Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)
16-18 November 2022, Maldives

Testing ground-truth errors in an automotive dataset for a DNN-based object detector

Boda Li[§]
WMG
University of Warwick
Coventry, United Kingdom
boda.li.1@warwick.ac.uk

Gabriele Baris[§]
Institute of Mechanical Intelligence
Sant'Anna School of Advanced Studies
Pisa, Italy
gabriele.baris@santannapisa.it

Pak Hung Chan
WMG
University of Warwick
Coventry, United Kingdom
pak.chan.1@warwick.ac.uk

Anima Rahman
WMG
University of Warwick
Coventry, United Kingdom
anima.rahman@warwick.ac.uk

Valentina Donzella
WMG
University of Warwick
Coventry, United Kingdom
v.donzella@warwick.ac.uk

***Abstract*—Given the promising advances in the field of Assisted and Automated Driving, it is expected that the roads of the future will be populated by vehicles driven by computers, partially or fully replacing human drivers. In this scenario, the first stage of the perception-decision-actuation pipeline will likely rely on Deep Neural Networks for understanding the scene around the vehicle. Typical tasks for Deep Neural Networks are object detection and instance segmentation, tasks relying on supervised learning and annotated datasets. As one can imagine, the quality of the labelled dataset strongly affects the performance of the network, and this aspect is investigated in this paper. Annotation quality should be a primary concern in safety-critical tasks, such as Assisted and Automated Driving. This work addresses and classifies some of the mistakes found in a popular automotive dataset. Moreover, some experiments with a Deep Neural Network model were performed to test the effect of these mistakes on network predictions. A set of criteria was established to support the relabelling of the testing dataset which was compared to the original dataset.**

***Index Terms*—automated driving, Deep Neural Network, dataset, environment perception, ground truth**

I. Introduction

In recent years, an increasing number of vehicles has been equipped with Advanced Driving Assistance Systems (ADASs) such as adaptive cruise control, lane keeping assist, parking assist, etc. The Society of Automotive Engineers (SAE) has defined six levels of driving automation to classify the *autonomy* a vehicle is capable of [1]. In the lower three levels (L0 to L2), the driver is in charge of most of the *dynamic driving tasks*, with an increasing number of tasks that can be delegated to the ADAS functionalities. In the higher three levels (L3 to L5), related to Automated Vehicles (AVs), the human driver is relieved by most of the driving tasks, including monitoring the environment, and L5 is full automation with unlimited *Operational Design Domain* (ODD).

Many ADAS and AV functionalities (i.e., lane centering, traffic sign recognition, etc.) strongly rely on cameras, and Deep Neural Networks (DNNs) using camera data are a promising solution (in terms of performance, flexibility, robustness, etc.) to implement these functionalities. DNNs are becoming more popular and pervasive every year, with current applications in numerous fields, from biology and medicine, to manufacturing, from automation to robotics and intelligent vehicles and drones, from structural monitoring to surveillance.

The basic idea behind neural networks is that they *learn* how to solve a problem using some training inputs, being then able to generalise on unseen data. A first categorisation of DNNs can be carried out in terms of *how* they learn, leading to *supervised* vs. *unsupervised learning*. In both cases, once the DNN has completed its learning, the weights of the neurons in the layers are frozen and the network can be deployed to evaluate unseen data. In unsupervised learning, the neural network is typically provided with a certain amount of *unlabelled* input data and it finds particular relationships and patterns in the data. Tasks such as clustering and dimensionality reduction (compression) belong to this category [4], [5]. On the other hand, in supervised learning, the network is provided with both input data and *target* values (i.e. expected outputs of the network). Tasks that fall into this category are, for example, image classification and segmentation [6], [7]. Thus, the main difference between supervised and unsupervised learning basically resides in the way in which the training data, namely the *dataset*, are prepared. While for unsupervised learning the collected data might need only some preliminary cleaning, sizing and pre-processing, in the case of supervised

[§] Authors contributed equally.

978-1-6654-7095-7/22/$31.00 © 2022 IEEE

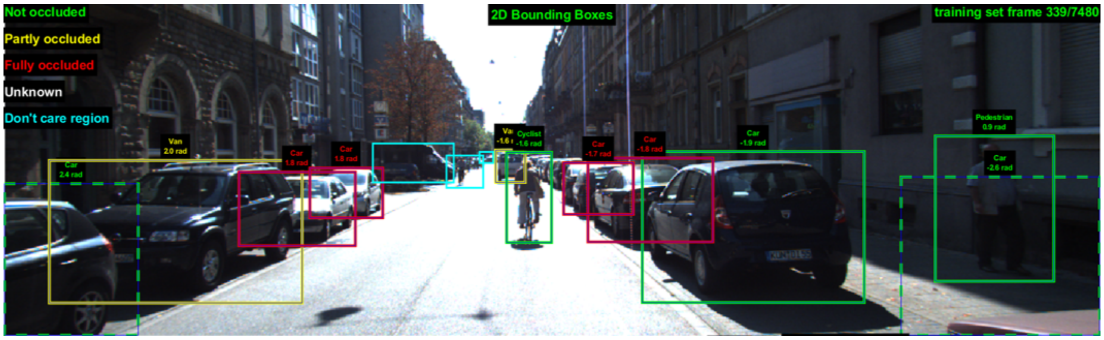


Fig. 1. Image from the KITTI dataset [2] with the ground truth bounding boxes, that can be used for DNN supervised learning; the object label is above the bounding box, the box colour indicates the the amount of occlusion.

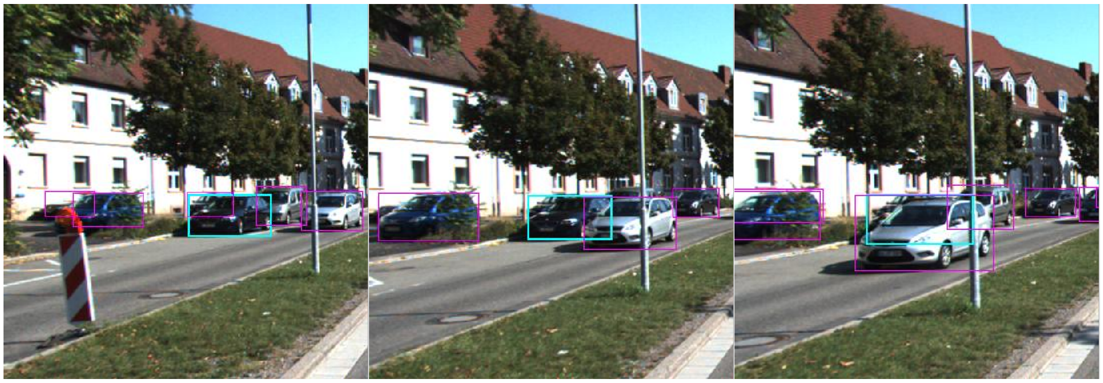


Fig. 2. Example of frames from a sequence in the KITTI MoSeg dataset [3]: the vehicle highlighted by the cyan bounding box has a box even when completely covered by the car moving on the road. Ground truth bounding boxes are in magenta.

learning, the dataset needs to undergo through a *labelling* (or annotation) step in which objects or areas of interest in the input data are recognised and associated with a class or a specific output, as shown in Fig. 1.

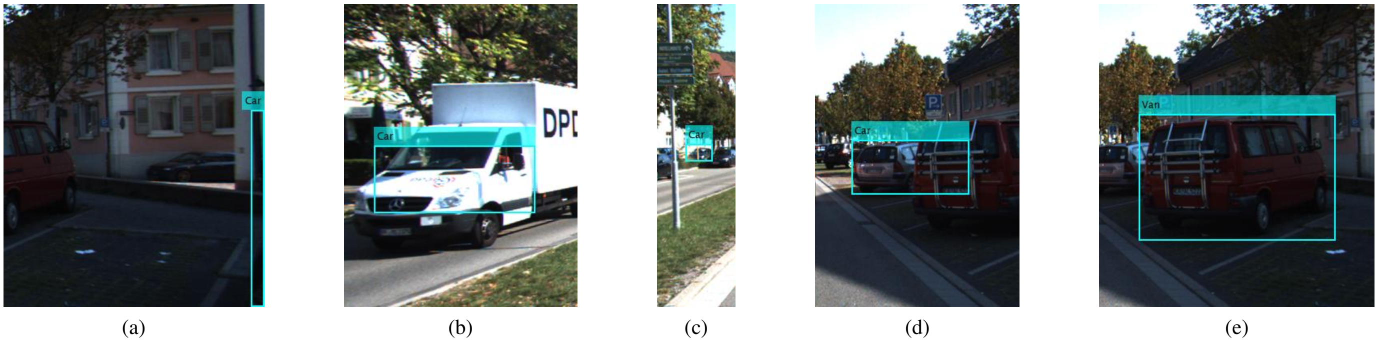
The annotation process is a vital part in the preparation of any dataset for supervised learning. The quality of the annotations strongly affects the performance of the network. This paper focuses on the quality of the bounding boxes in annotated dataset for 2D object detection using camera data. Moreover the considerations and conclusions drawn in this paper can be easily expanded to 2D and 3D object detection based on different sensors, such as LiDAR [8] and RADAR [9]. To understand the importance of labels, one can imagine to work on a image classification task with vehicles. The dataset contains the images of several types of vehicles (e.g. cars, vans, lorries, etc.). Let's assume that for the car class, the labels are not correct half of the times, namely half of the cars in the dataset has the label `car` and the other half has the `van` one. During training, the network would hardly learn the correct mapping between inputs and targets, since similar input features would be associated to two different classes. Errors in the labels may be of different types, and they will be described in this paper.

Depending on the use case, the criteria for annotating the dataset need to be properly defined. This paper is focused on the task of real-time 2D object detection for Automated Driving based on camera data. In this field, safety and reliability are a primary concern, thus an accurately annotated dataset is a crucial aspect to guarantee DNN best performance. The presented work shows the relationship between ground truth quality and DNN performance. A novel set of **relabelling criteria** is proposed, to reproducibly and consistently label automotive datasets.

This paper is structured as follows: Sec. II provides some background about dataset labelling in general and the dataset we analysed in this paper; Sec. III provides the details about the design methodology for the experiments; finally, Sec. IV provides an analysis of the experimental results.

II. Background

As mentioned in the previous section, supervised learning is based on the exploitation of big curated and labelled datasets. The quality of datasets (in terms of variability in the dataset, quantity of data, coverage of unexpected cases, etc.) and of bounding boxes and labels (in terms of correct labelling of objects, sizes of bounding boxes, correct classes, etc.) is key for the process of hyperparameter tuning (e.g. learning rate, batch size, solver, etc.). This tuning enables the creation of a trained DNN version with optimised performance for the specific dataset. This version will be then deployed in the specific system to be used to infer on new, unseen data, e.g. in our case we assume that a trained and optimised DNN version



(a) (b) (c) (d) (e)

Fig. 3. Examples of the errors in the analysed dataset: (a) false detection; (b) fully occluded vehicle behind the white lorry; (c) missing vehicle's bounding box; (d) partially occluded vehicle; (e) wrong sized bounding box.

TABLE I
ERRORS IDENTIFIED FROM A PRELIMINARY VISUAL INSPECTION OF BOUNDING BOXES IN KITTI MOSEG TESTING DATASET

Error Type	**Count**	**Description**
Missing	87	An object is present in the image but is not labelled
Incorrect	100	The ground truth bounding box does not highlight a object
Bad Fit	14	The bounding box fit is not appropriate for the identified object
Occlusion	550	The bounding box shows the bounds for the predicted object size, even if the object is occluded

will be deployed in automated vehicles to infer in real-time based on the video data stream produced by the cameras.

A. Errors and Bias in DNN models

The importance of understanding and diagnosing errors in computer vision and object detectors has been well known, and it has been early discussed in works such as the one presented by Hoiem *et al.* in 2012 [10]. This early work divided the false positives and false negatives identified by different object detectors and clearly identified that a component in the overall number of errors is due to confusion with background or unlabeled objects. The authors also investigated aspects of truncation, occlusion, deformation and object size. However, the considered models are outdated, and also the study is more focused on understanding the errors due to the model than how they are related not only to the model structure, but also to the quality of bounding boxes and labeled ground truth in the datasets.

Recent studies have analysed in more details how DNNs are learning, generalising, and how the extraction of different types of features has an impact on performance [11]. Some common techniques to enhance the out-of-distribution performance of DNNs (i.e. the DNN ability to generalise even when inferring on data with distribution differing from the original distribution of the training set) are based on forcing the model to learn from higher-level features, however these networks might be prone to annotation artifacts. In fact, a recent work on natural language inference has studied the relationship between annotation artefacts and natural language prediction [12]. This work highlights that there are several datasets used for Natural Language Inference containing annotation artifacts, and these artifacts are unavoidably related to biases in the models. However, the field of annotation artefacts needs to be explored in depth in the case of image recognition tasks, such as 2D object detection. There have been some works focusing on the bias introduced by incomplete and not sufficient data, raising concerns on the ethical implications of using *biased networks* and more importantly on their safety (e.g. a network trained to neglect truncated cars can mis-detect an immediate danger) [13].

B. Challenges related to datasets used for Assisted and Automated Driving

A study has focused in understating the relationship between dataset bias in combination with neutral or biased learning styles. It presents some very interesting results related to automated vehicles, considering, for example, the importance of learning using data from different cities, countries, etc. [14]. Building on these previous works, a very recent project has analysed the bias introduced by not considering environmental variability in the training dataset, comparing some traditional bench-marking datasets, i.e. the Microsoft COCO and PASCAL VOC datasets [15], [16], with the DAWN dataset, which is an automotive labelled dataset covering the main road stakeholders (i.e. car, motorbikes, vans, etc.) in adverse weather conditions (snow, fog, rain and sandstorm) [17], [18]. Interestingly, the paper demonstrates that several DNN based detectors are affected by a 'good weather bias' and perform poorly on data with variable adverse weather; moreover the paper proposed a new training technique to reduce this bias. Despite this interest in DNN bias due to data and models, to the best of our knowledge there are no papers exploring errors in the labels of the most widely used datasets, not even of bench-marking automotive datasets (e.g. KITTI, A2D2, panda, etc. [2], [3], [19], [20]), where errors in labels can affect the safety of the driving functions.

In particular, as a part of this work we have investigated a subpart of the KITTI dataset, which is widely used for bench-marking machine learning and DNNs for assisted and automated driving tasks. We have selected the KITTI MoSeg part of the full dataset, since it has temporally correlated frames and can be useful when investigating video data compression [21], [22]. The labels of the MoSeg dataset have been generated automatically using a simultaneous motion and vehicle detection DNN architecture. If, from one side, this automation means that a DNN can reduce the man-hours to carry out the labelling task, on the other side, as demonstrated in this work, there are several errors, of at least 4 different types (as described in the following section). One clear example of errors related to the labelling network architecture is legacy labels related to motion estimation, e.g. a detected vehicle in sequential frames is still detected (i.e. it has a ground truth bounding box) even if another object/vehicle appears in the frames obstructing the initially detected vehicle, as shown in Fig. 2.

III. Methodology

The ground truth bounding boxes from the KITTI MoSeg dataset were investigated to understand how incorrectly labelled data could affect the performance of neural networks. The KITTI MoSeg dataset contains about 1300 training images and 349 testing images [2], [3]. Compared to the original KITTI dataset, the images have been further processed to provide information such as optical flow. This dataset provides bounding box labels for `car` and `van` across sequential images from 6 videos in the training set and 2 videos in the testing set from the original KITTI dataset.

TABLE II
Number of original bounding boxes vs. added, removed, or resized ones when using the proposed criteria, Sec. III.A.

	Car	Van	Total
Original Number of Bounding Boxes	**2337**	**311**	**2648**
Added Bounding Boxes	292	75	367
Removed Bounding Boxes	515	14	529
Resized Bounding Boxes	1974	491	2465
Final Total of Bounding Boxes	1987	499	2486

A. Testing set inspection

After some preliminary activities based on DNNs and the KITTI MoSeg dataset [21], [22], a thorough inspection of the 349 frames in the test dataset was carried out. The test dataset was visualised by overlaying the ground truth bounding boxes onto their respective images. Each overlaid image was then visually inspected to roughly count and categorise any evident errors in the bounding boxes (BBs). Based on the identified mistakes in the dataset BBs, the errors were attributed to four different classes, as described in Table. I: (i) *missing*; (ii) *incorrect*; (iii) *bad fit*; (iv) *occlusion*. Some of the identified errors are shown in Fig. 3.

B. Initial criteria for labelling

A set of criteria, derived from the initial inspection, was agreed upon to re-label the MoSeg testing dataset and to reduce the number of incorrect ground truth BBs. The aim was to decrease the impact of incorrect ground truth BBs on the performance of a neural network. These criteria were also chosen based on the specific DNN task that we considered in this work, i.e. detection of vehicles from an assisted and automated driving perspective. Namely, our **relabelling criteria** were:

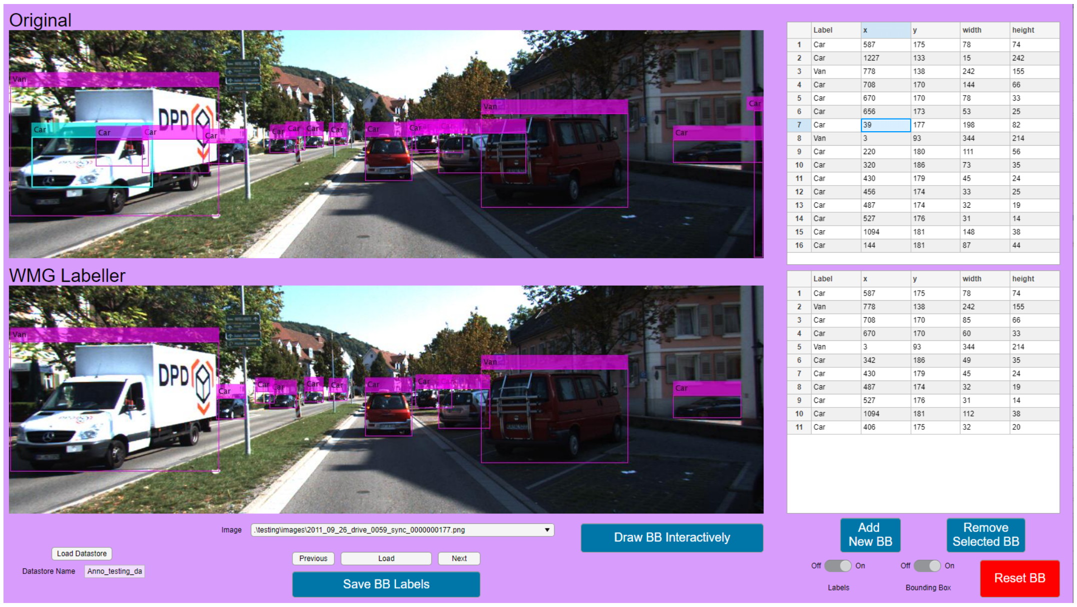
1) the bounding box shall be no less than 20 pixels in either the width or the height;
2) the bounding box shall contain all visible parts of the target object, with an error lower than 3 pixels;
3) the size of the bounding box shall not include any estimated or occluded parts of the target object unless criteria 2 is applicable;
4) at least more than half of one face of the target object (i.e. front, rear, left, right) must be visible to be eligible to be labelled.

C. Perception task

As previously mentioned, the bounding box annotation criteria should be designed to be specific for the use case. The KITTI dataset is widely used to support the development and testing of assisted and automated driving functions. These functions are implemented via the pipeline of *sensing, perceiving, planning* and *control*. The selected dataset can replace the data produced by the *sensing* step and processed by the *perception* step. In this work, a deep neural network based object detector, Faster R-CNN [23] with ResNet50 [24] as the backbone, was chosen to perform the *perception* step. The selected Faster R-CNN was already pre-trained on COCO train2017 [25] and it was fine-tuned for this study with the training set of the KITTI MoSeg dataset. It is worth noting that an error analysis on the MoSeg training set has not been carried out, but randomly sampling this part of MoSeg, errors similar to the ones identified in the testing set were observed. As a part of this study, the fine tuned DNN, thereafter named the selected Faster R-CNN, has been used to generate the prediction for the original dataset and the re-labeled one. The traditional performance metrics (mean Average Precision and Recall, mAP$_{50}$ and mAR$_{10}$ respectively [26]) have been used to evaluate the quality of the detections.

TABLE III
MoSeg and BoSeg testing sets evaluated using the selected Faster R-CNN; last 2 rows are the results when including only bounding boxes bigger than 50 pixels

Training dataset	**Testing dataset**	**mAP$_{50}$**	**mAR$_{10}$**
MoSeg train	MoSeg test	78.7%	53.0%
MoSeg train	BoSeg test	75.3%	50.0%
MoSeg train	MoSeg test$_{>50px}$	71.8%	56.4%
MoSeg train	BoSeg test$_{>50px}$	79.7%	63.1%



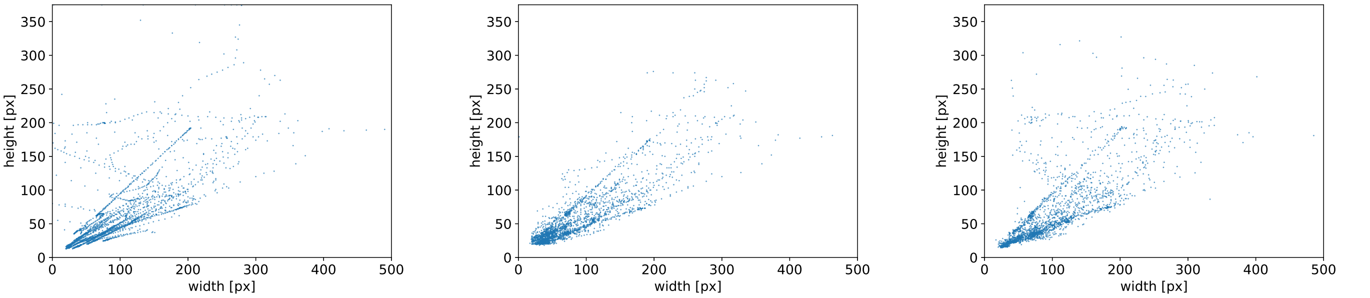
	Label	x	y	width	height
1	Car	587	175	78	74
2	Car	1227	133	15	242
3	Van	778	138	242	155
4	Car	708	170	144	66
5	Car	670	170	78	33
6	Car	656	173	53	25
7	Car	39	177	198	82
8	Van	3	93	344	214
9	Car	220	180	111	56
10	Car	320	186	73	35
11	Car	430	179	45	24
12	Car	456	174	33	25
13	Car	487	174	32	19
14	Car	527	176	31	14
15	Car	1094	181	148	38
16	Car	144	181	87	44

	Label	x	y	width	height
1	Car	587	175	78	74
2	Van	778	138	242	155
3	Car	708	170	85	66
4	Car	670	170	60	33
5	Van	3	93	344	214
6	Car	342	186	49	35
7	Car	430	179	45	24
8	Car	487	174	32	19
9	Car	527	176	31	14
10	Car	1094	181	112	38
11	Car	406	175	32	20

Image .\testing\images\2011_09_26_drive_0059_sync_0000000177.png

Datastore Name Anno_testing_da

Fig. 4. Screenshot of the annotation app created for this work. It can be used to draw bounding boxes and to compare original and amended ground truth.



(a) (b) (c)

Fig. 5. Width and height distribution of bounding boxes in (a) the original MoSeg dataset; (b) BoSeg dataset; (c) as detected by the selected Faster R-CNN.

IV. Results and Discussion

A bounding box annotation app was newly developed in Matlab. The app allows for pixel wise editing of the bounding boxes, see Fig. 4. This app enables the user to add, remove and resize BBs and to compare the original and amended BBs.

A. The re-labelled BoSeg dataset

Based on the four criteria set out in Sec. III, the majority of the bounding boxes were adapted in the MoSeg testing dataset, Tab. II; from now on the re-labelled dataset will be called BoSeg. Based on the criteria, 367 new bounding boxes were added (they where missing in the original dataset). Some of these added BBs are related to small vehicles not labelled in MoSeg (see Fig. 3c) and they will be further discussed in combination with detection performance. Moreover, 529 bounding boxes were removed; these removed bounding boxes would be related to vehicles that do not meet the established criteria, or to completely occluded vehicles (see Fig. 3b). In this image, it can be seen that there are several vehicles fully occluded behind the white van on the left. There are also legacy bounding boxes from vehicles that pass by the edge of the frame as seen in the right of Fig. 3a. Finally, most of the bounding boxes were resized by at least the width or the height. The majority of these resize steps were minor changes to fit the vehicle better as per the proposed 3 pixel error in criterion 2. However, there are also many bounding boxes which were resized due to occlusion covering a significant amount of the vehicle (see Fig. 3d) or unnaturally large bounding boxes (see Fig. 3e). In addition, 28 vehicles labelled as `van` were re-labelled into `car`; on the other hand, 155 vehicles originally labelled as `car` were re-labelled as `van`.

A further analysis of the BB size distributions has been carried out in Fig. 5, to compare MoSeg with the proposed BoSeg and the detections by the selected DNN. The interesting aspect is that BoSeg has an higher concentration of small BBs, and also BB with one dimension way bigger than the other are reduced with respect to MoSeg. These aspects have an effect on DNN detection, as discussed in the following section.

B. Faster R-CNN performance

Evaluating the detection using MoSeg and BoSeg ground truth, MoSeg performs slightly better in terms of the selected metrics. Moreover, due to class unbalance, the `van` and `car` were combined into a single `vehicle` class. However, as mentioned, the selected network has been trained on MoSeg. Moreover, BoSeg has an higher distribution of small BB and it was observed that the DNN is not always able to detect them. For this reason, the performance metrics was recalculated filtering out BB with a side smaller than 50 pixels, and in this case BoSeg performs better than MoSeg. This results can be attributed to the improved quality of the BBs in BoSeg. It is worth noting that we expect that the effect of re-labelling the training dataset will have a even higher impact on the selected Faster R-CNN performance; this work is currently ongoing.

V. Conclusion

In conclusion, this paper has presented a preliminary analysis of the effect of ground truth bounding box errors on deep neural network based detectors, in the specific context of perception for assisted and automated driving functions (even though the results can be easily extended to other applications). In particular, this work proposes a re-labelling of the testing set of the KITTI MoSeg dataset, a commonly used automotive dataset for benchmarking machine learning and computer vision algorithms. The ground truth errors have been classified and estimated, and this classification can be applied to different datasets. Moreover, some criteria for the annotation of the dataset have been proposed; to the best of our knowledge there are no established criteria for this process. The preliminary results show that the re-labeled BoSeg performs better than the original MoSeg when we neglect small BBs, however we expect even better results when the full MoSeg training set will be annotated according to the proposed criteria and used for fine tuning of the DNN. This work provides a framework to evaluate the impact of ground truth quality on DNN performance, future work will analyse different datasets, perception tasks, and network architectures.

Acknowledgment

Boda Li was supported by the State Scholarship Fund by China Scholarship Council (CSC) from the Ministry of Education of P.R. China. Dr Donzella acknowledges the Industrial Fellowships scheme of the Royal Academy of Engineering. The Authors thank HVM CATAPULT for the support, and Dr Anthony Huggett, ON Semiconductor, for fruitful discussions.

References

[1] SAE J3016_202104, “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” Society of Automotive Engineers, Warrendale (PA), USA, Standard, 2021.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets Robotics: The KITTI Dataset,” *International Journal of Robotics Research (IJRR)*, 2013.

[3] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, “MODNet: Moving Object Detection Network with Motion and Appearance for Autonomous Driving,” *arXiv preprint arXiv:1709.04821*, 2017.

[4] K.-L. Du, “Clustering: A neural network approach,” *Neural networks*, vol. 23, no. 1, pp. 89–107, 2010.

[5] D. Bank, N. Koenigstein, and R. Giryes, “Autoencoders,” *arXiv preprint arXiv:2003.05991*, 2020.

[6] W. Wang, Y. Yang, X. Wang, W. Wang, and J. Li, “Development of convolutional neural network and its application in image classification: a survey,” *Optical Engineering*, vol. 58, no. 4, p. 040901, 2019.

[7] F. Sultana, A. Sufian, and P. Dutta, “Evolution of image segmentation using deep convolutional neural network: a survey,” *Knowledge-Based Systems*, vol. 201, p. 106062, 2020.

[8] Y. Li and J. Ibanez-Guzman, “Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems,” *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[9] W. Jiang, Y. Ren, Y. Liu, and J. Leng, “Artificial neural networks and deep learning techniques applied to radar target detection: A review,” *Electronics*, vol. 11, no. 1, p. 156, 2022.

[10] D. Hoiem, Y. Chodpathumwan, and Q. Dai, “Diagnosing error in object detectors,” in *European conference on computer vision*. Springer, 2012, pp. 340–353.

[11] H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli, “The pitfalls of simplicity bias in neural networks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9573–9585, 2020.

[12] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith, “Annotation artifacts in natural language inference data,” *arXiv preprint arXiv:1803.02324*, 2018.

[13] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, “Undoing the damage of dataset bias,” in *European Conference on Computer Vision*. Springer, 2012, pp. 158–171.

[14] D. Danks and A. J. London, “Algorithmic bias in autonomous systems.” in *IJCAI*, vol. 17, 2017, pp. 4691–4697.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[16] M. Everingham and J. Winn, “The pascal visual object classes challenge 2012 (voc2012) development kit,” *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, vol. 8, p. 5, 2011.

[17] M. A. Kenk and M. Hassaballah, “Dawn: vehicle detection in adverse weather nature dataset,” *arXiv preprint arXiv:2008.05402*, 2020.

[18] A. Marathe, R. Walambe, K. Kotecha, and D. K. Jain, “In rain or shine: Understanding and overcoming dataset bias for improving robustness against weather corruptions for autonomous vehicles,” *arXiv preprint arXiv:2204.01062*, 2022.

[19] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn *et al.*, “A2d2: Audi autonomous driving dataset,” *arXiv preprint arXiv:2004.06320*, 2020.

[20] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, “The apolloscape dataset for autonomous driving,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 954–960.

[21] P. H. Chan, G. Souvalioti, A. Huggett, G. Kirsch, and V. Donzella, “The data conundrum: compression of automotive imaging data and deep neural network based perception,” in *London Imaging Meeting*, vol. 2021, no. 1. Society for Imaging Science and Technology, 2021, pp. 78–82.

[22] P. H. Chan, A. Huggett, G. Souvalioti, P. Jennings, and V. Donzella, “Influence of AVC and HEVC compression on detection of vehicles through Faster R-CNN,” *TechRxiv preprint*, 5 2022.

[23] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft COCO: Common Objects in Context,” 2014. [Online]. Available: https://arxiv.org/abs/1405.0312

[26] COCO Detection Evaluation Metrics. [Online]. Available: https://cocodataset.org/#detectioneval