A Recurrent CNN for Online Object Detection on Raw Radar Frames

Colin Decourt, Rufin VanRullen, Didier Salle, Thomas Oberlin

A recurrent CNN for online object detection on raw radar frames

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

***Abstract*—Automotive radar sensors provide valuable information for advanced driving assistance systems (ADAS). Radars can reliably estimate the distance to an object and the relative velocity, regardless of weather and light conditions. However, radar sensors suffer from low resolution and huge intra-class variations in the shape of objects. Exploiting the time information (e.g, multiple frames) has been shown to help to capture better the dynamics of objects and, therefore, the variation in the shape of objects. Most temporal radar object detectors use 3D convolutions to learn spatial and temporal information. However, these methods are often non-causal and unsuitable for real-time applications. This work presents RECORD, a new recurrent CNN architecture for online radar object detection. We propose an end-to-end trainable architecture mixing convolutions and ConvLSTMs to learn spatio-temporal dependencies between successive frames. Our model is causal and requires only the past information encoded in the memory of the ConvLSTMs to detect objects. Our experiments show such a method's relevance for detecting objects in different radar representations (range-Doppler, range-angle) and outperform state-of-the-art models on the ROD2021 and CARRADA datasets while being less computationally expensive.**

***Index Terms*—Computer Vision and Pattern Recognition, Radar Object Detection, Autonomous Driving, Radar imaging**

I. Introduction

TODAY, most advanced driving assistance systems (ADAS) use cameras and LiDAR sensors to perceive and represent the surrounding environment. Cameras provide rich visual semantic information about the environment, while LiDARs provide high-resolution point clouds of surrounding targets. Large-scale image-based and LiDAR-based datasets [1], [2] have enabled the development of perception algorithms for object detection and segmentation. Despite their high resolution, camera and LiDAR sensors suffer from high sensitivity to harsh weather (fog, snow, rain) and bad light conditions (night or sunny). On the contrary, radar operates at a millimetre wavelength, which can penetrate or diffract around tiny particles, making it a robust and crucial sensor for ADAS applications. Radar sensors also provide accurate localisation of surrounding targets and can estimate the velocity of vehicles and pedestrians in a single capture. However, the small number of public radar datasets and the lack of uniformity between them have slowed down research in deep learning for radar.

As shown in Figure 1, radar data can be represented either as target lists or as raw data spectra (or tensors). Target lists are the default representation of radar data. They are obtained after several processing steps, including signal processing (Fourier transforms), threshold algorithms (CFAR [3]), target tracking (Kalman filtering [4]), and classification algorithms. Targets lists representation contains low-level information such as position (x, y), azimuth θ, relative speed v_r and radar cross section σ of targets. Classification or segmentation algorithms can be applied to these data, such as in [5]–[8], to provide semantic information about the target. However, the target lists have suffered several pre-processing steps. They do not contain all the initial information, which might lower the performance of classification or segmentation (ghost targets, sparse point clouds). Instead of radar target lists, it is possible to consider raw radar data tensors (range-doppler (RD), range-azimuth (RA) or range-azimuth-doppler (RAD) spectra) to exploit all the information available in the radar signal. In the last two years, several raw data datasets [9]–[11] have been released to perform classification [12], [13], object detection [11], [14]–[16] or segmentation [11], [17], [18] on raw data tensors. However, most detection or segmentation methods are static. In other words, they use only a single image as input without exploiting the correlations among successive frames.

For automotive applications, time is key information which can be exploited to learn temporal patterns between successive frames in videos for example. In radar, given the modulation and the characteristic of the signal of FMCW (Frequency Modulated Continuous Wave) radars, the data includes temporal information (*e.g.* Doppler effect), which is a crucial value in autonomous driving. The use of time in radar makes it possible to learn the dynamics of the objects held in the radar signal, handle the variation in the shape of the object over time, and reduce the noise between successive frames (induced by the movement of the surrounding object and the vehicle itself). Recent efforts have been made to exploit temporal relationships between raw data radar frames using multiple frames for detection or segmentation tasks. Mainly, Ouaknine *et al.* [17] use 3D convolutions for multi-view radar semantic segmentation. [19] and [20] also take advantage of 3D convolutions for object detection on RA maps. In [21], Major *et al.* use ConvLSTM to detect cars in RA view and [22] processes sequences of two successive radar frames to learn the temporal relationship between objects.

This paper presents a new convolutional and recurrent neural network (CRNN) for radar spectra. Unlike most multi-frame radar object detectors, our model is causal, which means

[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.

The code and the pre-trained weights is available at https://github.com/colindecourt/record

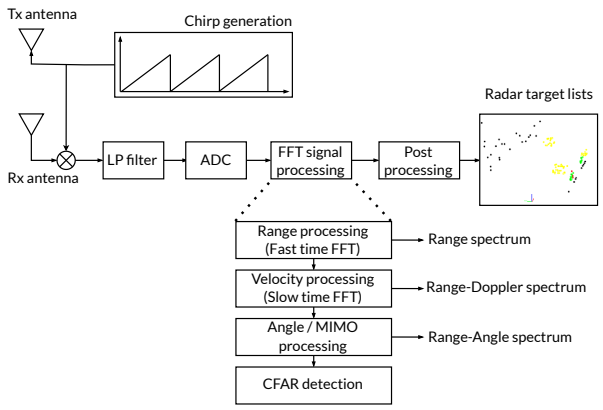


Fig. 1. FMCW radar overview

we only use past frames to detect objects. This characteristic is crucial for ADAS applications because such systems do not have access to future frames. To learn spatial and temporal dependencies, we introduce a model consisting of 2D convolutions and convolutional recurrent neural networks (ConvRNNs). Additionally, we use efficient convolutions and ConvRNNs (inverted residual blocks and Bottleneck LSTMs) to reduce the computational cost of our approach. Our model is end-to-end trainable and does not require pretraining or multiple training steps. We present a generic method that can process either RA, RD or RAD spectra and outperforms state-of-the-art architectures on different tasks. To our knowledge, this is the first fully convolutional recurrent network for radar spectra.

Section II introduces fundamental radar signal processing. Then in Section III, we describe our approach and the proposed model. IV presents the prior art to this work. We present the results of our experiments on the ROD2021 dataset [10] and the CARRADA [9] dataset in Section V. Finally, we discuss our results and conclude the paper in Section VI.

II. Radar background

Radar is an active sensor that transmits radio frequency (RF) electromagnetic (EM) waves and uses the reflected waves from objects to estimate the distance, velocity, and angle of arrival of these targets [23]. Automotive radars emit a particular waveform called FMCW. An FMCW radar periodically transmits P chirps (a frequency-modulated pulse whose frequency increases linearly with time) over M_{Rx} receiving antennas and M_{Tx} transmitting antennas to estimate the range, the velocity and the Direction-of-Arrival (DOA) of the targets. For the m^{th} antenna, we express a single FMCW pulse as:

$$s_n(t) = e^{j2\pi(f_c+0.5Kt)t} \quad 0 \leq t \leq T, \tag{1}$$

where f_c is the carrier frequency, K is a modulation constant, and T is the duration of the chirp (fast time).

As shown in Figure 1, it is possible to estimate the distance to the target from the ADC data by applying a first discrete Fourier transform along the fast time index (*i.e.* for every chirp). We obtain the velocity by computing a second discrete Fourier transform over the slow time index (*i.e.* for each chirp index). This second Fourier transform allows measuring the frequency shift between received chirps resulting in the Doppler frequency and hence the velocity of targets. These two successive Fourier transforms result in a range-Doppler (or range-velocity) spectrum.

From the RD spectrum, it is possible to obtain a list of targets which contains the position x, y, the radial velocity v_r, the DOA θ and the radar cross section σ. This is done by applying thresholding algorithms like CFAR [3], followed by the DOA estimation and post-processing steps (ego-motion compensation, Kalman filtering, classification) [23]. However, these operations reduce the amount of information in the signal.

In multiple receiving antenna scenarios, each antenna sees the reflected signal with a slight time delay. Computing a third discrete Fourier transform along the antenna array allows estimating targets' DOA and results in a range-azimuth-Doppler spectrum. Summing values along the Doppler dimension enable the computation of the range-angle spectrum. Compared to target lists, these representations contain much more information about the environment.

III. Spatio-temporal radar object detector

We aim to design a model to learn the implicit relationship between frames at different spatial and temporal levels recurrently. This section describes the architecture of our single-view spatio-temporal encoder and decoder. We also introduce a multi-view version of our model designed to learn spatial and temporal features in different views (*e.g.* RD, AD, and RA) simultaneously.

A. Problem formulation

Let us consider a sequence R of N radar frames ranging from time $k - N + 1$ to k such as: $R = \{r_{k-N+1}, \ldots, r_k\}$. We aim to find the locations of every object in the scene at time k, p_k, based on the N past frames. We define our model with two functions, the encoder E and the decoder D.

We consider a causal model, which means it uses only the past to predict the next time step. For each time step k of the sequence, we consider a recurrent convolutional encoder taking as input the frame at time k and a set of I previous hidden states $H_{k-1} = \{h^0_{k-1}, \ldots, h^i_{k-1}, \ldots, h^{I-1}_{k-1}\}$ with i the index of the recurrent unit if more than one are used. The encoder returns a set of features maps F_k and a set of updated hidden states $H_k = \{h^0_k, \ldots, h^i_k, \ldots, h^{I-1}_k\}$ such that:

$$E(r_k, H_{k-1}) = (F_k, H_k). \tag{2}$$

Because our encoder encodes the past N frames recurrently to predict the position of objects at time step k, our decoder is a fully convolutional decoder that takes as input the encoder's updated hidden states H_k (the memory) and the set of feature maps F_k (spatio-temporal feature maps) such that:

$$D(F_k, H_k) = p_k. \tag{3}$$

As we want to improve the classification accuracy more than the localisation accuracy, we use recurrent layers in the encoding phase only. In encoder-decoder architectures, the

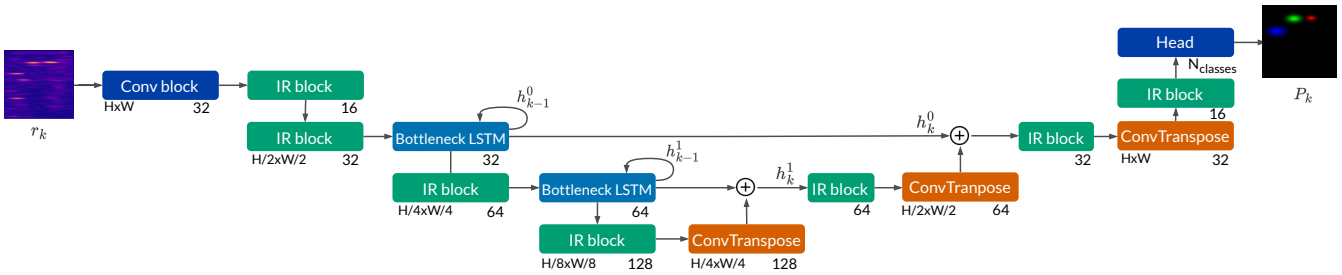


Fig. 2. Model architecture (RECORD). Rounded arrows on *Bottleneck LSTMs* stand for a recurrent layer. Plus sign stands for the concatenation operation. We report the output size (left) and the number of output channels (right) for each layer.

encoder learns to extract an abstract representation of the radar frame relative to the class while the decoder is used for localisation. Using recurrent layers only in the encoding phase allows the encoder to encode spatio-temporal relationships at the object level to improve the objects' representation.

B. Spatio-temporal encoder

An overview of our single-view architecture is shown in Figure 2. We propose a fully convolutional recurrent encoder (left part of Figure 2). In other words, our encoder mixes 2D convolutions and ConvRNNs. We use 2D convolutions to learn spatial information and reduce the size of inputs. To reduce the number of parameters of the model and its computation time, we use inverted residual (IR) bottleneck blocks from the MobileNetV2 [24] architecture instead of classic 2D convolutions for most of the convolutional layers of the model. IR bottleneck is a residual block based on depthwise separable convolutions that use an inverted structure for efficiency reasons. Then, we propose inserting ConvLSTM [25] cells between convolutional layers to learn the temporal relationship between frames. Similarly to the convolutions, we replace the classic ConvLSTM with an efficient one proposed in [26] by Liu and Zhu called bottleneck LSTM. Contrary to a classic ConvLSTM, authors replace convolutions with depthwise-separable convolutions, which reduces the required computation by a factor of eight to nine. Additionally, $tanh$ activation functions are replaced by $ReLU$ activation functions. In this work, we use two bottleneck LSTMs, as a result, $I = 2$. Such a layout enhances spatial features with temporal features and vice versa.

We follow the MobileNetV2 structure by first applying a full convolution to increase the number of channels followed by a single IR bottleneck block. Except for the first IR bottleneck block, we set the expansion rate γ to four. Next, we apply two blocks composed of three IR bottleneck blocks followed by a bottleneck LSTM to learn spatio-temporal dependencies. Because the computational cost of bottleneck LSTMs is proportional to the input size, we use a stride of two in the first IR bottleneck block to reduce the input dimension. We insert these bottleneck LSTMs in the middle of the encoder to not alter the spatial information too much. Finally, we refine the spatio-temporal feature maps obtained from the bottleneck LSTMs by adding three additional IR bottleneck blocks.

Because we treat data sequences, it is desirable to calculate normalisation statistics across all features and all elements for each instance independently instead of a batch of data (a batch can be composed of sequences from different scenes). As a result, we add layer normalisation before sigmoid activation on gates o_t, i_t and f_t in the bottleneck LSTM, and we adopt layer normalisation for all the layers in the model.

C. Decoder

As described in Section III-A, our decoder is a 2D convolutional decoder which takes as input the last feature maps of the encoder (denoted F_k) and a set of two hidden states $H_k = \{h^0_k, h^1_k\}$. Our decoder is composed of three 2D transposed convolutions followed by a single IR block with an expansion factor γ set to one, and a layer normalisation layer. Each transposed convolution block upsample the input feature map by two. Finally, we use two 2D convolutions as a classification/segmentation head (depending on the task) which projects the upsampled feature map onto the desired output.

The U-Net architecture [27] has popularised skip connections between the encoder and decoder. It allows precise localisation by combining high-resolution and low-resolution features. We, therefore, adopt skip connections between our encoder and our decoder to improve the localisation precision. To prevent the loss of temporal information in the decoding stage, we use the hidden states of each bottleneck LSTM (denoted by h^0_k and h^1_k in Figure 2) and concatenating them with the output of a transposed convolution operation to propagate in the decoder the temporal relationship learned by the encoder.

D. Multi-view spatio-temporal object detector

The preceding sections describe a spatio-temporal radar object detection architecture for single view inputs (*i.e.* RA or RD). However, using more than one view to represent targets in their entirety might be desirable. In other words, to simultaneously find the position (distance, angle), the velocity and the class of targets. In this section, we propose to extend the previous architecture to a multi-view approach. We follow the paradigm of Ouakine *et al.* [17] by replicating three times the encoder proposed in Section III-B (one for RA view, one for RD view and one for AD view, see Figure 2). Then, the latent space of each view is concatenated to create a multi-view latent space. We use two decoders to predict objects' positions in all dimensions (RA and RD). One for the RA

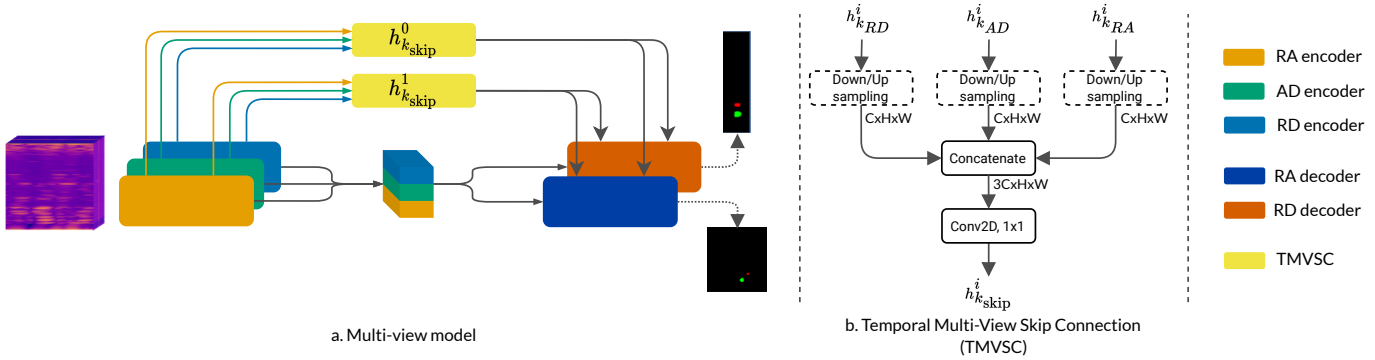


Fig. 3. Multi-view model architecture (MV-RECORD). We use the encoder described in Figure 2 for each view. Dashed boxes denote an optional operation applied only if the feature maps have different shapes. Gray arrows denote the same output.

view and one for the RD view. The multi-view latent space is the input of these decoders.

In Section III-C, we use the hidden states of each bottleneck LSTMs for the skip connection to add the temporal information in the decoding part. For the multi-view approach, we want to take advantage of the multi-view and the spatio-temporal approaches in the skip connections to supplement decoders with data from other views (*e.g.* add velocity information in the RA view). Similarly to the multi-view latent space, we concatenate the hidden states from RD, RA and AD views. This concatenation results in a set of concatenated hidden states $H_k = \{h^0_{k_{skip}}, h^1_{k_{skip}}\}$. We describe the operation to obtain H_k in Figure 3b. We concatenate H_k in the same way as in the single view approach. We call this operation Temporal Multi-View Skip Connections (TMVSC). Figure 3 illustrates the multi-view architecture we propose.

E. Training procedure

We propose two training methods to train RECORD and MV-RECORD: *online* and *buffer*, summarised in Figure 4. Let us denote by $R = \{r_{k-N+1}, \ldots, r_k\}$ a sequence of N radar frames ranging from time $k-N+1$ to k, $P = \{p_{k-N+1}, \ldots, p_k\}$ the objects' position in the sequence (the ground truth) and $\mathcal{L}$ the loss function we aim to minimise.

a) Buffer training: We adopt a many-to-one paradigm when training using the *buffer* approach. We train the model to predict only the position of the objects in the last frame r_k as shown in Figure 4a. Therefore, given a sequence of N radar frames, we minimise the following loss function:

$$\mathcal{L}(\hat{p}, p) = \mathcal{L}(\hat{p}_k, p_k) \tag{4}$$

where k is the last time step of the sequence. *Buffer* training forces the model to focus on a specific time window and to learn a global representation of the scene. However, in inference, the model must process N frames sequentially to make a prediction. Therefore, we propose to train the model differently using a many-to-many paradigm to improve the model's efficiency in inference.

b) Online training: We adopt a many-to-many paradigm when training using the *online* approach. We train the model to predict the position of the objects for every frame in the sequence R as shown in Figure 4b. Therefore, given a sequence of N radar frames, we minimise the following loss function:

$$\mathcal{L}(\hat{p}, p) = \sum_{k=1}^{N} \mathcal{L}(\hat{p}_k, p_k) \tag{5}$$

Online training pushes the model to use previous objects' positions to make a new prediction. It encourages the model to keep only relevant information from the previous frames. *Online* training requires training with longer sequences but allows data processing one by one (no buffer) in inference. In contrast to the *buffer* approach, the hidden states are not reset in inference.

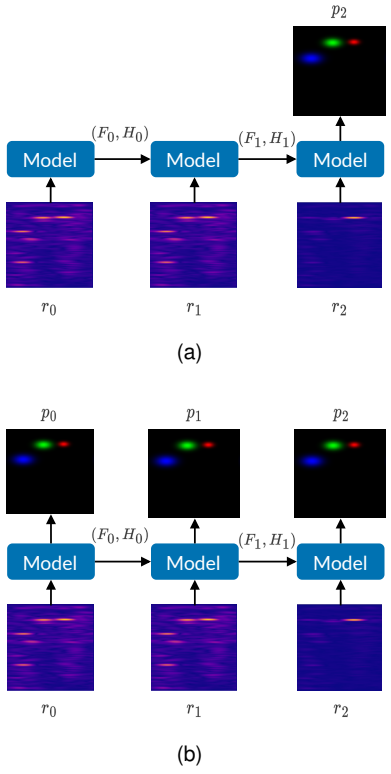


Fig. 4. Training procedures with $N = 3$. (a) Buffer training procedure (many-to-one). (b) Online training procedure (many-to-many).

IV. Related work

A. Sequential object detection and segmentation in computer vision

Object detection and segmentation are fundamental problems in computer vision. However, the majority of object detection and segmentation algorithms have been developed on static images. For some applications (robotics, autonomous driving, earth observation), processing sequences of images is desirable. Due to motion blur or object occlusion, it is sub-optimal to directly apply classical object detectors, or segmentation algorithms [28] on successive frames. To exploit the temporal information in sequences, optical flows [29], [30], recurrent networks with or without convolutions [26], [31]–[34], attention [30], [35], transformers [30], [36], aggregation methods [37] or convolutions [38], [39] were widely used.

In [31] and [26], authors propose to transform the SSD object detector in a recurrent model by adding ConvLSTM cells on the top of the regression and classification head for [31] and between the last feature map of the feature extractor and the detection head for [26]. However, these models predict bounding boxes (not segmentation masks) and learn to find temporal relationships only between feature maps. In [33], authors present a recurrent model for one-shot and zero-shot video object instance segmentation. Contrary to previous methods and ours, they use a fully recurrent decoder composed of upsampling ConvLSTM layers to predict instance segmentation masks. Another approach proposed by Sainte Fare Garnot and Landrieu in [35] consists in using temporal self-attention to extract multi-scale spatio-temporal features for panoptic segmentation of satellite image time series.

Despite some models being trained online (*i.e.* no access to future frames during training), [26], [31], [32], some models using a sequence of images are non-causal in inference and use the video in its entirety. Thus, [34] propose a causal recurrent flow-based method for online video object detection. Their method only uses the past and the current frame from a memory buffer and learns short-term temporal context using optical flow and long-term temporal context using a ConvLSTM. Nevertheless, this method needs to learn optical flow to get accurate results, which is not possible in radar.

B. Sequential object detection and segmentation in radar

In radar, object detection or segmentation algorithms [14], [16], [40]–[42] that do not use time suffer from low performances for similar classes such as pedestrians and bicyclists [14], [17]. According to the Doppler principle, motion information is held in the radar signal and should help to differentiate a pedestrian (non-rigid body, motion information widely distributed), and a car (rigid body, more consistent motion information) [19]. 3D convolutions are primarily used in radar to learn spatio-temporal dependencies between frames. Methods such as [19], [20], [41], [43] adopt 3D encoder-decoder architectures where they predict the position of objects for N successive frames. Despite their performances, these methods require a buffer of N frames in memory to work and are not really online methods in inference, as the convolutional kernel is applied over past and future frames. On the contrary, our approach doesn't access future frames either in training or inference. Additionally, the number of parameters of models using 3D convolutions is huge for real-time applications (34.5M for RODNet-CDC [19], 104M for RAMP-CNN [41]). Consequently, Ju *et al.* [20] introduced Dimension Apart Module (DAM), a lightweight module for spatio-temporal features extraction on RA maps that can be integrated into U-Net style network architecture. Alternatively, Ouaknine *et al.* propose in [17] to use 3D convolutions to encode the spatial information of the N past frames in an online setting. Similarly to other 3D convolutions-based methods, TMVA-Net [17] has a lot of parameters compared to our approach.

Kaul *et al.* [18] propose a model without 3D convolutions where the time information is stacked in the channel dimension. [44] aggregates point clouds of different time steps to increase the resolution of the radar point cloud. More recently, Liu *et al.* [8] propose an approach inspired by the transformer architecture and based on computer vision-based feature extractors to exploit temporal dependencies between objects in two successive frames. Finally, Major *et al.* [21] follow [45] by using a ConvLSTM over the features of a multi-view convolutional encoder. Even though this model is similar to ours, the LSTM cell is applied only to the learned cartesian output before the detection head, and the proposed model only detects cars. Additionally, this model produces bounding boxes, which is not very accurate for radar, is not end-to-end trainable and requires pre-training of a non-recurrent version of it before.

V. Experiments

A. Single view object detection

a) Dataset: We prototype and train RECORD on the ROD2021 challenge dataset[1], a subset of the CRUW dataset [10]. Due to its high frame rate (30 fps), this dataset is well-suited to evaluate the temporal models. The ROD2021 dataset contains 50 sequences (40 for training and 10 for testing) of synchronised cameras and raw radar frames. Each sequence contains around 800-1700 frames in four different driving scenarios, *i.e.*, parking lot (PL), campus road (CR), city street (CS), and highway (HW).

The provided data of the ROD2021 challenge dataset are pre-processed sequences of RA spectra (or maps). Annotations are confidence maps (ConfMaps) in range-azimuth coordinates that represent object locations (see Figure 2). According to [19] one set of ConfMaps has multiple channels, each representing one specific class label, *i.e.*, car, pedestrian, and cyclist. The pixel value in the cls-th channel represents the probability of an object with class cls occurring at that range-azimuth location. We refer the reader to [19] for more information about ConfMaps generation and post-processing. RA spectra and ConfMaps have dimensions 128×128.

b) Evaluation metrics: We use the metric proposed in [19] to evaluate the models on the ROD2021 challenge datasets. In image-based object detection, intersection over

[1] https://www.cruwdataset.org/rod2021

TABLE I
RESULTS OBTAINED ON THE TEST SET OF THE ROD2021 CHALLENGE FOR DIFFERENT DRIVING SCENARIOS (PL: PARKING LOT, CR: CAMPUS ROAD, CS: CITY STREET AND HW: HIGHWAY). WE REPORT THE BEST RESULTS OVER FIVE DIFFERENT SEEDS. THE BEST RESULTS ARE IN BOLD, AND THE SECOND BESTS ARE UNDERLINED.

Model	AP					AR					Params (M)	GMACS	Runtime (ms)
	Mean	PL	CR	CS	HW	Mean	PL	CR	CS	HW			
RECORD (buffer, ours)	<u>72.8</u>	95.0	<u>67.7</u>	48.3	**77.4**	**82.8**	<u>96.7</u>	73.9	**72.8**	**81.7**	0.69	5.0	61.6
RECORD (online, ours)	**73.5**	**96.4**	**72.5**	49.9	<u>72.5</u>	<u>81.2</u>	96.4	<u>78.1</u>	68.8	<u>77.6</u>	0.69	0.95	6.2
RECORD (no lstm, multi)	65.5	89.9	57.3	43.1	68.9	78.9	93.1	68.2	<u>71.5</u>	75.7	0.47	0.76	5.8
RECORD (no lstm, single)	59.5	85.7	48.5	39.11	64.4	75.1	90.8	62.4	68.9	69.6	0.44	0.35	5.7
DANet [20]	71.9	94.7	65.7	**51.9**	70.0	80.7	96.2	75.1	**72.8**	73.0	0.74	9.1	21.1
UTAE [35]	68.4	92.1	67.4	<u>51.4</u>	65.5	78.4	94.6	74.0	69.7	70.0	0.79	4.1	7.5
T-RODNet [46]	69.9	<u>95.6</u>	**72.5**	48.2	63.7	79.5	**97.2**	**79.1**	70.2	67.2	159.7	91.7	74.0

union (IoU) is mostly used to estimate how close the prediction and the ground truth (GT) are. For our single-view approach, as we predict the location of objects, we utilise the object location similarity (OLS) to match detection and GT. The OLS is defined as:

$$OLS = \exp \frac{-d^2}{2(s\kappa_{cls})^2} \tag{6}$$

where d is the distance (in meters) between the two points in the RA spectrum, s is the object distance from the radar sensor (representing object scale information) and κ_{cls} is a per-class constant that describes the error tolerance for class cls (average object size of the corresponding class). First, OLS is computed between GT and detection. Then the average precision (AP) and the average recall (AR) are calculated using different OLS thresholds ranging from 0.5 to 0.9 with a step of 0.05, representing different localisation error tolerance for the detection results. In the rest of this section, AP and AR denote the average precision and recall for all the thresholds.

c) Evaluation procedure: In the ROD2021 dataset, annotations of test sequences are unavailable.To train and evaluate our models and the baselines similarly, we selected 36 sequences out of 40 to train the model and four for validation. We use the validation set for early stopping. Once the models are trained, we test them on the test set. Finally, we update the prediction on the ROD2021 evaluation platform to evaluate the performance of each model. As for the ROD2021 challenge, the evaluation is done for 70% of the test set.

d) Competing methods: We compare our approach with several radar-based and image-based methods using sequences of multiple radar frames. For the radar-based approach, we first benchmark our model against DANet[2] [20], a 3D convolutional model which won the ROD2021 challenge. Because image-based models are too heavy for our application, we finally contrast our recurrent approach against the attention-based model UTAE [35], which is lighter than image-based approaches and which we can use causally. We found that decreasing the number of channels of UTAE and changing the positional encoding improved the performances (see Table IV). We also consider two variants of our model without LSTMs, one using the time along the channel dimension (*no lstm, multi)* and one using a single frame (*no lstm, single)*.

e) Experimental settings: We use sequences of 32 frames for the *online* training. For validation and testing, frames are processed one by one. For the *buffer* training, we use sequences of 12 frames in both training and evaluation. Here we reset the hidden states every 12 frames. We use this buffer approach for fair comparison with other baselines that also use a buffer, although it is less efficient than the online approach. We optimise our model using the Adam optimiser with learning rate (1×10^{-3}) for the buffer method and 3×10^{-4} for the online method. We decay the learning rate exponentially by a factor of 0.9 every ten epochs. We train all the models using a binary cross-entropy loss.

We use an early stopping strategy to stop training if the model does not improve for seven epochs. To avoid overfitting, we use a stride of four for the buffer model and eight for the online model (i.e., how many frames the model skips between each training iteration) in the training dataset. The stride is set to one in validation and testing as we process data on the fly. We apply different data augmentation techniques during training, such as horizontal, vertical and temporal flipping. We use these settings for all the baselines, except for DANet where we use the settings recommended by the authors. All the models were implemented using the Pytorch Lightning[3] framework and trained on an NVIDIA Quadro RTX 8000 GPU. We run all the models with five different seeds and report the best results in the next paragraph.

f) Results: Table I presents the results of our model and the baselines on the test set of the ROD2021 dataset. Our recurrent approaches generally outperform baselines for both AP and AR metrics; this remains true for most scenarios. The online version of RECORD obtains the best trade-off between performances and computational complexity (parameters, number of multiplications and additions and runtime). Despite having less GMACs than UTAE and DANet, the buffer version of RECORD is the slowest one among all the models. Indeed, for each new frame we need to process the 11 previous ones, which is inefficient. Results show that the online version should be preferred for real-time applications. Additionally, RECORD methods exceed 3D and attention-based methods

[2]The original implementation is not available so we implement it according to author's guidelines. We do not use test augmentation and ensemble learning in this paper.

[3]https://www.pytorchlightning.ai/

TABLE II
COMPARISON OF DIFFERENT TYPES OF CONVRNN. WE TRAIN ALL THE MODELS WITH THE SAME LOSS AND HYPERPARAMETERS. BOTTLENECK LSTM ACHIEVES THE BEST AP WHILE HAVING FEWER PARAMETERS AND GMACS.

ConvRNN type	AP	AR	Params (M)	GMACS
Bottleneck LSTM [26]	**69.8 ± 2.2**	80.2 ± 1.5	0.69	5.0
ConvLSTM [25]	66.63 ± 3.28	79.36 ± 2.39	1.0	11.6
ConvGRU [47]	69.7 ± 2.4	**81.2 ± 1.4**	0.94	9.8

TABLE III
COMPARISON OF DIFFERENT TYPES OF SKIP CONNECTIONS. RESULTS ARE AVERAGED OVER 5 DIFFERENT SEEDS ON THE ROD2021 TEST SET. CONCATENATION IS THE RECORD MODEL, ADDITION STAND FOR A MODEL WHERE WE ADD THE OUTPUT OF TRANSPOSED CONVOLUTIONS TO h_k^i, AND NO SKIP CONNECTION STANDS FOR A MODEL WITHOUT SKIP CONNECTIONS.

Skip connection	AP	AR	Params (M)
Concatenation	**69.8 ± 2.2**	80.2 ± 1.5	0.69
Addition	64.4 ± 5.3	**80.5 ± 1.1**	0.58
No skip connections	63.7 ± 6.2	78.7 ± 3.5	0.58

on static scenarios such as parking lot (PL) and campus road (CR). In PL and CR scenarios, the radar is static and the velocity of targets varies a lot, our recurrent models seem to learn variations of the target's speed better than other approaches. Surprisingly the attention-based method UTAE, initially designed for the segmentation of satellite images, obtains very competitive results with our method and the DANet model. We notice that the approach using the time in the channel dimension reaches lower AP and AR than their counterpart, which explicitly uses time as a new dimension. Finally, training our model without the time and using only a 2D backbone (*no lstm, single)* obtain the lower performance on the test set.

g) Ablation studies: We demonstrate the relevance of using bottleneck LSTMs instead of classic ConvGRUs or ConvLSTMs in Table II. Bottleneck LSTMs reduce the number of parameters and GMACS and achieve higher AP and AR than classic ConvRNNs. Additionally, we show in Table III the AP and the AR of our model with different skip connections. We show that concatenating temporal features with spatial features of the decoder (i.e., our RECORD model) reaches better AP and AR than a method without skip connections, or one where we add the temporal features to the spatial features of the decoder. Nevertheless, the concatenation of features increases the number of parameters and the number of GMACS of the model compared to other approaches.

h) UTAE performances improvement: Table IV depicts the performance improvement of UTAE [35] model with and without positional encoding and with a modified number of channels in the encoder and the decoder. We modify the number of channels of UTAE to match our architecture. We found that decreasing the model size and using positional encoding improve the model's performance. We define positional encoding as the time between the first and the k^{th} frames.

TABLE IV
PERFORMANCES IMPROVEMENT OF UTAE MODEL WITH AND WITHOUT POSITIONAL ENCODING AND WITH THE DEFAULT ARCHITECTURE (UNDERLINED LINE). RESULTS ARE OBTAINED ON THE TEST SET AND ON A SINGLE SEED.

# channels		Pos. enc.	AP	AR	Params (M)
Encoder	Decoder				
16, 32, 64, 128	16, 32, 64, 128	Yes	**68.4**	**78.4**	0.79
16, 32, 64, 128	16, 32, 64, 128	No	46.9	64.3	0.79
64, 64, 64, 128	32, 32, 64, 128	Yes	60.8	77.9	1.1

B. Multi-view semantic segmentation

a) Dataset: To demonstrate the relevance of our method, we train our model for multi-view object segmentation on the CARRADA dataset [9]. The CARRADA dataset contains 30 sequences of synchronised cameras and raw radar frames recorded in various scenarios with one or two moving objects. The CARRADA dataset provides RAD tensors and semantic segmentation masks for both RD and RA views. Contrary to the CRUW dataset, the CARRADA dataset only contains simple driving scenarios (static radar on an airport runway). The frame rate is 10Hz. The objects are separated into four categories: pedestrian, cyclist, car and background. The RAD tensors have dimensions $256 \times 256 \times 64$ and the semantic segmentation masks have respectively dimensions 256×256 and 256×64 for the RA and the RD spectra. For training, validation and testing, we use the dataset splits provided by the authors.

b) Evaluation metrics: We evaluate our multi-view model using the intersection over union (IoU). IoU is a common evaluation metric for semantic image segmentation, which quantifies the overlap between the target mask T and the predicted segmentation mask P. For a single class, IoU is defined as:

$$IoU = |\frac{T \cap P}{T \cup P}|. \tag{7}$$

We then average this metric over all classes to compute the mean IoU (mIoU).

c) Competing methods: We compare our multi-view model with state-of-the-art multi-view radar semantic segmentation models, namely MV-Net and TMVA-Net [17]. Additionally, we train a buffer and an online single-view variant of our RECORD model. We train two different models, one for the RA view and one for the RD view.

d) Experimental settings: As for the ROD2021 dataset, we use two evaluation settings for MV-RECORD: online and buffer. The CARRADA dataset has a significantly lower frame rate than the ROD2021 dataset. In order to match the same time as a single view model, we set the number of input frames to five for the buffer variant and ten for the online one, which corresponds to a time of respectively 0.5 and 1 second. We set the batch size to eight and optimise the model using Adam optimiser with a learning rate of 1×10^{-3} for both buffer and

TABLE V
RESULTS ON THE MULTI-VIEW APPROACH ON CARRADA DATASET. MV-RECORD STANDS FOR OUR MULTI-VIEW APPROACH. RECORD* STANDS FOR A SINGLE-VIEW APPROACH. THE BEST RESULTS ARE IN BOLD, AND THE SECOND BESTS ARE UNDERLINED.

	Model	IoU					Params (M)	GMACS	Runtime (ms)
		mIoU	Bg	Ped	Cycl	Car			
RA	MV-RECORD (buffer, ours)	**44.5**	99.8	<u>24.2</u>	**20.1**	<u>34.1</u>	1.9	22.2	281.8
	MV-RECORD (online, ours)	<u>42.4</u>	99.8	22.1	<u>11.1</u>	**36.4**	1.9	3.7	56.6
	RECORD* (buffer, ours)	34.8	99.7	10.3	1.4	27.7	0.69	8.2	116.1
	RECORD* (online, ours)	36.3	99.8	12.1	3.1	30.4	0.69	2.38	29.6
	TMVA-Net [17]	41.3	99.8	**26.0**	8.6	30.7	5.6	98.0	21.8
	MV-Net [17]	26.8	99.8	0.1	1.1	6.2	2.4	53.3	18.8
RD	MV-RECORD (buffer, ours)	**63.2**	99.6	**54.9**	**39.3**	<u>58.9</u>	1.9	22.2	281.8
	MV-RECORD (online, ours)	58.5	99.7	49.4	26.3	58.6	1.9	3.7	56.6
	RECORD* (buffer, ours)	58.1	99.6	46.6	28.6	57.5	0.69	6.1	58.5
	RECORD* (online, ours)	<u>61.7</u>	99.7	52.1	<u>33.6</u>	**61.4**	0.69	0.59	13.3
	TMVA-Net [17]	58.7	99.7	<u>52.6</u>	29.0	53.4	5.6	98.0	21.8
	MV-Net [17]	29.0	98.0	0.0	3.8	14.1	2.4	53.3	18.8

online methods except for the online multi-view model where the learning rate is set to $3 \times 10^{-}4$.

We decay exponentially the learning rate every 20 epochs with a factor of 0.9. We use a combination of a weighted cross-entropy loss and a dice loss with the recommended parameters described in [17] to train our model as we find it provides the best results. To avoid overfitting, we apply horizontal and vertical flipping data augmentation. We also use an early stopping strategy to stop training if the model's performance does not improve for 15 epochs. Training multi-view models is computationally expensive (around six days for TMVA-Net and five days for ours). As a result, we train models using the same seed as the baseline for a fair comparison. We use the pre-trained weights of TMVA-Net and MV-Net to evaluate baselines.

e) Results: Table V shows the results we obtain on the CARRADA dataset. Our multi-view approaches beat the state-of-the-art model TMVA-Net on the multi-view radar semantic segmentation task while using two times fewer parameters and requiring significantly fewer GMACS. Our approach seems to correctly learn the variety of objects' shapes without complex operations such as the atrous spatial pyramid pooling (ASPP) used in TMVA-Net. We notice that using recurrent units instead of 3D convolutions in a multi-view approach significantly helps to improve the classification of bicyclists and cars, especially on the RA view, where we double the IoU for bicyclists compared to TMVA-Net. However, bicyclists and pedestrians are very similar classes, and improving the detection performance of bicyclists leads to a loss in the detection performance of pedestrians for the RA view. In the RD view, MV-RECORD models outperform the TMVA-Net approach for all classes. We notice a huge gap in RA view performances between the CARRADA dataset and the CRUW dataset, as well as between the two views of the CARRADA dataset. We hypothesise that the small frame rate of the CARRADA dataset might cause these differences. Indeed, the RD view contains Doppler information, enabling one to learn the dynamics of targets. However, the RA view might not contain as much motion information as in the ROD2021 dataset, where the frame rate is higher, allowing the network to learn the dynamics of targets even in the RA view. Unfortunately, we cannot share the same analysis for the online multi-view approach. Compared to the results on the ROD2021 dataset, where the online approach performs better than the buffer one, we could not find proper training settings for the online multi-view model. Despite MV-RECORD online reaching higher IoU than TMVA-Net on the RA view, this model performs similarly with TMVA-Net on the RD view but has significantly lower IoU than the MV-RECORD buffer approach. We think these differences are mostly optimisation problems. Indeed, we show the online training outperforms the buffer training when using a single view on both the ROD2021 (Table I) and the CARRADA dataset. Especially on the RD view, our single view and online model outperforms TMVA-Net without using the angle information, with 8 times fewer parameters and less computations. This confirms that the low frame rate of the CARRADA dataset limits the motion information that the recurrent layers can learn. Finally, despite having fewer GMACS and parameters than TMVA-Net, our multi-view model (buffer) is much slower in inference than TMVA-Net and is unsuitable for real-time application. The online version is faster and should be preferred for real-time applications. Decreasing the size of the feature maps in the early layer of the network might help to increase the inference speed of the model. Also, we notice using a profiler that the LayerNorm operation takes up to 90% of the inference time for the multi-view models and up to 70% of the inference time for the single-view models. Replacing layer normalisation with batch normalisation should speed up the runtime of our approaches. Given the good results of the single-view approach (especially for the RD view), we recommend using our model for single-view inputs, as RECORD was originally designed for this single-view object detection.

C. Discussion

a) The difference with the results in the DANet paper [20]: Experiments in Section V-A show that DANet produces a 71.9 AP and 79.5 AR which is different from the results announced in the original paper. The code of DANet being unavailable, we implemented it according to the author's

guidelines. Although we obtained the same number of parameters announced for the DAM blocks, our implementation has 740k parameters instead of the 460k announced in the paper. Beyond the implementation, the training and evaluation procedure in our paper is different from the one in the DANet [20] paper. While DANet is trained on the entire training set, we trained it on 36 carefully chosen sequences for a fair comparison with other models. Also, DANet authors use the following techniques when testing the model to improve the performance: test-time augmentation (TTA), ensemble models and frame averaging. Because DANet predicts frames by a batch of 16 with a stride of four, the authors average the overlapping frames (12 in total) in inference. Together, those techniques boost the performance of DANet around ten points, according to the ablation studies in DANet's paper, which is coherent with the gap between our scores and the ones from DANet paper. While applying TTA, ensemble models and training all the models using all the sequences would certainly also improve the global performance of all the models in Table I, we preferred comparing the architectures on a simpler but fair evaluation.

b) General discussion: Here we discuss and analyse the results presented in Sections V-A and V-B. First, radar data differs from LiDAR and images. The most critical differences being 1) the data is simpler in terms of variety, size, and complexity of the patterns; and 2) the datasets are smaller. We thus believe that our lighter architectures are flexible enough, while being less sensitive than huge backbones and less prone to overfitting for radar data. This mainly explains why Bottleneck LSTMs perform better than ConvLSTMs/ConvGRUs (see Table II). Also, we think convolutional LSTMs are more adapted to radar sequences because 1) convLSTMs learn long-term spatio-temporal dependencies at multiple scales, which 3D convolution cannot do because of the limited size of the temporal kernel; 2) LSTMs can learn to weigh contributions of different frames which can be seen as an adaptive frame rate depending on the scenarios and the speed of vehicles; 3) Hidden states keep the position/velocity of objects in previous frames in memory and use it to predict the position in the next time steps. Indeed, we show that, except for MV-RECORD, which is hard to optimise, online methods generally perform better than buffer ones while having lower computational cost (GMACs and inference time).

To conclude, although multi-view methods are interesting for research purposes, we find them difficult and long to optimise. Current radar generations generally detect targets in the range-Doppler view and compute the direction of arrival for each detected target to save computation time. As the RAD cube is cumbersome to compute and store in memory, we suggest using our model on single-view inputs (RD or RA), depending on the desired application.

VI. Conclusion

In this work, we tackled the problem of online object detection for radar using recurrent neural networks. Contrary to well-known radar object detectors, which use a single frame to detect objects in different radar representations, we learn spatial and temporal relationships between frames by leveraging characteristics of FMCW radar signal. We propose a new architecture type that iteratively learns spatial and temporal features throughout a recurrent convolutional encoder. We designed an end-to-end, efficient, causal and generic framework that can process different types of radar data and perform various detection tasks (key point detection, semantic segmentation). Our methods outperform competing methods on both CARRADA and ROD2021 datasets. Notably, our models help distinguish pedestrians and cyclists better and learn the target variations better than 3D approaches.

The main challenge in the near future will be to embed this model onboard real cars. This will require a more involved training, a quantisation of the model, and improved data augmentation or domain adaptation strategies to cope with the limited amount of labelled data.

References

[1] 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.

[2] R. Urtasun, P. Lenz, and A. Geiger, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2012, pp. 3354–3361.

[3] 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.

[4] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[5] A. Palffy, E. Pool, S. Baratam, J. F. Kooij, and D. M. Gavrila, "Multi-class Road User Detection with 3+ 1D Radar in the View-of-Delft Dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4961–4968, 2022.

[6] 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.

[7] 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.

[8] J. Liu, W. Xiong, L. Bai, Y. Xia, T. Huang, W. Ouyang, and B. Zhu, "Deep Instance Segmentation with Automotive Radar Detection Points," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022, conference Name: IEEE Transactions on Intelligent Vehicles.

[9] 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.

[10] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 2815–2824.

[11] J. Rebut, A. Ouaknine, W. Malik, and P. Pérez, "Raw High-Definition Radar for Multi-Task Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17 021–17 030.

[12] M. Ulrich, C. Gläser, and F. Timm, "DeepReflecs: Deep Learning for Automotive Object Classification with Radar Reflections," in *2021 IEEE Radar Conference (RadarConf21)*, May 2021, pp. 1–6, iSSN: 2375-5318.

[13] K. Patel, W. Beluch, K. Rambach, M. Pfeiffer, and B. Yang, "Improving Uncertainty of Deep Learning-based Object Classification on Radar Spectra using Label Smoothing," in *2022 IEEE Radar Conference (RadarConf22)*, Mar. 2022, pp. 1–6.

[14] C. Decourt, R. VanRullen, D. Salle, and T. Oberlin, "DAROD: A Deep Automotive Radar Object Detector on Range-Doppler maps," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2022, pp. 112–118.

[15] M. Meyer, G. Kuschk, and S. Tomforde, "Graph Convolutional Networks for 3D Object Detection on Radar Data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2021, pp. 3060–3069.

[16] R. Franceschi and D. Rachkov, "Deep learning-based Radar Detector for Complex Automotive Scenarios," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 303–308.

[17] A. Ouaknine, A. Newson, P. Pérez, F. Tupin, and J. Rebut, "Multi-View Radar Semantic Segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 671–15 680.

[18] P. Kaul, D. De Martini, M. Gadd, and P. Newman, "RSS-Net: weakly-supervised multi-class semantic segmentation with FMCW radar," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 431–436.

[19] Y. Wang, Z. Jiang, Y. Li, J.-N. Hwang, G. Xing, and H. Liu, "RODNet: 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.

[20] B. Ju, W. Yang, J. Jia, X. Ye, Q. Chen, X. Tan, H. Sun, Y. Shi, and E. Ding, "DANet: Dimension Apart Network for Radar Object Detection," in *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 2021, pp. 533–539.

[21] B. Major, D. Fontijne, A. Ansari, R. Teja Sukhavasi, R. Gowaikar, M. Hamilton, S. Lee, S. Grzechnik, and S. Subramanian, "Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[22] P. Li, P. Wang, K. Berntorp, and H. Liu, "Exploiting Temporal Relations on Radar Perception for Autonomous Driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 17 071–17 080.

[23] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive Radars: A Review of Signal Processing Techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., 2015.

[26] M. Liu and M. Zhu, "Mobile Video Object Detection With Temporally-Aware Feature Maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Springer International Publishing, 2015, pp. 234–241.

[28] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[29] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-Guided Feature Aggregation for Video Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[30] Y. Yu, J. Yuan, G. Mittal, L. Fuxin, and M. Chen, "BATMAN: Bilateral Attention Transformer in Motion-Appearance Neighboring Space for Video Object Segmentation," *arXiv preprint arXiv:2208.01159*, 2022.

[31] X. Li, H. Zhao, and L. Zhang, "Recurrent RetinaNet: A Video Object Detection Model based on Focal Loss," in *International conference on neural information processing*. Springer, 2018, pp. 499–508.

[32] A. Pfeuffer, K. Schulz, and K. Dietmayer, "Semantic Segmentation of Video Sequences with Convolutional LSTMs," in *2019 IEEE intelligent vehicles symposium (IV)*. IEEE, 2019, pp. 1441–1447.

[33] C. Ventura, M. Bellver, A. Girbau, A. Salvador, F. Marques, and X. Giro-i Nieto, "RVOS: End-To-End Recurrent Network for Video Object Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[34] C. Zhang and J. Kim, "Modeling Long-and Short-Term Temporal Context for Video Object Detection," in *2019 IEEE international conference on image processing (ICIP)*. IEEE, 2019, pp. 71–75.

[35] V. S. F. Garnot and L. Landrieu, "Panoptic Segmentation of Satellite Image Time Series With Convolutional Temporal Attention Networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 4872–4881.

[36] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "SSTVOS: Sparse Spatiotemporal Transformers for Video Object Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 5912–5921.

[37] Y. Chen, Y. Cao, H. Hu, and L. Wang, "Memory Enhanced Global-Local Aggregation for Video Object Detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[38] F. Xiao and Y. J. Lee, "Video Object Detection with an Aligned Spatial-Temporal Memory," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[39] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 331–346.

[40] 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.

[41] 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.

[42] 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.

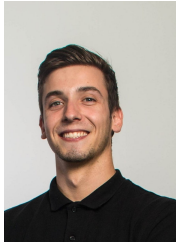
[43] Z. Zheng, X. Yue, K. Keutzer, and A. Sangiovanni Vincentelli, "Scene-Aware Learning Network for Radar Object Detection," in *Proceedings of the 2021 International Conference on Multimedia Retrieval*, 2021, pp. 573–579.

[44] D. Niederlöhner, M. Ulrich, S. Braun, D. Köhler, F. Faion, C. Gläser, A. Treptow, and H. Blume, "Self-Supervised Velocity Estimation for Automotive Radar Object Detection Networks," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 352–359.

[45] M. J. Jones, A. Broad, and T.-Y. Lee, "Recurrent Multi-frame Single Shot Detector for Video Object Detection," in *British Machine Vision Conference (BMVC)*, Sep. 2018.

[46] T. Jiang, L. Zhuang, Q. An, J. Wang, K. Xiao, and A. Wang, "T-rodnet: Transformer for vehicular millimeter-wave radar object detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2023.

[47] N. Ballas, L. Yao, C. Pal, and A. C. Courville, "Delving Deeper into Convolutional Networks for Learning Video Representations," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.



Colin Decourt is a PhD candidate as part of ANITI (Artificial Natural Intelligence Toulouse Institute). He received his Master's degree in telecommunications and artificial intelligence at Bordeaux Institute of Technology in 2020. He started his PhD with ISAE-SUPAERO, CerCo (CNRS UMR5549) and NXP Semiconductors in October 2020. His research focuses on scene understanding for automotive FMCW radars (targets detection, classification and tracking) using artificial intelligence.

His research interest includes image and radar processing, computer vision, and self-supervised learning.

Rufin VanRullen studied Mathematics and Computer Science, then quickly turned to Cognitive Sciences. During his PhD from Université Paul Sabatier, Toulouse, France (2000), he worked on neural coding and rapid visual processing under the supervision of Simon J. Thorpe. As a post-doctoral researcher at the California Institute of technology with Cristof Koch, he became interested in the mechanisms of visual attention and consciousness. Since 2002, he is a Research Director at the CNRS in the Brain and Cognition Research Center (CerCo) of Toulouse. His work in experimental and computational neuroscience explored the role of brain oscillations in cognition. In particular, he demonstrated that rhythmic brain activity makes our perception periodic—a rapid sequence of perceptual cycles, akin to a video sequence. More recently, his research focuses on AI and deep neural networks. He holds a Research Chair in the Artificial and Natural Intelligence Toulouse Institute (ANITI) and has received several European grants (European Young Investigator Award, ERC Consolidator grant, ERC Advanced grant) as well as the CNRS bronze medal in 2007.

Thomas Oberlin received the M.S. degree in applied mathematics from Université Joseph Fourier, Grenoble, France, in 2010, as well as an engineer's degree from Grenoble Institute of Technology. In 2013, he received the Ph.D. in applied mathematics from the University of Grenoble. In 2014, he was a post-doctoral fellow in signal processing and medical imaging at Inria Rennes, France, before joining as an Assistant Professor INP Toulouse – ENSEEIHT and the IRIT Laboratory, Université de Toulouse, France. Since 2019, he is an Associate Professor in artificial intelligence and image processing at ISAE-SUPAERO, Université de Toulouse, France.

His research interests are in signal, image and data processing and in particular time-frequency analysis, representation learning, and sparse/low-rank regularizations for inverse problems.

Since 2022, he serves as an Associate Editor for the IEEE Transactions on Signal Processing.