RADLER: Radar Object Detection Leveraging Semantic 3D City Models and Self-Supervised Radar-Image Learning

Supplementary Material

7. RadarCity Dataset

7.1. Sensor and Device Setup

The sensor platform for collecting the dataset contains a GoPro HERO11 Black camera and a 77GHz FMCW mmWave radar from Texas Instrument, model AWR1843Boost. The radar's field of view (FoV) after post-processing is 1-33.7m, ±60° horizontally. The GoPro camera uses the ultra-wide mode to enable a horizontal FoV of also ±60°. The frame rate of the camera recording is 30 frames per second (FPS), and for the radar, it is 15 FPS.

Attempts have been made to record radar data at 30 FPS to facilitate data alignment. However, under our radar configuration, the radar data recording could only last around five minutes at 30 FPS due to limited bandwidth for data transmission. The collected radar data is initially stored in the onboard storage and then transmitted to the computer via Ethernet, as depicted in Fig. 10. However, there is a mismatch in the data collection and transmission rate, causing the onboard storage to be filled up after a certain time. Subsequently, any collected data beyond the storage capacity will be abandoned, leading to data loss. To maintain data integrity, it is required to reduce the frame rate of the radar data recording to 15 FPS.

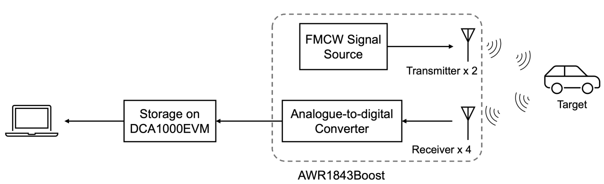

Figure 10. A simplified workflow for the radar data collection. Two transmitters emit signals from the FMCW signal source, while four receivers collect the signals reflected by the target. These signals are then converted to digital signals, stored in onboard storage of the DCA1000EVM board, and transmitted to the computer.

However, to ensure that every second camera image frame matches a radar frame, it is essential to start recording simultaneously on both the camera and radar. The proposed method uses a synchronized software trigger to start the recording on both devices. For the GoPro camera, a lab firmware is available, offering the functionality to synchronize the built-in clock of the GoPro camera with the computer's system clock. Additionally, the GoPro camera can start recording automatically at a given time.

The radar board is controlled through mmWave Studio, a GUI from Texas Instruments that allows configuration and control of mmWave sensor modules and collection of analog-to-digital data for offline analysis. A script has been developed to read the computer's system clock so the buttons to trigger the radar data recording can be clicked at a specified time.

As a result, the GoPro camera and radar can be triggered to record data automatically and simultaneously based on the computer's system clock.

7.2. Data Clusters

After collecting the data, they are organized into clusters based on object types and their positions in the scene. Tab. 3 shows detailed information about the data clusters. Data clusters 1 and 2 contain three common objects on the street: pedestrians, cyclists, and cars. These objects are also the targets of interest in many other machine learning models for radar object detection.

Data cluster 1 and 2 are further split based on where those objects are. Fig. 11a shows an example of data cluster 1 where the cars are on the driving lane, and a cyclist is riding on the bicycle lanes. Fig. 11b, an example data from data cluster 2, contains the same object classes. However, the cyclist there is riding on the pedestrian walk instead of the bicycle lane. This separation is prepared for further analysis of how to use the prior information from the semantic 3D city models, as the initial thought is that the traffic lanes in the street model should facilitate the detection of common objects traveling on them. Objects, such as the bus in Fig. 11c, are out of our current research interest.

7.3. Dataset Annotation Overview

Data clusters 1 and 2 were selected for training and evaluating the performance of RADLER against the baseline RODNet models. Based on the workflow of contrastive SSL, the data in data clusters 1 and 2 are further split into the pretext and downstream task datasets. The pretext task dataset has around 35 000 radar-image data pairs, while the downstream task dataset has 10 000 data pairs with 8000 for training and 2000 for testing. Although SSL can save the efforts of annotating the pretext task dataset, the downstream task dataset still needs to be annotated. In this case, the downstream task dataset is purely manually annotated, hence incurring some offsets in the annotations despite great carefulness. The distribution of the annotations of different object classes in the downstream dataset is presented in

Table 3. Data Clusters in the RadarCity Dataset.

Data Cluster	Object Classes	Objects' Position	Number of Data Pairs
1	pedestrians, cyclists, and cars	In the corresponding lanes	38 800
2	pedestrians, cyclists, and cars	Not in the corresponding lanes	5300
3	Besides mentioned above, trucks, vans, scooters, motorbikes, and bicycles for delivery, etc.	May or may not be in the corresponding lanes	18 700

(a) Data Cluster 1: the objects are in the corresponding lanes as the car and cyclist here.

(b) Data Cluster 2: the objects are not in the corresponding lanes.

(c) Data Cluster 3: objects other than cars, cyclists, and pedestrians.

Figure 11. Examples from different data clusters.

Fig. 12. More cars are annotated than pedestrians and cyclists since those cars are annotated repeatedly among the radar data frames.

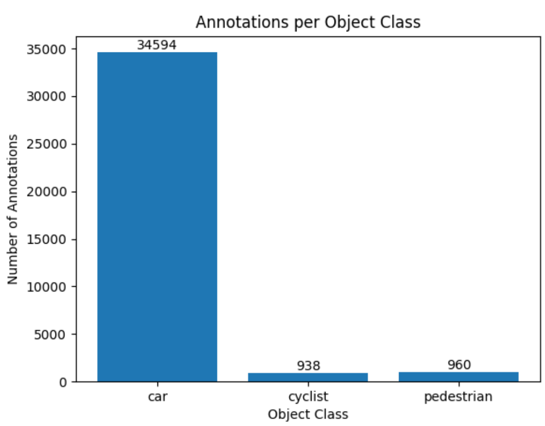

Figure 12. Statistics of the downstream task dataset annotations.

8. Evaluation Results

8.1. Impact of OLS in L-NMS

The OLS value used in L-NMS is chosen through an experimental evaluation. The results of mAP and mAR for RADLER and RODNet under different OLS values as thresholds for L-NMS are shown in Fig. 13. An observation applied for all is that mAP and mAR improve as the OLS value increases.

Notably, RODNet-HG performs better than other models at lower OLS values (from 0.2 to 0.4). This suggests that RODNet-HG produces more separated peaks in the ConfMaps, leading to fewer overlaps and, therefore, lower similarities between the peaks to have them survive the L-NMS. As the OLS value increases, RADLER, particularly the variant integrated with SDM, starts to outperform the others, with its advantage becoming most apparent at an OLS value of 0.8, where it achieves a mAP of nearly 95% and a mAR of around 96%.

While the exact reasons for this performance trend across different OLS values may not be immediately clear, this experiment highlights how the distribution of confidence values in the ConfMaps varies across models. The same OLS threshold applied to different models yields different target lists, indicating the differences in how the confidence values of a detected object from different models are numerically distributed on the ConfMaps, resulting in the quality of the target list generated through L-NMS.

8.2. Additional Qualitative Results

Fig. 14 shows more visualized detection results from the test data. Here, the OLS value for L-NMS is set to be 0.8 for all models. RADLER demonstrates a clear advantage over the RODNet models. As shown in every column, RADLER can produce a more accurate and confident detection for each

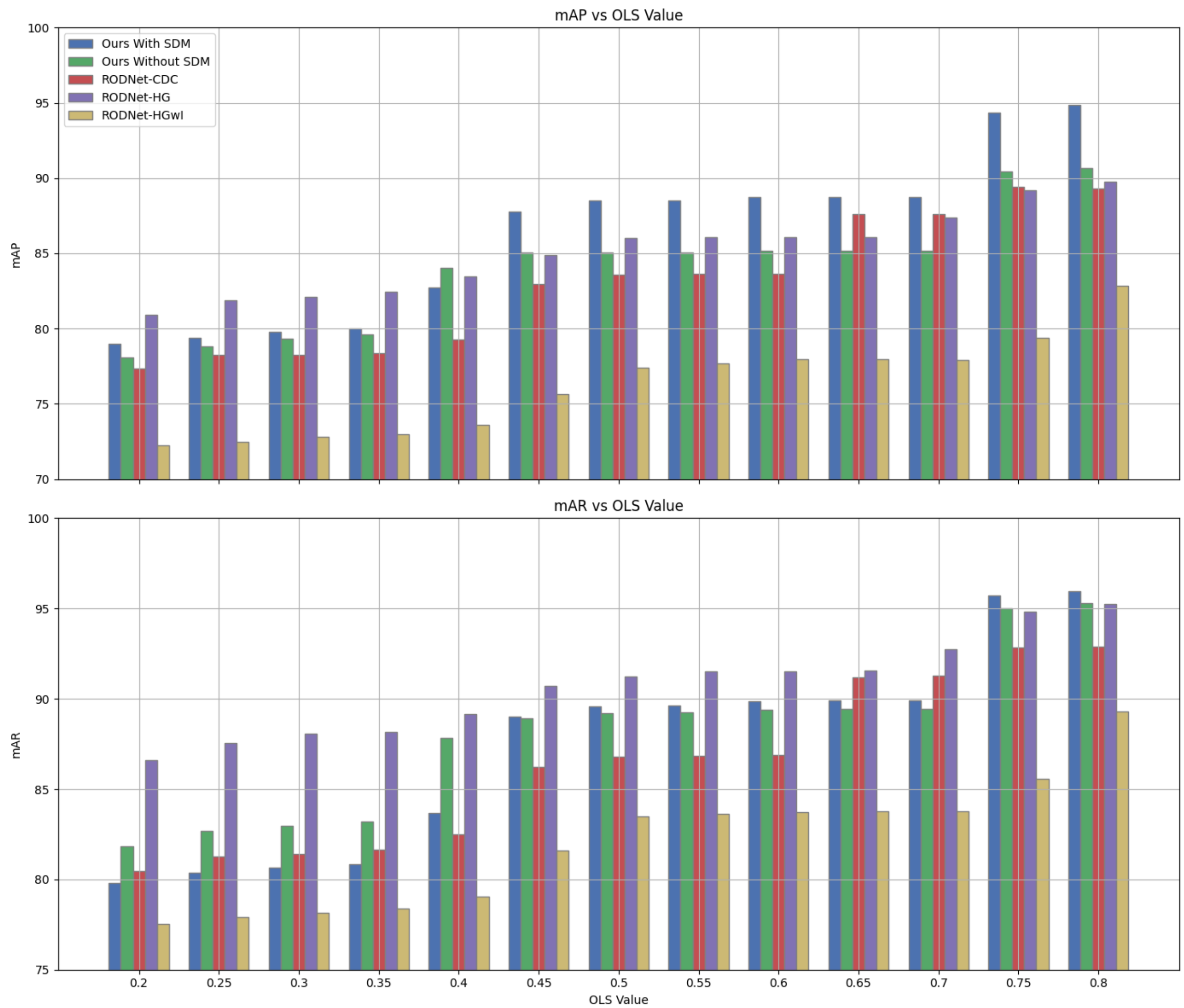

Figure 13. The mAP and mAR of RADLER and RODNet under different OLS values as the threshold for L-NMS.

object as the RODNet models tend to produce redundant detection for one object with lower confidence values.

In the 1st column, SDM contribute to the detection of the cyclist by increasing its confidence score by 0.09. Also, in the last column, the usage of SDM increases the confidence of the detected cyclist by 0.06.

However, SDM are not always contributive. In the 1st column, the detected car on the left in the middle of the street is 0.03 lower in confidence score compared to the detection from RADLER without SDM. Also, in the 3rd column, SDM have lowered the confidence of the detected cyclist by 0.03.

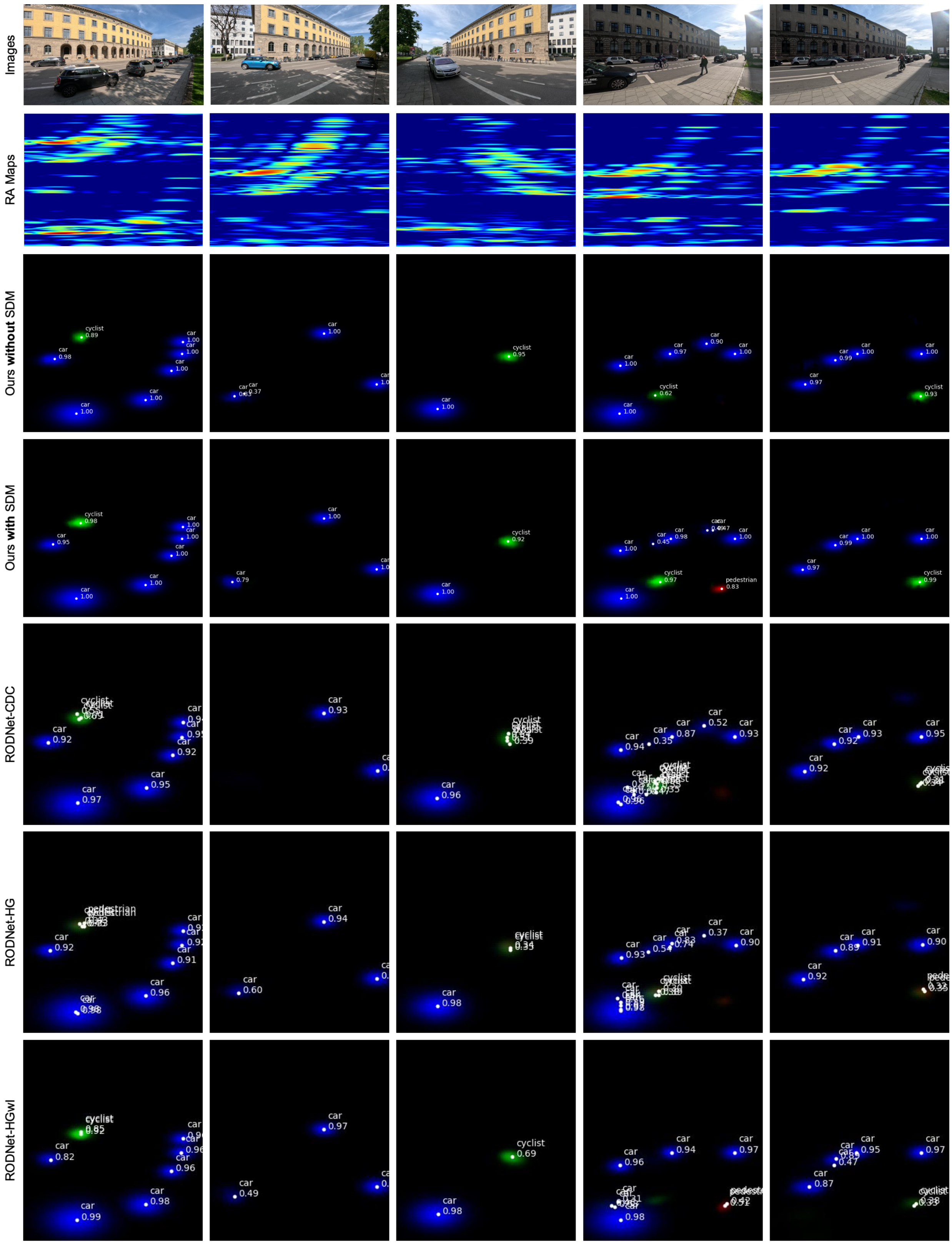

Figure 14. Example results: the first row displays the images, and the second row shows the corresponding RA maps. Subsequent rows display predicted ConfMaps from different models.