Domain-centric ADAS datasets

Václav Diviš[1,*], Tobias Schuster[2] and Marek Hrúz[3]

[1]University of West Bohemia, Sedláčkova 214, Pilsen 3 301 00, Czech Republic
[2]Siemens Technology, Software Systems & Processes, Research in Verification & Test
[3]University of West Bohemia, Faculty of Applied Sciences, Department of Cybernetics and New Technologies for the Information Society

Abstract

Since the rise of Deep Learning methods in the automotive field, multiple initiatives have been collecting datasets in order to train neural networks on different levels of autonomous driving. This requires collecting relevant data and precisely annotating objects, which should represent uniformly distributed features for each specific use case. In this paper, we analyze several large-scale autonomous driving datasets with 2D and 3D annotations in regard to their statistics of appearance and their suitability for training robust object detection neural networks. We discovered that despite spending huge effort on driving hundreds of hours in different regions of the world, merely any focus is spent on analyzing the quality of the collected data, from an operational domain perspective. The analysis of safety-relevant aspects of autonomous driving functions, in particular trajectory planning with relation to time-to-collision feature, showed that most datasets lack annotated objects at further distances and that the distributions of bounding boxes and object positions are unbalanced. We therefore propose a set of rules which help find objects or scenes with inconsistent annotation styles. Lastly, we questioned the relevance of mean Average Precision (mAP) without relation to the object size or distance.

Keywords

Advanced Driver-Assistance Systems, Trajectory Planning, Domain-centric Datasets, Object Detection, mean Average Precision

1. Introduction and Motivation

The shift from classical programming paradigms to Machine Learning-driven approaches (ML) is significant. Humans are nowadays often relying on decisions and recommendations made by Artificial Neural Networks (ANNs), however most of the applications are harmless. This is not the case of Advanced Driver-Assistance Systems (ADAS), where wrong decisions can lead to severe injuries [1]. The development of an ML-driven application within this field follows strict processes [2, 3], from acquiring data necessary for training to deploying and evaluating the models. These processes comply with functional safety standards [4, 5] but do not propose specific measures, nor concrete thresholds which the system should pass before being publicly released. Moreover, it is generally not feasible to collect the full variation of information in a stochastic environment such as public roads. This is why it is important to look deeper into the possibilities of diagnosing and analyzing unbalanced or missing information within large-scale datasets related to the operation domain, i.e. a data-centric approach [6].

A data-centric approach lays the focus on ensuring that the data clearly conveys what the AI must learn. In the case of higher levels of ADAS [7], the vehicle can control the movement in longitudinal and lateral directions, while relying predominantly on cameras, radars, and nowadays more frequently on LIDAR systems. One example of an ADAS function is the Adaptive Cruise Control (ACC). The task of ACC is to drive within the lane at a certain speed and keep a safe distance from any potential obstacle. The Time-To-Collision (TTC) for each object is calculated for the purpose of keeping a safe distance. If the TTC value decreases below a defined threshold, a braking or evasive maneuver will be initiated. Based on linear kinematic equations, where distance is equal to speed over time, the greater the speed difference between the subject-automated vehicle (ego vehicle) and the object in front of it, the shorter the TTC will be, hence the ACC needs to plan in advance. As a consequence, on highways, the ACC must incorporate objects at a further distance (on camera images objects will appear smaller) into the trajectory planning [8]. On the contrary, in cities, where the maximum speed is limited to 50km/h, the average area of an encountered object will be larger.

Motivated by these physical dependencies, we took a deeper look at the annotated objects and analyzed their statistical appearance. The examination was done in regard to the functionality of ACC, in state-of-the-art (SOTA) large-scale automotive datasets. Our contributions are as follows:

SafeAI 2023: The AAAI's Workshop on Artificial Intelligence Safety, Feb 13-14, 2023 Washington, D.C., US
*Corresponding main author.
divisvaclav@gmail.com (V. Diviš); tobias.schuster@siemens.com (T. Schuster); mhruz@ntis.zcu.cz (M. Hrúz)
https://gitlab.com/divisvaclav/ (V. Diviš)
0000-0001-9935-7824 (V. Diviš); 0000-0002-9421-8566 (M. Hrúz)

- We define a minimum safe distance $s_{min} = f(v, a_{max})$ for a variety of scenarios (ego speed v and weather dependent deceleration a_{max}). We calculate the safe distance based on German legislation, but the process can easily be adapted to any other legislation.
- We analyze the distribution of objects' bounding boxes' (BB) relative sizes, distances to ego vehicle, and positions in the datasets.
- We define a set of standardizable sanity checkers which help verify the quality of the collected data and mark ambiguously labeled data.
- We highlight the concrete missing information which is not part of the datasets and diagnose the cause.
- We propose an *automotive mean Average Precision (amAP)* metric, which is now related to the distance to the object, or relative BB size.

2. Datasets and Related Work

2.1. Automotive Datasets

As mentioned in Section 1, the prerequisite for the correct functionality of the ACC is information about object classes, sizes (used in case of overtaking or evasive maneuver), and distances of the objects to the ego vehicle. Several common automotive datasets are therefore not suitable for such a task, despite some of them providing LIDAR data, namely BDD100K [9] (no LIDAR data), CityScapes [10] (no bounding box annotations), Perl [11] (no 3D annotations) or Apollo Scape [12] (no images, only LIDAR). The other group of datasets contains images and LIDAR point clouds including the 2D as well as the 3D annotations. It is for this reason that we decided to take the following large-scale automotive datasets into account: **KITTI** [13], Audi Autonomous Driving Dataset (**A2D2**) [14], **Lyft Level 5** dataset [15], **nuScenes** dataset [16], **Waymo** Open Dataset [17] and the **ONCE** dataset [18].

Each dataset contains a various number of labeled objects like small vehicles (vehicle, ego vehicle, SUV, motorcycle, etc.), large vehicles (truck, bus, tram), pedestrians, and cyclists. Especially **KITTI** and **nuScenes** show high-class imbalance for some classes due to fine-grained classes. Furthermore, the selected datasets contain labeled camera images with 2D and 3D bounding boxes and the corresponding LIDAR point cloud information which provides distance information for each object. However, the size of the datasets, in terms of the number of labeled frames and captured ambient conditions, varies. For instance, A2D2 provides a dataset of 2D labeled images, but only a small part contains 3D bounding boxes. **KITTI**, **nuScenes**, **LyftLevel5** and **Waymo** reflect only urban areas whereas **A2D2** and **ONCE** also contain highways, country roads, tunnels etc. On top of that, the sensor setups vary as well, e.g. different camera resolutions. For the **KITTI** dataset, the authors used a LIDAR sensor and two stereo cameras (left and right), whereas for **Waymo** five LIDARs (restricted to 75m) and five cameras were used. **nuScenes** used six cameras and one LIDAR sensor as well as five radars, ONCE uses one LIDAR and seven cameras, and A2D2 five LIDARs and six cameras. However, for the ACC, only the front cameras are taken into account, resulting in a smaller amount of usable images.

The **KITTI** dataset is the smallest in terms of scenes and the least diverse, containing only sunny and cloudy daytime scenes. For a short period of time, the **Waymo** and **nuScenes** datasets provided the largest variety and amount of data and annotations; they are among the most widely-used autonomous driving datasets. Although the **ONCE** dataset recently set a new benchmark for the amount of driving hours and frames, **Waymo** contains the highest amount of 3D bounding boxes because **ONCE** focuses on self-supervised learning without labels. Table 1 gives an overview of important general information per dataset.

2.2. Dataset Analysis

The authors of the above-mentioned datasets compared their works based on the common aspects of the datasets as shown in Table 1. General properties like the number of driving hours are often used to compare datasets and to state an improvement. Moreover, the number of scenes, images, or annotations is often used to determine the quality of the datasets.

For instance, the authors of the **ONCE** [18] and **A2D2** [14] dataset focus on the number of annotations, the amount of driving hours, the adverse weather conditions, the time (day/night) and different locations (urban, highway, country roads) as well as countries/cities where the data was captured. However, specific requirements of driver assistance systems were not considered while creating or evaluating any of those datasets. The intent was rather to generate general datasets for a wide range of supervised and unsupervised learning tasks as well as driving functions.

The authors of **A2D2**, as well as **nuScenes**, focused on statistics relevant to the ACC and other assistance systems. Their work provides information about the distribution of the object distances for different classes as well as the absolute number of objects within the dataset. Additionally, the authors of **nuScenes** analyzed the distributions of the velocities of common objects like vehicles and bikes as well as bounding box dimensions.

The **KITTI** benchmark [13] is based on the performance analysis of neural networks on the size of BBs in pixels as a proxy for the distance of the ego vehicle

Table 1
Comparison of analyzed datasets. Cells with "-" indicate not mentioned in the original paper.

Dataset	Scenes	# images	# 3D Bounding Boxes	Ambient conditions	# classes	Year of release
ONCE [18]	1M	7M	417k	urban, highways, country roads; day/night; various weather	5	2021
nuScenes [16]	1k	1.4M	1.4M		23	2020
A2D2 [14]	-	-	-	urban, highways, country roads; day; various weather	14	2020
LyftLevel5 [15]	366	323k	1.3M	urban; day; various weather	9	2019
Waymo [17]	1150	1M	12M	urban; day/night; various weather	4	2019
KITTI [13]	22	15k	200k	urban; day; sunny, cloudy	8	2012

to the object. Their work in general follows the COCO evaluation methodology [19], but no physical distance information is used. The authors of **nuScenes** and **Waymo** did set a baseline for various detection tasks, yet without considering the distances to the different objects explicitly. The analysis closest to ours is done by the authors of the **ONCE** dataset. They analyzed the collected data regarding distance-wise mean Average Precision performance for 3D object detection using only point clouds. However, their distance thresholds were selected rather intuitively, whereas we specifically derive the distance from the domain safety requirements. Additionally, we analyze the spatial distribution of objects within the images as well as the bounding box/object size compared to the image size.

3. Background

Since the majority of related works only analyze datasets from a general ML perspective, omitting the point of data-centric paradigm, we decided to verify the SOTA automotive datasets in regard to a trajectory planning task. One part of our motivation is that forecasting the trajectory planning is conditioned by ego's vehicle velocity, which in case of higher value takes the further-distance objects into account. In order to be able to evaluate the sufficiency and quality of annotated objects within the datasets, we've chosen the Time-To-Collision as an instance to calculate the minimal safe distance from the ego vehicle.

For the sake of simplicity, we do not consider any obstacle heading from the opposite direction (on the collision course), since we are working with static images and thus don't have the information about the relative motion of the objects. As described in [20]: "The TTC value at instant t is defined as the time for two objects to collide if they continue at their present velocity and on the same path". Let's define the first object to be an obstacle (anything else than the ego vehicle) and the second object to be the ego vehicle. We consider the velocity of an obstacle to be equal to 0km/h (representing the stand-still object and therefore the worst-case scenario), and the ego vehicle's deceleration to be 7 m/s^2 (can variate within a range from 7 m/s^2 till 10 m/s^2 on dry roads [21],[22]). Vehicle deceleration can be seen as a function of adhesion between the tires and the road, which depends on the material used in the tires, material of the road, temperature, weather conditions, mounted braking system, and the mass of the vehicle.

Based on the definition of TTC, let us consider three driving scenarios:

- highway (recommended speed $130km/h \approx 36m/s$)
- country road (maximum speed $100km/h \approx 28m/s$)
- city (maximum speed $50km/h \approx 14m/s$)

We now compute the minimal safe distance which needs to be ensured in order to brake in time (without initiating any evasive maneuver), for the following case: ego vehicle is driving on the highway, the possible deceleration is equal to $7m/s^2$ and reaction delay is $0.0s$. Based on the kinematic equations of a linearly decelerating object, the distance which the object will travel is a function of time $s = s_0 + v_0 t - \frac{1}{2}at^2$, where time t is a function of deceleration $t = \frac{v_0 - v}{a}$. With a linear deceleration of $7m/s^2$, the vehicle, moving within the legal limits, will reach its standstill state in time $t = \frac{v_0 - v}{a} = \frac{36-0}{7} = 5.14s$. Within this time frame, the ego vehicle will travel a distance of $s = 0 + 36 \cdot 5.14 - \frac{1}{2} \cdot 7 \cdot 5.14^2 = 185.04 - 92.46 = 92.58m$. For completeness, the braking distance under the same weather conditions on country roads is $55.11m$ and in the city $14m$ as can be seen in Table 2. As mentioned earlier, this process can be generalized and repeated for

Table 2
All scenarios with calculated minimum safe distances.

Scenario	Max. allow speed [km/h]	Time to stand still [s]	Min. safe distance d_s [m]
highway	130	5.14	92.58
country road	100	4.00	55.11
city	50	2.00	14

any ambient conditions, type of vehicle, and speed limitations.

It is noticeable that the highway's maximum foresight boundary will be in reality limited by the physical properties of the camera or the maximum speed difference between the ego vehicle and the object. But most importantly, objects within those safe ranges must be part of a dataset (training and testing), otherwise, the system will deal with an epistemic uncertainty [23]. In order to be able to investigate the statistics of objects' appearances, we need a dataset that provides information about the object's distance. As mentioned in Section 2, some datasets such as Perl and BKK100, etc. are not suitable for this task. Consequently, we have chosen the following large-scale automotive datasets: **KITTI**, **Waymo** open Dataset, **A2D2** from Audi, **nuScenes**, **LyftLevel5** and **ONCE**, which contain 3D annotations and distances to the objects.

In regard to the functionality of trajectory planning, we focused on the following in-dataset object characteristics:

- distribution of a BB's relative size: in order to verify that a variety of objects' sizes is captured within the dataset,
- distribution of the distance between obstacles and the ego vehicle (with relation to minimum safe distances): in order to verify that objects in further distances are incorporated within the dataset,
- relation between BB's relative size and object distance from ego vehicle: to discover abnormality within the dependency,
- heatmap of an object's appearance density: to visualize the potential asymmetrical appearance of an object with relation to the ego vehicle,
- an optical flow between consecutive images; in order to identify a series of static images, which can lead to class imbalanced dataset.

As originally presented in [19] and further explored in [18], it seems to be reasonable to observe the mean Average Precision with relation to a specific object's size. Since the original authors clustered the object groups rather generally (small, middle, big), we propose to have clearly specified operational domain dependencies and incorporate the minimum safe distances as thresholds. Furthermore, the relation of AP to the objects' relative BB sizes or distance highlights detailed discrepancies between the model's performance trained on the same dataset. We therefore incorporate the minimal safe distance d_s of each scenario as a threshold for the creation of test subsets $\mathcal{A} \subset \mathcal{B} \subset \mathcal{C}$ from original test dataset $\mathcal{D}$. For instance $\mathcal{A}$ contains objects only in distance $> d_s(highway)$. For each subset the average precision can be calculated, representing a concrete value for a specific operational domain e.g. driving in the city. Equation 1 represents a different perspective on the average precision metric, which we call *automotive mean AP*.

$$automotive\ mAP = \frac{1}{n_c} \sum_{i=1}^{c} AP_{n_i}, \tag{1}$$

where n_c is a number of domain specific test-subsets.

4. Analysis of datasets

In this chapter we analyze all datasets from Table 1 for the characteristics mentioned in Section 3. In order to evaluate the model's generalization ability, the collected data has to be divided into two parts, namely training and validation. The training part is used for extracting the relevant features and finding a reasonable combination in order to build a high-level feature representation, whereas the validation part is used to evaluate the loss after each training cycle (epoch). The training loop usually ends when the loss, on a validation set, stagnates for several epochs [24]. Logic demands that both the training and validation parts should have uniformly distributed objects densities and their properties (e.g.size, distance). We therefore decided to evaluate the correlation between the theoretical uniform distribution and observed one by calculating the Wasserstein distance [25].

4.1. Analysis of outliers

There is no reason to assume that all data are flawlessly annotated since the task is usually done by several people, which increases the uncertainty of an inconsistent annotation style. We therefore designed plausibility check functions which allow us to indicate the potentially wrong annotated data and analyze them later on.

- $f_1 : return\,(0.0 \geq rel.bb.size > 1.0)$
- $f_2 : return\,(e^{-\frac{dist}{d} + long.cam.off})$
- $f_3 : return\ (intersec(bb_i, bb_j) > thr_{overlap})$
- $f_4 : return\,(opt.flow(img_{i-1}, img_i) > thr_{static})$

The function f_1 returns a bounding box whose relative size is out of the range $(0.0, 1.0]$. The outliers of otherwise exponentially decayed objects' size with respect to the distance to the ego are marked by function f_2. This function covers the quadratic dependency of the BB area and the mapping of the captured object within the real world to the pixel coordinates. The parameter $long.cam.off$ and the denominator d were found with the least squares optimization from the analyzed data and therefore are unique for each dataset and each class. The function f_3 highlights objects whose bounding boxes significantly overlap. The threshold of the relative intersection area can be defined by $thr_{overlap}$.

Examples of the outcomes for functions $f_1 - f_3$ are given in Figure 1. We further found that most of the datasets contain sequences of "stop and go" in a traffic jam, or idling at crossroads. These situations result in the recording of many similar images without objects or surrounding variation. Therefore we added f_4, which calculates a dense optical flow [26] from previous and actual images in the sequence and returns a positive flag in case the magnitude sinks bellow empirically defined threshold as seen in Figure 2. The higher the value of the magnitude, the more objects were moving from frame to frame. With this method, we could even identify if the data were recorded repeatedly in the same place [27] (when recorded in one session), but we only used it to discover static scenarios.

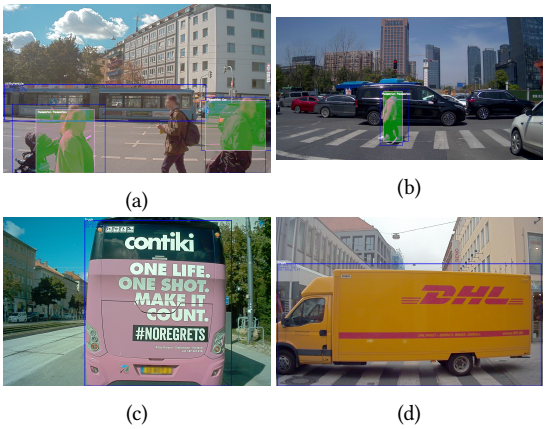

Figure 1: Examples of outliers (potentially wrongly annotated data) within the analysed datasets. Images (a, b) capture overlapping BBs, whereas images (c, d) contain objects where the relative BB size doesn't correlate with the object distance. We further encourage to analyse such a subset of filtered images and adapt the parameters of the outliers function $f_1 - f_4$ to make them suitable to the target domain.

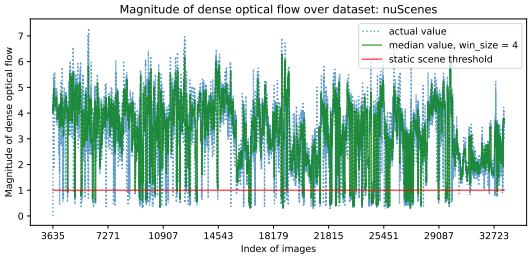

Figure 2: As visible on the figure, 2.3% of the nuScenes dataset are "stop and go" scenarios, where the magnitude of the dense optical flow drops below a static scene threshold. The threshold was recalculated for each dataset based on the image resolution.

4.2. Overall results

Exemplary results of the analysis of the objects' distances distribution as well as the relation between the relative BB size and object distance can be seen in Figures 3 and 4. Moreover, we show an example of object appearance variation (heatmap) of class *Vehicle* in the A2D2's dataset in Figure 5.

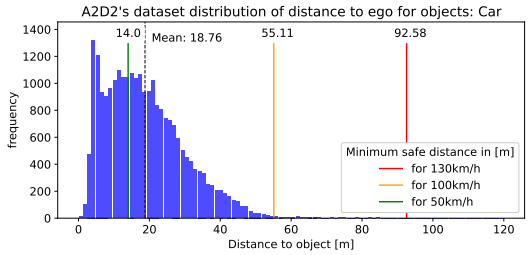

Figure 3: The information from subset (0.0 to 14.0) meters will be used in scenarios where a vehicle is driving less than 50km/h (idling on a crossroads, for instance). However, the A2D2 dataset doesn't contain any objects with the minimal safe distance necessary for highway driving (92.58m and more).

5. Concluding Remarks

To summarize, most of the SOTA autonomous driving datasets are generated with a focus on a large amount of scenes/driving hours/frames while considering different weather, location, and daylight conditions. However, as we discovered, none of the datasets contains a sufficient amount of information for safe autonomous driving on the highway (described in Section 3). Every analyzed dataset was lacking high-distance annotated objects, as can be seen in Table 3. Such a gap can be explained by the physical limitations of the camera (too low resolution),

Table 3
Comparison of the analysed datasets highlighting the best results in regard to the training of an object detector used for trajectory planning.

Research Questions	Class	ONCE	nuScenes	A2D2	LyftLevel5	Waymo	KITTI
Which dataset contains the **most** distant objects (driving $\geq 130km/h$)?	Ped.	0	6	0	**55**	0	0
	Veh.	0	**2464**	4	1574	0	0
Which datasets has the **most** uniform distribution according to the Wasserstein distance?	Ped.	0.86	0.82	0.80	**0.60**	0.65	0.89
	Veh.	0.74	0.57	0.81	**0.50**	0.53	0.70
Which datasets has the **least** outliers based on our sanity check filters (f1 - f3)?	All	212	4742	563	**146**	6935	288
Which dataset contains the **least** static images according to optical flow f4?	All	358	786	157	**59**	895	77

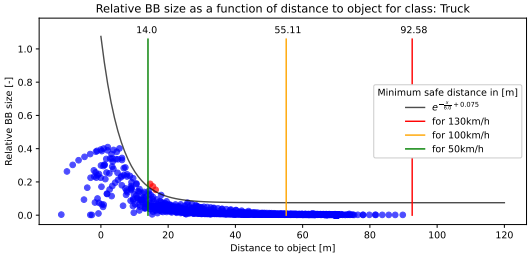

Figure 4: The black curve indicates the outliers' decision boundary of otherwise exponentially-decayed objects size with respect to the distance to the ego. Outliers can indicate rotated, or wrongly annotated (with unnecessary big margin) objects.

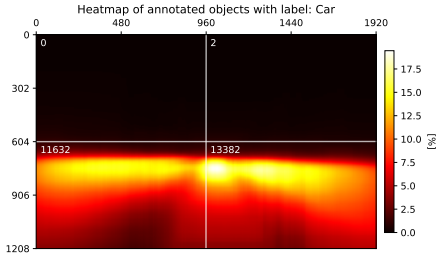

Figure 5: The number in the top left corner of each quadrant indicates the number of objects' appearances in the respective area. Contrary to the expected symmetrical heatmap, the vehicle appears with different quantities and sizes on both sides of the ego. The distribution is obviously unbalanced in the vertical and horizontal directions containing the majority of the objects in the lower half of the image. Such a statistical information can be used in post-processing by plausibility check of objects appearance.

the annotation style (objects under a certain pixel area were excluded from the annotation process), and the ambient conditions in which the dataset was recorded. In addition, a system trained on such a dataset would have to deal with epistemic uncertainty and look for additional sources of information (namely LIDAR).

Furthermore, all datasets contain predominantly small-sized objects (the highest MEAN value of the relative BB size of the class *Person* was 0.091 in the case of the **KITTI** dataset). For comparison, the same can be stated for the well-known COCO dataset [19], where the class *Person* has a MEAN relative BB size equal to 0.089. By generating heatmaps, we discovered that 99.8% of the objects appear only in the two lower quadrants of the image. Such information can lead to a significant downsizing of the field of view and the thereof acceleration of the detectors' inference time. The majority of overlapping BBs, with potentially wrong annotation styles, were extracted from a sequence of streams on crossroads. Such a static data sequence (9.55% in case of **nuScenes** dataset) contains a lot of similar features (the majority of surrounding objects are not moving) and could be removed from the dataset.

Finally, we defined and evaluated a reasonable set of rules, described in Section 4.1, which automatically proves the quality of the collected data from a domain-related perspective. We encourage the community to use our "domain-centric" approach in order to create a dataset under concrete functional constraints and train detectors on it. Our code and additional results are published on GitLab and can be publicly accessed. [1]

This work deepened our vision of a domain-centric ML approach in the automotive industry. To conclude, we outline some research directions which we are currently investigating: (a) Analysis of *automotive mAP* based on

[1] https://gitlab.com/arrk-fi/ObjectDetectionCriticality/-/tree/dependency_branch.

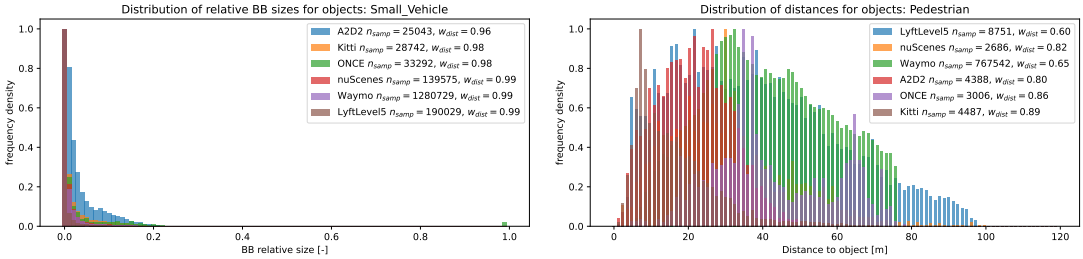

Figure 6: Left: We picked a *Small Vehicle* class where a various BB relative size density can be seen. It is visible that the majority of large objects has very small relative BB size. **Right:** We show the distance to object distributions of class pedestrian for each dataset, where n_{samp} is number of samples and w_{dist} is Wasserstein Distance between Uniform distribution $\mathcal{U}$ and the dataset distance to object distributions. The lower the value, the closer the two distributions are.

the relative BB size or distance to object with SOTA object detectors. (b) Object detector performance analysis on cleaned data (without outliers). (c) Dataset creation with respect to our domain-centric approach. (d) Combination of datasets in order to achieve a more uniform data distribution. (e) Data augmentation to compensate weak aspects in the datasets.

Acknowledgments

This work is partly funded by ARRK Engineering GmbH. The work has also been supported by the grant of the University of West Bohemia, project No. SGS-2022-017 and by the Technology Agency of the Czech Republic, project No. CK03000179.

References

[1] E. Schwalb, Analysis of safety of the intended use (sotif) (2019).

[2] W. W. Royce, Managing the development of large software systems: concepts and techniques, in: Proceedings of the 9th international conference on Software Engineering, 1987, pp. 328–338.

[3] K. Petersen, C. Wohlin, D. Baca, The waterfall model in large-scale development, in: International Conference on Product-Focused Software Process Improvement, Springer, 2009, pp. 386–400.

[4] ISO 26262:2018, Road vehicles — Functional safety (ISO 26262), Standard, International Organization for Standardization, 2018.

[5] ISO/PAS 21448:2019, Road vehicles - Safety of the intended functionality, Standard, International Organization for Standardization, 2019.

[6] A. Ng, Deep Learning AI data-centric ai competition, https://https-deeplearning-ai.github.io/data-centric-comp/, 2021. Accessed: 2022-08-16.

[7] SAE International, Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems, volume J3016, 2014.

[8] A. Gasparetto, P. Boscariol, A. Lanzutti, R. Vidoni, Path planning and trajectory planning algorithms: A general overview, Motion and operation planning of robotic systems (2015) 3–27.

[9] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, T. Darrell, BDD100K: A diverse driving video database with scalable annotation tooling, CoRR abs/1805.04687 (2018). URL: http://arxiv.org/abs/1805.04687. arXiv:1805.04687.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3213–3223.

[11] G. Pandey, J. R. McBride, R. M. Eustice, Ford campus vision and lidar data set, The International Journal of Robotics Research 30 (2011) 1543–1552.

[12] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, D. Manocha, Trafficpredict: Trajectory prediction for heterogeneous traffic-agents, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 6120–6127.

[13] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, The International Journal of Robotics Research 32 (2013) 1231–1237.

[14] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, P. Schuberth, A2d2: Audi autonomous driving dataset, 2020. URL: https://arxiv.org/abs/2004.06320. doi:10.48550/ARXIV.2004.06320.

[15] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. On-

druska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, V. Shet, Level 5 perception dataset 2020, https://level-5.global/level5/data/, 2019.

[16] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuscenes: A multimodal dataset for autonomous driving, in: CVPR, 2020, pp. 11621–11631.

[17] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al., Scalability in perception for autonomous driving: Waymo open dataset, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2446–2454.

[18] J. Mao, M. Niu, C. Jiang, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, C. Xu, et al., One million scenes for autonomous driving: Once dataset, 2021.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: ECCV, Springer, 2014, pp. 740–755.

[20] R. Van Der Horst, J. Hogema, Time-to-collision and collision avoidance systems (1993).

[21] I. für Unfallanalysen, Bremstabelle hamburg - institut für unfallanalysen - bremstabelle, https://unfallanalyse.hamburg/index.php/ifu-lexikon/bremsen/bremstabelle-a/, 2022. Accessed: 2022-09-14.

[22] A. Erd, M. Jaśkiewicz, G. Koralewski, D. Rutkowski, J. Stokłosa, Experimental research of effectiveness of brakes in passenger cars under selected conditions, in: 2018 Xi International Science-Technical Conference Automotive Safety, IEEE, 2018, pp. 1–5.

[23] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, in: NIPS, 2017, pp. 5574–5584.

[24] L. Prechelt, Early stopping-but when?, in: Neural Networks: Tricks of the trade, Springer, 1998, pp. 55–69.

[25] C. Villani, Optimal transport, old and new. notes for the 2005 saint-flour summer school, Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer 3 (2008).

[26] G. Farnebäck, Two-frame motion estimation based on polynomial expansion, in: Scandinavian conference on Image analysis, Springer, 2003, pp. 363–370.

[27] A. J. Davison, Real-time simultaneous localisation and mapping with a single camera, in: Computer Vision, IEEE International Conference on, volume 3, IEEE Computer Society, 2003, pp. 1403–1403.