Radar-PointGNN: Graph Based Object Recognition for Unstructured Radar Point-cloud Data

Svenningsson, Peter; Fioranelli, Francesco; Yarovoy, Alexander

DOI
10.1109/RadarConf2147009.2021.9455172

Publication date
2021

Document Version
Final published version

Published in
2021 IEEE Radar Conference

Radar-PointGNN: Graph Based Object Recognition for Unstructured Radar Point-cloud Data

Peter Svenningsson, Francesco Fioranelli and Alexander Yarovoy
Microwave Sensing, Signals and Systems (MS3)
Delft University of Technology
Delft, The Netherlands
{p.o.svenningsson, F.Fioranelli, a.yarovoy}@tudelft.nl

2021 IEEE Radar Conference (RadarConf21) | 978-1-7281-7609-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/RadarConf2147009.2021.9455172

***Abstract*—Perception systems for autonomous vehicles are reliant on a comprehensive sensor suite to identify objects in the environment. While object recognition systems in the LiDAR and camera modalities are reaching maturity, recognition models on sparse radar point measurements have remained an open research challenge. An object recognition model is here presented which imposes a graph structure on the radar point-cloud by connecting spatially proximal points and extracts local patterns by performing convolutional operations across the graph's edges. The model's performance is evaluated by the nuScenes benchmark and is the first radar object recognition model evaluated on the dataset. The results show that end-to-end deep learning solutions for object recognition in the radar domain are viable but currently not competitive with solutions based on LiDAR data.**

***Index Terms*—object detection, object recognition, radar, geometric deep learning, nuScenes**

I. Introduction

Autonomous driving and advanced driver-assistance systems require the perception of the surrounding environment. A subtask in perception is the detection and classification of objects in the environment, here referred to as object recognition to avoid semantic ambiguity in the radar research domain with regards to detection tasks. To aid in this task an autonomous system is equipped with a sensor suite which commonly includes camera, LiDAR and radar sensor modalities. Radar is an attractive sensor type for object recognition because of its robustness to weather and lighting conditions in addition to providing information on target velocities.

Commercially available radar sensors output a marked point-cloud consisting of radar detection points. The points are specified in two spatial coordinates and are marked with radial velocity u in relation to the sensor and radar cross-section σ.

A conventional object recognition pipeline for radar data commonly involves clustering the radar detection points and classifying the clusters based on the statistical attributes of its members, as described in [1], [2] and [3]. In contrast, state of the art object recognition models in the LiDAR domain [4] and the camera domain [5] are end-to-end deep learning models which leverage convolutional operations to extract local patterns from the data.

Recently, end-to-end deep learning models developed in the LiDAR domain have been adapted for radar data as explored in [6]. The radar detection points are embedded by a PointNet based pipeline, achieving encouraging results on a private dataset. However, the development of end-to-end deep learning models for object recognition in the radar domain remains an open research challenge.

Graph neural networks (GNNs) have recently shown promising results for object recognition task on LiDAR data as explored in [7]. Their work explores graph construction by connecting spatially proximal points. Local patterns are then extracted by convolutional operations over the graph-edges. The work explores a single shot detector which provides competitive results for object recognition in the automotive setting on the KITTI benchmark [8]. Other notable work explores the classification [9] and segmentation [10] of point-clouds using GNNs.

In this paper we propose an object recognition model for radar data which draws from the LiDAR Point-GNN presented in [7]. The model takes as input the radar detection points in a graph representation and uses graph convolutions to embed each radar detection point into a contextualized representation. The model outputs an object proposal for each point in the point-cloud. The proposals are thresholded based on a predicted clutter-score[1] and spatially overlapping proposals are suppressed based on non-maximum suppression [11].

The proposed model is evaluated on the nuScenes dataset [12] with the task of recognizing 10 classes of objects in the automotive setting, a listing of these classes are found in [12]. This model is one of the first published result on multi-object recognition for radar data on a publicly available dataset and is the only model evaluated on the nuScenes benchmark exclusively using the radar modality. The proposed method is evaluated in comparison to a PointNet++ encoder [13] and a performance upper bound.

In summary, the contributions of this paper are:

- An object recognition model in the radar domain based on graph convolutions.
- The first published method for multi-object recognition for radar data evaluated on the public dataset nuScenes.

[1]The clutter-score is an analogue to the objectness-score introduced in [5]. The score differentiates objects of interests from background clutter and is used in the non-maximum suppression algorithm as a proxy for prediction confidence.

II. Previous work

Some previous work explores object recognition in the radar domain by clustering the radar detection points and classifies the clusters based on the statistical attributes of its members, as described in [1], [2] and [3]. Other work have aimed to recognize objects by measuring how well the data coincides with a pre-defined template as described in [14] and [15].

Previous work in deep learning methods have explored the use of PointNet [6] and PointNet++ [16] for multi-object recognition and semantic segmentation respectively, achieving strong results on private datasets of radar point-cloud measurements.

In [17] the authors explored using a convolutional neural network (CNN) to extract local patterns from the radar data cube which are included as covariates to the radar detection points. The patterns extracted from the radar data cube was shown to increase the efficacy of a radar detection point classifier. Other work such as [18] have explored using image object recognition pipelines such as YOLO [5] on projections of the radar data cube which showed encouraging results on detecting static objects.

A. Feature extraction from sets.

Architectures here collectively referred to as PointNets are able to extract features from unordered sets such as point-clouds. The PointNet first proposed in [19] utilizes a shared multi-layered perceptron (shared-MLP) to embed points in a high-dimensional feature space. Global features are extracted by taking the maximum element in each feature dimension which are then concatenated to the pointwise feature vectors. The PointNet++ model presented in [13] expands the architecture by using a PointNet to pool the input features of points in small spatial regions. Regions of various sizes are used to capture differently sized structures. In addition, sampling and grouping strategies are employed to reduce the computational complexity – a necessity when considering highly dense point-clouds as input.

B. Feature extraction from graphs.

Graph neural networks (GNNs) define a convolutional operation along the edges of the input graph [20]. Node embeddings are generated by aggregating features along the graph-edges. The work in [7] proposes a one-shot object recognition model based on embeddings generated by graph convolutions and achieves competitive results on the KITTI dataset [8]. A LiDAR point-cloud is first regularized into a coarse three dimensional grid. A graph is then constructed by connecting spatially proximal voxels. Graph convolutions map the voxels to a contextualized embedding and a decoder architecture generates one object proposal from each voxel. Other notable works use GNNs to classify point-clouds [21] and to perform a semantic segmentation [22].

III. Proposed method

The model proposed by this work can be decomposed into five steps: graph construction based on the spatial distance between radar detection points, mapping the input features to a non-contextual embedding, employing graph convolutions to generate contextualized point embeddings and the generation of object proposals which are then suppressed based on prediction confidence and a predicted clutter-score. A diagram of the model architecture is found in Fig. 1.

A. Pre-processing

Formally we define a point-cloud as a set $\mathbb{P} = \{v_1, \ldots, v_n\}$ where $v_i = (x_i, m_i)$ is a point with spatial coordinates $x_i \in \mathbb{R}^2$ marked with the state vector $m_i \in \mathbb{R}^k$ representing additional point properties. The mark m may include properties measured by a sensor such as radar cross-section or embedding features generated by a neural network. A graph $G = (\mathbb{P}, E)$ is constructed with the radar detection points $v_i \in \mathbb{P}$ as vertices and edges

$$E = \left\{(i,j) | \left\|x_i - x_j\right\|_2 < r\right\}, \tag{1}$$

including self loops. In this work the radius r is set to 1 m.

To increase the density of the input point-cloud the radar measurements from the previous five radar frames have been translated and rotated to account for the movement of the measurement vehicle and are included in the point-cloud. A categorical covariate $T \in \{0, \ldots, 5\}$ is appended to the radar detection points to indicate the age of the measurement, providing the initial mark

$$m_i = (\sigma_i, u_i, T_i, \deg(v_i)), \tag{2}$$

where σ_i denotes radar cross-section, $u_i \in \mathbb{R}^2$ denotes the measured range-rate in a Euclidean coordinate system shared across the sensors, and $\deg(v_i)$ denotes the number of edges connected to v_i serving as a proxy for point density.

B. Data augmentation

With the aim to mitigate overfitting, noise is added to the samples drawn from the training dataset. The velocity $u \in \mathbb{R}^2$ is scaled by a factor $\alpha \sim \mathrm{unif}_1(0.8, 1.2)$. The point coordinate $x \in \mathbb{R}^2$ and radar cross-section σ are translated by $\Delta x \sim \mathrm{unif}_2(-0.1, 0.1)$ and $\Delta\sigma \sim \mathrm{unif}_1(-0.04, 0.04)$ respectively.

C. Graph convolution

In the main, this work follows the graph convolution defined in [7]. Edge features are constructed as

$$e_{i,j} \leftarrow f\left(x_j - x_i, m_j\right), \tag{3}$$

which is a function of the embedding of the transmitting point v_j and the relative position $x_j - x_i$. The operation is translationally invariant against global shift in the spatial coordinates.

A new embedding for point v_i is generated as

$$m_i \leftarrow g\left(\rho\left(\{e_{i,j} \mid j : (i,j) \in E\}\right), m_i\right) + m_i, \tag{4}$$

where $\rho(\cdot)$ denotes the max-pool function which pools the edge features $e_{i\cdot}$ directed to node i, and $g(\cdot)$ is some function which further embeds the pooled features. Note that a skip connection for the previous embedding m_i is included to

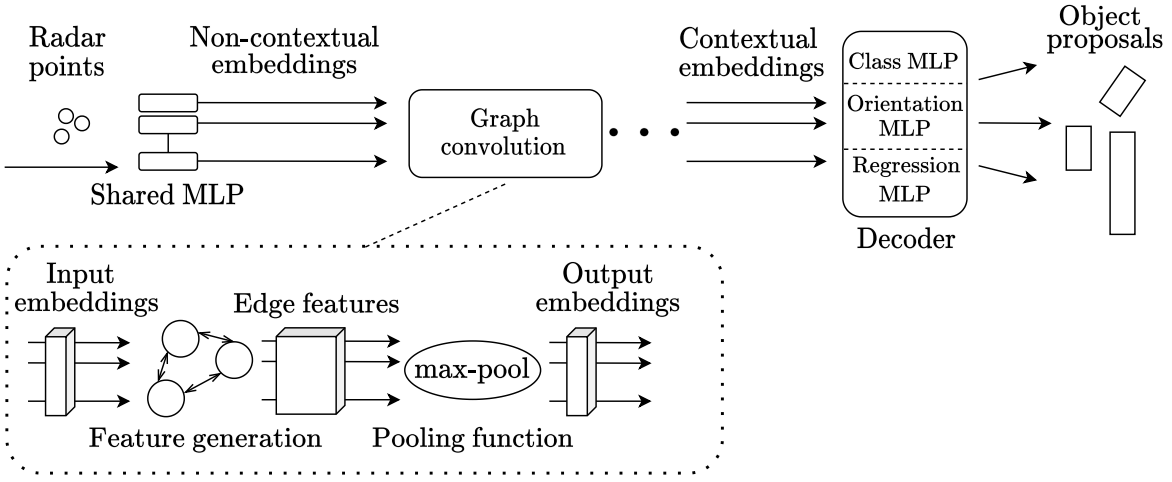

Fig. 1. An illustration of the proposed object recognition model. Note that $\cdots$ signifies that multiple graph convolutions are performed in sequence. The model generates one proposal from each radar detection point. These are then filtered based on a clutter-score threshold and non-maximum suppression.

improve gradient propagation. In this work the functions $f(\cdot)$ and $g(\cdot)$ are chosen as multi-layered perceptrons (MLPs).

In summary, a radar detection point is embedded by using a small neural network to extract features from radar detection points found within 1 m of its spatial location.

D. Network

After the pre-processing step, the point-graph is passed through a shared-MLP $f_{nc}(\cdot)$ which maps the point covariates $(\sigma_i, u_i, T_i, \deg(v_i))$ to a high dimensional feature space which is here referred to as the non-contextual embeddings. As visualized in Fig. 1 the point embeddings are then passed through a sequence of graph convolutions which extracts local patterns from the input - generating contextualized point embeddings. Lastly, a decoder consisting of three MLPs is used to generate a two dimensional object proposal from each embedded point.

An object proposal comprises a two-dimensional bounding box which approximates the physical extent of the object, a clutter-score which indicates if the object proposal is centered on clutter or on an object of interest, and a probability distribution over the object classes. The predicted bounding boxes are parametrized as:

$$\begin{aligned} \hat{h} &= \bar{h}_{class} + \delta_h, \\ \hat{w} &= \bar{w}_{class} + \delta_w, \\ \hat{x}_{center} &= x_{point} + \delta_x, \\ \hat{y}_{center} &= y_{point} + \delta_x, \end{aligned} \tag{5}$$

where $\delta_{\cdot}$ denotes the regressed scalar values output by the decoder, $\bar{h}_{class}$ denotes the median height for the predicted class and (x_{center}, y_{center}) denotes the center of the predicted bounding box. The object's velocity $\hat{u}'_i \in \mathbb{R}^+$ is predicted as squared scalar. It is assumed that the object's velocity coincides with the object's orientation.

The bounding box orientation $\hat{\phi}$ is predicted as a probability distribution over eight equisized bins. A probability distribution over the object-classes p_i and a clutter-score p'_i are also generated for each proposal.

E. Objective function

If a radar detection point is found within an annotated bounding box, the point is assigned the class label and box parameters of the annotation. All other points are assigned the *Background* class.

The network is trained on an objective function which evaluates a set of proposals in terms of how well they predict their respective class labels and how well the generated bounding boxes coincides with their respective annotation. The classification loss is defined as a sum of the cross-entropy loss for the clutter prediction task and the class prediction task,

$$\mathcal{L}_{cls} = \frac{c_{obj}}{|P|} \sum_{(p'_i, y'_i) \in P} \mathcal{L}_{ce}(p'_i, y'_i) + \frac{c_{cls}}{|B|} \sum_{(p_i, y_i) \in B} \mathcal{L}_{ce}(p_i, y_i),$$

where B is the set of points not annotated as *Background*.

A localization loss is calculated for any correct predictions using the Huber loss [23] for scalar predictions and cross-entropy for the orientation prediction defined as:

$$\mathcal{L}_{loc} = \frac{1}{|C|} \sum_{v_i \in C} \sum_{(\hat{q}, q) \in Q_i} c_q \mathcal{L}_{\text{Huber}}(\hat{q}, q) + c_\phi \mathcal{L}_{ce}(\hat{\phi}_i, \phi_i),$$

$$Q_i = \{(\hat{c}_i^{(x)}, c_i^{(x)}),\ (\hat{c}_i^{(y)}, c_i^{(y)}),\ (\hat{h}_i, h_i),\ (\hat{w}_i, w_i),\ (\hat{u}'_i, u'_i)\},$$

where C denotes the set of correct predictions.

With the aim to mitigate overfitting, L2 regularization is added to the objective function formulation calculated over all the parameters in the model as in,

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{loc} + \gamma \mathcal{L}_{reg}. \tag{6}$$

F. Optimization

The Adam optimization algorithm [24] was used to fit the model parameters. The learning rate followed a half-sinoid learning rate schedule.

$\text{iou}(\cdot)$: The intersection over union of two bounding boxes.

Input: $\mathcal{B} = \{b_1, \ldots, b_n\}$, $\mathcal{D} = \{d_1, \ldots, d_n\}$, T
$\mathcal{B}$ is the set of bounding boxes
$\mathcal{D}$ is the corresponding clutter-scores
T_{nms} is the overlap threshold value

Output: $\mathcal{M}$ is the output set of filtered bounding boxes

function NMS(B : bounding boxes, D : scores)
  $\mathcal{M} \leftarrow \{\}$
  while $B \neq \{\}$ **do**
    $i \leftarrow \text{argmin}(\mathcal{D})$
    $\mathcal{M} \leftarrow \mathcal{M} + b_i$
    for $b_j \in \mathcal{B}$ **do**
      if $\text{iou}(b_i, b_j) > T_{nms}$ **then**
        $\mathcal{B} \leftarrow \mathcal{B} - b_j$
        $\mathcal{D} \leftarrow \mathcal{D} - d_j$
  return $\mathcal{M}$

Fig. 2. The non-maximum suppression algorithm used to filter out overlapping bounding boxes.

G. Suppression

The network generates one object proposal per point in the point-graph. Any predictions with a clutter-score higher than T_{obj} are discarded. Spatially overlapping predictions are filtered by the non-maximum suppression algorithm specified in Fig. 2. Since intersecting objects are rarely found in the dataset the non-maximum suppression threshold is set as $T_{nms} = 0.01$.

IV. Experiments

In this section we define the problem setting and the model parameters used in the proposed method. The results are found in Table I.

A. Dataset

The proposed method is evaluated on the nuScenes object recognition benchmark [12]. The dataset contains annotations for 10 object-classes where the two most populated classes *Car* and *Pedestrian* with 493322 and 220194 annotations respectively comprise approximately 70% of all annotations. The annotations which are available at 2 Hz have in this work been linearly interpolated in time to acquire continuous annotations. Five FMCW radar units providing a full field of view were used to record the dataset. However, the technical specifications of the radar sensors are unavailable [12].

The dataset comprises 20 s driving sequences, 700 of which are used to fit the model and 150 are used as a validation set. The test set consists of 150 driving sequences without a publicly available ground truth and is evaluated by a third party [25]. The benchmark evaluates average precision (AP) over the classes in the dataset averaged over a selection of match distances. A number of localization metrics are also evaluated: average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE) and the average attribute error (ATE) which are formally defined in [12]. The average precision is calculated over recall and precision values greater than 10%.

1) Performance bound: Under the assumption that it is not possible to identify an object which has not generated a radar detection point we can construct an upper bound on the AP. Also, since the model predicts the object orientation as one of eight discrete values one can also construct a lower bound for the average orientation error. These bounds are calculated on the validation set, presented in Table I and are valid for both the PointNet and the R-PointGNN model.

B. Implementation details

In this work the non-contextualized embeddings are generated by an MLP with layer sizes $(4, 32, 64, 128, 512)$ separated by batch normalization and ReLU activation functions. The remaining MLPs used in this work have a hidden size of 512, have three layers and are separated by batch normalization and ReLU activation functions. Eight graph convolutions as defined in (3), (4) were used in the network.

Currently there are no other submissions to the nuScenes benchmark using only radar data. Therefore the presented method is benchmarked against a model where the graph convolutions have been replaced by a PointNet++ encoder. The MLPs used in the PointNet++ encoder have a depth of 3 and a hidden size of 512 and pools information at radii $0.2\,\text{m}$, $0.5\,\text{m}$ and $1\,\text{m}$. In contrast to the original implementation [13] the subsampling procedure has been removed to account for the sparseness of a radar point-cloud. Besides the feature extraction, the two models follow identical object recognition pipelines.

The model was fit using a base learning rate of 2×10^{-5} with loss constants $c_{obj} = 1, c_{cls} = 2$, and $c_q = 0.1, q \in Q$ over 50 epochs using early stopping based on the epoch loss evaluated on the validation set.

C. Results

The proposed method has been submitted to the nuScenes object recognition benchmark [12] evaluated by [25]. In Table I, II the proposed method is compared against the benchmark model, a performance upper bound and a state of the art LiDAR model.

Among the object recognition models which take radar data as input, the proposed Radar-PointGNN has the highest performance achieving an average precision of 10% for the *Car* class. A visualization of objects recognized by the Radar-PointGNN model on unseen data is found in Fig. 3.

Generally, the object recognition models based only on radar data perform poorly. The classification of many classes does not exceed the 10% recall threshold needed to calculate a meaningful average-precision in accordance with the nuScenes benchmark [12]. As indicated by the performance bound, a significant portion of the annotated objects have not provided strong enough reflections to generate a radar detection point which makes it difficult for the recognition model to reach the

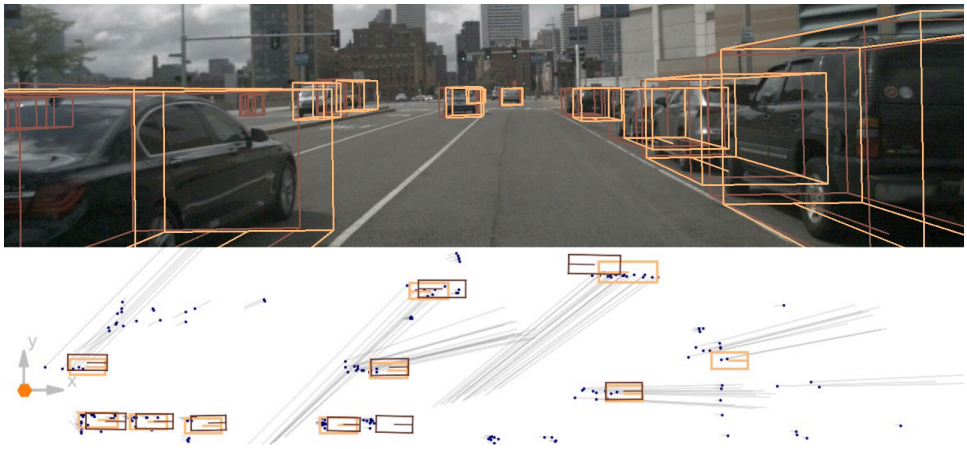

Fig. 3. Objects recognized are visualized with annotated objects, radar detection points and the measurement vehicle. The measured range-rate of a radar detection point is visualized with the vector with length proportional to the magnitude of the velocity.

TABLE I
THE PERFORMANCE METRICS EVALUATED ON THE NUSCENES TEST SET [1]

		True positive metrics			
	AP%	ATE	ASE	AOE	AVE
R-PointGNN (Ours)					
Car	**10.1**	**0.69**	0.20	**0.38**	**0.95**
PointNet					
Car	5.2	1.11	0.20	0.72	1.16
Performance bound [2]					
Car	58.0	-	-	0.01	-
Pedestrian	21.0	-	-	0.13	-
Barrier	29.0	-	-	0.14	-
Truck	69.0	-	-	0.29	-

[1] Class metrics with an AP lower than 1% are omitted.
[2] AP bound is calculated by observing the ratio of annotated objects that have generated any radar detection..

TABLE II
THE PERFORMANCE METRICS FOR A STATE OF THE ART LIDAR MODEL [1]

		True positive metrics			
	AP%	ATE	ASE	AOE	AVE
MEGVII					
Car	81.1	0.18	0.16	0.10	0.19
Pedestrian	80.1	0.14	0.30	0.42	0.22
Truck	48.5	0.36	0.19	0.08	0.23
Barrier	66.0	0.24	0.26	0.03	-

[1] Only a selection of the object-classes evaluated on the nuScenes test set are displayed for the MEGVII [26] model.

10% recall threshold for some classes. However, the model also performs poorly on uncommon classes like *Truck* which have a high performance bound.

Annotations and predictions are matched based on the distance of the bounding boxes' center position. In accordance with the nuScenes benchmark the AP metric is averaged over four maximum match-distances. The performance of the Radar-PointGNN model at different match distances is visualized in Fig. 4. The model shows poor performance on low maximum match-distances indicating that increasing the model's efficacy on the localization task would significantly increase the AP metric.

The performance of the state of the art LiDAR object recognition model MEGVII is shown in Table II. The radar based object recognition models are not yet competitive with the LiDAR based object recognition pipeline.

V. CONCLUSION

This paper presents an viable end-to-end deep learning object recognition pipeline for radar point-cloud data. A graph representation is used to define a parametrized convolutional operation which maps the radar points to a contextualized representation capable of generating object proposals with more efficacy than other point-cloud encoders. However, the presented method is not competitive with LiDAR based object recognition models, in part as a consequence of the sparsity of the point-cloud generated by automotive radars in comparison to those generated by LiDAR sensors.

Nevertheless, this work brings object recognition methods for radar in line with mature methodologies from the LiDAR and camera domain. The trend of automotive radar technology is moving in the direction of an increasing number of channels and an utilization of broader frequency bands. These changes entail an increased density of the produced radar detection point-cloud more similar to point-clouds generated by a LiDAR sensor. One may speculate that the method here presented would benefit from such changes as similar methods show strong results in the LiDAR domain.

The presented model has been evaluated on the nuScenes

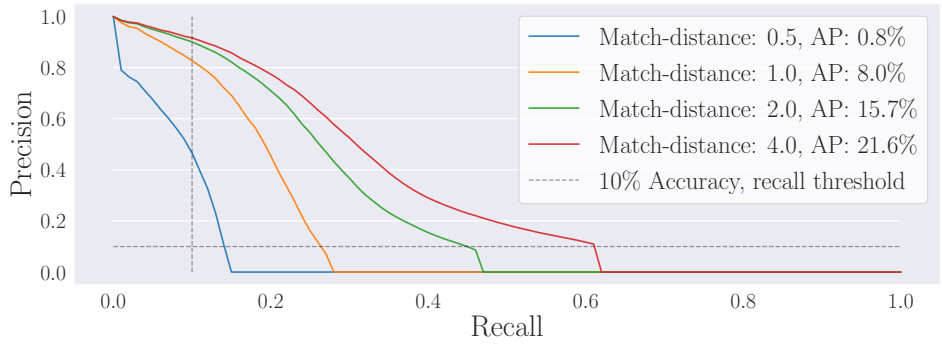

Fig. 4. The precision-recall curves for the *Car* class as predicted by the Radar-PointGNN model on the validation set. Note that the performance increases with the match-distance, indicating a poor localization of the predictions.

benchmark and is the first radar object recognition model evaluated on a public dataset. To gain a stronger understanding of the performance of this model it would be useful to evaluate conventional clustering-based object recognition pipelines on the nuScenes dataset. One may also consider to include low level sensor data as covariates to the presented object recognition model. These are topics for future work.

Acknowledgment

I wish to acknowledge the technical assistance and guidance provided by Samuel Scheidegger over the course of this project. I also thank nuTonomy for enabling a large body of research by making the nuScenes dataset publicly available.

References

[1] O. Schumann, C. Wöhler, M. Hahn, and J. Dickmann, "Comparison of random forest and long short-term memory network performances in classification tasks using radar," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, 2017, pp. 1–6.

[2] R. Prophet, M. Hoffmann, M. Vossiek, C. Sturm, A. Ossowska, W. Malik, and U. Lübbert, "Pedestrian classification with a 79 ghz automotive radar sensor," in *2018 19th International Radar Symposium (IRS)*. IEEE, 2018, pp. 1–6.

[3] N. Scheiner, N. Appenrodt, J. Dickmann, and B. Sick, "Radar-based feature design and multiclass classification for road user recognition," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 779–786.

[4] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[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)*. IEEE, 2019, pp. 61–66.

[7] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1711–1719.

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

[9] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[10] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3d graph neural networks for rgbd semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5199–5208.

[11] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*, 2011.

[12] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.

[13] 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*, 2017, pp. 5099–5108.

[14] F. Roos, D. Kellner, J. Dickmann, and C. Waldschmidt, "Reliable orientation estimation of vehicles in high-resolution radar images," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 9, pp. 2986–2993, 2016.

[15] J. Schlichenmaier, N. Selvaraj, M. Stolz, and C. Waldschmidt, "Template matching for radar-based orientation and position estimation in automotive scenarios," in *2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. IEEE, 2017, pp. 95–98.

[16] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Semantic segmentation on radar point clouds," in *2018 21st International Conference on Information Fusion (FUSION)*. IEEE, 2018, pp. 2179–2186.

[17] A. Palffy, J. Dong, J. F. Kooij, and D. M. Gavrila, "Cnn based road user detection using the 3d radar cube," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1263–1270, 2020.

[18] R. Pérez, F. Schubert, R. Rasshofer, and E. Biebl, "Deep learning radar object detection and classification for urban automotive scenarios," in *2019 Kleinheubach Conference*. IEEE, 2019, pp. 1–4.

[19] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[21] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[22] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4558–4567.

[23] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] EvalAI, "Evaluation systems for ai agents," Accessed Sept. 01 2020. [Online]. Available: https://evalai.cloudcv.org

[26] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3d object detection," *arXiv preprint arXiv:1908.09492*, 2019.