Raw High-Definition Radar for Multi-Task Learning

Julien Rebut[1] Arthur Ouaknine[1,2] Waqas Malik[3] Patrick Pérez[1]
1: Valeo.ai, Paris, France 2: Télécom Paris, Paris, France 3: Valeo North America Inc., San Mateo, CA

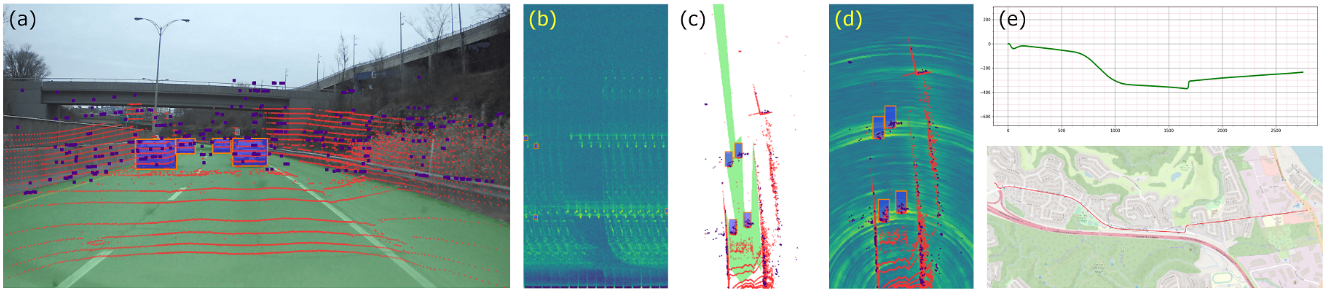

Figure 1. **Overview of our RADIal dataset**. RADIal includes a set of 3 sensors (camera, laser scanner, high-definition radar) and comes with GPS and vehicle's CAN traces; 25k synchronized samples are recorded is raw format. (a) Camera image with projected laser point cloud in red and radar point cloud in indigo, vehicle annotation in orange and free-driving-space annotation in green; (b) Radar power spectrum with bounding box annotations; (c) Free-driving-space annotation in bird-eye view, with vehicles annotated with orange bounding boxes, radar point cloud in indigo and laser point cloud in red; (d) Range-azimuth map in Cartesian coordinates overlayed with radar point cloud and laser point cloud; (e) GPS trace in red and odometry trajectory reconstruction in green.

Abstract

With their robustness to adverse weather conditions and ability to measure speeds, radar sensors have been part of the automotive landscape for more than two decades. Recent progress toward High Definition (HD) Imaging radar has driven the angular resolution below the degree, thus approaching laser scanning performance. However, the amount of data a HD radar delivers and the computational cost to estimate the angular positions remain a challenge. In this paper, we propose a novel HD radar sensing model, FFT-RadNet, *that eliminates the overhead of computing the range-azimuth-Doppler 3D tensor, learning instead to recover angles from a range-Doppler spectrum. FFT-RadNet is trained both to detect vehicles and to segment free driving space. On both tasks, it competes with the most recent radar-based models while requiring less compute and memory. Also, we collected and annotated 2-hour worth of raw data from synchronized automotive-grade sensors (camera, laser, HD radar) in various environments (city street, highway, countryside road). This unique dataset, nick-named* RADIal *for "Radar, LiDAR* et al.*", is available at* `https://github.com/valeoai/RADIal`.

1. Introduction

Automotive radars have been in production since the late 90s. They are the preferred, most affordable sensors for adaptive cruise control, blind spot detection and automatic emergency braking functions. However, they have a poor angular resolution, which hinders their use in automated driving systems. Indeed, such systems need a high level of safety and robustness, usually reached through redundancy mechanisms. While sensing is improved by fusing several modalities, the overall combination works only if each sensor achieves sufficient and comparable performances. High-definition (HD) imaging radar has emerged to meet these requirements. By using dense virtual antenna arrays, these new sensors achieve high angular resolution both in azimuth and elevation (horizontal and vertical angular positions, resp.) and produce denser point clouds.

With the rapid progress of deep learning and the availability of public driving datasets, *e.g.*, [4, 6, 12], the perception ability of vision-based driving systems (detection of objects, structures, markings and signs, estimation of depth, forecasting of other road users' movements) has considerably improved. These advances quickly extended to depth sensors such as laser scanners (LiDAR), with the help of specific architectures to deal with 3D point clouds [19, 42].

Dataset	Year	Scale	Radar data					Radar type	Modalities	Sequence	Annot. type
			ADC	RAD	RA or RD	PC	Doppler				
nuScenes [4]	2019	large	✗	✗	✗	✓	✓	LD	CLO	✓	3D boxes
Astyx [24]	2019	small	✗	✗	✗	✓	✓	HD	CL	✗	3D boxes
RadarRobotCar [1]	2020	large	✗	✗	✓	✗	✗	S	CLO	✓	✗
CARRADA [31]	2020	small	✗	✓	✓	✓	✓	LD	C	✓	2D boxes, seg.
RADIATE [38]	2020	medium	✗	✗	✓	✗	✗	S	CLO	✓	2D boxes
MulRan [17]	2020	medium	✗	✗	✓	✓	✗	S	CLO	✓	✗
Zendar [27]	2020	small	✗	✗	✓	✓	✓	HD	CL	✓	2D boxes
CRUW [41]	2021	medium	✗	✗	✓	✗	✗	LD	C	✓	point location
RadarScenes [36]	2021	large	✗	✗	✗	✓	✓	HD	CO	✓	point-wise
RADDet [43]	2021	small	✗	✓	✓	✗	✓	LD	C	✓	2D boxes
RADIal (ours)	2022	medium	✓	✓	✓	✓	✓	HD	CLO	✓	2D boxes, seg.

Table 1. **Publicly-available driving datasets with radar**. The dataset is 'small' (<15k frames), 'large' (>130k frames) or 'medium' (in between). The radar is low-definition ('LD'), high-definition ('HD') or scanning ('S') and its data is released in different representations, amounting to different signal processing pipelines: analog-to-digital converter ('ADC') signal, range-azimuth-Doppler ('RAD') tensor, range-azimuth ('RA') view, range-Doppler ('RD') view, point cloud ('PC'). The presence of Doppler information depends on the radar sensor. Other sensor modalities are camera ('C'), LiDAR ('L') and odometry ('O'). RADIal is the only dataset providing each representation of a HD radar, combined with camera, LiDAR and odometry, while proposing detection and free-space segmentation tasks.

Quite surprisingly, the adoption of deep learning for radar processing in this context is much slower, compared to the other sensors. This might be explained by the complex nature of the data and the lack of public datasets. Indeed, recent key contributions in the field of radar-based vehicle's perception have appeared together with the release of datasets. Interestingly, most of the recent works exploit the range-azimuth (RA) representation of the radar data (either in polar or Cartesian coordinates). Similar to a bird's eye view (see Figure 1d), this representation is easy to interpret and allows simple data augmentation with translations and rotations. However, one barely-mentioned drawback is that the generation of the RA radar maps incurs significant processing costs (tens of GOPS, see Section 6.5), which compromises its viability on embedded hardware. While novel HD radars offer better resolution, they make this computational complexity issue even more acute.

Owing to the promising capabilities of HD radars, our work attacks this issue to improve their practicality. In particular, we propose: (1) FFT-RadNet, an optimized deep architecture that processes HD radar data at reduced cost, toward two different perception tasks, namely vehicle detection and free-space segmentation; (2) An empirical analysis comparing various radar signal representations in terms of performance, complexity and memory footprint; (3) RADIal, the first raw HD radar dataset, including several other automotive-grade sensors, as described in Table 1.

The paper is organized as follows: Sections 2 and 3 discuss radar background and related work; FFT-RadNet and RADIal are introduced in Sections 4 and 5 resp.; Experiments are reported in Section 6, and Section 7 concludes.

2. Radar background

Radars are usually composed of a set of transmitting and receiving antennas. The transmitters emit electromagnetic waves which are reflected back to the receivers by the objects in the environment. Standard in the automotive industry [3, 13], a frequency-modulated continuous-wave (FMCW) radar emits a sequence of frequency-modulated signals called chirps. The frequency difference between the emission and reception is mostly due to the radial distance of the obstacle. This distance is thus extracted via a Fast Fourier Transform (FFT) along the chirp sequence (range-FFT). A second FFT (Doppler-FFT) along the time axis extracts the phase difference, which captures the radial velocity of the reflector. The combination of these 2 FFTs provides a range-Doppler (RD) spectrum for each receiving antenna (Rx), stored for all Rx in an RD tensor. The angle-of-arrival (AoA) can be estimated by using more than one Rx. A phase difference in the received signal is observed due to the small distance between Rx antennas. A common practice is to apply a third FFT (angle-FFT) along the channel axis to estimate this AoA.

Radar's capability to discriminate between two targets with same range and velocity but different angles is called its angular resolution. It is directly proportional to the antenna aperture, that is, the distance between the first and last antennas. The multiple inputs multiple outputs (MIMO) approach [9] is commonly used to improve the angular resolution without increasing the physical aperture: Angular resolution increases by a factor of 2 for each added emitting antenna (Tx). Denoting N_{Tx} and N_{Rx} the number of its Tx and Rx channels respectively, a MIMO system builds a virtual array of $N_{\text{Tx}} \cdot N_{\text{Rx}}$ antennas. In order to prevent emitted signals from interfering, the transmitters emit the same signal at the same time, but with a slight phase shift Δ_ϕ between two consecutive antennas. The downside of this approach is that the signature of each reflector appears N_{Tx} times in the RD spectrum, making the data interleaved.

To translate the AoA into an effective angle, one needs to calibrate the sensor. An alternative to the third FFT is to correlate in the complex domain the RD spectrum with a calibration matrix, to estimate the angles (azimuth and elevation). The complexity of this operation for a single point of the RD tensor is $\mathcal{O}(N_{\text{Tx}} N_{\text{Rx}} B_{\text{A}} B_{\text{E}})$, where B_{A} and B_{E} are the numbers of discretization bins for azimuth and elevation angles respectively in the calibration matrix. For a 4D representation in range-azimuth-elevation-Doppler, this operation would need to be performed for each point of the RD tensor.[1]

As a conclusion, for an embedded HD radar, traditional signal processing can not be applied as it is too resource greedy in terms of both computation requirements and memory footprint. For driving assistance systems, there is therefore the challenge of increasing radar's angular resolution while keeping the processing costs under control.

3. Related work

Radar datasets. Traditional radars offer a good trade-off between cost and performance. While they provide accurate range and velocity, they suffer from a low azimuth resolution leading to ambiguity in separating close objects. Recent datasets include processed radar representations such as the entire range-azimuth-Doppler (RAD) tensor [31, 43] or single views of this tensor –either range-azimuth (RA) [1, 17, 27, 38, 41] or range-Doppler (RD) [27]. These representations require large bandwidth to be transmitted as well as large memory storage. For this reason, datasets that include several modalities with numerous samples, such as nuScenes [4], provide only radar point clouds, a lighter representation. However, it is a limited processed representation and it is biased to the signal processing pipeline. Several other datasets use a 360° scanning radar [1, 17, 38]. However, its angular resolution is limited as with traditional radars and it does not provide Doppler information.

As discussed earlier, recent HD radars successfully reach an azimuth angular resolution below the degree using large arrays of virtual antennas. The Zendar dataset [27] provides range-Doppler and range-azimuth views for such a radar. Both Astyx [24] and RadarScenes [36] datasets contain HD radar data processed as point clouds.

To the best of our knowledge, there is no open-source HD radar dataset that provides raw data together with camera and LiDAR in various driving environments, a gap that our dataset is filling. Table 1 summarizes the characteristics of the publicly-available driving datasets with radar.

Radar object detection. Low-definition (LD) radar has been used for many applications such as hand gesture recognition [10], object or person detection at gates [15] and aerial monitoring [26]. For automotive applications, single views of the RAD tensor are chosen as input of specific neural network architectures to detect objects' signatures in the considered view, either RA [8, 40] or RD [28]. Differently, [44] uses a radar view to localise objects in the camera image and [2] proposes a two-stage approach to estimate the azimuth of a detected object using only RD views.

Specific architectures have been designed to ingest aggregated views of the RAD tensor to detect objects in the RA view [11, 23]. The entire tensor has also been considered, either for object detection in both RA and RD views [43] or for object localisation in the camera image [32].

A radar point cloud contains less information than RAD views due to the pre-processing that has been applied. However, [7, 35] explore this representation for 2D object detection with LR radar and [25] shows that HD radar point clouds can outperform LiDAR for this task.

None of these works mentions the pre-processing cost to generate the RAD tensor or the point cloud, which are taken as granted. In fact, a HD radar can not be used by previously-mentioned approaches as it would not fit on even the largest automotive embedded device. With [11] for instance applied to HD radar, the input data at each timestamp would occupy 450MB and require $4.5 \cdot 10^{10}$ FLOPS[2] for only one elevation (out of 11). To the best of our knowledge, there is no previous work on end-to-end object detection that is capable to scale with raw HD radar data.

Radar semantic segmentation. Semantic segmentation on radar representation has been less explored due to the lack of annotated datasets. The RA view has been a research topic for multi-class [16] and free-space [29] segmentation. The entire RAD tensor is considered for multi-view segmentation in [30]. Radar point cloud segmentation has also been explored to estimate bird-eye-view occupancy grids, either for LD [22, 39] or HD [33, 34, 37] radars.

Once again, none of these methods can scale to raw HD radar data to perform free-space segmentation for instance. Moreover, there is no previous work either on free-driving-space segmentation or semantic segmentation using only RD views of HD radar signals. In addition, there is no existing multi-task model that performs both radar object detection and semantic segmentation simultaneously. Next, we detail our approach toward reducing both memory and complexity to perform vehicle detection and free-driving-space segmentation with raw HD radar signals.

[1] Considering a HD radar with $0.2°$ of azimuth resolution over $180°$ of horizontal field of view (FoV) and 11 elevations, it would require 498 GFLOPS to be computed.

[2] For comparison, ResNet50 on a 256px image requires $4 \cdot 10^9$ FLOPS.

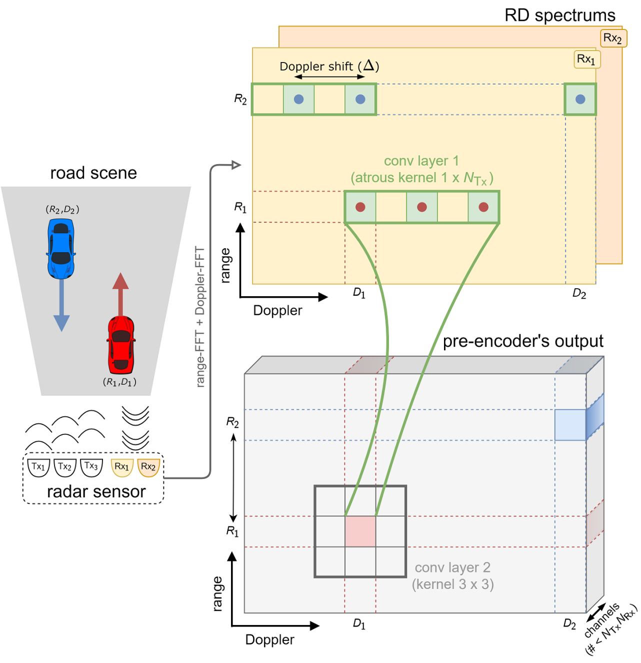

Figure 2. **Trainable MIMO pre-encoder.** Considering three transmitters ($N_{\text{Tx}}=3$) and two receivers ($N_{\text{Rx}}=2$), an object's signature is visible N_{Tx} times in the RD spectrum. The pre-encoder uses Atrous convolutions to organise and compress signatures in fewer than $N_{\text{Tx}}{\cdot}N_{\text{Rx}}$ output channels.

4. FFT-RadNet architecture

Our approach is motivated by automotive constraints: Automotive-grade sensors must be used and only limited processing/memory resources are available on the embedded hardware. In this context, the RD spectrum is the only representation that is practical for HD radar. Based on it, we propose a multi-task architecture, compatible with above requirements, which is composed of five blocks (see Fig.3):

- A pre-encoder reorganizing and compressing the RD tensor into a meaningful and compact representation;
- A shared Feature Pyramidal Network (FPN) encoder combining low-resolution semantic information with high-resolution details;
- A range-angle decoder building a range-azimuth latent representation from the feature pyramid;
- A detection head localizing vehicles in range-azimuth coordinates;
- A segmentation head predicting the free driving space.

4.1. MIMO pre-encoder

As explained in Section 2, the MIMO configuration implies one complex RD spectrum per receiver. This results in a complex 3D tensor of dimension $(B_{\text{R}}, B_{\text{D}}, N_{\text{Rx}})$, where B_{R} and B_{D} are the numbers of discretization bins for range and Doppler respectively. It is important to understand how a given reflecting object, say a car in front, appears in this data. Denote R the actual radial distance of this object to the radar and D its relative radial velocity expressed in Doppler effect. For each receiver, its signature will be visible N_{Tx} times, one per transmitter. More specifically, it will be measured at range-Doppler positions $(R,(D+k\Delta)[D_{\max}])_{k=1\cdots N_{\text{Tx}}}$, where Δ is the Doppler shift (induced by the phase shift Δ_ϕ in the transmitted signal) and $D_{\max}$ is the largest Doppler that can be measured. The measured Doppler values are modulo this maximum.

This signal intricacy calls for a rearrangement of the RD tensor that will facilitate a subsequent exploitation of the MIMO information (to recover angles) while keeping data volume under control. To this end, we propose a new trainable pre-encoder that performs such a compact reorganization of the input tensor (Fig. 2). In order to handle at best its specific structure along the Doppler axis, we use first a suitably-defined Atrous convolution layer that gathers Tx and Rx information at the right positions. The size of its kernel for one input channel is $1\times N_{\text{Tx}}$, hence defined by the number of Tx antennas, and its dilation amounts to $\delta = \frac{\Delta B_{\text{D}}}{D_{\max}}$, the number of Doppler bins corresponding to Doppler shift Δ. The number of input channels is the number N_{Rx} of Rx antennas. A second convolution layer, with a 3×3 kernel, learns how to combine these channels and compresses the signal. The two-layer pre-encoder is trained end-to-end with the rest of the proposed architecture.

4.2. FPN encoder

Using a pyramidal structure to learn multi-scale features is a common practice in object detection [20] and semantic segmentation [45]. Our FPN architecture uses 4 blocks composed of 3, 6, 6 and 3 residual layers [14] respectively. The feature maps of these residual blocks form the feature pyramid. This classic encoder has been optimized considering the nature of the data while controlling its complexity. The channel dimensions are in fact chosen to encode at best the azimuth angle over the entire distance range (*i.e.*, high resolution and narrow field of view at far range, low resolution and wider field of view at near range). To prevent losing the signature of small objects (typically few pixels in the RD spectrum), the FPN encoder performs a 2×2 downsampling per block, leading to a total reduction of the tensor size by a factor of 16 in height and width. For similar reasons and to avoid overlaps between adjacent Tx's, it uses 3×3 convolution kernels.

4.3. Range-angle decoder

The range-angle decoder aims to expand the input feature maps to higher resolution representations. This upscaling is usually achieved through multiple deconvolution layers whose output is combined with previous feature maps to preserve spatial details. In our case, the representation is

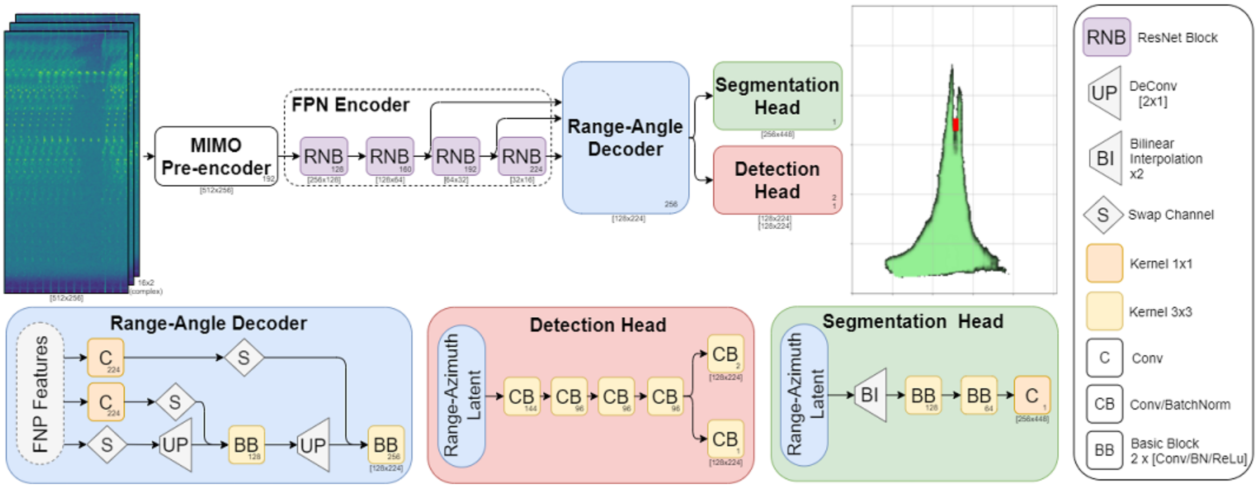

Figure 3. **Overview of FFT-RadNet**. FFT-RadNet is a lightweight multi-task architecture. It does not use any RA maps or RAD tensor which would require costly pre-processing. Instead, it leverages complex range-Doppler spectrums containing all the range, azimuth and elevation information. This data is de-interleaved and compressed by the MIMO pre-encoder. An FPN encoder extracts a pyramid of features which the range-angle decoder converts into a latent range-azimuth representation. Based on this representation, multi-task heads finally detect vehicles and predict the free driving space.

unusual due to the physical nature of the axes: The dimensions of the input tensor correspond respectively to range, Doppler and azimuth angle, whereas the feature maps that will be sent to subsequent task heads should correspond to a range-azimuth representation. Consequently, we swap the Doppler and azimuth axes to match the final axis ordering and then upscale the feature maps. However, the range axis has a lower size compared to the azimuth one, since it was decimated by a factor of 2 after each residual block, while the azimuth axis (formerly the channel axis) was increasing. Prior to these operations, we apply a 1×1 convolution to the feature maps from the encoder to the decoder. It adjusts the dimension of the azimuth channel to its final size, right before swapping the axes. The deconvolution layers upscale only the range axis, producing feature maps that are concatenated with those from the previous pyramid level. A final block of two Conv-BatchNorm-ReLU layers is applied, generating the final range-azimuth latent representation.

4.4. Multi-task learning

Detection task. The detection head is inspired from Pixor [42], an efficient and scalable single-stage model. It takes the RA latent representation as input and processes it using a first common sequence of four Conv-BatchNorm layers with 144, 96, 96 and 96 filters respectively. The branch is then divided in a classification and a regression pathways. The classification part is a convolution layer with sigmoid activation that predicts a probability map. This output corresponds to binary classification of each "pixel" as occupied or not by a vehicle. In order to reduce computational complexity, it predicts a coarse RA map, where each cell has a resolution of 0.8m in range and 0.8° in azimuth (*i.e.*, $1/4$ and $1/8$ of native resolutions resp. in range and azimuth). This cell size is enough to dissociate two close objects. Then, the regression part finely predicts the range and azimuth values corresponding to the detected object. To do so, a unique 3×3 convolution layer outputs two feature maps corresponding to the final range and azimuth values.

This two-fold detection head is trained with a multi-task loss composed of a focal loss applied to all the locations for the classification and of a "smooth L1" loss for the regression applied only on positive detections (see [42] for details of these losses). Let $\mathbf{x}$ be a training example, $\mathbf{y}_\text{clas} \in \{0,1\}^{B_\text{R}/4\times B_\text{A}/8}$ its classification ground truth and $\mathbf{y}_\text{reg} \in \mathbb{R}^{2\times B_\text{R}/4\times B_\text{A}/8}$ the associated regression ground truth. The detection head of FFT-RadNet predicts a detection map $\hat{\mathbf{y}}_\text{clas} \in [0,1]^{B_\text{R}/4\times B_\text{A}/8}$ and associated regression map $\hat{\mathbf{y}}_\text{reg} \in \mathbb{R}^{2\times B_\text{R}/4\times B_\text{A}/8}$. Its training loss reads:

$$\begin{aligned}\mathcal{L}_\text{det}(\mathbf{x},\mathbf{y}_\text{clas},\mathbf{y}_\text{reg}) = \text{focal}(\mathbf{y}_\text{clas},\hat{\mathbf{y}}_\text{clas}) + \\ \beta\,\text{smooth-L1}(\mathbf{y}_\text{reg}-\hat{\mathbf{y}}_\text{reg}),\end{aligned} \tag{1}$$

where $\beta > 0$ is a balancing hyper-parameter.

Segmentation task. The free-driving-space segmentation task is formulated as a pixel-level binary classification. The segmentation mask has a resolution of 0.4m in range and 0.2° in azimuth. It corresponds to half of the native range and azimuth resolutions while considering only half of the entire azimuth FoV (within $[-45^\circ, 45^\circ]$). The RA latent representation is processed by two consecutive groups of two Conv-BatchNorm-ReLu blocks, producing respectively 128 and 64 feature maps. A final 1×1 convolution outputs a

2D feature map followed by a sigmoid activation to estimate the probability of each location to be drivable. Let $\mathbf{x}$ be a training example, $\mathbf{y}_{\text{seg}} \in \{0,1\}^{B_{\text{R}}/2 \times B_{\text{A}}/4}$ its one-hot ground truth and $\hat{\mathbf{y}}_{\text{seg}} \in [0,1]^{B_{\text{R}}/2 \times B_{\text{A}}/4}$ the predicted soft detection map. The segmentation task is learnt using a Binary Cross Entropy loss:

$$\mathcal{L}_{\text{free}}(\mathbf{x}, \mathbf{y}_{\text{seg}}) = \sum_{(r,a)\in\Omega} \text{BCE}(\mathbf{y}_{\text{seg}}(r,a), \hat{\mathbf{y}}_{\text{seg}}(r,a)), \quad (2)$$

where $\Omega = [\![1, \frac{B_{\text{R}}}{2}]\!] \times [\![1, \frac{B_{\text{A}}}{4}]\!]$.

End-to-end multi-task training. The whole FFT-RadNet model is trained by minimizing a combination of the previous detection and segmentation losses:

$$\mathcal{L}_{\text{MTL}} = \sum_{\mathbf{x}} \mathcal{L}_{\text{det}}(\mathbf{x}, \mathbf{y}_{\text{clas}}, \mathbf{y}_{\text{reg}}) + \lambda \mathcal{L}_{\text{free}}(\mathbf{x}, \mathbf{y}_{\text{seg}}), \quad (3)$$

w.r.t. the parameters of the MIMO pre-encoder, of the FPN encoder, of the RA decoder and of the two heads; λ is a positive hyper-parameter that balances the two tasks.

5. RADIal dataset

As depicted in Table 1, publicly-available datasets do not provide raw radar signal, neither for LD radar nor for HD radar. Therefore, we built RADIal, a new dataset to allow research on automotive HD radar. As RADIal includes 3 sensor modalities –camera, radar and laser scanner–, it should also permit one to investigate the fusion of HD radar with other more classic sensors. The specifications of the used sensor suite are detailed in the Supplement. Except for the camera, all sensors are automotive-grade qualified. On top of that, the GPS position and full CAN bus of the vehicle (including odometry) are also provided. Sensor signals were recorded simultaneously in a raw format, without any signal pre-processing. In the case of the HD radar, the raw signal is the ADC. From this ADC data, all conventional radar representations can be generated: range-azimuth-Doppler tensor, range-azimuth and range-Doppler views or point cloud.

RADIal contains 91 sequences of about 1-4 minutes, for a total of 2 hours. This amounts to approximately 25k synchronized frames in total, out of which 8,252 are labelled with 9,550 vehicles (see details in the Supplement). Vehicles' annotation is composed of 2D boxes in the image plane along with the real-world distance to the sensor and the Doppler value (relative radial speed). The annotation of the radar signal is hard to achieve as the RD spectrum representation is not meaningful for the human eye.

Vehicle detection labels were first generated automatically using supervision from the camera and laser scanner. A RetinaNet model [21] was used to extract object proposals from the camera. Then, these proposals were validated when both radar and LiDAR agree on the object position from their respective point cloud. Finally, manual verification was conducted to reject or validate the labels. The free-space annotation was done fully automatically on the camera images. A DeepLabV3+ [5], pre-trained on Cityscape, has been fine-tuned with 2 classes (*free space* and *occupied*) on a small manually-annotated part of our dataset. This model segmented each video frame and the obtained segmentation mask was projected from the camera's coordinate system to the radar's one thanks to known calibration. Finally, already available vehicle bounding boxes were subtracted from the free-space mask. The quality of the segmentation mask is limited due to the automatic method we employed and to the projection inaccuracy from camera to real world.

6. Experiments

6.1. Training details

The proposed architecture has been trained on the RADIal dataset using exclusively the RD spectrums as input. The RD spectrum being composed of complex numbers, we stack its real and imaginary parts along the channel axis before passing it to the MIMO pre-encoder. The dataset has been split into Training, Validation and Test sets (approx. 70%, 15% and 15% of the dataset, respectively) in such a way that frames from a same sequence can not appear in different sets. We manually split the Test dataset in 'hard' and 'easy' cases. Hard cases are mostly situations where the radar signal is perturbed, *e.g.*, by interference with other radars, important side-lobes effects or significant reflections on metallic surfaces.

The FFT-RadNet architecture is trained using the multi-task loss detailed in Section 4.4 with the following hyper-parameters set-up empirically: $\lambda = 100$, $\beta = 100$ and $\gamma = 2$. The training process uses the Adam optimizer [18] during 100 epochs, with an initial learning rate of 10^{-4} and a decay of 0.9 every 10 epochs.

6.2. Baselines

The proposed architecture has been compared to recent contributions in the radar community. Most of the competing methods presented in Section 3 have been designed for LD radar and can not scale with HD radar data due to memory limitation. Instead, baselines with similar complexity have been selected regarding their input representation (range-azimuth or point cloud) for a fair comparison. Input representations (RD, RA or point cloud) are generated for the entire Training, Validation and Test sets using a conventional signal-processing pipeline.

Object detection with point cloud. The Pixor [42] method has been used to detect vehicles after voxelization of the radar point cloud into a 3D volume of

Model	Radar Input	Overall AP(%) ↑	Overall AR(%) ↑	Overall R(m) ↓	Overall A(°) ↓	Easy AP(%) ↑	Easy AR(%) ↑	Easy R(m) ↓	Easy A(°) ↓	Hard AP(%) ↑	Hard AR(%) ↑	Hard R(m) ↓	Hard A(°) ↓
Pixor [42]	PC	96.46	32.32	0.17	0.25	**99.02**	28.83	0.15	0.19	93.28	38.69	0.19	0.33
Pixor [42]	RA	96.56	81.68	**0.10**	0.20	96.86	88.02	**0.09**	0.16	**95.88**	**70.10**	**0.12**	0.27
FFT-RadNet (ours)	RD	**96.84**	**82.18**	0.11	**0.17**	98.49	**91.69**	0.10	**0.13**	92.93	64.82	0.13	**0.26**

Table 2. **Object detection performances on RADIal Test split**. Comparison between Pixor trained with point cloud ('PC') or range-azimuth ('RA') representations, and the proposed FFT-RadNet requiring only range-Doppler ('RD') as input. Our method obtains similar or better overall performances than baselines in both average precision ('AP') and average recall ('AR') for a 50% IoU threshold. It also reaches similar or better range ('R') and angle ('A') accuracy, showing that it successfully learns a signal processing pipeline that estimates the AoA with significantly fewer operations, as detailed in Table 4.

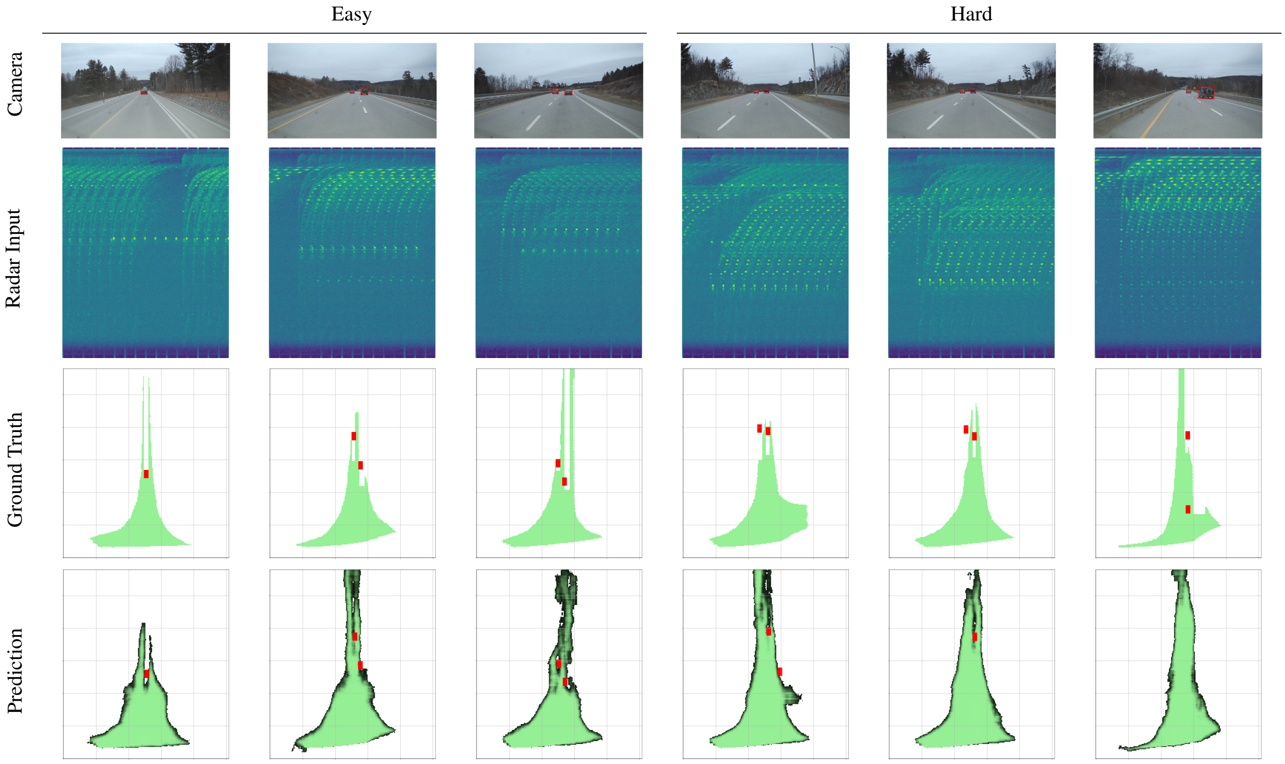

Figure 4. **Qualitative results for object detection and free-space segmentation on Easy and Hard samples.** Camera views (1st row) are displayed for visual reference only; RD spectrums (2nd row) are the only inputs to the model; Ground truths (3rd row) and predictions (4th row) are shown for both tasks. Note that there could be a projection error of the free driving space from camera to real world due to vehicle pitch variations.

$[0\,\mathrm{m}, 103\,\mathrm{m}] \times [-40\,\mathrm{m}, 40\,\mathrm{m}] \times [-2.5\,\mathrm{m}, 2.0\,\mathrm{m}]$ around the radar (longitudinal, lateral and vertical ranges), sampled at 0.1m in each direction. The size for this input 3D grid is thus $1030 \times 800 \times 45$. Pixor is a lightweight architecture intended to be real-time. However, its input representation generates 96MB of data, which becomes a challenge for embedded devices.

Object detection with RA tensor. As detailed in Section 3, several methods [11, 23] used views of the RAD tensor as input. However, the memory usage would be too extensive for HD radar data. As [23] showed that using only the RA view leads to better performance for object detection, we compared our method to a Pixor architecture without the voxelization module. It takes as input the RA representation in RADIal, of size 512×896 with range values in $[0\mathrm{m}, 103\mathrm{m}]$ and azimuth in $[-90°, 90°]$.

Free-space segmentation. We selected PolarNet [29] to evaluate against our approach. It is a lightweight architecture designed to process RA maps and predict free space. We re-implemented it to the best of our comprehension.

6.3. Evaluation metric

For object detection, the Average Precision (AP) and Average Recall (AR) are used considering an Intersection-

Over-Union (IoU) threshold of 50%. For semantic segmentation, the mean IoU (mIoU) metric is used on a binary classification task (*free* or *occupied*). The metric is computed on a reduced $[0\text{m}, 50\text{m}]$ range as the boundaries of the road surface are hardly visible beyond this distance.

6.4. Performance analysis

Object detection. Performances for object detection are reported in Table 2. We observe that FFT-RadNet using range-Doppler as input outperforms the Pixor baseline using PC as input (Pixor-PC) and reaches slightly better performances than the costly Pixor-RA baseline. The position accuracy, both in range and azimuth angle, is similar, and even better in angle, compared to Pixor-RA. These results show that our approach successfully learns the azimuth angle from the data. From a manufacturing viewpoint, note that this opens cost saving opportunities as the end-of-line calibration of the sensor is no longer required in the proposed framework. In the Easy Test set, FFT-RadNet delivers +1.6% AP and +3.6% AR compared to Pixor-RA. However, on the Hard test set, Pixor-RA performs the best. The RA approach does not struggle that much with the hard samples because the data is pre-processed by a signal processing pipeline that already solves some of these cases. In contrast, the performance with point-cloud input is much lower than all others. Indeed, the recall is low due to the limited number of points at far range.

Free-driving-space segmentation. The performance for the free-driving-space segmentation is provided in Table 3. We observe that FFT-RadNet significantly outperforms PolarNet by 13.4% IoU on average. This is partly explained by the lack of elevation information in the RA map, an information that is present in the RD spectrums.

6.5. Complexity analysis

FFT-RadNet has been designed first to get rid of the signal processing chains that transform the ADC data into either a sparse point cloud or denser representations (RA or RAD), without compromising the richness of the signal. Because the input data remains quite large, we designed a compact model to bound the complexity in terms of number of operations, as a trade-off between performance and range/angle accuracy. Moreover, the pre-encoder layer compresses significantly the input data. An ablation study has been performed to define the best trade-off between the size of the feature maps and the model's performance (details in the Supplement).

As shown in Tab. 4, FFT-RadNet is the only method not requiring the AoA estimation. As explained in Sec. 2, the pre-encoder layer compresses the MIMO signal containing all the information to recover both the azimuth and elevation angles. The AoA for the point-cloud approach generates 3D coordinates for a sparse cloud of around 1000 points on average, leading to 8 GFLOPS-worth of computing, prior to applying Pixor for object detection. To produce the RA or RAD tensor, AoA runs for each single bin of the RD map, but only considering one elevation. Such a model is thus unable to estimate the elevation of objects such as bridges or lost cargo (low object). For one elevation, the complexity is about 45 GFLOPS, but would increase up to 495 GLPOPS for all the 11 elevations. We have demonstrated that FFT-RadNet can cut these processing costs without compromising the quality of the estimation.

Model	Radar input	mIoU (%) ↑		
		Overall	Easy	Hard
PolarNet [29]	RA	60.6	61.9	57.4
FFT-RadNet	RD	**74.0**	**74.6**	**72.3**

Table 3. **Free-driving-space segmentation performances**. FFT-RadNet successfully approximates the angle information in the radar data while reaching better performance than PolarNet. Note that this performance is achieved by FFT-RadNet while simultaneously performing object detection, as our model is multi-task.

Method	Input size (MB) ↓	# Params. (10^6) ↓	Complexity (GFLOPS) ↓	
			AoA proc.	Model
PCL Pixor	98.30	6.93	8	741
RA Pixor	**1.75**	6.92	45*	761
FFT-RadNet	16.00	**3.79**	**0**	**584**

Table 4. **Complexity analysis**. The proposed method reaches the best trade-off between the size of the input, the number of parameters of the model and the computational complexity. Note that the AoA processing of the RA Pixor method (*) considers only a single elevation, otherwise it is up to 496 GFLOPS for the whole set of $B_{\text{E}} = 11$ elevations.

7. Conclusion

We introduced FFT-RadNet, a novel trainable architecture to process and analyse HD radar signals. We demonstrated that it effectively alleviates the need for costly preprocessing to estimate RA or RAD representations. Instead, it detects and estimates objects position while segmenting free driving space from the RD spectrum directly. FFT-RadNet slightly outperforms RA-based approaches while reducing processing requirements. The experiments are conducted on RADIal, a new dataset which is part of the work and contains sequences of automotive-grade sensor signals (HD radar, camera and laser scanner). Synchronized sensor data are available in a raw format so that various representations can be evaluated and further research can be conducted, possibly with fusion-based approaches.

References

[1] 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. 2, 3

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

[3] Graham M Brooker. Understanding millimetre wave fmcw radars. In *ICST*, 2005. 2

[4] 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, 2, 3

[5] 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. 6

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1

[7] Andreas Danzer, Thomas Griebel, Martin Bach, and Klaus Dietmayer. 2D car detection in radar data with PointNets. In *ITSC*, 2019. 3

[8] Xu Dong, Pengluo Wang, Pengyue Zhang, and Langechuan Liu. Probabilistic oriented object detection in automotive radar. In *CVPR Worshops*, 2020. 3

[9] B J Donnet and I D Longstaff. Mimo radar, techniques and opportunities. In *EuRAD*, 2006. 2

[10] S. Franceschini, M. Ambrosanio, S. Vitale, F. Baselice, A. Gifuni, G. Grassini, and V. Pascazio. Hand gesture recognition via radar sensors and convolutional neural networks. In *RadarConf*, 2020. 3

[11] 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, 7

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 1

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

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 4

[15] Xinrui Jiang, Ye Zhang, Qi Yang, Bin Deng, and Hongqiang Wang. Millimeter-wave array radar-based human gait recognition using multi-channel three-dimensional convolutional neural network. *Sensors*, 2020. 3

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

[17] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. Mulran: Multimodal range dataset for urban place recognition. In *ICRA*, 2020. 2, 3

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[19] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 1

[20] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[21] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 6

[22] Jakob Lombacher, Kilian Laudt, Markus Hahn, Jurgen Dickmann, and Christian Wohler. Semantic radar grids. In *IV*, 2017. 3

[23] 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, 7

[24] Michael Meyer and Georg Kuschk. Automotive radar dataset for deep learning based 3d object detection. In *EuRAD*, 2019. 2, 3

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

[26] Casian Miron, Alexandru Pasarica, and Radu Timofte. Efficient cnn architecture for multi-modal aerial view object classification. In *CVPR Workshops*, 2021. 3

[27] Mohammadreza Mostajabi, Ching Ming Wang, Darsh Ranjan, and Gilbert Hsyu. High resolution radar dataset for semi-supervised learning of dynamic objects. In *CVPR Workshop*, 2020. 2, 3

[28] Weichong Ng, Guohua Wang, Siddhartha, Zhiping Lin, and Bhaskar Jyoti Dutta. Range-Doppler detection in automotive radar with deep learning. In *IJCNN*, 2020. 3

[29] Farzan Erlik Nowruzi, Dhanvin Kolhatkar, Prince Kapoor, Elnaz Jahani Heravi, Fahed Al Hassanat, Robert Laganière, Julien Rebut, and Waqas Malik. Polarnet: Accelerated deep open space segmentation using automotive radar in polar domain. In *VEHITS*, 2021. 3, 7, 8

[30] Arthur Ouaknine, Alasdair Newson, Patrick Pérez, Florence Tupin, and Julien Rebut. Multi-view radar semantic segmentation. In *ICCV*, 2021. 3

[31] Arthur Ouaknine, Alasdair Newson, Julien Rebut, Florence Tupin, and Patrick Pérez. Carrada dataset: Camera and automotive radar with range- angle- doppler annotations. In *ICPR*, 2021. 2, 3

[32] 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

[33] 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. 3

[34] Robert Prophet, Gang Li, Christian Sturm, and Martin Vossiek. Semantic segmentation on automotive radar maps. In *IV*, 2019. 3

[35] 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. 3

[36] Ole Schumann, Markus Hahn, Nicolas Scheiner, Fabio Weishaupt, Julius F. Tilly, Jürgen Dickmann, and Christian Wöhler. Radarscenes: A real-world radar point cloud data set for automotive applications. In *ArXiv*, 2021. 2, 3

[37] Ole Schumann, Jakob Lombacher, Markus Hahn, Christian Wohler, and Jurgen Dickmann. Scene understanding with automotive radar. In *IV*, 2020. 3

[38] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. RADIATE: a radar dataset for automotive perception. In *ArXiv*, 2020. 2, 3

[39] 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. 3

[40] 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. 3

[41] Yizhou Wang, Gaoang Wang, Hung-Min Hsu, Hui Liu, and Jenq-Neng Hwang. Rethinking of radar's role: A camera-radar dataset and systematic annotator via coordinate alignment. In *CVPR Workshop*, 2021. 2, 3

[42] Bin Yang, Wenjie Luo, and Raquel Urtasun. PIXOR: real-time 3d object detection from point clouds. In *CVPR*, 2018. 1, 5, 6, 7

[43] Ao Zhang, Farzan Erlik Nowruzi, and Robert Laganière. Raddet: Range-azimuth-doppler based radar object detection for dynamic road users. In *CVR*, 2021. 2, 3

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

[45] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 4