TU Delft

Delft University of Technology

Track-Cued Radar Point Cloud Target Classification

Chen, Lihui; Hassan, Mujtaba; Ravindran, Satish; Wu, Ryan

DOI
10.1109/IEEECONF60004.2024.10942901

Publication date
2024

Document Version
Final published version

Published in
Conference Record of the 58th Asilomar Conference on Signals, Systems and Computers, ACSSC 2024

Citation (APA)
Chen, L., Hassan, M., Ravindran, S., & Wu, R. (2024). Track-Cued Radar Point Cloud Target Classification. In M. B. Matthews (Ed.), *Conference Record of the 58th Asilomar Conference on Signals, Systems and Computers, ACSSC 2024* (pp. 1354-1359). (Conference Record - Asilomar Conference on Signals, Systems and Computers). IEEE. https://doi.org/10.1109/IEEECONF60004.2024.10942901

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

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

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.

Track-Cued Radar Point Cloud Target Classification

Lihui Chen
Advanced Radar Solutions
NXP Semiconductors
San Jose, USA
phil.chen_1@nxp.com

Mujtaba Hassan
MS3 Group, EEMCS Faculty
Delft University of Technology
Delft, Netherlands
s.m.hassan@tudelft.nl

Satish Ravindran
Advanced Radar Solutions
NXP Semiconductors
San Jose, USA
satish.ravindran@nxp.com

Ryan Wu
Advanced Radar Solutions
NXP Semiconductors
San Jose, USA
ryan.wu@nxp.com

2024 58th Asilomar Conference on Signals, Systems, and Computers | 979-8-3503-5405-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/IEEECONF60004.2024.10942901

***Abstract*—A novel temporal-spatial object classification neural network model is proposed to improve the classification capability of tracked objects. It takes queued points of tracked objects using multiple frames as input, utilizes spatial and temporal information from these points for sampling and grouping as well as extracts hierarchical temporal-spatial features for target classification. Experimental results on a proprietary 4D Imaging Radar dataset and open-source 2D RadarScenes dataset demonstrate that the proposed tracker-cued radar point-cloud target classification method allows the model to learn meaningful appearance and motion features from sparse radar points data, and achieves accurate classification output as compared to a baseline method, while being efficient to run on edge hardware with limited resources.**

***Index Terms*—Object classification, temporal-spatial sampling, temporal-spatial grouping, temporal-spatial feature extraction.**

I. Introduction

Automotive radar has been extensively utilized in cars for many years as an essential sensor, primarily due to its robustness in harsh weather conditions, its capacity to measure Doppler information in the surrounding environment, and its cost-effectiveness. Recently, developments in radar technologies, coupled with the availability of open-source radar datasets, have attracted increased attention to radar for perception tasks in autonomous driving using deep learning.

Point cloud-based radar target tracking has been implemented in many radar products [1], which can provide target ID, position and velocity information at target level (represented as 2D bounding boxes or 3D bounding cubes). Classifying the tracked targets is a vital component of this perception stack. It allows a higher degree of scene understanding by allowing the system to recognize objects present in the scene as well as to provide essential information for downstream tasks such as sensor fusion and motion control including several deep learning applications [2].

There are existing techniques to classify objects based on point clouds [3], [4], [5], [6], [7], [8]. However, these techniques usually have certain limitations including:

- The sparsity of radar points presents challenges to classify small targets, especially for vulnerable road users such as pedestrians, bicyclists, and motorcyclists. Radar points on vulnerable road users could degenerate into single-point case, which cause most of the models to fail.
- Classification result would be sensitive to the segmentation quality in a single frame. When the segmentation quality drops, or tracking is inaccurate because of no detection in a specific frame, inconsistent classification results will be generated, even though the object is still being tracked.
- Computation cost of these techniques are very high, which makes these methods unsuitable to be implemented on edge processors.

In this paper, we propose a novel solution which overcomes the above mentioned limitations. We named our solution Tracker-cued Point-cloud Target classification. Fig. 1 shows a general pipeline of our proposed method. Our approach is based on the intuition that combining spatial and temporal information of points within a tracked object, should lead to more robust classification output than using the information from a single frame. The main components of our method include:

- Point set queue, which contains both temporal and spatial information of radar points.
- Hierarchical point set feature extraction achieved by cascading multiple temporal-spatial point set feature extraction modules.
- Temporal-spatial point set feature extraction modules that comprise of a temporal-spatial sampling, a temporal-spatial grouping, and a temporal-spatial feature extractor.

We evaluated our proposed classification methods on a proprietary 4D imaging radar dataset from one of our collaborators and the open-source RadarScenes dataset [9]. Our proposed solution outperforms the baseline single frame classifier [4] by providing an increase in classification accuracy of 8.6% for a set of 5 classes, and 5.3% for the combined VRU/Vehicle classes on the Imaging Radar dataset as well as an increase in classification accuracy of 8.5% for the set of 5 combined classes on RadarScenes dataset. The main contributions of the paper are as follows: (1) We designed a classification approach that uses multiple frames for classification of tracked targets which showed a significant improvement in classification performance as compared to a baseline approach [4] using single frame. (2) We proposed a novel temporal sampling and grouping scheme that helps to extract diverse and representative points coming from different spatial and temporal locations as well as used a temporal-spatial feature extractor to obtain hierarchical temporal-spatial features from these points (3) We developed a classification NN that requires

Authorized licensed use limited to: TU Delft Library. Downloaded on May 07,2025 at 09:09:52 UTC from IEEE Xplore. Restrictions apply.

low memory and compute that is suitable for implementation on real embedded automotive hardware.

The rest of this paper is organized as follows: Section II illustrates the approach used in this work. Section III describes the various experiments performed with a discussion on results. Section IV concludes this work.

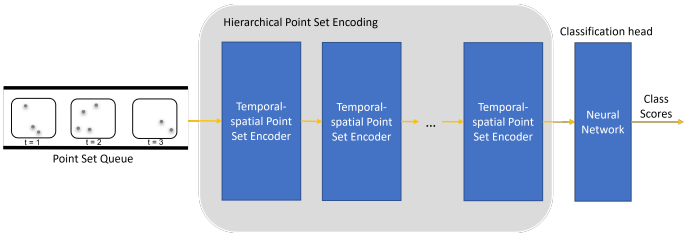


Figure 1: Overview of our model architecture.

II. Proposed Method

Our model is inspired from PointNet++ [4] which uses a hierarchical feature extraction NN module for classification of single frame lidar point cloud. However, we have several key differences from this model which include: (1). Usage of multiple frames of radar point clouds instead of single frame lidar point cloud to account for the sparsity in radar point clouds as well as to leverage additional radial cross section (RCS) and Doppler measurement (2). Point set queue, which contains both temporal and spatial information of radar points. (3). Temporal-spatial point set feature extraction modules comprising of a temporal-spatial sampling, a temporal-spatial grouping, and a hierarchical temporal-spatial feature extraction.

A. Point Set Queue

The input to the our proposed model is a First-In-First-Out (FIFO) queue that queues the points of a tracked object from multiple frames. Fig. 2 shows the general concept. Points that lie inside the bounding box of the tracked object are stored for each timestamp for a total of T frames. Each point contain six features: x, y, z, $Doppler$, RCS, $timestamp$. Depending on the scenario, each member of the queue can have different number of points. The purposes of the queue design are:

- Accumulating more points for classification even if object points are limited within one or all frames.
- Improving classification for extreme cases including zero-point case where tracking is off (Fig. 3);
- Introducing relative timestamps to incorporate temporal information for points from different frames. The temporal-spatial point set feature extraction module can extract motion features using such timestamps.

B. Temporal-Spatial Point Set Feature Extraction

The points in the queue are then passed to the temporal-spatial point set feature extraction module. This module is essentially an encoder that generate temporal-spatial features which are more reliable and accurate than spatial features obtained using spatial encoding alone. Fig. 1 shows the

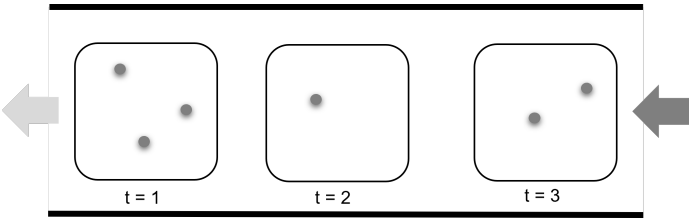


Figure 2: A FIFO Point-set Queue with Timestamps.

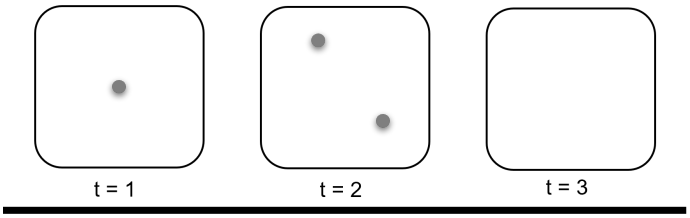


Figure 3: Case of zero-point when tracking is off.

placement of this module in the overall setup. Here, the hierarchical temporal-spatial encoder includes multiple individual temporal-spatial encoder stages cascaded in series. The purpose of using this cascading approach is to obtain encoder modules optimized for different characteristics. For example, the first temporal-spatial encoder stage operate on a first spatial scale (receptive field) optimized for detecting smaller features, e.g., by applying a smaller grouping radius, whereas the second temporal-spatial encoder stage operate on a second spatial scale optimized for detecting larger features, e.g., by applying a larger grouping radius. Numbers of samples and the balance between spatial and temporal offsets is varied across different encoder stages.

A more detailed view of each temporal-spatial encoder stage is shown in Fig. 4. Here, the input is the point set queue or abstracted point features obtained from the preceding encoder module. This encoder module includes a temporal-spatial sampling component, a temporal-spatial grouping component, and a NN component which extract hierarchical temporal-spatial features that are passed to the next stage of the module.

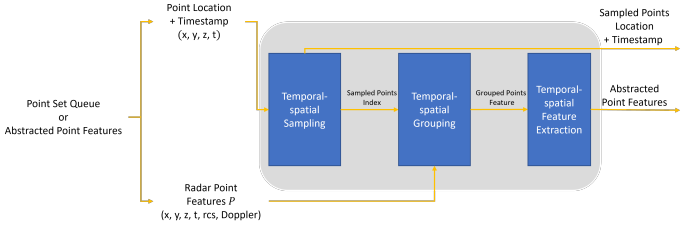


Figure 4: Temporal-spatial Point Set Feature Extraction.

C. Temporal-Spatial Sampling

Similar to [4], a sampling module is used to reduce the number of points for further processing. In the case of single frame point cloud, a farthest-point-sampling (FPS) approach is often used. The approach calculates the Euclidean distance from the current selected point to all other points, then the point with farthest distance is selected (sampled) as the next starting

Authorized licensed use limited to: TU Delft Library. Downloaded on May 07,2025 at 09:09:52 UTC from IEEE Xplore. Restrictions apply.

point. The goal of such sampling design is to preserve the appearance information of the object during down-sampling using only spatial coordinates. However, for our case, we are storing points coming from multiple timestamps. Since the object can be moving, using a sampling approach based on only spatial locations may not be the best method.

We propose a temporal-spatial sampling module that is designed to not only preserve the appearance, but also extract potential object motion. In order to obtain the samples, we modified the sampling distance by incorporating the timestamp (t). The distance is defined as a function of (x, y, z, t) as in Equation 1.

$$resizebox3.0in0.13ind_s(x, y, z, t) = \sqrt{(x_s - x_i)^2 + (y_s - y_i)^2 + (z_s - z_i)^2 + \lambda(t_s - t_i)^2} \quad (1)$$

where $\min(0, \Delta T_{\max}) \leq (t_s - t_i) \leq \max(0, \Delta T_{\max})$

Here, $d_s(x, y, z, t)$ is the Euclidean distance to a point s from the current sample, i which initially is the first sample. The calculated distance is the square root of the sum of the squares of offset components between points s and i in each of the dimensions x, y, z, and t. Time values are in units of relative timestamps, An adjustable parameter λ sets a balance between spatial components (x, y, z) and the temporal component (t). For example, setting λ to a large value ensures that the time component $\lambda(t_s - t_i)^2$ predominates over the spatial components, such that the next sample is certain to be selected from a different frame, assuming a timing constraint is satisfied. The timing constraint limits points that are eligible for selection as the next sample. If ΔT_{max} is 1, for example, then only points in the current frame or then immediately next time-adjacent frame are candidates for selection as the next sample. However, if ΔT_{max} is 0, then only points in the current frame can be selected. Points outside the range specified by ΔT_{max} are ignored, with no distances calculated for them, thus reducing the computational workload of the system. Once a point is selected as a sample, that point is removed from consideration from FPS operation going forward, such that each point can be sampled only once.

Fig. 5 shows an example arrangement for sampling radar points where ΔT_{max} is 1. Here, the sample at ΔT=1 (shown by green arrow) is selected since it has the highest distance from the current sample and is within the timing constraint. However, samples at ΔT=0 and ΔT=2 are not selected since the former has a higher distance than sample at ΔT=1 and the latter is outside the timing constraint.

D. Temporal-Spatial Grouping

As proposed in [4], a local grouping is typically used to exploit the local appearance structure of the point cloud by grouping each of the sampled point with its neighbors. A radius parameter "r" is used to defined the range of neighborhood point search. To achieve a similar goal, but with the context of temporal points, we scale the spatial radius r by the timestamp difference ΔT. Using this approach, points that are both spatially and temporally local to each other are prioritized to get grouped together.

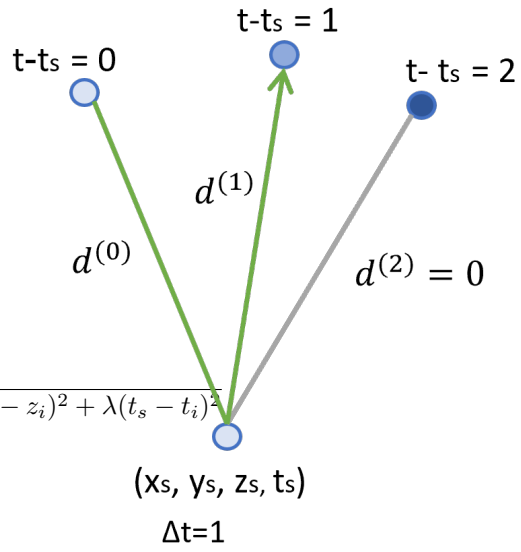


Figure 5: Example of temporal-spatial sampling encouraging temporal jumps.

Fig. 6 depicts grouping around a current sample point and illustrates how the scaled radius affect grouping of neighborhood points with different timestamp. Here, radar points are represented in space (x, y, z) and points from different frames are shown with different shading. Open circles represent points in the same frame as sample, hatched circles represent points in the immediately next frame (ΔT=+1), and solid circles represent points in the second frame (ΔT=+2). Just as the sampling is subject to timing constraints, so too is the grouping. Timing constraints applied when grouping around a current sample point are the same as those applied when sampling from the same sample point using FPS. For example, the current sample point may be associated with a ΔT_{max}=+1, because +1 was the ΔT_{max} applied when sampling from the same sample point. Thus, grouping around sample for this example is limited to the same frame that contains sample and to the immediately next frame in time. Any points shown with solid circles (ΔT_{max}=+2) are ignored for purposes of grouping. Grouping proceeds by applying different spatial radii for different frames. A first radius R_{local} is applied to points in the same frame as sample and a second radius R_{adj} is applied to points in the immediately time-adjacent frame (ΔT_{max}=+1). The grouping then groups together all points in the same frame as of the sample within the radius R_{local}, along with all points in the next frame within the radius R_{adj}. In general, the farther away a point is in time, the smaller the radius that is used for determining whether to include that point in the group. Mathematically , the grouping radius $R_{\Delta t}$ for different values of ΔT_{max} may be expressed as (2):

$$R_{\Delta T} = (\lambda_g)^{\Delta T} \cdot R_{local} \quad (2)$$

where R_{local} is the grouping radius for ΔT=0 (same frame) and λ_g is a scale factor. Both R_{local} and λ_g are adjustable parameters. For example, different values of R_{local} and/or λ_g

Authorized licensed use limited to: TU Delft Library. Downloaded on May 07,2025 at 09:09:52 UTC from IEEE Xplore. Restrictions apply.

are used by different encoder stages. Setting λ_g between 0 and 1 ensures that the grouping radius $R_{\Delta T}$ becomes smaller for larger values of ΔT.

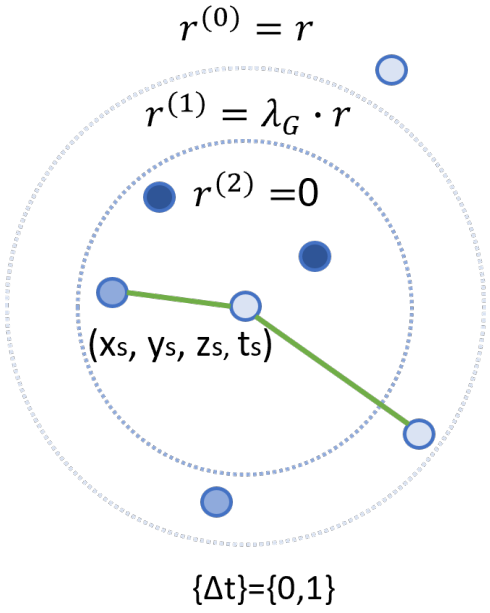


Figure 6: Temporal-spatial Grouping by scaling grouping radius.

E. Temporal-Spatial Feature Extraction

These temporal-spatial sampled points with their grouped neighbors are passed to a NN module. This consists of a series of a shared multi layer perceptron (MLP) layers to extract temporal-spatial features for each sampled point. By leveraging radar points across multiple frames, the temporal-spatial encoder is able to to build denser and more robust representations of features in the point cloud than could be achieved by limiting sampling and grouping to individual frames. These features at the current scale and are then passed to the next cascaded temporal-spatial feature extraction module to combine with the features extracted at next scale to obtain hierarchical features.

F. Classification Head

Finally, the hierarchical temporal-spatial features are passed to the classification head which is configured to provide output in the form of class labels. The classification head is implemented using a series of MLP layers followed by a softmax layer to give the probability of each label. The object is given the label of the class with the highest probability.

III. Experimental Data and Results

A. Dataset

In order to assess the performance of our classifier, a proprietary 4D Imaging Radar dataset is used. The 3D bounding box annotations with corresponding class labels and track IDs are available for each object in the scene. This dataset was used because it provides height information and has a high angular resolution that helps in obtaining a higher classification performance. Labels include car, truck, pedestrian, cyclist and motorcyclist.

To further validate our method, open-source 2D RadarScenes dataset is also used which provides ground truth annotation for moving objects with class labels and track IDs. This dataset was used because it is a popular dataset used in many automotive radar based research papers. Labels include car, pedestrian, pedestrian group, two-wheeler and truck.

B. Metrics

The performance of the tracked object classification can be measured by classification accuracy defined by (3). In the context of radar for ADAS/AD applications, for quantitative evaluation, the following accuracy metrics are measured:

- **Multi-class accuracy**: This is a single overall metric which provide the classification accuracy between all classes of interest (car, truck, pedestrian, cyclist, motorcyclist for the Imaging Radar dataset as well as car, pedestrian, pedestrian group, two-wheeler, truck for RadarScenes dataset).
- **VRU/Vehicle accuracy**: The above mentioned classes can be categorize into two groups: vulnerable road users (VRUs) and vehicles. This metric measures classification accuracy for these two merged classes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where TP represents true positive, TN true negative, FP false positive and FN false negative respectively.

C. Results

The performance and computation cost of the proposed methodology comparing to the baseline method [4] is listed in Table I, Table II for the Imaging Radar dataset and RadarScenes dataset respectively. It can be observed that there is a huge improvement in classification performance for classifier using our methodology as compared to the baseline method. Overall, the multi-class accuracy is increased by 8.6% and VRU/Vehicle accuracy is increased by 5.3% for the Imaging Radar dataset whereas the multi-class accuracy is increased by 8.5% for RadarScenes dataset respectively. This is because the higher number of frames, novel temporal sampling and grouping and hierarchical temporal-spatial feature extraction, used by our proposed model allow the classifier to extract accurate and robust features that helps in classification of different objects. Lastly, a higher classification performance is observed on the Imaging Radar dataset than RadarScenes. This is because the Imaging Radar dataset has a better angular resolution than RadarScenes dataset and provides additional height information.

Fig. 7 shows the confusion matrices for a classifier using PointNet++ (Fig. 7a) and our proposed model (Fig. 7b) respectively for all of the 5 classes. Here, we can see that the accuracy is increased for each of the 5 classes using

Authorized licensed use limited to: TU Delft Library. Downloaded on May 07,2025 at 09:09:52 UTC from IEEE Xplore. Restrictions apply.

Table I: Performance comparison on the Imaging Radar dataset.

Method	PointNet++	Our model
Multi-class Accuracy	87.5%	96.1%
VRU/Vehicle Accuracy	92.7%	98.0%

Table II: Performance comparison on RadarScenes dataset.

Method	PointNet++	Our model
Multi-class Accuracy	80.5%	89.0%

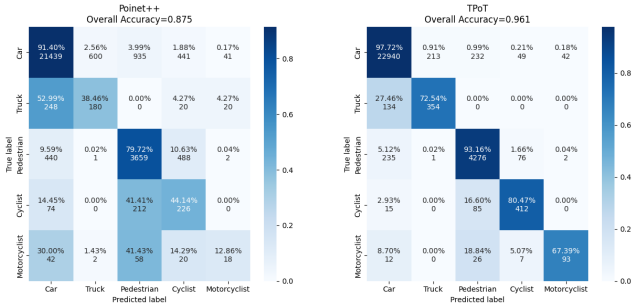
our proposed model. The highest increase can be seen for cyclist, motorcyclist and pedestrian classes. This is because these VRU objects have lower number of points per frame and have an irregular motion pattern that can confuse a single frame classifier. Using multiple frames, allow the classifier to utilize more points per object as well as capture the motion of these VRU objects better, helping in classification.

Fig. 8 shows the confusion matrices for a classifier using PointNet++ (Fig. 8a) and our proposed model (Fig. 8b) respectively for all of the 5 classes. Here, we can see that the accuracy is increased for each of the 5 classes using our proposed model. The highest increase can be seen for two-wheeler and pedestrian group classes. This is because of the lower number of points per frame large and the irregular motion pattern for these classes that can confuse a single frame classifier. Using multiple frames, allow the classifier to utilize more points per object as well as to capture the motion of these VRU objects better, helping in classification. A drop in accuracy is seen for the pedestrian class where some pedestrians are assigned as pedestrian group class. This may be due to the nature of labels of the RadarScenes dataset where the difference between pedestrian and pedestrian group label is that "pedestrian group label is assigned where no certain separation of individual pedestrians can be made" [10]. This means that for close-by pedestrians, they are sometimes assigned as a pedestrian group class instead of multiple instances of pedestrian class. So, features learned for a pedestrian group is similar to the features of multiple instances of close-by pedestrian class. When multiple frames are used, there is a higher probability that these close-by pedestrians will be classified as a pedestrian group class because points from multiple frames will get aggregated and these will produce features similar to that coming from a pedestrian group class causing the classifier to confuse them with pedestrian group class.

D. Robustness to Segmentation Quality

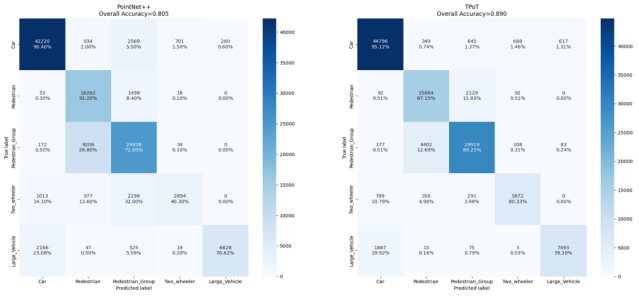
In real life ADAS/AD applications, segmentation errors of tracked objects are almost inevitable due to many factors, such as trajectory prediction errors, crowded scenes etc. For this reason, being robust to different segmentation errors is another key performance indicator of the classification module. From the point cloud perspective, segmentation error could cause two issues:

- Missing real object points;



(a) PointNet++ (b) Our Model

Figure 7: Confusion Matrix for the Imaging Radar Dataset.



(a) PointNet++ Classification. (b) Our Model

Figure 8: Confusion Matrix for RadarScenes Dataset.

- Including noise points (points that does not belong to object);

The problem is illustrated by Fig. 9. Blue circles are real object points, while red triangles represent noise points. When the tracker output (red cube) is far from the real location (blue cube), real object points are lost, and noise points are added to the input. This results in a disrupted point cloud that may confuse the classifier.

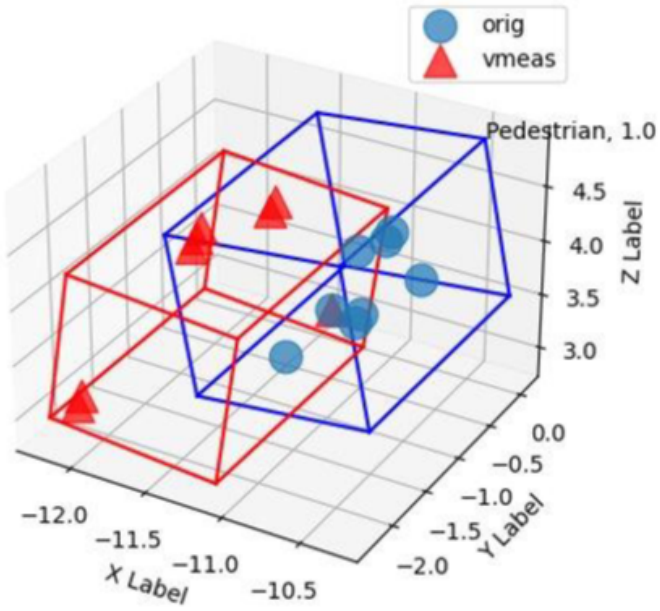


Figure 9: Point segmentation errors in tracking.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 07,2025 at 09:09:52 UTC from IEEE Xplore. Restrictions apply.

To measure the segmentation quality of point cloud, we use the V-measure metric (as proposed in [7]), which includes the following two scores (sub-metrics):

- **Homogeneity score**: indicates the portion of real object points included as as shown in (4).

$$Homogeneity = 1 - \frac{H(Y_{true}|Y_{pred})}{H(Y_{true}} \tag{4}$$

- **Completeness score**: indicates the portion of real object points to the total points (including noise) as shown in (5).

$$Completeness = 1 - \frac{H(Y_{pred}|Y_{true})}{H(Y_{pred}} \tag{5}$$

To reflect the overall quality, V-measure is used which is the harmonic mean between homogeneity and completeness scores. It can be calculated by (6).

$$V - Measure = \frac{2 \cdot Homogeneity \cdot Completeness}{Homogeneity + Completeness} \tag{6}$$

Fig. 10 gives a comparison of the performance between the baseline method and our proposed model for tracked objects with different V-measures. Comparing to the baseline method, the proposed methodology is more robust to different levels of segmentation errors. This effect is visible especially for lower V-measure since the point cloud is severely disturbed for these cases causing the baseline method to fail. On the other hand, our proposed model is able to retain the performance because it can take advantage of multiple frames aided with temporal sampling and grouping.

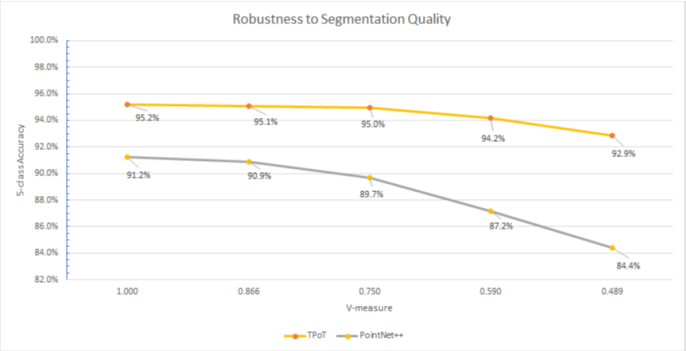


Figure 10: Robustness to segmentation errors.

E. Resource Requirements

Table III gives a comparison of the resource requirements between the baseline classifier and our proposed model. It can be seen that our model has a much lower resource requirement with a reduced memory footprint of around 5 times (78k instead of 382k activations), reduced parameter requirement of around 6 times (20k instead of 119k parameters) and a reduced compute requirement of around 30 times (28k instead of 728k multiply accumulates (MACs)). This can be achieved because the temporal-spatial sampling and grouping modules are highly efficient in extracting features from radar point cloud data. Therefore, the neural network part in both the feature extraction module and the classification head can be extremely lightweight.

Table III: Resource Requirements for a Single Inference

Method	PointNet++	Our Model
Memory Footprint	382k	78k
Number of Parameters	119k	20k
Number of MACs	728k	28k

IV. Conclusion

In this work, we propose a custom object classification NN model that uses multiple frames to solve the problem of target classification. A queue of points coming from multiple timestamps is passed to a hierarchical temporal-spatial feature extraction module using a temporal scheme for sampling and grouping. The output of this module is passed to a classification head that outputs labels. Applying our strategy showed a consistent improvement in classification performance as compared to a baseline classifier using PointNet++ [4]. Moreover, our method is more robust to segmentation errors as compared to the baseline method because of using multiple frames. Lastly, the overall classification methodology requires much less memory and processing power on the die which is extremely important when deploying to edge processors.

References

[1] M. Hassan, F. Fioranelli, A. Yarovoy, and S. Ravindran, "Radar multi object tracking using dnn features," 11 2023, pp. 1–6.

[2] F. J. Abdu, Y. Zhang, M. Fu, Y. Li, and Z. Deng, "Application of deep learning on millimeter-wave radar signals: A review," *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/1951

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings - IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, vol. 2017-December, 2017.

[5] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Advances in Neural Information Processing Systems*, vol. 2018-December, 2018.

[6] 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*, 2019.

[7] 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?" 07 2020, pp. 1–8.

[8] T. Akita and S. Mita, "Object tracking and classification using millimeter-wave radar based on lstm," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1110–1115.

[9] O. Schumann, J. Lombacher, M. Hahn, C. Wohler, and J. Dickmann, "Scene Understanding with Automotive Radar," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, 2020.

[10] "Radarscenes dataset labeling," https://radarscenes.com/dataset/labeling/, accessed: May 1, 2024.

Authorized licensed use limited to: TU Delft Library. Downloaded on May 07,2025 at 09:09:52 UTC from IEEE Xplore. Restrictions apply.