Virtual Reality and Digital System Twins in the Development and Testing of Trainable Highly Automated Driving Decision Making in Shunting Operations

Schäfer, S.[1*]; Yöndem, S. F.[1]; Aliziyad, N.[1]; Cichon, M.[2]

[1]Institute of Vehicle Technology Nuremberg (IFZN),
Nuremberg Institute of Technology (NIT)
Keßlerplatz 12
90489 Nuremberg
[1*]Mail: steffen.schaefer@th-nuernberg.de
Phone: +49 911 5880 1789
[2]Institute of Vehicle System Technology (FAST),
Karlsruhe Institute of Technology (KIT)

Abstract

In the development and testing of highly automated systems, virtual environments such as physics engines and game development frameworks offer a suitable approach for the flexible design of test scenarios [1], [2]–[4]. This paper discusses the application of Unreal Engine 5® (UE5) in testing camera based automated driving functions of shunting operations, i.e., driving on sight. It presents the basic idea of mining training data sets for Computer Vision-based intelligences and highlights the potential of virtual reality in training adaptive algorithms for driving decision making in railroads.

Keywords: Automated Driving, ATO, Virtual Tests, Unreal Engine, Training Data Set, Simulated Sensors, Machine Learning, Computer Vision

1 Introduction

The shift of freight traffic to rail transport systems is further driven by the current policy strategy offensive [5]–[7]. In order to cope with the increasing transport volume and the simultaneous academization of professions, high-performance systems such as Automated Train Operation Systems (ATO) are needed to replace the function of the shunting driver as a driver model. Its feasibility in principle was already demonstrated at the Munich North shunting yard in 2017 [8]. Following this, currently the operational reliability and system safety is investigated. Therefore a holistic scenario-based validation method is under development in order to test these functionalities in a way that is appropriate to the application and relevant to the context [9]. A simulated railway environment is developed to independently serve as raw sensor data sources during both algorithm development and end-of-line testing. For a sustainable and comprehensive mapping of the eventuality spectrum, photorealistic virtual tests are pursued. The simulation environment (LAB) enables the generation of a huge variety of scenes, which is of interest in the context of collecting training data sets for machine learning purposes. Therefore, this paper first analyses the requirements on synthetically generated image data sets (virtual images) within a study on photorealism. Real comparison data sets of basic sensor architectures are collected during field operation and their simulatively emulated counterparts are reconstructed. These comparable data sets (data twins) are fed to different pre-trained object detection and classification algorithms. Their confidence is evaluated on both data sets. It is shown to what extent the image classification Confidence Score (CS) can be increased by means of model-like parameter variations. For the second stage an adaptive algorithm is trained on virtual images rendered using the rail specific UE5-LAB environment. The performance of the artificial intelligence-based object classification is then evaluated on the real world recordings.

Due to increases in rendering performance and the flexibility in scenario and object design, the possibilities for data set creation surpassed limitations mentioned in the literature [10]. Furthermore, developments on frameworks of artificial intelligences, i.e. later versions of neural network-based CV systems are promising and lead to the central research question.

Can virtual images from the virtually rendered railway environment (UE5-LAB) serve as a training dataset for a CV system that should operate in the real world and what are the requirements for the dataset?

2 State of research

The integration of Virtual Reality (VR) into technical processes was demonstrated at the latest in 1993 with the development of a teleoperation system for space activities [11]. Today, the technology is not only used for human-machine-interfaces but also for machine-machine-interfaces, such as described in the following sections.

2.1 Virtual Reality as a machine playground

The migration of VR into the development processes of perception based autonomous systems is known as an established technology from parallel industries [2]–[4]. The basic adaptation of this approach to the railroad sector is demonstrated in [9]. Along a developed tool chain, a virtual closed-loop test bench was developed for scenario-based testing of highly automated shunting functions [12]. The structure of the simulative laboratory test bench for automated systems is shown in Figure 1. Equivalent to the Operational Design Domain (ODD, [13]), different test scenarios (a) are rendered using the UE5 framework (b). This rendering describes the graphical representation of each static and dynamic element of a scene in the field of view [14]. In accordance with the procedures in the field, each movement of the locomotive is initiated by a shunting task (c). Digital models of camera, localization and Light Detection and Ranging system (LiDAR) are integrated to synthesize the respective virtual sensor data stream in the first-person perspective, post-processed and visualized in (d). From there the emulated data is forwarded via ethernet using the sensor specific User Datagram Protocol (UDP) streams. The receiver software element is the middleware of the autonomous System Under Test (SUT). This middleware, running either on laboratory (e) or the target hardware (f), serves as a publisher and subscriber infrastructure for the SUT. The exact data processing depends on the middleware software framework used. For instance, an autonomous system under development [15] is based on the Robot Operating System open source software libraries (ROS), another system currently under test is using the Eclipse zenoh™ protocol.

Within a closed loop test bench, the sensor data is processed by the SUT (e,f), which in turn forwards the driving decision to the low-level control. Represented by a co-simulated model of the vehicle kinematics or dynamics (g), the low-level closes the loop by controlling the velocity of the vehicle inside the UE5 simulation. The feeding of simulated data sets, including an explicit shunting task, LiDAR point clouds, camera and position data in the Universal Transverse Mercator (UTM) format, into automated systems has already been demonstrated in [9].

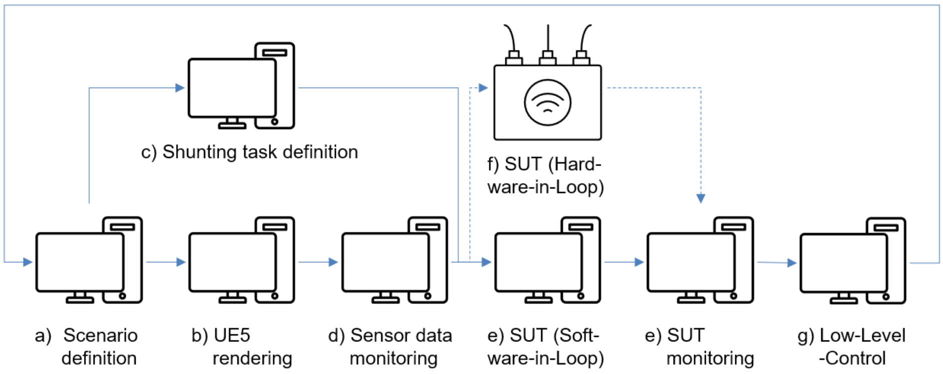

c) Shunting task definition
f) SUT (Hardware-in-Loop)
a) Scenario definition
b) UE5 rendering
d) Sensor data monitoring
e) SUT (Software-in-Loop)
e) SUT monitoring
g) Low-Level -Control

Figure 1: Simulative closed loop test bench for highly automated driving decision making as System under Test (SUT)

2.2 Image based object classification

For obstacle detection on railway tracks the diffusion of machine learning into the autonomous systems is outlined e.g. in [16], [17]. Training neural networks to detect and recognize static railroad signs is presented in [18] and extended to more complex elements such as light signals in [19]. Supervised learning methods require large data sets, whereof each has to be labelled in a manner appropriate to the topic. In [16], the architecture of the YOLOv3 convolutional neural network is applied. The network is trained for five different classes with 7 412 images and tested against 880 images. Hence, a massive data set is required for each class. [16]

For the evaluation of the algorithm different metrics are considered. These start with the precision (P), which is defined as the ratio of true positives of a class to the number of images predicted to show this class. The Recall (R) as the ratio of the sum of true and false positive predictions to the total number of this class. The Intersection over Union (IoU) defines the common ratio of the detected bounding box and the labelled (true) bounding box. A threshold value for the IoU sets the minimum of congruence to be achieved before a classification is done. For each class this threshold can be varied over an interval [min,max]. This leads to a variation of the P and R values. The area below the P-R-curve gives the Average Precision (AP). The average of the AP@[min,max] over all classes gives the mean Average Precision (mAP@[min,max]). [20]

In [14], the improvements in computer graphics and image rendering are investigated. The findings summarize a comparison between the human visual system and a CV system and confirm an engineering benefit in the field of artificial intelligence and object classification. In [21] a CV-based steering angle calculation for automotive applications is presented and tested in CARLA and validated through performing in-field test. In [22], the application of Deep Learning as a new area of machine learning is understood as an approach to

“achieve the imitation of the human brain”

within the control of an aerial vehicle. In order of training the autonomous object detection, image classification and path planning algorithms, AirSim, based on Unreal Engine 4 ® is used to obtain the data set.

In [10], it is stated, that virtually rendered images cannot represent a complete data set for an end-to-end CV based automatic steering angle control system in the real world. It is stated, that this is due to the lack of scenarios on the one hand, and the image quality on the other. In addition, precise manual labelling is time-consuming and therefore expensive.

2.3 Photorealism as key to virtual testing

Photorealism aims to make digital content resemble a photograph rather than mimic human vision. Based on that, it is not the human evaluation of the degree of photorealism that is required, but rather that one of CV systems.

The essential elements for creating photorealistic renderings begin in the lighting. On Earth, the atmosphere modifies sunlight into varying daytime light. Without it, sunlight would be harsh, white, and unvarying. Shadows would be sharp and sunsets abrupt. The atmosphere is responsible for the dynamic, visually engaging daylight we experience. [23].

The uncanny valley concept [24] represents a decrease in positive emotional responses to human-like characters that appear almost, but not completely, real [23]. Audiences appreciate non-realistic computer graphics characters like those in Toy Story [25], but may feel disturbed by characters that closely mimic humans yet possess an indistinguishable non-human quality. The uncanny valley applies not only to characters but also to animated or static elements and entire environments. Although inanimate objects do not evoke the same emotional response, a subtler uncanny feeling occurs when digital images approach photorealism yet lack certain believability.

The challenge is to identify and address the elusive factors that impact photorealism to overcome the uncanny valley. The size of the object in the environment should be accurate. [23]

3 Evaluating photorealism

With the goal of migrating UE5 from the entertainment industry into the engineering process of safety-relevant driving decision-making, the degree of reality simulation becomes relevant. To ensure that both virtually performed tests and virtual images provide reliable and meaningful information, the similarity of the data twins (real/virtual) must be demonstrated. The aim of this chapter, the first step of this research, is to answer the question which elements and settings of a simulated image are crucial for a CV system and which parameters can be used to influence them.

Two different cameras were used for in-field image recording. The according parameters are listed in Table 1. To create realistic and comparable virtual images, the specific characteristics of the cameras were considered for the digital models inside UE5.

Table 1: Camera and lens specifications

Parameter	Basler acA4024-29 µm with a KOWA LM8JC3M2	Sony HDR-HZ1
Resolution in Megapixel	12.2	11.9
Resolution H x V in Pixel	4 024 x 3 036	4 608 x 2 592
Sensor Format in “	1/1.7	1/2.3
Framerate in fps	13.4	29.97
Mono/ Colour	Black/ White	RGB
Lens focal length in mm	8	17.1
f-number	F1.4	F2.8

3.1 Method

For the object detection, trainable neural networks were used due to their high accuracy and adaptability [26]. The algorithms Faster R-CNN, YOLO, SSD, and CenterNet were employed. These algorithms were selected for their ease of implementation using the open-source machine learning framework *Apache MXNet* and the *GluonCV* toolkit, which simplify the development of CV-object detection and classification systems.

The models were trained and tested for the classes *human* and *train* of the COCO dataset [27], a comprehensive object detection, segmentation, and captioning dataset. The performances of the models were evaluated using CSs, which measure the likelihood that the detected object is genuinely the object of interest.

The CS is defined as the product of the objectness score and the class probability of an object detection algorithm. Finally, the visual aspects of the surroundings were improved based on the keys to photorealism discussed in Chapter 2 and further evaluations were conducted. The correct dimensioning for each major object in the simulation was considered. Objects such as trains and signals were first designed in CAD and then adjusted in *Blender* before being imported to UE5. The general test and improvement loop, shown in Figure 2, is run four times.

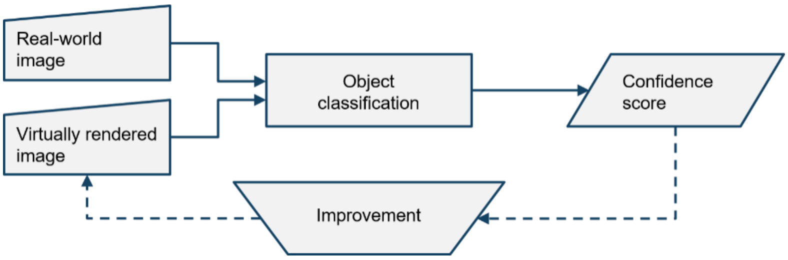


Figure 2: Proposed improvement loop for comparing captured and virtually rendered images

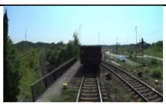
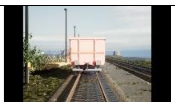
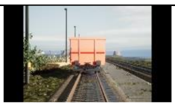
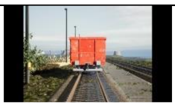
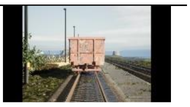
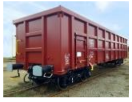
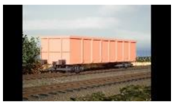
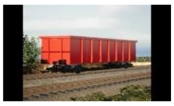
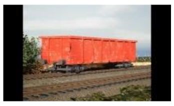
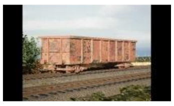
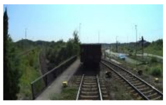
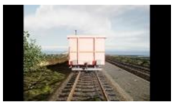
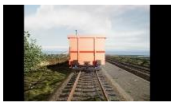
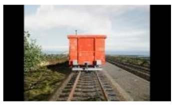
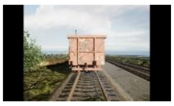
The research on photorealism was divided into two phases. Phase one aimed to study the basic suitability of the proposed object detection method, while phase two focused on examining the level of photorealism and its effect on the CSs.

To identify potential differences in CSs, first the scores on generated images were compared with those on similar real-world images. This step was crucial in order to confirm the viability of the proposed method for comparing the two image sets.

For the human-class, Faster R-CNN showed minimal differences in the CS between real and virtual images, with a mean deviation in thousandths. YOLO on the other hand showed slightly negative average differences, indicating that its CS on virtual humans is in general higher than on the real-world image. Different categories of wagons were tested. The mean difference using Faster R-CNN is ranged between -0.02 and -0.06, meaning the CS on virtual images with wagons is higher compared to the real image. For the same classes, YOLO showed a positive mean difference (0.03-0.08), suggesting a lower CS for virtual images. Out of 30 images, 16 resulted in a CS difference of less than 6 %. Excluding the undetected images, the measurable difference in CS between real-world and virtual images was 40 %. Due to this noticeable deviation two additional algorithms, i.e. Single Shot Detector (SSD) and CenterNet were consulted for the evaluation. The mean of all four algorithms was calculated and employed as a metric in phase two.

For phase two, images captured during in-field recording drives were considered. The test data sets include the presence of humans, steel sheets, dwarf signals, crows and different types of wagons. Four virtual images were considered during each test, using both camera models in two distinct camera positions (0° and 30° local rotation). The different camera positions were used to minimize the influence of using only one object orientation, such as only the front perspective. A test was conducted to assess the impact of different materials on the confidence score of the object detection algorithm for the wagons Eanos-x 056 and Tads 961. Both real-world and virtual images were compared using various materials applied to the wagons. Four types of materials were used in this study: Basic, Automotive, Imperfection and Megascan, each of which differs in the level of detail and thus appearance. The comparison of the according confidence scores for both cameras in each position is listed in Table 2.

Table 2: Comparison of the mean confidence score of four object detection algorithms for an Eanos-x 056 wagon with different materials, cameras and perspectives

	Real-world	Basic	Automotive	Imperfection	Megascan
Sony, Position 1					
Metric	0.85	0.36	0.29	0.75	0.28
Sony, Position 2					
Metric	0.66	0.47	0.67	0.85	0.45
Basler, Position 1					
Metric	0.85	0.36	0.24	0.79	0.19

Basler, Position 2					
Metric	0.66	0.78	0.92	0.92	0.83
Overall mean	0.76	0.49	0.53	0.83	0.44

The overall mean provides the average metric value for each material type across both camera models and positions. The imperfection material showed the highest overall mean (0.83), indicating the best average performance among the materials. In contrast, the Megascans material showed the lowest overall mean (0.44), indicating the poorest average performance. For the Tads 961 wagon the highest overall mean CS of 0.70 was achieved when using the Megascans material, indicating the best performance among all tested materials. Conversely, the imperfection material yielded the lowest overall mean CS (0.53), suggesting the weakest performance on average. It is essential to recognize that both materials were downloaded from Megascan, but the material types and textures used differed. The reason for the scatter is still the subject of ongoing research. The basic and automotive materials demonstrated similar mean CSs, ranging from 0.49 to 0.59.

The subsequent test cases evaluated the impact of surrounding changes on the object classification algorithms. Four different environments were designed, which included the presence of a bush, the use of different grass materials, placing the object on grass without a rail track, and positioning the wagon on a road. It was observed that the presence of a bush caused the shadow of the wagon to appear larger. Using the starter content grass material resulted in lower photorealism in the environment, as the grass lacked imperfections and appeared excessively green. Positioning the wagon on grass without a rail track led to a reduced CS, as trains are typically found on rail tracks. Further scenarios inspected the effect of changing materials of objects or objects themselves in the background. The impact of altering the material of a background building was investigated. The results showed that changing the background material had a minimal effect on the CS, with changes ranging between -1% to 2%. Inspection of the effect of daytime and the according lighting (sunlight, dusk, night, lamp illumination during night) showed, that the period directly after sunset had the lowest mean CS. This was due to the absence of sunlight and lamps were not turned on. After the lamps were turned on, the mean CS increased and ranged from 0.43 to 0.59. As the sun becomes visible the mean CS increased to a range of 0.84-0.89.

In Phase one, pre-trained neural networks showed similar results on real and virtual images, with a resulting detection rate of 97%. In Phase two, the virtual images resulted in a lower detection rate: out of 316 total images, 90 objects remained undetected yielding a 72% detection rate. It is also to consider that some images in Phase two were expected to be undetected, such as wagons occluded by bushes or wagons on roads.

3.2 Conclusion on photorealism

The study on photorealism should highlight the potential benefits of virtual training datasets and evaluate whether their application is advantageous for the current use case, such as automated shunting. As a representation of the autonomous system, we used four object detection algorithms: Faster R-CNN, YOLOv3, SSD, and CenterNet, which were trained on real-world images from the COCO dataset. Our method however did not measure the photorealism of an image directly as it was not the objective of the study. Instead, it relied on the CSs of object detection algorithms to compare virtual images to similar real-world images.

The impact of the surrounding and lighting was categorized. Overall, the findings suggest that CV systems are able to detect objects known from the real world in rendered images. Furthermore, it is shown, that material selection can significantly influence the performance of object detection algorithms. The study concludes with the motivation to reverse the methodical approach and design a virtual training dataset for a supervised learning algorithm for real-world use.

4 Design of a virtual image training data set for in-field object classification

The next step was to replace the generic COCO dataset with a training dataset specific to a particular use case. Therefore, virtual images obtained from the photorealistic simulation environment were used. This data set aims to serve for the training of a deep learning algorithm for signal detection and classification. The validity of the virtual training data is shown using the example of the dwarf signal, as shown in Figure 3, in each case for white (Sh1, go) and red (Hp0, stop). Following up on the research presented in [20] where the YOLOv5 deep network is proposed for object detection and recognition, this research uses the improved later version YOLOv8.

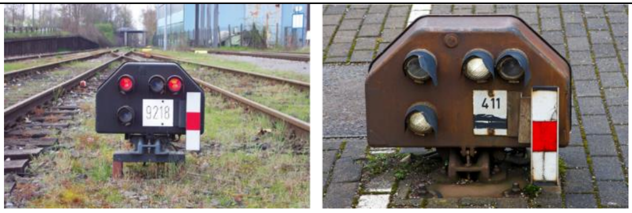


Figure 3: Dwarf signal, Hp0 red (left) and Sh1 white (right)

In this section, initially the requirements on training neural networks are outlined. The chosen network is trained in two stages, in order to demonstrate the impact of the scenario design. Afterwards the integration of the classifier into the virtual test bench is shown and the live object classification is demonstrated. In the final step, recordings from the field are used to test the classifier on 345 real world images for both detail levels presented in the sections 4.2 and 4.3 respectively.

4.1 Requirements on training data sets and general processing

According to the suggestions of the developer, the later, advanced data set was created along the following instructions. First and foremost, appropriate diversity is required. For instance, the weather distribution observed in the field is relevant. For this purpose, the weather report of the target marshalling yard recorded during the last years was considered. In addition, consistency with the application scenario is required. This means that the operation environment of the CV system was designed with typical elements such as tracks, rail vehicles, buildings and people wearing high-visibility vests. Third, class balance is required. This means that training data of at least similar quantity has to be available for each respective class to be recognized.

The training data set was exported from UE5. Before feeding the images to the YOLOv8 algorithm, the classes were labelled. For this purpose, the software tool *Make Sense AI* was used. Every class appearing in a single image was framed by a bounding box, designed as tight as possible and assigned to the corresponding class.

In order to increase the variety of the images and to simulate disturbances a step called data augmentation was carried out using the python library *albumentations*. The augmentation was reviewed using the *pybboxes* library. The images were saved in .jpeg format along with the according label document containing the coordinates of the bounding boxes and the class description in a .txt file, both in the same folder. The augmentation techniques used are rotation (between -30° and +30°), brightness (between

-15% and +15%), blur (up to 5 pixels) and noise (up to 8% of pixels). After sorting out irrelevant images, the data augmentation more than doubled the data set.

4.2 Model One – Simple data set

The first training data set was set up with 800 images in a variety of weather and lighting conditions, such as shown in Figure 4. The data set was extended using the described data augmentation techniques to 1 916 images in sum. The elements used include academic examples and do not represent the railroad guidelines conformity yet. In this stage elements from the periphery such as buildings, vehicles etc. were not considered.

For the training process the initial (lr0) and final (lr1) learning rate were set to the default value of 0.01 each. The image size was set to 640 x 640 pixels, the number of images per batch to two, IoU threshold to 0.5 and the patience to 50. The model used is the large one, i.e. *Yolov8l.pt*. The images were split for training/test/validation purpose as 80/10/10.

The entire training took 424 iterations, processed in about 20 hours using a Nvidia Titan Xp graphics card. After finishing the training, the mAP value is 0.92. The bounding box loss is 0.41 and class loss is 0.25. A low box loss value indicates a high IoU.

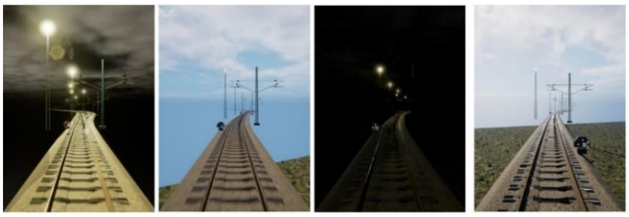


Figure 4: Excerpt of the virtual training images

4.3 Model Two – Advanced data set

In the second stage of testing the integration of YOLO into the object detection and classification process, the training data set was set up closer to the requirements given in section 4.1. In order to measure the impact of the training data, the settings of the training process were maintained.

The data set from model one (section 4.2) was extended to 3 897 images (1 655 without the augmentation) focusing on the following details. The simulation of different surroundings, regarding different illumination conditions, camera perspectives and angles, realistic scenarios including locomotives, trees and florals, terrain, rail track colors etc. such as shown in Figure 5 was applied.

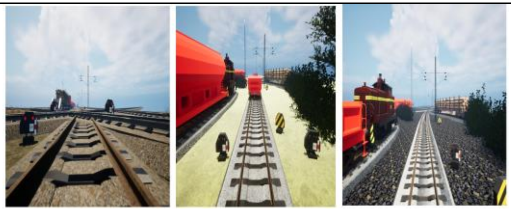


Figure 5: Examples of a coherent and realistic environment for a more advanced and comprehensive training data set

Compared to the first training, the advanced training data set resulted in a higher accuracy of the classificatory, shown in Figure 6. The mAP value increased from 0.92 to 0.95, the bounding box loss decreased from 0.41 to 0.38 and the class loss decreased from 0.25 to 0.22, proving that the extension of the dataset has a positive effect on the accuracy of the model. The abbreviations (train and val) in Figure 6 refer to the data set used for training and validation during the training process. The graphs show typical asymptotic behavior. For the visualization of the training progress the training iterations, the so-called epochs, are plotted on the abscissa, the respective value on the ordinate. On the left side the improvement on the IoU (decreasing box loss) is shown on both, training (above) and validation data (below). A similar but faster evolution is observed for the right identification of a certain class in an image (decrease of cls_loss). Furthermore the evolution of the metrics P, Rm mAP@50 as well as the mAP@[50;95] is printed over the epochs. It can be seen, that the quality of the chosen neural network quickly improved in the beginning and came to a saturation after a few iterations. When comparing the evolution of the mAP50 and the mAP@[50;95] it becomes clear, that the higher confidences coming in line with higher IoU thresholds take more training iterations.

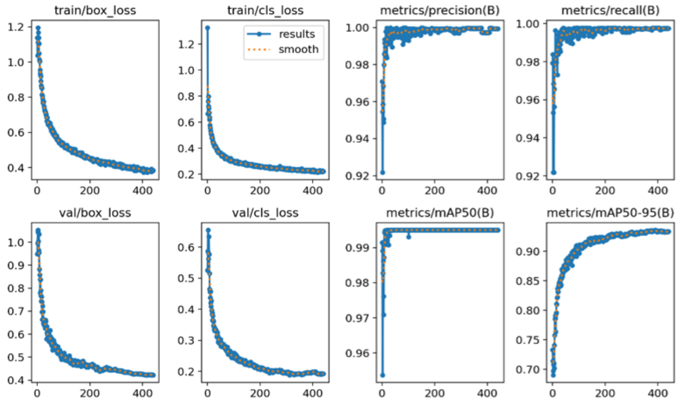


Figure 6: Result of the second training run of Yolov8 using virtual images only

4.4 Integration of YOLO into the virtual test bench

After completing the training, the virtual test bench, presented in section 2.1, was used to validate and demonstrate the CV-based object detection and classification. Therefore, the images captured by the camera model inside UE5 were streamed out by UDP to a local PC running a Python script. After setting up the network interface the packages are reassembled to images in the RGBA format. These were then converted into the RGB format and fed frame by frame to the CV system based on YOLOv8. The script live draws a bounding box around detected objects and classifies them as either red or white dwarf signal and annotates the according CS. This gives a simple approach to live monitor the decision making of artificial intelligences in driving decision making.

4.5 Testing results on real data

The main intension of this research was to verify a neural network trained on virtual data for object detection and classification on real-world data. Therefore, 345 images captured during in-field recordings were considered. The results of the comparatively high scores for both virtual test runs of the earlier sections can serve as a reference metric. In Table 3, the results for both algorithms, based on the simple (Model One) and advanced data set (Model Two) are listed. Thereby, the metrics are given for both, virtually generated images from the LAB environment and the real-world validation images from the FIELD. The table also includes the results of a further experimental model trained on the exact same parameters and requirements as in Test Two, but using a higher resolution of the images (1 280 x 1 280 Pixel).

Table 3: Results of the neural network algorithms trained on virtual images and tested on real world data (FIELD) and generated virtual data (LAB)

Model	mP@0.5	mAP@[0.5, 0.95]	mR@0.5	Accuracy	F1 Score	Inference time in ms
One (LAB)	0.995	0.92	1.00	1.00	1.00	11.6
One (FIELD)	0.593	0.3	0.474	0.424	0.56	11.6
Two (LAB)	0.995	0.943	0.999	0.99	0.99	11.6
Two (FIELD)	0.968	0.705	0.951	0.93	0.96	11.6
Exp. (LAB)	0.994	0.941	0.998	0.99	0.99	43.1

Exp. (FIELD)	0.912	0.668	0.856	0.899	0.87	43.1

The simple training data set of Model One led to very good results (scores > 0.9) when tested on virtual tracks in the LAB environment. The scenes were simple and no other objects included but the track and the signals. When applying the same CV model to the FIELD data, it becomes obvious, that the confidences are comparatively low with an mAP of about 30 %. Model Two, included a data set with more details on the railway environment. The impact on the mAP on LAB data is recordable, the improvement on FIELD data more than doubled the score to over 70 %. Furthermore, the mean recall more than doubled from Model One to Two on FIELD images. The improvement of the system confidence can also be seen in the confusion matrix, shown in Figure 7.

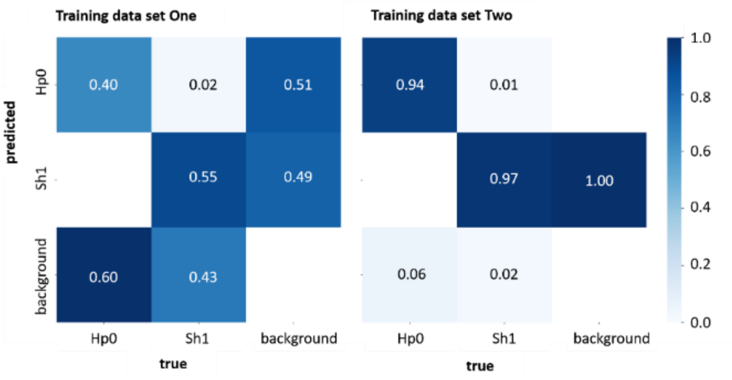


Figure 7: Confusion matrix. Based on the training data set of test one (left) and test two (right). Validated on the same real-world data.

The confusion matrix represents the actual true class on the abscissa, and the predicted class on the ordinate. In Figure 7, the results are presented for the CNN-based CV-System trained on the simple data set (left) and on the advanced data set (right). The improvement on the CV system trained on the advanced data set is indicated by the high scores of true/true values. Using model One, only 40 % of Hp0 signals were correctly detected, whereas model Two showed 94 % correct results on that class. Wrong predictions of 60 % with model One lowered to a tenth of that with model Two. The Sh1 signal was correctly detected in 55 % of cases using model One and increased to 97 % with model Two. Furthermore, it can be noted, that less false positives were detected with model Two. Compared to the virtual control quantity, the mAP on real data is still lower but improvements on the training data set promise a further enhancement of the YOLO classifier. At least the advanced training data set showed CSs of above 87% which is estimated as good.

5 Conclusions and future work

In this paper the utilization of convolutional neural networks for railway specific objects was demonstrated. It is shown that the COCO training data gives a point to start research about virtual training data for field applications. Furthermore, extracting training data from the scenario simulator as described in [9],[12] is demonstrated. It is shown, that an improvement on background details in the training data set has a positive impact on the classification and the accompanied CS. The integration of YOLO as an object classification algorithm into the virtual test bench is demonstrated. Furthermore, it is shown, that virtually rendered training data can be used for object detection and classification in the field.

Future work will investigate decisive details in order to improve the mAP score. Parallelly the approach is rolled out to other sensors such as LiDARs. For a more efficient training the data generation including the required and appropriate labelling is examined with regard to its automatability. The research team is aiming to develop a school bench for autonomous systems.

Acknowledgement

Chapter two of this paper is based on results of the cooperative research project VAL of the consortium partners DB Cargo AG and the NIT, funded by the Federal Ministry of Digital Affairs and Transport; FKZ 5320000013, EBA Az. 8fd/003-1255#008-VAL.

Literature

[1] Pegasus, „Schlussbericht für das Gesamtprojekt PEGASUS: Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen“, Jan. 2020.

[2] K. Neumann-Cosel, „Virtual Test Drive“, München, 2014.

[3] W. Guerra, E. Tal, V. Murali, G. Ryou, und S. Karaman, „FlightGoggles: Photorealistic Sensor Simulation for Perception-driven Robotics using Photogrammetry and Virtual Reality“, in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China: IEEE, Nov. 2019, S. 6941–6948. doi: 10.1109/IROS40897.2019.8968116.

[4] S. Shah, D. Dey, C. Lovett, und A. Kapoor, „AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles“, in *Field and Service Robotics*, M. Hutter und R. Siegwart, Hrsg., in Springer Proceedings in Advanced Robotics, vol. 5. Cham: Springer International Publishing, 2018, S. 621–635. doi: 10.1007/978-3-319-67361-5_40.

[5] SHIFT2RAIL, „SHIFT2RAIL STRATEGIC MASTER PLAN“, Jan. 2014. [Online], Accessed on August 14th 2023: https://shift2rail.org/wp-content/uploads/2021/06/Shift2Rail-Master-Plan_approved-by-S2R-GB.pdf

[6] RAILFREIGHT FORWARD EUROPEAN RAILFREIGHT VISION 2030, „30 by 2030: Rail Freight strategy to boost modal shift“, Jan. 2021. [Online], Accessed on August 14th 2023: https://www.railfreightforward.eu/sites/default/files/usercontent/white_paper-30by2030-150dpi6.pdf

[7] UN Climate Technology Centre & Network, „Modal shift in freight transport“, 1. Januar 2017. [Online], Accessed on August 14th 2023: https://www.ctc-n.org/technologies/modal-shift-freight-transport

[8] M. Cichon und R. Schaal, *Vollautomatische Abdrücklokomotive - Machbarkeitsstudie und Aufbau eines Demonstrators*. 2018.

[9] S. Schäfer, L. Greiner-Fuchs, T. Hofmeier, und Cichon, Martin, „Entwicklung eines echtzeitfähigen virtuellen Laborprüfstands (Simulationsumgebung) für das szenariobasierte Testen und Validieren hochautomatisierter Fahrentscheidungs-und Steuerungssysteme von Rangierlokomotiven“, in *19. Internationale Schienenfahrzeugtagung Dresden*, März 2023, S. 108.

[10] W. Yuan, M. Yang, C. Wang, und B. Wang, „VRDriving: A Virtual-to-Real Autonomous Driving Framework Based on Adversarial Learning“, *IEEE Trans. Cogn. Dev. Syst.*, Bd. 13, Nr. 4, S. 912–921, Dez. 2021, doi: 10.1109/TCDS.2020.3006621.

[11] S. Wenzel und H. J. Claßen, „Die Anwendung von Virtual Reality bei Telerobotik“, in *Virtual Reality*, H. J. Warnecke und H.-J. Bullinger, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, S. 259–269. doi: 10.1007/978-3-642-88650-8_20.

[12] L. Greiner-Fuchs, S. Schäfer, T. Hofmeier, und M. Cichon, „Database-supported methodical approach for the development of a toolchain for the evaluation of ATO functions using a scenario-based test methodology“, in *Proceedings of the Fifth International Conference on Railway Technology: Research, Development and Maintenance*, in Paper 13.4. Montpellier: Civil-Comp Press, 2022. doi: 10.4203/ccc.1.13.4.

[13] British Standards Institution (BSI), *Operational design domain (ODD) taxonomy for an automated driving system (ADS). Specification*. London, United Kingdom: BSI Standards Limited, 2020.

[14] A. Singh, M. Kumar, und A. Saxena, „Analysis of Computer Vision for Graphics and Animation“, in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India: IEEE, März 2023, S. 804–807. doi: 10.1109/ICACCS57279.2023.10112718.

[15] M. Cichon und R. Falgenhauer, „Integrales ATO-System für Rangier-Aufgaben“, in *Tagungsband Rad-Schiene-Tagung 2023*, Dresden, März 2023.

[16] R. M. Prakash, M. Vimala, S. Keerthana, P. Kokila, und S. Sneha, „Machine Learning based Obstacle Detection for Avoiding Accidents on Railway Tracks“, in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India: IEEE, Mai 2023, S. 236–241. doi: 10.1109/ICICCS56967.2023.10142837.

[17] R. Sattiraju, J. Kochems, und H. D. Schotten, „Machine learning based obstacle detection for Automatic Train Pairing“, in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, Trondheim, Norway: IEEE, Mai 2017, S. 1–4. doi: 10.1109/WFCS.2017.7991962.

[18] S. Mikrut, Z. Mikrut, A. Moskal, und E. Pastucha, „Detection and Recognition of Selected Class Railway Signs“, *Image Processing & Communications*, Bd. 19, Nr. 2–3, S. 83–96, Sep. 2014, doi: 10.1515/ipc-2015-0013.

[19] G. Karagiannis, S. Olsen, und K. Pedersen, „Deep Learning for Detection of Railway Signs and Signals“, in *Advances in Computer Vision*, K. Arai und S. Kapoor, Hrsg., in Advances

in Intelligent Systems and Computing, vol. 943. Cham: Springer International Publishing, 2020, S. 1–15. doi: 10.1007/978-3-030-17795-9_1.

[20] A. Staino, A. Suwalka, P. Mitra, und B. Basu, „Real-Time Detection and Recognition of Railway Traffic Signals Using Deep Learning“, *J. Big Data Anal. Transp.*, Bd. 4, Nr. 1, S. 57–71, Apr. 2022, doi: 10.1007/s42421-022-00054-7.

[21] S. Shafique, S. Abid, F. Riaz, und Z. Ejaz, „Computer Vision based Autonomous Navigation in Controlled Environment“, in *2021 International Conference on Robotics and Automation in Industry (ICRAI)*, Rawalpindi, Pakistan: IEEE, Okt. 2021, S. 1–6. doi: 10.1109/ICRAI54018.2021.9651414.

[22] S. Wang, J. Chen, Z. Zhang, G. Wang, Y. Tan, und Y. Zheng, „Construction of a virtual reality platform for UAV deep learning“, in *2017 Chinese Automation Congress (CAC)*, Jinan: IEEE, Okt. 2017, S. 3912–3916. doi: 10.1109/CAC.2017.8243463.

[23] E. Dinur, *The Complete Guide to Photorealism: For Visual Effects, Visualization and Games*, 1. Aufl. New York: Routledge, 2021. doi: 10.4324/9780429244131.

[24] M. Mori, K. MacDorman, und N. Kageki, „The Uncanny Valley [From the Field]“, *IEEE Robot. Automat. Mag.*, Bd. 19, Nr. 2, S. 98–100, Juni 2012, doi: 10.1109/MRA.2012.2192811.

[25] M. Henne, H. Hickel, E. Johnson, und S. Konishi, „The making of Toy Story [computer animation]“, in *COMPCON ’96. Technologies for the Information Superhighway Digest of Papers*, Santa Clara, CA, USA: IEEE Comput. Soc. Press, 1996, S. 463–468. doi: 10.1109/CMPCON.1996.501812.

[26] N. O’Mahony *u. a.*, „Deep Learning vs. Traditional Computer Vision“, in *Advances in Computer Vision*, K. Arai und S. Kapoor, Hrsg., in Advances in Intelligent Systems and Computing, vol. 943. Cham: Springer International Publishing, 2020, S. 128–144. doi: 10.1007/978-3-030-17795-9_10.

[27] T.-Y. Lin *u. a.*, „Microsoft COCO: Common Objects in Context“, 2014, doi: 10.48550/ARXIV.1405.0312.

Assignment of thematic focus: The proposed contribution topic is basically assigned to highly automated (assisted/automated/autonomous - 3A) rail freight transport, but is transferable to all automatable track-guided driving processes on sight. The presented virtual or digital tools are already used for (simulated) automated dispatching and can provide valuable insights in incident management. The goal of the work is to increase the competitiveness of rail freight transport and contributes to the focus on mobility management in perspective.

Author

Schäfer, Steffen

Research associate at the Institute of Vehicle Technology at Nuremberg Institute of Technology. Since 2014, conducting research in international teams on automations and innovations to improve the sustainable mobility situation. Studied aerospace engineering at RWTH Aachen University and researches the influence of virtual systems on the diffusion of artificial intelligences into the driving decision making of railroad systems.