A NOVEL POINT CLOUD EXTRACTION AND FEATURE DESIGN APPROACH FOR RADAR-BASED OBJECT CLASSIFICATION

Yi Zhou,[1,2,3,5], Miguel López-Benítez[5], Limin Yu[2], Yutao Yue[134*]*

[1]*Institute of Deep Perception Technology, JITRI, Wuxi 214000, China*
[2] *School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China*
[3]*XJTLU-JITRI Academy of Industrial Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China*
[4]*Department of Mathematical Sciences, University of Liverpool, Liverpool L69 7ZX, UK*
[5]*Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 7ZX, UK*
**Email: limin.yu@xjtlu.edu.cn; yueyutao@idpt.org*

Keywords: RADAR POINT CLOUD, CLASSIFICATION, FEATURE DESIGN

Abstract

Radar-based object detection has been extensively studied, with many researchers employing deep learning models like PointNet for processing radar point clouds. However, a common challenge in this domain is dealing with sparse point clouds generated by commercial radar sensors. From a neural network perspective, managing redundant and noisy data is generally more feasible than handling sparse data. Some researchers have attempted to address this challenge by exploring the use of radar raw data, such as Range-Angle-Doppler (RAD) tensors, as input. However, RAD tensors are dense 3D tensors that demand significant memory and computational resources for storage and processing. In our study, we argue that the current point cloud representation is practical, but the point clouds generated by traditional detection pipelines tend to be overly sparse. To tackle these issues, we investigate point cloud extraction and feature design to better align with the requirements of deep learning models. Specifically, we propose a pipeline that employs binary segmentation for point extraction to minimize information loss. Additionally, we leverage local Signal-to-Noise Ratio (SNR) information to assign weights to each point, effectively mitigating the impact of redundant and noisy points. Our experimental results, based on our dataset, demonstrate that using our feature set can significantly enhance the accuracy of the PointNet model without requiring modifications to the model's architecture.

1 Introduction

Millimeter-wave (mmWave) radars have found widespread use in advanced driver assistance systems (ADAS) applications due to their maturity, compact size, and cost-effectiveness [1]. While lidar remains a dominant sensor for autonomous driving, its high cost limits its widespread adoption. Consequently, researchers have increasingly turned their attention to mmWave radars. The growing demands for high-level autonomous driving require improved spatial resolution from radar sensors. Multiple-Input Multiple-Output (MIMO) techniques have significantly enhanced the performance of modern radar sensor. However, compared to advances in waveform and antenna design, the signal processing pipeline in commercial radars has seen limited progress. Traditional radar signal processing pipelines enable reliable detection and tracking of moving targets, but they have limited classification capability and poor detection ability for static objects. These limitations restrict their adoption in autonomous driving scenarios.

The practical automotive radar detection framework can be divided into two main components: the sensor end and the central computing end. The signal processing pipeline is efficiently implemented onboard the radar sensor end, where the radar point cloud is generated as an output. This radar point cloud is then transmitted to the computing units, where it is possible to apply deep learning models to provide reliable detection results. In the traditional radar pipeline, the Constant False Alarm Rate (CFAR) detector is employed to extract point detections. Additionally, unsupervised clustering methods, such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN), are used to propose clusters from the radar point cloud. Significant effort has been devoted to feature set design and selection for the cluster.

In recent years, researchers have been exploring deep learning methods for radar perception without the need for traditional feature engineering. These neural networks can accept various representations of radar data as input, including ADC signals [2], range profiles [3], range-angle (RA) maps [4], RAD tensors [5,6,7], point clouds [8], and the concatenation of CFAR detections with the RA map [9]. However, despite the popularity of using pre-CFAR data as input in many recent research papers, these methods may not be practical for real-world radar sensors. This is due to the limited computational resources available at the edge, making it challenging to process raw radar data using deep learning techniques. One potential solution is to transfer the pre-CFAR data to a central computing unit for processing. However, this design approach is not widely adopted due to the limited transmission bandwidth required to transfer large pre-CFAR data. Additionally, from an algorithmic perspective, pre-CFAR data has disadvantages, including inaccurate annotation and

redundant computation wasted on background cells. In this work, we focus on using the radar point cloud as input. The traditional radar detection pipeline with the point target assumption faces difficulties in two aspects. Firstly, estimating the threshold for CFAR proves challenging, resulting in the loss of spatial information of weak scattering points. The RD CFAR detector also faces challenges with static object detection. Secondly, small objects may be misidentified as clutter or noise during the clustering stage. Due to information loss during the CFAR and DBSCAN processes, the power of deep learning-based classifiers cannot be fully exploited.

In this work, we explore a simple yet effective method to increase the point cloud density by treating the detection as a dynamic binary segmentation task. The resulting point cloud is noisy, but we design an SNR-based attention mechanism to assign weights to the points. By modifying the point cloud density and feature dimension, our PointNet-based model outperforms other models with multi-scale architecture and powerful local feature detectors.

The remainder of this article is organized as follows: Section 2 provides highlights of research progress in radar point cloud processing. Section 3 presents a detailed explanation of our proposed point cloud extraction and feature set design method. Section 4 introduces the dataset and experimental settings. In Section 5, the experimental results are thoroughly analyzed. Finally, Section 6 concludes this article.

2. Related Works

In traditional radar point cloud processing, hand-crafted features are extracted for each radar cluster and utilized as inputs for machine learning classifiers. Scheiner *et al.*[10] extensively explore feature design for the random forest and LSTM classifiers. These features can be broadly categorized into three groups: statistical features based on fundamental radar measurements like range, angle, amplitude, and Doppler; geometric features describing the spatial distribution of detections within radar clusters, including measurements related to bounding boxes; and micro-Doppler features capturing fine-grained motion information by examining the distribution of Doppler values within clusters. Experimental findings have highlighted the value of geometric features and micro-Doppler features as complementary additions to the standard sensor output. In their subsequent work [11], they introduce a feature selection strategy to identify optimized feature sets for ensemble classifiers. Although ensemble classifiers often utilize most features, individual classifier results suggest that range and Doppler features hold greater importance, while angular and shape-related features tend to be excluded during feature selection due to the low angular resolution of the automotive radar sensors employed.

With the introduction of large-scale radar dataset and advancement in model architecture, deep learning methods outperform traditional machine learning classifiers across various tasks. The deep learning methods for processing radar point clouds can be categorized into two paradigms.

The first paradigm involves using PointNet variants to process radar point clouds. PointNet's architecture was initially designed for point clouds with higher density and detailed local structures, but it can be adapted for radar point clouds with some modifications to input dimensions. PointNet [12] extracts point-wise features using shared MLPs and aggregates these features in the max-pooling layer. PointNet++ [13] extends this structure by creating a hierarchical architecture through sampling and grouping point features at multiple scales. Recent improvements focus on enhancing local geometric feature extraction using techniques like convolution [14], graph neural networks [15], or attention mechanisms [16]. However, the recent work [17] suggests that a deeper architecture consisting of multiple lightweight residual MLP modules can achieve good performance without complex local geometry extractors.

The second paradigm involves converting point clouds into dense representations, such as grids or pillars, and then applying CNN for processing. However, one challenge with this approach is the potential loss of distribution information when aggregating point cloud data into grids. To address this, Kohler *et al.* [18] leverage the kernel point convolutions to improve the encoding of local point cloud contexts during grid rendering. Liu *et al.* [19] build pillar representation and density features through kernel density estimation (KDE) in feature encoding stage.

Scheiner *et al.* [20] conduct a comprehensive comparison of various methods for radar point segmentation tasks, including DBSCAN+LSTM, PointNet++ with DBSCAN, PointPillars, grid mapping + YOLOv3, and PointNet++ with LSTM. Their experimental results revealed that the YOLOv3 architecture achieved the best performance, followed closely by a modular approach combining PointNet++, a DBSCAN algorithm, and an LSTM network. PointPillars, on the other hand, performed significantly worse than the other methods, possibly due to the low resolution of the radar data used in their experiments.

3. Methodology

3.1 Point Cloud Extraction

We follow the traditional radar signal processing pipeline which applies range FFT and Doppler FFT to obtain the radar RD map. The next step is to extract point clouds from the RD map. The traditional method involves firstly apply CFAR detector to RD map, then resolve the angle for each detection. This detection pipeline is widely used for several reasons: radar typically exhibits low spatial resolution, allowing it to be effectively modelled as point targets, and the limited bandwidth of the Controller Area Network (CAN) protocol expects fewer detections for real-time transmission. However, with the emergence of high-resolution radar, many researchers argue that the traditional detection pipeline sacrifices valuable spatial information and overlooks static objects. Furthermore, deep learning models are known to be more resilient to input noise than information loss. The introduction of automotive ethernet further allow the transmission of large-scale point cloud data. Nevertheless, as discussed in the previous section, we suggest that directly processing raw radar data is

computationally intensive and cannot achieve real-time performance. To obtain a radar representation that is both sufficient and feasible, we propose a method to extract point clouds from the RA map using image binary segmentation method.

Regarding the radar RA map, we treat it as an image and apply binary segmentation methods, which are commonly used to distinguish between foreground and background in images. However, traditional image segmentation methods are typically applied globally, which can be problematic when dealing with the non-uniform energy scale of different object classes, such as pedestrians with a low Radar Cross Section (RCS). To address this, we utilize a sliding window technique to extract local patches from the radar data and perform segmentation within each window.

For binary segmentation, we utilize Otsu's method, which iteratively calculates an optimal threshold by minimizing within-class variance and maximizing between-class variance. The threshold t^* equation is computed as follows

$$t^* = \arg\max_t \frac{\left(\mu_T \cdot P(t) - \mu(t)\right)^2}{P(t) \cdot \left(1 - P(t)\right)} \qquad (3.1)$$

where $t^*$is the optimal threshold, μ_T is the average intensity within the window, $P(t)$ is the cumulative sum of probabilities up to intensity level t and $\mu(t)$ is the cumulative mean up to intensity level t.

The threshold is then applied to each window, assigning cells below the threshold to the background class and those above or equal to the threshold to the point cloud detection list. Once we have the RA point cloud, we determine the Doppler velocity by identifying the peak value along the Doppler dimension in the RAD tensor.

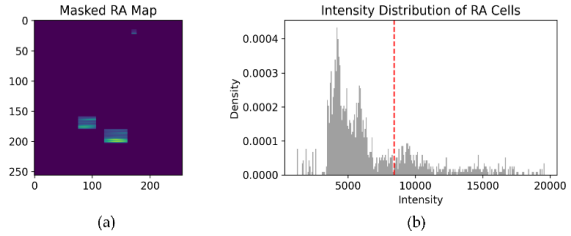


Figure 3.1 Intensity Distribution and Segmentation Threshold

3.2 Feature Extraction

Suppose the region proposal method is applied to extract the possible bounding box in RA map, then we select the foreground cells within the bounding box as the point cloud representation for each object. Since radar has higher dimensions of features in addition to the spatial distribution, we further investigate the feature extraction methods for radar point cloud.

The features can be categorized into two types: point-level features and cluster-level features. Point-level features include the intensity and the Doppler information of individual detections. Cluster-level features include the number of points and the feature distribution within each cluster. One limitation of point-level features is that they do not fully capture the relationships between points. In dense point clouds like lidar data, relationships between points can be captured using multi-scale down sampling. This approach takes advantage of the dense spatial information available in lidar point clouds. However, for radar point clouds, many adjacent points are filtered out due to the peak detection characteristics of the CFAR detector. As a result, the spatial relationships between points in radar data become less apparent. In our work, we address this issue by using the local SNR as a descriptor of the relationships between adjacent cells. Local SNR provides a measure of the signal strength relative to the surrounding noise cells. By incorporating SNR as a descriptor, we can effectively capture the relatively distribution of intensity within each cluster.

Similar to the operations in the CFAR detector, we calculate the local SNR in a sliding window mechanism. As illustrated in Figure 3.2, we split the window around the test cell into two sets: training cells and guard cells. The training cells are used to estimate the average power level of clutter or noise present in the radar data. They provide a reference for the background signal. On the other hand, the guard cells consist of cells around the peaks, which are caused by the side-lobe of the radar beam. Guard cells serve to prevent these side-lobe peaks from affecting the SNR calculation. The local SNR is then defined as the ratio of the intensity of the test cell to the average power level estimated from the training cells. This calculation allows us to obtain the SNR value for each detection, providing a measure of the signal strength relative to the background noise or clutter. As depicted in Figure 3.2, we calculate SNR in both RA and RD maps to model the energy distributional features along different dimensions.

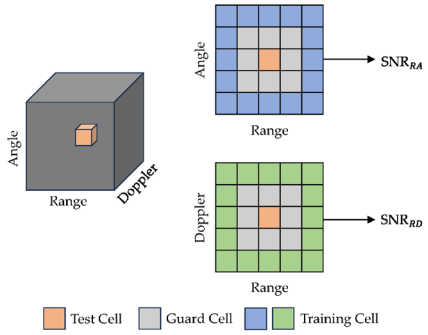


Figure 3.2 Local SNR for RA map and RD map

3.2 Attention Module for Intensity Features

When testing the vanilla PointNet on radar datasets, several observations were made. Firstly, it was found that not all points in the radar data carry equal importance. If we apply peak grouping to significantly reduce the density of point cloud, the performance drops but not very much. This finding suggests that certain points, particularly those with high relative intensity, are more crucial for classification. The second observation relates to the impact of modifying the SNR by adding Gaussian white noise to the test set. The classifier's

performance significantly dropped in low SNR cases. The detection methods returned numerous noisy data points, affecting the spatial distribution of the point cloud and consequently impacting classification accuracy.

To address these issues and improve the model's robustness, an attention mechanism is proposed, associating weight maps to data points to achieve a more reliable performance. The point properties, such as range and Doppler, represent physical measurements with clear interpretations. Applying attention to these features in earlier layers will lead to unrealistic measurements. Among all these features, intensity is better suited to be interpreted as the confidence feature. In our feature design, we have included two additional local SNR features. These two SNR features can be considered as transformed versions of intensity and provide energy distributional information in two different views. To fuse the intensity information, we have designed a lightweight attention module, as illustrated in Figure 3.3. The SNR in the RA dimension serves as query, and the SNR in the RD dimension serves as key. No additional weight matrix is used, as the SNR can be regarded as a pre-calculated adaptive transformation applied to intensity. It also saves computation and avoid overfitting during training. The attention score matrix is obtained by calculating the dot product between the query and key matrices and applying the softmax function. Next, the intensity is multiplied with a learnable parameter matrix to obtain the value matrix. The transformed intensity is then multiplied with the attention score matrix to obtain the attention-aware representation of intensity.

$$I' = \text{softmax}(SNR_{\text{RA}}(I) \cdot SNR_{\text{RD}}(I)^T)V(I) \quad (3.2)$$

where I is the intensity, SNR_{RA} and SNR_{RD} are the pre-calcaulated local SNR in RA and RD maps respectively, V is a learnable weight matrix, I' is the attention-aware representation of the intensity after applying the attention mechanism. Then, this enriched representation is concatenated with the other features and fed into subsequent layers of PointNet.

The attention module offers two main benefits: firstly, it reduces the input dimension, thereby decreasing the model size and enabling higher computational efficiency. Secondly, it enhances the model's ability to recognize critical points and relationships among data points, resulting in improved classification accuracy.

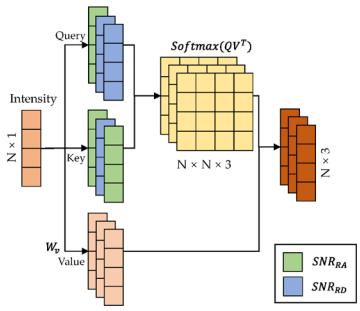


Figure 3.3 Attention Module to Fuse Intensity Features

3.3 Radar PointNet

In this project, the classification model is built on the PointNet architecture, as depicted in Figure 3.4. PointNet utilizes shared MLPs applied point-wise, followed by a global max-pooling to extract global feature of the input point cloud. However, the vanilla PointNet architecture is designed for high-resolution 3D point cloud data and may not be suitable for radar data. Radar point cloud is inherently sparse, making it difficult to visually identify local geometrical features.

The main characteristics of radar point clouds are their sparsity and high dimensionality. As illustrated in the previous section, our radar point cloud consists of 7 dimensional features, including range, azimuth, Doppler velocity, number of points, intensity, SNR in RA dimension, and SNR in RD dimension. For the point-level features, we can directly feed them into the PointNet to extract point-wise features. For the cluster-level feature, such as number of points, we can add it into the global features after the max-pooling layer. Therefore, the vanilla PointNet for radar point cloud takes a (Batch, N, 6) dimension point cloud as input to the PointNet head. If we employ the attention module to intensity features, the input size can be reduced to (Batch, N, 4). Then, the number of point feature is appended to the 1D features after max pooling to get a (Batch, 1024) dimensional global feature. Finally, the MLP-based classification head is used to predict the class distribution of the point cloud.

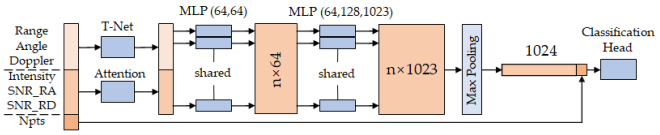


Figure 3.4 Radar PointNet Architecture

3.4 Focal Loss

Given that many radar datasets suffer from data imbalance, often stemming from limited scenarios and challenges in annotating vulnerable road users, we employ the focal loss as a solution to tackle this issue. The focal loss is a tailored loss function designed specifically to address class imbalance problems in multi-class classification tasks. Its main objective is to enhance the model's performance by assigning greater weights to challenging, misclassified instances, while reducing the weight of correctly classified ones. This strategic weighting allows the model to pay more attention to difficult examples and effectively handle datasets with imbalanced class distributions.

Mathematically, the focal loss for multi-class classification is defined as follows:

$$L_{focal} = -\sum_{i=1}^{N}((1-p_i)^{\gamma} \cdot \log(p_i)) \quad (3.3)$$

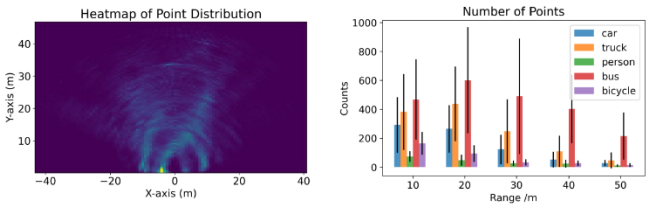
where N represents the number of classes, p_i is the predicted probability of the i-th class, and γ is the focal parameter that controls the degree of downweighting.

4 Datasets and Experiments

4.1 Datasets

Recently, several radar datasets have been made public available for benchmarking deep learning methods. To investigate on feature design of point cloud, it is essential to have a dataset that includes radar raw data. Consequently, we have selected the dataset used in RADDet [7]. The radar sensor used in this dataset is the TI AWR1843-BOOST 77GHz radar, positioned on the sidewalks and facing the main roads. In the azimuth plane, the radar comprises 2 transmit (Tx) and 4 receive (Rx) antennas, forming a virtual 8-antenna receiving array with a 3dB beamwidth of 15 degrees. The waveform is configured for short range mode, offering a maximum range of 50 meters. The dataset provides synchornized radar RAD data and stereo camera data. The radar RAD data is represented as a three-dimensional tensor with dimensions of (256, 256, 64) corresponding to range, angle, and Doppler, respectively. It has a range resolution of 0.19 m per bin, an angular resolution of 0.34 degrees per bin, and a velocity resolution of 0.42 m/s per bin.

The original dataset was designed for the detection purpose with radar RAD tensor as inputs. We regenerated the classification dataset by applying CFAR to get the point clouds, cropping the points within the bounding boxes and associating the corresponding labels. The resulting dataset contains 12,211 frames of radar point clouds for the classification task. The annotations cover six categories, namely 'person', 'bicycle', 'car', 'bus', and 'truck'. However, the data distribution is imbalanced, with the 'car' category being dominant, accounting for 60.5%, while 'person' and 'truck' account for 21.3% and 14.2%, respectively. The remaining data belong to 'bus' and 'bicycle'. From Figure 4.1 (a), we can clearly observe that the distribution of objects is predominantly concentrated in the short-range area. From Figure 4.1 (b), we notice a decreasing trend in the number of points as the range increases. This suggests that objects farther away from the radar sensor tend to have fewer detected points, which aligns with the expected behavior of radar reflections. Additionally, we observe that vulnerable road users, including pedestrians and bicycles, contain a much smaller number of points compared to other objects. This phenomenon is a result of their weak reflection characteristics, making them challenging to detect accurately with radar.



(a) (b)

Figure 4.1 Dataset Statistics

4.2 Evaluation Metrics

In our paper, we have employed a range of classification metrics to comprehensively evaluate the model's performance. Accuracy serves as a fundamental metric for measuring the overall correctness of predictions. Additionally, we utilize balanced accuracy, which is defined as the average of recall obtained for each class, especially useful for assessing models on our imbalanced datasets. For a finer-grained evaluation of the model's performance across different classes, we calculate precision, recall, and F1 score. Precision measures the model's ability to make correct positive predictions, recall assesses the model's capacity to capture all positive instances, and the F1 score provides insights into the model's performance on individual classes by balancing precision and recall.

4.3 Experiment Settings

For the input data, we utilized a uniform sampling process to upsample or downsample each point cloud into 225 points. With a batch size of 32 and a feature dimension of 5, the resulting input data is shaped as (32, 225, 5). To ensure a fair evaluation of our model's performance, we adopted a random shuffle and split strategy to create three subsets: 80% for training, 10% for validation, and 10% for testing. This partitioning scheme allowed us to train the model on the majority of the data while assessing its performance on an independent subset for better test of generalization.

Regarding the training criterion, we set the gamma value to 1 for the focal loss and used a weight of 0.001 for the affine transformation loss. During testing, we employed the standard cross entropy loss criterion to evaluate the model and selected the best candidate based on its performance. For the training settings, we utilized the Adam optimizer with a learning rate of 0.01 to optimize the model's parameters during training. The maximum number of epochs for training was set to 100. Additionally, we employed the early stopping technique with a patience value of 10 epochs, which enabled us to halt training if the model's performance did not improve for a specified number of epochs. This strategy helps prevent overfitting and saves computational resources.

Instead of our proposed radar PointNet, we also compared other popular models for point cloud processing, including PointNet [13], PointNet++[14], PointConv[16], PointMLP[17] and PointCloudTransformer[18]. Since these models are originally designed for high resolution point cloud classification, we modify their network structures to better suit sparse radar point cloud. Most of these models have a hierarchical architecture inspired by PointNet++, so we take PointNet++ as an example to illustrate how we adapted the model architecture. Specifically, the first Set Abstraction (SA) layer operates on 64 points, with a sampling radius of 2 units and 32 samples per point. It applies an MLP with hidden layer sizes [64, 64, 128]. The second SA layer operates on 16 points, with a sampling radius of 4 units and use an MLP with hidden layer sizes [128, 128, 256]. The third layer groups all points together and applies an MLP with hidden layer sizes [256, 512, 1024].

5 Results Analysis

5.1 Comparison of Classifiers

Table 1 provides a comparison of different models based on their accuracy, balanced accuracy, number of parameters (Params), and floating-point operations (FLOPs). Among the models evaluated, our proposed radar PointNet achieves the second highest accuracy of 84.55%, outperforming the other models. It also demonstrates the highest balanced accuracy of 61.80%, indicating better performance in handling imbalanced datasets. Since we mainly modify the input to the vanilla PointNet, compared to the vanilla PointNet, the accuracy improve significantly, indicating the effectiveness of our feature design.

Interestingly, the Random Forest classifier are even higher than the other deep learning models. It indicates that the global statistics play a more important role than local spatial distribution. PointNet++ show competitive results with accuracies of 84.75% and balanced accuracy of 55.15%, which indicate the multi-scale features are important for the classification task. This in line with the results from random forest which indicates that the vanilla PointNet is too weak to extract global features from the radar point cloud, thus has the weaker performance. PointMLP achieves an accuracy of 81.89 with a relatively low parameter count and FLOPs. The transformer architecture PCT display very lower accuracy of 61.50%, while having highest FLOPs. The transformer architecture further enhances the local geometric structure, however not suitable for radar point cloud with irregular distribution.

Overall, the table highlights the importance of global statistics within the radar point cloud and the transformer model can finds hard to extract the attention map from the local structure due to the randomly distribution of the scattering points. Therefore, our SNR-based attention is more reasonable for radar point cloud and can significantly improve the PointNet performance.

Table 1 Classification Performance Comparison

Model	Accuracy (%)	Balance Acc(%)	Params (M)	FLOPs (M)
RandomForest	82.51	46.66	-	-
PointNet	81.64	43.33	1.604	66.307
PointNet++	84.75	55.15	1.467	44.721
PointConv	82.39	44.92	19.561	96.374
PointMLP	81.89	52.60	0.294	10.961
PCT	61.50	20.54	2.933	113.230
Ours	84.55	61.80	1.606	66.270

To better understand our improvements over vanilla PointNet, we plot the confusion matrix of the vanilla PointNet (a) and our model (b) in Figure 5.1. From the left, we can find the truck and bus are very hard to discriminate. All the predicted bus are actually bus. This is because of the imbalanced dataset and the similar geometric point distribution between these two classes. After we apply the intensity features and focal loss, we can find most of the bus and truck are classified correctly. However, the bicycle class show some performance degradation, as it is confused with car and person, despite have different geometric distribution. This further inspire the future work which balance between intensity and geometric features.

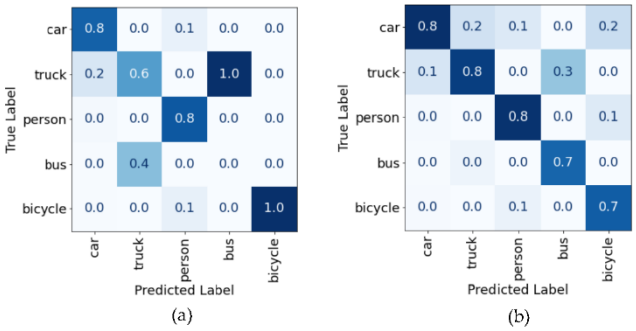


Figure 5.1 Confusion Matrix

5.2 Point Detection Methods

In this section, we compare the point detection methods, including the RA CFAR, RD CFAR, our proposed binary segmentation and the grid mapping representation. Figure 5.2 shows the mask of detections extracted from different CFAR detectors. The RA CFAR show significant spreads in angle dimension due to low angular resolution. Therefore, we apply a peak grouping for better localization of the point target. In Figure 5.3, we visualize an example of the detected point cloud of a vehicle given three different detectors. In addition, the grid mapping representation is built upon the binary segmentation by accumulating the detection result in a grid map and resize the image into the size of [224,224].

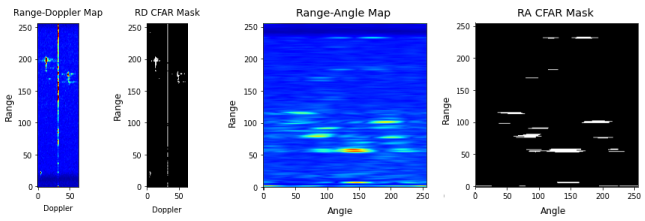


Figure 5.2 Detection Mask by CFAR Detectors

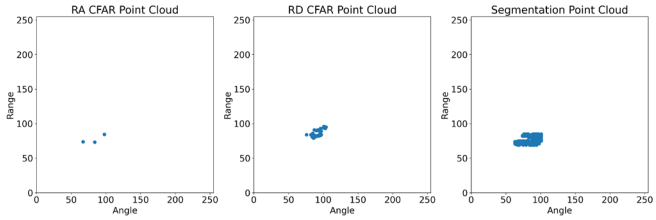


Figure 5.3 Comparisons of Point Cloud Density

Tabel 3 investigates the effect of spatial density. From the table, we can find our method will return far more point detections per object than traditional CFAR detector and achieve the best performance. Although the points detected by our methods is much noisy than CFAR detector, the neural network is robust to these noisy input and benefits with simply increase the point

density. The grid mapping representation lower the accuracy since the improper quantization.

Table 3 Effect of Spatial Density

Method	Average Point Number	Accuracy(%)
RA CFAR	5	77.33
RD CFAR	38	81.15
Binary Seg	172	84.55
Grid Mapping	302 (occupied pixels)	79.91

5.3 Feature Importance

Table 4 investigate the feature importance for different classes using F1 score. To mitigate the effects feature design, we use the vanilla PointNet as the base classifier. When using all features, the Car and Person classes achieve relatively high F1 scores, but the Truck, Bus and Bicycle classes perform poorly. Conver the coordinate from Cartesian to Range Doppler index leads to poorer performance, possibly due to the irregular shapes in RD map confuse the classifier than the simple point cluster in Cartesian coordinate. Excluding the Doppler feature decreases overall performance, particularly for Truck, Bus, and Bicycle. Omitting the Number of Points feature significantly impacts the Bicycle class. It decreases truck and increase bus performance. Since these two classes are easily confused with each other, the performance change does not reveal significant information. This also applies to the intensity where the same phenomenon happens. The absence of the Intensity feature leads to minor performance decreases. Overall, we can find use all features leads to the best performance.

Table 4 Feature Importance

Feature	F1 Score				
	Car	Truck	Person	Bus	Bicycle
All	90.39	46.93	88.75	25.00	31.58
XY to RD	88.33	11.27	89.24	0.00	11.76
w/o Doppler	85.84	7.97	82.77	0.00	0.00
w/o Npts	88.77	28.76	85.59	51.61	0.00
w/o Intensity	87.60	9.56	87.90	53.33	26.67

5.4 Effect of SNR-based Attention

Table 5 investigates the effectiveness of our proposed attention mechanism for intensity features. In this analysis, "multi-head" implies the use of three query-key pairs: SNR_{RA}- SNR_{RA} , SNR_{RD}- SNR_{RD} and SNR_{RA}- SNR_{RD}, and "single head" means we only employ the SNR_{RA}- SNR_{RD} head. The table results suggest that the utilization of attention modules can significantly enhance the precision of truck and bus identification and the recall of the bicycle class. Furthermore, the multi-head mechanism introduces slight additional improvements across most of the classes.

In Figure 5.4, we select some data examples for each class and plot the attention map for the point cloud. We can find the SNR attention mechanism performs a similar way as kernel fitting. For the close range detections the peak of intensity are assigned to more weights, while the detections at distance all the weights are assigned to points spanning the whole objects.

Table 5 Effectiveness of SNR Attention

Classifier	Precision				
	Car	Truck	Person	Bus	Bicycle
Multi-Head	82.81	92.19	89.30	73.33	86.21
Single-Head	83.97	79.17	82.82	66.67	69.44
w/o Attention	81.65	72.22	82.18	0.00	100.00
	Recall				
	Car	Truck	Person	Bus	Bicycle
Multi-Head	97.50	23.05	91.50	57.89	39.06
Single-Head	93.79	22.27	95.97	52.63	39.06
w/o Attention	96.53	15.23	92.84	0.00	1.56

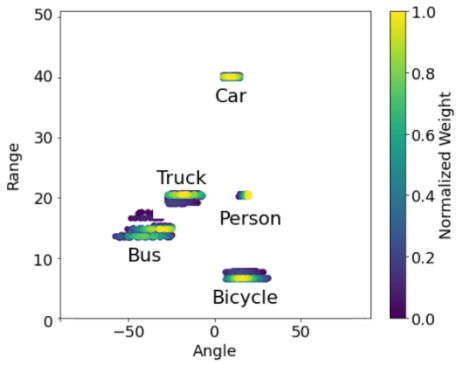


Figure 5.4 Attention Visualization

6 Conclusion

Most radar point cloud analysis work has primarily focused on designing model architectures. However, in this study, we've undertaken an investigation into the impact of point cloud generation and feature set design on radar point cloud classification tasks. We've proposed a straightforward yet highly effective method to increase point cloud density and enrich the feature set. Our approach involves reinterpreting the detection task as a dynamic binary segmentation problem. While the resulting point cloud may exhibit noise, we have innovatively introduced an SNR-based attention mechanism designed to assign weights to individual points. By strategically adjusting point cloud density and feature dimension, our radar PointNet model consistently outperforms alternative models.

In future research, we plan to explore the incorporation of features into the point sampling mechanism used in hierarchical model architectures. Additionally, we will investigate the balance between geometrical features and intensity features to further enhance radar point cloud classification.

7 Acknowledgements

This work received financial support from Jiangsu Industrial Technology Research Institute (JITRI) and Wuxi National Hi-Tech District (WND).

8 References

[1] Zhou, Y., Liu, L., Zhao, H., et al.:'Towards deep radar perception for autonomous driving: Datasets, methods, and challenges', Sensors, 2022, 22(11), p.4208

[2] Giroux, J., Bouchard, M., Laganiere, R.:'T-FFTRadNet: object detection with swin vision transformers from raw adc radar signals', arXiv preprint, 2023, 2303.16940

[3] Liu, Y., Wang, F., Wang, N., et al.:'Echoes Beyond Points: Unleashing the Power of Raw Radar Data in Multi-modality Fusion', arXiv preprint, 2023, 2307.16532

[4] Wang, Y., Jiang, Z., Li, Y., et al.:'RODNet: A real-time radar object detection network cross-supervised by camera-radar fused object 3D localization. IEEE Journal of Selected Topics in Signal Processing', 2021, 15(4), pp.954-967

[5] Ouaknine, A., Newson, A., Pérez, P., et al.: 'Multi-view radar semantic segmentation'. In Proc. Int. Conf. Computer Vision, Virtual, October 2021, pp. 15671-15680

[6] Gao, X., Xing, G., Roy, S., et al.:'Ramp-cnn: A novel neural network for enhanced automotive radar object recognition', IEEE Sensors Journal, 2020, 21(4), pp.5119-5132.

[7] Zhang, A., Nowruzi, F.E., Laganiere, R:'Raddet: Range-azimuth-doppler based radar object detection for dynamic road users', Conf. Robots and Vision, Virtual, May 2021, pp. 95-102

[8] Danzer, A., Griebel, T., Bach, M., et al.:'2d car detection in radar data with pointnets', IEEE Intelligent Transportation Systems Conference, Auckland New Zealand, October 2019, pp. 61-66

[9] Shao, Z., Zhang, X., Xu, X., et al.:'Cfar-guided convolution neural network for large scale scene sar ship detection', IEEE Radar Conference, San Antonio, USA, May 2023, pp. 1-5

[10] Scheiner, N., Appenrodt, N., Dickmann, J., et al.:'Radar-based feature design and multiclass classification for road user recognition', IEEE Intelligent Vehicles Symposium, Changshu, China, June 2018, pp. 779-786

[11] Scheiner, N., Appenrodt, N., Dickmann., et al.:'Radar-based road user classification and novelty detection with recurrent neural network ensembles', Paris, France, IEEE Intelligent Vehicles Symposium, June 2019, pp. 722-729

[12] Qi, C.R., Su, H., Mo, K., et al.:'Pointnet: Deep learning on point sets for 3d classification and segmentation', Proc. IEEE Conf. Computer Vision and Pattern Recognition Honolulu, Hawaii, July 2017, pp. 652-660

[13] Qi, C.R., Yi, L., Su, H., et al.:'Pointnet++: Deep hierarchical feature learning on point sets in a metric space', Advances in neural information processing systems, 2017, 30

[14] Wu, W., Qi, Z., Fuxin, L.:'Pointconv: Deep convolutional networks on 3d point clouds', Proc. IEEE Conf. Computer Vision and Pattern Recognition, Long Beach, USA, June 2019, pp. 9621-9630

[15] Wang, Y., Sun, Y., Liu, Z., et al.:'Dynamic graph cnn for learning on point clouds', ACM Transactions on Graphics, 2019, 38(5), pp.1-12.

[16] Ma, X., Qin, C., You, H., et al.:'Rethinking network design and local geometry in point cloud: a simple residual mlp framework', Int. Conf. Learning Representations, Virtual, October 2021

[17] Guo, M.H., Cai, J.X., Liu, Z.N., et al.:'Pct: Point cloud transformer', Computational Visual Media, 2021, 7, pp.187-199

[18] Köhler, D., Quach, M., Ulrich, M., et al.:'Improved Multi-Scale Grid Rendering of Point Clouds for Radar Object Detection Networks', arXiv preprint, 2023, 2305.15836

[19] Liu, J., Zhao, Q., Xiong, W., et al.:'SMURF: spatial multi-representation fusion for 3d object fetection with 4d imaging radar', arXiv preprint, 2023, 2307.10784

[20] Scheiner, N., Kraus, F., Appenrodt., et al.:'Object detection for automotive radar point clouds–a comparison', AI Perspectives, 2021, 3(1), pp.1-23