Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
©2023 ESREL2023 Organizers. *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P261-cd

Complementary object detection: Improving reliability of object candidates using redundant detection approaches

Waldemar Boschmann

Chair of Dynamics and Control, University of Duisburg-Essen, Duisburg, Germany
E-mail: waldemar.boschmann@uni-due.de

Fateme Bakhshande

Chair of Dynamics and Control, University of Duisburg-Essen, Duisburg, Germany
E-mail: fateme.bakhshande@uni-due.de

Dirk Söffker

Chair of Dynamics and Control, University of Duisburg-Essen, Duisburg, Germany
E-mail: soeffker@uni-due.de

In actual advanced technical applications, like autonomous driving, Machine Learning is utilized. Most of these methods work well in certain and/or trained situations but can fail in unknown or uncertain situations. Therefore, overreliance might lead to safety-critical situations. Detecting objects appears as a key task for the safe operation of automated systems, like autonomous vehicles. To address potential failures of an object detection system, different redundant approaches can be used. Recent research aims for fusion, combining different modalities and architectures to utilize their advantages. It can be assumed that a combination of diverse approaches compensates each other's drawbacks and leads to improved reliability and robustness of the final prediction. In this contribution the fusion of detections of multiple detection systems at a detection level is studied using different opinion pooling strategies. The predicted detection score is calibrated using the true positive rate at a score level. This results in a standardized score over different detection approaches. Afterwards detection candidates of different approaches are associated and a new detection candidate is generated in a fusion stage. Therefore missed or false positive detections of one apporach can be compensated based on a redundent set of predicted object candidates. The aim is to highlight certain detections and to reduce the detection score of false positive detections. The fused approach generates a 3 % improvement in comparison to the best individual results of single approaches, additionally improved robustness is achieved.

Keywords: Object detection, decision fusion, redundent systems, opinion pooling, autonomous driving, robust object detection.

1. Introduction

In modern technical applications such as autonomous driving, learning-based methods have become increasingly popular due to their ability in providing valuable information for decision-making. However, the reliability of these approaches can be compromised in uncertain or unknown situations, which may lead to safety-critical outcomes. To address the potential underperformance of one approach, a set of diverse and redundant approaches can be used. The predictions of multiple approaches can be analyzed, associated, and fused to provide a final result. This paper investigates the fusion of multiple object detection approaches to improve reliability and robustness in safety-critical applications. The study involves generating detection candidates on the nuScenes dataset [Caesar et al. (2020)] using multiple detection methods and applying a k-fold cross-validation scheme. An opinion pooling-based fusion strategy is employed based on the prior association between object candidates from different approaches. The results of different configurations are compared to a baseline. The contribution of this study is structured as follows. In section 2, an overview of object detection, object detection fusion, and situational dependencies is presented. In section 3, the applied approach is introduced, including calibration of the initial detections, followed by association, building the opinion profile, and finally fusing the detections. In section 4, the obtained results are discussed. Finally, a summary, conclusion, and outlook are presented.

2. Related Work

In this section, common approaches for object detection are discussed, focusing on the differences between different modalities. Furthermore, the concept of decision fusion in object detection and situational variation are reviewed and examples of early, middle, and late fusion methods are provided.

2.1. *Object detection*

In recent years, object detection using image-based approaches has gained significant research interest, especially for classification tasks. State-of-the-art techniques are divided into two main categories, one-stage and two-stage. One-stage systems , compare Redmon et al. (2016) and Liu et al. (2016), directly learn detection from the entire image and can be trained end-to-end. Although this results in a simple architecture and low inference time, they tend to perform worse than two-stage approaches. Two-stage approaches rely on region proposals, which can either be pre-computed, f.e. Girshick (2015), or obtained from a region proposal network, Ren et al. (2015). The region proposal network can operate on a feature map obtained from the image or on additional sensor data, as presented in Nabati and Qi (2019).

However, image-based approaches are known to be sensitive to lighting conditions and texture variations of the objects being detected. Under challenging conditions, such as varying brightness, noise, or objects with poor texture like repeating patterns or uniform surfaces, their performance drops significantly. Additionally, accurate estimation of 3D positions from 2D images has its own challenges.

LiDAR-based techniques can achieve high performance of 3D object detection due to the available depth information. The availability of information depends on distance, object size, or shape, affecting the detectability of objects. However, LiDAR data only provide sparse depth information, and textural information is not available. To overcome these limitations, recent approaches transform the point cloud data into voxel or pillar representations, resulting in a dense structure. Features are extracted voxel- or pillar-wise and passed to a region proposal network for the final detection stage [Zhou and Tuzel (2018); Yan et al. (2018); Lang et al. (2019); Yin et al. (2021)].

2.2. *Object detection and decision fusion*

Fusion aims to combine existing advantages by utilizing complementary aspects of available signals, information, or decisions. Fusion in the context of object detection can be classified as early fusion, middle fusion, or late fusion, Feng et al. (2019). Late fusion of predicted candidates can be denoted as decision fusion. Early fusion can be performed by projections or augmentation of the available sensor data. For example, Corral-Soto and Liu (2020) projected a LiDAR point cloud into a front view depth map and used it as an overlay for camera images. In Vora et al. (2020) camera images are used to augment a point cloud by adding pixel color as additional information. Middle fusion can be performed in various ways, examples are: Nabati and Qi (2019) proposed the generation of region proposals based on radar points and evaluated the generated proposals with an image-based detection system like Fast R-CNN. Feng et al. (2020) generated separate features for images and point clouds. Proposals are generated in 3D through the LiDAR network. Features from LiDAR and image domain are presented to a fusion network performing scoring and bounding box regression using proposals and region of interest features. Proposals from radar or image domain are suggested for practical applications. Late fusion aims for the fusion of high level features or preliminary predictions. For example, in Qi et al. (2018) a pre-trained model is used to predict 2D region proposals on image data. Each proposal transformed into a frustum, limiting the search space for a LiDAR-based detection pipeline. Despite the improvement, this method requires detections in both domains, camera and

LiDAR. In Pang et al. (2020) 2D and 3D predictions are fused by a lightweight fusion network. Fusion was performed based on the predicted score, intersection in the 2D image plane, and distance in 3D. The fusion result was a new score map. It can be assumed that the used model can learn distance dependencies. Further influences are not represented. Decision fusion is commonly applied using different opinion pooling strategies. In Dietrich and List (2016) different strategies like linear, geometric, and multiplicative opinion pooling are discussed.

2.3. *Situational variation in object detection*

Real-world applications are facing a diverse set of situations. Situations are defined by environmental influences introducing uncertainties. These environmental influences are represented in the sensor data, as well as the obtained predictions of single or multiple detection systems. Influences like weather or illumination of the scene can affect the whole detection range. In Corral-Soto and Liu (2020) complementary sensors are analyzed considering the advantages and disadvantages and compared with an early fusion approach. The results are shown over different artificial darkness levels and distance rings. Clear dependencies can be observed and are quantified as average precision values. In Yin et al. (2021) decreasing performance of anchor-based approaches during dynamic situations like turning maneuvers, is reported. In the case of redundant systems, multiple predictions for potential objects can be available. Additional information can be obtained based on agreement or conflicts in the available predictions. In Redmon et al. (2016) it is shown that the combination of diverse detection approaches can lead to improved overall performance.

It can be concluded that indicators for detection performance can be diverse. Autonomous systems need to be aware of varying uncertainties introduced by situational variations.

3. New object candidate fusion approach

3.1. *Detection pipeline*

The initial object candidates are produced using LiDAR-based detection approaches. In this contribution, one modality is used, but theoretically, multimodal approaches can be fused in the same manner. Using the implementation of OpenMMLab (2020) four different configurations are defined. The configuration is based on PointPillar introduced in Lang et al. (2019) and CenterPoint proposed in Yin et al. (2021) using different voxel and pillar grids. The approaches are trained on the nuScenes dataset [Caesar et al. (2020)]. The standard split is discarded and a 5-fold cross-validation scheme is used. Training is performed using all available classes on 3 folds. The remaining two folds are used for calibration and validation. To obtain the initial set of object candidates, 10 iterations are trained. The baseline is obtained from the best and worst performance of the available approaches, as shown in tab. 1.

3.2. *Calibration of detection score*

Due to the different detection pipelines and varying performance based on different classes, the predicted detection score can show different characteristics. Furthermore, the detection score does not correlate with a transparent metric or value. Therefore, the predicted detection score is calibrated as

$$tpr = f_{cal}(score), \tag{1}$$

mapping the score to the correlated true-positive-rate tpr using the calibration function f_{cal}. Similar to the tpr the detection-rate dr and miss-rate mr are calculated as

$$dr = f_{dr}(dist), \tag{2}$$

and

$$mr = 1 - f_{dr}(dist), \tag{3}$$

estimating the dr and mr based on a particular distance using the estimation function f_{dr}. The functions f_{cal} and f_{dr} are obtained from a curve fitting process. For the fitting linear, sigmoid and logarithmic models are defined. Underpopulated

Table 1. Single approach performance overview and baseline definition.

	↑Average Precision [%]					↑True Positive scores [%]					
	0.5	1.0	2.0	4.0	mAP	trans	scale	orient	vel	attr	mTP
	Car										
CP_V_01	74.83	83.73	86.47	87.67	83.17	82.14	84.80	87.39	74.94	**85.50**	82.96
CP_V_0075	75.20	83.45	86.08	87.28	83.00	**82.59**	84.83	**87.49**	**75.93**	85.28	**83.22**
CP_P_02	72.12	81.27	84.26	85.73	80.84	81.49	84.13	78.60	70.26	83.20	79.53
PP	68.26	81.50	85.75	87.47	80.74	78.27	84.22	80.41	75.91	84.31	80.62
BL top	75.20	83.73	86.47	87.67	83.17	82.59	84.83	87.49	75.93	85.50	83.22
BL bottom	68.26	81.27	84.26	85.73	80.74	78.27	84.13	78.60	70.26	83.20	79.53
Proposed: Best fused	**77.72**	**86.79**	**89.54**	**90.74**	**86.20**	82.53	**84.84**	86.16	75.06	84.32	82.58
Δ top [pp]	2.52	3.06	3.08	3.07	3.02	-0.06	0.01	-1.33	-0.87	-1.18	-0.64
Δ bottom [pp]	9.46	5.52	5.29	5.01	5.46	4.26	0.71	7.56	4.80	1.13	3.05
	Pedestrian										
CP_V_01	80.40	82.42	83.56	84.82	82.80	84.56	70.49	66.65	76.09	92.87	78.13
CP_V_0075	82.62	84.01	85.02	86.12	84.44	**86.34**	70.91	67.29	**77.95**	92.85	79.07
CP_P_02	74.34	76.24	77.46	79.17	76.80	83.96	70.82	70.73	75.23	94.40	79.03
PP	71.72	73.08	74.30	76.13	73.81	84.15	70.58	67.46	74.63	94.60	78.28
BL top	82.62	84.01	85.02	86.12	84.44	86.34	70.91	70.73	77.95	94.60	79.07
BL bottom	71.72	73.08	74.30	76.13	73.81	83.96	70.49	66.65	74.63	92.85	78.13
Proposed: Best fused	**82.87**	**84.23**	**85.08**	**86.13**	**84.58**	86.16	**71.03**	**77.16**	76.68	**95.87**	**81.38**
Δ top [pp]	0.25	0.23	0.05	0.01	0.13	-0.17	0.12	6.43	-1.27	1.27	2.31
Δ bottom [pp]	11.15	11.15	10.78	10.00	10.77	2.20	0.54	10.51	2.05	3.02	3.25

Interpretation: Left to right: Average Precision with distance threshold in [m]; mAP denotes mean Average Precision; True Positive scores based on true positive detections at distance threshold 2.0 m; translation score, scale score, orientation score, velocity score, attribute score; mTP denotes mean True Positive scores. Details on calculation can be found in Caesar et al. (2020); CP_V denotes CenterPoint_Voxel, CP_P denotes CenterPoint_Pillar, PP denotes PointPillar, BL denotes baseline; Best performance in bold

sections are up-sampled to ensure uniform distribution. This is relevant to avoid a bias towards a specific section during the fitting process. The results are evaluated using R^2. The fit with the highest R^2 value is used for f_{cal} and f_{dr}.

3.3. *Association of detection candidates*

Detections generated according to sec. 3.1 are associated using different distance measures similar to the association used in Wang et al. (2021). Distance d is calculated based on euclidean distance L^2

$$d_{L^2} = \frac{||dist||_2}{dist\_threshold}, \quad (4)$$

with $dist$ being the center distance between the target object and the ego vehicle, intersection over unit (IoU)

$$d_{Iou} = 1 - IoU, \quad (5)$$

with $IoU = \frac{|A \cap B|}{|A \cup B|}$, and generalized IoU (GIoU)

$$d_{GIoU} = 1 - GIoU, \quad (6)$$

with $GIoU = IoU - \frac{|C \backslash (A \cup B)|}{|C|}$ and C as smallest enclosing convex object [Rezatofighi et al. (2019)]. Based on the distance, Hungarian algorithm for bipartite matching between detections of two detection pipelines is performed. The output provides matched and unmatched detections. This is repeated for all available combinations of detection pipelines. Based on the obtained matches, instances are defined. Instances represent a cluster of detections referring to the same physical object. Each detection in an instance is from a unique detection pipeline. Each detection can only be assigned to one instance. Therefore, in this example, an instance consists of a maximum of four detections.

3.4. *Fusion of detection candidates*

For the fusion of detection candidates, an opinion pooling strategy is applied. The related opinion profile is build based on the obtained instance. If a detection of a detection pipeline is represented, the probability P is assumed to be

$$P_i = tpr = f_{cal}(score). \tag{7}$$

If a detection of a detection pipeline is not represented, P is assumed to be

$$P_i = mr = 1 - f_{dr}(dist), \tag{8}$$

with the distance as situational parameter. The distance of the missing detection is estimated by the average distance of available detections within the instance. Furthermore, a weight is calculated for each opinion by using the match information. Each match contains the two detections A and B and a normalized distance measure d in the range $[0, 1]$. The weight is calculated by

$$w_i = 0.1 + \sum\nolimits_{n=1}^{N} \frac{tpr(A_n) + tpr(B_n)}{2}(1 - d_n), \tag{9}$$

where N is the number of matches containing a detection related to detection pipeline i. The weight of missing and single detections is defined to be 0.1 since a 0.9 threshold is applied in the association step. Therefore, the weight for instances with only one detection is uniform. The weights w_i are not normalized. Since it is not intended to estimate a new detection, a particular detection is selected based on a selection rule. Afterward, a new detection score is estimated based on a pooling rule. The selection of the final detection box is performed by either selecting according to the highest score or by the highest weight. If two detections are available, the weight is identical and selection is performed by the highest score. If only one detection is available, the selection is obsolete. Following Dietrich and List (2016) the pooling will be performed as average pooling

$$P_A = \frac{1}{m} \sum\nolimits_{i=1}^{m} P_i, \tag{10}$$

linear pooling

$$P_A = \sum\nolimits_{i=1}^{m} w_i * P_i, \tag{11}$$

or geometric pooling

$$P_A = \sum\nolimits_{i=1}^{m} P_i^{\frac{1}{w_i}}, \tag{12}$$

where m is the number of opinions in the profile.

3.5. *Validation Strategy*

Evaluation of the results is performed using the classes ’car’ and ’pedestrian’. After the fusion the final predictions are evaluated using the ground truth information. Average precision is calculated using different distance thresholds and different true-positive metrics are calculated at distance threshold 2 m following the nuScenes evaluation scheme. The evaluation metrics are obtained for all 10 iterations. Reported metrics are always the mean of the 10 individual iterations. The performance difference is calculated by subtracting the baseline from the fused result. Performance difference is reported in percentage points (pp). The best result is identified by averaging mAP and mTP over all considered classes.

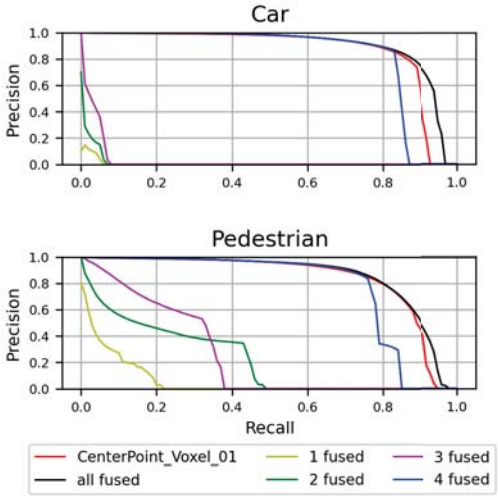

Fig. 1. Average precision for classes ’car’ and ’pedestrian’ at distance threshold 2 m; CenterPoint_Voxel_01 is shown as reference; The n value in ”n fused” denotes the number of fused predictions

4. Application and Results

The introduced fusion strategy is applied using 36 different configurations. To improve the read-

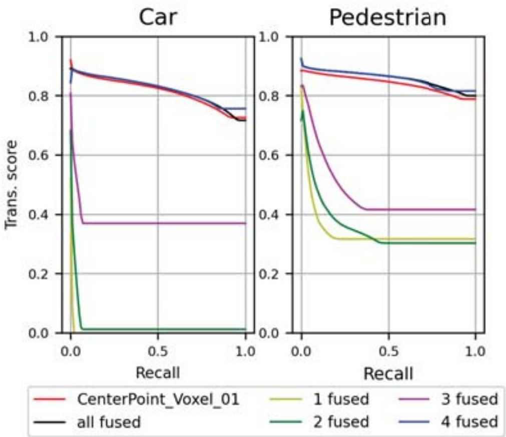

Fig. 2. Translation score for classes 'car' and 'pedestrian' at distance threshold 2 m; CenterPoint_Voxel_01 is shown as reference; The n value in "n fused" denotes the number of fused predictions

ability of the results, only the difference to the 'baseline top' is shown. The results for the average precision are shown in fig. 3. The results for the true-positive scores are shown in fig. 4. In tab. 1 the best configuration for object candidate fusion is compared to the baseline. The 'Best fused' result is obtained using IoU distance, linear pooling, and selection based on highest weight. It can be seen that mAP is increased for 3.02 pp for class 'car' and mTP is increased for 2.32 pp for class 'pedestrian' compared to baseline top. In the case of the detection of pedestrians, no significant improvement in mAP can be achieved. The reason can be the worse association ability of smaller objects. In fig. 1 average precision curves are shown for detections with four or fewer detections separately. It can be seen that class 'pedestrian' has a high ratio of fused detections based on less than four predictions compared to class 'car'. While the TP score for the translation is not improving on average, it can be seen that fused detections based on four detections still show a better performance than the reference (fig. 2). Overall, it can be concluded that fusing multiple detections helps to improve the overall performance. Fusion of 'pedestrian' detections remains difficult, but for cases with successful association improvement for mAP and mTP can be achieved. Furthermore, the fused results are at least equal to the average precision of the best available individual detection systems. It can be assumed that the fused results compensate for the performance deficits of single approaches of the available detection systems, without explicitly choosing the ones. In this sense, a robustness enhancement can be obtained.

5. Summary and Conclusion

In this contribution, an opinion pooling-based fusion strategy for 3D object detection has been presented. Different configurations have been tested and compared on different object classes. An improvement considering different metrics was achieved and improved robustness is obtained. The results among the tested configurations have shown that IoU leads to better associations than the other distance measures. The different opinion pooling rules seem to perform similarly, with only slight performance changes. An improvement can be achieved if box selection is performed based on the highest weight, calculated using distance and tpr of associated detections. This can be explained due to the better representation of unity among different detections. Miss classifications at the detection stage have not been considered, but might have a relevant impact on the results. Future work will focus on the quantification of the actual reliability of detections obtained from a standalone approach or fusion-based approaches based on situational variables.

Acknowledgement

We acknowledge support by the European Regional Development Fund (ERDF), grant-no. EFRE-0801714.

References

Caesar, H., V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom (2020). nuscenes: A multimodal dataset for autonomous driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11618–11628.

Corral-Soto, E. R. and B. Liu (2020). Understanding strengths and weaknesses of complementary sensor modalities in early fusion for

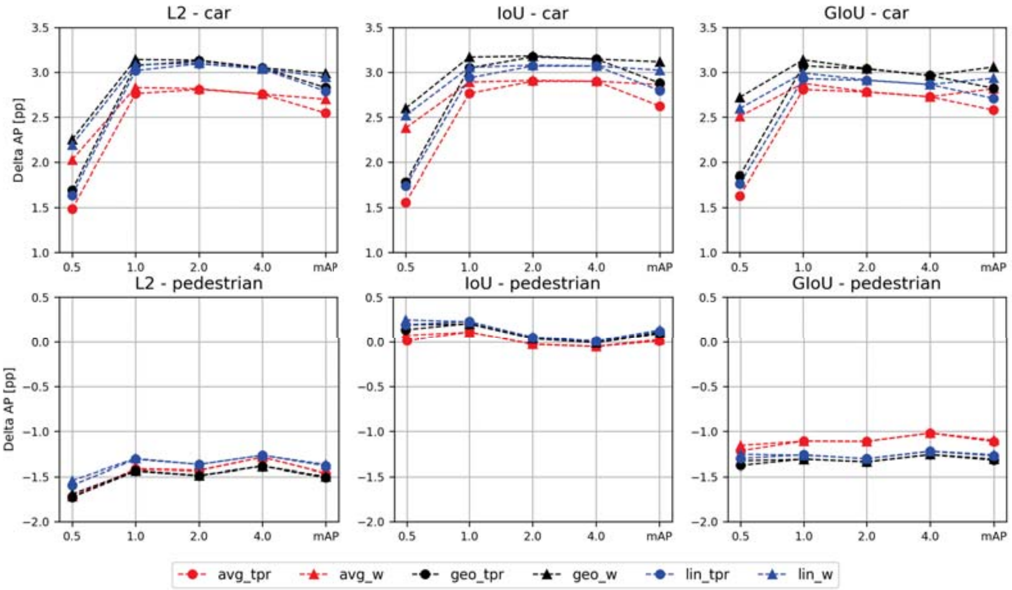

Fig. 3. Average precision for different configurations and classes. Legend should be read as follows: 'avg' denotes average pooling; 'geo' denotes geometric pooling; 'lin' denoted linear pooling; 'tpr' denoted selection by highest tpr; 'w' denotes selection by highest weight

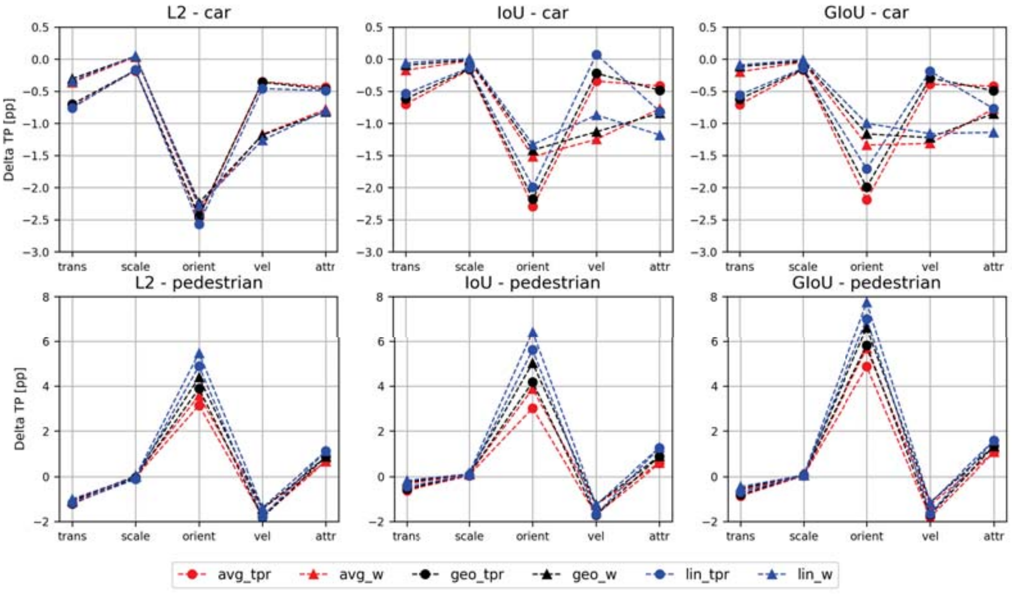

Fig. 4. True-positive metrics for different configurations and classes. Legend should be read as follows: 'avg' denotes average pooling; 'geo' denotes geometric pooling; 'lin' denoted linear pooling; 'tpr' denoted selection by highest tpr ; 'w' denotes selection by highest weight

object detection. *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1785–1792.

Dietrich, F. and C. List (2016). Probabilistic opinion pooling. In A. Hajek and C. Hitchcock (Eds.), *Oxford Handbook of Philosophy and Probability*. Oxford: Oxford University Press.

Feng, D., Y. Cao, L. Rosenbaum, F. Timm, and K. C. J. Dietmayer (2020). Leveraging uncertainties for deep multi-modal object detection in autonomous driving. *2020 IEEE Intelligent Vehicles Symposium (IV)*, 877–884.

Feng, D., C. Haase-Schuetz, L. Rosenbaum, H. Hertlein, F. Duffhauss, C. Gläser, W. Wiesbeck, and K. C. J. Dietmayer (2019). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems 22*, 1341–1360.

Girshick, R. B. (2015). Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1440–1448.

Lang, A. H., S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom (2019). Pointpillars: Fast encoders for object detection from point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12689–12697.

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg (2016). Ssd: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, and M. Welling (Eds.), *Computer Vision – ECCV 2016*, Cham, pp. 21–37. Springer International Publishing.

Nabati, R. and H. Qi (2019). Rrpn: Radar region proposal network for object detection in autonomous vehicles. *2019 IEEE International Conference on Image Processing (ICIP)*, 3093–3097.

OpenMMLab (2020). MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`.

Pang, S., D. D. Morris, and H. Radha (2020). Clocs: Camera-lidar object candidates fusion for 3d object detection. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10386–10393.

Qi, C., W. Liu, C. Wu, H. Su, and L. J. Guibas (2018). Frustum pointnets for 3d object detection from rgb-d data. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 918–927.

Redmon, J., S. K. Divvala, R. B. Girshick, and A. Farhadi (2016). You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

Ren, S., K. He, R. B. Girshick, and J. Sun (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence 39*, 1137–1149.

Rezatofighi, H., N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 658–666.

Vora, S., A. H. Lang, B. Helou, and O. Beijbom (2020). Pointpainting: Sequential fusion for 3d object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4603–4611.

Wang, Q., Y. Chen, Z. Pang, N. Wang, and Z. Zhang (2021). Immortal tracker: Tracklet never dies. *ArXiv abs/2111.13672*.

Yan, Y., Y. Mao, and B. Li (2018). Second: Sparsely embedded convolutional detection. *Sensors 18*(10).

Yin, T., X. Zhou, and P. Krähenbühl (2021). Center-based 3d object detection and tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11779–11788.

Zhou, Y. and O. Tuzel (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4490–4499.