A Combined Multiple Object Detection and Tracking Network on FMCW Radar Data

Yutian Lei
yutianle@andrew.cmu.edu

Yuheng Qiu
yuhengq@andrew.cmu.edu

Haomin Shi
haomins@andrew.cmu.edu

Zilin Si
zsi@andrew.cmu.edu

Abstract

With the benefits of robustness on different environments and low cost comparing to the vision-based and LiDAR-based methods, object detection and tracking with radar data is a promising approach for perception tasks in autonomous driving. In this work, we propose a combined multiple object detection and tracking network on frequency-modulated continuous-wave (FMCW) radar data. The object detection and tracking framework consists of 1) a detection branch to generate confident maps for detecting multi-class objects from radar sequences, and 2) a tracking branch to realize object tracking. To incorporate spatial information into temporal features extracted by 3D encoder, a novel channel fusion attention (CFA) module is deployed upon the above architectures. Due to the absence of a public radar tracking dataset, we labeled the CRUW dataset, a camera-radar object detection dataset with a semi-automatic annotation to gain tracking ground truth. Finally, we show our proposed method can achieve a satisfying detection and tracking accuracy by giving experimental results and comparing to the performance of the baseline model.

1 Introduction

Object detection and tracking is essential to the safety of autonomous driving. Camera and LiDAR have long been used to perform this task, and there has been plenty of researches on this topic. However, both of them are sensitive to lighting and weather conditions, and LiDAR is too expensive for commercial autonomous vehicles. Alternatively, radar with its low cost and robust performance under different weather conditions has shown its value to autonomous driving. Therefore, object detection and tracking under outdoor environment with radar is a quite new trending topic worth exploring.

Most of existing methods on radar-based object tracking [**?**, **?**, **?**] take advantages of the Doppler information. But due to limited data transfer bandwidth, it is common to drop Doppler data to get more accurate range and azimuth measurements [**?**] in practice. Therefore, new methods that do not rely on the Doppler data need to be studied.

While a well-designed and trained neutral network nowadays has been able to achieve promising results object tracking and detection, there are few researches on exploiting DNNs for object tracking using radar signals.

In this paper, we proposed an one-stage end-to-end network which predicts object detection and tracking for the given radar data without additional tracklet association or re-identification to increase the robustness and accuracy for object detection and tracking with radar range-azimuth data, as shown

Preprint. Under review.

in Figure 1. The model is supervised by the detection ground truth provided from the CRUW dataset and the tracking ground truth generated by our proposed semi-automatic annotation technique.

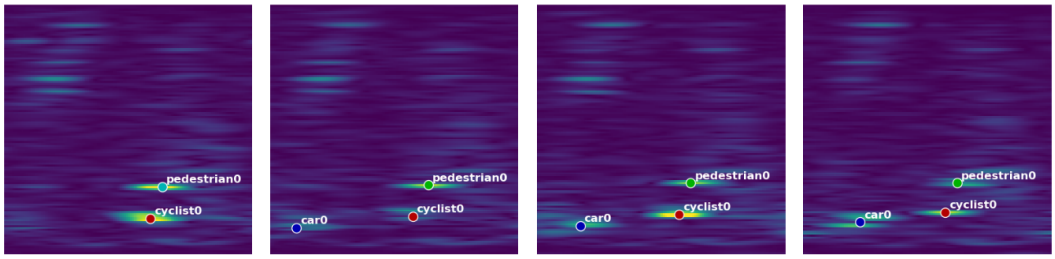

Figure 1: Illustration of object detection and tracking on range-azimuth Radar data. Confidence maps of detection as well as associated features are extracted from a sequence of RF images by an end-to-end deep neural network.

2 Related Work

Object Detection and Tracking: Image-based object detection methods [**?**, **?**] achieved real-time performances, which is fundamental and crucial for many robotics applications. In recent years, vision-based object tracking approaches have also made significant progress under deep learning frameworks. Object tracking models are typically following two main paradigms, tracking-by-detection [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**] and tracking-by-segmentation[**?**, **?**, **?**]. They generated bounding boxes (detection) or per-pixel segments (segmentation) for the objects of interests and then linked objects into trajectories via data association. [**?**] developed an end-to-end tracking architecture with a dedicated optimization process, which is capable of fully exploiting both target and background appearance information for target model prediction. FAMNet [**?**] refined feature extraction, affinity estimation and multi-dimensional assignment in a single network. [**?**] proposed a method to address multiple object tracking(MOT) by defining a dissimilarity measure based on object motion, appearance, structure, and size. As there is no bounding box definition in our RF images nor the resulting output ConfMaps, traditional MOT method is not applicable in our task. A new method should be designed for radar object detection and tracking task.

Learning of Radar Data: Some previous work [**?**, **?**] focused on deriving ego-motions or semantic environment by extracting features from radar point clouds. However, it sacrificed run-time efficiency that makes it not suitable for real time tasks. Apart from handcrafted feature-based methods, there are also learning-based radar object detection systems. Palffy et al. [**?**] proposed a CNN-based radar object detection pipeline, and achieved relatively high precision and recall with Range-Azimuth images and Doppler dimension. But it had one drawback that its data annotation was human hand labeled, which requires too much human effort on large dataset. Xu et al. [**?**] constructed an improved U-net with FMCW radar signal and included auto-labeling inputs, while most of the training data were still human labeled. Yizhou et al. [**?**] focused on radar object detection, and proposed a camera-radar fusion algorithm to generate confidence maps from networks. They designed an auto-encoder neural network to learn object locations and object classes. They also released an open source dataset including all camera images and FMCW radar data. The limitation of their work is that they cannot detect or distinguish close objects, and their pipeline is hard to generalize to other tasks such as tracking and motion planning. Daniel et al. [**?**] introduced another Doppler dimension of the radar data and used end-to-end CNN for detection and classification. However, this work only generates sparse points but not a full inference of object position. Finally, Bence et al. [**?**] utilized Range-Azimuth-Doppler data and constructed a LSTM model to generate bounding boxes with human labeled ground truth data. In our work, all of the data are camera labeled, which makes the task highly scalable and automatic. And we believe that it is crucial to the object detection task in self-driving which requires a huge amount of data.

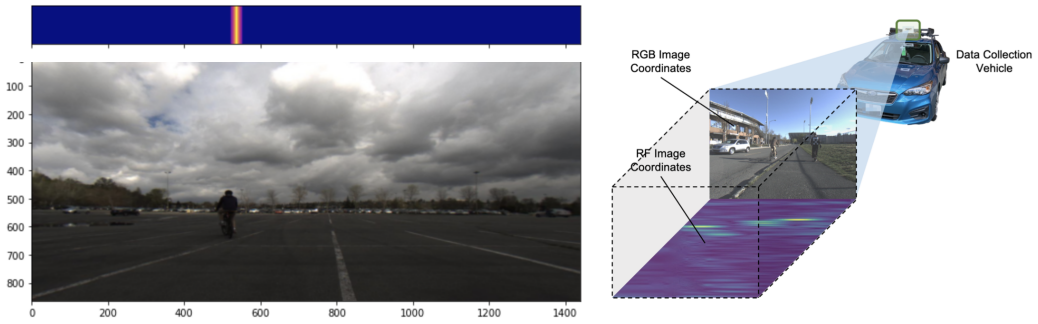

Figure 2: A projection from radar to image. We can re-projecte the x-axis of the object based on the pin-hole camera re-projection model

3 Dataset Description

CRUW Dataset: CRUW Dataset[1] is a public camera-radar dataset for autonomous driving. There are total 3.5 hours with 30 FPS of camera-radar data collected from parking lot, campus road, city street and highway scenarios. And about 260K objects are included in the dataset. Sensors on the data collection platform include a pair of stereo cameras (FLIR BFS-U3-16S2C-CS) and two 77GHz FMCW radar antenna arrays (TI AWR1843 + DCA1000) which are calibrated and synchronized. CRUW dataset provides object annotations for RF images which include object classes and centers.

Semi-automatic Annotation for Tracking Ground Truth: To incorporate object tracking with the radar sensor, we need to label the raw radar data with object categories and tracking ids. In order to keep the annotation effort manageable, a semi-automatic method is proposed to associate object detection into trajectories via tracklet association. Specifically, the image-based object detection is generated by Mask-RCNN [**?**], and then the detection results are fed into the SORT [**?**] tracking algorithm. Given the objects information from images, we can project the tracked object to the radar field. As shown in Figure 2, the radar-image projection is based on the pin-hole camera model. we can calculate the projection on the x-axis given the (1), where F_x is the focal length on x-axis, C_x is the center of the camera on x-axis, x_c and z_c are translation from the camera to radar sensor.

$$x = C_x + F_x \frac{\rho_c \sin\theta_c + x_c}{\rho_c \cos\theta_c + z_c} \tag{1}$$

Also, the labeled data can be re-projected from RF-image space to the image space so that the generated annotations can be human-corrected.

4 Model

Inspired by [**?**, **?**], our proposed method for the radar detection and tracking consists of six components as shown in Figure 3: (1) a 2D-CNN encoder backbone to extract detection features from a single key radar frame, (2) a 3D-CNN encoder backbone to extract tracking features from a radar clip; (3) a channel fusion and attention (CFA) module ;(4) Atrous Spatial Pyramid Pooling (ASPP) modules, (5) decoder modules, and (6) task-specific prediction heads for detection and tracking.

The 2D-CNN encoder extracts detection features which provides spatial information. The ASPP module extracts multi-scale features, which are crucial for detection tasks because the the objects locate in different ranges and have different sizes. Finally, the 2D decoder takes output features from ASPP and the skip connected features from encoder to generate final detection results.

Similarly, 3D encoder extracts tracking features which provides temporal-spatial information. The CFA module introduced by [**?**] combines channels from both 2D and 3D features. We adopt this module because tracking results are highly dependent on the detection performance and features. When combining both spatial and temporal-spatial features, the system is able to generate temporal

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

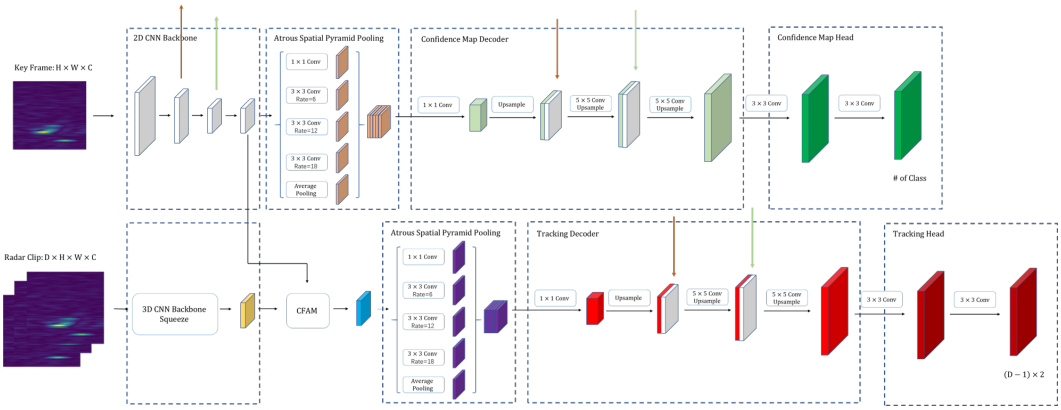

Figure 3: The Structure of Our Proposed Model

action offsets. Finally, the 3D decoder takes the output features from CFA module and the skip connected features from 2D encoder to produce tracking offsets.

2D-CNN Encoder Backbone We deploy ResNet-18 as the basic architecture in our 2D-CNN branch to extract detection features. The input of the this module is the key frame with the shape $[H \times W \times C_i]$, which is the most recent frame of the input radar clip. The last feature F_3 of 2D backbone, with the shape of $[\frac{H}{16} \times \frac{W}{16} \times C_o]$ is fed in to the ASPP module and the CFA module, and the intermediate features F_1 and F_2, with the shape of $[\frac{H}{4} \times \frac{W}{4} \times C_1]$ and $[\frac{H}{8} \times \frac{W}{8} \times C_2]$ respectively, are saved, which will be fed into the detection and tracking decoders later.

3D-CNN Encoder Backbone As 3D-CNNs are able to capture motion information by applying convolution operation not only in space dimension but also in time dimension, the 3D-CNN is utilized to extract spatio-temporal features used for tracking. The basic 3D-CNN architecture in our framework is 3DResNet-18. The input to the 3D backbone is a clip of the radar data, which is composed of a sequence of D frames in time order, and has a shape of $[D \times H \times W \times C_i]$, while the last conv layer of 3D backbone outputs a feature map of shape $[1 \times \frac{H}{16} \times \frac{W}{16} \times C_o]$. The depth dimension of the output feature map is reduced to 1 such that output features can be squeezed to $[\frac{H}{16} \times \frac{W}{16} \times C_o]$ in order to match the output feature of 2D-CNN.

Channel Fusion and Attention (CFA) Module The CFA module aggregates features from 3D and 2D encoders by taking both of their outputs as input. The output generated from the 3D network's has a size of $[\frac{H}{16} \times \frac{W}{16} \times C_o]$, and the output generated from the 2D network has a size of $[\frac{H}{16} \times \frac{W}{16} \times C_2]$. And they get concatenated to form a input of $[\frac{H}{16} \times \frac{W}{16} \times (C_0 + C_2)]$, which is later fed into the CFA module. The output of the CFA module is $[\frac{H}{16} \times \frac{W}{16} \times C_3]$, which smoothly combines the features from both spatial and spatial-temporal information.

Atrous Spatial Pyramid Pooling (ASPP) Module DeepLabv2 [**?**] proposed atrous spatial pyramid pooling (ASPP), where parallel atrous convolution layers with different rates capture multi-scale information. The ASPP is adpodted in our model to probe the input feature with multiple filters that have complementary effective fields of view, thus capturing 2D detection features as well as 2D-3D fused features at multiple scales.

The adopted ASPP module consists of one 1×1 convolution, three 3×3 convolutions with rates = $(6, 12, 18)$ and an average pooling layer. The resulting features from all five branches are then concatenated and passed to the corresponding decoder.

DeepLabV3+ Decoder The decoder module follows DeepLabV3+ [**?**] with small modification for radar data. The features from ASPP are first passed through 1×1 convolution and then bilinearly upsampled by a factor of 2. The output features are then concatenated with the corresponding low-level features F_2 from the 2D-CNN backbone that have the same spatial resolution. It's noticed that another 1×1 convolution on the low-level features is used to reduce the number of channels, since

the corresponding low level features usually contain a large number of channels which may outweigh the importance of the rich encoder features. After the concatenation, another 5×5 convolution is applied to refine the features followed by another simple bilinear upsampling by a factor of 2, and similarly, the output feature is then concatenated with the corresponding low-level feature F_1. Finally, the output feature is passed through a 3×3 convolution and a bilinearly upsampling layer by a factor of 4. The final output feature is then fed into the corresponding header.

Head The head module simply consists of two 3×3 convolution layers to get the detection and tracking outputs.

For detection branch, the head outputs a $\hat{Y} \in [0,1]^{[H \times W \times C_d]}$ confidence map, where C_d is the number of classes. As each of the ground truth center $p = [x_p, y_p]$ is spread onto a confidence map $Y \in [0,1]^{H \times w \times C_d}$ using a Gaussian kernel $Y = exp(-\frac{(x-x_p)^2+(y-y_p)^2}{2\sigma_p^2})$, where σ_p is an object size-adaptive standard deviation [**?**]. If two Gaussians of the same class overlap, we take the element-wise maximum. The training objective is a penalty-reduced pixel-wise L1 Loss.

$$L_d = \sum_{xyc} (\hat{Y}_{xyc} - Y_{xyc})^2 \tag{2}$$

For tracking branch, the head outputs an offset of shape $\hat{O} \in [H \times W \times 2(D-1)]$. The offset indicates the center of each object in current time frame to the center of the object with the same track id in the previous $D-1$ time frames. Thus, the association between the objects in different time frames can be built on the offsets of its center. We also use L1 loss for the offset prediction, which is only activated at pixels neighboring to the center of each object.

$$L_o = \begin{cases} \sum_{xyc} (\hat{O}_{xyc} - O_{xyc})^2, & \text{if } \exists i, (x - x_{pi})^2 + (y - y_{pi})^2 \leq \eta \\ 0, & \text{otherwises} \end{cases} \tag{3}$$

So that the totol loss of the model will be

$$L = \alpha L_d + \beta L_o \tag{4}$$

where α, β are weights to balance the relative value of tracking and detection losses.

5 Evaluation Metrics

The target of our model is to get the location, class information and tracking identity for each object in the input radar frames. Thus, both detection and tracking metrics should be established for radar data to evaluate our model.

Detection For traditional visual object detection tasks, the bounding boxes' location and size are the main criterion to evaluate accuracy. However, for both the ConfMaps and RF images, there is no bounding box definition but the probability distribution in the map. Therefore, a new metric, Object Location Similarity (OLS) is used to define the correlation between two detected objects in terms of distance and the scale in the ConfMaps. Formally the OLS is defined as

$$OLS = exp\{\frac{-d^2}{2(s\kappa_{cls})^2}\} \tag{5}$$

where d is the distance between two points in an RF image, s is the distance for an object to the radar sensor to include the scale information. Closer the object is to the radar sensor, larger the scale is. And κ_{cls} is the error tolerance for each object class. Here OLS is interpreted as a Gaussian distribution, where d serves as the bias, and $(s\kappa_{cls})^2$ serves as the variance.

During the evaluation, the OLS is firstly calculated between each detection result and ground truth in every frame. Then different thresholds from 0.5 to 0.9 with a step of 0.05 for OLS are applied to calculate the average precision (AP) and average recall (AR).

Tracking The well-established CLEAR MOT metrics [**?**, **?**] is adapted for multi-object tracking to our task with modification for radar data. Formally, the ground truth of a sequence radar data with T time frames. Each frame consists of a set of N non-empty confidence probability maps $P = p_1, ..., p_N$ where $p_i \in [0, 1]^{[h \times w]}$ and N be the number of object classes. Each object t has a track id $id_t \in \mathbb{Z}^+$. The output of a MOTS method is a set of K non-empty hypothesis probability $H = h_1, ..., h_K$ with $h_i \in [0, 1]^{[h \times w]}$, where track id $id_h \in \mathbb{Z}^+$ is assigned by our model.

Hence, the mapping $c : H \rightarrow P$ from hypothesis to ground truth can simply be defined using the the proposed object location similarity (OLS) as

$$c(h) = \begin{cases} \arg\max_{p \in P} OLS(h, p), & if \max_{p \in P} OLS(h, p) > 0.5 \\ \varnothing, & otherwise \end{cases} \tag{6}$$

The set of true positives $TP = \{h \in H \mid c(h) \neq \varnothing\}$ is comprised of hypothesis probability which are mapped to ground truth. And false positives are hypothesis probability that are not mapped to any ground truth mask, $FP = \{h \in H \mid c(h) = \varnothing\}$ The set of false negatives $FN = \{p \in P \mid c^{-1}(m) = \varnothing\}$ contains the ground truth which are not covered by any hypothesis.

Then, let $pred : P \rightarrow P \cup \varnothing$ denote the latest tracked predecessor of a ground truth, or $\varnothing$ if no tracked predecessor exists. The set of identity switches (IDS) is then defined as the set of ground truth masks whose predecessor was tracked with a different id, by

$$IDS = \{p \in P \mid c^{-1}(p) \neq \varnothing \wedge pred(m) \neq \varnothing \wedge id_{c^{-1}(m)} \neq id_{c^{-1}(pred(m))}\} \tag{7}$$

Given the definitions in CLEAR MOT metrics [**?**], we propose the multi-object tracking accuracy of radar (MOTAR) based on object location similarity as

$$MOTAR = 1 - \frac{|FN| + |FP| + |IDS|}{|P|} = \frac{|TP| - |FP| - |IDS|}{|P|} \tag{8}$$

and the multi object tracking precision of radar (MOTPR) as

$$MOTPR = \frac{\sum_{h \in TP} OLS(h, c(h))}{|TP|} \tag{9}$$

where $|S|$ is the number of element in set S.

6 Experiments and Results

In this section, we will first present the detection performance of our baseline model RODNet, then show and analyze the quantitative results of our ablation study on different modules of our model for both detection and tracking tasks. Finally, the qualitative results of our model will be shown.

6.1 Baseline Performance

We choose the RODNet [**?**] as our baseline model as it is the state-of-the-art work on radar object detection. However, as the size and the annotation format of the dataset used in the baseline was different from the released CRUW dataset, which is used in this project, it's necessary for us to implement and train the baseline model on our dataset to make our detection result comparable. For the baseline model, we use its open-sourced implementation RODNet based on the repository[2]. We trained the model with hourglass architecture which gives the best performance in the paper[**?**]. All the other hyper-parameters are kept same as descried in the original paper.

The quantitative detection results of the baseline trained on CRUW datset is shown in Table 1. It's noted that the AP and AR are both worse than the results from paper where they were reported as AP = 85.98 and AR = 87.86. The gap of the detection accuracy, as far as we concerned, is mainly resulted from the size of training dataset, as the dataset released from the official website (with 40 video sequences) is a more than ten times smaller than the dataset described from the paper (464

[2]https://github.com/yizhou-wang/RODNet

video sequences). Furthermore, our performance is comparable with the results on the leaderboard of the competition held by the authors, ROD2021 Challenge @ ICMR 2021[3], which further makes our results reasonable.

6.2 Quantitative Performance of Detection

Table 1: Ablation study on Object Detection.

Backbone	ASPP	Skip	AP	$AP^{0.5}$	$AP^{0.7}$	$AP^{0.9}$	AR	$AR^{0.5}$	$AR^{0.7}$	$AR^{0.9}$
Resnet-18			65.31	76.53	64.28	58.18	74.41	79.48	86.81	65.32
	✓		72.25	81.79	74.93	64.24	78.86	81.33	79.15	69.99
		✓	73.41	82.93	72.88	67.21	80.11	84.74	82.56	73.21
	✓	✓	**78.13**	**84.63**	**80.32**	**71.69**	**86.39**	**88.97**	**86.22**	**79.14**
Baseline: RodNet			66.24	75.23	65.12	60.11	73.72	80.66	84.89	67.05

We provide our quantitative results as well as the ablation study results for detection task on CRUW dataset compared with RODNet, which are shown in Table 1. We can see that our AP and AR are higher than the baseline with ASPP and skip connection modules.

Effectiveness of ASPP Module To prove the effectiveness of the ASPP module, we did experiments on substituting the ASPP module with a layer that repeats and stacks the 2D features to make it same shape of the output feature of ASPP. The average precision improved by 7% and average recall improved by 4% with the ASPP module, which proves that getting multi-scale features is essential for the network to detect objects correctly and distinguish false positives.

Effectiveness of Skip Connection from 2D Encoder to 2D Decoder In the proposed model, the intermediate features of the 2D encoder is skip connected to the 2D decoder as commonly applied in the image encoder-decoder framework like [**?**, **?**]. We also trained model without skip connection structure to illustrate the effectiveness of this structure. It's found that average recall was 6% higher and average precision was 8% higher with skip connection. This showed that skip connection recovers spatial information lost during downsampling, and combining compressed latent features with early features for a more compact model greatly improves detection performance.

6.3 Quantitative Performance of Tracking

Table 2: Ablation study on Object Tracking.

Backbone	CAF	Skip	$MOTAR$	$MOTPR$
Resnet-18			60.93	62.35
	✓		80.38	79.01
		✓	65.13	68.92
	✓	✓	**86.22**	**89.15**

We provide our quantitative results as well as the ablation study results for tracking task on CRUW dataset, which are shown in Table 2.

Effectiveness of CFA Module To check the effectiveness of the CFA module, we trained our model without CFA module by directly passing the output feature of 3D CNN encoder (repeated and stacked so that it has same channel with of output of CFA module) to the ASPP module. It's demonstrated that the MOTAR and MOTPR significantly increased (around 20%) with CFA module added. The insight behind the experiment is straightforward: the tracking performance is highly depended on the the detection results of the key frames. That's why incorporating the detection feature into the tracking branch can help the network better predict and generate the tracking results, and thus improve the tracking performance of the network.

[3]https://competitions.codalab.org/competitions/28019

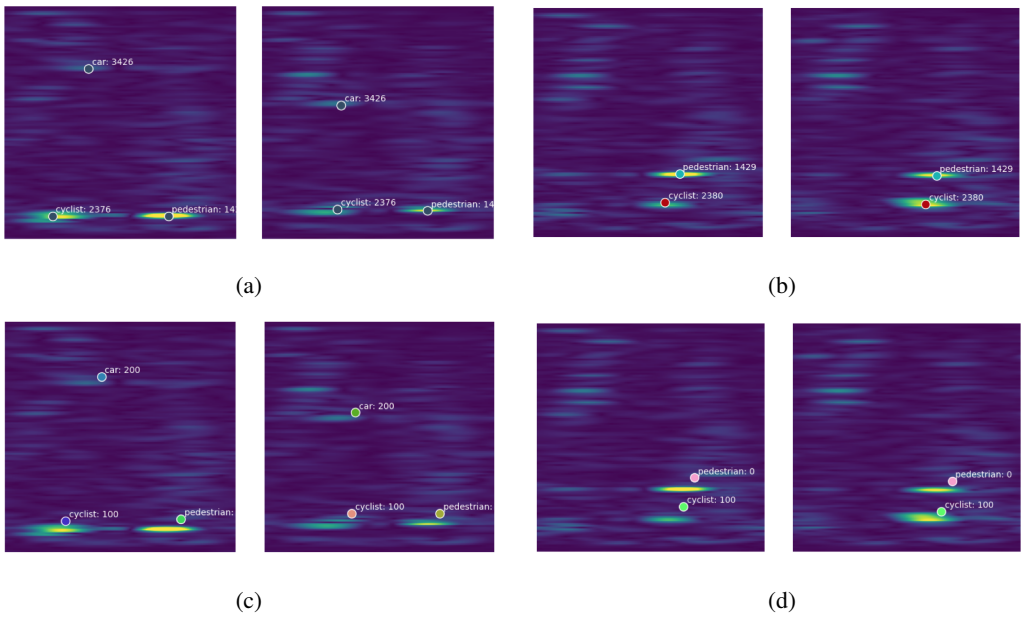

Figure 4: Detection and tracking results (c)(d) from two sequences which are compared with ground truth (a)(b). The object classes and tracking ids are correctly estimated.

Effectiveness of Skip Connection from 2D Encoder to 3D Decoder As motioned before, we also skip connected the features from 2D encoder to 3D decoder to further incorporate and recover the spatial dimensionality in the trakcing branch. To check the effectiveness of the CFA module, we trained our model without this structure by directly replicating and concatenating the output feature in tracking decoder. The experiments demonstrated that the MOTAR and MOTPR will increase (around 5-10%) with this structure, which further indicate the significance of detection (or spatial) feature in the learning of tracking results.

6.4 Qualitative Results

We have visualized our results, including Confmaps with detection and tracking labels on it as show in Figure 4. We can see that in most cases, even with 3 objects, the model generates satisfactory results. In the results, we can see the model can successfully detect, classify and track objects compared with the ground truth.

7 Conclusion and Discussion

In this project, we first designed and implemented a semi-automatic labeling technique to generate ground truth data for object tracking. And we proposed a novel network that detects and tracks objects in radar data simultaneously. Our experiments demonstrates that our detection performance is much better than the re-trained baseline model. Through the ablation study, we found out that: ASPP module and skip connections are crucial for object detection; CFA model helps generating features that combines temporal and temporal-spatial features. Our final tracking inference reaches 12 FPS in average with satisfactory results on detection and tracking qualitatively and quantitatively.

8 Future Work

Efficiency The efficiency mostly reflects on the real-time achievement. However, from our experiments, our model currently can achieve around 12 FPS, which doesn't fulfill real-time requirements. Also, with the number of objects getting larger, the post-processing time increases quadratically. So, we would like to lighten our model but still maintain the same level accuracy to improve its efficiency.

Multi-modal system Generally for a complete self-driving system, it cannot just depend on a single sensor input. Multi-sensor fusion can greatly improve the system's robustness. So we would like to extend our work to combine with other sensor data such as vision or LiDAR data. Possible fusion methods include loose coupling, which means doing probabilistic inference on all tracking results, and tight coupling, which means feeding data into a single network.

Comparison with traditional methods In this project, we have not done comparison with other traditional radar object detection methods. In the future, we plan to do so.

Further ablation studies In addition to ResNet-18, we plan to experiment with different 2D-CNN backbones in our ablation study to get more comprehensive results on what kind of network produces the best result.

9 Division of Work

Yutian Lei: Model implementation and training

Yuheng Qiu: Ground truth generation and semi-automatic labeling

Zilin Si: Baseline, Pre-processing and post-processing of data

Haomin Shi: Baseline, Dataloading and evaluation metrics

10 Github Link

Here is our private link: Private Link. Please send email to haomins@andrew.cmu.edu to get view access.