Delft University of Technology

Automatic Labelling & Semantic Segmentation with 4D Radar Tensors

Sun, Botao; Roldan, Ignacio; Fioranelli, Francesco

DOI
10.1109/ICASSP49660.2025.10890149

Publication date
2025

Document Version
Final published version

Published in
ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings

Citation (APA)
Sun, B., Roldan, I., & Fioranelli, F. (2025). Automatic Labelling & Semantic Segmentation with 4D Radar Tensors. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. https://doi.org/10.1109/ICASSP49660.2025.10890149

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository
as part of the Taverne amendment.

More information about this copyright law amendment
can be found at https://www.openaccess.nl.

Otherwise as indicated in the copyright section:
the publisher is the copyright holder of this work and the
author uses the Dutch legislation to make this work public.

ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10890149

Automatic Labelling & Semantic Segmentation with 4D Radar Tensors

Botao Sun, Ignacio Roldan, Francesco Fioranelli
Microwave Sensing, Signals & Systems (MS3) Group, Dept. of Microelectronics, TU Delft
Delft, The Netherlands
email:B.Sun-6@student.tudelft.nl; i.roldanmontero@tudelft.nl; f.fioranelli@tudelft.nl

***Abstract*—In this paper, an automatic labelling process is presented for automotive datasets, leveraging on complementary information from LiDAR and camera. The generated labels are then used as ground truth with the corresponding 4D radar data as inputs to a proposed semantic segmentation network, to associate a class label to each spatial voxel. Promising results are shown by applying both approaches to the publicly shared *RaDelft* dataset, with the proposed network achieving over 65% of the LiDAR detection performance, improving 13.2% in vehicle detection probability, and reducing 0.54 m in terms of Chamfer distance, compared to variants inspired from the literature.**

***Index Terms*—automotive radar, automatic labelling, semantic segmentation**

I. INTRODUCTION

Radar's robustness in adverse weather and ability to provide dynamic information makes it a valuable complement to cameras and LiDAR in advanced driver-assistance systems (ADAS) [1]. Although deep learning methods for semantic segmentation are well-developed for RGB images and LiDAR point clouds (PCs), their application in radar remains underexplored, especially with 4D radar data that includes additional elevation information [2] [3] [4] [5]. This paper addresses this research gap by proposing a method to perform semantic segmentation directly on the range-azimuth-elevation-Doppler (RAED) tensor. Additionally, a novel automatic labelling process is introduced to generate point-by-point multi-class labels in the *RaDelft* dataset, thus enabling joint detection and classification using radar data.

II. PROPOSED AUTOMATIC LABELLING PROCESS

The work in this paper uses the *RaDelft* dataset, with 4D radar tensors, LiDAR Point Clouds, and RGB images for 16975 frames across 7 different driving scenes recorded in Delft. Additional details on the dataset collection setup are discussed in [6]. 4D radar tensors are generated from the raw data by applying two 2D FFTs to create RAED tensors. Each radar tensor has two channel dimensions: a Range-Azimuth-Doppler (RAD) tensor containing power values, and another one containing single elevation values from 1 to 34, corresponding to the highest power return. Consequently, the dimension of the RAED tensor is $2 \times 128 \times 240 \times 500$ ($N_C \times N_D \times N_A \times N_R$), representing channel, Doppler, azimuth, and range, respectively.

For semantic segmentation, point-by-point multi-class labels are necessary. However, the *RaDelft* dataset does not provide labels, hence an automatic labelling process is proposed and explained in the following sections. The whole process is illustrated in Fig. 1.

A. Automatic labelling approach

1) Preliminary object label generation: Benefiting from the rapid development of 3D object detection only using LiDAR PCs, rough object labels can be generated by applying a pre-trained detection model directly to the LiDAR PCs. Here, we use the anchor-free 'Part-A2 model' [8] trained on the KITTI dataset [9]. As the pre-trained model is not specifically designed or trained for the *RaDelft* dataset, its performance is not optimized. The KITTI dataset uses a 64-layer Velodyne LiDAR, while the *RaDelft* uses a 128-layer RoboSense LiDAR, resulting in denser PCs. This increased density particularly affects the detection of nearby bicycles, which may be misclassified as ground or buildings. To mitigate this, we first downsample the LiDAR PCs to reduce their density before applying the pre-trained model. A high-confidence threshold of 0.5 is set to ensure detection accuracy and retain only bounding boxes with class information. Using seven bounding box parameters such as center coordinates x, y, z, bounding box size d_x, d_y, d_z, and rotation angle $heading$ along the z-axis, the vertex coordinates in Cartesian coordinates are generated. Points within each bounding box are then assigned the corresponding class.

To align with the radar parameters, the whole LiDAR PCs are then cropped to fit the radar view with an azimuth angle from $-70°$ to $70°$, a maximum unambiguous range of 51.4 m, and an elevation angle from $-15°$ to $15°$. In this cropping process, the transformation from LiDAR to radar view is applied. After cropping, the PCs are transformed back to the LiDAR view for calibration. Notably, LiDAR detects ground points, which can form dense clusters and negatively impact radar detection. To mitigate this, the ground points can be adaptively eliminated by 'Patchwork++' [10] so that only four classes of targets remain, i.e., 'scenario objects', 'pedestrians', 'vehicles', and 'bicycles'.

2) Calibration process: Despite setting a high confidence threshold, some mis-labelling events still occur. To address this, RGB images can be used for calibration. As the camera is placed directly below the radar, this alignment ensures that its view overlaps with the central field of view of the radar, which is more crucial than the side view. We employ the transformer-

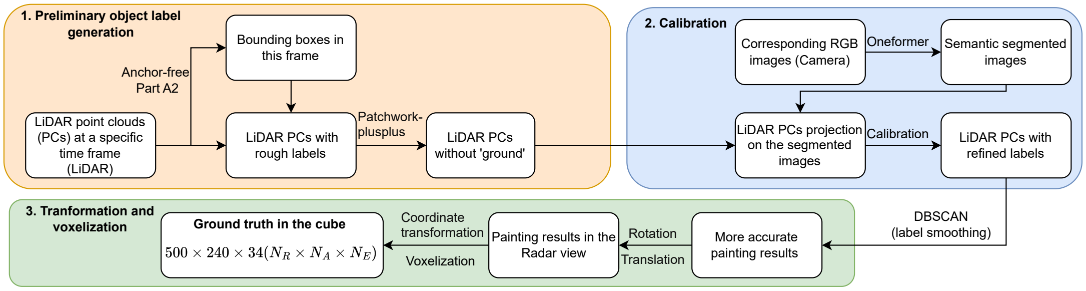
1. Preliminary object label generation
Bounding boxes in this frame
Anchor-free Part A2
LiDAR point clouds (PCs) at a specific time frame (LiDAR)
LiDAR PCs with rough labels
Patchwork-plusplus
LiDAR PCs without 'ground'
2. Calibration
Corresponding RGB images (Camera)
Oneformer
Semantic segmented images
LiDAR PCs projection on the segmented images
Calibration
LiDAR PCs with refined labels
DBSCAN (label smoothing)
3. Tranformation and voxelization
Ground truth in the cube
$500 \times 240 \times 34(N_R \times N_A \times N_E)$
Coordinate transformation
Voxelization
Painting results in the Radar view
Rotation
Translation
More accurate painting results

Fig. 1. Block diagram of the proposed automatic labelling process using LiDAR point clouds (PCs) and RGB images [7]. First, preliminary labels on LiDAR PCs are generated by a pre-trained object detection model. Semantic information from the images is then used to calibrate key labels in the central view, followed by label consistency adjustment using DBSCAN. Finally, point-by-point multi-class labels are generated by coordinate transformation and voxelization.

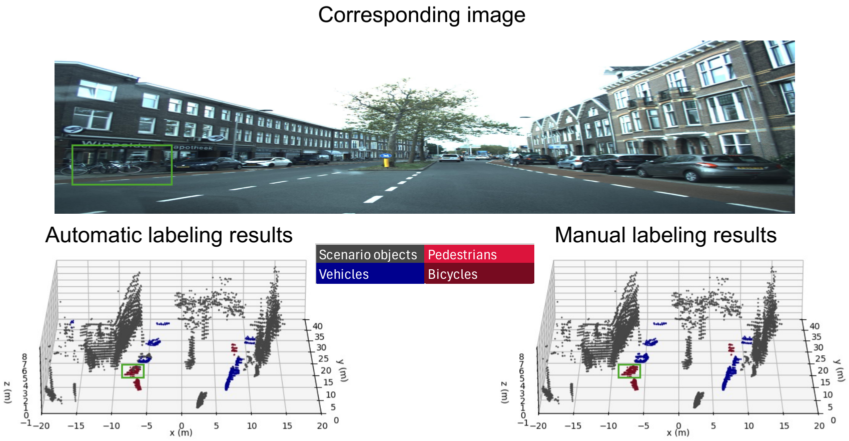

Fig. 2. Automatic vs manual labelling results for a complex scene, with a reference camera image provided. The color bar distinguishes the 4 labelled classes.

based pre-trained model 'OneFormer' [11] to extract semantic information from camera data. Inspired by [12], extrinsic calibration parameters between the camera and LiDAR can be used to estimate the relative rigid transformation between them, i.e., a projection from the 3D LiDAR PCs to the 2D camera images to align the PCs with the image pixels. A 25-m distance threshold is also set to use semantic information from the camera as labels, so that the new labels for the points close to the sensors are assigned to labels generated from the camera, while the distant points retain their original labels.

Despite combining semantic information from two sensors and focusing on central view targets, labelling imperfections remain, such as incomplete labelling of large objects due to small bounding boxes. To address this, label consistency adjustment is applied using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to identify clusters in the PCs [13]. In each cluster, labels are assigned to the same class via majority voting. To detect small clusters and differentiate closely located targets, we set $\epsilon = 0.6$, and $minPts = 100$. Finally, refined labels are generated.

3) Tranformation and voxelization: As the radar data is structured as a cube in polar coordinates, directly using non-uniform LiDAR PCs in Cartesian coordinates as labels is infeasible. Thus, we first rotate and translate the LiDAR PCs to align with the radar view and voxelize the LiDAR PCs to convert them into a cube with the dimension $500 \times 240 \times 34$ ($N_R \times N_A \times N_E$) representing range, azimuth, & elevation, respectively. Each voxel in the cube requires a label, defined by majority voting as 0 for **'empty'**, 1 for **'scenario objects'**, 2 for **'pedestrians'**, 3 for **'vehicles'**, and 4 for **'bicycles'**.

TABLE I
QUANTITATIVE METRICS OF THE DESIGNED AUTOMATIC LABELLING RESULTS COMPARED TO MANUAL LABELLING RESULTS.

Class	**Precision**	**Recall**	**F1-score**
Scenario objects	0.98	0.99	0.99
Vehicles	0.95	0.81	0.88
Bicycles	0.79	0.87	0.83
Pedestrians	0.67	0.72	0.69

B. Automatic labelling results

1) Qualitative results: Fig. 2 compares the proposed automatic vs manual labelling results for a sample frame. The majority of points are correctly labeled in the automatic labelling process, with the exception of some points within the green box highlighted in the figure. There, multiple bicycles are closely parked, complicating the generation of bounding boxes during the preliminary object labelling process. However, through the calibration process, points within the 25m range are correctly calibrated to the ‘bicycles’ class.

2) Quantitative results: For a quantitative evaluation of the proposed automatic labelling, we randomly selected 50 frames from 7 different scenes for manual expert labelling. Precision, recall, and F1-score metrics were then used to compare the automatic labelling with the manual labels. The results are shown in Table I. The performance for ‘scenario objects’ and ‘vehicles’ is satisfactory, while the performance for ‘bicycles’ and ‘pedestrians’ is slightly lower, which is expected with the greater difficulty in automatically labelling smaller targets.

III. PROPOSED SEMANTIC SEGMENTATION NETWORK

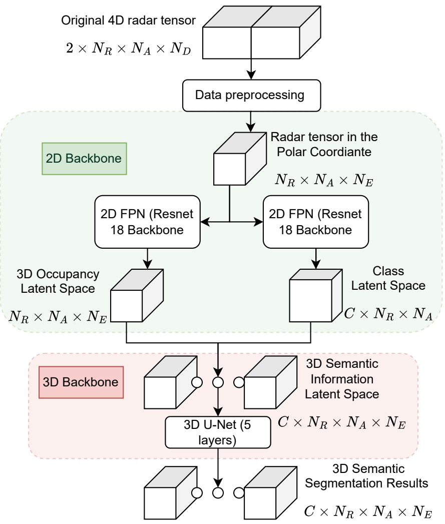
Original 4D radar tensor
$2 \times N_R \times N_A \times N_D$
Data preprocessing
2D Backbone
Radar tensor in the Polar Coordiante
$N_R \times N_A \times N_E$
2D FPN (Resnet 18 Backbone
2D FPN (Resnet 18 Backbone
3D Occupancy Latent Space
$N_R \times N_A \times N_E$
Class Latent Space
$C \times N_R \times N_A$
3D Backbone
3D Semantic Information Latent Space
$C \times N_R \times N_A \times N_E$
3D U-Net (5 layers)
3D Semantic Segmentation Results
$C \times N_R \times N_A \times N_E$

Fig. 3. Proposed radar semantic segmentation approach [7]. A radar tensor in polar coordinates with dimensions $N_R \times N_A \times N_E$ is generated after data preprocessing. Next, a 2D backbone with two individual branches is constructed to generate the 3D occupancy latent space and the class latent space, respectively. These latent spaces are then combined through broadcasting to form a new 3D semantic segmentation latent space. Finally, a 3D backbone is used to produce the output, providing the probability of occurrence for each class at every voxel.

A. Proposed network architecture for semantic segmentation

The original RAED tensor from the *RaDelft* dataset has dimensions of $N_C \times N_D \times N_A \times N_R$, where the elevation information is embedded within the second dimension of N_C, making it less intuitive to use. Moreover, the ground truth generated from LiDAR lacks Doppler information. To address this, the RAED tensor is transformed into a Range-Azimuth-Elevation (RAE) tensor with dimensions $500 \times 240 \times 34$ ($N_R \times N_A \times N_E$). In this transformation, each voxel’s value is the average power from the corresponding spatial location in the original radar tensor in polar coordinates.

Starting from this tensor, the proposed semantic segmentation network begins with a 2D backbone to generate two individual feature maps, as shown in Fig. 3. The first branch uses a Feature Pyramidal Network (FPN) [14] structure with a Resnet-18 backbone to extract the multi-scale features and finally generate a 3D occupancy latent space with the dimension $N_R \times N_A \times N_E$. The second branch also utilizes an FPN structure with a ResNet-18 backbone, but with different channels to produce the class latent space with dimensions $C \times N_R \times N_A$ ($C = 4$ in this work, i.e., 4 classes of interest). These two feature maps are then combined, utilizing the broadcast property, to construct a 3D semantic information latent space with dimensions $C \times N_R \times N_A \times N_E$. The second part of the network uses a 3D backbone to integrate the 3D occupancy with class feature maps. A 3D U-Net [15] with five layers is constructed for this purpose. This architecture produces 3D semantic segmentation results with dimensions $C \times N_R \times N_A \times N_E$, and is more computationally efficient than using a 3D convolutional network.

A combination of weighted cross-entropy (wCE) loss and soft-dice (SDice) loss is used for training. The wCE loss is designed for voxel-wise classification and addresses class imbalance, while the SDice loss is effective for shape segmentation. The network is trained using 80% of the data from five of the seven scenarios, with the remaining 20% reserved for validation. The two remaining scenarios are used for testing as they are completely unseen for the network.

B. Semantic segmentation results

1) Qualitative results: The 3D semantic segmentation outputs are transformed into Cartesian coordinates to compare with LiDAR ground truth visually. Fig. 4 shows the generated radar PCs with class information, the corresponding LiDAR PCs ground truth, and the reference camera image. The results indicate that the generated radar PCs successfully detect most targets within the field of view (FoV). Close vehicles are accurately segmented. Among all vehicles in the FoV, those bounded by the purple box are dynamic, and their segmentation benefits from the radar’s Doppler information. More distant pedestrians or vehicles are harder to segment. For static bicycles on the left hand side, some are correctly segmented into their class, while others are incorrectly classified as vehicles due to their dense arrangement. This indicates that radar is more effective at segmenting close-by and dynamic targets compared to others.

2) Quantitative results: Since radar PCs are noisier than those from LiDAR and include reflections also from optically occluded objects, traditional semantic segmentation metrics such as Intersection over Union (IoU) are not suitable. To simultaneously evaluate detection & segmentation performance, we calculate the detection probability (P_d) and false alarm rate (P_{fa}) for all classes and for each individual class. Notably, we put the ‘pedestrians’ and ‘bicycles’ in the same class to consider vulnerable road users (VRUs). Additionally, the Chamfer distance is computed for all classes, ‘scenario objects’ only, and all targets to measure the similarity between their PCs.

As there are no existing multi-class semantic segmentation methods in 3D space using RAED radar tensors as input and LiDAR PCs as ground truth, direct comparisons are challenging. Thus, we compare our approach with two adapted

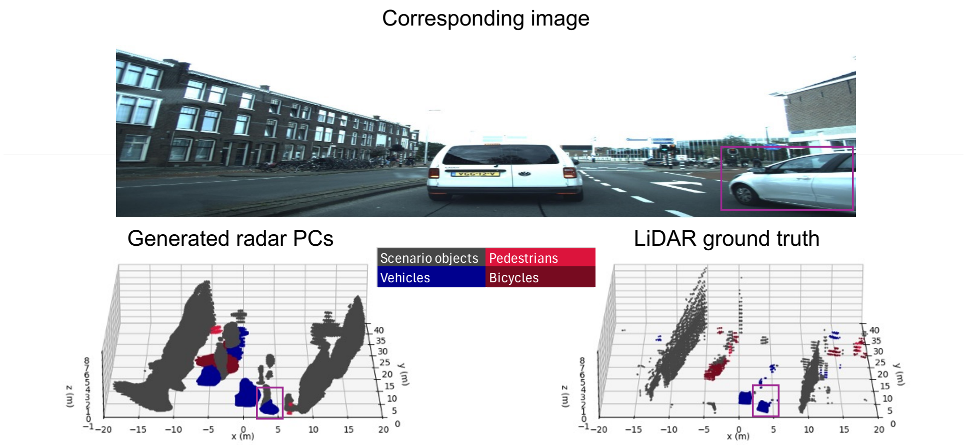

Fig. 4. Generated radar PCs with class information and LiDAR ground truth are presented for a complex scene. The corresponding RGB image is provided for reference. The color bar with the 4 labeled classes is the same as the one used in the labelling process.

TABLE II
PERFORMANCE OF THE PROPOSED SEMANTIC SEGMENTATION NETWORK VS VARIANTS INSPIRED FROM THE LITERATURE. RESULTS IN () ARE FOR VERSIONS OF EACH NETWORK TRAINED WITH THE CLASSES 'PEDESTRIANS' & 'BICYCLES' COMBINED INTO A 'VRU' CLASS. THE BEST PERFORMANCE FOR EACH METRIC IS MARKED IN BOLD.

Methods	P_{d_All} ↑	P_{fa_All} ↓	P_{d_Scenario} ↑	P_{d_Vehicles} ↑	P_{d_VRU} ↑	CD_{All} ↓	CD_{Scenario} ↓	CD_{Targets} ↓
Baseline (VRU)	63.1% **(65.1%)**	3.55% (3.09%)	60.7% **(61.6%)**	34.8% **(47.9%)**	23.1% (25.3%)	2.15m (1.86m)	2.45m (2.15m)	6.73m (5.79m)
Baseline + Res (VRU)	63.0% (63.2%)	3.41% (2.98%)	60.9% (58.4%)	35.9% (42.6%)	24.2% **(25.8%)**	1.97m **(1.77m)**	2.23m **(2.08m)**	5.95m **(5.33m)**
Variant 1 [16] (VRU)	54.2% (60.4%)	3.05% (3.38%)	49.9% (57.9%)	32.9% (34.7%)	21.3% (22.4%)	2.34m (2.31m)	2.79m (2.61m)	7.53m (7.40m)
Variant 2 [17] (VRU)	42.4% (46.8%)	**2.43%** (2.77%)	39.1% (42.1%)	20.8% (28.7%)	13.7% (16.8%)	2.81m (2.69 m)	3.55m (3.23 m)	8.14m (7.79m)

methods from the literature. The first comparison is with the network proposed by [16], which uses the original data from the *RaDelft* dataset as input and adapts their data-driven detection network into a semantic segmentation network, named 'Variant 1'. The second comparison is with [17], where the input is made sparse by retaining only the top 10% of values, and the detection head is removed to modify the network into a semantic segmentation network, named 'Variant 2'. We also consider an improved baseline, where the 3D U-Net in the 3D backbone is modified with a residual structure, 'Baseline + Res'. For each network, we also train a version where 'pedestrians' & 'bicycles' are merged into a 'VRU' class.

The performance evaluation uses the following metrics:

- P_{d_All}: Detection probability for all points.
- P_{d_Scenario}: Detection probability for 'scenario objects'.
- P_{d_Vehicles}: Detection probability for vehicles.
- P_{d_VRU}: Detection probability for VRUs.
- P_{fa_All}: False alarm rate for all points.
- CD_{All}: Chamfer distance for all points.
- CD_{Scenario}: Chamfer distance for 'scenario objects'.
- CD_{Targets}: Chamfer distance for all targets except 'scenario objects'.

Table II presents the metrics averaged across all test data. Radar point clouds achieve over 65% of the LiDAR detection performance and nearly 50% for P_{d_Vehicles}. Compared to the two competing methods, the VRU version of the 'Baseline + Res' and 'Baseline' demonstrate the highest metrics except P_{fa_All}. The generally low P_{d_VRU} comes from the low speed of VRUs, leading to their classification as 'scenario objects'. The relatively high value of CD_{Targets} results from not distinguishing static targets from dynamic ones in the training and test process, causing some static targets to be classified as 'scenario objects', i.e., parked cars. The VRU versions generally perform better than the normal ones due to reduced class complexity in the semantic segmentation process.

IV. CONCLUSIONS

An automatic approach for labelling is presented for automotive datasets, using complementary information from camera and LiDAR. The generated labels are used together with related 4D radar data as inputs to a designed semantic segmentation network to associate class labels to each spatial voxel. Promising results are shown by applying both approaches to the publicly shared *RaDelft* dataset, with the proposed network achieving more than 65% in P_d, improving 13.2% in P_{d_Vehicles}, and reducing 0.54 m in CD_{All} compared to variants inspired from the literature. The dataset and generated labels are publicly available at https://data.4tu.nl/datasets/4e277430-e562-4a7a-adfe-30b58d9a5f0a, with more details of the proposed method in [7].

REFERENCES

[1] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *Ieee Access*, vol. 8, pp. 2847–2868, 2019.

[2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[5] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[6] I. Roldan, A. Palffy, J. F. P. Kooij, D. M. Gavrila, F. Fioranelli, and A. Yarovoy, "A deep automotive radar detector using the radelft dataset," *IEEE Transactions on Radar Systems*, vol. 2, pp. 1062–1075, 2024.

[7] B. Sun, "Autolabeling & semantic segmentation with 4d radar tensors," *TU Delft MSc Thesis*, 2024. [Online]. Available: https://repository.tudelft.nl/record/uuid:f01462b1-0446-481e-9333-9b8d3a488f14

[8] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2647–2664, 2020.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[10] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3D point cloud," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 13 276–13 283.

[11] J. Jain, J. Li, M. T. Chiu, A. Hassani, N. Orlov, and H. Shi, "Oneformer: One transformer to rule universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2989–2998.

[12] J. Domhof, J. F. Kooij, and D. M. Gavrila, "A joint extrinsic calibration tool for radar, camera and lidar," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 571–582, 2021.

[13] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[15] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*. Springer, 2016, pp. 424–432.

[16] I. Roldan, A. Palffy, J. F. Kooij, D. M. Gavrila, F. Fioranelli, and A. Yarovoy, "See further than cfar: a data-driven radar detector trained by lidar," in *2024 IEEE Radar Conference (RadarConf24)*. IEEE, 2024, pp. 1–6.

[17] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, "K-radar: 4d radar object detection for autonomous driving in various weather conditions," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3819–3829, 2022.