# Deep Open Space Segmentation using Automotive Radar


1st Farzan Erlik Nowruzi
*University of Ottawa*
Ottawa, Canada
fnowr010@uottawa.ca

2nd Dhanvin Kolhatkar
*Sensor Cortek Inc.*
Ottawa, Canada
dhanvin@sensorcortek.ai

3rd Prince Kapoor
*Sensor Cortek Inc.*
Ottawa, Canada
prince@sensorcortek.ai

4th Fahed Al Hassanat
*Sensor Cortek Inc.*
Ottawa, Canada
fahed@sensorcortek.ai

5th Elnaz Jahani Heravi
*Sensor Cortek Inc.*
Ottawa, Canada
elena@sensorcortek.ai

6th Robert Laganiere
*University of Ottawa*
Ottawa, Canada
laganier@eecs.uottawa.ca

7th Julien Rebut
*Valeo*
Paris, France
julien.rebut@valeo.com

8th Waqas Malik
*Valeo*
Bietigheim-Bissingen, Germany
waqas.malik@valeo.com





***Abstract*—In this work, we propose the use of radar with advanced deep segmentation models to identify open space in parking scenarios. A publically available dataset of radar observations called SCORP was collected. Deep models are evaluated with various radar input representations. Our proposed approach achieves low memory usage and real-time processing speeds, and is thus very well suited for embedded deployment.**



***Index Terms*—Deep Learning, Radar, Dataset, Semantic Segmentation, Parking, Autonomous Driving**


## I. Introduction

Levels of autonomy in driving systems are categorized between level 1 driver assistance systems to fully autonomous level 5 systems. Radar is a low-powered sensor which has been used in the automotive industry for a few decades, offering a less expensive depth estimation solution than lidar. They are a crucial component of autonomous vehicles due to their capability to observe objects and their instantaneous velocities, and their robustness in harsh weather conditions.

There are numerous deep learning model architectures which show state-of-the-art performance on various computer vision applications [3], [5], but they are generally used with input sensors such as camera and lidar. Despite its capabilities, radar still depends on traditional signal processing techniques, and is seldom used with modern deep learning methods. This could be attributed to the unintuitive nature of its information representation, and to the lack of publicly available datasets. Once radar echoes are collected, they require various conversions from time domain to frequency domain to be understandable by human experts. Furthermore, annotating these signals is a challenging task that requires careful consideration.

The contributions presented in this paper are as follows:

1) The first publically available comprehensive dataset including raw radar inputs along with ground-truth open space annotations.
2) A deep segmentation approach that consumes radar signals to estimate open space in a parking lot. Our proposal provides comparable performance to much larger models while being faster and smaller.
3) A comprehensive study of various radar modalities for the purpose of open area segmentation.
4) Evaluation of multiple deep learning approaches on the collected dataset.

## II. Literature Review

Recent work using deep learning for semantic segmentation can use two approaches to generate a refined mask: using an *encoder-decoder* architecture to recover fine segmentation predictions [5], and through the use of *atrous convolutions* to avoid decimating the input's resolution [4]. The former's compuational cost can vary depending on the decoder's architecture, while the latter comes with a sizeable increase in memory footprint, due to the use of large feature maps in the network. *Encoder-decoder* architectures typically make use of an image classification network as its *encoder*, while building a *decoder* to recover fine features for segmentation. *Fully Convolutional Network* (FCN) [5] uses transposed convolutions to upsample the output of the encoder and concatenation with low-level features from the encoder to generate a refined segmentation mask. The use of atrous convolutions for segmentation was pioneered in the DeepLabv2 network [4] by employing varying rates for extracting features and segmenting object at different scales. [3] adds a small decoder to a *DeepLabv2* encoder.

Currently, there are a few public datasets available for ADAS applications that, to some extent, include radar information [1], [2]. The radar *Robotcar* dataset [2] was released for scene understanding analysis with radar data. The dataset includes radar, lidar, camera, GPS, and IMU observations. The radar data was collected using a special purpose sensor with much higher resolution and range than average industrial radar and is not designed based on the requirements of the automotive industry. Although this dataset does not provide the raw signal, azimuth-range representation still provides more insight into the radar data than the detected points of [1]. The major disadvantage of this dataset is the lack of ground-truth bounding box information for the objects at the time of writing of this paper. Annotating the radar data is

especially challenging. It is hard to understand the information as displayed in common representations. To address this issue, one can rely on the use of a visual or depth estimation sensor in combination with radar to create the ground truth data.

Most radar processing currently uses traditional techniques. In the case of occupancy grid mapping, it is common to use *Inverse Sensor Models* (ISM), followed by Bayesian filtering. Sless et al. [6] proposes a U-Net inspired segmentation architecture which takes a Bird's Eye View (BEV) input and generates a mask containing a prediction for each pixel: occupied, unoccupied or unobservable. Formulating the problem as a three-class segmentation problem shows an important improvement when compared to traditional methods. This is expanded by [7] by adding of a fourth, *unknown*, class. This latter approach relies more heavily on the certainty of the predictions.

## III. Dataset

To the best of our knowledge, there are no publicly available datasets for radar with accessible Analog-to-Digital Converter (ADC) signals and annotations. To overcome this problem, we collect our own dataset. We moved a car equipped with a side view radar and camera in a parking lot with the objective of identifying the drivable open spaces in the scene. A Linear Frequency Modulation Continuous Wave (LFMCW) radar with 76 Ghz frequency in Multiple Input Multiple Output (MIMO) mode is utilized to collect the environment observations. The usage of the Time Division Multiplexing (TDM) MIMO mode results in 8 virtual channels for the radar information: 2 Tx elements transmit sequentially and 4 Rx elements receive coherently. The resulting dataset is made up of 3913 frames, collected in 11 driving sequences.

To collect any radar data, the first step is to select a set of parameters for the signal. Table I shows the details of the configuration used to capture the data.

TABLE I
DETAILS OF THE CONFIGURATION USED WITH RADAR.

| | |
|---|---|
| Frequency | 76 *Ghz* |
| Maximum Range | 15 *m* |
| Range resolution | 0.12 *m* |
| Unambiguous Velocity | 10.5 *m/s* |
| Velocity resolution | 0.33 *m/s* |
| Field of View | 90° |

A camera is also used in our data capture to assist with the annotation of the data. We fix it to the same frame-rate as the radar and capture images of size 1280x960 pixels. The captured visual information by camera will later be used to create the ground truth labels. All communication between various components and their synchronization is managed through the *Robot Operating System (ROS)* software.

### A. Radar Processing Pipeline

The signals at each Rx element have the same amplitude but different phase values that represent angular spectrum once converted to frequency domain. The radar echoes are transformed to a 3D tensor of Samples-Chirps-Antennas (SCA). The SCA tensor consists of complex numbers. This is the earliest level to process the radar information. At this stage, all the information is in the time domain and there is no spatial coherence between the values. This entails that any model applied to this stage should explicitly or implicitly include layers to extract spatial coherency.

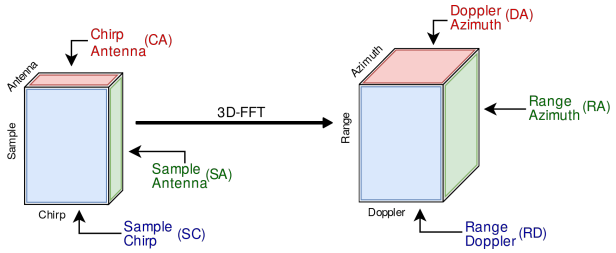


Fig. 1. Radar data representation. (Left) SCA representation. Arranging data along antennas results in Sample-Chirp (SC) view, arranging along chirps results in Sample-Antenna (SA), and Chirp-Antenna (CA) is achieved by arranging data along sample dimension. (Right) Fourier transformation is applied along all three dimensions. Arranging values along azimuth results in Range-Doppler (RD), arrangement along doppler results in Range-Azimuth (RA), and arrangement along range results in Doppler-Azimuth (DA).

Applying Fast Fourier Transform (FFT) along Samples, Chirps, and Antenna dimension results in Range-Doppler-Azimuth (RDA) representation.

Range-Azimuth (RA) is the spatial representation of the received signals. It represents the Bird-Eye-View (BEV) of the environment in polar coordinates. A Polar to Cartesian transformation is regularly used on this representation to calculate the direction of arrival (DoA) point-cloud map in Cartesian coordinates for detected objects. In our dataset, we store SCA, RDA and DoA tensors to cover all the various mainstream levels of inputs to any system.

### B. Annotation Challenge

Annotating radar data is an extremely challenging task as the echo-responses are not easily understandable for human. DoA point cloud is an easier representation to understand, but the level of information is coarse, such that it is extremely difficult to use for annotation. To overcome this issue, we employ the sequence of monocular camera images collected in synchronization with radar. As the calibration of a single monocular camera to the radar sensor is prohibitively difficult, we rely on scene reconstruction techniques to extract odometry information. Once an odometry trace is calculated, we use this trace to accumulate the radar DoA's. The open source software of [8] is used to perform 3D reconstruction and extract the odometry trace.

Open space annotations from accumulated DoA are propagated to corresponding frames using the odometry and radar intrinsic parameters. This ground truth generation pipeline is shown in Figure 2.

As the annotations are propagated from 3D reconstruction, labels include values for locations that are not in the direct

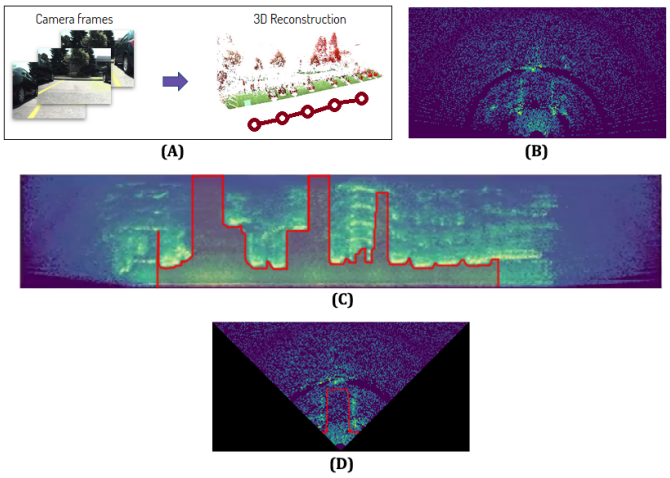


Fig. 2. Ground truth generation pipeline. (A) Camera frames are used to reconstruct the scene and extract odometry. (B) Cartesian DoA from a single frame. (C) Accumulated Cartesian DoA which is used to generate initial annotations. (D) Annotations are distributed to the corresponding frames, cropped for the radar's field of view, and are manually adjusted.

line of sight of the radar. Even though a radar can still detect free space in those regions, they are removed to be consistent with the single frame based annotations of [2].

Note that the inputs to our proposed models only include the radar data, and the output is generated as the occupancy grid map in Cartesian coordinates. Also, it is worth mentioning that the annotation task can be handled much more easily if a laser depth sensor such as Lidar would be used.

The SCORP dataset is available at this link [1].

## IV. Model Architecture

We propose and compare three deep learning approaches inspired by recent work in the field of semantic segmentation: *DeepLabv3+* [3], *Fully Convolutional Networks* (FCN) [5] and *FCN_tiny*. We implement all three of these segmentation models with MobileNet-v2 as their feature extractor.

The DeepLabv2 architecture for segmentation is implemented through two core concepts: reducing the output stride of the feature extractor while using atrous convolutions to generate larger feature maps, and performing atrous spatial pyramid pooling (ASPP) to cover a wider range of object sizes. The *DeepLabv3+* architecture iterates on the architecture of the ASPP, and appendes a small decoder to the DeepLabv2 encoder network, which upsamples the feature extractor's output and combines them with features from earlier layers. We implement a complete version of the DeepLab architecture, referred to as *DeepLabv3+* in our experiments. This version uses atrous convolutions of rates 2, 4 and 6 in its ASPP module. The model's output is then resized to the input size through bilinear interpolation.

The FCN segmentation architecture is a simple encoder-decoder method which uses transposed convolutions to upsample feature maps by a factor of 2. The encoder's output is upsampled and concatenated with lower level feature maps, thereby recovering detailed spatial information. Two of these upsampling steps are used, generating output feature maps 8 times smaller than the network's input. These are then resized to the input size through bilinear interpolation.

Finally, we experiment with a small variation of the FCN architecture, FCN_tiny, with a reduced number of feature maps in each MobileNetv2 layer by 75%, and using a depth of 8 in the decoder feature maps. The resulting model has a much lower number of parameters than the other models.

## V. Experiments

We perform a series of experiments that address various aspects of application of deep learning to the radar sensor. We used three distinct data representation, namely, RAD, RA, and DoA. The combination of these representations as input, Polar and Cartesian outputs, along with the model architectures outlined in previous section, results in our list of experiments. We further evaluate the effect of implicit vs explicit Polar to Cartesian coordinate transformation. We use 3193 frames for training, and 720 for evaluation (9 and 2 driving sequences, respectively).

Mean Intersection-over-Union (Mean-IoU), commonly used for semantic segmentation tasks [5] [4], is selected as the evaluation metric. Mean-IoU is calculated as the average of the Intersection-over-Union (IoU) metric of each class.

### A. Input Modalities

The goal of this experimentation is to identify the effect of various input and output representations on the model performance. We compare three distinct input modalities:

- **RAD:** RAD is a 3D tensor and the convolutions are applied along the last (Doppler) dimension. This input is the polar representation of radar frames for each individual Doppler channel that is generated from the third FFT along antenna dimension of SCA.
- **RA:** RAD input Tensor is summed along the Doppler dimension and the logarithmic value of the matrix is named as RA. This representation is the actual value that traditional methods use to extract the location of their detections. Same as RAD, information in RA is also in polar coordinates.
- **DoA:** DoA input matrix is a Cartesian Bird-Eye-View (BEV) generated from RA. The pixel values in the matrix represent the power received by the radar sensor at that location. The benefit of this modality is its metric coherence with convolutional kernels. This is important as the same convolutional filter in every location of this representation represents the same receptive field.

In order to use segmentation results in various tasks, they need to be in the Cartesian coordinate system. However, the predictions in polar coordinates can be simply converted to Cartesian system. To isolate the effect resulting from having annotations in two different domains, and compare the Polar inputs to Cartesian inputs fairly, we utilize two output representations:

[1] www.sensorcortek.ai/publications/

- **Cartesian ground-truth:** As discussed in section III-B, the open-space is annotated in a parking lot using DoA-input Tensor. Then, the annotated points are used to generate a mask for open-space segmentation. However, we confined the field-of-view of radar to 90°. Cartesian ground truth can be used with all three input models.
- **Polar ground-truth:** After generating the Cartesian ground-truth masks, the annotated points are transformed into the Polar coordinate system. For training, we cropped the RA and RAD input tensors to match the selected field of view.

We conducted experiments where the input tensor is in one domain (i.e. Polar), while the output mask is in another domain (i.e. Cartesian). We expect that the model architecture should learn to adapt the transformation and generate comparable results. Table II shows the results of these experiments. As expected, having the input and output in the same domain in all cases resulted in better performance than learning the domain transformation internally.

We can further observe that using RAD as the input provides the best mean-IOU. This outcome is due to the descriptive information present in RAD that are manually summarized in RA and DoA representations. It is apparent that the model is extracting a better mid-level representation than the manual compression achieved through summation or coordinate transformation done by RA and DoA. RA beats the DoA in performance. This shows that a model using convolutional kernels defined with Cartesian coordinate in mind, is capable of adapting them to the Polar usecase. From a sensor point of view, far points in the BEV map of DoA have much lower information density compared to the closer points. This imbalance is a reason for the lower performance of the DoA input.

TABLE II
MEAN-IOU REPORTED FOR DIFFERENT MODEL ARCHITECTURES.

| Input | Label | FCN_tiny | FCN | DeepLabV3+ |
|---|---|---|---|---|
| RAD-Input | RA-Mask | 83.61 | 83.76 | 82.88 |
| RAD-Input | DoA-Mask | 73.24 | 78.05 | 73.92 |
| RA-Input | RA-Mask | 81.99 | 82.59 | 81.14 |
| RA-Input | DoA-Mask | 77.96 | 78.24 | 77.22 |
| DoA-Input | DoA-Mask | 79.00 | 80.75 | 78.05 |

### B. Model Analysis

*Tensorflow* is used as a back-end platform for training and evaluation of these models. We used *Trainable Softmax Cross-Entropy loss* with initial learning rate of 0.005. This loss is based on Softmax Cross-Entropy loss, with the addition of trainable parameters used to weight each class during loss calculation, thereby avoiding the need to hand-pick appropriate weighting parameters. The optimizer used is RMSProp with a momentum and a decay factor of 0.9. All training of our model architectures was undertaken on *Nvidia Geforce RTX 2080 TI*. Based on our experiments, we noticed that all of our implemented model architecture reached to their optimized weights within 22-30K steps.

Training results show that the FCN model is clearly the better model. In all cases, it achieves superior performance than its opponents. FCN_tiny, a compact version of FCN with only $210k$ parameter - less than $10\%$ of FCN's size - performs comparably to the much larger DeepLabV3+ model. This shows that atrous convolutions are not as helpful as skip architectures for this task. This is most likely caused by the local nature of useful information for this task, which reduces the need for larger atrous rates. Downsampling the input by a factor of 32 provides sufficient information without significantly affecting the quality of the features. On the other hand the holes in an atrous convolution hampers the use of local information.

The proposed FCN_tiny model achieves 324 FPS on the *RTX 2080 TI* GPU and 267 FPS on the *Intel Core i9-9900K* CPU. As such, it is $8\%$ faster than FCN on GPU (301 FPS) and more than twice as fast as FCN on CPU (118.56 FPS).

## VI. CONCLUSION

In this paper, we evaluated various representations of radar data as inputs to deep models, various deep model architectures, and the effect of the polar to Cartesian transformation. FCN_tiny has slightly worse performance than FCN while being an order of magnitude smaller, making it the perfect candidate for Radar-on-Chip integration. To the best of our knowledge, SCORP constitutes the first comprehensive dataset that provides ADC information.

Employing temporal models would increase the performance of occupancy map predictions. We keep this aspect as a topic to address in our future research. We hope this paper and the new dataset will provide an insight into the inner workings of the radar sensor, and enable an increasing number of researchers to easily access the radar data and further develop this field.

## REFERENCES


[1] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," arXiv preprint arXiv:1903.11027, 2019

[2] D. Barnes, M. Gadd, P. Murcutt, P. Newman and I. Posner, "The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset," arXiv preprint arXiv:1909.01300, 2019

[3] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," IEEE European Conf. on Computer Vision (ECCV), pp. 801-818, 2018

[4] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A.L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," IEEE trans. on Pattern Analysis and Machine Intelligence (PAMI), pp. 834-848, 2017

[5] J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 3431-3440, 2015

[6] L. Sless, G. Cohen, B.E. Shlomo and S. Oron, "Road Scene Understanding by Occupancy Grid Learning from Sparse Radar Clusters using Semantic Segmentation," arXiv preprint arXiv:1904.00415, 2019

[7] D. Bauer, L. Kuhnert and L. Eckstein, "Deep, spatially coherent Inverse Sensor Models with Uncertainty Incorporation using the evidential Framework," IEEE Intelligent Vehicles Symp. (IV), pp. 2490-2495, 2019

[8] P. Moulon and P. Monasse and R. Marlet, "Adaptive Structure from Motion with a Contrario Model Estimation," Proc. of the Asian Computer Vision Conference (ACCV), 2012