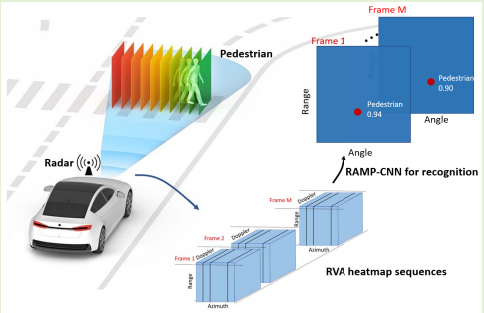
# RAMP-CNN: A Novel Neural Network for Enhanced Automotive Radar Object Recognition

Xiangyu Gao, Guanbin Xing, *Member, IEEE*, Sumit Roy, *Fellow, IEEE*, and Hui Liu, *Fellow, IEEE*

***Abstract*—Millimeter-wave (mmW) radars are being increasingly integrated into commercial vehicles to support new advanced driver-assistance systems (ADAS) by enabling robust and high-performance object detection, localization, as well as recognition - a key component of new environmental perception. In this paper, we propose a novel radar multiple-perspectives convolutional neural network (RAMP-CNN) that extracts the location and class of objects based on further processing of the *range-velocity-angle* (RVA) heatmap sequences. To bypass the complexity of 4D convolutional neural networks (NN), we propose to combine several lower-dimension NN models within our RAMP-CNN model that nonetheless approaches the performance upper-bound with lower complexity. The extensive experiments show that the proposed RAMP-CNN model achieves better average recall (AR) and average precision (AP) than prior works in all testing scenarios (see Table.III). Besides, the RAMP-CNN model is validated to work robustly under the nighttime, which enables low-cost radars as a potential substitute for pure optical sensing under severe conditions.**




***Index Terms*— Automotive radar, object recognition, convolutional neural network, multiple-perspectives, range-velocity-angle heatmap.**

## I. Introduction

THE millimeter-wave (mmW) radars provide highly accurate object detection and localization (range, velocity and angle), largely independent of environmental conditions [1]. Thus, they are fast becoming indispensable in providing critical sensory inputs for environmental mapping in future autonomous vehicle operations. In challenging conditions - nighttime, glaring sunlight, snow, rain or fog - the utility of pure optical sensing (camera and lidar) is diminished [2]; hence the primary objective of this paper is to enable low-cost mmW radar as a potential substitute. To achieve this, radars should deliver semantic environment perception close to what optical sensors provide.

The evolution of radar-based object recognition algorithms has been driven by recent advances in automotive radar hardware using chirp or frequency modulated continuous wave (FMCW) over 77-81 GHz RF bandwidth with integrated digital CMOS and packaging resulting in low-cost radar-on-chip system [3]. Texas Instrument (TI)'s state-of-art 77 GHz FMCW radar chips and evaluation boards - AWR1443, AWR1642, and AWR1843 - are built with low-power 45-nm RF CMOS process and enable unprecedented levels of integration in an extremely small form factor [4]. Other vendors (e.g. Uhnder) have recently unveiled a new, *all-digital* phase modulated continuous wave (PMCW) radar chip capable of synthesizing multiple-input and multiple-output (MIMO) radar capability with 192 virtual receivers, thereby obtaining very high angular resolution [5]. Such high-resolution radars perform similar functions to lidars, i.e., generate dense point-cloud maps from object returns in the vicinity at a fraction of the cost of lidar systems.

FMCW radars transmit a linear frequency modulated signal; the received signal reflected from a target is mixed with the transmitted signal to obtain the beat frequency, which is a function of the round trip delay and therefore can be mapped directly to range [6]. Similarly, transmitting a train of equispaced FMCW chirps (also called a frame) allows Doppler velocity estimation for target that undergoes (relative) radial motion. Such radial motion induces a phase shift over the chirps in a range resolution cell, which is used to compute the Doppler radial velocity [6]. Finally, the use of multiple transmitters and receivers enables azimuth localization of

Manuscript received August 25, 2020; revised October 27, 2020; accepted November 1, 2020. Date of publication November 5, 2020; date of current version January 15, 2021. This work was supported by the CMMB Vision–University of Washington Department of Electrical Engineering (UWEE) Center on Satellite Multimedia and Connected Vehicles. The associate editor coordinating the review of this article and approving it for publication was Dr. Michail Antoniou. *(Corresponding author: Xiangyu Gao.)*

Xiangyu Gao, Guanbin Xing, and Sumit Roy are with the Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: xygao@uw.edu; gxing@uw.edu; sroy@uw.edu).

Hui Liu is with the Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195 USA, and also with Silkwave Holdings, Hong Kong (e-mail: huiliu@uw.edu).

Digital Object Identifier 10.1109/JSEN.2020.3036047

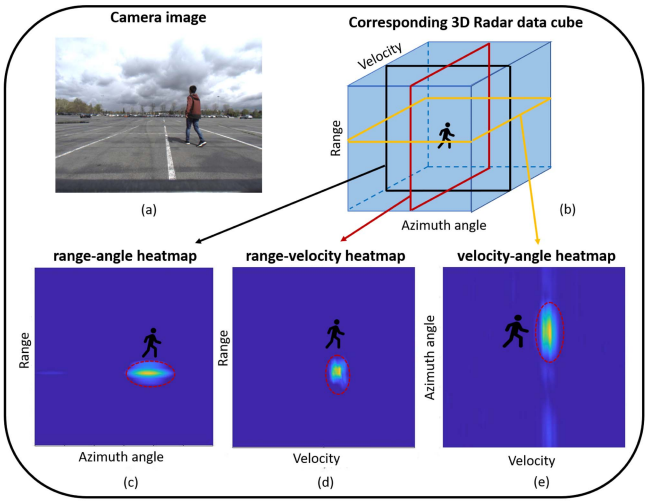



Fig. 1. Abstraction of single frame input radar data and corresponding camera image: (a) The camera image of the pedestrian; (b) The range-velocity-angle radar data cube, three cross profiles of it are shown as figure (c), (d) and (e); (c) The range-azimuth angle heatmap; (d) The range-velocity heatmap; (e) The velocity-azimuth angle heatmap.

target by appropriate beamforming processing of the multiple transmitted waveforms reflected by the target to receiver array [7]. In summary, the analog-to-digital converted (ADC) raw radar data - has 3 dimensions: samples (fast time), chirps (slow time), and receivers) - can be mapped to the 3D radar cube with 3 new dimensions: range, Doppler velocity, and angle. In this paper, we adopt the 3-DFFT [8] to obtain the 3D radar cube that is named the range-velocity-angle (**RVA**) heatmap.[1] [2]

The small form factor of TI 77 GHz boards - while a desirable feature - limits the number of antennas that can be integrated, resulting in poor angular resolution (see Table. I). Specifically, two targets at the same distance and radial velocity are not resolved in angle if separated by less than resolution beamwidth; even if resolvable, the spatial dimension is not well-defined. Hence to achieve reliable object recognition using such hardware, [8], [9] have sought to exploit the unique movement patterns over time for different classes of objects, i.e. rely on temporal patterns over multiple frames rather than spatial discrimination from single-frame data.

Traditional radar object recognition algorithms are based on the statistical signal processing and manual feature selection [10], [11], [12]. For example, it is usual to apply the constant false alarm rate (CFAR) [13] algorithm and DBSCAN clustering [14] algorithm to detect the location of objects.

[1]In general, by the heatmap we refer to the complex image resulting from the FFT operations. When for visualization purposes, we take the amplitude value of the complex heatmap.

[2]In this paper, angle represents the azimuth angle if not specified.

[3]In the calculation of velocity resolution, $\lambda$ is the wavelength of transmitted signal, $T_c$ is the duration of one chirp, $T_c = \frac{1}{N_c f_F}$.

[4]In the calculation of angle resolution, $N_{Rx}$ is the number of elements of receiver array (including the MIMO virtual receiver), $d$ is the separation between receive antenna pair.

[5]The operating velocity range is $(-V_{max}, V_{max})$. Similarly, the operating angle range is $(-\theta_{max}, \theta_{max})$

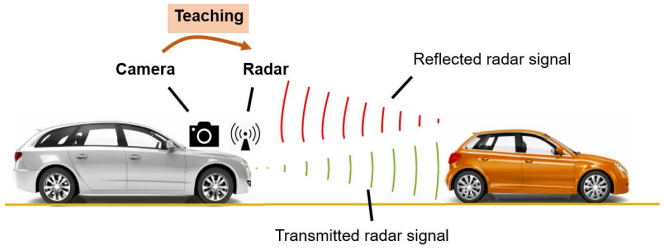



Fig. 2. The vehicular radar-camera system.[6]

Then the predefined features of detected objects, such as SNR, range profile, Doppler spread [10], number of detections, and spatial distance [12] are extracted and combined to determine the classes of objects. Recent advances in deep learning (DL) have promoted novel approaches for automating feature selection. For general DL methods, the algorithmic time complexity can be reduced enormously by implementing suitable pre-processing on the input data. However, too much pre-processing risks losing key information embedded in the raw data; we therefore seek the right trade-off between performance and efficiency.

While several prior works [8], [9], [15]–[17] explore radar object recognition with various input data formats using NN, none has ever combined the spatial and temporal domain information well, i.e., by jointly processing the 3D radar cube *sequences* (from multiple frames). Our fundamental contribution is a *deep learning* network design with 3D radar cube sequences as input that approaches performance upper bound by exploiting all available information (Section IV-A).

However, it is impractical to implement the 4D (3D from radar cube plus time sequences) convolution processing as the resulting computation complexity is unacceptable for real-time perception. Therefore, we propose to combine several lower-dimension (3D) models, which nonetheless exceeds the performance of prior methods with acceptable computation complexity (see Table. VII and VIII). Basically, each 3D radar cube (RVA heatmap) is sliced into 2D images from 3 perspectives, that is, range-angle (RA) heatmap, range-velocity (RV) heatmap, and velocity-angle (VA) heatmap. The RA, RV, and VA heatmap sequences are then processed by three parallel DL models to generate different feature bases, which are fused to make the object recognition decision (see Fig. 5). We name above radar network architecture RAMP-CNN.

Supervised learning methods need a huge amount of training data and corresponding ground truth labels, a challenge for every DL-based approach. In particular, human labeling of radar data is unreliable even in good conditions, in contrast to labeling camera images. To solve this problem, we propose to set up a vehicular radar-camera system as shown in Fig. 2 and 12 to collect the synchronized radar data and camera images for building the UWCR dataset. Cameras are used for teaching radars the locations (range and angle) and classes of objects under good light and weather conditions, which is achieved by implementing the object detection and depth estimation algorithm on captured images (Section IV-C).

[6]Fig. 2 is modified from [18] and the abstract figure is modified from [19].

TABLE I
PARAMETERS AND CONFIGURATIONS OF TI'S AWR1843 FMCW RADAR [4], [8]

| Parameter | Calculation Equation | Configuration | Value |
|---|---|---|---|
| Range resolution ($R_{\text{res}}$) | $R_{\text{res}} = \frac{c_0}{2B} = 0.23$ m | Frequency ($f_{\text{c}}$) | 77 GHz |
| Velocity resolution ($V_{\text{res}}$) | $V_{\text{res}} = \frac{\lambda}{2N_{\text{c}}T_{\text{c}}} = 0.065$ m/s [3] | Sweep Bandwidth ($B$) | 670 MHz |
| Angle resolution ($\theta_{\text{res}}$) | $\theta_{\text{res}} = \frac{\lambda}{N_{\text{Rx}}d\cos\theta} \approx 15°$ [4] | Sweep slope ($S$) | 21 MHz/$\mu$s |
| Max operating range ($R_{\text{max}}$) | $R_{\text{max}} = \frac{f_{\text{s}}c_0}{2S} = 28.5$ m | Sampling frequency ($f_{\text{s}}$) | 4000 Ksps |
| Max operating velocity ($V_{\text{max}}$) [5] | $V_{\text{max}} = \frac{\lambda}{4T_{\text{c}}} = 8.3$ m/s | Num of chirps in one frame ($N_{\text{c}}$) | 255 |
| Max operating angle ($\theta_{\text{max}}$) | $\theta_{\text{max}} = \sin^{-1}\left(\frac{\lambda}{2d}\right) = 90°$ | Num of samples of one chirp ($N_{\text{s}}$) | 128 |
| | | Num of transmitters, receivers | 2, 4 |
| | | Frame rate ($f_{\text{F}}$) | 30 FPS |

To further improve the performance of RAMP-CNN model, we propose the following two approaches to avoid overfitting in the training stage, which have been validated in Ablation study (Section VII-B).

*Radar Data Augmentation Algorithms:* Data augmentation encompasses a suite of techniques that enhance the size and quality of training dataset such that better DL models can be built. Traditional image augmentation algorithms include geometric transformations, color space augmentations, mixing images, etc. In Section V, we propose 4 basic data augmentation operations - flipping, translating, interpolating, and mixing - that work for radar data by accounting for radar imaging physics: energy loss with range and nonuniform angular resolution.

*New Loss Function Design:* The feature basis from the RA, RV, and VA input - named RA, RV, and VA features respectively - are fused within the feature fusion module. When fusing, RA features remain unchanged while RV and VA features are mapped to the range-angle domain. The resulting issue is that the NN that takes the fused features as input may well give more weights to the straightforward and accessible RA features than other velocity-based features, leading to overfitting. Therefore, to push NN to effectively utilize RV and VA features, we add a new term - that only takes RV and VA inputs - to original loss function (Section IV-D).

In summary, the main contributions of this paper are fourfold:

- Design a novel temporal-spatial RAMP-CNN model that jointly processes the 3D radar cube sequences to achieve superior performance than all prior works, as validated by extensive testing with our UWCR dataset.
- The proposed RAMP-CNN model is validated to work robustly under the nighttime, where cameras (and other passive optical sensors) are largely ineffective.
- For training the RAMP-CNN model, we propose and establish a vehicular radar-camera system that uses cameras to teach radars the locations and classes of objects under the good light and weather conditions.
- To avoid overfitting in training stage, we propose the modified data augmentation algorithms suitable for radar data and design a new loss function that pushes the RAMP-CNN model to utilize more velocity-related features.

The rest of this paper is organized as follows. Several relevant prior works are commented in Section II. The principle of radar preprocessing algorithm is introduced in Section III. The RAMP-CNN architecture and radar data augmentation algorithms are presented in Section IV and V. We describe the experiment details including the evaluation results in Section VI and analyze the RAMP-CNN model in Section VII. In the end, we conclude the paper and propose future work.

## II. RELATED WORK

We comment on the relevant prior works [8], [9], [15]–[17], [20] that have attempted radar object classification with various radar data input formats.

Reference [9] uses the short-time Fourier transform (STFT) intensity (heatmap) as the input and implements several existing DL models to extract micro-Doppler patterns from it, with up to 93% recognition accuracy when evaluated for three classes discrimination: car, pedestrian and cyclist. Further, [8] proposes a new framework to pre-process the raw radar data and enhance radar object classification by incorporating not only the micro-Doppler pattern but also the spatial information. However, the above methods are two-stages architectures where the regions of interest (i.e., the locations of objects) need to be found before the classification, which usually takes longer inference time than one-stage methods. Besides, in [8], [9], there are several prepossessing procedures (i.e., CFAR detection, DBSCAN clustering, etc) before feeding the radar input to NN, which may make the information incomplete.

Reference [16] presents a single shot detection and classification system in urban automotive scenarios, which is based on the YOLO system applied to the pre-processed range-Doppler-angle power spectrum with 77 GHz FMCW radar. To feed the 3D radar power spectrum into 2D YOLO network, [16] condenses the angular domain by choosing the maximum. The range-angle domain is the main perspective to observe the objects' reflection ability and shapes such that condensing angular domain is not the best choice. Similarly, [20] also takes the range-Doppler radar data as input and predicts object class with a CNN.

Reference [15] illustrates a DL-based vehicle detection solution that operates on the absolute-valued range-velocity-angle radar tensor. The ability of accuracy vehicle detection in high way scenario mainly relies on recognizing the energy

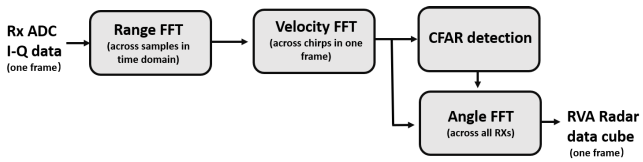



Fig. 3. Basic radar signal processing chain.

distribution on the range and angle dimension (i.e., the contribution of Doppler dimension to detection is small). This may be hard for solving the various object recognition problem we have.

Reference [17] shows a radio object detection network (RODNet) to detect objects purely from the processed radar data in the format of range-angle heatmap sequences. [17] mainly exploits the temporal information behind the change of spatial patterns across frames. However, for each frame, [17] just randomly picks one range-angle heatmap of a chirp signal, which is convenient but gives up the abundant velocity information behind the phase change across chirps.

Besides object classification, a bunch of NN-based radar applications have been attempted, such as human activity classification [21], [22], hand-gesture recognition [23], and the armed and unarmed personnel recognition [24].

## III. Radar Data Preprocessing

An FMCW radar transmits a train of FMCW chirp signals - named a frame - and then mixes the received echo with the local reference (transmitted signal) to yield the intermediate frequency (IF) signal. The IF signal is digitized by Quadrature ADC and then processed by the 3-DFFT algorithm shown in Fig.3. The 3-DFFT algorithm consists of 3 discrete fast Fourier transforms (DFFT) that estimate the spectrum of range, Doppler velocity, and angle respectively. The third FFT (Angle FFT) is performed across receivers at every cell of the range-velocity spectrum (Velocity FFT output). Before that, we implement the CFAR [13] algorithm on range-velocity spectrum and then compensate the Doppler-induced phase change [25] at the locations where CFAR produced detections.

### A. Range Estimation

For the target at range $r$, the resulting beat signal has a frequency $f_{\mathrm{b}} = \frac{S2r}{c_0}$ in the IF band, where $S$ is the slope of a chirp signal, and $c_0$ is speed of light. To estimate the beat frequency, a fast Fourier transform (*Range FFT*) is used to convert the time domain IF signal into the frequency domain; the peaks in the resulting spectrum is used to detect resolved objects. The resolution of FFT-based range estimation is determined by the swept RF bandwidth $B$ of the FMCW system [13], i.e., $R_{\mathrm{res}} = \frac{c_0}{2B}$. In our experiments with the TI system, the FMCW signal is configured for 670 MHz swept bandwidth, and the expected range resolution is 0.23 m.

### B. Velocity Estimation

Any radial object motion $\Delta r$ (shown in Fig. 4) relative to the radar between consecutive chirps will cause a frequency shift $\Delta f_{\mathrm{b}} = \frac{2S\Delta r}{c_0}$ as well as a phase shift $\Delta\phi_v = 2\pi f_{\mathrm{c}} \frac{2\Delta r}{c_0} = \frac{4\pi v T_{\mathrm{c}}}{\lambda}$ in the beat signal [6], [8], where $f_{\mathrm{c}}$ is the carrier frequency, $v$ is the object velocity, $T_{\mathrm{c}}$ is the chirp period, and $\lambda$ is the wavelength. Compared to the beat frequency shift, the phase shift is more sensitive to the object movement [6]. Hence, it is common to execute a fast Fourier transform (*Velocity FFT*) across the chirps to estimate phase shift and then transform it to velocity estimation. The velocity resolution of this method is given by: $V_{\mathrm{res}} = \frac{\lambda}{2N_{\mathrm{c}}T_{\mathrm{c}}}$ [6], where $N_{\mathrm{c}}$ is the number of chirps in one frame. The expected velocity resolution is 0.065 m/s given the configuration $N_{\mathrm{c}} = 255$, and $T_{\mathrm{c}} = 120\,\mu$s.

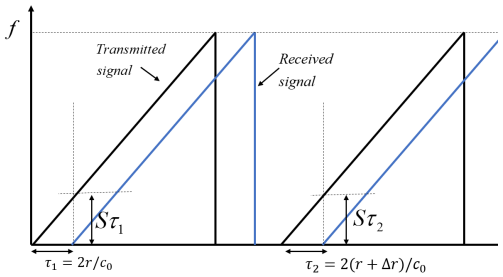



Fig. 4. The illustrations of the range and velocity measurement.

### C. Angle Estimation

Angle estimation is conducted via processing the signal at a receiver array composed of multiple elements. The return from a target located at far field and angle $\theta$ results in the steering vector $\boldsymbol{a}_{\mathrm{ULA}}(\theta) = [1, e^{-j2\pi d \sin\theta/\lambda}, \cdots, e^{-j2\pi(N_{\mathrm{Rx}}-1)d\sin\theta/\lambda}]^{\mathrm{T}}$ as the uniform linear array output [26], where $d$ denotes the inter-element distance. Hence a fast Fourier transform across Rx elements (*Angle FFT*) can easily resolve objects with different arrival angles $\theta$ [7], [8].

TI chips [27] provide the MIMO radar capability that forms a larger virtual array with orthogonal transmit waveforms [28], which also enables a greater degree of freedom capability and better angle resolution [7]. We adopted the TDM-MIMO [7], [29] configuration with 2 Tx and 4 Rx for all collected data such that the resulting virtual array consists of 8 elements.

For non-stationary targets, the motion-induced phase errors should be compensated on the virtual antennas (elements corresponding to the second Tx in case of TDM-MIMO) before the Angle FFT. According to [25], these virtual elements are corrected via rotating phase by $\frac{\Delta\phi_v}{2}$, half of the estimated Doppler phase shift, where $v$ is obtained from the CFAR detection results on the range-velocity spectrum.

The angle resolution for FFT processing is known to be $\theta_{\mathrm{res}} = \frac{\lambda}{N_{\mathrm{Rx}} d\cos\theta}$ [6]. For boresight $\theta = 0°$, $N_{\mathrm{Rx}} = 8$, and $d = \frac{\lambda}{2}$, the angle resolution is approximated to 15°.

## IV. RAMP-CNN Model: A Convolutional Neural Network for Radar Data

### A. 3-Perspectives Autoencoders Design

As shown in Fig. 5, the main body of the RAMP-CNN architecture is composed of 3 convolutional autoencoders. These autoencoders extract features from the heatmap

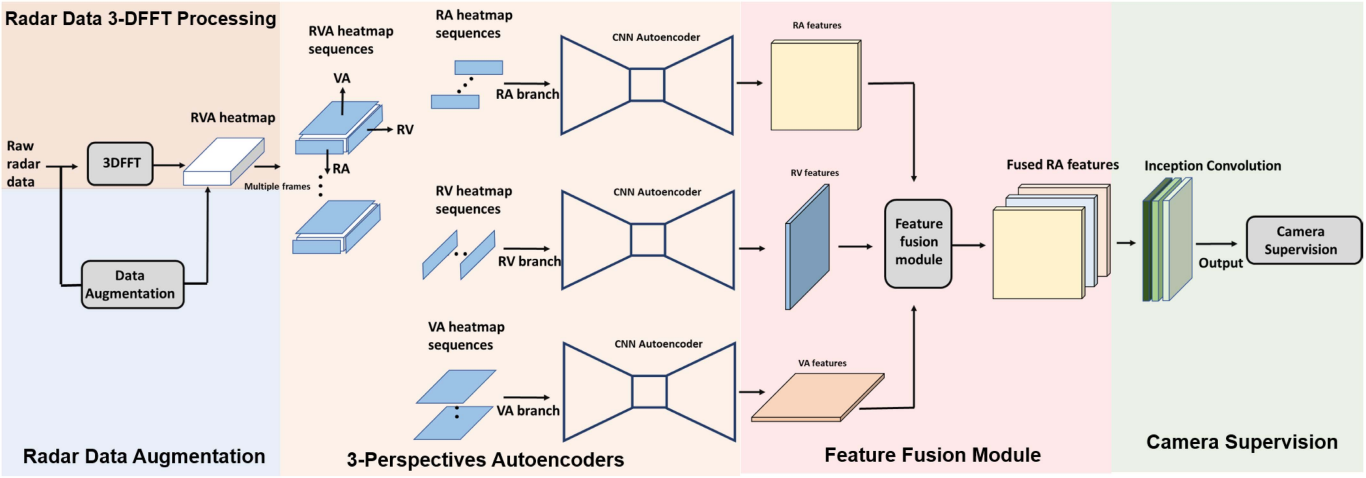



Fig. 5. The architecture of RAMP-CNN model.

sequences of different perspectives - that is, RA, RV, and VA respectively.

***Why Convolutional Autoencoder:*** CNNs are known for excellent feature extractor in some tasks, e.g, object detection, and segmentation. The convolutional autoencoders (CAE) - consisting of an encoder and a decoder - render a compact feature representation of the input, by learning the optimal filters that minimize the reconstruction error [30], [31]. The output feature representation/basis is the same size grid as input and the cell value of grid is the feature strength. Each (input) image-plane pixel location maps to multiple feature grid indices. Thus, operations such as weighted-sum followed by a suitable non-linearity on the feature grid cells can be used to determine the presence of particular-type objects at one location. The parameters for such operations (i.e., weights and bias) can be trained by suitable NN iteration.

***The Physical Significance of Network Design :*** The first CAE processes the complex-valued RA heatmap sequences with 3D *conv* layers and *transposed conv* layers. Similar to [17], we pick one RA heatmap from each frame to form the heatmap sequences, and the singled out RA heatmap is obtained by computing Range FFT and Angle FFT at an arbitrarily selected chirp.[7] For the RA heatmap sequences input, those 3D convolution operations take advantage of not only the object's spatial patterns in a single frame but also the temporal information behind the change of spatial patterns across frames. Some aspects of spatial patterns - like the distribution of reflection intensity - directly contribute to object recognition, e.g., larger objects (vehicles) contain more stronger-reflectors than small objects (humans).

As the RA heatmap input is with the complex-valued format, the temporal change of spatial patterns across multiple frames can be expressed as the change in both amplitude and phase. Particularly, the phase change of mmW signal along time is sensitive to the object movement, e.g. 1mm position movement results in phase shift $\Delta\phi = \pi$ for 77 GHz radar.

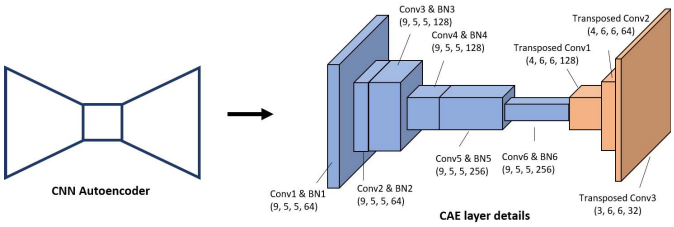



Fig. 6. Details of Convolutional AutoEncoder (CAE), consist of six 3D *conv* layers and three 3D *transposed conv* layers.

While the sampling rate of RA heatmap input is a bit lower (30 FPS), we still believe the embedded phase shift would provide additional benefit compared to the amplitude-only input.

The second and third CAE process the absolute-valued RV and VA heatmap sequences respectively.[8] The RV and VA heatmaps are calculated from the original RVA heatmap by summing the power over the omitted dimension. What two CAEs have in common is: in single RV or VA heatmap, they extract features from the distribution of range-velocity or velocity-angle cells; while across multiple heatmaps, they extract object's movement patterns from the change of radial velocity with time. These two CAEs essentially utilize the abundant velocity-based information behind the phase change across chirps within each frame, which is the biggest difference from [17].

To illustrate, different classes of moving objects exhibit different movement patterns. From Fig. 7, we can visualize the movement patterns of a pedestrian as the change of radial velocity with time. The velocity versus time relationship is also known as the STFT heatmap [8] that highlights specific micro-Doppler signatures of human gait.

***Network Details:*** We adopt the 3D Convolutional-De-Convolutional [30] (shown in Fig. 6) model as our CAE, which is effective in summarizing spatio-temporal patterns from raw data into high-level semantics.

[7]For the range bin where there exists CFAR detections, we pick its maximum-intensity velocity and use it to compensate the Doppler-induced phase error on virtual receiver elements before Angle FFT.

[8]We adopt the absolute-valued RV and VA heatmap here, since the phase change we are interested in have been preprocessed with Velocity FFT and been represented in the Doppler domain.

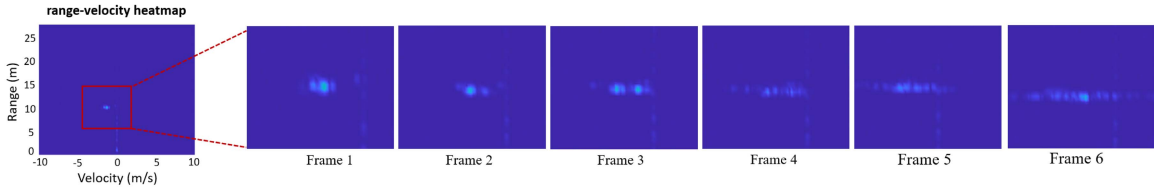



Fig. 7. Visualization of a pedestrian's movement patterns: we zoom in partial regions of 6 range-velocity heatmaps of a moving pedestrian. The x-axis is velocity and the y-axis is the range. We can observe that this pedestrian has a small location movement over 6 frames but a big change of velocity patterns.

Each CAE includes six 3D *conv* layers and three 3D *transposed conv* layers. All 3D *conv* layers are followed by a *bn* (batch-normalization) layer and the *ReLU* activation function. The first two 3D *transposed conv* layers are followed by the *PReLU* activation function. The layer details (including parameter selection) of CAE are presented in Fig. 6. For illustration, the first blue cuboid part in Fig. 6 represents the 3D *conv* layer and a *bn* layer. The kernel size of the 3D *conv* layer is (9, 5, 5) and the number of output feature channels is 64.

To preserve the phase information in RA heatmap input, we represent complex-valued heatmap by two real-valued channels that store the real part and imaginary part respectively in the first CAE following [32]. While for RV and VA heatmap inputs, it suffices to only keep the absolute value and use the one-channel representation.

## B. Feature Fusion Module

The feature basis extracted from RA, RV, VA inputs all support the final classification decision. Our final output is expressed as an image in RA domain, which means the RA feature can be directly inputted to the network to obtain the corresponding RA-format output. The key issue is how to further use RV and VA features to support an improved final classification. This is similar to initial human perception using the visual sensor (eyes) supported by supplementary sense organs (ears, nose) for final determination. A person with impaired eyesight will rely more on other sensors, e.g. acquire initial angle information/feature via the ear.[9]

The above analogy applied to radar processing suggests how to use the VA feature - that provides good azimuth angle information and no range information. As shown in Fig. 8, VA feature is condensed along the velocity dimension by summing, then the condensed vector is replicated in the missing dimension - range. Similarly, we replicate the RV feature along the angle dimension. Thereafter, we concatenate all features along the channel dimension and input them to the deep network for classification decisions.

*Convolution Layers After Feature Fusion Module:* There are two *conv* layers that take the fused features as input and make recognition decisions: one 3D *inception* layer, and one ordinary 3D *conv* layers with kernel size (3, 3, 3). Note that the ordinary *conv* layer operates on time, range and angle dimension, while the *inception* layer operates on channel, range and angle dimension of fused features. To avoid collapsing the time dimension on *inception* layer, we repeat the operation on each timestamp and concatenate all inception results along time dimension.

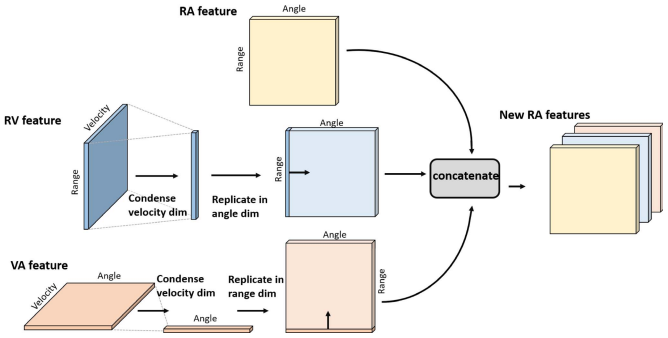



Fig. 8. The structure of feature fusion module.

The 3D *inception* layer includes 3 convolution kernels: (3, 5, 5), (3, 3, 3), (3, 1, 21). The first two kernels allow NN to take advantage of multi-level feature extraction, i.e. it extracts both general $5\times5$ size feature and local $3\times3$ size features. The last kernel with dilation 6 is used to push the NN to observe a larger area in angle - hence to solve the false alarm problem on the side-lobes.[10] We make it the dilated convolution - with angular kernel size 21 and dilation 6 - to cover almost all angle cells, as well as to reduce complexity.

## C. Camera Supervision

The established radar-camera system shown in Fig. 12 generates synchronized camera images and raw radar data. As shown in Fig. 9, we apply the object detection [33] and depth estimation [34], [35] algorithm on captured camera images to obtain the locations (i.e., range and azimuth angle) and classes of all objects of interest in the scene. Under the good light and weather conditions, the obtained information from cameras is treated as ground truth for supervising the output of RAMP-CNN. Note that camera assisted radar learning only happens in the training stage, while in testing, radar acts independently.

To ease the training burden, we use the center keypoint to represent the existence of objects following [36]. For each ground truth center point $\mathbf{p}$ with location $(p_r, p_\theta)$, class id $p_c$

[9]The ear does not provide good range localization, and hence suggests an equal probability prior to range.

[10]The side-lobe in radar heatmap is easy to be recognized as objects of a certain class. This is because the convolution-kernel operators of CAEs do not force each feature to be global (i.e., to span the entire visual field) [31].

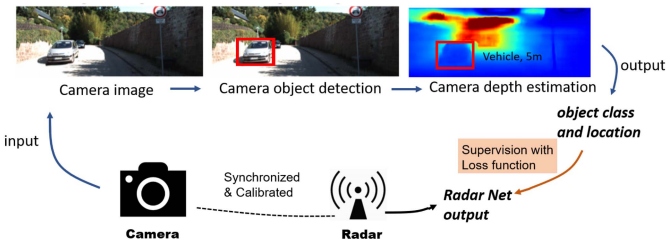



Fig. 9. The framework of camera teaching radar for the training stage.

and frame id $p_t$, we compute its Gaussian representation:

$$Y_{t,r,\theta,c} = \begin{cases} \exp(-\dfrac{(r-p_r)^2+(\theta-p_\theta)^2}{2\sigma_p^2}) & \text{if } c = p_c \text{ and } t = p_t \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\sigma_p$ is an object size-adaptive standard deviation. We then splat all ground truth center points onto $Y \in [0,1]^{D\times W\times H\times C}$, and take the element-wise maximum if two Gaussians of the same class and same frame overlap.[11] $Y$ is used as the ground truth in loss function.

### D. Loss Function for All-Perspectives Learning

Let $X_{\mathrm{RA}}, X_{\mathrm{RV}}, X_{\mathrm{VA}}$ be the RA, RV and VA input heatmap sequences, the aim of RAMP-CNN model is to predict center-point heatmap sequences $\hat{Y} \in [0,1]^{D\times W\times H\times C}$ in RA domain, where $\hat{Y}_{t,r,\theta,c} = 1$ corresponds to a detected center point at range $r$, azimuth angle $\theta$, frame $t$ and class $c$, while $\hat{Y}_{t,r,\theta,c} = 0$ represents background. The prediction $\hat{Y}$ includes a map for every frame time. The center point types of each map include $C = 3$ classes of objects: pedestrian, cyclist, and car.

For the prediction $\hat{Y}$ and ground truth $Y$, the training objective is a modified penalty-reduced pixelwise logistic regression with focal loss [37], [38]:

$$L_{\hat{Y}Y} = \frac{-1}{N_{\mathrm{obj}}} \sum_t \sum_r \sum_\theta \sum_c \times \begin{cases} \kappa(1-\hat{Y}_{t,r,\theta,c})^\alpha \log(\hat{Y}_{t,r,\theta,c}) & \text{if } Y_{t,r,\theta,c} = 1 \\ \kappa(1-Y_{t,r,\theta,c})^\beta (\hat{Y}_{t,r,\theta,c})^\alpha & \\ \quad \times \log(1-\hat{Y}_{t,r,\theta,c}) & \text{if } Y_{t,r,\theta,c} = 0 \\ & \text{and } Y_{t,r,\theta,\bar{c}} > 0 \\ (1-Y_{t,r,\theta,c})^\beta (\hat{Y}_{t,r,\theta,c})^\alpha & \\ \quad \times \log(1-\hat{Y}_{t,r,\theta,c}) & \text{otherwise} \end{cases} \tag{2}$$

where $\alpha$ and $\beta$ are hyper-parameters of focal loss [38], and $N_{\mathrm{obj}}$ is the number of objects in ground truth. The normalization by $N_{\mathrm{obj}}$ is chosen as to normalize all positive focal loss instances to 1. Compared to [38], we add a new scalar hyper-parameter $\kappa$, which put more loss/focus at the region where objects exist to shorten the training time. In this paper, we choose $\kappa = 4$ and following [37], we use $\alpha = 2$ and $\beta = 4$ in all our experiments.

[11]The symbols $D$, $W$, $H$ and $C$ here represent the size of $Y$ on time, range, azimuth angle and class dimension respectively.

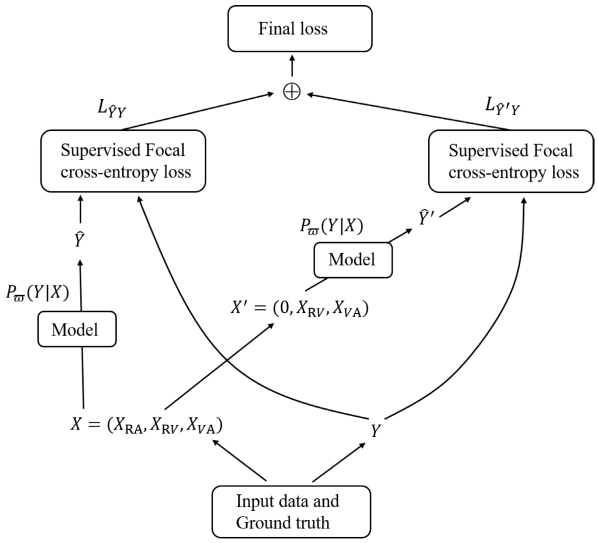



Fig. 10. The loss function for all-perspectives supervision.

When designing the loss function, we also take account of the fact that NN may well give more weights to the straightforward and accessible RA features than other velocity-based features, leading to overfitting. This point can be illustrated with the above human perception example again. A person with unimpaired eyesight will not rely much on the other sensors (ears, nose), thus resulting in weaker supplementary function compared to a person with impaired eyesight.[12]

The above analogy applied to loss function design suggests how to make NN fully utilize all three perspectives (RA, RV and VA) and particularly enhance the supplementary function provided by RV and VA perspective. We add a new loss constraint $L_{\hat{Y}'Y}$ besides the original loss term $L_{\hat{Y}Y}$ mentioned above. To obtain $L_{\hat{Y}'Y}$, we set $X_{\mathrm{RA}} = 0$ in the new loss term, i.e. we input $X = (0, X_{\mathrm{RV}}, X_{\mathrm{VA}})$ to the NN $P_w(Y|X)$ again such that getting the new prediction $\hat{Y}'$. Then $\hat{Y}'$ is also supervised by ground truth $Y$ with (2) to obtain $L_{\hat{Y}'Y}$.

The final loss is the weighted sum of two terms:

$$L_{loss} = L_{\hat{Y}Y} + \gamma L_{\hat{Y}'Y} \tag{3}$$

where $\gamma$ is the hyper-parameter to balance two terms, chosen to be $\gamma = 0.5$ in this paper.

## V. Radar Data Augmentation Algorithms

Many image data augmentation algorithms have been proposed to increase the amount of *relevant* data and prevent the NN from overfitting, thus essentially boosting overall performance. However, most of the existing data augmentation algorithms cannot be applied to the radar data because of a few key differences from traditional RGB images - complex inputs, energy loss with range, and nonuniform resolution in the angular domain. In this section, we focus on 4 basic data augmentation operations and explain how to apply them to radar data: flipping, translating, interpolating, and mixing.

[12]To train this supplementary function for a non-disabled person, it is better to create a situation where eyes are not working, e.g., blindfolding

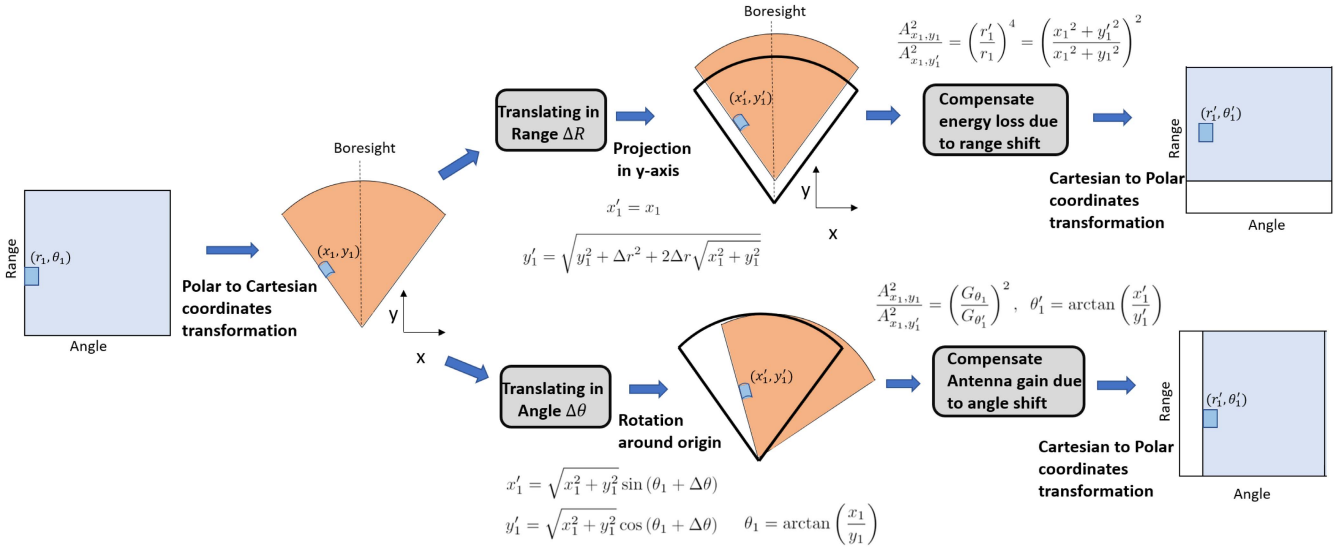



Fig. 11. The design of translating in range and angle.

***Horizontal Flipping***: The horizontal flipping operation will swap the left and right part of the processed 3D radar cube along azimuth angle dimension. This operation can be applied to radar data directly as to images because radar has symmetric property (resolution and antenna gain) in the angular domain.

***Translating in Range***: With the translating in range operation, we will do a pre-defined range shift $\Delta r$ for all objects in RA domain. As shown in Fig. 11, for the first step, we transform the polar-coordinates[13] $(r, \theta)$ radar data into the uniform Cartesian-coordinates $(x, y)$ radar data with the well-known projection: $x = r\sin\theta$, $y = r\cos\theta$. This relation is nonlinear, therefore the new cell in Cartesian coordinates is not rectangular, interpolation and down-sampling operations are needed to sample the Cartesian plane uniformly.

Assume that there is a target at location $(r_1, \theta_1)$ with corresponding Cartesian-coordinates $(x_1, y_1)$, and the distance from target to radar boresight is fixed even with range shift. Then, the range shift $\Delta r$ for this target is equivalent to shift y-axis while keeping the x-axis fixed in the Cartesian plane. This maps the previous cell $(x_1, y_1)$ to the new cell $(x_1', y_1')$ with relation: $x_1' = x_1$, $y_1' = \sqrt{y_1^2 + \Delta r^2 + 2\Delta r r_1}$, where $r_1 = \sqrt{x_1^2 + y_1^2}$, and $r_1' = \sqrt{x_1'^2 + y_1'^2} = r_1 + \Delta r$.

Without considering the Doppler phase change, the above translating in range operations can be approached by shifting $\lfloor -\frac{2M_r S\Delta r}{c_0 f_s} \rceil$ cells[14] in the polar-coordinates range spectrum/profile and then changing the phase across antennas:

$$\frac{\phi_{r_1',q}}{\phi_{r_1,q}} = \frac{qd\sin\theta'}{qd\sin\theta} = \frac{r_1}{r_1 + \Delta r} \tag{4}$$

where $M_r$ is the number of points for Range FFT, $\phi_{r_1,q}$ is the original phase of $q^{th}$ Rx for the target at $r_1$, $\phi_{r_1',q}$ is the phase of $q^{th}$ Rx after projecting the target to $r_1'$. The phase of Rx changes as the azimuth angle of target changes when applying the translating in range operation.

Meanwhile, the energy loss due to range shift needs to be compensated according to the radar range equation [13]:

$$\frac{A_{x_1,y_1}}{A_{x_1',y_1'}} = \left(\frac{r_1'}{r_1}\right)^2 = \left(\frac{r_1 + \Delta r}{r_1}\right)^2 \tag{5}$$

where $A$ is the signal amplitude.

***Translating in Angle***: With the translating in angle operation, we will do a pre-defined angle shift $\Delta\theta$ for all objects in RA domain. The angle shift in polar plane is equivalent to the rotation around origin in Cartesian plane. Therefore, as shown in Fig. 11, after transforming to the Cartesian-plane data, we use the following relation to map the target in cell $(x_1, y_1)$ to the new cell $(x_1', y_1')$: $x_1' = r_1\sin(\theta_1 + \Delta\theta)$, $y_1' = r_1\cos(\theta_1 + \Delta\theta)$, where $\theta_1 = \arctan(\frac{x_1}{y_1})$, and $\theta_1' = \arctan(\frac{x_1'}{y_1'}) = \theta_1 + \Delta\theta$.

If there is no more than one target in a range bin, we can approximate above translating in angle operation by shifting $\lfloor \frac{M_\theta d(\sin\theta_1 - \sin\theta_1')}{\lambda} \rceil$ cells in the polar-coordinates angular spectrum, where $M_\theta$ is the number of points for Angle FFT.

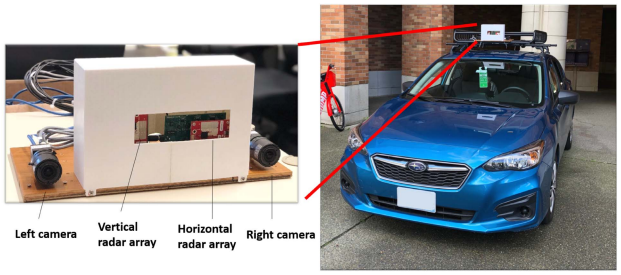
Based on radar range equation [13], we also need to compensate for the antenna gain $(G)$ loss due to angle shift:

$$\frac{A_{x_1,y_1}}{A_{x_1',y_1'}} = \frac{G_{\theta_1}}{G_{\theta_1'}} \tag{6}$$

***Interpolating***: The interpolating operation is to fill in the blanks (the white stripes in the last two images of Fig. 11) left by the translating operation. We utilize the environment noise for interpolating in order to imitate the situation where there is no object in the blank area. The environment noise samples are obtained by sorting all data of the 3D radar cube with amplitude and then taking the bottom (smallest) 5% of it.

[13]The processed 3D radar cube is represented as the polar-coordinates format in RA domain, which is non-uniform for the representation of objects.

[14]Left shift when the shifting cell number is positive, otherwise right shift.




(a) (b)

Fig. 12. Radar-camera data capture platform: (a) This platform consists of 2 FLIR cameras and two perpendicular radars from TI - the right radar is with the 1D horizontal antenna array, and the left one is with the 1D vertical antenna array. (b) Data capture platform mounted on a vehicle with front view.


Fig. 13. 8 scenario examples in the collected UWCR dataset: row 1, 3 are the camera images; row 2, 4 are the corresponding radar range-azimuth angle heatmaps.

***Mixing***: The mixing operation is to add up two 3D radar cubes, which could remain unchanged or could have been done with other augmentation techniques - like flipping and translating.

## VI. Experiment

### A. UW Camera-Radar (CR) Dataset

A large camera image and radar raw data (I-Q samples post demodulated at the receiver) dataset for various objects have been collected for multiple scenarios - parking lot, curbside, campus road, city road, freeway, etc. - by a vehicle-mounted platform that is driven (see Fig. 12(b)). In particular, significant effort was placed in collecting data for situations where cameras are largely ineffective, i.e. under challenging light conditions. We show the camera images and radar range-angle heatmaps of several scenario examples in our UWCR dataset at Fig. 13.

The data collection platform shown in Fig. 12(a) consists of 2 FLIR cameras (left and right) and two TI AWR1843 EVM radars [27]. Two radar EVM boards are placed to form a '2D' antenna array system[15] that can provide more abundant object information. We place one radar array horizontally and the other one vertically to collect the data from both range-azimuth angle domain and range-elevation angle domain. We only use the radar data from horizontal array so far, and we will incorporate the vertical array data into the future work.

TABLE II
Dataset Distribution for Training and Test

| | Augmented data | Training set [14] | Testing set |
|---|---|---|---|
| Frames | 26462 | 67198 | 25098 |
| Included ped. cyc. car [15] | 53275, 18840, 18731 | 55203, 24742, 46446 | 14541, 7607, 7471 |
| | Parking Lot | Curbside | On-road | nighttime |
| Frames | 9900 | 7200 | 4398 | 3600 |
| Included ped. cyc. car | 5750, 4501, 2700 | 3581, 1172, 2039 | 5210, 1934, 2732 | - [16] |

As discussed in Section IV-C, the binocular cameras are synchronized with radars, and they can provide the location and class of semantic objects after we implement the Mask R-CNN detection model [33] and unsupervised depth estimation model [34], [35] on the captured camera images. The semantic object detection results and depth estimation results generated from cameras are manually calibrated and then saved as the requisite ground truth for the following training and evaluation.

### B. Data Processing

*3-DFFT Preprocessing:* We implement the 3-DFFT [8] algorithm on the raw I-Q radar data samples to obtain the RVA heatmap sequences. The FFT on range, angle, velocity dimensions are all 128 points. We choose input frame number $M = 16$. Therefore, the size of the preprocessed input data is $128 \times 128 \times 128 \times 16$ that corresponds to [range bins $\times$ angle bins $\times$ velocity bins $\times$ frame number].

*Data Augmentation:* We implement the proposed data augmentation algorithms on the processed RVA heatmap sequences, which include flipping, range translating, angle translating, and mixing the input data. The augmented data is saved locally and **only** used as part of the training data to avoid overfitting.

*Training and Test Sets:* We partition our UWCR dataset into the training set and test set. Any nighttime scenario data cannot be used in the training set as the corresponding low-light camera images cannot provide the ground truth labels. So all nighttime data is placed into the test set for qualitative performance evaluation only, i.e, the performance of nighttime data isn't evaluated with numerical metrics.

The data distribution for the training, augmented, and test set are shown in Table. II (row 1-3). Note that the training set in table doesn't count augmented data, and the whole training data are the collection of training set and augmented data. The test set is divided into 4 scenarios: parking lot, curbside, on-road, and nighttime. Table. II also shows their data distribution.

[15]Here, '2D' is equivalent to two perpendicular 1-D arrays.
[16]The training set here doesn't count the augmented data.
[17]Pedestrian (ped), cyclist (cyc).
[18]The nighttime data are not labeled, so we don't count the number of different classes of objects here.

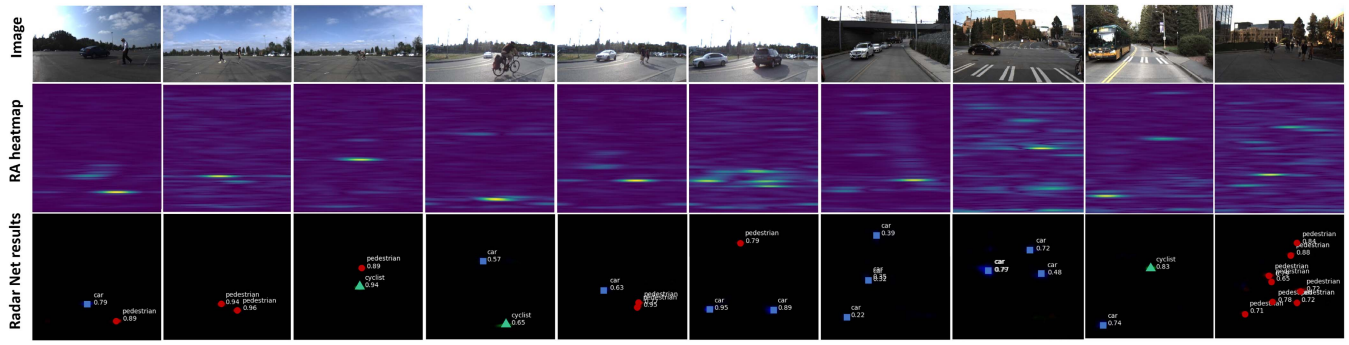



Fig. 14. 10 test examples from the parking lot scenario, curbside scenario, and on-road scenario: Column 1-3 are the parking lot scenario; Column 4-6 are the curbside scenario; Column 7-10 are the on-road scenario. For each column, the top row image is the synchronized camera image for visualization, the second row image is the corresponding radar RA heatmap, and the bottom row image is the visualization of the RAMP-CNN model results.

TABLE III
PERFORMANCE COMPARISON BETWEEN DIFFERENT MODELS

| Model | Overall | | Parking Lot Scenario | | Curbside Scenario | | On-road Scenario | |
|---|---|---|---|---|---|---|---|---|
| | AP | AR | AP | AR | AP | AR | AP | AR |
| CDMC [8] | 30.55% | 54.79% | 65.74% | 76.56% | 28.68% | 53.93% | 4.88% | 25.76% |
| RODNet-HG [17] | 71.84% | 76.03% | 93.87% | 95.36% | 61.65% | 70.09% | 41.97% | 53.04% |
| RODNet-CDC [17] | 71.46% | 78.15% | 92.72% | 95.07% | 64.01% | 71.97% | 46.52% | 58.61% |
| Prop. RAMP-CNN | **81.23%** | **84.25%** | **97.38%** | **98.37%** | **79.25%** | **84.21%** | **57.07%** | **64.85%** |

## C. Experiments

***Baselines:*** We compare the RAMP-CNN model with RODNet-CDC [17], RODNet-HG [17], the state-of-art radar object classification models, as well as the CDMC [8], a model that fully exploits the micro-Doppler signatures of moving objects.

***Training:*** We train the RAMP-CNN model and retrain the RODNet-CDC, RODNet-HG, CDMC following below details.

*a) Proposed RAMP-CNN model:* We train the RAMP-CNN model on complete training set (includes augmented data) with Cyclic learning rate (minimum $5 \times 10^{-6}$, maximum $5 \times 10^{-5}$, and cycle duration 860 iterations) [39], batch size 5 for the first 10 epochs. Then we continue to train the RAMP-CNN model with Step learning rate (starts from $5 \times 10^{-6}$, and decays 0.2 every 5 epochs), batch size 4 for the next 24 epochs. We use the Adam gradient descent optimizer [40] and 1 TITAN RTX GPU for the training of all experiments. To verify the capability of the proposed radar data augmentation algorithms, we also train a new RAMP-CNN model following the same procedures as above, but without augmented data.

*b) RODNet-CDC and RODNet-HG:* We train the RODNet-CDC and RODNet-HG model with Cyclic learning rate (same as above), batch size 4 for 10 epochs, and then train them with Step learning rate (same as above), batch size 4 for the following 22 epochs. The gradient descent optimizer is Adam [40]. The loss function for RODNet-CDC is the Minimum Square Error (MSE) provided by PyTorch, and the loss function for RODNet-HG is Cross Entropy. Note that the training set for RODNet-CDC, RODNet-HG and CDMC [8] model doesn't include the augmented data.

*c) CDMC:* Following [8], we generated about $1.2 \times 10^5$ concatenated STFT heatmaps in total from the training set. By feeding the training STFT heatmaps to the VGG16 classifier, we trained the model from scratch with the batch size 5, learning rate $1 \times 10^{-4}$ for the first 10 epochs, and learning rate $1 \times 10^{-5}$ for the next 10 epochs. The gradient descent optimizer is also Adam [40] and the loss function is the Cross Entropy provided by TensorFlow.

## D. Evaluation Metrics

We use the average precision (AP) and average recall (AR) to evaluate performance, which are calculated from the true positive, false positive, and false negative rates in (7). Here, true positive (tp) represents correctly located and classified instances, false positive (fp) represents the false alarm, false negative (fn) represents the missed detection and/or incorrectly classified instance.

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}, \quad \text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{7}$$

We adopt the CFAR [13] and threshold 0.2 to filter out the target center points from prediction $\hat{Y}$. Whether the targets are correctly located is determined by a object size-adaptive distance threshold, i.e., if the distance between the prediction and ground truth is smaller than the threshold, we assume the prediction is correctly located.

## E. Evaluation Results

We test the RAMP-CNN under 4 different scenarios: parking lot, curbside, on-road, and nighttime (See Fig. 14 and Fig. 15 for testing examples). The parking lot scenario is manually controlled to have moving and/or static objects at a

clear parking lot. For the curbside scenario, we set up the data collection platform on the curbside and then record multiple moving objects on a clear road. The on-road scenario is more like an autonomous driving scenario where we drive around and record all objects on the city road. For the nighttime scenario, we collect the data under challenging light conditions where cameras are largely ineffective. The testing results of RAMP-CNN and other baselines are shown in Table. III.

*The Parking Lot Scenario:* The parking lot test set has the data of 9900 frames which contain 5750 pedestrians, 4501 cyclists, and 2700 cars. The parking lot scenario is relatively easy for the object recognition task as the background is clean and the objects are few. The RAMP-CNN model achieves **nearly perfect** performance (97.38% AP, 98.37% AR), and beat all prior works. We show 3 test examples in Fig. 14.

*The Curbside Scenario:* The curbside test set has the data of 7200 frames which contain 3581 pedestrians, 1172 cyclists, and 2039 cars. This scenario allows multiple objects to appear at the same time and some of them to be close, so that it is harder than parking lot scenario. The AP (79.25%) and AR (84.21%) of RAMP-CNN model have **around 15% improvement** over the best results of prior work - RODNet-CDC model (64.01% AP, 71.97% AR). We show 3 test examples in Fig. 14.

*The On-Road Scenario:* The on-road test set has the data of 4398 frames which contain 5210 pedestrians, 1934 cyclists, and 2732 cars. This test set is collected from the city-road driving experiments which include several challenging situations, e.g., the strong reflections from the environment, a large number of cars in the field of view, crowded pedestrians, etc. The RAMP-CNN model obtains **10% improvement in AP** and **6% improvement in AR** over the RODNet-CDC baseline. We show 3 test examples in Fig. 14.

*The Nighttime Scenario:* We test the RAMP-CNN under nighttime to support the objective of this paper - advance the cause of radar as a low-cost substitute for optical sensors that fail under such severe conditions. As shown in Fig. 15, the RAMP-CNN model performs as well as under the daytime scenario, i.e. radar is impervious/robust to sunlight change. As it is hard to implement the ground truth labeling on nighttime set, we don't numerically evaluate the performance here.

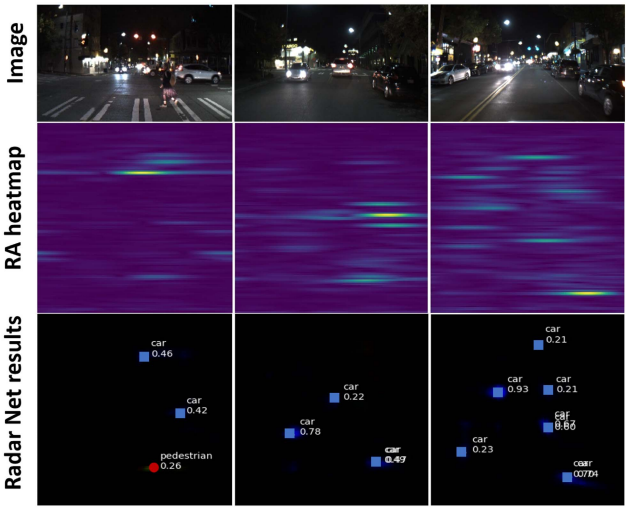



Fig. 15. 3 test examples from the nighttime scenario. The arrangement is same as Fig. 14.

TABLE IV
DATASET DISTRIBUTION FOR STATIC OBJECT SCENARIO AND MOVING OBJECT SCENARIO

| | Static Object Scenario | Moving Object Scenario |
|---|---|---|
| Frames | 3600 | 7200 |
| Included ped. cyc. car | 1250, 1800, 1800 | 2883, 1083, 1702 |

TABLE V
THE PERFORMANCE OF RAMP-CNN MODEL FOR STATIC OBJECT SCENARIO AND MOVING OBJECT SCENARIO

| Model | Static Object Scenario | | Moving Object Scenario | |
|---|---|---|---|---|
| | AP | AR | AP | AR |
| CDMC [8] | 32.34% | 50.73% | 28.68% | 53.93% |
| RODNet-HG [17] | 56.26% | 61.36% | 61.68% | 70.11% |
| RODNet-CDC [17] | 65.02% | 69.91% | 64.01% | 71.97% |
| RAMP-CNN | **67.58%** | **70.27%** | **79.25%** | **84.21%** |

## VII. ANALYSIS AND ABLATION STUDY

### A. Impact of Adding Temporal Information

Compared to prior works, the proposed RAMP-CNN model fully exploits the temporal information behind the chirps within one frame, as well as the change of spatial information (range-angle info.) across frames; hence we expect it to essentially achieve performance improvements for moving objects. To verify this, we choose a part of the data from the parking lot and curbside scenario, and redivide them into the static object set and moving object set. The distribution of these two sets is shown in Table. IV.

We evaluate the performance of RAMP-CNN, CDMC, RODNet-HG, and RODNet-CDC model on the static object set and moving object set respectively. From the evaluation results (Table. V), we know that for static object scenario, the performance of RAMP-CNN (AP around 67%, AR around 70%) in the same level with RODNet-CDC, but much better than RODNet-HG model; for moving object scenario, there is a performance gap (about 15% AP and 13% AR) between RAMP-CNN model (AP 80%, AR 84%) and other baselines (AP around 65%, AR around 71%). The results above verify that the added temporal information in RAMP-CNN model is helpful for the recognition with moving objects.

### B. Ablation Study

An ablation study refers to removing some "part/module" of the model or algorithm, and seeing how that affects performance. In this section, we will study the contribution of two parts of RAMP-CNN model - data augmentation and new training loss - to the performance.

Following the training procedure mentioned in Section VI-C, we train two 'incomplete' RAMP-CNN

TABLE VI
ABLATION STUDY

| Model | Data Augmentation | New training loss | Overall | | Parking lot scenario | | Curbside scenario | | On-road scenario | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | AP | AR | AP | AR | AP | AR | AP | AR |
| RAMP-CNN | | ✓ | 76.78% | 81.39% | 95.91% | 97.21% | 66.47% | 75.83% | 52.79% | 62.66% |
| RAMP-CNN | ✓ | | 76.93% | 81.41% | 93.90% | 95.46% | 75.52% | 81.81% | 54.10% | 61.86% |
| RAMP-CNN | ✓ | ✓ | **81.23%** | **84.25%** | **97.38%** | **98.37%** | **79.25%** | **84.21%** | **57.07%** | **64.85%** |

models - one removes the data augmentation part, the other one replaces the proposed training loss with the ordinary focal loss [37], [38]. The trained models are all evaluated on 4 test sets mentioned above for comparison with the performance of the 'complete' RAMP-CNN model (presented in Table. III). The experiment results of the ablation study are shown in Table. VI.

The experiments between the RAMP-CNN model with and without data augmentation confirm that the proposed augmentation algorithms help to boost model performance by avoiding overfitting. For illustration, the data augmentation algorithms make RAMP-CNN get 12% AP improvement and 8% AR improvement in the curbside scenario, and get 4% AP improvement and 2% AR improvement in the on-road scenario.

The experiments between the RAMP-CNN model with and without the proposed training loss function verify that the proposed training loss helps improve performance as well by pushing the RAMP-CNN model to learn more Doppler-related features. Specifically, RAMP-CNN obtains around 4% AP improvement and 3% AR improvement in both parking lot scenario and curbside scenario as well as the on-road scenario.

### C. Complexity Analysis

In this section, we analyze the time complexity and space complexity of different models.

***Time Complexity***: Time complexity is the amount of time it takes to run the algorithm. We count the number of floating-points operations (**FLOPs**) required by algorithm, to measure time complexity. The time complexity of the overall CNN is the sum of the time complexity of all *conv* layers [19] [41]:

$$\textbf{Time} \sim \mathcal{O}\left(\sum_{l=1}^{N_{\text{conv}}} I_l^n \cdot K_l^n \cdot C_{l-1} \cdot C_l\right) \tag{8}$$

where $N_{\text{conv}}$ is the number of *conv* layers, $n$ is the dimension of convolution kernels (1-dim, 2-dim or 3-dim convolution), $I$ is the size of feature map, $K$ is the size of convolution kernel, $C_l$ is the number of output channels of the $l^{th}$ *conv* layer, that is, number of convolution kernels of this layer.

Another indicator to measure a model's time complexity is the training or prediction time. If time complexity is too high, it will lead to a large amount of time for model training and prediction. Therefore, we also measure the frame-level prediction/testing time for different models to evaluate the time complexity. We show the results in Table. VII.

[19]The time cost of fully connected layers and pooling layers is not involved in the this formulation. These layers typically take 5-10% computational time.

TABLE VII
TIME COMPLEXITY ANALYSIS

| Model | FLOPs | Prediction time (per frame) |
|---|---|---|
| RODNet-CDC | $4.75 \times 10^{11}$ | 11.2 ms |
| 4D-CDC [18] | $1.64 \times 10^{14}$ | - [19] |
| RAMP-CNN | $1.41 \times 10^{12}$ | 31.1 ms |

***Space Complexity***: Space complexity quantifies the amount of memory needed by an algorithm to run as a function. This consists of two parts: the total number of parameters (first term of (9)), and the occupied memory of the feature map output at all layers (second term of (9)).

$$\textbf{Space} \sim \mathcal{O}\left(\sum_{l=1}^{N_{\text{conv}}} K_l^n \cdot C_{l-1} \cdot C_l + \sum_{l=1}^{N_{\text{conv}}} I_l^n \cdot C_l\right) \tag{9}$$

We show the space complexity results, as well as the number of layers for different models in Table. VIII

From Table. VII and Table. VIII, we know that compared to the 4D-CDC model, RAMP-CNN needs almost **100** times fewer FLOPs, around **half** amount of parameters, and **35** times smaller feature map size. For practical application, this means RAMP-CNN would not only run 100 times faster than 4D-CDC for both training and prediction, but also take 35 times less memory. That confirms the claimed statement - RAMP-CNN has much less computation complexity than the 4D model.

Also, compared to RODNet-CDC model, the time and space complexity of RAMP-CNN is around 3 times higher. That, however, means the performance improvement of RAMP-CNN model comes at the expense of increased complexity.

### D. Summary

The proposed RAMP-CNN model achieves significant performance improvement over prior works on radar object recognition under parking lot, curbside, and on-road scenario, which establishes a new state-of-art baseline. In some hard cases, the radar object recognition functionality of RAMP-CNN might still be poor for supporting autonomous driving

[20]To compare the complexity between one 4D model and RAMP-CNN model model, we replace the 3D convolution kernels in RODNet-CDC model with the 4D convolution kernels and call the new model 4D-CDC.

[21]Note: we didn't implement the 4D-CDC model, so the prediction time and layer numbers are ignored here.

[22]The number of *conv* layers and *transposed conv* layers in models. For RODNet-CDC and RAMP-CNN model, the layers are all 3D; while for 4D-CDC model, all layers are 4D.

TABLE VIII
SPACE COMPLEXITY ANALYSIS

| Model | Parameters amount | Feature map size | Layers number [20] |
|---|---|---|---|
| RODNet-CDC | $3.47 \times 10^7$ | $6.31 \times 10^7$ | 6, 3 |
| 4D-CDC | $1.79 \times 10^8$ | $6.58 \times 10^9$ | 6, 3 |
| RAMP-CNN | $1.04 \times 10^8$ | $1.89 \times 10^8$ | 20, 9 |

presently.[23] However, it can be further improved in the future via incorporating more preprocessing to increase spatial resolution or adopting advanced radar platform with more antennas.

RAMP-CNN is also verified to work at the nighttime scenarios, where cameras are largely ineffective due to the low-light. Further, prior works [1], [2], [43] have been showing that mmW radars are with excellent environmental resistance and robustness because the millimeter-wave is less attenuated by fog, rain, or snow. Therefore, we have reason to believe that RAMP-CNN can be applied to these adverse conditions as a good substitute for optical sensors. However, due to the difficulty of capturing data in such circumstances locally, this must be left for future work.

There are several other advantages of applying RAMP-CNN to mmW radars - it has excellent range localization ability because of the centimeter-level range resolution (∼3.75 cm with 4 GHz sweep bandwidth). As shown in Fig. 14 (column 10), RAMP-CNN can resolve multiple close pedestrians with range and localize them separately. Besides, RAMP-CNN model has great generalization for the input data with a higher dimension. For example, if we add the elevation dimension (from the vertical radar array) to the current RAMP-CNN input, then the formed 5D data can still be sliced and processed by several lower-dimension (3D) models that nonetheless achieve better performance with acceptable computation complexity.

RAMP-CNN model fully exploits the temporal information behind the chirps in one frame, as well as the change of spatial information (range-angle info.) across frames. Thus, the performance of RAMP-CNN particularly for moving objects, shows significant improvements relative to state-of-art.

The ablation study shows that both the proposed data augmentation algorithms and training loss are helpful for boosting the performance of RAMP-CNN. It is worth noting that major performance improvement comes from the main body of RAMP-CNN model (3-Perspectives model); the cumulative impact of all elements in the RAMP-CNN architecture results in the promising performance improvement, at the expense of increased complexity.

## VIII. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel RAMP-CNN model for radar object recognition that can obtain the location (range and azimuth angle) and class of the objects in each frame by inputting the 3D radar cube sequences. The RAMP-CNN model fully exploits the temporal information behind the chirps in one frame, as well as the change of spatial information across frames, which makes it achieve significant performance improvement over the previous work. For future work, we are continuing to explore how to effectively utilize the radar data and create more sensible radar networks based on radar data properties.

[23]Based on the performance of camera object detection in [42], we infer that the acceptable/desired average precision and recall for radar would be 0.8 with different classes of objects.

## ACKNOWLEDGMENT

The authors would like to thank Yizhou Wang of Information Processing Lab, University of Washington, for the source codes of RODNet [17] that was used for the baseline evaluation in this article.

## REFERENCES

[1] K. Yoneda, N. Suganuma, R. Yanase, and M. Aldibaja, "Automated driving recognition technologies for adverse weather conditions," *IATSS Res.*, vol. 43, no. 4, pp. 253–262, Dec. 2019.
[2] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 103–111, Jun. 2019.
[3] I. Bilik, O. Longman, S. Villeval, and J. Tabrikian, "The rise of radar for autonomous vehicles: Signal processing solutions and future research directions," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 20–31, Sep. 2019.
[4] *AWR1843 Single-Chip 77- 79-GHz FMCW Radar Sensor datasheet*, Texas Instrument, Dallas, TX, USA, 2018.
[5] V. Giannini, M. Goldenberg, and A. Eshraghi, "9.2 A 192-virtual-receiver 77/79 GHz GMSK code-domain MIMO radar system-on-chip," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 164–166, Feb. 2019.
[6] C. Iovescu and S. Rao, *White Paper: The Fundamentals Millimeter Wave Sensors*. Dallas, TX, USA: Texas Instrument, 2017.
[7] S. Rao, *White paper: MIMO Radar*. Dallas, TX, USA: Texas Instrument, 2017.
[8] X. Gao, G. Xing, S. Roy, and H. Liu, "Experiments with mmWave automotive radar test-bed," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 1–6.
[9] A. Angelov, A. Robertson, R. Murray-Smith, and F. Fioranelli, "Practical classification of different moving targets using automotive radar and deep neural networks," *IET Radar, Sonar Navigat.*, vol. 12, no. 10, pp. 1082–1089, Oct. 2018.
[10] H. Rohling, S. Heuel, and H. Ritter, "Pedestrian detection procedure integrated into an 24 GHz automotive radar," in *Proc. IEEE Radar Conf.*, pp. 1229–1232, 2010.
[11] A. Bartsch, F. Fitzek, and R. H. Rasshofer, "Pedestrian recognition using automotive radar sensors," *Adv. Radio Sci.*, vol. 10, pp. 45–55, Sep. 2012.
[12] N. Scheiner, N. Appenrodt, J. Dickmann, and B. Sick, "Radar-based feature design and multiclass classification for road user recognition," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1–5.
[13] M. A. Richards, *Fundamentals of Radar Signal Processing*. New York, NY, USA: McGraw-Hill, 2005.
[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.
[15] B. Major *et al.*, "Vehicle detection with automotive radar using deep learning on range-azimuth-Doppler tensors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–5.
[16] F. Schubert, R. Rasshofer, and E. Biebl, "Deep learning radar object detection and classification for urban automotive scenarios," in *Proc. Kleinheubach Conf.*, Sep. 2019, pp. 1–4.
[17] Y. Wang, Z. Jiang, X. Gao, J.-N. Hwang, G. Xing, and H. Liu, "Rodnet: Object detection under severe conditions using vision-radio cross-modal supervision," 2020, *arXiv:2003.01816*. [Online]. Available: https://arxiv.org/abs/2003.01816

[18] *Automotive Adaptive Cruise Control Using FMCW Technology*. Accessed: Oct. 21, 2020. [Online]. Available: https://www.mathworks.com/help/phased/ug/automotive-adaptive-cruise-control-using-fmcw-technology.html
[19] *From Adas to Driver Replacement-is Actual Radar Performance Good Enough*. Accessed: Oct. 21, 2020. [Online]. Available: https://www.analog.com/en/technical-articles/from-adas-to-driver-replacement.html
[20] I. Roldan *et al.*, "DopplerNet: A convolutional neural network for recognising targets in real scenarios using a persistent range–Doppler radar," *IET Radar, Sonar Navigat.*, vol. 14, no. 4, pp. 593–600, Apr. 2020.
[21] W. Ye, H. Chen, and B. Li, "Using an end-to-end convolutional network on radar signal for human activity classification," *IEEE Sensors J.*, vol. 19, no. 24, pp. 12244–12252, Dec. 2019.
[22] F. Luo, S. Poslad, and E. Bodanese, "Human activity detection and coarse localization outdoors using micro-Doppler signatures," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8079–8094, Sep. 2019.
[23] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna Doppler radar with deep convolutional neural networks," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3041–3048, Apr. 2019.
[24] Q. Chen, Y. Liu, F. Fioranelli, M. Ritchie, B. Tan, and K. Chetty, "DopNet: A deep convolutional neural network to recognize armed and unarmed human targets," *IEEE Sensors J.*, vol. 19, no. 11, pp. 4160–4172, Jun. 2019.
[25] J. Bechter, F. Roos, and C. Waldschmidt, "Compensation of motion-induced phase errors in TDM MIMO radars," *IEEE Microw. Wireless Compon. Lett.*, vol. 27, no. 12, pp. 1164–1166, Dec. 2017.
[26] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, Jul. 1996.
[27] *xWR1843 Evaluation Module (xWR1843BOOST) Single-Chip mmWave Sensing Solution*, Texas Instrument, Dallas, TX, USA, 2018.
[28] N. Pandey, "Beamforming MIMO radar," Nat. Inst. Technol. Rourkela, Rourkela, India, Tech. Rep. 212EC6192, Jul. 2014.
[29] H. Sun, F. Brigui, and M. Lesturgie, "Analysis and comparison of MIMO radar waveforms," in *Proc. Int. Radar Conf.*, Oct. 2014, pp. 1–6.
[30] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5734–5743.
[31] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, Jun. 2011, pp. 52–59.
[32] M. Zhao *et al.*, "Through-wall human pose estimation using radio signals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7356–7365.
[33] K. He, G. Gkioxari and R. Girshick, "Mask R-CNN," 2017, *arXiv:1703.06870*. [Online]. Available: https://arxiv.org/abs/1703.06870
[34] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. CVPR*, 2017, pp. 270–279.
[35] Y. Wang, Y.-T. Huang, and J.-N. Hwang, "Monocular visual object 3D localization in road scenes," in *Proc. 27th ACM Int. Conf. Multimedia*, New York, NY, USA, 2019, pp. 917–925.
[36] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," 2018, *arXiv:1808.01244*. [Online]. Available: https://arxiv.org/abs/1808.01244
[37] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*. [Online]. Available: https://arxiv.org/abs/1904.07850
[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017, *arXiv:1708.02002*. [Online]. Available: https://arxiv.org/abs/1708.02002
[39] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Dec. 2017, pp. 464–472.
[40] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, Dec. 2014, pp. 1–8.
[41] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5353–5360.
[42] S. Devi, P. Malarvezhi, R. Dayana, and K. Vadivukkarasi, "A comprehensive survey on autonomous driving cars: A perspective view," *Wireless Pers. Commun.*, vol. 114, pp. 2121–2133, May 2020.
[43] Y. Golovachev, A. Etinger, G. Pinhasi, and Y. Pinhasi, "Millimeter wave high resolution radar accuracy in fog conditions-theory and experimental verification," *Sensors*, vol. 18, p. 2148, Jul. 2018.

**Xiangyu Gao** received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2018, and the M.S. degree in electrical and computer engineering from the University of Washington, Seattle, WA, USA, in 2020, where he is currently pursuing the Ph.D. degree. His research interests include sensor fusion for autonomous driving, statistical signal processing, and deep learning.

**Guanbin Xing** (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Peking University, Beijing, China, in 1996 and 1999, respectively, and the Ph.D. degree in electrical engineering from the University of Washington in 2004.

He has over 15 years of experience as a Senior System Architect in the wireless communications and digital broadcasting industry. In 2017, he joined CMMB Vision–UW EE Center on Satellite Multimedia and Connected Vehicles, as a Research Scientist, working on the mmWave radar signal processing and machine learning-based sensor fusion solutions for autonomous driving.

**Sumit Roy** (Fellow, IEEE) received the B.Tech. degree in electrical and computer engineering from IIT Kanpur, Kanpur, India, in 1983, the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 1985 and 1988, respectively, and the M.A. degree in statistics and applied probability in 1988.

From 2001 to 2003, he was a Senior Researcher at the Intel Wireless Technology Lab, where he was involved in systems architecture and standards development for ultrawideband wireless PANs and next-generation high-speed wireless LANs. Since 2008, he has been a Science Foundation of Ireland's E.T.S. Walton Fellow for a sabbatical at University College Dublin, Dublin, Ireland. From 2014 to 2015, he spent a sabbatical year at Microsoft Research, Bengaluru, India, as an Erskine Fellow at the University of Canterbury, Christchurch, New Zealand, and as a Short Term Visiting Foreign Expert at Shanghai Jiao Tong University. He is currently an Integrated Systems Professor of Electrical and Computer Engineering with the University of Washington, Seattle, WA, USA. His research interests include fundamental analysis/design of wireless communication and sensor network systems spanning a diversity of technologies and system application areas: next-gen wireless LANs and 5G/beyond 5G cellular networks, heterogeneous network coexistence, spectrum sharing and software-defined radio platforms, and vehicular and airborne networks.

Dr. Roy was elevated to a Fellow of the IEEE, for his contributions to multiuser communications theory and cross-layer design of wireless networking standards. He currently serves on the Executive Committee of the National Spectrum Consortium dedicated to efficient spectrum sharing between Federal licensed and civilian systems. He has served as an Associate Editor for all major ComSoc publications at various times, including the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and as a ComSoc Distinguished Lecturer (2017-2018).

**Hui Liu** (Fellow, IEEE) received the B.S. degree in electrical engineering from Fudan University, Shanghai, China, in 1988, and the Ph.D. degree in electrical engineering from the University of Texas at Austin, Austin, TX, USA, in 1995.

He was a Full Professor and an Associate Chairman with the Department of Electrical Engineering, University of Washington, Seattle, WA, USA, and a Chair Professor and the Associate Dean of the School of Electronic, Information, and Electrical Engineering, Shanghai Jiao Tong University. He was one of the principal designers of the 3G TD-SCDMA mobile technologies. He was the Founder of Adaptix, which pioneered the development of OFDMA-based mobile broadband networks (mobile WiMAX and 4G LTE). He is currently the President and the CTO with Silkwave Holdings, and an Affiliated Professor with the University of Washington. He has authored over 80 journal articles and two textbooks and holds 70 awarded patents. His research interests include broadband wireless networks, satellite communications, digital broadcasting, and multimedia signal processing. He contributed to the global standards for broadband cellular and mobile broadcasting. He was a recipient of the 1997 NSF CAREER Award, the Gold Prize Patent Award in China, three IEEE best conference paper awards, and the 2000 ONR Young Investigator Award.