The validation of simulation for testing deep learning-based object recognition

Ella P. Fokkinga[a], Merel E. te Hofsté[a], Richard J.M. den Hollander[a], Remco van der Meer[a], Frank P.A. Benders[a], Frank B. ter Haar[a], Veronique E. Marquis[a], Melvin van Berkel[a], Jeroen M. Voogd[a], Thijs A. Eker[a], and Klamer Schutte[a]

[a]TNO - Defense, Safety and Security, Oude Waalsdorperweg 63, the Hague, the Netherlands

ABSTRACT

The military is looking to adopt artificial intelligence (AI)-based computer vision for autonomous systems and decision-support. This transition requires test methods to ensure safe and effective use of such systems. Performance assessment of deep learning (DL) models, such as object detectors, typically requires extensive datasets. Simulated data offers a cost-effective alternative for generating large image datasets, without the need for access to potentially restricted operational data. However, to effectively use simulated data as a virtual proxy for real-world testing, the suitability and appropriateness of the simulation must be evaluated. This study evaluates the use of simulated data for testing DL-based object detectors, focusing on three key aspects: comparing performance on real versus simulated data, assessing the cost-effectiveness of generating simulated datasets, and evaluating the accuracy of simulations in representing reality. Using two automotive datasets, one publicly available (KITTI) and one internally developed (INDEV), we conducted experiments with both real and simulated versions. We found that although simulations can approximate real-world performance, evaluating whether a simulation accurately represents reality remains challenging. Future research should focus on developing validation approaches independent of real-world datasets to enhance the reliability of simulations in testing AI models.

Keywords: Deep learning; Simulation; Verification and validation; Object detection; Synthetic-to-real gap

1. INTRODUCTION

Artificial Intelligence (AI)-based computer vision methods are becoming increasingly important for military, automotive, and other high-risk applications. Especially during operational use in a military context, vast amounts of sensor data are coming in, from both manned and unmanned platforms. Handling such a large data volume requires automated analysis methods for autonomous and/or decision-support systems. Deep Learning (DL) has emerged as the most effective methodology for computer vision tasks such as object detection, target identification, and tracking. DL models rely strongly on the availability of large datasets for both *training* and accurately *testing* their performance and reliability under diverse operational conditions.

The acquisition of large and varied datasets for military purposes is challenging. The restricted nature of military environments often limits data availability. Secondly, the landscape of military engagement is constantly evolving with new threats, technologies, and tactics. In addition, the models must perform in various environments, terrains, and weather conditions, depending on the operational domain. These factors complicate the gathering and maintenance of datasets that represent the required operational conditions. Moreover, the costs of live exercises, including logistics, equipment, and personnel, can be high and obtaining datasets under the required environmental conditions is challenging.

The use of simulated data could address these challenges.[1–6] It offers a cost-effective method to generate large datasets of images, without requiring access to actual operational recordings. Simulation provides full control over object and scene characteristics, allowing for rapid adaptation to new circumstances and exposing DL models to diverse conditions tailored to specific operational requirements.

Corresponding author: Ella P. Fokkinga. E-mail: ella.fokkinga@tno.nl

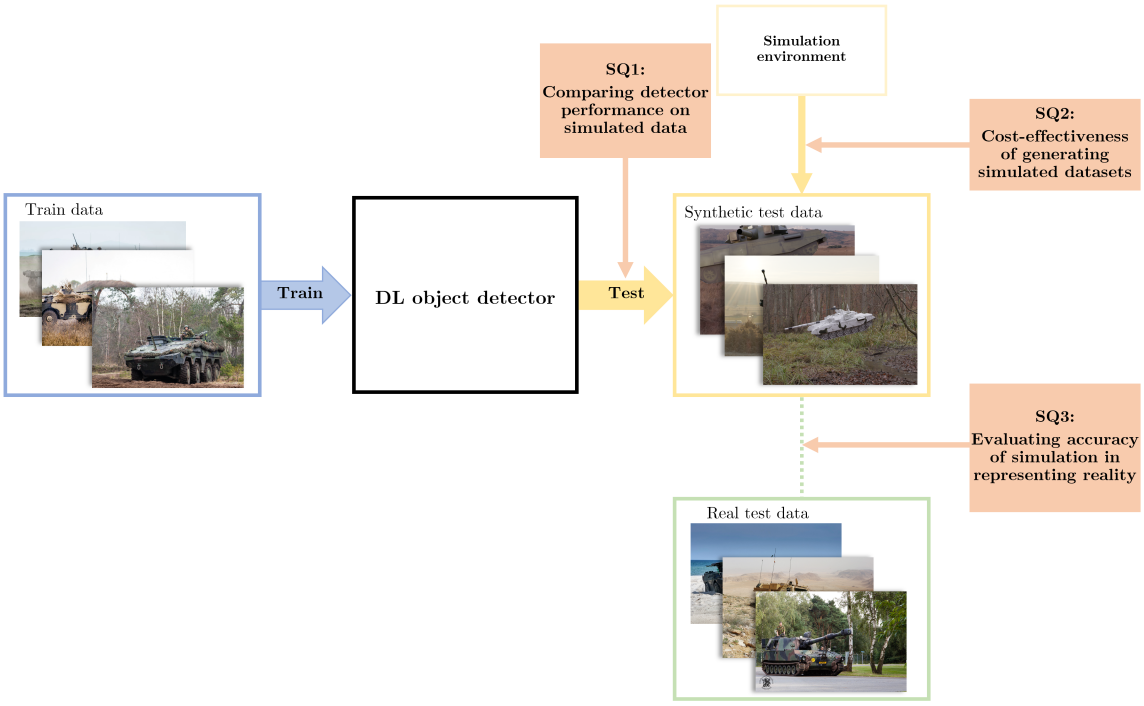


Figure 1: Overview of the three key aspects of using simulated data in the testing phase of a DL-based object detector addressed in this study. Firstly (SQ1): assessing how the object detector's performance on simulated data compares to its performance on real-world data under varying conditions. Secondly (SQ2): evaluating the cost-effectiveness of generating simulated datasets. Finally (SQ3): evaluating how accurately the simulation environment represents the real-world, without depending on large, annotated real-world datasets. Note that while examples images of military vehicles are used in this figure, we selected an automotive use-case for this study.

The use of simulated data for development of AI-models is a well-researched topic, particularly in the training phase.[1,2,5–14] However, for simulated data to function as a virtual proxy for real-world data in the testing process of the DL model, its suitability and appropriateness must be evaluated. Research on using simulation during the testing phase of a model is less extensive. In this paper, we examine the following research question: *"How can simulated data be effectively used in the testing phase of a DL-based object detector?"* This question is further divided into three sub-questions, as illustrated in Fig. 1:

- **Sub-question 1 (SQ1):** *"How does the object detector's performance on simulated data compare to its performance on real-world data under varying conditions?"* This question focuses on evaluating the object detector's performance across different conditions, in both simulated and real-world datasets, examining how well the simulation follows the real-world performance trends.
- **Sub-question 2 (SQ2):** *"How cost-effective is it to generate a simulated dataset that is suitable for testing an object detector?"* This question addresses the efficiency and practicality of creating a simulated dataset that meets the requirements for testing.
- **Sub-question 3 (SQ3):** *"How can we evaluate the accuracy of a simulated environment in representing reality, without relying on large, annotated real-world datasets?"* This question explores methods to validate the realism and representativeness of the simulated environment, focusing on broader patterns and behaviors expected in real-world data beyond just performance metrics.

It is important to note that, in this work, and especially for *SQ3*, the emphasis is on similarity in terms of performance, not necessarily on other aspects of similarity. To answer the research questions, we chose an

automotive DL-based object detection use-case. In this setting, we apply the object detector to recognize vehicles under varying circumstances, using both real and simulated data. By analysing the performance difference of the object detector under different conditions and comparing the results between real and simulated data, we investigate what the simulation reveals about performance on real-life imagery (*SQ1*). To understand the costs-effectiveness of simulating a dataset (*SQ2*), we do not only use existing public datasets, but also simulate an automotive dataset internally. Finding a method to test whether the simulation represents reality is difficult, especially without a large, annotated, real dataset for comparison. Therefore, we explore other patterns in the model behavior that could indicate the reliability of our tests on the simulation (*SQ3*).

2. RELATED WORK

2.1 The use of simulation in the testing phase of AI models

The use of simulated data for training AI-models is a well-researched topic,[1,2,5–14] far more so than for the testing phase. While the benefits of using simulation for training are recognized, it is challenging to determine whether models perform similarly on simulated data as they would on real-world data. Rosenzweig et al. (2021)[15] address this by presenting a novel framework to measure and validate the transferability of testing results from simulated to real-world data, aligning closely with our study's objectives. Their work focuses on semantic segmentation models for autonomous driving, using a generative label-to-image synthesis model to simulate data from real-world labels. Their approach includes three types of transferability measures: correlation of performance metrics, error distribution analysis, and a discriminator model to distinguish real and simulated data. While this framework is effective in assessing effectively to which extent testing results on simulated data reflect real-world performance, it still relies on the availability of a real dataset for validation.

Other studies have explored specific aspects of the testing phase where using simulated data can be effective. For instance, simulated data is relevant for stress-testing, where an algorithm is pushed to its limits to identify the conditions under which it begins to fail. Pérez-García et al. (2023)[16] used simulated data to identify and address shortcomings in biomedical imaging models, which often underperform due to dataset shifts and spurious correlations. The data was simulated via generative image editing using text-to-image diffusion models to modify biomedical images in a controlled and realistic manner. In a military context, stress-testing could involve testing DL models under extreme conditions that are rare or too dangerous to replicate during real exercises.

Furthermore, simulated data allows for testing against distributional shifts. Breugel et al. (2024)[6] explored the generation of simulated data conditional on shift information, to quantify a model's sensitivity to distributional shifts. The study evaluated scenarios with and without prior knowledge on the shift. In military operations, environmental factors such as weather and terrain, or new technologies like advanced camouflaging techniques and stealth materials, can change the operational landscape. It is important to validate that significant changes in operational data compared to the training data have a minimal effect on model performance.

2.2 Automotive use-case

For this study, we selected an automotive use-case, because it is a well-established field for applying simulations in AI model training, offers literature on simulation-based testing, and there is availability of datasets with both real and corresponding simulated data. In the Virtual KITTI paper by Gaidon et al. (2016),[17] the authors developed a method to (semi-)automatically generate simulated datasets based on real-world data for various computer vision tasks in both training and testing. They produced five sequences of 'synthetic clones', closely mimicking real-world data from the original KITTI Vision Benchmark Suite,[18] an annotated database consisting of images captured by a driving car in Karlsruhe. This virtual dataset allowed them to demonstrate that DL algorithms pre-trained on real data behave similarly in real and virtual worlds by comparing multiple tracking metrics, concluding that it can function as a proxy for testing multiple object tracking algorithms. In 2020, a successor, the Virtual KITTI 2 dataset, was released with more photo-realistic features due to upgrades in the Unity game engine.[19] Re-running experiments with the Virtual KITTI 2 dataset showed that the performance gap for multiple object tracking between real and virtual data remained small. As part of our experiments, we will use the real and the Virtual KITTI 2 'clone' dataset.

In 2022, Sun et al. introduced SHIFT, another large automotive synthetic dataset developed for autonomous driving.[20] It comprises over 70 hours of driving and 2.5 million annotated frames. The paper focuses on modelling

both discrete and continuous domain shifts to evaluate an AI model's robustness. Domain shifts include weather conditions, time of day, traffic levels, and camera orientation. Unlike Virtual KITTI, SHIFT does not have a direct one-to-one correspondence with real-world imagery. Instead, the real-world BDD100K database,[21] a large annotated set with various weather conditions, is used for overall comparison. The study found similar trends in mean average precision (mAP) degradation under various domain shifts for both SHIFT and BDD100K, suggesting that the simulated dataset is consistent with the real dataset. However, object detection models performed best in overcast scenes for real-world data, while in SHIFT, the best performance was in clear images. Additionally, mAP values for SHIFT were slightly higher than for BDD100K, indicating that simulated data might be less noisy or too controlled. Due to the absence of an one-to-one corresponding real dataset, we decided not to use the SHIFT dataset.

2.3 Performance degration under adverse conditions

Our study focuses on an automotive use-case of vehicle detection performance under varying conditions, a prevalent topic in literature.[22–28] Object detectors often show decreased performance in adverse weather conditions or other scenarios they were not specifically trained on.[24] Research has shown that training models in diverse weather conditions can improve robustness. For instance, Rothmeier et al. (2023)[27] demonstrated that incorporating simulated adverse weather datasets during training can enhance detection performance.

3. METHODS

To address the defined research questions (Fig. 1), we conducted a series of experiments using two datasets, with both a real-world and its corresponding simulated variant. This approach allows us to systematically compare and analyze the performance and reliability of the DL-based object detector under different conditions.

3.1 DL-based object detection

We evaluated the performance of two DL-based object detectors, specifically two versions of YOLOv8 pretrained on the MS-COCO dataset,[29] focusing on three vehicle classes: car, truck, and bus. The evaluations were executed across various scenes and under different weather and lighting conditions.

Performance degradation or changes are expected when a detector operates under conditions different from those it was trained on. Given that YOLOv8 was only pretrained on the MS-COCO dataset and not specifically on adverse weather conditions or difficult urban scenes, we hypothesized it would exhibit performance differences under these varied conditions. This evaluation aims to demonstrate the correspondence in performance estimation between real and simulated data, addressing *SQ1: "How does the object detector's performance on simulated data compare to its performance on real-world data under varying conditions?"*.

Our objective is to show that DL object detectors' performance varies under different conditions in the real world and then assess whether the simulation represents the reality by observing if a similar pattern occurs in the simulated environment. However, our real datasets do not include very extreme conditions, such as rainstorms or heavy fog, so significant performance changes might not be observed. To address this, we used both a larger YOLO model (YOLOv8l) and a smaller one (YOLOv8n). Under mild adverse conditions, we expect a fairly low performance degradation for the larger model. In contrast, the smaller model is expected to show a more noticeable decrease in performance, helping us determine if similar performance degradation patterns occur in both real and simulated environments.

Both the YOLOv8l and YOLOv8n models were applied with a confidence threshold of 0.1 for the relevant MS-COCO vehicle classes.

3.2 Datasets

We used a publicly available simulated dataset but also created our own simulated dataset internally. This dual approach not only provides more data for our experiments but also helps us gather insights into the costs and benefits of simulating data. The process of simulating our dataset is described within Section 3.2.2. Insights gained by going through this process will be used in the discussion to answer *SQ2: "How cost-effective is it to generate a simulated dataset that is suitable for testing an object detector?"*.

Table 1: The five scenes of the KITTI dataset[18] for which a simulated variant is available in the virtual KITTI 2 dataset.[19]

Scene	Description
01	crowded urban area
02	road in an urban area and a busy intersection
06	mostly stationary camera at a busy intersection
18	long road in the forest with challenging imaging conditions
20	highway

3.2.1 KITTI

The original KITTI Vision Benchmark Suite[18] is a database consisting of images captured by a windshield camera in a vehicle driving through Karlsruhe, Germany, including rural areas and on highways. The car is equipped with two high-resolution and grayscale video cameras. The dataset is designed for multiple computer vision tasks and includes annotations. Weather conditions during the recordings ranged from clear to partially cloudy, with variations in lighting conditions, such as sunny days, overcast skies, and sequences recorded during twilight or dawn.

A high quality, one-to-one simulation of the KITTI dataset, known as Virtual KITTI 2, is readily available. For this simulated dataset, five real-world KITTI videos were selected from the original dataset. These five scenes are described in Table 1.

An example frame of each scene is provided in Fig. 2. We expect the object detector to perform differently across these scenes, as reported in the original paper by Gaidon et al. (2016),[30] making it useful for analysing the correspondence between real and simulation (*SQ1*, Fig. 1).

Figure 2: Example frames from the real KITTI dataset (left) and the virtual clone frame of the Virtual KITTI 2 dataset (right) for the five different scenes.[18,19] From top to bottom: 01, 02, 06, 18, 20. These scenes come with respectively 447, 223, 270, 339 and 837 frames.

To create the virtual KITTI dataset, Gaidon et al. outlined five steps: (1) acquisition of real-world data for calibration purposes, (2) creation of a virtual copy of the real-world data using computer graphics engine Unity version 2018.4 LTS,[31] (3) automatic generation of different lighting and weather conditions, (4) automatic generation of detailed ground-truth annotations, (5) verification of the validity of the simulated data using multiple metrics. The Virtual KITTI 2 dataset[32] was chosen for our experiments due to its enhanced photorealism,

resulting from upgrades to the gaming engine. Although various environmental conditions were simulated in step (3), we did not include them in our analysis because corresponding real-world data is unavailable. Our analysis was conducted using only the real and the 'clone' (simulated) data. In step (4), annotations were automatically generated for all vehicles in view. To increase the correspondence between reality and simulation, we filtered the bounding boxes based on the smallest sizes present in the annotations of the real dataset, by removing the bounding boxes with either $w < 15$ or $h < 15$ pixels. The annotated classes for both real and simulation that we take into account are car, truck, and bus.

3.2.2 INDEV

The INDEV dataset, recorded by TNO, consists of videos recorded at various intersections in the Netherlands. For this study, we selected four 15-minute videos recorded at different times from the same camera position at an intersection in Eindhoven. These videos were chosen due to their varied weather (dry, raining) and lighting conditions (daylight at 14:00 and 21:00, night at 23:15 and 22:30). An example frame from each set is shown on the left side in Fig. 3.

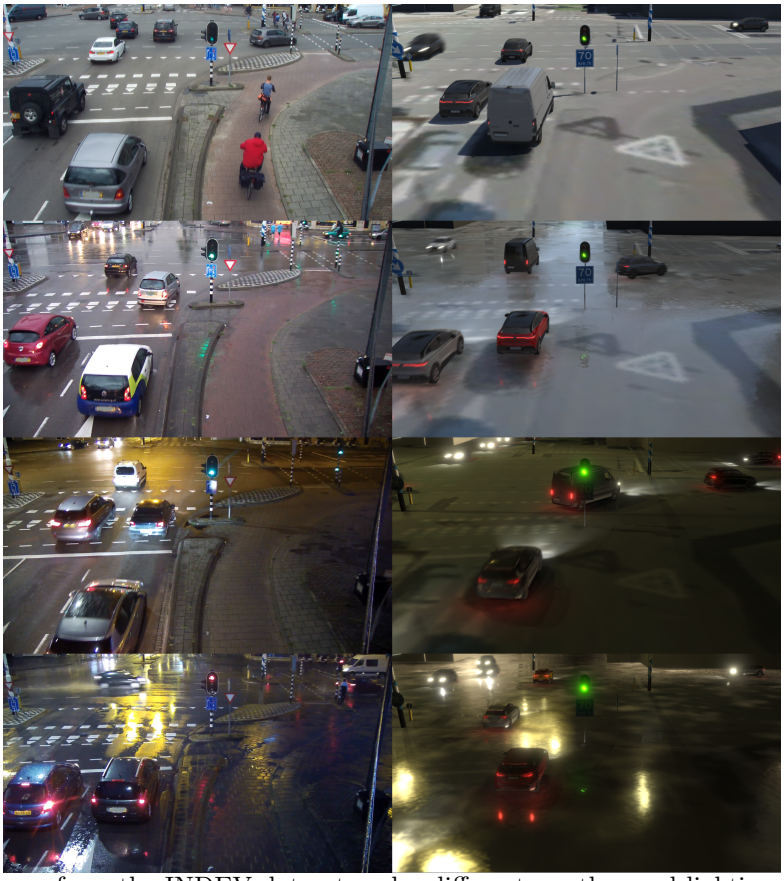


Figure 3: Example frames from the INDEV dataset under different weather and lighting conditions. Left: Real dataset frames. Right: Simulated dataset frames. For each real video we have annotated 60 frames; each simulated dataset contains 1000 frames.

The original videos were recorded at 60 frames per second (fps). For this study, 60 frames of each of the four videos were selected, resulting in a total dataset of 240 frames. The relevant automotive classes (car, bus, truck) were annotated using the Computer Vision Annotation Tool (CVAT),[33] an open-source interactive annotation tool for videos and images.

Based on the four real-world frame sets, we procedurally generated simulated data with matching environmental conditions (*SQ1, SQ2*). Because the object detector should be able to recognize vehicles anywhere in

the scene, vehicle models were scattered procedurally across the road lanes instead of manually replicating the original dataset exactly. This approach makes it much more feasible to rapidly generate large amounts of data.

The intersection where the original videos were recorded was reconstructed in the Unity game engine version 2022.3.7,[31] using aerial imagery from PDOK, a platform with geo-datasets of Dutch governments,[34] and building models from 3DBAG, an open 3D building dataset covering the Netherlands.[35] For each of the four original datasets, the environmental conditions were recreated by altering the global lighting settings and adding post-processing effects. The night scenes had no direct sunlight, but were illuminated by the street lights and the vehicle headlights instead. A bloom effect was added to simulate overexposure at night. For the rainy scenes, the specularity and albedo of the ground were adjusted to generate reflections that are similar in intensity and clarity to those that can be observed in the original videos.

In addition to these environmental conditions, motion blur, occurring when vehicles move at high speeds, can be observed in the original data. To replicate this, we added a motion blur post-processing effect to the simulation. We assigned a random velocity to each vehicle, resulting in pixels getting blurred across a corresponding distance along the direction of travel.

For all four simulated scenes, 1000 annotated frames were generated by randomly placing vehicle models along splines following the road lanes that are within view of the camera. These splines slightly extended beyond the view of the camera, resulting in vehicle models that might only be partially visible. Vehicles with insufficient space in front or behind them were discarded. The number of vehicles ranged from 1 to 20 per frame, consistent with the real data.

During the annotation process of the real frames, we observed a very low number of buses. To enhance the comparability between the real and simulated datasets, we decided to exclude buses from the simulation and placed only car and truck models in the simulation with a specified car to truck ratio of $8:1$. Including color variations, a total of 27 different vehicle types were used. Example frames from each set are shown on the right side of Fig. 3.

The annotated real frames are used for performance estimation and comparison with simulation (*SQ1*). Additionally, to analyze patterns in the data (*SQ3*), for which ground truth annotations are not required, we collected 1000 snapshots from each video, taken within 60 minutes surrounding the timestamps mentioned above. This ensures an equal number of real and simulated frames for pattern analysis.

3.3 Performance analysis

To evaluate the performance (*SQ1*) of the object detectors on both real and simulated datasets, we focused on comparing the AP values. The AP is a popular metric for object detection tasks that combines classification accuracy with localisation accuracy. We define AP as follows:

$$\text{AP@0.25} = \frac{1}{n}\sum_{k=1}^{k=n} P_{k,\text{IoU}=0.25} \tag{1}$$

where precision P is calculated per class k and per Intersection over Union (IoU) threshold by dividing the true positive detections by the total number of detections. We consider three relevant vehicle classes, thus $n = 3$. Localisation accuracy is determined by how well the predicted bounding boxes match the ground truth, measured by the IoU.[36] For this study, we use a rather low IoU threshold of 0.25 to prioritize the detection and correct classification of vehicles over the precision of bounding box localisation. For simplicity, we will refer to AP@0.25 as AP in the remainder of this work.

Additionally, we report the numbers of false positives (FP) and false negatives (FN), to gain more insight into the errors made by the models.

The AP value is computed per frame. To compare the performance on real and simulated and datasets, we perform statistical tests. For both datasets, the mean AP values are compared under the following hypotheses, referring to AP as X for simplicity:

$$H_0 : \mu(X_{real}) = \mu(X_{sim}) \tag{2}$$
$$H_1 : \mu(X_{real}) \neq \mu(X_{sim}) \tag{3}$$

We use the unpaired t-test statistic:

$$t_{unpair} = \frac{\bar{X}_{real} - \bar{X}_{sim}}{\sqrt{\frac{S^2_{real}}{n_{real}} + \frac{S^2_{sim}}{n_{sim}}}} \tag{4}$$

where $\bar{X}_{real}, \bar{X}_{sim}$ are the sample averages and S^2_{real}, S^2_{sim} the sample variances of the unpaired variables.[37] Given that the numbers of samples n_{real} and n_{sim} are both relatively large, the test statistic approaches a normal distribution.

As the real and simulated KITTI data are available in pairs, we also compute the paired comparison by subtracting the values X_{real} and X_{sim} for each corresponding frame. This leads to the hypotheses:

$$H_0 : \mu(X_{real} - X_{sim}) = 0 \tag{5}$$
$$H_1 : \mu(X_{real} - X_{sim}) \neq 0 \tag{6}$$

We then calculate the paired t-test statistic:

$$t_{pair} = \frac{\overline{X_{real} - X_{sim}}}{S/\sqrt{n}} \tag{7}$$

with S being the sample standard deviation of the difference. Again, this statistic approximates a normal distribution for a large number of samples n. For the procedural generation of the INDEV dataset, we only use the unpaired t-test since the real and simulated data do not form pairs.

In addition to comparing mean values using the tests, we compare the distributions. For this purpose, we employ the two-sample Kolmogorov-Smirnov ($K-S$) test[38] which compares the cumulative distribution functions (CDF) of both real and simulated AP values. Similar scenes could namely be expected to produce similar distributions of the AP values. The hypotheses are:

$$H_0 : F_{real}(X) = F_{sim}(X) \tag{8}$$
$$H_1 : F_{real}(X) \neq F_{sim}(X) \tag{9}$$

where $F_{real}(X), F_{sim}(X)$ are the CDFs of X_{real} and X_{sim}, respectively. The $K-S$ test uses the measure:

$$KS = \sup_x |F_{real}(x) - F_{sim}(x)| \tag{10}$$

to test if the differences between the two distributions are significantly different. For all tests, we use a significance level of $\alpha = 0.025$ to account for the two-sided nature of the tests, corresponding to a nominal significance level of $\alpha = 0.05$.

3.4 Pattern analysis

The previous experiments aim to determine how the performance measured on simulated data compares to the performance measured on real-world data, addressing *SQ1* (Fig. 1). In this way, we can assess how accurately the simulation reflects real-world performance. However, the primary goal of using simulation for testing DL object detectors is to reduce the dependency on large annotated real-world datasets. Thus, a direct comparison between real and simulated datasets may not always be feasible to test whether the simulation accurately reflects reality, leading us to *SQ3: "How can we evaluate the accuracy of a simulated environment in representing reality, without relying on large, annotated real-world datasets?"*.

To address this, we investigate whether other patterns correspond between simulation and real-world datasets. Often, object detection performance varies with the distance of the vehicles to the camera; for instance, vehicles

might be harder to detect when they are either very close or very far away from the camera. Both situations can occur at the edge of the images and often result in truncated objects.

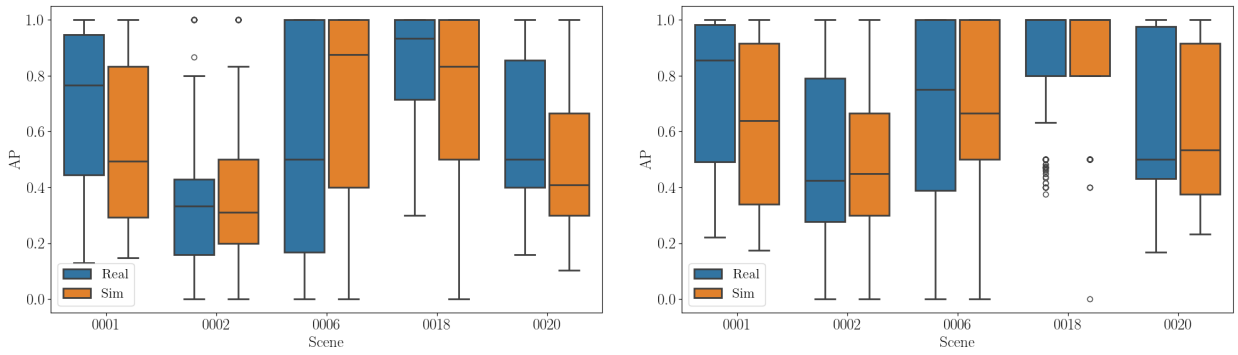
To determine whether this pattern is present for our use-case and, if so, whether it corresponds between real and simulated data, we analyze the relationship between the confidence of the prediction and the vertical location of the bounding box. Both datasets were recorded with the camera at a constant position and tilt angle relative to the ground, allowing us to use the bottom coordinate of the predicted bounding box as an indicator of the object's distance from the camera. Since the y-axis starts at the top of the image, a higher y-coordinate indicates that the object is closer to the camera, while a lower y-coordinate corresponds to a greater distance from the camera. To gain a better understanding of the relationship, we perform a regression analysis using a second-order order polynomial, fitted with an ordinary least squared approach which examines the impact of the predictor (*distance*) on the dependent variable (*confidence*). Next, we conduct an Analysis of Covariance (ANCOVA) to determine if the relationship between distance and confidence differs between real and simulated datasets. This analysis controls for covariates and highlights whether confidence levels vary consistently across both data types. Finally, to further compare the distributions, without assuming the underlying data to be normally distributed, we compute the Empirical Cumulative Distribution Function (ECDF) for both datasets, providing a cumulative view of how object distances were distributed. The difference between the distributions are analyzed via the $K - S$ test.

4. RESULTS

4.1 KITTI

4.1.1 Performance Analysis

The boxplots in Fig. 4 show that the AP on the real and simulated datasets exhibit a similar pattern, in average value as well as in error margins and outliers. The highest performance is consistently achieved on *scene 18*, with *scene 2* and *20* being more challenging.



(a) YOLOv8n (b) YOLOv8l

Figure 4: Box plots showing the AP for the different scenes of the KITTI dataset, for both the real and simulated data.

Results for the statistical tests on the mean AP are provided in Table 2. There is a large variation in the correspondence between real and simulation over the different KITTI scenes. Based on the unpaired t-test, *scene 01* and *scene 20* are found to be statistically significant different for both YOLO models. *Scene 06* is only found to be different for the smaller YOLOv8n model. The paired t-test however indicates that only *scene 02* for YOLOv8n and *scene 06* for YOLOv8l are not significantly different between the real-world and simulated data.

Similar to the results of the means test, the $K - S$ distribution test shows that mainly *scene 01* and *20* are largely different. *Scene 02* and *18* result in a similar AP distribution, not showing a significant difference for both models. The p-values for *scenes 06* and *18* differ a lot between the two models, with *scene 06* even being significantly different for YOLOv8n and not YOLOv8l. This indicates dependence on the specific model used.

Table 2: Average AP values over the frames for the real and simulated KITTI datasets.

	YOLOv8n					YOLOv8l				
Scene	**real**	**sim**	**unpaired p-value**	**paired p-value**	**KS-test p-value**	**real**	**sim**	**unpaired p-value**	**paired p-value**	**KS-test p-value**
01	0.70	0.55	<0.001*	<0.001*	<0.001*	0.75	0.63	<0.001*	<0.001*	<0.001*
02	0.36	0.35	0.373	0.137	0.743	0.50	0.50	0.484	0.014*	0.525
06	0.57	0.64	0.024*	0.008*	0.007*	0.63	0.66	0.224	0.031	0.330
18	0.82	0.79	0.086	0.006*	0.330	0.86	0.88	0.145	<0.001*	0.054
20	0.61	0.50	<0.001*	<0.001*	<0.001*	0.65	0.60	0.004*	<0.001*	<0.001*

*: statistically significant difference.

Table 3: Statistics of the real and simulated KITTI data and the predictions as made by the YOLOv8l model. GT = Ground Truth.

	# GT objects		**# predicted objects**		**# false positives**		**# false negatives**	
Scene	**real**	**sim**	**real**	**sim**	**real**	**sim**	**real**	**sim**
01	2898	5327	5581	4559	3028	913	345	1681
02	1226	1417	1969	1468	1099	524	403	473
06	762	747	1169	811	571	211	163	147
18	1413	1413	2503	1591	1022	267	114	89
20	6404	8535	9603	7653	4551	1915	1350	2774

In the real KITTI data, many more false positives are predicted by both YOLO models than in the simulated data. These statistics are shown for YOLOv8l in Table 3. After visual inspection of the results on the real data, we found that this can partly be explained by predictions located at occluded cars, that are not correctly annotated as ground truth. In the simulated data however, the annotations are automatically generated. We attempted to prevent this difference by filtering out annotations in the simulated data with $w < 15$ or $h < 15$ pixels. However, this is an arbitrary threshold, and differences in accuracy and level of detail between the manually and automatically generated annotations are still present. The authors of the virtual KITTI[30] mention this gap as well, stating that mainly "corner cases" - with heavy truncation or occlusion - are inconsistently or inaccurately labeled.

The total # ground truth objects and # predicted objects for the YOLOv8l models are also listed in Table 3. An example annotated frame is shown in Figure 5, where the bounding boxes predicted by YOLOv8l are shown on the left. For the real image the bounding boxes marked as false positives are plotted on the right. As can be seen, these are incorrectly classified as false positives as they are plotted over non-annotated vehicles. For the simulated image, the bounding boxes marked as false negatives are plotted mostly over occluded vehicles which are, even for the human eye, hardly visible.

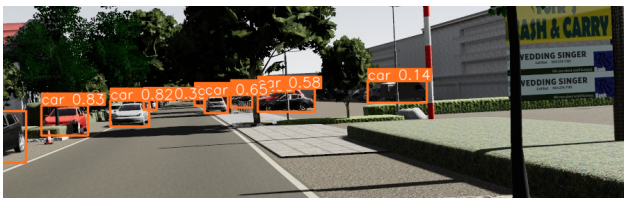
Thus, the difference in annotations between real and simulated explains some of the false positives and negatives. Globally, we see more predictions in the real data than in the simulated data, which could be explained by noise present in the real data. However, while for *scene 18* the total amount of GT labels are the same, the # predictions and # false positives for real are much higher. Upon inspection, we found that many objects that are present in the real data were in fact not cloned into the virtual image. An example of this can be found in Fig. 6.

4.1.2 Pattern Analysis

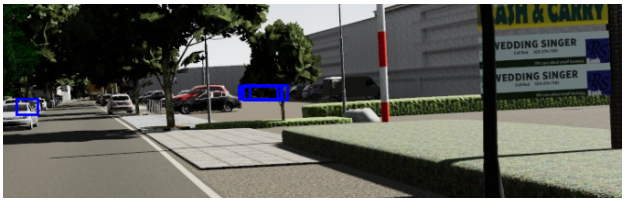
To analyze the relationship between the position of the bounding box and the corresponding confidence, scatter plots of the real and simulated images of *scene 01* are shown in Figure 7. A second-order order polynomial was fitted with a R^2 value of 0.46 (see Table 6 in the Appendix), suggesting a moderate fit. The positive coefficient for distance (7.41) suggests that confidence increases with the y-coordinate, meaning confidence is higher at closer distances. This relationship is moderated by a significant negative quadratic term (-4.19), indicating a

(a) Real - predictions

(b) Real - FP



(c) Simulated - predictions



(d) Simulated - FN

Figure 5: Zoomed in on frame 46 of *scene 01* from (virtual) KITTI showing the predictions of the YOLOv8l model on the left and the incorrectly classified 'false positives' (FP) for the real image and 'false negatives' (FN) for the simulated image on the right.

(a) Real

(b) Simulated

Figure 6: Frame 68 of (virtual) KITTI *scene 18* showing discrepancies between the images. In the simulated image three cars are present, where in the real image at least five cars are visible.

nonlinear pattern where confidence initially increases as distance decreases but then declines after reaching a certain threshold. This supports our initial hypothesis that detected objects near the edge of the image have lower confidence scores.

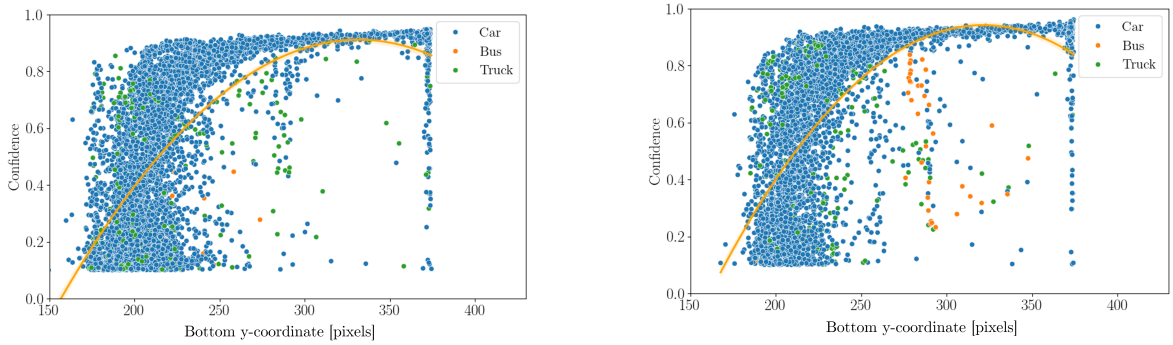
The ANCOVA model applied on the YOLOv8l results supports the findings in Section 4.1.1 for *scene 02*, as the coefficients for the second-order order polynomial of the simulated dataset are not statistically different from the real dataset. The observations of this scene are shown in Figure 8. Regarding YOLOv8l, there is no statistical evidence for similarity in the other scenes. The ANCOVA results for the YOLOv8n model show that the polynomials fitted for the real and simulated data of *scene 01* and *scene 06* are similar. However, the second-order order polynomials do not fit well to the data, as only a low R-squared value is obtained.

The ECDFs of the observations in relation to distance are depicted in Fig. 9. The two-sided $K - S$ test for this ECDF resulted in a p-value of < 0.001 suggesting to reject the null-hypothesis $F_{\text{sim}} = F_{\text{real}}$. It should be noted however that, where we would expect the camera viewpoint to be identical in the real and simulated frames of KITTI, the camera position in simulation appears to be slightly tilted upwards compared to the camera viewpoint in the real frames. This could partly explain the shifted distribution in the ECDF plots.

4.2 INDEV

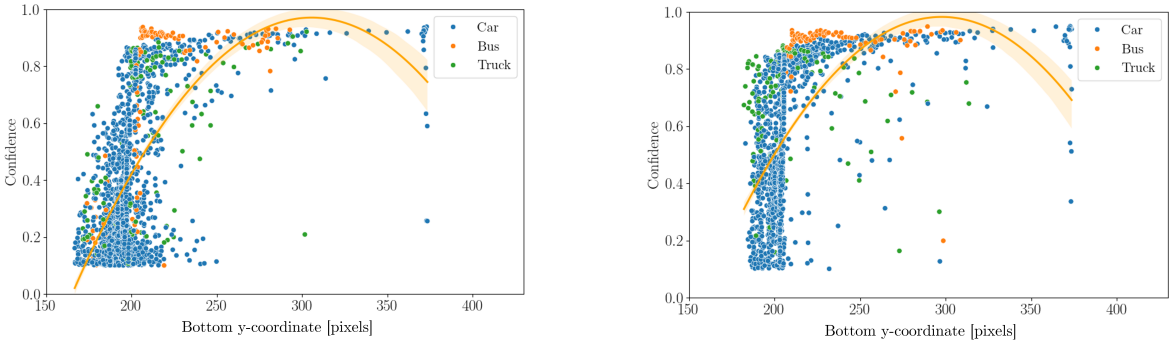
4.2.1 Performance Analysis

For the INDEV dataset, a smaller difference in AP is observed for the different weather and lighting conditions, when compared to scene differences in the KITTI dataset (Fig. 4 vs. Fig. 10). The rain at night time is the only condition that consistently results in a lower AP for the real data. This decrease is replicated in the simulation. However, only changing from day to night already results in this performance decrease, indicating that adding



(a) Real (b) Simulated

Figure 7: Scatter plot of the relationship between confidence and bottom y-coordinates of the bounding boxes predicted by YOLOv8l, on the KITTI dataset in all frames of *scene 01*.



(a) Real (b) Simulated

Figure 8: Scatter plot of the relationship between confidence and bottom y-coordinates of the bounding boxes predicted by YOLOv8l, on the KITTI dataset in all frames of *scene 02*.

rain does not affect the detection performance. This is confirmed by similar results for daylight with and without rain.

The average AP values are reported in Table 4. While in the boxplots in Figure 10, the results on the simulation and the real dataset appear to be comparable, the statistical tests show that both the averages and the distributions of the APs are statistically different (for all the unpaired t-tests and the $K - S$ test $p < 0.001$). As mentioned, the AP for INDEV only considers car and truck classes. Upon further inspection of the performance split per class, we see that especially in the simulation, the truck class is very often not detected or wrongly predicted as car. Focusing the performance only on the car class improves comparability of the real and simulated data, but still mainly yields significantly different results (see Fig. 15 in the Appendix).

In Table 5, the # false positives and # false negatives are listed as the fraction of the total # predictions, considering the difference in # of frames (60 vs. 1000 for real vs. simulation respectively). In all conditions and both models, except for YOLOv8l in *night-rain*, the # false negatives is higher in the simulation. Distant cars are often missed in the night simulations, for which an example is depicted in Fig. 11. The # false positives on the other hand, is higher for real, except for the YOLOv8l model during both night-time conditions. Again, this suggests an increased amount of noise in the surroundings in the real data - even more during daytime. The examples in Fig. 11 demonstrate this difference in noise level. Finally, we see a correspondence in the kind of mistakes that occur in both the real-world and simulated data. For instance, the airplane class is regularly predicted on vehicles in the top corners and, while this is more prevalent in the simulation, distant cars are often not detected (Fig. 11).

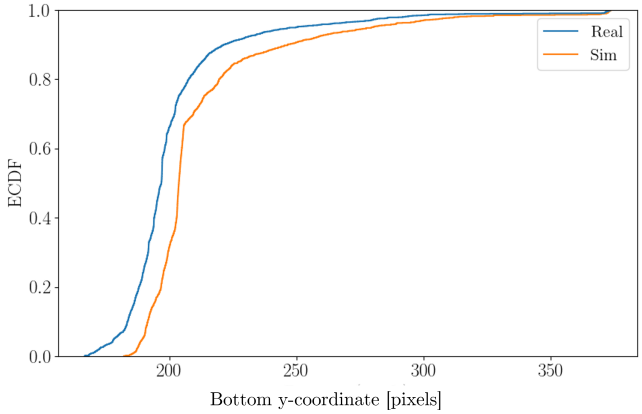
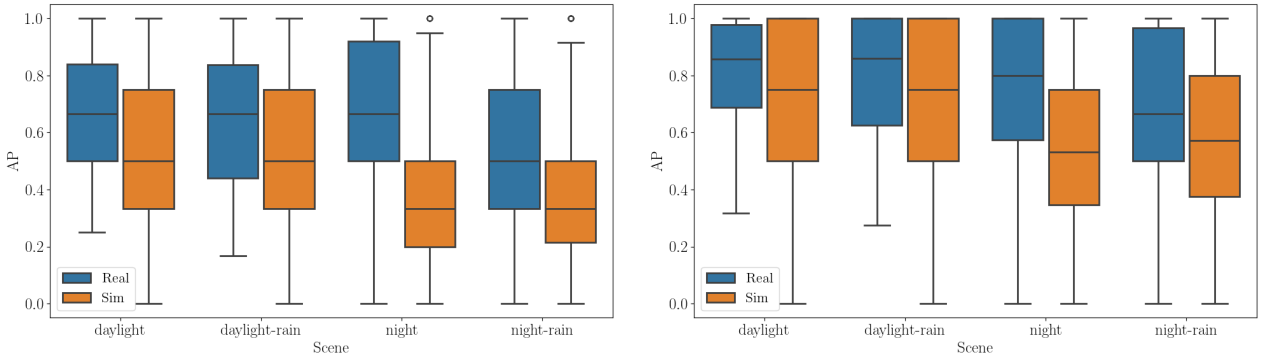


Figure 9: The Empirical Cumulative Distribution Function (ECDF) for the bottom y-coordinates of the bounding boxes predicted by the YOLOv8l model on both the real and simulated data of KITTI *scene 02*. While the distributions' shapes appear to be similar, the $K-S$ test indicates a significant difference.



(a) YOLOv8n (b) YOLOv8l

Figure 10: Box plots showing the AP for the different conditions of the INDEV dataset, for both the real and simulated data.

4.2.2 Pattern analysis

Scatter plots were made to visualize the relationship between the location of the object in the image and the corresponding confidence. The simulated images were procedurally generated and do not follow the approach of a one-to-one copy like Virtual KITTI. Because of this, there are some differences between the datasets: 1) the image size is different for images of the both datasets, and 2) the locations of the vehicles in the frame do not necessarily follow the same distribution in both datasets, as they were placed procedurally in the virtual images.

To account for these changes, the distance was first normalized, and subsequently scaled with the probability of a vehicle located at this distance. The results are visualized in Fig. 12, and the data seems less suited to fit a polynomial curve. This is especially apparent for the simulated dataset, where the R^2 value is only 0.041 (see Table 7 in the Appendix for all results). The ECDF is plotted for the predictions in the real and simulated dataset. Even though the curves appear to be similar in Fig. 13, $K-S$ testing indicates that the two distributions are statistically different.

Table 4: The average APs for the INDEV dataset. All results were statistically different for both the unpaired t-test and the $K-S$-test (all $p < 0.001$) and are thus not reported in this table.

	YOLOv8n		**YOLOv8l**	
Scene	**real**	**sim**	**real**	**sim**
daylight	0.67	0.55	0.79	0.73
daylight-rain	0.65	0.54	0.79	0.72
night	0.64	0.36	0.76	0.56
night-rain	0.53	0.38	0.61	0.59

Table 5: Statistics of the real and simulated INDEV data and the predictions as made by the YOLOv8l model. GT = Ground Truth.

	# GT objects		**# predicted objects**		$\frac{\#falsepositives}{\#predictions}$[%]		$\frac{\#falsenegatives}{\#GTobjects}$[%]	
Scene	**real**	**sim**	**real**	**sim**	**real**	**sim**	**real**	**sim**
daylight	461	6083	429	5384	12	10	18	20
daylight-rain	332	6086	363	5456	23	11	16	20
night	201	6289	172	4810	10	18	22	37
night-rain	213	6188	157	4931	12	18	35	33

5. DISCUSSION

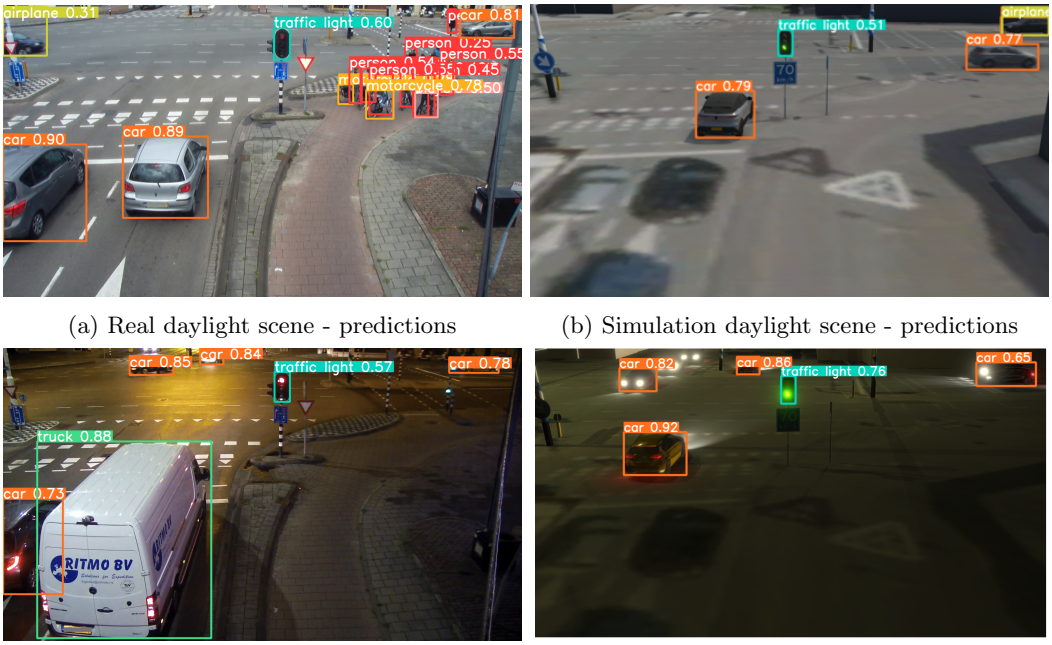
In this work, we researched how simulated data can be used effectively in the testing phase of a DL-based object detector. Our study addressed three key aspects (Fig. 1) through a series of experiments using two automotive datasets — one publicly available (KITTI) and one internally developed (INDEV) — with both real and simulated versions of the scenes.

5.1 Comparing performance across different conditions on simulated vs. real data (SQ1)

First, we evaluated how the object detector's performance on simulated data compares to its performance on real-world data under varying conditions (*SQ1*). For the real INDEV dataset, we observed less performance degradation under the different conditions than expected. Despite using a smaller YOLO model in an attempt to capture more pronounced performance changes, only slight performance decreases were observed, particularly for the nightly rain setting (AP dropping 0.67 to 0.53 from *daylight* to *night-rain*). Interestingly, the difference over the conditions was similar or even smaller in comparison with the larger model, suggesting that model size may not always negatively correlate with greater sensitivity to condition changes.

In contrast, the real KITTI dataset showed more significant performance differences across scenes (Table 2, e.g. AP ranging from 0.36 to 0.82 for *scene 01* and *18* respectively for YOLOv8n). This dataset demonstrated the expected increase in performance difference when switching to a smaller model, with more pronounced drops in performance for specific scenes. While some changes in performance were observed, future work should focus on settings where changes are more notable. This would allow for stronger statements about the correspondence of these changes on real and simulated data. For now, especially for INDEV, the differences are slight, with the exception of *night-rain* conditions.

The use of rain settings in the INDEV simulation did not significantly impact the AP (for YOLOv8n e.g. 0.55 to 0.54 for *daylight* to *daylight-rain*, 0.36 to 0.38 for *night* to *night-rain*, and see Fig. 10 and Fig. 15). While rain combined with night-time reduces performance more substantially (-0.11 AP for YOLOv8n), switching to night-time alone impacts performance much more in simulation than in reality. This difference shows that our simulated data, while looking realistic to the human eye, especially compared to for instance the Virtual KITTI 2 rain simulation,[30] still has limitations in accurately replicating real-world conditions. Real datasets under adverse conditions like rain or night-time may introduce noise and artifacts not perfectly captured in the simulation, leading to different performance metrics. Similar challenges are found in the SHIFT dataset,[20] where shifts in weather types are attempted but not perfectly aligned with real-world data.



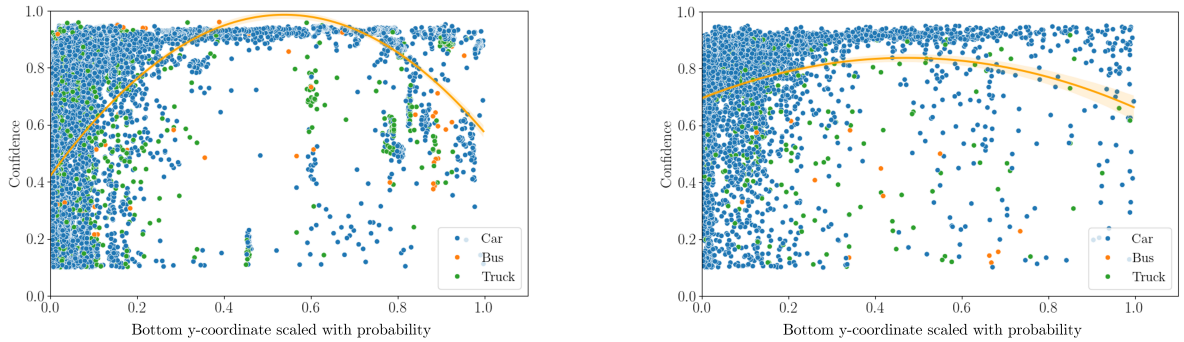
(a) Real daylight scene - predictions

(b) Simulation daylight scene - predictions

(c) Real night scene - predictions

(d) Simulation night scene - predictions

Figure 11: The top row shows that false positive predictions of the airplane class occur in similar situations on both the real and simulated data. In addition, it is clear that there is much more noise present in the surroundings of the real data, while the simulation is much simpler and less crowded. This is less present in the night scenes (bottom row), considering the absence of most bicycles and persons.



(a) Real

(b) Simulated

Figure 12: Scatter plots of the relationship between confidence and normalized, scaled bottom y-coordinate of bounding boxes predicted by YOLOv8l, on the INDEV dataset on all frames of the *daylight* scene.

The difference in mean AP was evaluated using commonly applied t-tests, which require the assumption of normality in the data. When both groups have at least 30 data points, the Central Limit Theorem generally justifies assuming a normal distribution.[39] For smaller sample sizes, the Shapiro-Wilk test can be used to assess normality. In our case, with the smallest sample size being 60, it was reasonable to assume a normal distribution. A non-parametric alternative, like the $K - S$ test, could be used to avoid reliance on the data's underlying distribution. This approach would be particularly useful when working with diverse datasets, which is often the case during the testing of a deep learning model.

Furthermore, when comparing the mean AP versus the distribution of the AP for KITTI, we observe differences between the t-test and the $K - S$ test results. This discrepancy suggests that conclusions may vary depending on whether we seek comparable mean performance or a similar distribution. Therefore, the choice of

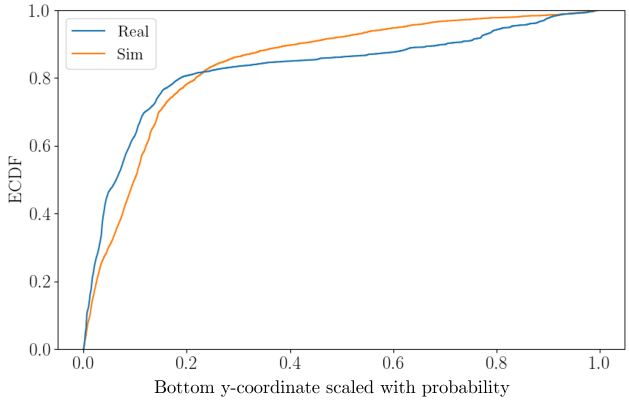


Figure 13: The Empirical Cumulative Distribution Function (ECDF) for the normalized bottom y-coordinate of the bounding boxes predicted by the YOLOv8l model on both the real and simulated data of the *daylight* scene. While the distributions appear to be similar, the $K-S$-test indicates a significant difference.

test is not just statistical, but also practical. It depends on how the outputs of the DL model are intended to be applied in real-world scenarios. For instance, in applications where consistent performance across diverse conditions is important, a closer examination of the distribution might be more informative than a simple comparison of means. A related consideration is whether simulations need to be photo-realistic or whether they simply need to yield comparable performance. The same principle applies when training AI-models with simulated data; the simulation does not necessarily have to be (photo-)realistic, as long as the required information can be extracted from the images and the model performs well on real-world test sets.[7, 40]

5.2 Cost-effectiveness of generating a simulated dataset (SQ2)

By simulating the INDEV dataset, we explored the cost-effectiveness of generating a simulated dataset suitable for testing an object detector (*SQ2*). The procedural generation of this dataset, as described in Section 3.2.2, proved to require relatively little effort to achieve a performance on the car class that is comparable to that obtained with real-world data. When including the truck class, the difference in performance increases (comparing Fig. 10 vs. Fig. 15 in the Appendix). This issue is partly attributed to the lower ratio of trucks to cars in the real dataset compared to the simulation, which used a probability ratio of 8 cars to 1 truck.

So, while this difference in class balance causes a discrepancy in the prediction performance, it also highlights a key benefit of simulation: the ability to control and modify class balance. In real data, achieving a balanced dataset is challenging due to certain objects naturally occuring more frequently than others. This imbalance can lead to biased model performance, favoring majority classes while underperforming on minority classes. For example, in a military vehicle detection context, common vehicles like jeeps or trucks might vastly outnumber rare vehicles like tanks. Simulation allows for artificial adjustment of class distributions, addressing this issue.

As described in Section 5.1, challenging conditions such as rain or night-time are difficult to simulate accurately. Initial procedural generation of simulations may require relatively little effort, achieving a high degree of realism. However, especially the more complex scenarios might require additional fine-tuning. This fine-tuning process, aimed at a better replication of certain environmental factors like for instance lighting, reflections, and noise inherent in real-world conditions, can significantly increase the overall costs of generating a simulated dataset. This trade-off between the level of effort invested in generating the simulation and the resulting cost-effectiveness must be considered. Balancing these factors is important, as the additional investment in fine-tuning may not always result in a proportional improvement in simulation accuracy.

Overall, the performance estimated on real and simulated KITTI data was comparable. However, *scenes 01* and *20* consistently showed significant differences, likely due to discrepancies in annotations. Real KITTI data often contains inaccuracies, such as missed occluded objects, while simulated datasets like Virtual KITTI provide

precise automatic annotations. This discrepancy is present in all scenes, but probably in a higher degree for these scenes, supported by their increased difference in # GT objects present (Table 3). In our INDEV dataset, manual annotation was limited due to the substantial effort required, resulting in a large difference in sample size and weakening the power of statistical analysis on the AP (Section 4.2.1). Annotation efforts and accuracy are an important factor determining object detection importance.[41,42] Especially the careful annotation of small objects is a challenging topic,[43,44] in which the automatic generation of annotation in simulations could help in finding a solution.

5.3 Evaluating accuracy of simulation in representing reality (SQ3)

Finally, we explored methods to evaluate the accuracy of simulated environments in representing reality, without relying on large, annotated real-world datasets (*SQ3*). While AP provides an estimate of the DL detector's performance, it does not directly offer insights into how accurately the simulation represents reality. The challenge lies in validating whether the AP obtained from simulated data can reliably serve as a proxy for the AP from real data. This validation again requires access to annotated real datasets, which are often scarce.

Our findings highlight the difficulty of guaranteeing that AP derived from simulated data will consistently reflect the performance on real-world data, complicating the use of simulations as a substitute for real-world data. To address this, we explored other informative characteristics, such as the impact of object distance on prediction confidence. We have demonstrated that several statistical tests on patterns in the data reveal that simulated and real datasets for both KITTI and INDEV are statistically different. These results highlight the difficulty in developing simulations that closely match real-world data with high fidelity.

Techniques like ANCOVA and computation of ECDFs, in combination with visual analysis of the results, can provide valuable insights into the disparities between real and simulated datasets, without the need for annotations. However, more refined pattern analysis techniques to evaluate the reliability of simulated datasets should be researched. One approach could be to perform localized regression analyses on specific regions of the image, such as areas with varying densities of predictions (e.g., high-confidence predictions in the middle of the image versus low-confidence predictions at the edges) or use a non-parametric curve to fit the data and recognize patterns. This can help in identifying whether the simulated data captures the nuances of real-world conditions across different scenarios and avoids assumptions of the underlying data distribution. Testing other hypotheses, such as the model's confidence in predicting occluded objects, could also be beneficial. Since these statistics can be readily obtained from simulations, they could provide a more detailed understanding of the model's behaviour under various conditions.

We propose that when creating simulations to test DL models, one should ensure that statistical tests are in place that align with the specific requirements of the application. These tests should guide the fine-tuning of simulations to ensure that they meet the necessary standards for the intended real-world scenarios.

5.4 Conclusions and future work

While this study demonstrates promising results for using simulated data in testing DL-based object detectors, several areas require further research to enhance the validation and application of simulation. While the quick procedural simulation of the INDEV dataset has shown potential, a method has yet to be found to validate the use of simulated data without relying on real-world datasets.

For this study, we have focused more on testing the simulation than on the development of a simulation as accurate as possible. While this was out of our scope for now, future research should include improvements in the accuracy of simulated environments, taking into account sensor noise, environmental factors, and object dynamics.

While it is unlikely that simulated data will completely replace the need for real data in the near future, we do see more apparent possibilities for specific aspects of the testing phase. Two key areas are stress-testing and the identification of challenging conditions. Stress-testing can be used to analyze a model's behaviour under extreme conditions, such as an increased motion blur, reduced resolution, darker environments, more intense rain, or lower-quality 3D models. Understanding how and when these factors degrade model performance can provide valuable insights into the robustness of object detectors and help identify failure modes where the

model's performance drops significantly. Additionally, simulations could be used to quickly identify challenging conditions for models, such as foggy weather, as demonstrated in datasets like SHIFT and KITTI.[20,30] Once these conditions are identified, efforts can be focused on collecting real data in these specific scenarios, optimising the use of limited data collection resources. However, our work indicates that accurately simulating these challenging conditions is difficult, and it remains uncertain whether the simulations provide a reliable indication of real-world performance.

In conclusion, while simulated data holds promise for the testing phase of DL-based object detectors, significant challenges remain especially in finding a method to determine that a simulation represents reality with sufficient accuracy. Future work should focus on enhancing the representativeness of simulations, developing robust methods to test the simulation's reliability, and analysing in which specific part of the testing phase simulations can be the most effective. These efforts are important to advance the use of simulation as a reliable tool for model validation and performance analysis.

ACKNOWLEDGMENTS

This work was partly funded by the Netherlands Ministry of Defense under the V2113 MAVERIC (Military Autonomous VERification & validation in Complex environments) research program and partly by TNO's Appl.AI program.

REFERENCES

[1] Akyon, F. C., Eryuksel, O., Ozfuttu, K. A., and Altinuc, S. O., "Track Boosting and Synthetic Data Aided Drone Detection," *AVSS 2021 - 17th IEEE International Conference on Advanced Video and Signal-Based Surveillance* (2021).

[2] Reddy Nandyala, N. and Kumar Sanodiya, R., "Underwater Object Detection Using Synthetic Data," *2023 11th International Symposium on Electronic Systems Devices and Computing, ESDC 2023* (2023).

[3] Huang, J., Yin, J., Wang, S., and Kong, D., "Synthetic Data: Development Status and Prospects for Military Applications," *Mechanisms and Machine Science* **145**, 979–992 (2024).

[4] Liegl, C. J., Nickchen, T., Strunz, E., Horn, A., Coppenrath, A., Uysal, U., Ruß, M., and Luft, F., "Simulation: The Great Enabler?," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **13866 LNCS**, 312–325 (2023).

[5] Luotsinen, L. J., Kamrani, F., Lundmark, L., and Sabel, J., "Deep learning with limited data: A synthetic approach," (2021).

[6] van Breugel, B., Seedat, N., Imrie, F., and van der Schaar, M., "Can you rely on your model evaluation? improving model evaluation with synthetic test data," *Advances in Neural Information Processing Systems* **36** (2024).

[7] Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., and Birchfield, S., "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in [*IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*], **2018-June**, 1082–1090, IEEE Computer Society (apr 2018).

[8] Peng, X., Sun, B., Ali, K., and Saenko, K., "Learning Deep Object Detectors from 3D Models,"

[9] Kattakinda, P., Levine, A., and Feizi, S., "Invariant Learning via Diffusion Dreamed Distribution Shifts," (nov 2022).

[10] Voetman, R., Aghaei, M., and Dijkstra, K., "The Big Data Myth: Using Diffusion Models for Dataset Generation to Train Deep Detection Models," (jun 2023).

[11] Fang, H., Han, B., Zhang, S., Zhou, S., Hu, C., and Ye, W.-M., "Data Augmentation for Object Detection via Controllable Diffusion Models," in [*WACV*], (2024).

[12] Feng, C.-M., Yu, K., Liu, Y., Khan, S., and Zuo, W., "Diverse Data Augmentation with Diffusions for Effective Test-time Prompt Tuning," tech. rep.

[13] Eker, T. A., Heslinga, F. G., Ballan, L., den Hollander, R. J. M., and Schutte, K., "The effect of simulation variety on a deep learning-based military vehicle detector," (October), 26 (2023).

[14] Heslinga, F. G., Eker, T. A., Fokkinga, E. P., van Woerden, J. E., Ruis, F. A., den Hollander, R. J., and Schutte, K., "Combining simulated data, foundation models, and few real samples for training object detectors," in [*Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II*], **13035**, 44–55, SPIE (2024).

[15] Rosenzweig, J., Brito, E., Kobialka, H.-U., Akila, M., Schmidt, N. M., Schlicht, P., David Schneider, J., Hüger, F., Rottmann, M., Houben, S., and Wirtz, T., "Validation of simulation-based testing: Bypassing domain shift with label-to-image synthesis," in [*2021 IEEE Intelligent Vehicles Symposium Workshops*], 182–189 (2021).

[16] Pérez-García, F., Bond-Taylor, S., Sanchez, P. P., van Breugel, B., Castro, D. C., Sharma, H., Salvatelli, V., Wetscherek, M. T. A., Richardson, H., Lungren, M. P., Nori, A., Alvarez-Valle, J., Oktay, O., and Ilse, M., "RadEdit: stress-testing biomedical vision models via diffusion image editing," (2023).

[17] Gaidon, A., Wang, Q., Cabon, Y., and Vig, E., "VirtualWorlds as Proxy for Multi-object Tracking Analysis," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2016-December**, 4340–4349 (dec 2016).

[18] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R., "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)* (2013).
[19] Cabon, Y., Murray, N., and Humenberger, M., "Virtual KITTI 2," (jan 2020).
[20] Sun, T., Segu, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., and Yu, F., "Shift: a synthetic driving dataset for continuous multi-task domain adaptation," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 21371–21382 (2022).
[21] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T., "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in [*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*], 2636–2645 (2020).
[22] Zhou, H., Ma, J., Tan, C. C., Zhang, Y., and Ling, H., "Cross-Weather Image Alignment via Latent Generative Model with Intensity Consistency," *IEEE Transactions on Image Processing* **29**, 5216–5228 (2020).
[23] Rothmeier, T., Wachtel, D., Von Dem Bussche-Hunnefeld, T., and Huber, W., "I Had a Bad Day: Challenges of Object Detection in Bad Visibility Conditions," *IEEE Intelligent Vehicles Symposium, Proceedings* **2023-June** (2023).
[24] Rothmeier, T. and Huber, W., "Let it Snow: On the Synthesis of Adverse Weather Image Data," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* **2021-September**, 3300–3306 (sep 2021).
[25] Marathe, A., Ramanan, D., Walambe, R., and Kotecha, K., "WEDGE: A multi-weather autonomous driving dataset built from generative vision-language models,"
[26] Wang, J., Xu, M., Xue, H., Huo, Z., and Luo, F., "Joint image restoration for object detection in snowy weather," *IET Computer Vision* (2024).
[27] Rothmeier, T., Huber, W., and Knoll, A. C., "Time to Shine: Fine-Tuning Object Detection Models with Synthetic Adverse Weather Images,"
[28] Yang, Y., Zhang, H., Katabi, D., and Ghassemi, M., "Change is Hard: A Closer Look at Subpopulation Shift," *Proceedings of Machine Learning Research* **202**, 39584–39622 (2023).
[29] Jocher, G., Chaurasia, A., and Qiu, J., "YOLO-v8 by Ultralytics," tech. rep. (2023).
[30] Gaidon, A., Wang, Q., Cabon, Y., and Vig, E., "Virtual worlds as proxy for multi-object tracking analysis," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 4340–4349 (2016).
[31] Unity Technologies, "Unity," (2023). Game development platform.
[32] Cabon, Y., Murray, N., and Humenberger, M., "Virtual kitti 2," *arXiv preprint arXiv:2001.10773* (2020).
[33] Opencv, "Opencv/cvat: Annotate better with CVAT, the industry-leading data engine for machine learning. used and trusted by teams at any scale, for data of any scale."
[34] de Kaart (PDOK), P. D. O., "Over pdok." https://www.pdok.nl/over-pdok (2024). Accessed: 2024-08-06.
[35] Peters, R., Dukai, B., Vitalis, S., van Liempt, J., and Stoter, J., "Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands," (2022).
[36] Padilla, R., Netto, S. L., and Da Silva, E. A., "A survey on performance metrics for object-detection algorithms," in [*2020 international conference on systems, signals and image processing (IWSSIP)*], 237–242, IEEE (2020).
[37] Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P., and Meester, L. E., [*A Modern Introduction to Probability and Statistics: Understanding why and how*], Springer Science & Business Media (2006).
[38] Wikipedia contributors, "Kolmogorov–smirnov test — Wikipedia, the free encyclopedia," (2024). [Online; accessed 31-July-2024].
[39] Fischer, H., [*A history of the central limit theorem: from classical to modern probability theory*], vol. 4, Springer (2011).
[40] Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., and Birchfield, S., "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," in [*2019 International Conference on Robotics and Automation (ICRA)*], 7249–7255, IEEE (2019).
[41] Ma, J., Ushiku, Y., and Sagara, M., "The effect of improving annotation quality on object detection datasets: A preliminary study," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 4850–4859 (2022).

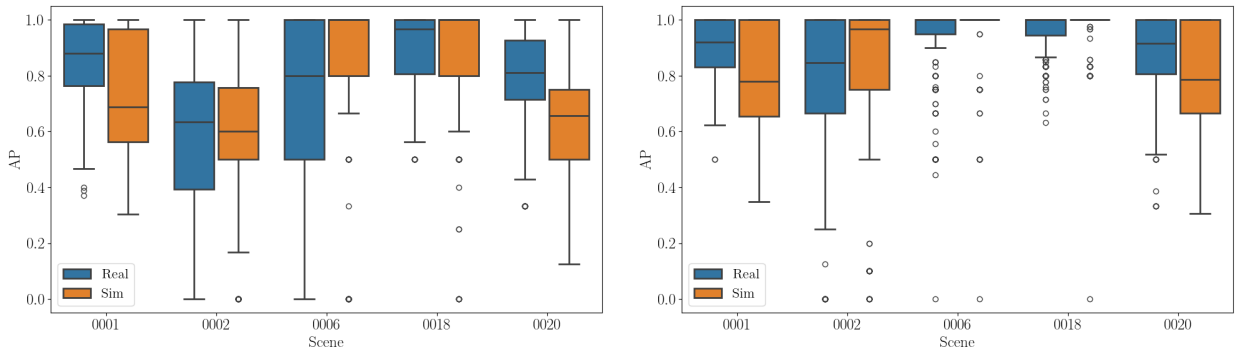
[42] Salari, A., Djavadifar, A., Liu, X., and Najjaran, H., "Object recognition datasets and challenges: A review," *Neurocomputing* **495**, 129–152 (2022).

[43] Tong, K., Wu, Y., and Zhou, F., "Recent advances in small object detection based on deep learning: A review," *Image and Vision Computing* **97**, 103910 (2020).

[44] Cheng, G., Yuan, X., Yao, X., Yan, K., Zeng, Q., Xie, X., and Han, J., "Towards large-scale small object detection: Survey and benchmarks," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

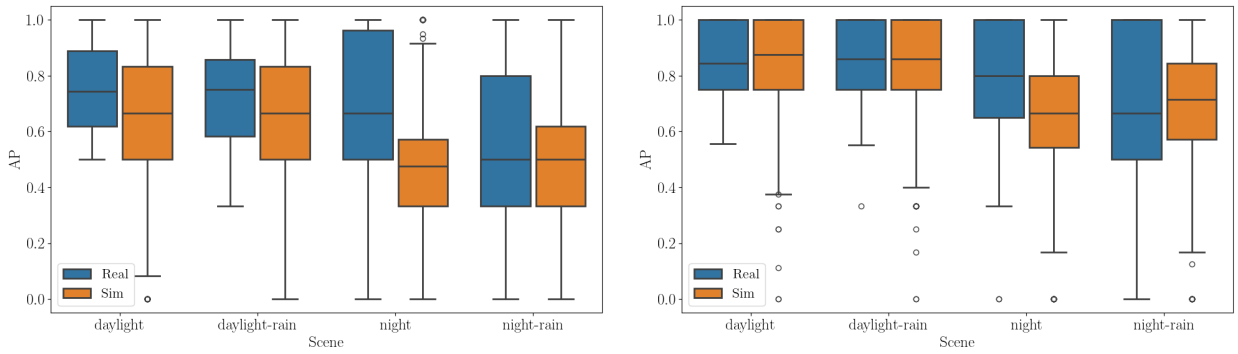
Appendix

This appendix lists additional results on the KITTI and INDEV datasets, including boxplots for the AP and statistical tests for the confidence-distance relationship.



(a) YOLOv8n (b) YOLOv8l

Figure 14: KITTI AP boxplot, for only the car class.



(a) YOLOv8n (b) YOLOv8l

Figure 15: INDEV AP boxplot, for only the car class.

Table 6: KITTI R-squared value for a second-order polynomial fit on the confidence and bottom y-coordinate of bounding boxes as predicted by the YOLOv8 models. $K-S$ statistic for $\mu_o : F_{\text{real}} = F_{\text{sim}}$. All $K-S$ statistics and p-values for the ECDF of the distances were statistically different ($p < 0.001$) and thus not reported in this table. *: Based on ANCOVA, at least one of the coefficients of the second-order order polynomial fit is significantly different compared with the polynomial fit of the real data.

	YOLOv8n			**YOLOv8l**		
	R-squared		$K-S$	**R-squared**		$K-S$
Scene	**real**	**sim**		**real**	**sim**	
01	0.43	0.34	0.18	0.45	0.43*	0.17
02	0.31	0.20*	0.38	0.41	0.33	0.36
06	0.37	0.29	0.38	0.42	0.12*	0.41
18	0.51	0.28*	0.42	0.57	0.41*	0.45
20	0.32	0.28*	0.17	0.45	0.37*	0.14

Table 7: INDEV R-squared value for a second-order polynomial fit on the confidence and normalized, scaled, bottom y-coordinate of bounding boxes as predicted by the YOLOv8 models. $K-S$ statistic for $\mu_o : F_{\text{real}} = F_{\text{sim}}$. All $K-S$ statistics and p-values for the ECDF of the normalized distances were statistically different ($p < 0.001$) and and thus not reported in this table. *: Based on ANCOVA, at least one of the coefficients of the second-order order polynomial fit is significantly different compared with the polynomial fit of the real data.

	YOLOv8n			**YOLOv8l**		
	R-squared		$K-S$	**R-squared**		$K-S$
Scene	**real**	**sim**		**real**	**sim**	
daylight	0.09	0.09*	0.08	0.27	0.04*	0.16
daylight-rain	0.22	0.06*	0.21	0.33	0.03*	0.27
night	0.07	0.10*	0.27	0.33	0.09*	0.35
night-rain	0.11	0.07*	0.16	0.15	0.08*	0.23