth images of pedestrians and vehicles. Figure 18(a) shows the IoU between the generated depth image of pedestrians with different sensing overheads. *CoSense*'s median IoU only drops to 0.4 from 0.55 for 30% of sensing, and the difference holds for 90% of samples. Similarly, we observe a 3.5% drop on median MS-SSIM with similar sensing overheads, i.e., 30%; however, the drop is negligible on the 90^{th} percentile (see Figure 18(b)). Finally, Figure 18(c) shows that we can predict pedestrians with 0.09 m median error; however, there is a 90^{th} percentile error increase with the drop in sensing overhead. Mean depth prediction error increases to 0.35 m from 0.16 m for 90% of samples when sensing overhead is reduced to 30% from 100%. The amount of error on the mean depth of the pedestrian is tolerable in practical settings, especially when detecting pedestrians ahead of time. Figure 19 also shows similar results for vehicles with a slight improvement in IoU and MS-SSIM and a slight deterioration in depth error since vehicles have more rigid structures and are typically farther away than the pedestrians. *The high IoU and MS-SSIM and low depth error with low sensing overhead show that CoSense is capable of performing both networking and sensing duties without a significant drop of performance in both of them.*

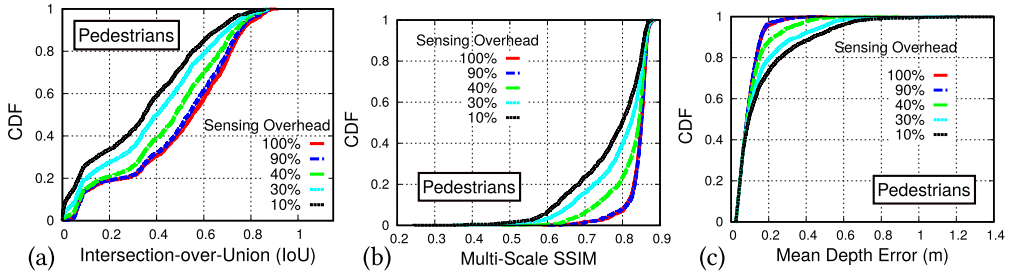

Fig. 18. Different metrics on test samples for pedestrians with various sensing overheads: (a) IoU. (b) MS-SSIM. (c) Mean depth prediction error in meters.

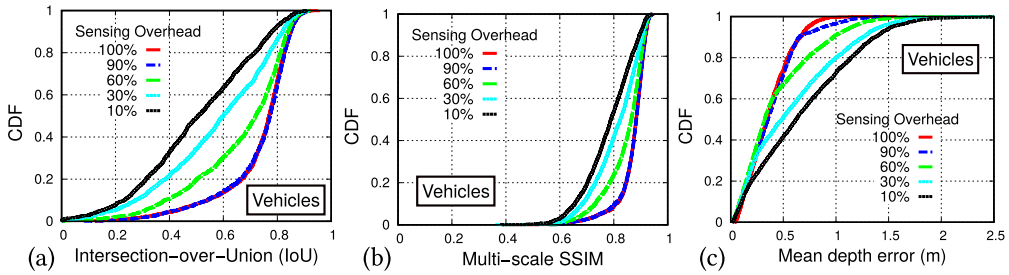

Fig. 19. Different metrics on test samples for vehicles with various sensing overheads: (a) IoU. (b) MS-SSIM. (c) Mean depth prediction error in meters.

5.2.4 Pedestrian and Vehicle Miss Rate. We now evaluate *CoSense* in predicting the number of pedestrians and vehicles passing through the scene per second. To count the number of pedestrians, we initially use all available sensing samples and assume YOLOv5 detection on camera images as the ground truth in clear weather conditions. Next, we track pedestrians using bounding overlap and identify the times when each pedestrian enters or leaves the data frames to count the total number of pedestrians in each frame. If a frame was dedicated to networking, we count any pedestrians that left or entered the frame as a miss count. For instance, if the sensing overhead is reduced to 30%, we count all pedestrian changes during the remaining 70% of the time as missed counts. We aggregate these numbers over one second and define the Miss Rate. We follow a similar process for counting the number of vehicles missed per second. If *CoSense* fails to predict the bounding box of a pedestrian or vehicle, we count it as a miss.

Figures 20(a)–(c) show the number of pedestrians missed with and without *CoSense* at various sensing overheads. At 30% sensing overhead, without *CoSense*, we may miss up to 8 pedestrians per second. In contrast, the maximum miss rate is reduced to 2 from 8 with *CoSense*'s context-aware object detection network. Similarly, Figures 21(a)–(c) show the number of missed vehicles with identical sensing overheads. At 30% sensing overhead, without *CoSense*, we may miss up to 10 vehicles per second, but *CoSense* reduces this to 3 vehicles per second at most. We observe a higher miss rate for vehicles compared with pedestrians because they are far from the mmWave device; hence, some reflections from them may be missed in some data samples.

5.2.5 Velocity Estimation. While the location and count of the objects and pedestrians are important, their velocity plays a crucial role in determining the collision probability in ensuring road safety. To estimate the velocity and direction of objects, the mmWave device can send multiple

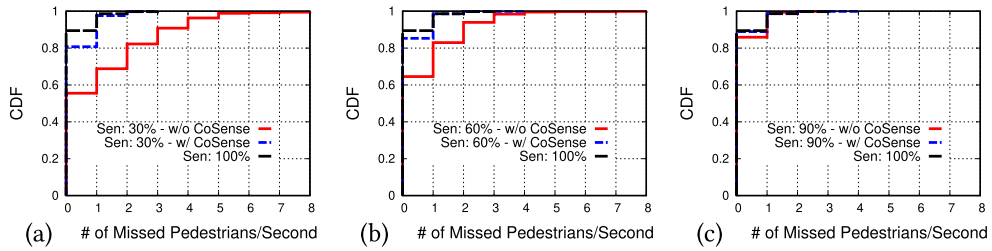

Fig. 20. Number of pedestrians missed per second near the traffic intersection with and without *CoSense* for (a) 30% sensing overhead, (b) 60% overhead, snf (c) 90% overhead.

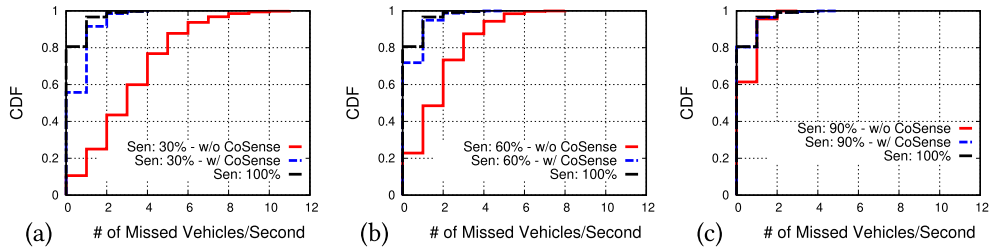

Fig. 21. Number of vehicles missed per second near the traffic intersection with and without *CoSense* for (a) 30% sensing overhead, (b) 60% overhead, and (c) 90% overhead.

back-to-back signals within a short time and measure Doppler shift. The resulting signals can be analyzed to determine the range, azimuth, and elevation of multiple points for different objects in 3D, along with their velocity. However, due to specularity, only a few points from an object can be obtained, making it difficult to segment these points for pedestrians or vehicles on point cloud data or mmWave heatmaps. Therefore, to label different points in the heatmap and track objects accurately, we use stereo images of corresponding heatmaps and apply YOLOv5 object detection to calculate bounding boxes for pedestrians and vehicles. We then compare the overlapping of bounding boxes on consecutive images to track different objects accurately. Additionally, we calculate the mean depth of each object and apply translation to estimate its range for the mmWave device. Finally, we compare the estimated range and velocity of the objects to all the points from the mmWave device to locate the closest match.

Figures 22(a)–(f) show examples of estimated velocity of pedestrians and vehicles on static heatmaps, dynamic heatmaps, and actual camera images. Positive velocity values indicate objects moving toward the mmWave device and vice versa. For example, Figure 22(b) shows that *CoSense* correctly labels a truck (blue circle) moving away from the device at 15.67 m/s (~35 mph), which matches the expected speed of vehicles on the given road. It also predicts three pedestrians on the mmWave heatmap (green circle), two of which have no movement and one with a velocity of -1.1 m/s. However, we observe that *CoSense*'s accuracy in labeling pedestrians and vehicles is affected by their range and azimuth angles, as only a few reflections arrive from the object, leading to incorrect labeling. For instance, Figure 22(a) shows that *CoSense* labels a pedestrian waiting for traffic lights with a speed of -2.238 m/s, while the actual speed is observed from the truck moving away, leading to an incorrect prediction for the label and direction. Nevertheless, *CoSense* accurately predicts the velocity of pedestrians and vehicles on camera images, particularly for pedestrians, as they are closer to the camera and provide error-free depth values.

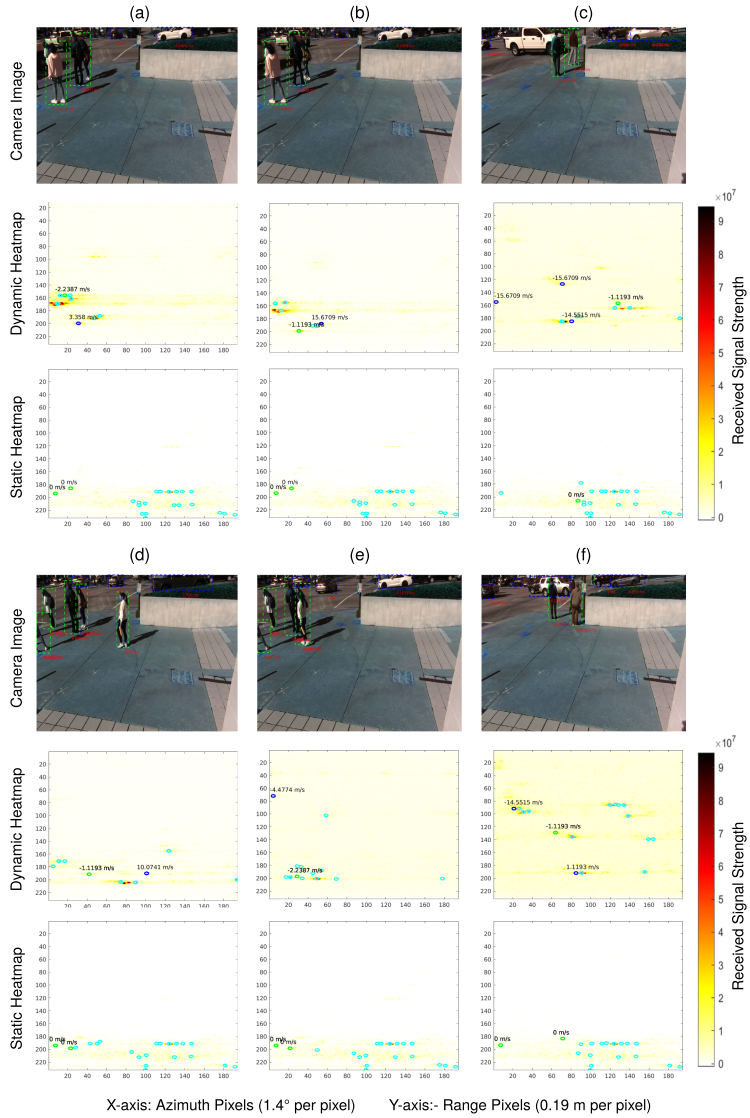

Fig. 22. Camera image, dynamic heatmap, and static heatmap with the velocity of pedestrians and vehicles.

Figure 23 presents the velocity estimation error for detected pedestrians and vehicles. For pedestrians, the estimated velocity has a median error of 0.4 m/s and a 90th percentile error of 1.65 m/s. For vehicles, the estimated velocity has a median error of 1.51 m/s and a 90th percentile error of 4.8 m/s. The larger velocity error for vehicles is due to their higher speeds and distance from the mmWave device, which results in erroneous depth values and, in turn, higher velocity estimation error. In the future, we plan to explore multi-perspective collaborative sensing that could improve the accuracy velocity measurement by using a closer-by picocell (see Section 7).

5.2.6 Pedestrian Detection Under Foggy Conditions. We now evaluate the performance of *CoSense* under foggy conditions. To create a controlled and realistic experiment, we use artificial fog generated by a water-based fluid fog machine, following the methods described in previous

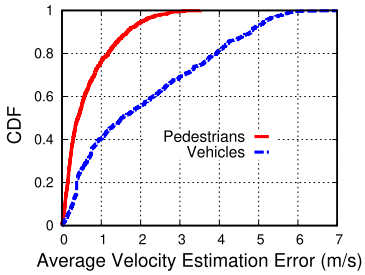

Fig. 23. Average velocity estimation error with mmWave device compared with the camera image velocity estimation for pedestrians and vehicles.

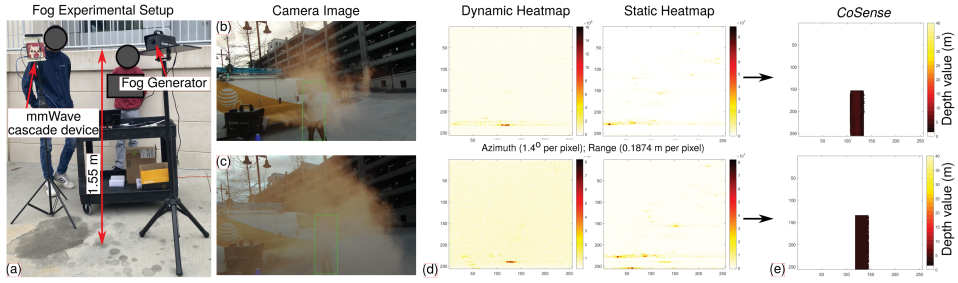

Fig. 24. (a) Experimental setup of fog trials for pedestrian detection at ~2.5 m. (b–c) Two camera images under foggy conditions. (d–e) Dynamic heatmaps, static heatmaps, and *CoSense*'s generated bounding boxes for the two pedestrians.

Table 5. Summary of Pedestrian Detection under Medium and Poor Visibility

CoSense	#Samples	Depth Error (median)	Depth Error (90^{th}%-ile)	IoU (Median)	IoU (90^{th}%-ile)	MS-SSIM (Median)	MS-SSIM (90^{th}%-ile)
Pedestrian 1	152	0.03 m	0.05 m	0.78	0.84	0.93	0.94
Pedestrian 2	184	0.02 m	0.13 m	0.89	0.93	0.94	0.95

works [25, 78, 79]. Figure 24(a) shows an experimental setup with our setup, which includes the DFM-400S fog machine [64]. We collect data samples from two pedestrians with different body somatypes, who stood in a natural pose at a distance of approximately 2.5 m from the setup. We process the mmWave samples through the pre-trained bounding box generator model and compare the output in foggy conditions with ground truth in clear conditions. Figures 24(b)–(c) show two sample RGB images under medium and poor visibility, in which it is difficult to detect the pedestrians. However, the dynamic heatmap of Figures 24(d)–(e) show a concentrated energy peak at the range of approximately 2.5 m, corresponding to the sway movement of the pedestrians during foggy conditions. The *CoSense* deep learning model leverages these unique heatmap features to accurately identify the bounding box of the pedestrians and predict their range (Figure 24(d)). Table 5 summarizes *CoSense*'s performance on over 100 data samples for each pedestrian. We observe a median IoU of 0.78 and 0.89, median MS-SSIM of 0.93 and 0.94, and median depth errors of 0.03 m and 0.02 m for *pedestrian 1* and *pedestrian 2*, respectively. This high accuracy is expected since mmWave signals can easily penetrate through fog.

5.2.7 Pedestrian Detection Under Different Light Conditions. Vision-based sensors (cameras, LiDAR) do better than mmWave devices to detect pedestrians during clear weather because they capture more detail of the object. However, camera images fail to capture the information about the

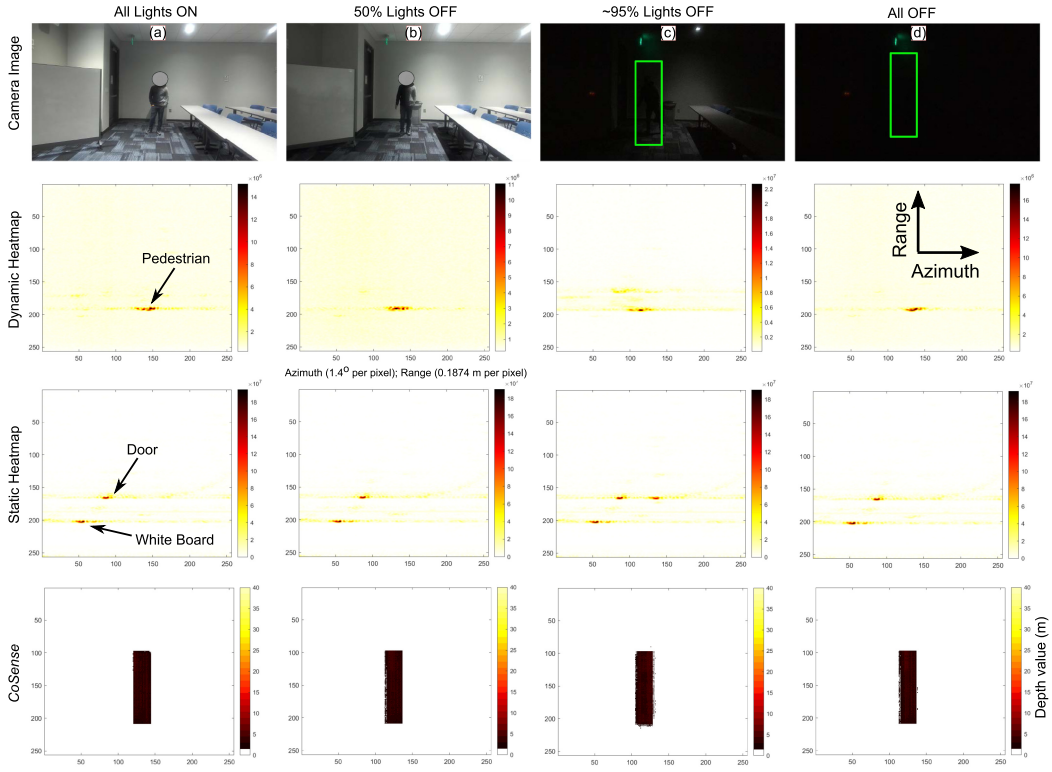

Fig. 25. Experimental setup under different light conditions. We have light bulbs at 20 different locations on the ceiling: (a) all lights ON, (b) 50% lights OFF, (c) ~95% lights OFF, and (d) all lights OFF.

object in front of them during low-light conditions or nighttime, resulting in missed classification and incorrect bounding box generation. Although it would be ideal for performing experiments in outdoor environments, it is not feasible for the following reasons. *First,* we cannot control the lights on the street and it may not be safe to turn off the street lights completely during live traffic flow. *Second,* we cannot control the number of vehicles and pedestrians and their positions. However, we can test the core idea that mmWave devices work in all light conditions by controlling the lights of the indoor environment. Suppose that we can show that mmWave reflections are not affected by ambient light and show pedestrian detection in the indoor environment. These results indicate that mmWave devices can still get a reflection from pedestrians in outdoor environments at different light conditions because the medium of transmission of mmWave, i.e., air, remains unchanged in the indoor and outdoor environments. To this end, we ask the pedestrian to stay inside the classroom and turn the lights on and off while we capture the data samples, including the mmWave heatmaps and camera images. To simulate different light conditions, we start the data collection with normal light conditions, i.e., all lights are ON. After a few seconds, we ask the pedestrian to turn off 50% of the lights, collect data samples for a few more seconds, then turn off ~ 95% of lights, and finally turn all lights off.

Figure 25 shows the summary of the performance of *CoSense* under different light conditions with camera images and static and dynamic heatmaps. The dynamic heatmap represents the reflections from the moving pedestrian, and the static heatmap represents the static objects such as the door, drywall, and white board. When all the light bulbs or even 50% of lights are ON, we can see the

Table 6. Model Size and Inference Time Required for *CoSense*

Model Size (MB)	Inference time (ms)
50.89	31.33

pedestrian in the camera image, and any vision-based method can detect a 2D bounding box of the pedestrian. Similar to the camera-based methods (i.e., YOLOv5), *CoSense* also accurately generates the 2D bounding box of the pedestrian in such cases. However, the camera image captures nothing in poor light conditions, and vision-based methods fail to detect the object (see camera images of Figures 25(c)–(d)). In contrast, the mmWave device still gets the reflection from the pedestrian and other static objects (see dynamic and static heatmaps of Figures 25(c)–(d)), which eventually enables *CoSense* to generate the 2D bounding box of the pedestrian (see Figures 25(c)–(d)). CoSense *consistently generates a 2D bounding box of the pedestrian accurately in all light conditions, showing its robustness to night conditions.*

5.2.8 Runtime Complexity of CoSense. Finally, we evaluate the runtime complexity of *CoSense* by analyzing its average inference time and model size. So far, the training, validation, and inference of *CoSense* have been conducted on a computationally powerful GPU server (2-core RTX A6000) to speed up the process. However, since our system aims to monitor traffic intersections in real time, it is important to also evaluate the inference time on a general-purpose CPU. Thus, we first train *CoSense*'s model on the GPU server and then evaluate its inference time on an 8-core AMD CPU. Table 6 shows the *CoSense*'s inference time and memory size. We observe that *CoSense*'s model is lightweight, with a memory size of only 50.89 MB, making it suitable for deployment on inexpensive networking devices in the future. Furthermore, on average, the model takes 31.33 ms to generate bounding boxes for a single data sample on the CPU. In the case in which the picocell, installed at the traffic intersections, has very low computational power, the data can also be uploaded to a remote GPU server for inference. This ensures that the system continues to operate efficiently, even in resource-constrained environments.

6 Related Works

Millimeter-Wave Networks for Joint Communication and Sensing: The integration of communication and sensing capabilities in a single device has been one of the focuses of beyond 5G and 6G network architectures [80–82], but it poses a challenge to achieve without negatively impacting either functionality [83]. In recent years, mmWave sensing technology has found diverse applications in areas such as gesture sensing, posture identification, pedestrian and vehicle detection, high-resolution image generation, and see-through occlusion [15, 24, 25, 84, 85]. However, these applications are typically standalone and do not address the challenges of joint networking and sensing within a device. Previous research has explored the use of mmWave for communication purposes, such as identifying optimal picocell locations, creating 5G coverage maps, and predicting dominant reflectors to estimate signal strength [74, 86, 87], but without sharing the mmWave device for sensing applications. Some previous works have investigated the coexistence of communication and sensing on a single mmWave device, but they lack experimental results from real hardware and provide only simulated evaluations or require modifications to the standard frame structure and waveform patterns that might be difficult to incorporate into practical networking devices [88, 89]. SPARCS [90] uses the sparse recovery method to enable integrated communication and sensing. However, the method is focused on sensing for indoor environments and evaluated at 60 GHz with 1.76 GHz bandwidth. In contrast, this article presents *CoSense*, a deep learning augmented model that enables the coexistence of communication and sensing on a

single mmWave device without modifying the existing frame structure or waveforms for outdoor environments.

Millimeter-Wave Sensing of Outdoor Environments: The ability of mmWave signals to work under poor visibility or no light has enabled multiple applications, such as privacy non-invasive pose reconstruction and exercise monitoring, high-resolution image generation, liquid and fruit sensing, and robot navigation. [13, 91–93]. However, all the applications are designed and evaluated in indoor, controlled settings. Also, the ability of mmWave to penetrate through rain particles and fog has made it suitable for range detection of objects from vehicles under harsh weather conditions [43, 94], but obtaining fine object details like those provided by vision cameras and LiDARs is challenging due to the limited antenna size at mmWave frequencies [23, 95, 96]. In outdoor scenarios, only a few research works have used mmWave devices for fine-resolution monitoring. Among them, a previous work [24] used horizontal and vertical mmWave devices with 12 virtual channels to detect pedestrians, bicyclists, and cars using a deep learning approach, but required dedicated sensing hardware that continuously sampled the target scene. Another work [25] used cascade mmWave devices to find the bounding boxes of incoming and outgoing vehicles around the ego vehicle but did not consider pedestrian detections or coexistence with networking operations. In contrast, *CoSense* uses cascade mmWave devices to collect data and a deep learning framework to detect both vehicles and pedestrians at traffic intersections while coexisting with networking operations. In the future, we expect *CoSense* to facilitate the exploration of joint networking and sensing applications with mmWave devices in other areas.

7 Discussion & Future Works

Compatibility of *CoSense* with 5G NR Devices: Due to the unavailability of 5G picocells, we evaluate *CoSense* with an mmWave cascade device [28]. The device employs **Frequency Modulated Continuous Wave (FMCW)** signals to estimate the reflection profile of the signal, enabling analysis and detection of objects in the environment. Although 5G picocells do not have the capability to generate FMCW signals, it is worth noting that they utilize **Sound Reference Signal (SRS)** packets [30] for measuring channel quality, determining data transmission rates, and decoding packets. These SRS packets contain valuable information about the environment, essentially serving as an equivalent to the signal reflection profile. Furthermore, 5G picocells can leverage NLOS paths through beam steering, directing signals towards strong reflectors to establish links with user devices. This functionality provides insights into the reflectivity of objects in the vicinity [95]. Therefore, we believe that *CoSense* can operate on 5G NR devices without necessitating any hardware modifications. Unfortunately, current open-source 5G NR devices [97, 98] do not grant user access to SRS packets; thus, we are unable to evaluate *CoSense* on a real 5G NR device. One factor that determines the performance of sensing tasks is the range resolution of the system, which is directly related to the bandwidth usage. Even though our setup supports up to 4 GHz bandwidth, we only use 800 MHz; thus, our system is close to the currently available bandwidth ranges of 5G deployments. *CoSense* achieves a given object detection performance with a limited bandwidth of 800 MHz. We know that in lower frequency ranges such as 28 GHz, 39 GHz, and 47 GHz, bandwidth is less than 4 GHz but more than 800 MHz (850 MHz at 28 GHz and 1.6 GHz at 39 GHz) [99]. Due to similar bandwidth usage, we anticipate our system to perform as expected in 5G NR devices. In the future, we will evaluate *CoSense* on real 5G NR devices once their functionalities become open sourced.

Collaborative Sensing from Multiple Picocells: Although *CoSense* accurately predicts pedestrians and vehicles for the majority of samples irrespective of the number of pedestrians and vehicles

in the frame due to better range (∼ 0.19 m) and angle (∼ 1.4°) resolution, there are still some instances in which it fails to detect them. For example, *CoSense* might only detect a single pedestrian when there are two pedestrians very close to each other. Additionally, we have observed high errors in velocity estimation for vehicles that are far away from the mmWave device (approximately 35–40 meters). To address these issues, we plan to explore collaborative sensing from multiple picocells located close by on traffic poles to capture the same scene from different perspectives. This will require synchronizing and exchanging mmWave sensing samples between picocells, and existing approaches such as the Integrated Access and Backhaul [37] in the 5G NR standard can facilitate them. By combining reflected signals from multiple picocells, we can compensate for objects that are not detected by one picocell from one perspective. Since we expect there to be more than one picocell in a given environment, we can also collaborate among them and optimize the data scheduling to design a "slot sequence" so that they could sense the environment opportunistically at different timestamps. This will improve both the accuracy of object detection and networking performance for each picocell.

Model Generalization to Other Picocells and Evaluation at Diverse Traffic Intersections: Our experiments are designed and evaluated with an mmWave device that has 192 virtual channels and uses 800 MHz bandwidth for capturing reflections. However, 5G picocells with different form-factors and costs can have different antenna sizes (e.g., 64 or 256 antennas), affecting their resolution in range, azimuth angle, and elevation angle, and accuracy in detecting objects. Another factor that could affect object detection accuracy is the height of picocells, as those placed at higher elevations might have more attenuated reflected signals. In the future, we will investigate the performance of *CoSense* with different antenna sizes, operational bandwidths, and picocell heights, and find potential avenues for amending the model and improving the accuracy. Furthermore, our data samples were collected at a downtown traffic intersection near our office building, with a speed limit of 35 miles per hour and busy traffic flow. Other intersections may have different densities of pedestrians, speed limits, and road structures (e.g., three-way, four-way, or five-way). In the future, we will evaluate our model by collecting more data samples from diverse intersections with different picocell placements.

8 Conclusion

CoSense is designed to enable the coexistence of networking and sensing on next-generation mmWave picocells for traffic monitoring and pedestrian safety at traffic intersections. The system proposes the use of 5G picocells, which operate at mmWave frequency bands and provide higher data rates and higher sensing resolution than traditional wireless technology. *CoSense* designs customized deep learning models that recover missing information in space and time about the target scene and solve the challenges with the coexistence of networking and sensing. The system is evaluated on diverse data samples captured at traffic intersections and demonstrates high accuracy in detecting pedestrians and vehicles. *CoSense* offers a promising solution for improving traffic monitoring and pedestrian safety using next-generation ubiquitous networking devices in all weather conditions.

A Appendix

A.1 Cascade mmWave Device Calibration

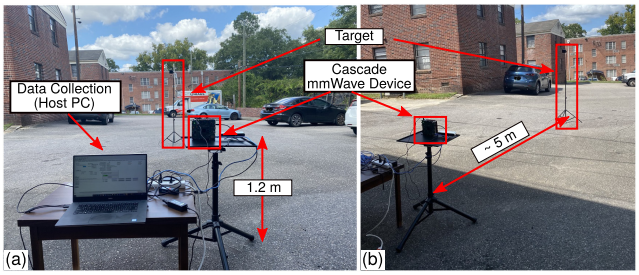

Fig. 26. Millimeter-wave cascade device calibration setup. (a) Front view. (b) Side view. Note that the device needs to be calibrated offline only once.

The mmWave cascade device in our experiments consists of four separate chipsets, each controlling a set of transmit and receive antennas. Each chipset introduces a slightly different offset in time and phase of their received signals, potentially leading to inaccurate estimations of objects and pedestrians' range and locations. This is because we need synchronized signals across all transmit and receive antenna pairs to focus the energy in a certain direction. To mitigate this issue, the device must be calibrated once offline since the offsets between the chipsets are fixed at the design time. This is accomplished by collecting mmWave reflections from a single thin target at a specific distance and using these reflections to estimate correction parameters that provide the necessary gain for each antenna to achieve high power in the main lobe beam of the device [100].

In *CoSense*, the cascade device is calibrated by pointing the setup toward a thin reflector located at a distance of approximately 5 meters from the center of the device (at 0° in azimuth and elevation). The device is positioned approximately 1.2 m above the ground to minimize ground reflections and to avoid other potential reflections (see Figures 26(a)–(b)). Data samples are collected for approximately 2 s, experiments are repeated 100 times, and a calibration program [101] is executed to generate the accurate weights for each virtual antenna. Then, these weights are used for all subsequent real experiments in traffic intersections. Figures 27(a)–(b) show the target's range/azimuth plots for uncalibrated and calibrated cases. We can see that energy is more focused towards a single location close to the thin reflector after the device is calibrated.

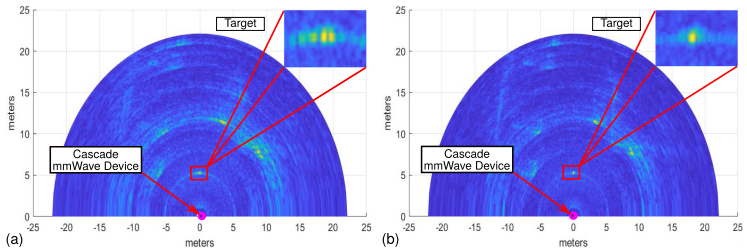

Fig. 27. Range/azimuth plots. (a) Before calibration. (b) After calibration. Note that the sharpness increases after calibration.

A.2 More Details on 5G Network Simulation

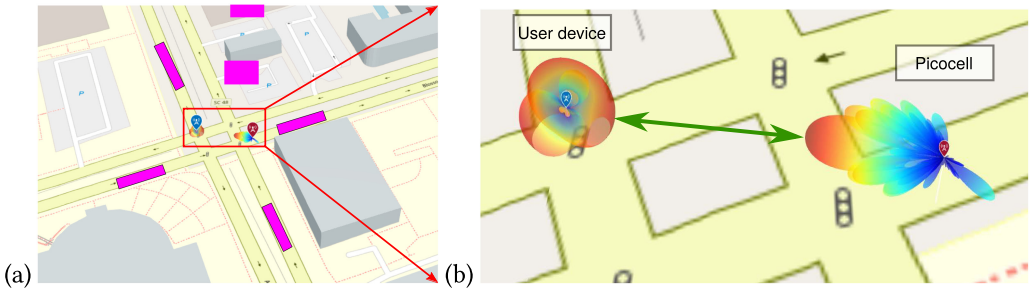

Fig. 28. (a) An open-street map view of the site with the virtual location of picocell and user device around the traffic intersection. (b) Zoom-in view of picocell's directional beam pattern and user device's omnidirectional beam pattern.

ALGORITHM 1: *CoSense*'s 5G Network Simulation

Initialize picocell ← [f_c=77 GHz, height = $h_{picocell}$, position = ($Lat_{picocell}$, $Long_{picocell}$), ant_size = (8, 8)];
UEs ← [position = (Lat_{UE}, $Long_{UE}$), height = h_{UE}, ant_size = (2, 2)]

for UE in UEs **do**
Estimate the channels for each UE using the Ray-Tracing method between the picocell and UE.
end for

Define the Frame Structure, Numerology, and Slot Sequence as follows:

- SCS ← (15 KHz or 30 KHz or 60 KHz or 120 KHz or 240 KHz)
- Numerology ← (0 or 1 or 2 or 3 or 4)
- Number of Resource Blocks (NRBs) ← (<= 275)
- Number of Frames ← N_{Frame} (total time ← N_{Frame}× 10 ms)
- Duplex Mode ← Half-Duplex (TDD)
- Find opportunistic idle time, and schedule networking and sensing;
  // Example Schedule Pattern ← ['N', 'N', 'N', 'S', 'S', 'N', 'N', 'N', 'N', 'N'] ['N'←Networking Slot, 'S'←Sensing Slot; 2 ms for sensing and 8 ms for networking]

for slot in TotalSlots (SlotsPerFrame ×N_{Frame}) **do**
Set the carrier's number of slots to the current slot.
Calculate the schedule, Sounding Reference Signal (SRS), Channel State Information (CSI), Cyclic Redundancy Check (CRC), and Physical Downlink Shared Channel (PDSCH) values.
Picocell encodes and transmits the data to the UEs.
Decode data at UEs and record *DataState*
Record S-Slot timestamps ($S_{timestamps}$) with Slot Sequence
end for

$Throughput_{UEs}$ ← *DataState*

return $Throughput_{UEs}$, $S_{timestamps}$ (Slot Sequences)

References

[1] Governors Highway Safety Association (GHSA). 2022. New Projection: U.S. Pedestrian Fatalities Reach Highest Level in 40 Years. (2022). Retrieved from https://www.ghsa.org/resources/news-releases/GHSA/Ped-Spotlight-Full-Report22

[2] U.S. Department of Transportation —Federal Highway Administration. 2022. About Intersection Safety. (2022). Retrieved from https://safety.fhwa.dot.gov/intersection/about/

[3] Centers for Disease Control and Prevention. 2022. Pedestrian Safety. (2022). Retrieved from https://www.cdc.gov/transportationsafety/pedestrian_safety/index.html

[4] SAE International. 2022. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. (2022). Retrieved from https://www.sae.org/standards/content/j3016_202104/
[5] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Shiyang Wang, Sen Lin, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. 2020. Towards 3D human pose construction using WiFi. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*.
[6] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
[7] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C. Miller. 2014. 3D tracking via body radio reflections. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14)*.
[8] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. 2015. Capturing the human figure through a wall. In *Proc. of ACM SIGGRAPH Asia*.
[9] CCTV Security Pros. CCTV. ([n. d.]). Retrieved from https://www.cctvsecuritypros.com
[10] RoboRealm. Microsoft Kinect, 2013. ([n. d.]). Retrieved from http://www.roborealm.com/help/MicrosoftKinect.php
[11] Vicon. ([n. d.]). Retrieved from https://www.vicon.com/applications/life-sciences/gait-analysis-neuroscience-and-motor-control/
[12] Dongzhu Xu, Anfu Zhou, Xinyu Zhang, Guixian Wang, Xi Liu, Congkai An, Yiming Shi, Liang Liu, and Huadong Ma. 2020. Understanding operational 5G: A first measurement study on its coverage, performance and energy consumption. In *ACM SIGCOMM*.
[13] Edward M. Sitar IV, Moh Sabbir Saadat, and Sanjib Sur. 2022. A millimeter-wave wireless sensing approach for at-home exercise recognition. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*.
[14] Aakriti Adhikari, Hem Regmi, Sanjib Sur, and Srihari Nelakuditi. 2022. MiShape: Accurate human silhouettes and body joints from commodity millimeter-wave devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 3 (2022).
[15] H. Regmi, Sabbir Saadat, Sanjib Sur, and Srihari Nelakuditi. 2021. SquiggleMilli: Approximating SAR imaging on mobile millimeter-wave devices. *Proc. of ACM IMWUT* (2021).
[16] Chen Wang, Jun Shi, Zenan Zhou, Liang Li, Yuanyuan Zhou, and Xiaqing Yang. 2020. Concealed object detection for millimeter-wave images with normalized accumulation map. *IEEE Sensors Journal* 21, 5 (2020).
[17] Adib Nashashibi, Fawwaz T. Ulaby, and Kamal Sarabandi. 1996. Measurement and modeling of the millimeter-wave backscatter response of soil surfaces. *IEEE Transactions on Geoscience and Remote Sensing* 34, 2 (1996).
[18] Chenshu Wu, Feng Zhang, Beibei Wang, and K. J. Ray Liu. 2020. mSense: Towards mobile material sensing with a single millimeter-wave radio. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020).
[19] Mostafa Alizadeh, Hajar Abedi, and George Shaker. 2019. Low-cost low-power in-vehicle occupant detection with mm-Wave FMCW radar. In *2019 IEEE SENSORS*. IEEE.
[20] N. Munte, A. Lazaro, R. Villarino, and D. Girbau. 2022. Vehicle occupancy detector based on FMCW mm-Wave radar at 77 GHz. *IEEE Sensors Journal* (2022).
[21] Ting Wu, Theodore S. Rappaport, and Christopher M. Collins 2015. The human body and millimeter-wave wireless communication systems: Interactions and implications. In *2015 IEEE International Conference on Communications (ICC'15)*.
[22] Sanjib Sur, Vignesh Venkateswaran, Xinyu Zhang, and Parmesh Ramanathan. 2015. 60 GHz indoor networking through flexible beams: A link-level profiling. In *Proc. of ACM SIGMETRICS*.
[23] 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*.
[24] 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).
[25] 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*.
[26] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. (2014). https://arxiv.org/abs/1411.1784
[27] Texas Instruments. 2023. MMWCAS-RF-EVM. (2023). Retrieved from https://www.ti.com/tool/MMWCAS-RF-EVM
[28] Texas Instruments. 2023. MMWCAS-DSP-EVM. (2023). Retrieved from https://www.ti.com/tool/MMWCAS-DSP-EVM

[29] Stereo Labs. 2023. ZED 2. (2023). Retrieved from https://www.stereolabs.com/zed-2/
[30] MATLAB. 2023. CDL Channel Model Customization with Ray Tracing. (2023). Retrieved from https://www.mathworks.com/help/5g/ug/cdl-channel-model-customization-with-ray-tracing.html
[31] Jiabo He, Sarah Erfani, Xingjun Ma, James Bailey, Ying Chi, and Xian-Sheng Hua. 2021. IoU: A family of power intersection over union losses for bounding box regression. *Advances in Neural Information Processing Systems* 34 (2021).
[32] Alain Hore and Djemel Ziou. 2010. Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*. IEEE, 2366–2369.
[33] James William and Charles Stein. 1992. Estimation with quadratic loss. *Breakthroughs in Statistics: Foundations and Basic Theory* (1992), 443–460.
[34] Verizon. 2023. Fair 5G. (2023). Retrieved from https://fair5g.org/sites/default/files/cwa_sacramento_verizon_june_2019_1_1.pdf
[35] AT&T. 2023. Driving Connected Transportation to the Next Level. (2023). Retrieved from https://about.att.com/blogs/2022/5g-connected-transportation.html
[36] Arvind Narayanan, Eman Ramadan, Jason Carpenter, Qingxu Liu, Yu Liu, Feng Qian, and Zhi-Li Zhang. 2020. A first look at commercial 5G performance on smartphones. In *Proceedings of The Web Conference 2020*.
[37] Erik Dahlman, Stefan Parkvall, and Johan Skold. 2018. *5G NR: The Next Generation Wireless Access Technology*. Elsevier.
[38] Luca Chiaraviglio, Ahmed Elzanaty, and Mohamed-Slim Alouini. 2021. Health risks associated with 5G exposure: A view from the communications engineering perspective. *IEEE Open Journal of the Communications Society* 2 (2021).
[39] Ali A. Zaidi, Robert Baldemair, Hugo Tullberg, Hakan Bjorkegren, Lars Sundstrom, Jonas Medbo, Caner Kilinc, and Icaro Da Silva. 2016. Waveform and numerology to support 5G services and requirements. *IEEE Communications Magazine* 54, 11 (2016).
[40] Bram van Berlo, Amany Elkelany, Tanir Ozcelebi, and Nirvana Meratnia. 2021. Millimeter wave sensing: A review of application pipelines and building blocks. *IEEE Sensors Journal* 21, 9 (2021).
[41] Texas Instruments. 2023. Automotive mmWave Radar Sensors . (2023). Retrieved from https://www.ti.com/sensors/mmwave-radar/automotive/overview.html
[42] Hermann Rohling, Steffen Heuel, and Henning Ritter. 2010. Pedestrian detection procedure integrated into an 24 GHz automotive radar. In *2010 IEEE Radar Conference*.
[43] Angela H. Eichelberger and Anne T. McCartt. 2016. Toyota drivers' experiences with dynamic radar cruise control, pre-collision system, and lane-keeping assist. *Journal of Safety Research* 56 (2016).
[44] 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)*.
[45] Google. 2023. Open Street Map. (2023). Retrieved from https://www.openstreetmap.org/#map=4/38.01/-95.84
[46] Nima Hatami, Yann Gavet, and Johan Debayle. 2018. Classification of time-series images using deep convolutional neural networks. In *10th International Conference on Machine Vision (ICMV'17)*.
[47] A. Anbarasi, T. Ravi, V. S. Manjula, J. Brindha, S. Saranya, G. Ramkumar, and R. Rathi. 2022. A modified deep learning framework for arrhythmia disease analysis in medical imaging using electrocardiogram signal. *BioMed Research International* 2022 (2022).
[48] Tianbo Gu, Zheng Fang, Zhicheng Yang, Pengfei Hu, and Prasant Mohapatra. 2019. mmSense: Multi-person detection and identification via mmWave sensing. In *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*.
[49] Zihao Zhao, Yuying Song, Fucheng Cui, Jiang Zhu, Chunyi Song, Zhiwei Xu, and Kai Ding. 2020. Point cloud features-based kernel SVM for human-vehicle classification in millimeter wave radar. *IEEE Access* 8 (2020), 26012–26021.
[50] Congzheng Han and Shu Duan. 2019. Impact of atmospheric parameters on the propagated signal power of millimeter-wave bands based on real measurement data. *IEEE Access* 7 (2019), 113626–113641.
[51] Vindhya Devalla, Rajesh Singh, Amit Kumar Mondal, and Vivek Kaundal. 2012. Design and development of object recognition and sorting robot for material handling in packaging and logistic industries. *International Journal of Science and Advanced Technology* 2, 9 (2012).
[52] Di Feng, Ali Harakeh, Steven L. Waslander, and Klaus Dietmayer. 2021. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021).
[53] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 2016. Monocular 3D object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[54] Chethan Kumar, R. Punitha, and others. 2020. YOLOv3 and YOLOv4: Multiple object detection for surveillance applications. In *2020 3rd International Conference on Smart Systems and Inventive Technology (ICSSIT'20)*. IEEE.

[55] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2017. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV'17)*.
[56] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. 2016. Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv preprint arXiv:1606.08921* (2016).
[57] Michael Tschannen, Olivier Bachem, and Mario Lucic. 2018. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069* (2018).
[58] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. In *ACM International Conference on Neural Information Processing Systems*.
[59] Hao Ge, Yin Xia, Xu Chen, Randall Berry, and Ying Wu. 2018. Fictitious GAN: Training GANs with historical models. In *European Conference on Computer Vision ECCV 2018*.
[60] PyTorch. 2023. Models and Pre-Trained Weights. (2023). Retrieved from https://pytorch.org/vision/stable/models.html
[61] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[62] Texas Instruments. 2023. AWR2243 Dataset. (2023). Retrieved from https://www.alldatasheet.com/datasheet-pdf/pdf/1245311/TI/AWR2243.html
[63] Texas Instruments. 2023. MMWAVE-STUDIO. (2023). Retrieved from https://www.ti.com/tool/MMWAVE-STUDIO
[64] Donner. 2023. DMF-400S. (2023). Retrieved from https://donnerca.com/
[65] 2023. Optimizers. https://keras.io/api/optimizers/
[66] Open-Source. 2023. Python. (2023). Retrieved from https://www.python.org/downloads/release/python-3100/
[67] Open-Source. 2022. TensorFlow. (2022). Retrieved from https://www.tensorflow.org/
[68] Open-Source. 2023. PyTorch. (2023). Retrieved from https://pytorch.org/
[69] NVIDIA. 2023. RTX A6000. (2023). Retrieved from https://www.nvidia.com/en-us/design-visualization/rtx-a6000/
[70] Google. 2023. Cloud TPU. (2023). Retrieved from https://cloud.google.com/tpu
[71] Theodore S. Rappaport. 2002. *Wireless Communications: Principles and Practice*. Prentice Hall.
[72] S. H. Oh and Noh-Hoon Myung. 2004. MIMO channel estimation method using ray-tracing propagation model. *Electronics letters* 40, 21 (2004), 1.
[73] Daniel C. Araújo, André L. F. de Almeida, Johan Axnäs, and João C. M. Mota. 2014. Channel estimation for millimeter-wave very-large MIMO systems. In *2014 22nd European Signal Processing Conference (EUSIPCO'14)*.
[74] Hem Regmi and Sanjib Sur. 2022. Argus: Predictable millimeter-wave picocells with vision and learning augmentation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 1 (2022).
[75] E. O. Christopher, Harpreet S. Dhillon, and R. Michael Buehrer. 2019. Single-anchor localizability in 5G millimeter wave networks. *IEEE Wireless Communications Letters* 9, 1 (2019).
[76] Lifang Feng, Hongbing Yang, Rose Qingyang Hu, and Jianping Wang. 2018. MmWave and VLC-based indoor channel models in 5G wireless networks. *IEEE Wireless Communications* 25, 5 (2018).
[77] Sajjad Hussain and Conor Brennan. 2019. Efficient preprocessed ray tracing for 5G mobile transmitter scenarios in urban microcellular environments. *IEEE Transactions on Antennas and Propagation* 67, 5 (2019).
[78] Yosef Golovachev, Ariel Etinger, Gad A. Pinhasi, and Yosef Pinhasi. 2019. Propagation properties of sub-millimeter waves in foggy conditions. *Journal of Applied Physics* 125, 151612 (2019).
[79] Yosef Golovachev, Ariel Etinger, Gad A. Pinhasi, and Yosef Pinhasi. 2018. Millimeter wave high resolution radar accuracy in fog conditions — theory and experimental verification. *MDPI Sensors* 18, 7 (2018).
[80] Kinza Shafique, Bilal A. Khawaja, Farah Sabir, Sameer Qazi, and Muhammad Mustaqim. 2020. Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. *IEEE Access* 8 (2020).
[81] Yuanhao Cui, Fan Liu, Xiaojun Jing, and Junsheng Mu. 2021. Integrating sensing and communications for ubiquitous IoT: Applications, trends, and challenges. *IEEE Network* 35, 5 (2021), 158–167. DOI: http://dx.doi.org/10.1109/MNET.010.2100152
[82] Andre Bourdoux, Andre Noll Barreto, Barend van Liempd, Carlos de Lima, Davide Dardari, Didier Belot, Elana-Simona Lohan, Gonzalo Seco-Granados, Hadi Sarieddeen, Henk Wymeersch, and others. 2020. 6G white paper on localization and sensing. *arXiv preprint arXiv:2006.01779* (2020).
[83] Mostafa Zaman Chowdhury, Md. Shahjalal, Shakil Ahmed, and Yeong Min Jang. 2020. 6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions. *IEEE Open Journal of the Communications Society* 1 (2020).
[84] Google. 2015. Project Soli. (2015). Retrieved from https://www.google.com/atap/project-soli/

[85] Timothy Woodford, Xinyu Zhang, Eugene Chai, and Karthikeyan Sundaresan. 2022. Mosaic: Leveraging diverse reflector geometries for omnidirectional around-corner automotive radar. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*.

[86] Arvind Narayanan, Eman Ramadan, Rishabh Mehta, Xinyue Hu, Qingxu Liu, Rostand A. K. Fezeu, Udhaya Kumar Dayalan, Saurabh Verma, Peiqi Ji, Tao Li, Feng Qian, and Zhi-Li Zhang. 2020. Lumos5G: Mapping and predicting commercial mmWave 5G throughput. In *Proceedings of the ACM Internet Measurement Conference*.

[87] Teng Wei, Anfu Zhou, and Xinyu Zhang. 2017. Facilitating robust 60 GHz network deployment by sensing ambient reflectors. In *Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation*.

[88] Qixun Zhang, Xinna Wang, Zhenhao Li, and Zhiqing Wei. 2021. Design and performance evaluation of joint sensing and communication integrated system for 5G mmWave enabled CAVs. *IEEE Journal of Selected Topics in Signal Processing* 15, 6 (2021).

[89] J. Andrew Zhang, Fan Liu, Christos Masouros, Robert W. Heath, Zhiyong Feng, Le Zheng, and Athina Petropulu. 2021. An overview of signal processing techniques for joint communication and radar sensing. *IEEE Journal of Selected Topics in Signal Processing* 15, 6 (2021).

[90] Jacopo Pegoraro, Jesus O. Lacruz, Michele Rossi, and Joerg Widmer. 2022. SPARCS: A sparse recovery approach for integrated communication and human sensing in mmWave systems. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'22)*.

[91] Yucheng Xie, Ruizhe Jiang, Xiaonan Guo, Yan Wang, Jerry Cheng, and Yingying Chen. 2022. mmFit: Low-effort personalized fitness monitoring using millimeter wave. In *2022 International Conference on Computer Communications and Networks (ICCCN'22)*. IEEE.

[92] Zhicheng Yang, Parth H. Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. 2016. Monitoring vital signs using millimeter wave. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*.

[93] Xiangyu Gao, Sumit Roy, and Guanbin Xing. 2021. MIMO-SAR: A hierarchical high-resolution imaging algorithm for mmWave FMCW radar in autonomous driving. *IEEE Transactions on Vehicular Technology* 70, 8 (2021).

[94] 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).

[95] R. Schulpen, L. A. Bronckers, A. B. Smolders, and U. Johannsen. 2020. 5G millimeter-wave NLOS coverage using specular building reflections. In *2020 14th European Conference on Antennas and Propagation (EuCAP'20)*. IEEE.

[96] Qijia Guo, Jie Liang, Tianying Chang, and Hong-Liang Cui. 2019. Millimeter-wave imaging with accelerated super-resolution range migration algorithm. *IEEE Transactions on Microwave Theory and Techniques* 67, 11 (2019).

[97] Junfeng Guan, Arun Paidimarri, Alberto Valdes-Garcia, and Bodhisatwa Sadhu. 2021. 3-D imaging using millimeter-wave 5G signal reflections. *IEEE Transactions on Microwave Theory and Techniques* 69, 6 (2021).

[98] NSF PAWR COSMOS Team. 2023. COSMOS: Cloud Enhanced Open Software Defined Mobile Wireless Testbed for City-Scale Deployment. (2023). Retrieved from https://cosmos-lab.org/

[99] Emerson, Inc. 2023. mmWave: The Battle of the Bands . (2023). Retrieved from https://www.ni.com/en/shop/wireless-design-test/what-is-mmwave-transceiver-system/mmwave–the-battle-of-the-bands.html#:~:text=By%20comparison%2C%2028%20GHz%20offers,GHz%20bandwidth%20in%20the%20US

[100] Muhammet Emin Yanik, Dan Wang, and Murat Torlak. 2019. 3-D MIMO-SAR imaging using multi-chip cascaded millimeter-wave sensors. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP'19)*.

[101] Texas Instruments. 2023. TI mmWave Studio. (2023). Retrieved from https://www.ti.com/tool/MMWAVE-STUDIO

Received 8 May 2023; revised 15 April 2024; accepted 3 May 2024