Multi-View Radar Semantic Segmentation

Arthur Ouaknine[1,2]   Alasdair Newson[1]   Patrick Pérez[2]   Florence Tupin[1]
Julien Rebut[2]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
[2]valeo.ai, Paris, France
arthur.ouaknine@telecom-paris.fr

Abstract

Understanding the scene around the ego-vehicle is key to assisted and autonomous driving. Nowadays, this is mostly conducted using cameras and laser scanners, despite their reduced performance in adverse weather conditions. Automotive radars are low-cost active sensors that measure properties of surrounding objects, including their relative speed, and have the key advantage of not being impacted by rain, snow or fog. However, they are seldom used for scene understanding due to the size and complexity of radar raw data and the lack of annotated datasets. Fortunately, recent open-sourced datasets have opened up research on classification, object detection and semantic segmentation with raw radar signals using end-to-end trainable models. In this work, we propose several novel architectures, and their associated losses, which analyse multiple "views" of the range-angle-Doppler radar tensor to segment it semantically. Experiments conducted on the recent CARRADA dataset demonstrate that our best model outperforms alternative models, derived either from the semantic segmentation of natural images or from radar scene understanding, while requiring significantly fewer parameters. Both our code and trained models are available at https://github.com/valeoai/MVRSS.

1. Introduction

Radar sensors have been used in the automotive industry for the last two decades, e.g., for automatic cruise control or blind spot detection. They have become the sensor of choice for applications requiring time to collision as they provide, besides localization, the relative velocity thanks to the Doppler information. However, these radars have been hindered previously by their poor angular resolution.

The shift from driving assistance to automated driving drastically increased the necessary level of performance,
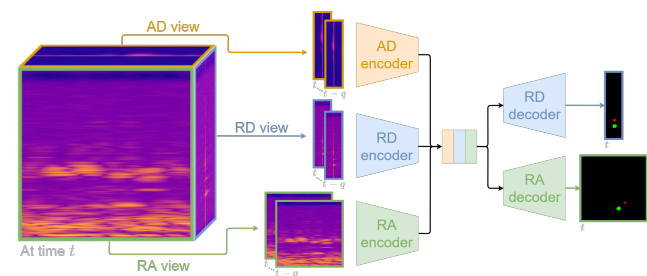

Figure 1. **Overview of our multi-view approach to semantic segmentation of radar signal**. At a given instant, radar signals take the form of a range-angle-Doppler (RAD) tensor. Sequences of $q+1$ 2D views of this data cube are formed and mapped to a common latent space by the proposed multi-view architectures. Two heads with distinct decoders produce a semantic segmentation of the range-angle (RA) and range-Doppler (RD) views respectively ('background' in black, 'pedestrian' in red and 'cyclist' in green in this example).

robustness and safety. Safety requires redundancy mechanisms at all levels of the system: From the sensing parts to the final decision modules. Redundancy at the sensor level can be reached using three sensors of different nature such as camera, lidar and radar.

While deep learning has brought about major progress in the automotive use of cameras and lidars – for object detection and segmentation in particular – it is only recently that it has also embraced radar signals. In fact, even though the radar technology has greatly improved, the signal processing pipeline has remained the same for years. This sensor is now a source of interest since public datasets have been released [3, 6, 15, 29, 33, 34, 45].

Radar scene understanding is in its infancy, but it provides key information to compensate for weaknesses of the other sensors. The radar data presented as a range-angle-Doppler (RAD) tensor contains signatures of the objects surrounding the car with enough details to distinguish them in the representation. Unlike object detection using bound-

ing boxes, semantic segmentation is appropriate for this task, since the object signatures have extremely variable sizes and may be mixed up due to the sensor's resolution.

In this work, we propose an approach to multi-view radar semantic segmentation, illustrated in Fig. 1, that exploits the entire data while addressing the challenges of its large volume and high level of noise. The segmentation is performed on the range-Doppler and range-angle views, which suffices to deduce the localisation and the relative speed of objects. Our first contribution is a set of lightweight neural network architectures designed for multi-view semantic segmentation of radar signal. The second contribution is a set of loss terms to train models on these tasks while preserving coherence between the multi-view predictions. Experiments show that our best model outperforms other methods considered for radar semantic segmentation in both quantitative and qualitative evaluations.

We present automotive radar sensing and related works in § 2, then our contributions and the methods that we compare (§ 3), experiments (§ 4) and conclusions (§ 5).

2. Background

2.1. Automotive radar sensing

A radar is an *active* sensor that emits electromagnetic waves, which are received back after being reflected in the environment [5, 17]. Standard in automotive, a frequency-modulated-continuous-wave (FMCW) radar emits a sequence of frequency-modulated signals called chirps. The received signal in the time domain is recorded in a 3D tensor indicating the chirp index, the chirp sampling and the corresponding receiver antenna index. It is usually named analog-to-digital converter (ADC) signal. The object distance is extracted using a Fourier transform along the chirp sequence (Range-FFT). A Doppler-FFT is then applied along the chirp sampling axis to estimate the phase difference and deduce the radial velocity of the reflective surface. Finally, an Angle-FFT processes the signal through the pairs of antennas to estimate the view angle to the object. This sequence of FFTs results in the range-angle-Doppler (RAD) tensor, a 3D data cube of complex numbers where each axis amounts to discretised values of the corresponding physical measurement. With conventional FMCW radars, the RAD tensor is usually not available as it is too computing intensive to estimate. CFAR algorithm [39] is typically applied to extract objects in range-Doppler while a sparse point cloud is obtained using beamforming.

From RAD tensor to multiple views. The intensities (squared modulus) in the RAD tensor provide information on the power backscattered by moving objects; See a visualization in Supplementary Material. Due to the coherent imaging, the backscattered signal presents strong fluctuations also known as speckle phenomenon and well studied

by Goodman [18]. This phenomenon can be modeled as a multiplicative noise. The averaging of few samples (operation known as multi-looking) reduces the noise strength, whereas the logarithmic transform induces a variance stabilization of the resulting signal [13].

Let $\mathbf{X}^{\text{RAD}}$ be the complex RAD tensor. Averaging its intensities across one of its dimensions leads to three possible 2D views: RA, RD and AD (Fig. 1). For instance, the RA view, expressed in decibels, is defined as:

$$\mathbf{x}^{\text{RA}}[r, a] = 10 \log_{10} \left(\frac{1}{N_{\text{D}}} \sum_{d=1}^{N_{\text{D}}} \left| \mathbf{X}^{\text{RAD}}[r, a, d] \right|^2 \right), \quad (1)$$

with $|.|$ the modulus and N_{D} the number of Doppler bins. Turning RAD tensors into views aims both to compress substantially the data representation and to reduce its noise. In practice, this reduces the size of the data by a factor of 50.

2.2. Related work

Low-cost FMCW radar is used in applications such as hand-gesture recognition [12, 19, 23, 25, 42, 48, 49, 50, 53, 54], human activity recognition [55], fall motion recognition [44], in-vehicle passenger detection [2] or surveillance [21]. This section will focus on automotive applications.

Raw radar signal is rich but complicated and noisy. Its processing pipeline can deliver a sparse point cloud (PC). Scene analysis can be done at various stages of this chain.

Point-cloud architectures. Point clouds are the most common interfaces of commercialized automotive radar. This representation has been broadly explored for object detection [11, 30], non-line-of-sight occluded object localisation and tracking [41] and occupancy grid segmentation in the bird's eye view (BEV) [26, 37, 38, 43, 47]. Despite good performances, PC approaches suffer from the radar preprocessing inefficiency. Filtering is usually performed to remove noise and simplify the signal representation. Unfortunately, this leads to the loss of useful information.

Single-view methods. An alternative to PC representation is to exploit the signal before CFAR filtering. Since the 3D RAD representation is cumbersome, it is usually considered by slicing or aggregating the tensor along one dimension.

The range-angle (RA) view, with its intuitive polar representation of the scene, is preferable for tasks like object classification [36], detection [14] and localisation [51], and for odometry [1]. In particular, [22] proposes a method to segment static and moving objects in RA view, which we compare to (see Section 3.2 for details).

The range-Doppler (RD) view is particularly useful since it indicates the relative speed of scene reflectors at each range bin. It is not widely explored though because of its less direct interpretability. Recent automotive works use Doppler spectrograms for vehicle classification [7] and RD views for object detection [32, 52] or source separation [8].

Brodeski *et al*. [4] also propose a two-stage method to detect objects in RD view and infer their position in simple and limited contexts.

Single-view methods suffer from a lack of information, as they do not exploit the entire radar data. Moreover, predictions on a single view are not able to provide both the position and the relative velocity of a detected object.

Multi-view methods. Recent works exploit the entire information in the RAD tensor. Handling this bulky representation is challenging. Palffy *et al*. [35] use a multi-stage method on only a sub-part of the tensor and detect objects in camera images. Major *et al*. [28] create 2D views by sum-pooling over each of the tensor's axes. Their multi-view representations are processed by a single network specialised in RA object detection. In the same vein, Gao *et al*. [16] also pre-process the RAD tensor into summed 2D views. Then, specialised auto-encoders extract features from each view which are fused to localise objects in the RA view. See further details on this approach in Sec. 3.2.

To the best of our knowledge, there is no previous work on multi-view radar semantic segmentation. In the next sections, we will describe our method which is able to both process the entire RAD tensor and segment radar multi-views to predict localisation and relative velocity of objects.

3. Methods and architectures

We now present in Sections 3.1 and 3.2 several methods for image segmentation and radar scene understanding, to which we compare our work. They are chosen for their performance and their relevance to the radar semantic segmentation task. Except RSS-Net [22], these architectures were not originally designed for radar semantic segmentation, nor to handle multiple views of a data volume. Consequently, for each of them, we have trained two models independently for range-Doppler and range-angle segmentation respectively. More details are provided in Section 4.2.

In Section 3.3, we then introduce the three proposed architectures for multi-view radar semantic segmentation.

3.1. Image-based competing methods

Long *et al*. [27] propose Fully Convolutional Networks (FCN), consisting of convolutional and down-sampling layers followed by transposed convolutions ("up-convolutions"). The final representations are processed by a 1D convolution with softmax to predict a category for each pixel. Several versions are proposed depending on the feature-map scales used to generate the output. FCN has been used for semantic segmentation of radar data in [34], where FCN-8s version achieves the best performance.

The U-Net architecture [40] is composed of equal-depth down-sampling and up-sampling pathways linked by skip connections. Originally used for medical images, it is especially well suited for small-object segmentation.

The DeepLabv3+ [10] is a popular encoder-decoder model for semantic segmentation of natural images. The encoder uses "atrous" separable convolutions which increase the receptive field of the network. The proposed Atrous Spatial Pyramidal Pooling (ASPP) layer [9] combines atrous convolutions with various dilation rates to learn multi-scale features followed by a 1D convolution.

3.2. Radar-based competing methods

Kaul *et al*. [22] propose RSS-Net, specialised in radar semantic segmentation, in particular for range-angle representations. Its architecture is similar to DeepLabv3+ with an encoder composed of 8 atrous convolutional layers, an ASPP module and a convolutional decoder with upsampling. The architecture is designed to reduce the dimension of the feature maps in the encoder, leading to excellent performance for range-angle BEV semantic segmentation.

Gao *et al*. [16] propose the Radar Multiple-Perspective Neural Network (RAMP-CNN) for object detection in RA representations. They aggregate the RAD tensor into 2D radar views which are processed by separate encoder-decoders with 3D convolutional layers. The final range-angle features are processed by a 3D inception module. RAMP-CNN achieves state-of-the-art performance in localising and classifying multiple objects in range-angle views.

3.3. Proposed multi-view methods

We propose three lightweight neural network architectures specialised in multi-view radar semantic segmentation, whose general principle is illustrated in Fig. 1. They take a temporal stack of radar views as their input and process them with dedicated encoders. The generated feature maps are fused in a shared latent space from which different decoders predict semantic segmentation maps for each output view. Since neither encoders nor decoders share weights, we keep them simple. Thus, there are specialised parts of the network in each view, and a reasonable number of parameters altogether. Further details about the architectures are provided in Supplementary Material.

Multi-view network (MV-Net). We first propose (see Figure 2 (1)) a baseline in the form of a double encoder-decoder architecture that processes stacked RD and RA views and predicts simultaneously the RD and RA semantic segmentation maps. It is important to note that we can deduce the third view (AD) from the RA and RD views, and it is therefore not necessary to segment it in the output. This is the case for all the architectures we propose.

Each encoder is composed of two blocks, each one with two sequences of convolution, batch normalisation and activation layers. The two blocks are separated by a max-pooling operation to down-sample the feature maps along the range axis (Doppler's resolution being lower, we keep it unchanged). The feature maps from both encoders are
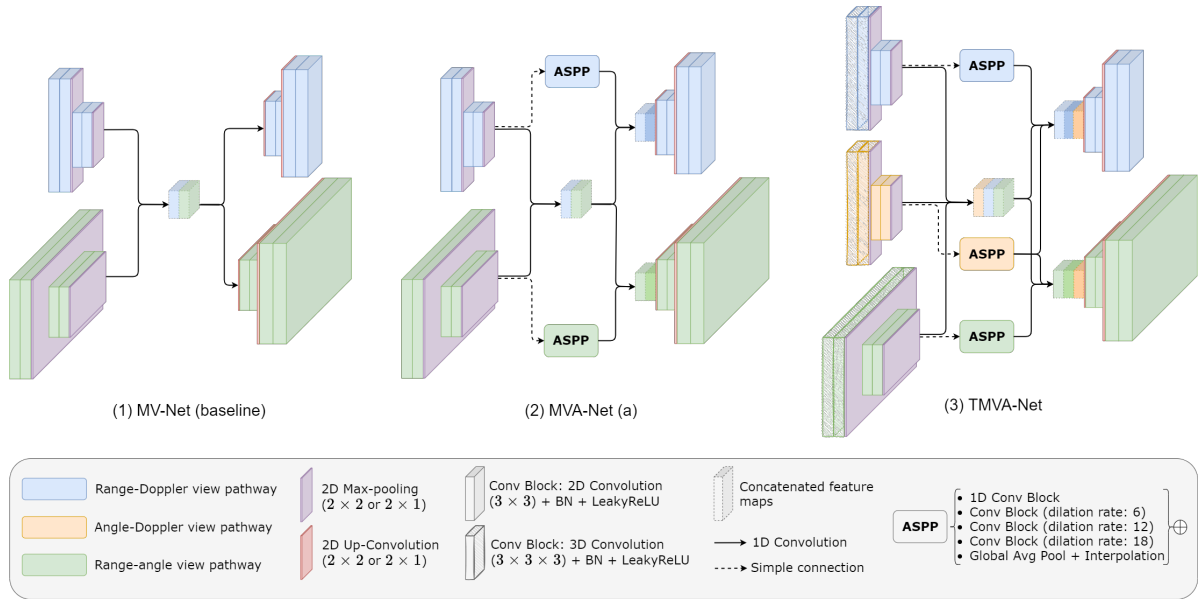
Figure 2. **Multi-view architectures for radar semantic segmentation.** (1) The multi-view network (MV-Net), considered as a baseline, is composed of two encoders, two decoders and a common latent space. (2) The MVA-Net(a) has an additional ASPP module for each view pathway; See text for variant MVA-Net(b); (3) The TMVA-Net architecture is similar to the MVA-Net(a) with an additional angle-Doppler encoder while exploiting the temporal dimension with 3D convolutions at the top of the encoders. The detailed architectures are provided in Supplementary Material.

down-sampled, processed by a 1D convolution and stacked into a common latent space. Since the input views are stacked according to the time dimension, this linear combination of the feature maps aims to learn temporal correlations. The features in the shared latent space are then processed with 1D convolution layers and used as the input of each decoder. There are two decoders predicting respectively the range-Doppler and range-angle semantic segmentation maps. Each one is composed of two blocks with two sequences of convolution, batch normalisation and activation layers. The up-sampling between the blocks is carried out by up-convolutions. A final 1D convolution performs a combination of the outputs of each decoder to generate K feature maps, where K is the number of classes. A softmax operation is then applied pixel-wise to the K feature maps generating soft masks.

Multi-view network with ASPP modules (MVA-Net). The ASSP module [9] used in DeepLabv3 [10] allows features to be jointly learned at different scales at a given depth of the network with no need for larger kernels or additional parameters. As shown in RSS-Net [22], it is well suited for radar semantic segmentation since the objects' signatures can vary considerably. Our second architecture, MVA-Net, introduces the use of this ASSP module at the end of each decoder of our MV-Net baseline. The generated multi-scale feature maps are concatenated, processed by a 1D convolution and stacked to the input

of each corresponding decoder. In effect, we propose two variants of MVA-Net: MVA-Net(a), shown in Figure 2 (2), consists in a double encoder-decoder architecture with ASPP modules to process and segment RD and RA views; MVA-Net(b) has an additional encoding branch learning features from the AD view. Similarly to the other encoding branches, the AD backbone generates features that are stacked in the common latent space. However, the outputs of its ASPP module contain both the angle and Doppler information. Thus the multi-scale feature maps from the AD pathway are stacked to the inputs of both the RD and RA decoders.

Temporal multi-view network with ASPP modules (TMVA-Net). The temporal dimension provides valuable information for radar semantic segmentation. It helps in estimating the shape of an object's signature despite high noise levels, and distinguishing objects that are close together with similar velocities. Our third architecture, which we will show to have the best performances, TMVA-Net in Fig. 2 (3), extends MVA-Net by explicitly leveraging the temporal dimension. The first block's 2D convolutions are replaced by 3D convolutions in each encoder branch, making it able to learn the spatio-temporal characteristics with limited increase in the number of parameters. Since 3D convolutions require a large number of parameters, full 3D-convolutional encoders, such as in [16], have not been retained. Hence, TMVA-Net is composed of three encoders

with 3D and 2D convolutions, one for each input view. Each one of them has a dedicated ASPP module. The feature maps generated from each encoding backbone are stacked into a shared latent space. From there, two decoders segment respectively the RD and RA views. They take as input the stacked features from the processed latent space and the multi-scale feature maps from the dedicated ASPP modules.

3.4. Losses

In what follows, we use the generic notation $f_\theta(\mathbf{x}) = \mathbf{p}$ for a segmentation model with parameters θ, input $\mathbf{x}$ and output $\mathbf{p}$. Training f_θ amounts to minimising w.r.t. θ a suitable loss function, given training examples $\mathbf{x}$ with ground truth $\mathbf{y}$. The architectures presented in Secs. 3.1-3.2 take single-view inputs stacked in the temporal dimension and predict, for each target view, a soft segmentation mask with class "probabilities" for each bin. For instance, the output of a model processing only the RA view is $f_\theta(\mathbf{x}^{\mathrm{RA}}) = \mathbf{p}^{\mathrm{RA}} \in [0,1]^{M \times N \times K}$, if $M \times N$ is the size of the view and K the number of classes. Our architectures, detailed in Sec. 3.3, take instead multi-view inputs: either $\mathbf{x} = (\mathbf{x}^{\mathrm{RD}}, \mathbf{x}^{\mathrm{RA}})$ or $\mathbf{x} = (\mathbf{x}^{\mathrm{RD}}, \mathbf{x}^{\mathrm{AD}}, \mathbf{x}^{\mathrm{RA}})$. In both cases, their goal is to predict soft masks $\mathbf{p} = (\mathbf{p}^{\mathrm{RD}}, \mathbf{p}^{\mathrm{RA}})$ for both RD and RA views.

The following section details the loss functions applied to each segmented view to train the proposed architectures. We also introduce a 'coherence' loss to enforce consistency between the predictions over the two views of the scene. Finally a combination of these loss terms is proposed.

Weighted Cross Entropy. Semantic segmentation models that predict a score for each class at each pixel are usually trained by minimising a Cross-Entropy (CE) loss function. This loss is not ideal for unbalanced segmentation tasks such as the radar semantic segmentation, since the optimisation process tends to focus on the classes that are most represented. In the present case, background and speckle noise dominate, in comparison to the signatures of the objects we wish to detect. In the same manner as RSS-Net [22], we employ a weighted Cross-Entropy (wCE) loss to tackle this issue.

Given a training example $\mathbf{x}$, let $\mathbf{y} \in \{0,1\}^{M \times N \times K}$ be its one-hot ground truth and $f_\theta(\mathbf{x}) = \mathbf{p} \in [0,1]^{M \times N \times K}$ the associated prediction. The wCE loss function is defined as:

$$\mathcal{L}_{\mathrm{wCE}}(\mathbf{y}, \mathbf{p}) = -\frac{1}{K} \sum_{k=1}^{K} w_k \sum_{(m,n) \in \Omega} \mathbf{y}[m,n,k] \log \mathbf{p}[m,n,k],$$
(2)

where $\Omega = [\![1,M]\!] \times [\![1,N]\!]$, and w_k's are normalized positive weights. Weight w_k is inversely proportional to the frequency of class k in the training set, that is $w_k \propto \left(\sum_{\mathbf{y}} \sum_{(m,n) \in \Omega} \mathbf{y}[m,n,k] \right)^{-1}$. The fewer the bins with ground-truth class k, the larger w_k becomes.

Soft Dice. Object signatures in radar representations often correspond to small regions. This is a well known issue in medical image segmentation, where the Dice metric (detailed in Section 4.1) is usually reformulated in a function called Dice loss, ranging between 0 and 1. Milletari *et al.* [31] have proposed the Soft Dice (SDice) loss defined as:

$$\mathcal{L}_{\mathrm{SDice}} = \frac{1}{K} \sum_{k=1}^{K} \left[1 - \frac{2 \sum_{(m,n)} \mathbf{y}[m,n,k] \mathbf{p}[m,n,k]}{\sum_{(m,n)} \mathbf{y}^2[m,n,k] + \mathbf{p}^2[m,n,k]} \right],$$
(3)

where $(m,n) \in \Omega$, as in Eq. 2. This formulation has proved useful for 2D and 3D medical image semantic segmentation, including for small objects.

Coherence. The objective of multi-view radar semantic segmentation is to simultaneously segment several views of the aggregated RAD tensor. The objects we wish to detect are observed in the different radar views, thus it is clear that a certain coherence must be maintained between the segmented views. For example, one view should not represent a pedestrian, while another represents a cyclist. We introduce a coherence loss (CoL) to preserve a consistency between the predictions of the model. The procedure to calculate this loss is illustrated in Supplementary Material.

Let $(\mathbf{p}^{\mathrm{RD}}, \mathbf{p}^{\mathrm{RA}})$ be the segmentation maps predicted by the model f_θ after the softmax operation. These two maps are aggregated by applying a $\max(.)$ operator along the axis that they do not share (either the Doppler or the angle). The two resulting maps of same size, denoted $\tilde{\mathbf{p}}^{\mathrm{RD}}$ and $\tilde{\mathbf{p}}^{\mathrm{RA}}$, contain the highest probability of each range bin for each class. In other words, they indicate if the model predicts a high probability to observe a category at a given distance. The coherence loss is the mean squared error (MSE) between these maximum range probability vectors. It encourages the network to predict high probability values at the same distance and in the same class for both views. The CoL, in the interval $[0, 1]$, is defined as:

$$\mathcal{L}_{\mathrm{CoL}}(\mathbf{p}^{\mathrm{RD}}, \mathbf{p}^{\mathrm{RA}}) = \left\| \tilde{\mathbf{p}}^{\mathrm{RD}} - \tilde{\mathbf{p}}^{\mathrm{RA}} \right\|_{\mathrm{F}}^2,$$
(4)

where $\| \cdot \|_{\mathrm{F}}$ denotes the Frobenius norm.

Combination of losses. The CE loss is specialised in pixel-wise classification and does not consider spatial correlations between the predictions. The SDice is particularly effective for shape segmentation, but it is difficult to optimise as a single loss function due to its gradient formulation. Finally, the CoL is useful where neither the CE nor the SDice is able to leverage a coherence between the prediction of the RD and the RA views. To combine the different strengths of these losses, we propose the following final loss to train multi-view architectures:

$$\mathcal{L} = \lambda_{\mathrm{wCE}}(\mathcal{L}_{\mathrm{wCE}}^{\mathrm{RD}} + \mathcal{L}_{\mathrm{wCE}}^{\mathrm{RA}}) + \lambda_{\mathrm{SDice}}(\mathcal{L}_{\mathrm{SDice}}^{\mathrm{RD}} + \mathcal{L}_{\mathrm{SDice}}^{\mathrm{RA}}) + \lambda_{\mathrm{CoL}}\mathcal{L}_{\mathrm{CoL}},$$
(5)

where λ_{wCE}, λ_{SDice} and λ_{CoL} are weighting factors set empirically.

4. Experiments

We present in this section the experimental evaluation of our models. We describe first the datasets and the evaluation metrics that we utilize. We then explain the modifications made to the methods we compare to. Finally, we give details concerning the experiments and analyse their results quantitatively and qualitatively.

4.1. Datasets and evaluation metrics

Datasets. The CARRADA dataset [34] contains synchronised camera and automotive radar recordings with 30 sequences of various scenarios with one or two moving objects. The radar views are annotated using a semi-automatic pipeline. This is the only publicly-available dataset providing RAD tensors and dense semantic segmentation annotation for both RD and RA views. The objects are separated into four categories: *pedestrian*, *cyclist*, *car* and *background*. The provided RAD tensors have dimensions $256 \times 256 \times 64$. The experiments presented in Section 4.3 use the dataset splits proposed by the authors, denoted CARRADA-Train, CARRADA-Val and CARRADA-Test.

For qualitative evaluation only, we also employ in-house sequences of more complex urban scenes with synchronised camera and radar data. For these sequences, the RAD tensors have the same dimensions as in CARRADA while the resolution in range is divided by two. The radar views are not annotated, which does not allow quantitative evaluation.

Evaluation metrics. A classic performance metric in semantic segmentation is the intersection over union (IoU): Given annotated test inputs, the IoU for a given class is defined as the percentage $\frac{|A \cap B|}{|A \cup B|}$, where A is the set of locations predicted as stemming from this class and B is the ground-truth set of locations for the same class. From the perspective of a single object in a given scene, the IoU measures how well and how completely it is segmented. Averaging this metric over all classes yields the mean IoU (mIoU) score. Another related, yet slightly different metric, is provided by the Dice score: For a given class and with same notations as above, it is defined as $\frac{2|A \cap B|}{|A|+|B|}$. For global performance, it is averaged over all classes into the mean Dice (mDice). Seeing segmentation as a local 1-*vs.*-all classification problem for each class, the Dice amounts to the harmonic mean of the precision and recall (a.k.a. F1 score). The IoU and Dice metrics are considered as complementary; We report both of them in our experiments.

4.2. Implementation of competing methods

This section describes the architectures we use for comparisons. For each method, one model is trained specifically
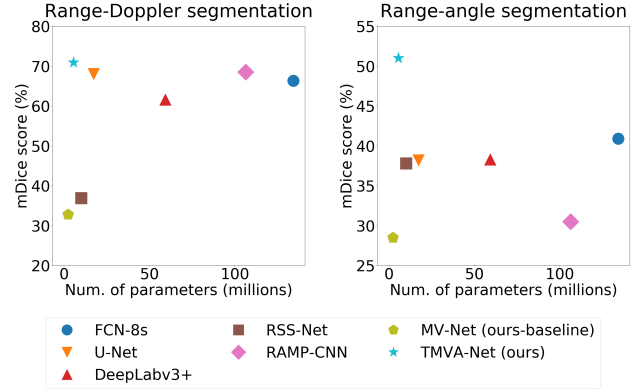

Figure 3. **Performance-*vs.*-complexity plots for all methods in RD and RA tasks.** Performance is assessed by mDice (%) and complexity by the number of parameters (in millions) *for a single task*. Top-left models are the best performing and the lightest. Only our models, MV-Net and TMVA-Net, are able to segment both views simultaneously. For all the other methods, two distinct models must be trained to address both tasks, which doubles the number of actual parameters.

for single-view semantic segmentation of either RD or RA. Details concerning pre-processing procedures are provided in the Supplementary Material.

The non-radar-based architectures have been used "as is": The FCN-8s architecture is based on a VGG16 [46] backbone; DeepLabv3+ uses a ResNet-101 [20]; The U-Net architecture is identical to the one in [40].

In the experiments with RSS-Net, the number of downsampling layers in the encoding part has been reduced to be trained with lower resolution inputs.

The RAMP-CNN architecture dedicated to the RD view has been adapted with two major changes. Firstly, the fusion module has been modified to aggregate and duplicate the feature maps to suit the range-Doppler space. Secondly, the size of the output feature maps has been reduced on the Doppler axis using an additional convolutional layer with 3×3 filters and a stride factor of 4. For both RD and RA segmentation tasks, an additional 1D convolutional layer with a softmax operation processes the last feature maps to predict segmentation maps.

4.3. Training and results

Training procedures. Methods presented in Section 3 are trained using CARRADA-Train and CARRADA-Val splits and tested on the CARRADA-Test. At each timestamp of a radar sequence, the provided RAD tensor is processed according to the method presented in Section 2.1.

For each frame, q past frames are also considered for both training and testing phases: The views from $t - q$ to t are stacked into the time-t input (Fig. 1). For the methods that do not explicitly process the time dimension, $q = 2$ for a total input sequence length of 3. Time-based methods

View	Method	# Param. (M)	IoU (%)					Dice (%)				
			Bkg.	Ped.	Cyclist	Car	mIoU	Back.	Ped.	Cyclist	Car	mDice
RD	FCN-8s [27]	134.3	99.7	47.7	18.7	52.9	54.7	99.8	24.8	16.5	26.9	66.3
	U-Net [40]	17.3	99.7	51.0	33.4	37.7	55.4	99.8	67.5	50.0	54.7	68.0
	DeepLabv3+ [10]	59.3	99.7	43.2	11.2	49.2	50.8	99.9	60.3	20.2	66.0	61.6
	RSS-Net	10.1	99.3	0.1	4.1	25.0	32.1	99.7	0.2	7.9	40.0	36.9
	RAMP-CNN	106.4	99.7	48.8	23.2	54.7	56.6	99.9	65.6	37.7	70.8	68.5
	MV-Net (ours-baseline)	2.4*	98.0	0.0	3.8	14.1	29.0	99.0	0.0	7.3	24.8	32.8
	TMVA-Net (ours)	5.6*	99.7	52.6	29.0	53.4	58.7	99.8	68.9	45.0	69.6	70.9
RA	FCN-8s [27]	134.3	99.8	14.8	0.0	23.3	34.5	99.9	25.8	0.0	37.8	40.9
	U-Net [40]	17.3	99.8	22.4	8.8	0.0	32.8	99.9	36.6	16.1	0.0	38.2
	DeepLabv3+ [10]	59.3	99.9	3.4	5.9	21.8	32.7	99.9	6.5	11.1	35.7	38.3
	RSS-Net	10.1	99.5	7.3	5.6	15.8	32.1	99.8	13.7	10.5	27.4	37.8
	RAMP-CNN	106.4	99.8	1.7	2.6	7.2	27.9	99.9	3.4	5.1	13.5	30.5
	MV-Net (ours-baseline)	2.4*	99.8	0.1	1.1	6.2	26.8	99.0	0.0	7.3	24.8	28.5
	TMVA-Net (ours)	5.6*	99.8	26.0	8.6	30.7	41.3	99.9	41.3	15.9	47.0	51.0

Table 1. **Semantic segmentation performance on the CARRADA-Test dataset for range-Doppler (RD) and range-angle (RA) views**. The number of trainable parameters (in millions) for each method corresponds to a single view-segmentation model; Two such models, one for each view, are required for all methods but ours. In contrast, the number of parameters reported for our methods ('*') corresponds to a single model that segments both RD and RA views. The RSS-Net and RAMP-CNN methods have been modified to be trained on both tasks (see Sec. 4.2). Performances are evaluated with the Intersection over Union ('IoU') and the Dice score per class, and their averages, 'mIoU' and 'mDice', over the four classes. The best scores are in red and bold type, the second best in blue and underlined.

using 3D convolutions have specific sequence lengths: $q = 8$ for RAMP-CNN and $q = 4$ for TMVA-Net.

The competing architectures have been trained with the CE loss, except for the RSS-Net, which is trained with a wCE using the formulation in [22]. Our methods are trained using the combination of loss terms detailed in Section 3.4. We used our formulation of the wCE loss (Section 3.4) with weights computed on CARRADA-Train.

All the training procedures use the Adam optimiser [24] with the recommended parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$). Since each method has its own set of hyper-parameters, further details are provided in the Supplementary Material, namely batch sizes, learning rates, learning rate decays, numbers of epochs and corresponding pre-processing steps for each one of them. Training was performed using the PyTorch framework with a single GeForce RTX 2080 Ti graphic card.

Quantitative results. The performance for both RD and RA semantic segmentation tasks on CARRADA-Test are shown in Table 1. Our proposed TMVA-Net achieves the best scores for both mDice and mIoU metrics and for both segmentation tasks. Moreover, our methods are the only ones to perform both tasks simultaneously. TMVA-Net also provides the best trade-off between performance and number of parameters for both tasks, as illustrated in Figure 3 with mDice metric (similar plots with mIoU metric are presented in Supplementary Material). Note that the number of parameters reported for each method in Table 1, in Figure 3 and in Supplementary Material corresponds to a single trained model, while competing methods require two independent models (hence twice more parameters) to perform

View	Method	# Param.	mIoU	mDice
RD	MV-Net (baseline)	2.4M	29.0	32.8
	MVA-Net (a)	3.6M	48.9	60.4
	MVA-Net (b)	4.8M	52.9	64.3
	TMVA-Net	5.6M	**59.3**	**71.5**
RA	MV-Net (baseline)	2.4M	26.8	28.5
	MVA-Net (a)	3.6M	28.1	31.1
	MVA-Net (b)	4.8M	36.7	43.9
	TMVA-Net	5.6M	**40.1**	**49.3**

Table 2. **Ablation study of the proposed architectures.** Each architecture has been trained using the wCE+SDice combination loss. TMVA-Net delivers the best performances under both mIoU and mDice metrics and for both RD and RA views.

both RD and RA segmentation tasks.

Ablation Studies. We report in Table 2 the performance of the four architectures we have introduced in Section 3.3. It shows that the additional ASPP modules in MVA-Net(a) boosts the performance relative to MV-Net for both RD and RA segmentation. The performance is further improved with MVA-Net(b) by the additional encoder that extracts features from the AD view and provides relevant information to separate object signatures. Finally, TMVA-Net is the most effective regardless of the metric, thanks to its ability to learn spatio-temporal features with 3D convolutions. The temporal dimension indeed helps distinguish between objects and the speckle noise, and to categorise them according to the shape variations.

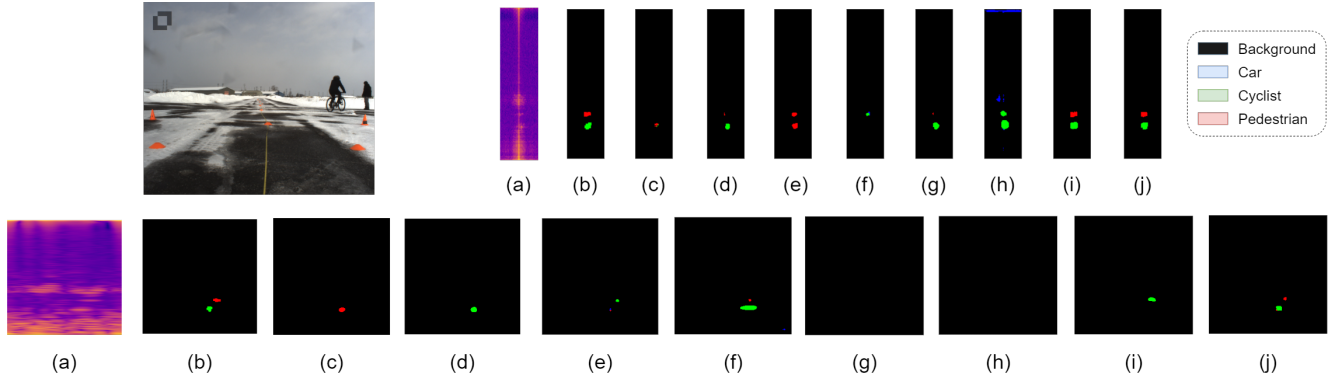We assess in Table 3 the importance of the different

Figure 4. **Qualitative results on a test scene of CARRADA**. (*Top*) camera image of the scene and results of the RD segmentation; (*Bottom*) Results of the RA Segmentation. (a) Radar view signal, (b) ground-truth mask, (c) FCN8s, (d) U-Net, (e) DeepLabv3+, (f) RSS-Net, (g) RAMP-CNN, (h) MV-Net (our baseline w/ wCE+SDice loss), (i) TMVA-Net (ours, w/ wCE+SDice loss), (j) TMVA-Net (ours, w/ wCE+SDice+CoL loss).

	RD view		RA view	
Loss	mIoU	mDice	mIoU	mDice
CE	56.1	67.8	39.1	48.3
SDice	58.5	70.3	37.1	44.8
wCE	51.1	62.8	34.3	41.1
CE+SDice	45.2	54.0	38.8	46.9
wCE+SDice	**59.3**	**71.5**	<u>40.1</u>	<u>49.3</u>
wCE+SDice+CoL	<u>58.7</u>	<u>70.9</u>	**41.3**	**51.0**

Table 3. **Ablation study of the combination of losses.** Each individual or combination of loss(es) is used to train a TMVA-Net model. Our proposed combination (wCE+SDice+CoL) reaches the best mIoU and mDice for the RA view and the second best scores for the RD view.

losses on the performance of TMVA-Net. The best combination of two loss terms is wCE+SDice for both tasks. The performance is further improved on the RA segmentation task by adding the proposed CoL term, while slightly reduced on RD views. This loss improves the coherence between the tasks by better detecting objects in the range-angle views as discussed in the following section.

Qualitative results. We show in Figure 4 qualitative results of each method on a scene from CARRADA-Test. The results of TMVA-Net (i-j) display well segmented RD views in terms of localisation and classification. Only TMVA-Net with CoL (j) is able to localise and classify both objects in the RD and RA views. The enforcement of the coherence of predictions across views succeeds in correctly classifying the same objects in the two views. This is not the case for TMVA-Net without CoL, as illustrated in the example (i), where the model predicts a cyclist instead of a pedestrian in the RA view. Moreover, the coherence loss also helps to discover new objects: In (i), TMVA-Net predicts a single

object in the RA view, while in (j), it localises and classifies both objects well with the help of CoL.

Qualitative results on additional urban scenes, which are not part of CARRADA, are provided in the Supplementary Material. They show that our methods unlike others can generalise well on RD and RA views. Indeed, TMVA-Net succeeds in learning object signatures on the CARRADA dataset and recognizing them in a different environment.

5. Conclusions and perspectives

In this work, we propose lightweight architectures for multi-view radar semantic segmentation and a combination of loss terms to train them. Our methods localise and delineate objects in the radar scene while simultaneously determining their relative velocity. We show that both the information from the RAD radar tensor and from its temporal evolution are important to conduct these tasks. Our methods significantly outperform competing architectures specialised either in natural image semantic segmentation or in radar scene understanding. Preliminary experiments also show qualitatively that they generalize better to new complex urban scenes.

We hope that our work will encourage the community to release larger datasets to emphasize the importance of the radar sensor and to explore new architectures for radar semantic segmentation.

Our future investigations will focus on improving the segmentation of cyclists and pedestrians, which remain difficult to distinguish. Exploiting radar properties could be interesting to improve both RAD tensor aggregation and class-specific data augmentation methods for the benefit of our learning algorithms.

References

[1] Roberto Aldera, Daniele De Martini, Matthew Gadd, and Paul Newman. What could go wrong? Introspective radar odometry in challenging environments. In *ITSC*, 2019. 2

[2] Mostafa Alizadeh, Hajar Abedi, and George Shaker. Low-cost low-power in-vehicle occupant detection with mm-wave fmcw radar. In *Sensors*, 2019. 2

[3] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford Radar RobotCar dataset: a radar extension to the Oxford RobotCar dataset. In *ICRA*, 2020. 1

[4] Daniel Brodeski, Igal Bilik, and Raja Giryes. Deep radar detector. In *RadarConf*, 2019. 3

[5] Graham Brooker. Understanding millimetre wave FMCW radars. In *ICST*, 2005. 2

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: a multimodal dataset for autonomous driving. In *CVPR*, 2020. 1

[7] Samuele Capobianco, Luca Facheris, Fabrizio Cuccoli, and Simone Marinai. Vehicle classification based on convolutional networks applied to FMCW radar signals. In *TRAP*, 2018. 2

[8] Hou Changbo, Hua Lijie, Liu Guowei, and Lin Yun. Radar signal separation and recognition based on semantic segmentation. In *DSA*, 2020. 2

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. In *TPAMI*, 2017. 3, 4

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 3, 4, 7

[11] Andreas Danzer, Thomas Griebel, Martin Bach, and Klaus Dietmayer. 2D car detection in radar data with PointNets. In *ITSC*, 2019. 2

[12] B. Dekker, S. Jacobs, A.S. Kossen, M.C. Kruithof, A.G. Huizing, and M. Geurts. Gesture recognition with a low power FMCW radar and a deep convolutional neural network. In *EURAD*, 2017. 2

[13] Charles-Alban Deledalle, Loic Denis, Sonia Tabti, and Florence Tupin. MuLoG, or how to apply gaussian denoisers to multi-channel SAR speckle reduction? In *Trans. Image Process.*, 2017. 2

[14] Xu Dong, Pengluo Wang, Pengyue Zhang, and Langechuan Liu. Probabilistic oriented object detection in automotive radar. In *CVPR Worshops*, 2020. 2

[15] Xiangyu Gao, Guanbin Xing, Sumit Roy, and Hui Liu. Experiments with mmWave automotive radar test-bed. In *ACSSC*, 2019. 1

[16] Xiangyu Gao, Guanbin Xing, Sumit Roy, and Hui Liu. RAMP-CNN: a novel neural network for enhanced automotive radar object recognition. In *Sensors*, 2020. 3, 4

[17] Antoine Ghaleb. *Micro-Doppler analysis of non-stationary moving targets in radar imaging*. PhD thesis, Telecom Paris, 2009. 2

[18] J. W. Goodman. Some fundamental properties of speckle. In *Optical Society of America*, 1976. 2

[19] Souvik Hazra and Avik Santra. Radar gesture recognition system in presence of interference using self-attention neural network. In *ICMLA*, 2019. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[21] F. Jin, A. Sengupta, S. Cao, and Y. Wu. Mmwave radar point cloud segmentation using gmm in multimodal traffic monitoring. In *RADAR*, 2020. 2

[22] Prannay Kaul, Daniele De Martini, Matthew Gadd, and Paul Newman. RSS-Net: weakly-supervised multi-class semantic segmentation with FMCW radar. In *IV*, 2020. 2, 3, 4, 5, 7

[23] Youngwook Kim and Brian Toomajian. Hand gesture recognition using micro-Doppler signatures with convolutional neural network. In *IEEE Access*, 2016. 2

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 7

[25] Wentai Lei, Xinyue Jiang, Long Xu, Jiabin Luo, Mengdi Xu, and Feifei Hou. Continuous gesture recognition based on time sequence fusion using MIMO radar sensor and deep learning. In *Electronics*, 2020. 2

[26] Jakob Lombacher, Kilian Laudt, Markus Hahn, Jurgen Dickmann, and Christian Wohler. Semantic radar grids. In *IV*, 2017. 2

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3, 7

[28] Bence Major, Daniel Fontijne, Amin Ansari, Ravi Teja Sukhavasi, Radhika Gowaikar, Michael Hamilton, Sean Lee, Slawomir Grzechnik, and Sundar Subramanian. Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In *ICCV Workshops*, 2019. 3

[29] Michael Meyer and Georg Kuschk. Automotive radar dataset for deep learning based 3D object detection. In *EuRAD*, 2019. 1

[30] Michael Meyer and Georg Kuschk. Deep learning based 3d object detection for automotive radar and camera. In *EuRAD*, Oct 2019. 2

[31] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 5

[32] Weichong Ng, Guohua Wang, Siddhartha, Zhiping Lin, and Bhaskar Jyoti Dutta. Range-Doppler detection in automotive radar with deep learning. In *IJCNN*, 2020. 2

[33] F. E. Nowruzi, D. Kolhatkar, Prince Kapoor, E. J. Heravi, R. Laganiere, Julien Rebut, and Waqas Malik. Deep open space segmentation using automotive radar. In *ICMIM*, 2020. 1

[34] A. Ouaknine, A. Newson, J. Rebut, F. Tupin, and P. Pérez. CARRADA dataset: camera and automotive radar with range-angle-Doppler annotations. In *ICPR*, 2020. 1, 3, 6

[35] Andras Palffy, Jiaao Dong, Julian F. P. Kooij, and Dariu M. Gavrila. CNN based road user detection using the 3D radar cube. In *RAL*, 2020. 3

[36] Kanil Patel, Kilian Rambach, Tristan Visentin, Daniel Rusev, Michael Pfeiffer, and Bin Yang. Deep learning-based object classification on automotive radar spectra. In *RadarConf*, 2019. 2

[37] Robert Prophet, Anastasios Deligiannis, Juan-Carlos Fuentes-Michel, Ingo Weber, and Martin Vossiek. Semantic segmentation on 3D occupancy grids for automotive radar. In *IEEE Access*, 2020. 2

[38] Robert Prophet, Gang Li, Christian Sturm, and Martin Vossiek. Semantic segmentation on automotive radar maps. In *IV*, 2019. 2

[39] Hermann Rohling. Radar CFAR thresholding in clutter and multiple target situations. In *T-AES*, 1983. 2

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 3, 6, 7

[41] Nicolas Scheiner, Florian Kraus, Fangyin Wei, Buu Phan, Fahim Mannan, Nils Appenrodt, Werner Ritter, Jürgen Dickmann, Klaus Dietmayer, Bernhard Sick, and Felix Heide. Seeing around street corners: non-line-of-sight detection and tracking in-the-wild using Doppler radar. In *CVPR*, 2020. 2

[42] Moritz Scherer, Michele Magno, Jonas Erb, Philipp Mayer, Manuel Eggimann, and Luca Benini. TinyRadarNN: combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars. In *IoT*, 2020. 2

[43] Ole Schumann, Jakob Lombacher, Markus Hahn, Christian Wohler, and Jurgen Dickmann. Scene understanding with automotive radar. In *IV*, 2020. 2

[44] Yogesh Shankar, Souvik Hazra, and Avik Santra. Radar-based non-intrusive fall motion recognition using deformable convolutional neural network. In *ICMLA*, 2019. 2

[45] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. RADIATE: a radar dataset for automotive perception. In *ArXiv*, 2020. 1

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

[47] Liat Sless, Gilad Cohen, Bat El Shlomo, and Shaul Oron. Road scene understanding by occupancy grid learning from sparse radar clusters using semantic segmentation. In *ICCV Workshops*, 2019. 2

[48] Yuliang Sun, Tai Fei, Shangyin Gao, and Nils Pohl. Automatic radar-based gesture detection and classification via a region-based deep convolutional neural network. In *ICASPP*, 2019. 2

[49] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with Soli: exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *UIST*, 2016. 2

[50] Yong Wang, Xiuqian Jia, Mu Zhou, Xiaolong Yang, and Zengshan Tian. Rammar: RAM assisted mask R-CNN for FMCW sensor based HGD system. In *ICC*, 2019. 2

[51] Yizhou Wang, Zhongyu Jiang, Yudong Li, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. RODNet: a real-time radar object detection network cross-supervised by camera-radar fused object 3D localization. In *SP*, 2021. 2

[52] Guoqiang Zhang, Haopeng Li, and Fabian Wenger. Object detection and 3d estimation via an FMCW radar using a fully convolutional network. In *ICASSP*, 2020. 2

[53] Zhenyuan Zhang, Zengshan Tian, Ying Zhang, Mu Zhou, and Bang Wang. u-DeepHand: FMCW radar-based unsupervised hand gesture feature learning using deep convolutional auto-encoder network. In *Sensors*, 2019. 2

[54] Zhenyuan Zhang, Zengshan Tian, and Mu Zhou. Latern: dynamic continuous hand gesture recognition using FMCW radar sensor. In *Sensors*, 2018. 2

[55] JianPing Zhu, HaiQuan Chen, and WenBin Ye. Classification of human activities based on radar signals using 1D-CNN and LSTM. In *ISCAS*, 2020. 2