DAROD: A Deep Automotive Radar Object Detector on Range-Doppler maps

Colin Decourt, Rufin Vanrullen, Didier Salle, Thomas Oberlin

DAROD: A Deep Automotive Radar Object Detector on Range-Doppler maps

Colin Decourt[1,2,3,4], Rufin VanRullen[1,2], Didier Salle[1,4], Thomas Oberlin [1,3]

Abstract— **Due to the small number of raw data automotive radar datasets and the low resolution of such radar sensors, automotive radar object detection has been little explored with deep learning models in comparison to camera and lidar-based approaches. However, radars are low-cost sensors able to accurately sense surrounding object characteristics (e.g., distance, radial velocity, direction of arrival, radar cross-section) regardless of weather conditions (e.g., rain, snow, fog). Recent open-source datasets such as CARRADA, RADDet or CRUW have opened up research on several topics ranging from object classification to object detection and segmentation. In this paper, we present DAROD, an adaptation of Faster R-CNN object detector for automotive radar on the range-Doppler spectra. We propose a light architecture for features extraction, which shows an increased performance compare to heavier vision-based backbone architectures. Our models reach respectively an mAP@0.5 of 55.83 and 46.57 on CARRADA and RADDet datasets, outperforming competing methods.**

I. Introduction

In the last decade, the increasing number of advanced driver-assistance systems (ADAS) has led to an increase in the number of sensors embedded in the car including camera, lidar and radar. These sensors together enable the vehicle to depict the surrounding environment and adapt its behaviour depending on it. Nowadays, most intelligent vehicles use camera and lidar for ADAS applications as they provide high-resolution output and high-performance in 3D object detection and classification tasks. Because of poor angular resolution, radar sensors have been neglected for object classification and detection tasks, and used mostly for blind spot detection or automatic cruise control. Yet, radar sensors seem particularly suited for critical and real-time automotive applications such as automatic emergency braking, because they are not hampered by light or weather conditions and they provide information such as range and velocity of the surrounding objects. Paired with camera and lidar sensors, radar could bring redundancy at sensors level to improve safety in the vehicles. In this paper we propose a new deep learning model for object detection and classification using radar data.

[1]Artificial and Natural Intelligence Toulouse Institute, Université de Toulouse, France

[2]CerCO, CNRS UMR5549, Toulouse

[3]ISAE-SUPAERO, Université de Toulouse, 10 Avenue Edouard Belin, Toulouse 31400, France

[4] NXP Semiconductors, Toulouse, France

This work has been funded by the Institute for Artificial and Natural Intelligence Toulouse (ANITI) under grant agreement ANR-19-PI3A-0004.

Computations performed to train our model have been carried out using the OSIRIM platform, administered by IRIT and supported by CNRS, Région Occitanie, the French Government and ERDF.

Object detection and classification is one of the main tasks in computer vision, for which deep neural networks have achieved a major breakthrough in the past decade. Such approaches have been successfully applied to lidar and camera [1], [2] but the dearth of publicly available annotated radar datasets has slowed down research in object detection and segmentation from radar data. As illustrated in Figure 1, radar data can be represented either as a target lists (point clouds) or as raw data tensors (Range-Doppler or Range-Angle-Doppler maps). Target lists, which are the default radar data format, contain very low level information such as the position of the targets all around the vehicle, their velocities and their radar cross-sections. As an example, target lists can be used as input to deep neural network for object classification [3] or object segmentation [4]. However, the filtering techniques applied to the radar signal to obtain target lists lead to a loss of useful information contained in raw data tensors. Instead, raw data tensors and deep neural networks can be used to replace and improve traditional techniques for object detection, classification and segmentation without losing information. Recently, radar datasets and challenges such as CARRADA [5], RADDet [6] or CRUW [7], where radar data is provided as raw data tensors, have opened up research on new deep learning methods for automotive radar ranging from object detection [6], [8], [9] to object segmentation [10].

In this work, we propose a new model for object detection and classification using Faster R-CNN [11] algorithm based only on Range-Doppler (RD) maps. The use of RD maps instead of Range-Angle-Doppler (RAD) tensors is motivated by the fact that RAD tensors are more computationally demanding to produce for radar Micro Controller Units (MCUs). We propose a lightweight backbone for Faster R-CNN object detection, adapted to range-Doppler data. We design our model to handle the complexity of the RD maps and the small size of radar objects while trying to keep the processing pipeline as efficient as possible. Experiments on CARRADA and RADDet datasets show that our model can help improving object detection and classification performance on radar data and outperforms competing methods.

The paper is organised as follows. Section II presents the related work and some background on radar processing. Section III then introduces our model. The experiments and results are gathered in Section IV, while Section V discusses and concludes the paper.

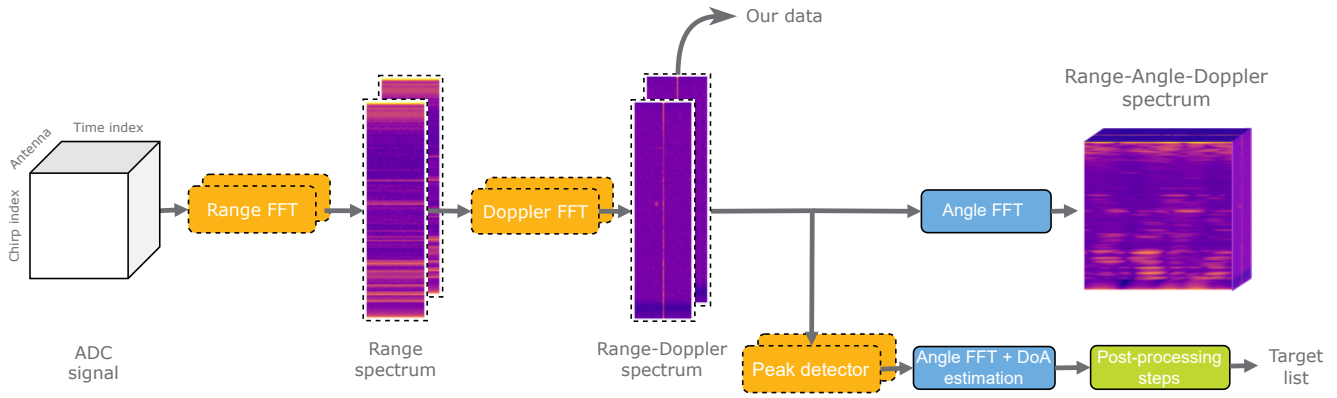

Fig. 1. Illustration of the radar signal processing pipeline. Orange boxes denote range and Doppler processing operations, blue boxes correspond to angle processing operations and the green box represents the post-processing.

II. Background and Related Work

A. Object detection for computer vision

In computer vision, the object detection task has been widely explored over the past few years thanks to many challenges such as Pascal VOC [12], COCO [13] or more specifically to ADAS application, KITTI [14]. Object detectors can be split into two categories: single-stage detectors and two-stages detectors. Single-stage detectors such as YOLO [15], SSD [16] or Retina-Net [17] are architectures which directly predict a bounding box and a class given an input image. However, this simplicity results in lower performance than two-stage detectors such as Faster R-CNN [11], R-FCN [18] or FCOS [19]. Two-stage detectors are generally composed of a region proposal network (RPN) that first proposes areas in the input image that potentially contain an object and then predicts a bounding box and a class for each of these regions. This two-step approach allows two-stage detectors to be more accurate than single-stage detectors but often slower.

B. Radar pipeline

Radar is an active sensor that transmits electromagnetic wave signals, which get reflected by objects in their field of view [20]. By capturing the reflected signal, a radar system can determine the range, velocity and angle of the objects. For automotive applications, most radars transmit a frequency-modulated continuous-wave signal (FMCW) in order to measure range as well as angle and velocity. An FMCW radar emits chirp signal, a signal whose instantaneous frequency increases or decreases linearly with time. Usually, an FMCW radar receives N samples of M chirps signal over T_x antennas resulting in a $N \times M \times T_x$ output tensor containing the received signal in the time domain. We called this tensor the Analog to Digital Converted (ADC) signal.

As illustrated in Figure 1, the distance information is extracted by performing a fast Fourier transform (FFT) over the ADC samples within one chirp. The velocity information is extracted by performing a second FFT over the chirp index to estimate the phase difference between chirps and deduce the Doppler shift, resulting in a range-Doppler spectrum. Finally, a 3^{rd} FFT (or angle FFT) or more advanced algorithms is applied in the antenna dimension to extract the angle information and finally generate Range-Angle-Doppler tensor (or RAD cube). Because the RAD tensor is too intensive to compute, targets are usually detected on the RD spectrum using peak detection algorithms such as CFAR [21]. Then, radar reflections are obtained using angle FFT or beamforming methods and some post-processing steps (ego-motion compensation, Kalman filtering).

C. Deep learning for automotive radar

Many research works have proposed to leverage the power of deep learning to improve some parts of the radar processing chain, ranging from target detection and classification to direction of arrival (DoA) estimation.

1) Reflection-based methods: In the automotive industry, the most common representation of radar data is a list of targets around the vehicle. Researchers widely use this representation for target recognition [22], [3], segmentation [4], [23], [24], ghost target detection [25] or 3D radar-camera object detection [26]. The sparsity of radar reflections allows development of lightweight, efficient and high-performance neural networks running on edge-computing. Nonetheless, this sparsity, due to the filtering techniques applied to the signal and the post-processing steps presented in Figure 1, results in a loss of valuable information contained in the raw radar signal.

2) Raw data-based methods: To overcome this, several works consider lower level representations, mainly the RD, RA or RAD tensors. Because RAD tensors aggregate both distance, velocity and angle information together, there is an increasingly number of works using this representation. Major et al. [27] and Gao et al. [28] propose similar architectures, merging each view into a single 2D tensor to detect and identify targets in RA maps. Paffly et al. [29] exploit the RAD view to detect and classify road-user objects using the radar data, while enriching the radar detection with the RAD cube. In [10], Ouaknine et al. use the RAD

cube representation to segment objects both in the RA and the RD maps using lightweight segmentation architecture which exploit the temporal information. 3D object detection is also explored by Zhang et al. in [6] using a YOLO-like architecture.

Since RA maps provide angle information, thus allowing to detect targets around the car, it has been explored extensively for different tasks. Patel et al. [30] extract regions of interest (ROI) from RA maps to classify targets. In a similar manner, Akita et al. [31] simultaneously track and classify targets using extracted ROI from RA maps. Recently, as part of the CRUW challenge [7] new object detection and classification architectures were proposed on RA maps such as [32] and [9].

The range-Doppler spectrum has been considered more recently in the following works. Perez et al. [33] use convolutional neural networks (CNN) for vulnerable road user classification. Similarly, Khalid et al. [34] use CNN and long-short term memory networks for target identification. [35] and [36] use YOLO-like architectures for object detection and classification. However, squaring and up-sampling steps in [35] might result in loss of information in the Range-Doppler spectrum. Finally Ng et al. [37] propose a U-Net-like architecture to replace CFAR algorithm, thus detecting targets.

As mentioned in section II-B, although RAD tensors provide the most informative data, they are cumbersome to compute for radar processors. In the purpose of object detection, RA views are not adequate representations either, since they do not account for Doppler which is a crucial information. Besides, RA maps usually suffer from a poor angular resolution caused by a small number of antennas in the FMCW radar. Instead, we hypothesise that the RD spectrum contains enough information for both detection and classification tasks in automotive radar. Angular information might be computed for each target afterwards in a post-processing step, either using standard techniques or with AI as done by Brodeski et al. in [38].

III. Methodology

In this section we present a lightweight Faster R-CNN architecture for object detection in Range-Doppler spectra. Given a RD map as input, we use a convolutional neural network to learn relevant features, as in Faster R-CNN. Following the features extraction, a region proposal network (RPN) is used to propose regions in the spectrum that contain potential targets. To generate region proposals, a small network is slid over the learned convolutional feature map. For each point in the feature map, the RPN learns whether an object is present in the input image at its corresponding location and estimates its size. This is done by placing a set of *anchors* on the input image for each location on the output feature map. These anchors indicate possible objects in various sizes and aspect ratios at this location. We invite the reader to refer to the Faster-RCNN paper [11] for more information about RPN and anchors. Next, the bounding box proposals from the RPN are used to pool features from the

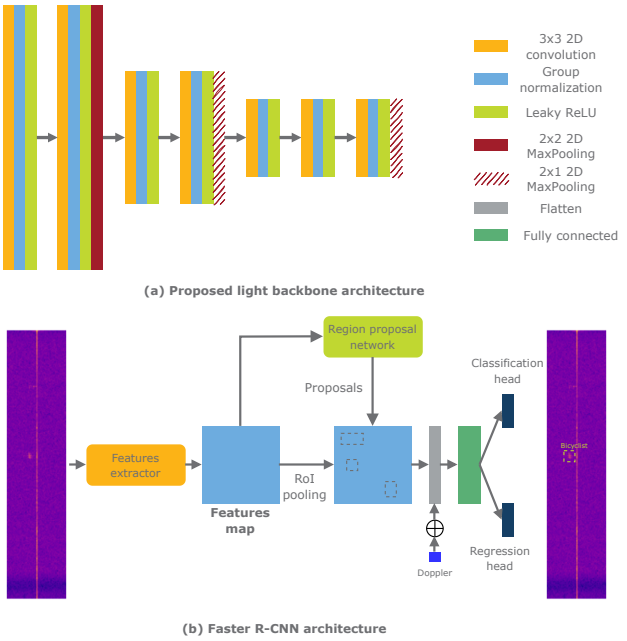

Fig. 2. Model architecture (lightweight Faster R-CNN architecture)

backbone feature map. These features are used to classify the proposals either as background or object and to predict a bounding box using two sibling fully connected layers. This second part is named Fast-RCNN. We depict this pipeline in Figure 2b.

We show in Figure 3 two RD maps with radar signatures of some objects in the captured scene of RADDet dataset. Even though those RD maps seem complex, the information they contain remains of low complexity, contrary to camera images which are bigger and more diverse in terms of textures, orientations, geometry, lighting, etc. Although being more noisy, RD maps have fixed orientation and their objects exhibit more similar patterns and shapes.

To account for those differences, we modify Faster R-CNN to include a lighter backbone and a modified RPN. Our backbone is derived from the VGG architecture [39] and contains 7 convolutional layers. Though residual networks are state-of-the-art architectures for features extraction, residual connections increase the complexity of the model and are difficult to implement in hardware. Figure 2a depicts this lightweight backbone architecture. To keep the processing pipeline as simple and efficient as possible, we decide to not resize the spectrum and to process it as it is, resulting in an input of size 256×64. The backbone is composed of two blocks with two 2D convolutions and one block with three 2D convolutions. Following each convolutional block, we apply a 2D max pooling operation to down-sample the size of the input. We down-sample the Doppler dimension only by a factor of two after the first block, to minimise loss of the Doppler information, which is useful for classification. Then, we obtain a feature map size 32×32 which was found to lead to the best performance. The number of channels for each block of convolutions are respectively set to 64, 128

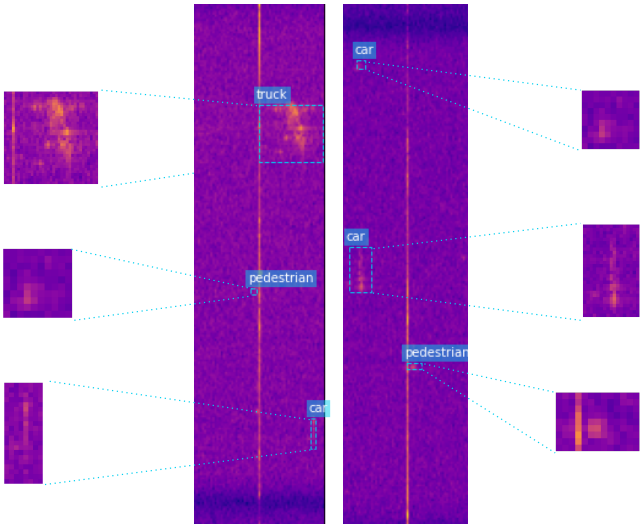

Fig. 3. Two RD maps of RADDet dataset, along with the bounding boxes around objects and their zoom.

and 256.

The next step is to defined the anchors used by the RPN so as to capture the diversity of shape and size in the objects. In this work we use 3 scales and 3 aspect ratios for anchors generation, yielding to 9 anchors at each position in the feature map. The mean size of objects in RD maps is 8×8, we use this size as reference for anchors scale. For smaller and bigger objects we respectively use scale of 4 and 16 resulting in anchors scales of size 4, 8 and 16. Additionally, we set aspect ratios to $\frac{1}{4}$, $\frac{1}{2}$ and $\frac{1}{8}$. Contrary to Faster R-CNN paper where aspect ratios are set to 1, 2 and $\frac{1}{2}$, we choose aspect ratios where denominators are multiple of 4 to account for the $\frac{1}{4}$ ratio between input height and width. To reduce the computational complexity of the model, we don't consider all the combinations of scales and ratios. We only consider combinations containing scale 8 and ratio $\frac{1}{4}$ resulting in less anchors generated per image (5 at each position). We find these settings provide the best performances.

Since the RD spectrum is not translation invariant (the velocity is a characteristic of the target), we decide to add this information to the feature vector used for classification and bounding box regression. This feature vector corresponds to the flattened region proposed by the RPN. We compute the velocity by extracting the position of the pixel with the highest intensity in proposed region of interest (ROI). Knowing the velocity resolution of the radar δ_v and the position of the highest pixel in the ROI p, we compute the velocity using the following formula: $v = \delta_v * p$. We notice a slight improvement in the performances using the Doppler velocity as extra feature.

We optimise the model using the loss function described in [11]. To take into account the uncertainty of the annotations, we slightly modify the intersection over union (IoU) thresholds to assign an anchor and a detection to a ground-truth box. We set this threshold to 0.5 for the RPN (instead of 0.7 for Faster R-CNN) and to 0.3 for Fast R-CNN (instead of 0.5 for Faster R-CNN). As there is not a lot of objects in our RD data, we randomly sample 32 proposals (instead of 512 for Faster R-CNN) to compute the loss of the RPN and Fast R-CNN.

IV. Experiments and Results

A. Datasets and competing method

We train our model on the two publicly available radar datasets CARRADA [5] and RADDet [6]. For the CARRADA dataset we use the segmentation masks as reference to create our bounding boxes by drawing a box around masks. Regarding the RADDet dataset, we extract the RD maps by summing the values of the RAD tensors over the angles dimension. We use the same bounding boxes provided by the authors of the RADDet dataset by only taking coordinates along the range and the Doppler dimension. We use default train/val/test distribution of CARRADA dataset. For RADDet, we randomly split the train into training and validation set with a 9:1 ratio. For testing, we use the provided test set.

We compare our model DAROD, made of the lightweight backbone and the simplified Faster R-CNN architecture displayed in Figure 2, with the RADDet model [6]. To the best of our knowledge it is the only published object detector designed for radar data. We modify the RADDet model to train it only with RD maps as input instead of RAD tensors. We also consider the variant termed RADDet RAD which corresponds to the original RADDet model (train on RAD tensors) evaluated only on the range and the Doppler dimensions, using the pretrained weights provided in [6]. As a second baseline we consider the state of the art in computer vision, by selecting the Torchvision[1] Faster R-CNN implementation using the default hyper-parameters, namely a resizing of the input from 256×64 to 800×800 and a ResNet50+FPN backbone pretrained on ImageNet. FPN (Feature Pyramid Network) is a feature extractor that takes a single-scan image of an arbitrary size, and outputs proportionally sized feature maps at multiple levels thus allowing to detect object at different scales. In addition, we train the Torchvision Faster R-CNN without the pre-traning on ImageNet to evaluate the impact of this pre-training on the results.

B. Training setting and evaluation metrics

We use the Adam optimiser with the recommended parameters. A learning rate of 1×10^{-4} is used for all our experiments. The batch size is set to 4 for CARRADA dataset and to 16 for RADDet dataset. Our model is trained over respectively 50 and 80 epochs for CARRADA and RADDet datasets. As Faster R-CNN object detector contains several hyper-parameters, we perform grid-search over some carefully chosen parameters to improve the performance of our model. We randomly use horizontal and vertical flipping as data augmentation strategies.
We evaluate our model using the mean average precision (mAP), a well-known metric for evaluating object detectors.

[1] https://github.com/pytorch/vision

TABLE I

RESULTS OF DIFFERENT MODELS ON CARRADA AND RADDET DATASETS. BEST RESULTS ARE SHOWN IN BOLD, SECOND BEST ARE UNDERLINED.

Dataset	Model	IoU 0.3			IoU 0.5			# params (M)	Inference time
		mAP	Precision	Recall	mAP	Precision	Recall		
CARRADA	DAROD (ours)	70.68	**76.73**	52.52	55.83	**68.34**	46.03	**3.4**	**25.31 ms**
	Faster R-CNN (pretained)	**71.08**	51.70	72.97	**61.56**	47.86	67.21	41.3	37.19 ms
	Faster R-CNN (from scratch)	64.21	45.90	**74.17**	52.93	41.59	**67.40**	41.3	37.19 ms
	RADDet RD	48.59	61.31	42.56	18.57	36.73	25.50	7.8	74.03 ms
RADDet	DAROD (ours)	**65.56**	**82.31**	47.78	46.57	**68.23**	38.74	**3.4**	**25.31 ms**
	Faster R-CNN (pretrained)	58.47	52.17	56.92	**49.55**	47.78	51.77	41.3	37.19 ms
	Faster R-CNN (from scratch)	49.16	32.33	**61.46**	40.84	29.37	**55.29**	41.3	37.19 ms
	RADDet RD	38.42	78.20	29.77	22.87	60.41	20.55	7.8	74.03 ms
	RADDet RAD [6]	38.32	68.80	26.83	17.13	46.55	16.99	8	75.2 ms

We consider mAP at IoU thresholds 0.3 and 0.5 to take into account the uncertainty on the annotations, which is generated semi-automatically for both datasets. In addition, we provide precision and recall at IoU thresholds 0.3 and 0.5. All the experiments are conducted using the Tensorflow[2] deep learning framework along with an Nvidia RTX 2080Ti GPU.

C. Results

Table I shows the performance of our model on CARRADA and RADDet datasets[3]. In a nutshell, our DAROD model clearly outperforms the RADDet method on both datasets, while it remains competitive with Faster R-CNN. When pre-trained on ImageNet, Faster R-CNN leads to the best mAP in 3 cases with DAROD being second best, the positions being inverted in the last experiment (RADDet dataset and IoU at 0.3).

Generally, we observe that DAROD achieves good precision scores but medium recall. This suggests that our model is accurate when detecting targets (eg. correctly classifies them) but seems to struggle to detect all the targets, resulting in missed targets. We draw the same conclusion for the RADDet model which obtains decent precision scores but low recall, hence impacting mAP@0.3 and mAP@0.5. The Faster R-CNN model achieves sufficient precision scores and good recall, resulting in more false positives but less missed targets, which may be better for critical applications. For DAROD, we aimed to optimise mAP, which measures the global performance of object detector. We might be able to improve the recall by reducing the selectivity of our model during training and in the post-processing step, or by decreasing the penalty of classification errors.

Finally, we remark the pretraining of Faster R-CNN backbone on the ImageNet dataset helps to improve the detection performance. Particularly, it drastically improves the precision score but doesn't seem to impact the recall score.

A critical point in automotive radar is the computational load of the different models. We compute the FLOPS (floating point operations per second) of the different models and represent it as a function of the performance in Figure 4. Not surprisingly, radar based approaches are far more efficient than Faster R-CNN that uses up-sampling and deeper backbones. RADDet model is the model with the lowest number of FLOPS as it is inspired from the single stage detector YOLO [15]. The number of FLOPS required by DAROD is slightly bigger than RADDet, but stays reasonable to run on MCUs.

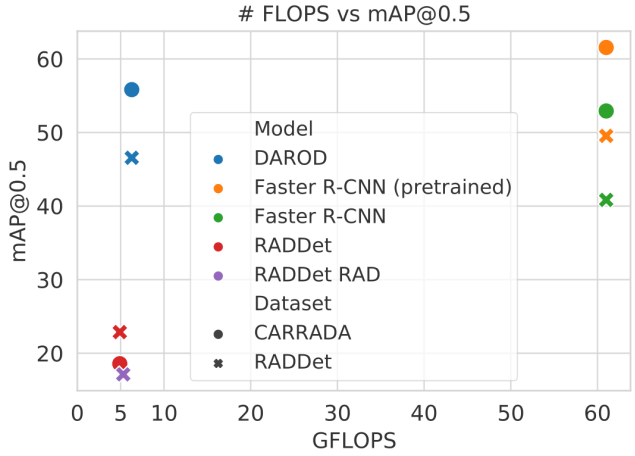

Fig. 4. Number of FLOPS vs. mAP@0.5 for DAROD, Faster R-CNN (pretrained or trained from scratch), RADDet and RADDet RAD.

V. DISCUSSION

We show in section IV-C that our model achieves much better mAP at threshold 0.3 and 0.5 than RADDet model on CARRADA and RADDet datasets. We would like to emphasise the RADDet model was specifically designed to process RAD tensors instead of RD spectra. Therefore, it might be inefficient on RD spectra and the results might suffer from this difference. For this reason, we evaluate the RADDet model on the range and the Doppler dimensions only, using the pre-train model provided by the authors. Results are given in Table I as the RADDet RAD model.

At similar feature map resolution as DAROD, the Faster R-CNN model we compare with achieves good performance. However regarding the large number of parameters and FLOPS of this model, the gain in performance we observe doesn't improve by far the results. We conclude we don't need to use very deep convolutional neural networks to

[2]https://www.tensorflow.org/

[3]We train all the models 10 times and we show the best results for each in Table I

extract meaningful information from radar data.

Although camera images are very different from RD maps, pre-training the weights of Faster R-CNN leads to an improvement of 7 to 9 points in mAP, which outperforms DAROD in 3 of the 4 cases. Pre-training the backbone of DAROD might also lead to a significant increase of performance. But this is not trivial, since it requires a well-suited dataset in terms of shape and complexity, this is thus left for future works.

We demonstrate that a simple and light backbone performs well for object detection and classification tasks contrary to deeper image based backbones. We also show our model outperforms similar radar based approach. However, our model doesn't take into account the temporal information of radar data which could be useful to build more accurate radar object detector. Research on this needs to be conducted.

Even though our model is the lightest in terms of number of parameters, the number of floating point operations it requires remain big to be embedded. Replacing the 2D convolutions by depth-wise separable convolutions or transforming our model into a single stage detector could be a solution to improve the efficiency of our backbone.

Finally, the approach we propose remains inspired from computer vision algorithms and might not be the most suitable and efficient approach for automotive applications. Working directly with the complex Range-Doppler spectrum to predict the position of objects in 3D (range, angle, velocity) instead of bounding boxes on RD spectra could address this problem.

We hope this work helps for the development of new deep learning approaches for object detection and/or classification using the RD spectra, in order to develop more efficient and high-performance object detectors for radar.

REFERENCES

[1] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[2] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4490–4499.

[3] N. Scheiner, O. Schumann, F. Kraus, N. Appenrodt, J. Dickmann, and B. Sick, "Off-the-shelf sensor vs. experimental radar - How much resolution is necessary in automotive radar classification?" in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 2020, pp. 1–8.

[4] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Semantic Segmentation on Radar Point Clouds," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 2179–2186.

[5] A. Ouaknine, A. Newson, J. Rebut, F. Tupin, and P. Pérez, "CARRADA Dataset: Camera and Automotive Radar with Range- Angle-Doppler Annotations," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 5068–5075.

[6] A. Zhang, F. E. Nowruzi, and R. Laganiere, "RADDet: Range-Azimuth-Doppler based Radar Object Detection for Dynamic Road Users," in *2021 18th Conference on Robots and Vision (CRV)*, 2021, pp. 95–102.

[7] Y. Wang, G. Wang, H.-M. Hsu, H. Liu, and J.-N. Hwang, "Rethinking of Radar's Role: A Camera-Radar Dataset and Systematic Annotator via Coordinate Alignment," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 2809–2818.

[8] M. Meyer, G. Kuschk, and S. Tomforde, "Graph Convolutional Networks for 3D Object Detection on Radar Data," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 3053–3062.

[9] Z. Zheng, X. Yue, K. Keutzer, and A. Sangiovanni Vincentelli, "Scene-aware learning network for radar object detection," in *2021 International Conference on Multimedia Retrieval*, New York, NY, USA, 2021, p. 573–579.

[10] A. Ouaknine, A. Newson, P. Pérez, F. Tupin, and J. Rebut, "Multi-view radar semantic segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 671–15 680.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012)."

[13] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *ECCV*, 2014.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *ECCV*, 2016.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[18] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-Based Fully Convolutional Networks," in *30th International Conference on Neural Information Processing Systems*, 2016, p. 379–387.

[19] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully Convolutional One-Stage Object Detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9626–9635.

[20] M. A. Richards, *Fundamentals of Radar Signal Processing*, 2nd ed. New York: McGraw-Hill Education, 2014.

[21] S. Blake, "Os-cfar theory for multiple targets and nonuniform clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 6, pp. 785–790, 1988.

[22] N. Scheiner, N. Appenrodt, J. Dickmann, and B. Sick, "Radar-based Road User Classification and Novelty Detection with Recurrent Neural Network Ensembles," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 722–729.

[23] Z. Feng, S. Zhang, M. Kunert, and W. Wiesbeck, "Point Cloud Segmentation with a High-Resolution Automotive Radar," in *AmE 2019 - Automotive meets Electronics; 10th GMM-Symposium*, 2019, pp. 1–5.

[24] A. Danzer, T. Griebel, M. Bach, and K. Dietmayer, "2D Car Detection in Radar Data with PointNets," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 61–66.

[25] F. Kraus, N. Scheiner, W. Ritter, and K. Dietmayer, "Using Machine Learning to Detect Ghost Images in Automotive Radar," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7.

[26] M. Meyer and G. Kuschk, "Deep Learning Based 3D Object Detection for Automotive Radar and Camera," in *2019 16th European Radar Conference (EuRAD)*, 2019, pp. 133–136.

[27] B. Major, D. Fontijne, A. Ansari, R. T. Sukhavasi, R. Gowaikar, M. Hamilton, S. Lee, S. Grzechnik, and S. Subramanian, "Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 924–932.

[28] X. Gao, G. Xing, S. Roy, and H. Liu, "RAMP-CNN: A Novel Neural Network for Enhanced Automotive Radar Object Recognition," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 5119–5132, 2021.

[29] A. Palffy, J. Dong, J. F. P. Kooij, and D. M. Gavrila, "CNN Based Road User Detection Using the 3D Radar Cube," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1263–1270, 2020.

[30] K. Patel, K. Rambach, T. Visentin, D. Rusev, M. Pfeiffer, and B. Yang, "Deep Learning-based Object Classification on Automotive Radar Spectra," in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–6.

[31] T. Akita and S. Mita, "Object Tracking and Classification Using Millimeter-Wave Radar Based on LSTM," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1110–1115.

[32] Y. Wang, Z. Jiang, Y. Li, J.-N. Hwang, G. Xing, and H. Liu, "ROD-Net: A Real-Time Radar Object Detection Network Cross-Supervised by Camera-Radar Fused Object 3D Localization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 954–967, 2021.

[33] R. Pérez, F. Schubert, R. Rasshofer, and E. Biebl, "Single-Frame Vulnerable Road Users Classification with a 77 GHz FMCW Radar Sensor and a Convolutional Neural Network," in *2018 19th International Radar Symposium (IRS)*, 2018, pp. 1–10.

[34] H.-u.-R. Khalid, S. Pollin, M. Rykunov, A. Bourdoux, and H. Sahli, "Convolutional Long Short-Term Memory Networks for Doppler-Radar Based Target Classification," in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–6.

[35] R. Pérez, F. Schubert, R. Rasshofer, and E. Biebl, "Deep Learning Radar Object Detection and Classification for Urban Automotive Scenarios," in *2019 Kleinheubach Conference*, 2019, pp. 1–4.

[36] K. Fatseas and M. J. Bekooij, "Neural Network Based Multiple Object Tracking for Automotive FMCW Radar," in *2019 International Radar Conference (RADAR)*, 2019, pp. 1–5.

[37] W. Ng, G. Wang, Siddhartha, Z. Lin, and B. J. Dutta, "Range-Doppler Detection in Automotive Radar with Deep Learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[38] D. Brodeski, I. Bilik, and R. Giryes, "Deep Radar Detector," in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–6.

[39] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.