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

Xiangyu Gao
University of Washington
Seattle, WA 98195
xygao@uw.edu

Guanbin Xing
University of Washington
Seattle, WA 98195
gxing@uw.edu

Sumit Roy
University of Washington
Seattle, WA 98195
sroy@uw.edu

Hui Liu
University of Washington
Silkwave Holdings
huiliu@uw.edu

Abstract

Millimeter-wave (mmW) radars are being increasingly integrated into commercial vehicles to support new Adaptive Driver Assisted Systems by enabling robust and high-performance object detection, localization, as well as recognition - a key component of environmental perception. In this paper, we propose a novel radar multiple-perspectives convolution 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 neural network (NN), we propose to combine several lower-dimension NN models within RAMP-CNN model that nonetheless completely exploits the temporal and spatial information with lower complexity. The extensive experiments show that our model achieves better AP and AR than prior works in all testing scenarios. Besides, RAMP-CNN is validated to work robustly under the nighttime, which enables low-cost radars as a potential substitute for pure optical sensing under severe conditions.

1 Introduction

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

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 that is a function of the round trip delay and therefore can be mapped directly to range [1]. Similarly, transmitting a train of equispaced FMCW chirps (also called a frame) allows Doppler velocity estimation for scenarios where the target 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 [1]. Finally, the use of multiple transmitters and multiple receivers (MIMO) configuration on radar enables azimuth localization of targets, by appropriate digital signal processing of the multiple transmitted waveforms reflected by the target to receiver array [2]. In summary, the ADC raw radar data - that has 3 dimensions:

Machine Learning for Autonomous Driving Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

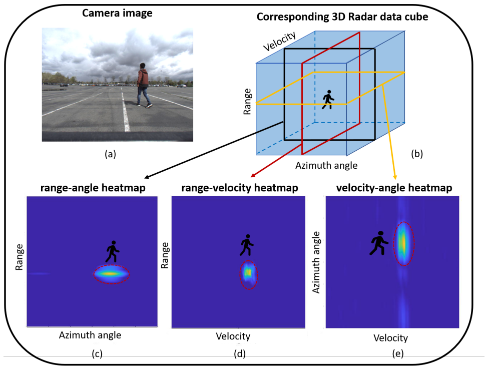


Figure 1: Abstraction of single frame input radar data and corresponding camera image: (a) The camera image of the pedestrian; (b) The range-velocity-angle radar data cube, three cross profiles of it are shown as figure (c), (d) and (e); (c) The range-azimuth angle heatmap; (d) The range-velocity heatmap;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

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 (Range FFT, Velocity FFT, and Angle FFT) [3] to obtain the 3D radar cube and name the generated radar cube range-velocity-angle (**RVA**) heatmap [1] [2].

The small form factor of 77 GHz FMCW radar boards - while a desirable feature - also limits the number of antennas that can be integrated, resulting in poor angular resolution. Specifically, two targets at the same distance and same radial velocity are not resolved in angle if separated by less than the resolution beamwidth, and even if resolvable, the spatial dimension is not well-defined. Hence to achieve reliable object recognition using such hardware, prior works [3, 4] 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.

While several prior works [3, 4, 5, 6, 7] explore radar object recognition with various input data formats using neural networks, none has ever combined the spatial and temporal domain information, i.e., by jointly processing the 3D radar cube *sequences* (from multiple frames). Our fundamental contribution is a *deep learning* network design that approaches performance upper bound since the joint processing can exploit all available information (see Section 3.1).

However, it is impractical to implement 4D (above 3D plus time sequences) convolution processing as the resulting computational complexity is unacceptable for real-time perception. Therefore, we propose to use the combination of several lower-dimension (3D) models that nonetheless exceed the performance of prior methods with acceptable complexity (see Table. 4). 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, as shown in Fig. 1. 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. 2). We name above radar network architecture RAMP-CNN.

To avoid the overfitting when training RAMP-CNN model, we propose a new loss function that pushes NN to learn more from the RV and VA features and avoids it to give more weights to the straightforward and accessible RA features than the velocity-based features (see Section 3.3).

2 Related Work

We comment on the relevant prior works [3, 4, 5, 6, 7] that have attempted radar object detection with various radar data input formats.

[4] 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. Further on, [3] propose a new framework to cess the raw radar data and enhance radar object classification by incorporating the additional spatial information. However, the above methods are all two-stages architecture where the regions

[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.

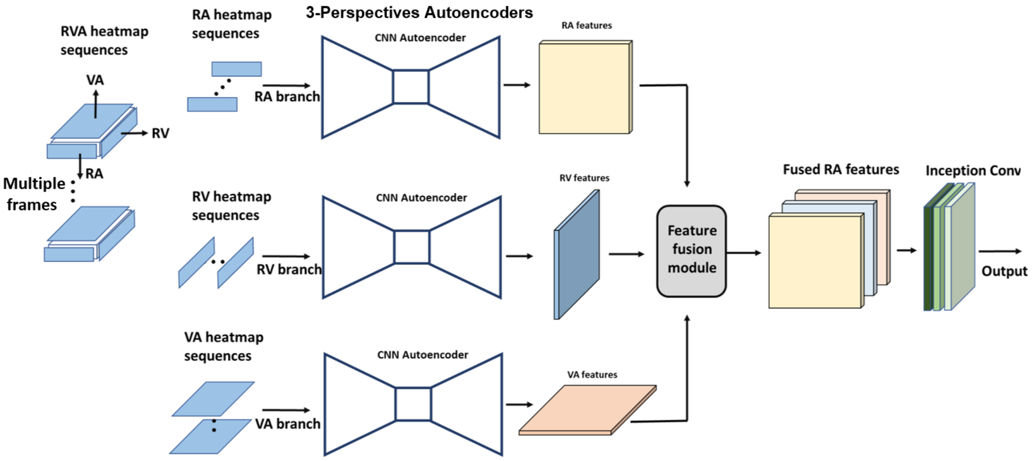


Figure 2: The architecture of RAMP-CNN model

of interest (i.e., the location of objects) need to be found before the classification. Besides, several prepossessing procedures (i.e., CFAR detection, DBSCAN clustering, etc) are adopted before feeding the radar input to neural network, which may make the information incomplete.

[6] presents a single shot detection and classification system in urban automotive scenarios, which is based on the YOLO system applied to the cessed range-Doppler-angle power spectrum with a 77 GHz FMCW radar. To feed the 3D radar power spectrum into the 2D YOLO network, [6] condense the angle domain by choosing the maximum. The range-angle domain is the main perspective to observe the objects' reflection ability and shapes such that condensing the angle is not the best choice.

[5] illustrates a deep-learning-based vehicle detection solution that operates on the image-like range-velocity-angle radar tensor. The ability of its accuracy vehicle detection in the high way scenario mainly relies on recognizing the energy distribution on the range and angle dimension (i.e., the contribution of the Doppler dimension to detection is small). This may be hard for solving the various object detection and classification problem we have.

[7] shows a radio object detection network that acts purely on the processed radar data in the format of RA heatmap sequences. The adopted autoencoder structure in [7] that processes the RA heatmap sequences exploits the temporal information behind the change of spatial patterns across frames. However, for each frame, [7] 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 the chirps.

3 RAMP-CNN model: A Convolutional Neural Network for Radar Data

3.1 3-Perspectives Autoencoders

As shown in Fig. 2, the main body of the RAMP-CNN architecture is composed of 3 convolutional autoencoders (CAE). The autoencoders extract features from the heatmap sequences of different perspectives - that is, RA, RV, and VA respectively.

The Physical Significance of Network Design : The first CAE processes complex-valued RA heatmap sequences with 3D convolution (*conv*) and transposed convolution (*transposed conv*) layers. Similar to [7], 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 [3] at an arbitrarily selected chirp. Those 3D *conv* 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. The phase change of mmW signal along time is sensitive to the object movement, e.g. 1 mm position movement results in phase shift $\Delta\phi = \pi$ for 77 GHz radar. 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 [3] RV and VA heatmap sequences respectively. 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 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 [7].

Network Details: We adopt the 3D Convolutional-De-Convolutional [8] model as our CAE, which is effective in summarizing spatio-temporal patterns from raw data into high-level semantics. Each CAE includes six 3D *conv* layers and three 3D *transposed conv* layers. All 3D *conv* layers are followed by a batch-normalization layer and the *ReLU* activation function. The first two 3D *transposed conv* layers are followed by the *PReLU* activation function.

To preserve the phase information in RA heatmap input, we represent complex-valued heatmap by two real-valued channels that store the real and imaginary parts following [9] (i.e., similar to RGB channels). While for RV and VA heatmap inputs, it suffices to only keep the absolute value and use the one-channel heatmap.

3.2 Feature Fusion Module

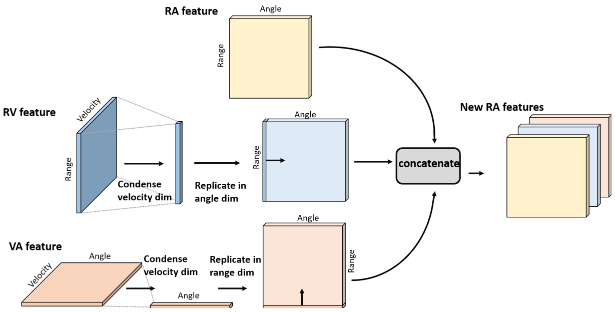


Figure 3: Feature Fusion Module

The key issue is how to transform the RV and VA features to RA domain such for supporting an improved final classification. In Fig. 3, the VA feature is condensed along the velocity dimension by summing, then the condensed vector is replicated in the missing dimension - range, and similarly for RV feature. Thereafter, we concatenate all features along channel dimension and input them to following network. The below analogy applied to radar processing suggests how to use VA feature - that provides good azimuth angle information but no range information.

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 the other sensors, i.e. acquire initial angle information/feature via the ear [4].

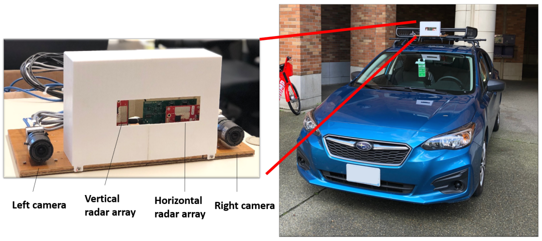
Convolution Layers after Feature Fusion Module: There are two *conv* layers that take as input the fused features and make recognition decisions: one 3D inception layer, and one ordinary 3D convolution layers with kernel size (3, 3, 3). Note that the ordinary *conv* layer operates on time, range and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

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

[3] 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.

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

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



(a) (b)

Figure 4: 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.

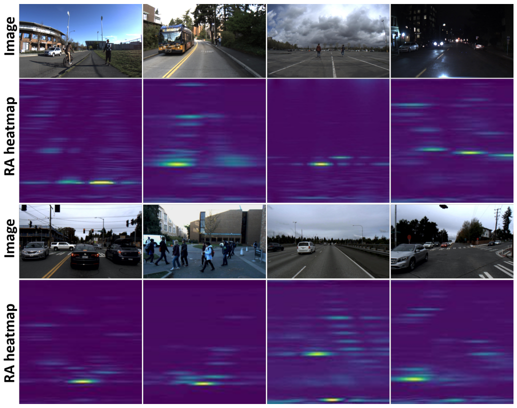


Figure 5: 8 scenario examples in the collected UWCR dataset: row 1, 3 are the camera images; row 2, 4 are the corresponding radar range-azimuth angle heatmaps.

3.3 Loss Function for All-perspectives Learning

To ease the training burden, we use the center keypoint to represent the existence of objects following [11]. For each ground truth center point $\mathbf{p}$ with location (p_r, p_θ), class id p_c and frame id p_t, we compute its Gaussian representation, and then splat all Gaussians onto a heatmap $Y \in [0,1]^{D\times W\times H\times C}$, and take the element-wise maximum if two Gaussians overlap [6]. Y is used as ground truth.

Let $X_{\text{RA}}, X_{\text{RV}}, X_{\text{VA}}$ be the input RA, RV and VA heatmap sequences, the aim of RAMP-CNN model is to predict center-point heatmaps [12] $\hat{Y} \in [0,1]^{D\times W\times H\times C}$ in RA domain, where $\hat{Y}_{t,r,\theta,c} = 1$ corresponds to a detected target at range r, azimuth angle θ, 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 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 [12, 13]:

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

where α and β are hyper-parameters of the focal loss [13], and N_{obj} is the number of objects in ground truth [7]. Compared to [13], we add a new scalar hyper-parameter κ, 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 [12], we use $\alpha = 2$ and $\beta = 4$ in all our experiments.

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 [8].

The above analogy applied to loss function design suggests how to make NN fully utilize all three perspectives 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 $X_{\text{RA}} = 0$ in the new loss term, i.e., we input $X = (0, X_{\text{RV}}, X_{\text{VA}})$ to NN $P_w(Y|X)$ again such that getting the new prediction $\hat{Y}'$. $\hat{Y}'$ is also supervised by ground truth Y with (1).

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

[7] The normalization by N_{obj} is chosen as to normalize all positive focal loss instances to 1.

[8] 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

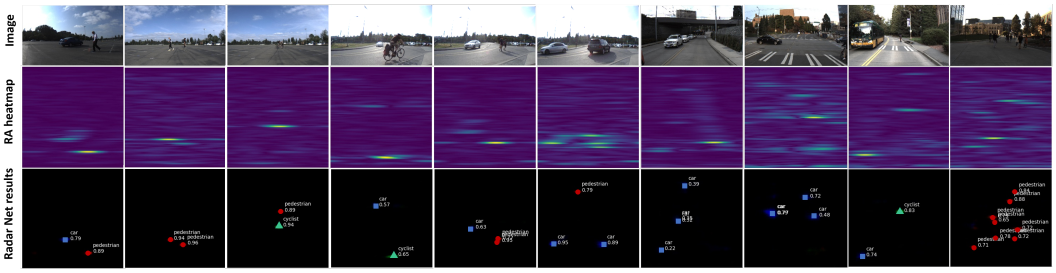


Figure 6: 10 test examples from the parking lot scenario (col 1-3), curbside scenario (col 4-6), and on-road scenario (column 7-10). For each column, the top row is the synchronized camera image for visualization, the second row is radar RA heatmap, and the bottom row is the visualization [9]of the RAMP-CNN model results.

Table 1: Performance comparison between different models

Model	Overall		Parking Lot Scenario		Curbside Scenario		On-road Scenario	
	AP	AR	AP	AR	AP	AR	AP	AR
CDMC [3]	30.55%	54.79%	65.74%	76.56%	28.68%	53.93%	4.88%	25.76%
RODNet-HG [7]	71.84%	76.03%	93.87%	95.36%	61.65%	70.09%	41.97%	53.04%
RODNet-CDC [7]	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%**

The final loss is the weighted sum of two terms:

$$L_{loss} = L_{\hat{Y}Y} + \gamma L_{\hat{Y}'Y} \tag{2}$$

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

4 Experiment

4.1 UW Camera-Radar (CR) Dataset

A large camera (image) and radar (raw 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 by a vehicle-mounted platform that is driven (see Fig. 4(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 several examples in our UWCR dataset at Fig. 5.

The binocular cameras in the data collection platform (Fig. 4(a)) are synchronized with radars, and they can provide the location and class of semantic objects via the Mask R-CNN object detection [14] and unsupervised depth estimation [15, 16] on captured 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.

4.2 Experiments

Pre-processing and Training: We implement the 3-DFFT [3] on raw radar data to obtain the RVA heatmap sequences. The FFT for all are 128 points. We choose input frame number $M = 16$. Therefore, the size of input data is $128 \times 128 \times 128 \times 16$. We train the RAMP-CNN with Adam optimizer, and data augmentation (flipping, tranalating, interpolating, and mixing).

Baselines: We compare the RAMP-CNN model with RODNet-CDC [7], RODNet-HG [7] (double the *conv* layers of RODNet-CDC), the state-of-art radar object detection models, as well as the CDMC [3], a model that fully exploits the micro-Doppler signatures of moving objects.

Evaluation Metrics: We use the Average Precision (AP) and Average Recall (AR) to evaluate the performance - using tp, fp, and fn [10]. We adopt the CFAR [17] algorithm and threshold 0.2 to filter

[9] In Fig. 6 and 7, we use different icons to visualize the occurrence of targets.

[10] Here, the 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.

out the target center points from prediction $\hat{Y}$. Whether the targets are correctly located is determined by a object size-adaptive distance threshold.

4.2.1 Experiment 1

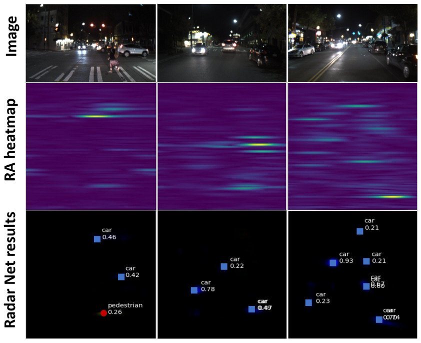


Figure 7: 3 test examples from the nighttime scenario

We train the RAMP-CNN with the whole training set and test it separately under the **daytime** parking lot, curbside, city-road, as well as the **nighttime** city road.

The daytime scenario: As shown in Table.1, for parking lot scenario, RAMP-CNN achieves **nearly perfect** performance (97.38% AP, 98.37% AR), and beat all prior works. For curbside scenario, it has **around 15% improvement** over the state-of-art results. For city-road scenario, RAMP-CNN obtains about **10% improvement in AP** and **6% improvement in AR** over RODNet-CDC baseline. Several testing examples are shown in Fig. 6.

The nighttime scenario: The RAMP-CNN model is also tested under nighttime (where cameras are largely ineffective) to advance the cause of radar as a low-cost substitute for optical sensors that fail under such conditions. Shown in Fig. 7, RAMP-CNN 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 the nighttime test set, we don't numerically evaluate the performance of RAMP-CNN model here.

4.2.2 Experiment 2

We redivide part of data into the **static object** scenario and **moving object** scenario, and test them separately.

Table 2: Performance comparison between different models for static objects and moving objects

Model	Static object scenario		Moving object scenario	
	AP	AR	AP	AR
CDMC [3]	32.34%	50.73%	28.68%	53.93%
RODNet-HG [7]	56.26%	61.36%	61.68%	70.11%
RODNet-CDC [7]	65.02%	69.91%	64.01%	71.97%
RAMP-CNN	67.58%	70.27%	79.25%	84.21%

As shown in Table. 2, for **static object** scenario, the performance of RAMP-CNN (AP around 67%, AR around 70%) is **in the same level** of the RODNet-CDC baseline; while for **moving object** scenario, there is **a performance gap (about 15% AP and 13% AR)** between RAMP-CNN model (AP around 80%, AR around 84%) and other baselines (AP around 65%, AR around 71%).

The results proves the RAMP-CNN model achieves better performance for moving object recognition, since it fully exploits the temporal information behind the chirps within one frame, as well as the change of spatial information (range-angle information) across frames.

4.3 Discussion

The proposed RAMP-CNN model achieves significant performance improvement over the previous works on the recognition of radar objects under daytime scenario. Although in some cases the radar object recognition functionality might still be thought of as poor for supporting autonomous driving, RAMP-CNN establishes a new state-of-art baseline and can be further improved via incorporating more pre-processing or spatial resolution.

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 of mmW radar ($\sim$3.75 cm with 4 GHz sweep bandwidth). As shown in Fig. 6 (column 10), RAMP-CNN can resolve and localize multiple close pedestrians with range. Besides, RAMP-CNN has good generalization for other 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 will nonetheless achieve better performance with relatively low computational complexity.

5 Analysis and Ablation study

5.1 Ablation Study

We retrain a new RAMP-CNN model that replaces the proposed training loss with the ordinary focal loss [12, 13]. The experiment results shown in Table. 3 verify that new training loss helps improve performance by pushing the RAMP-CNN model to learn more Doppler-related features. Specifically, the RAMP-CNN model obtains around 4% AP improvement and 3% AR improvement in both parking lot scenario and curbside scenario as well as the on-road scenario.

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 promising performance improvement, at the expense of increased complexity.

Table 3: Ablation Study

Model	New training loss	Overall		Parking lot scenario		Curbside scenario		On-road scenario	
		AP	AR	AP	AR	AP	AR	AP	AR
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%**

5.2 Complexity Analysis

Time complexity: We count the number of floating-points operations (**FLOPs**) required by all convolutional layers [18], as well as the frame-level prediction time, to measure the time complexity.

Space complexity: Space complexity quantifies the amount of memory needed by an algorithm to run. In Table. 4, we adopt the number of parameters, size of the all feature map output to measure it.

From Table. 4, we know that RAMP-CNN model needs almost **100** times fewer FLOPs, around **half** amount of parameters, and **35** times smaller feature map size, compared to 4D-CDC model. For practical application, this means RAMP-CNN would not only run 100 times faster than 4D-CDC for both training and inference, but also take 35 times less memory, which **confirms the claimed statement**. Also, compared to RODNet-CDC, the time and space complexity of RAMP-CNN is around 3 times higher. That, however, means the performance improvement of RAMP-CNN **comes at the expense of increased complexity**.

6 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 and spatial information, which makes it achieve significant performance improvement over the previous work. For future work, we are continuing to explore how to utilize the radar data effectively and create a more sensible radar neural network based on radar data properties.

[11] 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.

[12] Note: we didn't implement the 4D-CDC model, so the prediction time is ignored here.

Table 4: Complexity Analysis

Model	**Time Complexity**		**Space Complexity**	
	FLOPs	Prediction time (per frame)	Parameters amount	Feature map size
RODNet-CDC	4.75×10^{11}	11.2 ms	3.47×10^{7}	6.31×10^{7}
4D-CDC [11]	1.64×10^{14}	- [12]	1.79×10^{8}	6.58×10^{9}
RAMP-CNN	1.41×10^{12}	31.1 ms	1.04×10^{8}	1.89×10^{8}

Acknowledgement

This work was supported by CMMB Vision – UWEE Center on Satellite Multimedia and Connected Vehicles.

References

[1] C. Iovescu and S. Rao, *White paper: The Fundamentals of Millimeter Wave Sensors*. No. SPYY005, Texas Instrument, 2017.

[2] S. Rao, *White paper: MIMO Radar*. No. SWRA554A, Texas Instrument, 2017.

[3] X. Gao, G. Xing, S. Roy, and H. Liu, "Experiments with mmwave automotive radar test-bed," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1–6, 2019.

[4] 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 Navigation*, vol. 12, no. 10, pp. 1082–1089, 2018.

[5] B. Major, D. Fontijne, A. Ansari, R. Teja Sukhavasi, R. Gowaikar, M. Hamilton, S. Lee, S. Grzechnik, and S. Subramanian, "Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[6] R. Pérez, F. Schubert, R. Rasshofer, and E. Biebl, "Deep learning radar object detection and classification for urban automotive scenarios," in *2019 Kleinheubach Conference*, pp. 1–4, Sep. 2019.

[7] 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.

[8] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] M. Zhao, T. Li, M. A. Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human pose estimation using radio signals," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7356–7365, June 2018.

[10] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," pp. 52–59, 06 2011.

[11] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," 2018.

[12] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2017.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2017.

[15] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[16] Y. Wang, Y.-T. Huang, and J.-N. Hwang, "Monocular visual object 3d localization in road scenes," in *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, (New York, NY, USA), p. 917–925, Association for Computing Machinery, 2019.

[17] M. A. Richards, *Fundamentals of Radar Signal Processing*. US: McGraw-Hill Professional, 2005.

[18] K. He and J. Sun, "Convolutional neural networks at constrained time cost," pp. 5353–5360, 06 2015.