Journal of Intelligent Manufacturing (2022) 33:2157–2165
https://doi.org/10.1007/s10845-022-01983-4

Testing the reliability of monocular obstacle detection methods in a simulated 3D factory environment

Marius Wenning[1] · Anton Akira Backhaus[1] · Tobias Adlon[1] · Peter Burggräf[1]

Received: 29 November 2021 / Accepted: 13 June 2022 / Published online: 11 July 2022

Abstract

Automated driving in public traffic still faces many technical and legal challenges. However, automating vehicles at low speeds in controlled industrial environments is already achievable today. A reliable obstacle detection is mandatory to prevent accidents. Recent advances in convolutional neural network-based algorithms hypothetically allow the replacement of distance measuring laser scanners with common monocameras. In this paper, we present a photorealistic 3D simulated factory environment for testing vision-based obstacle detecting algorithms preceding field tests on the safety–critical system. We further test two obstacle detection methods employing state-of-the-art semantic segmentation and depth estimation in a range of challenging test scenarios. Both models performed well under common factory settings. Some edge cases, however, lead to vehicle crashes.

Keywords Automated factory transport · Visual obstacle detection · Autonomous transport

Introduction

Fully automated driving is considered one of the most significant contemporary challenges in computer science. The related problem complexity can be decreased by addressing specific use cases thereby limiting the operational design domain. For example, closed, artificially lit factories provide a very static and controlled environment. Here, it seems achievable for automated driving cars to reach a car park from the production line end. Automating this tedious and repetitive job could minimize manpower and most importantly reduce the risk of accidents (Wenning et al., 2020) and thus facilitate car manufacturing in general. Further, automated factory transportation can be extended to Automated Guided Vehicles (AGV). They play a crucial role in factory and warehouse logistics by transporting materials or products on the premises. Regardless of the application, automated systems in a human environment require a reliable obstacle detection (OD). This paper focuses on OD with a monocular frontal camera.

This sensor setup is simple compared to most self-driving systems that use an expensive LiDAR sensor suite, radar and multiple cameras (Feng, et al., 2020). Due to mandatory driver assistance systems, a monocular camera will soon belong to the standard car sensor setup. Since most factories are set-up for (human) vision, computer vision that imitates the human brain's image processing is arguably sufficient for robust OD. Given that the cars are also equipped with electric actuators, they can be automated on the factory premises.

To utilize data acquired through optical sensors, suitable computational strategies are required. In recent years, Convolutional Neural Network (CNN)-based algorithms made monocular OD increasingly viable. In automated driving, these algorithms have been built to perform complex scene understanding in public traffic. The simpler use case of automated factory transport requires only the binary classification *Stop/Go* and performs in the limited operational design domain of a factory. Consequently, we tested whether a CNN-based algorithm is able to reliably detect a range of obstacles. To answer this question, we contribute an application specific dataset, which consists of photorealistic images

✉ Marius Wenning
marius.wenning@rwth-aachen.de

Anton Akira Backhaus
antopost@gmail.com

Tobias Adlon
t.adlon@wzl.rwth-aachen.de

Peter Burggräf
p.burggraef@wzl.rwth-aachen.de

[1] Werkzeugmaschinenlabor, RWTH Aachen University, Aachen, Germany

Fig. 1 Overview of factory environment simulated in Unreal Engine 4

from a virtual factory environment (Fig. 1). The simulation was built in Unreal Engine 4 and enables the generation of an arbitrary number of noiseless ground truth segmentations and depth data in a variety of scenarios that would otherwise be dangerous or costly to provide. The dataset comprises safety–critical scenarios, which pose challenges to vision-guided autonomous vehicles, for example detecting humans with floor-colored clothing, fire outbreaks, vehicles with large floor clearance and unknown objects. Using these, we evaluate the viability of two CNN-based methods including semantic segmentation (SeS) and monocular depth estimation (DE) for OD. The SeS algorithm classifies each image pixel into one of two classes, either obstacle or drivable path. The DE provides a distance value for each image pixel. Using a threshold, the algorithm can deduce a binary output to control an automated vehicle.

In the following, we present previous work that investigated why and how DE and SeS CNN models commonly fail. We then elaborate on both OD algorithms and the virtual data generation method. Next, we introduce six test scenarios and, based on these, evaluate and discuss both OD methods. Lastly, we provide a summary and suggest future work.

Related work

Monocular depth estimation

Autonomous navigation is inevitably based on detailed depth analysis of the surrounding scene. Hence, almost all vehicles that work autonomously rely on a depth sensor. Monocular vision-based DE is limited by an inherent ambiguity problem since there is an infinite number of 3D correspondences to a 2D image. CNN-based methods deal with the ambiguity problem by implicitly learning depth cues similar to humans. Understanding the effects of depth cues on CNNs is vital for safety–critical DE tasks, as it may resolve potentially fatal edge cases that can result in fatal failure. Depth cues that are relevant for CNN-based methods have only recently become a research focus. Dijk & Croon, 2019 investigated which high-level depth cues are learned for detecting cars by varying the appearance and position of road objects as well as scene color in the KITTI dataset (Geiger & Urtasun, 2012). Here, vertical position, shadow darkness, texture gradient and the distinctiveness of the object drove the inferred object's depth accuracy. In a study aimed to identify low-level features that govern DE accuracy (Hu et al., 2019), Hu et al. found that CNN models selected edges depending on their importance for inference of scene geometry, not on their intensity. In accordance with van Dijk and de Croon's hypothesis, the lower edge of an object establishing the contact point to the ground seemed to be most important. However, Hu et al. further argued that depth estimators preferably detect the boundary and the inside region of each individual object.

While both studies offer valuable insights into the importance of various depth cues, most current research is limited to outdoor traffic scenes and provides little insight into DE behavior in edge cases and the reliability of depth estimators in a factory setting.

Semantic segmentation

While SeS has been shown to be useful in the field of automated driving, CNNs in general are still widely considered to be black-boxes which raises concerns in safety–critical applications. Eykholt, et al., 2017 demonstrated traffic sign detector failure upon stop sign modifications with black and white stickers. Similarly, person detectors failed because of adversarial designs on clothing (Xu, et al., 2019). The same principle of adversarial attacks on image classifiers can be extended to SeS (Bär et al., 2021). While these adversarial attacks are man-made, similar situations could arise in real-world applications causing critically delayed reactions of moving vehicles to obstacles.

Research has focused on low-level feature attacks and their effects on SeS network output. Unlike for DE networks (see above), for SeS little is known on the effects of high-level image alterations, for example adding novel objects to the scene or changing object appearance.

Additionally, the quality of CNN models heavily depends on the quality of training data: For example, the KITTI (Geiger & Urtasun, 2012) and Cityscapes (Cordts, et al., 2016) datasets are among the most used datasets for benchmarking computer vision models in traffic settings. However, Johnson et al. (Johnson-Roberson et al., 2016) pointed out that vehicles are mostly seen from the same angle and image regions causing the CNN to underperform in fringe situations.

When developing a vision-based OD system for safety–critical applications, simulation testing reduces risks and minimizes development costs. Some simulated environments for traffic scenes already exist (Dosovitskiy et al., 2017;

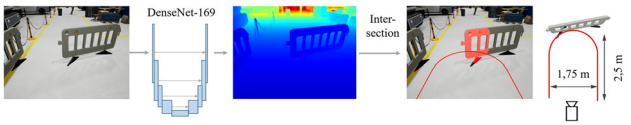

Fig. 2 Full model pipeline of depth estimation method for obstacle detection

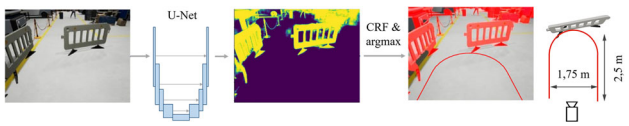

Fig. 3 Full model pipeline of semantic segmentation method for obstacle detection

Johnson-Roberson et al., 2016; Li et al., 2019; Pollok et al., 2019), but there are no such photorealistic industrial environments for testing vehicle automation to the best of our knowledge. Furthermore, most efforts have been directed at the neural networks on a low feature level and only few studies have pointed out vulnerabilities in DE and SeS models. Therefore, we aim to close this knowledge gap by investigating the robustness of OD methods in an industrial environment.

Implementation

Depth estimation model

For DE we use DenseDepth (Alhashim & P. Wonka, 2018) which consists of an auto-encoder architecture. The encoder part is equipped with a feature extractor, DenseNet-169 (Huang et al., 2016). The model is trained on RGB images and 8-bit grayscale images as ground truth depth where the pixel intensity denotes the distance. The recorded depth is truncated at 15 m and scaled to the values 0–255. While training DenseDepth, a random subset of data is color augmented as described in the original paper (Alhashim & P. Wonka, 2018). Due to the overabundance of synthetic data, the training time can be limited to 20 epochs.

We use an $\cap$-shaped area extruded along the vertical axis (Fig. 2) to deduce a binary decision value. If an object enters the $\cap$-volume, it should trigger a stop of the vehicle.

When the floor appearance changes (e.g., dark shadows, dirt), depth estimates can become noisy. These depth uncertainties become higher with larger distances. To guarantee a robust OD, the $\cap$-volume needs to be lifted. Different noise variances can be taken into account by raising the floor tolerance for each pixel individually. The values are determined by taking the maximum errors of all non-obstacle frames of the training set. Since these tolerance values lead to no false positives, we lower them by 80%. Thus, the risk of ignoring low obstacles is reduced without considerable accuracy loss.

Semantic segmentation model

We use binary SeS to classify anything that is not part of the traversable floor. A single RGB image is used as the CNN's input. The output is a per-pixel probability distribution for obstacles. A subsequent conditional random field (CRF) that refines the segmentation mask is followed by the argmax function to produce a binary mask. The vehicle must stop if any part of the obstacle mask enters the (not extruded) $\cap$-shaped region, cf. Figure 3. This method is 2D and thus assumes that only non-overhanging obstacles are present within the scene since only the obstacle's contact points to the floor are considered. This means that the obstacle distance will be overestimated when the point that first enters the $\cap$-shape is elevated from the ground.

For SeS, we employed the U-Net (Ronneberger et al., 2015) architecture with additional dropout layers (Adams, 2021). Dropout layers reduce over-fitting by dropping a different random sub-set of nodes at each training epoch (Hinton et al., 2012). For enhancing the generalization power, a light data augmentation is applied to 50% of training images per epoch. The augmentation sequence consists of small varying degrees of desaturation, color channel intensity variation, darkening/lightening, Gaussian blurring as well as perspective transformation, horizontal flipping (i.e. mirroring), and horizontal translation. The training comprises 20 epochs as with the DE model.

Virtual training data

A virtual camera recorded the data along a pre-generated trajectory with a constant pitch of -30°. The traversable path went around the center assembly area. It was marked by yellow tape on either side. To ensure that the camera captured a unique view and emulated autonomous vehicle motion, a Bézier curve was generated by placing the control points semi-randomly along the traversable path with bias to the path center. Lastly, the curve was converted into a set of approx. 2150 equidistant points representing each labelled image capturing position. A point distance of about 4 cm and a theoretical frame rate of 60 fps corresponded to a vehicle speed of 8.6 km/h. The vehicle speed determines the size of the $\cap$-shape since faster vehicle speeds require longer deceleration distances. Given high quality images, the obstacle detection capabilities are independent from vehicle speed for the employed algorithms.

We used UnrealCV (Qiu & Yuille, 2016) to capture RGB images, segmentation masks and depth images (Fig. 4). The object mask consisted of only two classes: floor and obstacle. Depth and object masks were saved as 8-bit grayscale png

Fig. 4 Ground truth data output of UnrealCV used for training of SeS and DE model: RGB (left), object mask (center), depth (right)

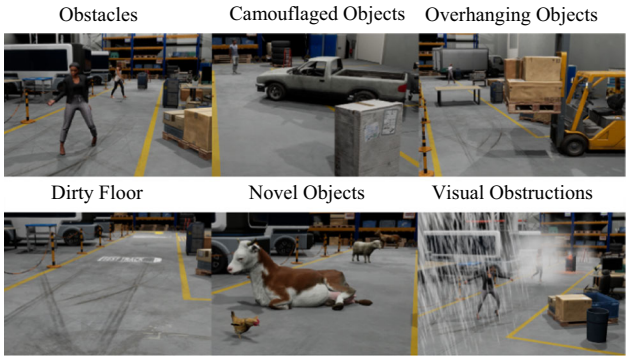

Fig. 5 Testing scenarios in industrial environment posing challenges to visual obstacle detection

images at 640*480 pixels. RGB images were compressed and saved as jpg of the same size. We generated 15,170 training and 4370 testing images.

Test cases

We devised six test scenarios representing safety and robustness requirements (Fig. 5). The scenarios used the same camera trajectory for control purposes. Some tests were less extensive and were performed only on certain segments of the training course. The scenarios were detailed as follows:

Obstacles: This run was a benchmark for further tests. It contained 15 obstacles of different shapes and sizes that can be commonly found in factories such as humans, chairs, fire extinguishers, barriers, and boxes.

Camouflaged objects: A major challenge for computer vision models is detecting objects similar to the background. For this purpose, six gray obstacles were placed on the path, including a person and a vehicle with the same texture as the floor.

Overhanging objects: These were objects elevated above the floor such as forklifts with raised cargo or large vehicles with clearance. Even for laser scanner-equipped AGVs these obstacles can be difficult to detect based on their frontal, floor-based focus (Ullrich & Albrecht, 2019). This run contained six challenging objects.

Dirty floor: Dirt and tire marks accumulate on factory floors. This test investigated vision model robustness to changes in floor appearance. Here, previous scenarios were employed with additional dirt, tire marks and three floor decals. The test case did not contain obstacles.

Novel objects: Depending on their purpose, most objects found in a factory are often square-shaped, made of wood or metal and are clearly visible. This run aimed to test performance on novel objects that are absent from training data such as animals, plants, and mythical creatures.

Visual obstructions: Visual obstructions pose a high risk to visual object detection. Benefitting from the wide range of applications that simulations allow to test for, this run aimed to test the models' ability to detect objects obstructed by rain, light smoke, heavy smoke, and sparks.

Table 1 Software and libraries employed

	Main libraries	Software
Data generation	Factory Data Generator UnrealCV 0.4.0 Office Scene Scanned 3D People Pack Open World Demo Collection Factory Environment Collection Vehicle Variety Pack	Unreal Engine 4.25 with UnrealCV plugin Windows 10
Evaluation	U-Net DenseDepth	Ubuntu 18.04

Software

Table 1 gives an overview on the employed software. Additionally, we provide the training and test data and the factory environment simulated in Unreal Engine 4.

Evaluation

To evaluate the OD capability of both models introduced in Sect. 3, we processed all test cases' images. Frames were marked as *stop* when any pixel of the ground truth depth was within the $\cap$-volume. False negatives (FN) comprised close obstacles that were not detected. Contrarily, false positives (FP) comprised close but undetected obstacles. A true positive (TP) was a correctly classified frame with an obstacle. Using the f1-score, the results on each test case can be condensed to one number per model:

$$f_1 = \frac{TP}{TP + 0.5 * (FP + FN)}$$

Table 2 sums up the models' results. In addition to the f1-score, we provide the maximum number of consecutive

Table 2 Performance metrics of Depth Estimation (DE) and Semantic Segmentation (SeS) on test cases

Test set	Model	f1-score	Max. FN series
Obstacles	DE	**0.97**	**7**
	SeS	**0.97**	16
Camouflaged Objects	DE	**0.98**	**2**
	SeS	0.96	8
Overhanging Objects	DE	**0.86**	**29**
	SeS	0.76	41
Dirty floor	DE	**–***	–*
	SeS	–*	–*
Novel objects	DE	**0.93**	**25**
	SeS	0.92	26
Visual obstructions	DE	0.75	**1**
	SeS	**0.81**	16

Boldfaced numbers indicate a better performance
*Note that the Dirty Floor set lacks obstacles

frames that were classified as FNs as these could result in a crash. Since each frame is taken approx. 4 cm apart and the ∩-shaped detection area is 2.5 m long, a FN series of >62 translates to a full-speed crash. Considering a braking distance of 1 m due to signal processing delays and vehicle inertia, the maximum FN series decreased to 37 frames. This threshold was exceeded in the *Overhanging Objects* set and nearly approached in several fringe cases.

For further analysis, we plotted the frame-wise errors of *stop/go* predictions for each model and run. FPs only caused the vehicle to stop unnecessarily while an FN can lead to a collision. The former are indicated in orange and the latter in red. Long red areas indicate a major obstacle detection failure. The graph identifies obstacles that were particularly challenging for specific models. The ground truth labels were determined with the same stopping criterion based on ground truth depth and semantic images. Images with the frame number indicated in the bottom right are provided as context for the *stop/go* graph.

Obstacles

Based on Fig. 6, both models detected the obstacles. For both, the person at frame 209 was detected with a seven-frame delay due to the protruding arm (Fig. 7, top). Here, the vehicle could still stop in time with the given framerate and speed of the camera (4 cm/frame). The models differ most around frame 1200. Here, a canvas barrier slouched towards the camera (Fig. 7, bottom). Despite the complex geometry, the DE model was able to detect it sufficiently early. The SeS model detected it slightly later due to the overhanging cloth.

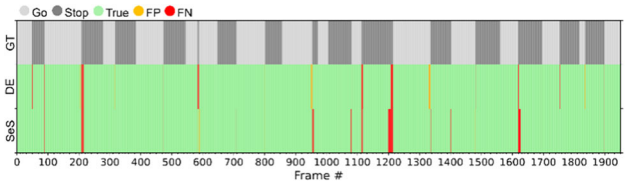

Fig. 6 Truthfulness of predictions in chronological order of *Obstacles* set

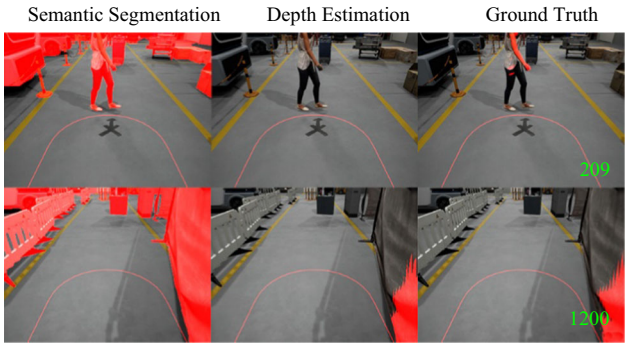

Fig. 7 Slightly late detected person (top); detecting a slouching canvas barrier (bottom)

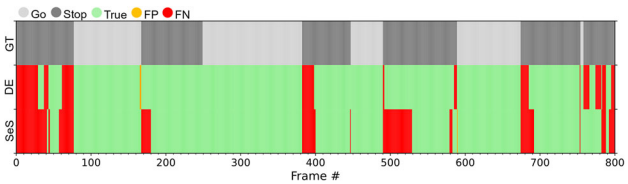

Fig. 8 Truthfulness of predictions in chronological order of the *Overhanging Obstacles* set

Overhanging obstacles

Obstacles with overhanging parts have shown to be the main source of FN in the previous tests and are the most challenging to detect, even for depth-aware models. The SeS model lacks depth awareness and consequently failed on most obstacles. Here, the focus of the experiment is on the DE. Figure 8 shows that DE improved the detection of elevated and overhanging obstacles, however, many late detections indicate a low reliability. Further, many frames were only classified correctly because of the dark shadows cast by overhanging objects. The shadows caused true positive detections in both models. However, shadows depend on the lighting and are not reliable indicators for object detection. Importantly, late detections and shadow-caused TPs reflect a limited reliability of the DE.

The run started with a forklift in a highly elevated position. Only the cargo was visible while the vehicle itself lied outside of the frame making this a particularly difficult obstacle to detect. Indeed, both models overlooked this obstacle. The predicted depth image displayed in Fig. 9 (top) suggests that

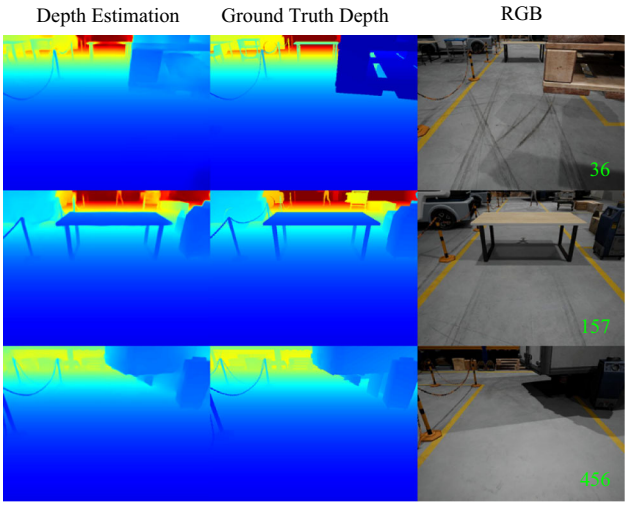

Fig. 9 Incorrectly predicted depth of a raised pallet (top); accurate predictions of a table (center) and back of a truck (bottom)

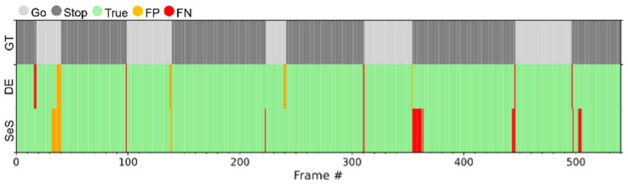

Fig. 10 Truthfulness of predictions in chronological order of *Camouflaged Obstacles* set

the position of the pallet was estimated further back in the image where its bottom edge crossed over to the floor.

Interestingly, the table (frames 167 – 248) and truck (frames 490 – 588) were very accurately predicted, thus triggering almost frame-perfect stops. The truck prediction is especially surprising considering that there was no clear contact point to the ground visible in the image, Fig. 9 (bottom). There was, however, a dark shadow directly under both objects, which seemed to have a positive effect on prediction accuracy.

Camouflaged obstacles

Both models detected all objects based on the frame-wise predictions in Fig. 10. However, some segmentation outputs were sparse and others overcomplete (e.g., frames 40–50). Notably, even incompletely segmented obstacles were never fully overlooked since there was always some part (e.g. edges or lighting) distinguishable from the floor.

SeS further failed to segment the dark rocker panel and bottom tire of the SUV in Fig. 11, possibly due to its similarity to a shadow indicating a risk of late detection for objects with dark lower bottom edges. In contrast, DE worked surprisingly well for connecting the rocker panel to the vehicle.

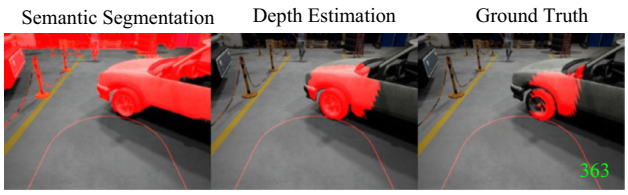

Fig. 11 When detecting the car the dark rocker panel is mistaken for a shadow

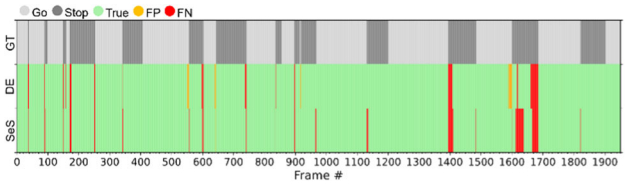

Fig. 12 Truthfulness of predictions in chronological order of *Novel Objects* set

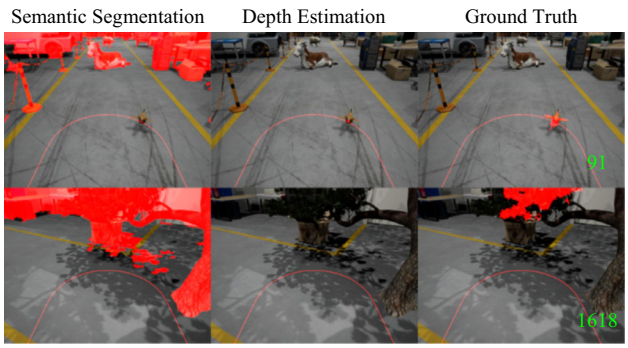

Fig. 13 Running chicken ignored by SeS (top left) and barely caught by DE (top center); overhanging tree branch ignored by both models (bottom)

The final obstacle (frames 496 – end), a human wearing gray shoes and trousers, posed difficulties to the SeS in few frames. Here, the shoes have little texture, are relatively flat and thus were harder to distinguish from the floor compared to the trousers that have quite pronounced texture and shadows.

Novel objects

As seen in Fig. 12, both models detected the novel objects similarly well in most instances. At frames 90 and 91 the SeS model completely failed to segment the running chicken, which is also barely registered by the DE model (Fig. 13, top). The reasons for this are unclear; possibly the tire marks, small dimensions and/or low contrast decreased the accuracy of both CNNs.

Furthermore, an analysis of frames 1600 – 1685 (passing under a tree branch) revealed that the SeS model classified dark shadows falsely as obstacles and thus confirms the difficulty of detecting overhanging obstacles. Although both

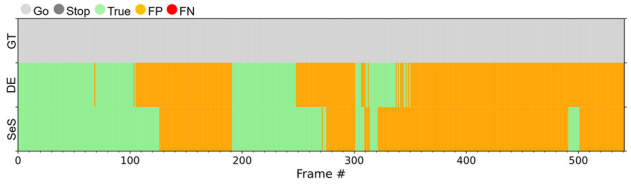

Fig. 14 Truthfulness of predictions in chronological order of *Dirty Floor* set

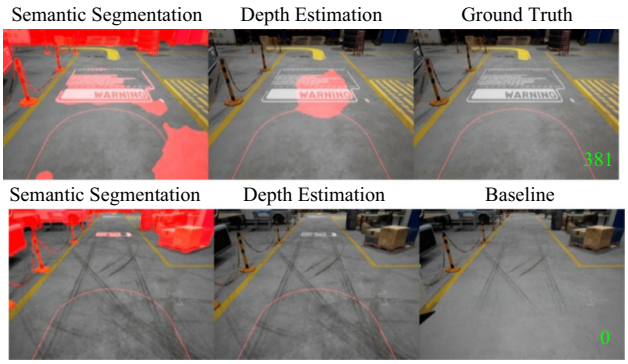

Fig. 15 False detection of white decal in both models (top); high robustness against dirty floor (bottom). The baseline image shows the floor conditions during training

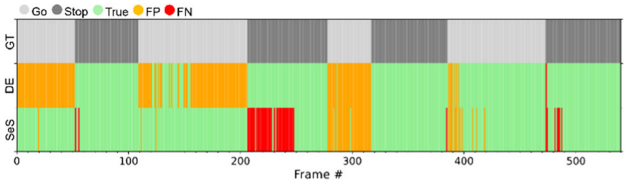

Fig. 16 Truthfulness of predictions in chronological order of *Visual Obstructions* set

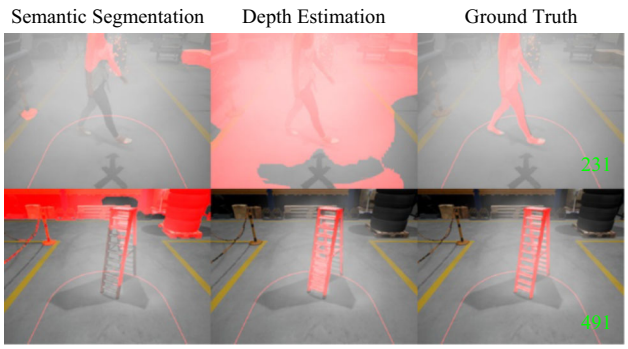

Fig. 17 Thick smoke causes the SeS to not detect the person. The DE detects the smoke as an obstacle (top). Thinner smoke around a latter still shows similar results in the SeS and less (bottom)

models triggered a stop (Fig. 12 frames 1600 – 1685), this is caused by shadows and the tree trunk. The actual obstacle, the branch, goes completely unnoticed (Fig. 13, bottom).

Dirty floor

In this test no obstacles were present. However, Fig. 14 shows that a changed floor appearance caused many FP detections in both models.

The test revealed that both models treated decals that differ greatly from the learned floor appearance (present in frames 130 – 191, 389 – 491) as obstacles. It is also noteworthy that bright yellow decals are mostly ignored by the SeS, unlike white ones (Fig. 15, top). It is the opposite with the DE model. The reasons for this, however, are unclear.

Notably, both models are robust to additional markings applied during normal usage of the floor (i.e., dirt and tire marks). This indicates a good transferability as light dirt and tire marks are represented in the training set. Figure 15 (bottom) depicts a comparison between the floor as seen in the training data and a dirty floor.

Visual obstructions

Figure 16 shows that both models performed poorly to detect obstacles when facing visual obstructions. The DE model estimated all pixels too close, thus initiating a break at every visual obstruction except light smoke (Fig. 17, bottom). This suggests that it performs reliably up to a certain smoke density.

The SeS model is relatively robust to rain and classified sparks as obstacles. More importantly, thick smoke caused obstacles to go largely undetected (Fig. 17, top). This is likely due to the loss of contrast. The entire scene took on a gray tone similar to the floor and thus detected the person at frames 207 – 277 far too late.

Discussion

The main findings are as follows: Firstly, the two evaluated methods are capable of reliably detecting obstacles under normal circumstances. However, some fringe cases pose problems for safe operation of the AGV. Secondly, overhanging obstacles were most difficult to detect. Shadows facilitated the detection and caused an overly optimistic performance in our dataset. Bad lighting conditions could further reduce obstacle detection capabilities. Moreover, the task became increasingly difficult when the object-floor contact point was occluded. In this case the depth estimator viewed the elevated object as lying on the floor further in the background. Distinct shadows from elevated objects significantly improved DE accuracy. Both findings support van Dijk's and de Croon's hypotheses (Dijk & Croon, 2019). Additionally, accuracy decreased for complex objects, especially for parts without direct connection to the floor.

Thirdly, the SeS model was slightly affected by camouflaged objects and tended to produce incomplete segments. Rare cases of objects small and flat that do not cast a shadow on themselves require additional research. Based on our investigation, however, it is unlikely that floor-like objects cause crashes. In reality every object has at least a slight texture and casts a shadow on itself or has different reflective properties that distinguish it from the floor.

Fourthly, both methods are robust to novel objects. This is unsurprising for SeS because binary classification classified all regions dissimilar to the floor as obstacles. For accurate DE this is more noteworthy since it requires consideration of the geometry to produce an accurate depth image. This indicates that DE heavily relies on low-level cues which can be learned from training on different objects.

Fifthly, when relying solely on the SeS model, smoke can cause a vehicle crash. It reduced feature contrasts and tinted the entire scene in gray similar to the floor. The DE model reacted to thick smoke by treating it as a close object, therefore coming to an early halt. Although a certain level of robustness to visual obstructions is desirable, a smoke breakout inside a factory should trigger a halting of all automated vehicles to ensure a safe evacuation. The SeS model fails in this regard.

Finally, introducing floor decals, which are not included in the training set will cause unwanted stops. Additional dirt, on the other hand, is tolerable to a degree. Conclusively, the models must be retrained when floor appearance is changed significantly.

Conclusion and future work

In this paper, we propose two methods for obstacle detection (OD) of automated vehicles in an industrial environment based on CNNs for semantic segmentation and depth estimation, respectively. Additionally, we devise a 3D virtual environment for generating RGB images as well as ground truth segmentation and depth images. Thus, we contribute an application-specific dataset, which addresses the OD challenges in industrial environments. The test data includes six scenarios featuring normal, camouflaged, overhanging, novel obstacles, visual obstructions and an altered floor appearance. Both models were able to detect obstacles reliably in most circumstances. However, overhanging obstacles in particular were shown to pose a safety threat due to being common and difficult to detect even for the depth estimation model.

Future work should address a better detection of overhanging obstacles that are a source of high risk. One possible approach is to add more similar obstacles to the training data of the depth estimator. Drone research, for instance, has resolved this problem. Here, often a single camera is employed for navigation. A model ensemble that combines SeS, DE and feature expansion (Beyeler et al., 2009; Mori & Scherer, 2013) for the upper image region could be a solution. Another promising approach is that of Mancini et al. (Mancini et al., 2018) who argue that object detectors implicitly learn sizes and proportions of objects belonging to the training domain. By utilizing this knowledge, it is possible to create more robust depth estimators.

The two CNNs tested in this paper are not representative. In future work, new OD models should be tested on our here presented application-specific dataset. The results provided in this paper can be used for benchmarking. By reproducing the results on real world data, the simulation should be validated and, if algorithms provide satisfying results in the simulation, fine-tuned.

While training and testing of machine learning algorithms already require large amounts of data, the statistical proof of safety require significantly more data (Kalra & Paddock, 2016). Here, an automated generation of new test scenarios as presented by (Pollok et al., 2019) could be a solution to enable more comprehensive studies.

Acknowledgements This work is part of the research project "AIMFREE" that is funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) on a joint initiative to fund research and development in the field of electromobility (Funding Number: 01MV19002A) and supported by the project management agency German Aerospace Center (DLR-PT). The author is responsible for the content.

Author contributions MW: Conceptualization, Writing – Review & Editing, Supervision; AAB: Methodology, Software, Data Curation, Writing – Original Draft; TA: Writing – Review; PB: Funding Acquisition, Resources.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Adams S., U-Net with Additional Dropout Layers. [Online]. Available: https://github.com/seth814 (accessed: Apr. 22 2021)

Alhashim and P. Wonka (2018) High quality monocular depth estimation via transfer learning Available: http://arxiv.org/pdf/1812.11941v2

Bär, A., et al. (2021). The vulnerability of semantic segmentation networks to adversarial attacks in autonomous driving: enhancing extensive environment sensing. *IEEE Signal Processing Magazine, 38*(1), 42–52. https://doi.org/10.1109/MSP.2020.2983666

Beyeler, A., Zufferey, J.-C. and D. Floreano (2009) Vision-based control of near-obstacle flight

Cordts, M. et al., (2016) The cityscapes dataset for semantic urban scene understanding [Online]. Available: https://arxiv.org/pdf/1604.01685

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017) CARLA: An Open Urban Driving Simulator [Online]. Available: http://arxiv.org/pdf/1711.03938v1

Eykholt E. et al. (2017) Robust Physical-World Attacks on Deep Learning Models [Online]. Available: http://arxiv.org/pdf/1707.08945v5

Feng, Di., et al. (2020). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2020.2972974

Geiger A. and Urtasun R. (2012) Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov R.R. (2012) Improving neural networks by preventing co-adaptation of feature detectors Available: https://arxiv.org/pdf/1207.0580

Hu, J., Zhang, Y. & Okatani T. (2019) Visualization of Convolutional Neural Networks for Monocular Depth Estimation," [Online]. Available: http://arxiv.org/pdf/1904.03380v1

Huang, G., Liu, Z., van der Maaten, L. & Weinberger K.Q. (2016) densely connected convolutional networks [Online]. Available: https://arxiv.org/pdf/1608.06993

Johnson-Roberson, M., Barto, C, Mehta, R., Sridhar, S. N., Rosaen, K. & Vasudevan, R. (2016) Driving in the matrix: can virtual worlds replace human-generated annotations for real world tasks? [Online]. Available: http://arxiv.org/pdf/1610.01983v2

Kalra, N., & Paddock, S. M. (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part a: Policy and Practice, 94*, 182–193. https://doi.org/10.1016/j.tra.2016.09.010

Li, X., Wang, Y., Yan, L., Wang, K., Deng, F., & Wang, F.-Y. (2019). ParallelEye-CS: A new dataset of synthetic images for testing the visual intelligence of intelligent vehicles. *IEEE Transactions on Vehicular Technology, 68*(10), 9619–9631. https://doi.org/10.1109/TVT.2019.2936227

Mancini, M., Costante, G., Valigi, P., & Ciarfuglia, T. A. (2018). J-MOD 2: joint monocular obstacle detection and depth estimation. *IEEE Robot. Autom. Lett., 3*(3), 1490–1497. https://doi.org/10.1109/LRA.2018.2800083

Mori, T. & Scherer S. (2013) First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles

Pollok, T., Junglas, L., Ruf, B., & Schumann, A., et al. (2019). UnrealGT: Using Unreal Engine to Generate Ground Truth Datasets. In G. Bebis (Ed.), *Lecture Notes in Computer Science, Advances in Visual Computing* (pp. 670–682). Springer International Publishing.

Qiu, W. and Yuille, A. (2016) UnrealCV: connecting computer vision to unreal engine Available: http://arxiv.org/pdf/1609.01326v1

Ronneberger, O., Fischer, P. and Brox, T. (2015) U-Net: convolutional networks for biomedical image segmentation [Online]. Available: http://arxiv.org/pdf/1505.04597v1

Ullrich, G., & Albrecht, T. (2019). *Fahrerlose Transportsysteme*. Springer Fachmedien Wiesbaden.

van Dijk T. and G. de Croon (2019) How do neural networks see depth in single images?

Wenning, M., Kawollek, S., & Kampker, A. (2020). Automated driving for car manufacturers' vehicle logistics. *at - Automatisierungstechnik, 68*(3), 222–227. https://doi.org/10.1515/auto-2019-0087

Xu K. et al. (2019) Adversarial T-shirt! evading person detectors in a physical world [Online]. Available: http://arxiv.org/pdf/1910.11099v3