Difference Analysis of Synthetic and Real Weather Effects in RGB Images

Theresa Maienschein[1,2][0009−0009−9668−9589], Rebecca Schmidt[2][0000−0002−9249−3812], Joachim Rüter[2][0000−0002−5559−5481], and Stefan Krause[2][0000−0001−6969−0036]

[1] Otto von Guericke University Magdeburg, FIN, ISG, Universitätsplatz 2, 39106 Magdeburg, Germany
theresa.maienschein@ovgu.de

[2] German Aerospace Center (DLR), Institute of Flight Systems, 38108 Braunschweig, Germany
{rebecca.schmidt,joachim.rueter,stefan.krause}@dlr.de

Abstract. Accurate environmental perception is crucial for autonomous vehicles but weather conditions such as rain or strong light can significantly impair the performance of camera-based systems. To ensure reliable operation under diverse weather conditions, simulation-based testing is increasingly employed. Currently, however, there is no established method to verify the realism of such simulations. Against this background, this work investigates the simulation of lighting and rain effects in Unreal Engine 5 by analyzing differences between real and simulated RGB images. An image dataset under different real-world weather conditions is collected and subsequently reproduced in the simulation environment. The resulting datasets are then compared using various metrics to assess how well simulated images reproduce the characteristics of real-world data that are relevant for computational perception, prioritizing quantitative image similarity over the physical correctness of the underlying scene. The results illustrate that real weather effects are characterized by complex dependencies - such as time of day, cloud cover, and camera-induced influences - which must be taken into account in simulations in order to achieve higher realism of the data. In addition, the difference analysis performed provides avenues for future research and improvements in simulation environments.

Keywords: Autonomous Vehicles · Perception · Synthetic Images · Image Similarity · Simulation-based Testing

1 Introduction

Reliable environmental perception through the use of various sensor technologies, like RGB cameras, is a key requirement for the safe operation of autonomous systems such as robots, self-driving cars, and Unmanned Aerial Vehicles (UAVs). It forms the basis for numerous safety-critical functions, such as collision avoidance or, in the case of UAVs, the identification of suitable landing sites. It has been shown that various weather conditions, such as rain, snow, and fog, can significantly impair the performance of environmental perception and thus pose a safety-related risk [30]. To enable the operation in such conditions, it is necessary to develop suitable approaches to increase the robustness of perception systems. In addition, efficient test procedures are needed to systematically evaluate new systems under adverse weather conditions. Conducting such tests in real-world environments, for example in flight tests, involves considerable challenges and risks. Simulations with variable and specifically adjustable weather conditions represent a promising alternative. However, because of the sim-to-real gap [6], the results of such tests are only meaningful if the underlying simulation is sufficiently realistic. In this context, realism does not necessarily refer to a completely physically accurate representation of the environment, but rather to a *phenomenologically accurate* representation that reflects the relevant properties of the environmental conditions for the virtual sensors and perception algorithms.

The literature on environmental perception under adverse weather conditions repeatedly indicates that there is a particular lack of datasets that explicitly take weather phenomena into account [15,25,30]. For this reason, numerous studies deal with the generation of new datasets or the expansion of existing ones [15,25,28]. However, these often consist of synthetically generated or augmented data with extensive scenes that do not focus exclusively on weather phenomena and do not have an existing real counterpart. For a systematic evaluation of the phenomenological correctness of the weather phenomena in simulation, a dataset with real data and minimized external influencing factors is required. Therefore, this work compares a real-world dataset of various weather conditions against a simulated counterpart featuring clear and rainy weather in a matching environment. The datasets are used to work towards developing a method for analyzing the differences between real and synthetic RGB images depicting weather phenomena in order to verify the phenomenological correctness of the synthetic data.

The main contributions of this work are twofold. First, a new real-world dataset depicting various weather conditions with minimal other influences is created, as well as a corresponding simulated dataset. The datasets are available at https://doi.org/10.5281/zenodo.20767236. Second, differences between the generated datasets are investigated using common image difference analysis metrics and used to evaluate the phenomenological correctness of the simulation. The proposed approach is translatable to other weather phenomena for future work.

2 Related Work

Research in the field of environmental perception under adverse weather conditions requires suitable datasets, but such weather phenomena are often missing from widely used datasets. Numerous studies attempt to close this gap by generating their own datasets. These are mostly synthetically created using augmentations [10,25] or simulations [15,28]. While some approaches are based on the physical properties of weather phenomena, such as raindrops [10,25], others are based on optical appearance [15,16]. Most of these studies mentioned pursuing the goal of generating datasets for training environmental perception systems. Accordingly, complete scenes are depicted, which make it difficult to analyze weather influences in isolation. Overall, it appears that real-world datasets with systematically complemented synthetic data are rarely created.

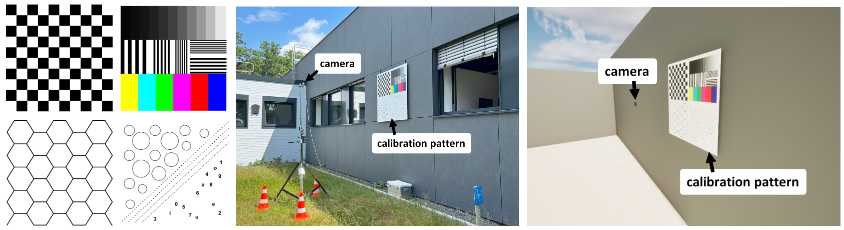
For difference analyses between synthetic and real-world images, numerous methods have been developed. Classifiers [29] and the evaluation of Machine Learning (ML) algorithm performance on real-world data [13,21] are two common methods for evaluating synthetic training data for ML models. However, one disadvantage of such ML-oriented approaches is that the decision process of ML models is opaque for humans. Accordingly, it is difficult to determine why it may be possible for the ML model to distinguish between real and synthetic data. Notably, [22] highlights color differences between synthetic and real-world images as having a significant influence on the generalization problems of ML models.

Another family of approaches aims at the direct evaluation of individual image properties such as brightness, contrast, or sharpness. There are often several different methods for such approaches [4,18]. While there is the disadvantage that these methods usually only provide global results and do not provide information about local structures, such methods are nevertheless helpful in identifying differences between images [12,19].

In addition to analyzing individual image properties, methods such as edge and corner detection can also be used to obtain information about the image content. These methods provide information about differences in the structures of images. Common edge detection methods include the Canny algorithm [7], the Sobel operator [24], and the Laplace filter [17]. The Harris Corner Detector [11] and Features from Accelerated Segment Test (FAST) [20] are particularly widely used for corner detection. An example of the application of such methods to compare two image datasets can be found in [8], where the differences between the datasets are evaluated based on their edge density, among other things, but also on simple image properties such as contrast and luminance.

While the analysis of image properties is based on individual images, so-called similarity measures are suitable for directly comparing two or more images. Such methods can be pixel-based, such as the mean squared error (MSE) [26], or combine several image properties, as in the case of the structural similarity index (SSIM) [27]. In practice, such methods are used, for example, to identify redundant images within a dataset [14], or to evaluate image quality by comparing original and reconstructed images using MSE and SSIM [23].

As shown, numerous methods covering a broad spectrum of low- to high-level image properties can be found in the literature. These methods differ in terms of their interpretability and are sometimes only suitable for specific applications. Previous work has often shown that the selection of methods is tailored to the respective application. In contrast, this work aims to evaluate the similarity be-



(a) Optical testing pattern (b) Real-world setup (c) Unreal Engine 5-based simulation setup

Fig. 1: Experimental setup for dataset generation

tween simulated and real images for applications of environmental perception, making the results applicable to both traditional and ML-based methods. Therefore, a selection of different approaches is presented below that take into account both low-level and high-level image features, thereby providing comprehensive insight into the data.

3 Dataset Generation

To examine whether simulated weather effects in images are phenomenological correct to perception algorithms, both real and simulated images need to be analyzed in the context of how computers see images. Furthermore, to ensure that the analysis highlights only differences caused by the effects under investigation - namely weather effects and their sim-to-real gap - a controlled environment is needed to minimize unwanted influences. This is especially necessary, as in datasets with dynamic and complex scenes like usually seen in real-world scenarios it is complicated to find the underlying reason for different results of the difference analysis. Since no existing dataset focuses exclusively on weather effects while enabling a comparison between real and simulated images, a new dataset was created, as described below.

To minimize effects of motion, a static scene was chosen for all images. Furthermore, a testing pattern was created as the main subject. This testing pattern, shown in Fig. 1a, features various structures and color scales, inspired by test cards used in various domains like camera calibration or movie technology, to enable different types of tests for this and future work, while simultaneously reducing the complexity of the scene dramatically. For example, the checkerboard pattern can be used to evaluate contrast, while hexagonal patterns are suitable for edge and corner detection. Color scales, in turn, can be used to assess changes in color.

Real-world Setup For the real-world setup the testing pattern is printed on a 1.5 m×1.5 m aluminum panel. The pattern is hung at a height of 3.45 m (top edge) on an exterior wall facing west. A GoPro HERO11 Black camera is positioned 2.5 m away from the pattern. The camera is centered on the pattern and the following settings were used: continuous recording with one image every five minutes, 5k resolution (5568x4872px), autoexposure activated, auto focusing activated and linear mode. The real-world setup is shown in Fig. 1b.

Simulation Setup Unreal Engine 5.3 is used as the simulation engine. The simulation environment is modeled on the real structure, as illustrated in Fig. 1c. The same georeference is set and the buildings were simplified. The windows are omitted due to the complexity of associated effects, and all such effects are eliminated from the simulation. The camera and optical marker are positioned according to the real-world measurements. The position of the sun and the ambient color are calculated using the inbuild SunSky Actor of the Unreal Engine [1]. Furthermore, a Niagara System with the options of adjusting the intensity and type of the rain is created for the rain simulation. The drop size of each raindrop (excluding drizzle) is randomly chosen from an inteval of 0.5-5mm, fitting the definition provided by the German Meteorological Service [3]. The length of the raindrops seen by the camera depend on the velocity of the raindrops and the camera exposure time. Based on [9], the velocity of raindrops (excluding drizzle) is assumed to be 6m/s. Lastly the number of raindrops is adopted according to the definitions for light, moderate and strong rain [2]. The intensity is calculated based on the defined amount of rain per time and the volume of the raindrops. Furthermore, the direction and speed of the wind can be adjusted according to the real wind. The simulated wind influences the falling raindrops in their fall direction. In addition falling raindrops hitting certain objects, like the optical test pattern, leave small drops on the objects, simulating real raindrops on objects. Unlike what is observed in reality no raindrops have been simulated on the camera lens so far.

4 Characterization of the Datasets

Over a period of 7 weeks in the summer months, 9793 real-world images are captured. These images are taken at all times of the day. Therefore, many images taken at night are black. In this work, only bright images taken at daytime or twilight are used. The filtering was automated based on a brightness threshold. Furthermore, these images are taken under different weather phenomena. Most images are taken at either clear, cloudy or rainy weather conditions. Only 10 images are taken while drizzle occurred, even fewer images show hail, fog, or thunderstorms. To identify the weather condition shown in the images, the weather broadcast of the nearby Brunswick airport weather station, which updates every 30 minutes, is used.

Out of the images taken at daytime or twilight, only those showing clear or rainy weather conditions and those which are taken a maximum of 10 minutes before or after a relevant weather broadcast are analyzed in this work. These images are then manually checked for undesired effects, which are not considered in the simulation environment, and therefore need to be sorted out of the dataset. In clear weather images, these effects are reflections from windows of surrounding

Table 1: Characterization of the generated datasets

Name	Number of Images	Type of origin	Weather-phenomena	Variations
Real-Complete	9793	real	clear, rain, cloudy drizzle, hail, thunderstorm, fog	original
Real-Clear	429	real	clear	original & cropped
Real-Rain	166	real	rain (all intensities)	original & cropped
Sim-Clear	429	simulated	clear	original & cropped
Sim-Rain	166	simulated	rain (all intensities)	original & cropped

buildings or lens flare. In rainy images the following three effects were searched for: falling raindrops, raindrops on the optical test pattern, and raindrops on the camera lens. The sorting results in two datasets, one for each weather condition. The clear weather dataset is called *Real-Clear* in the following and contains 429 images. The rainy images dataset is addressed by *Real-Rain* and consists 166 images. Both datasets are created in two variations: first with the original images as taken by the camera, in the following addressed by *original* and second with automatically cropped images, only showing the optical test pattern, in the following addressed by *cropped*.

Both datasets are recreated in the simulation environment. For the recreation, the date, the timestamp, and the weather information of each real image are used as input in the simulation. Consequently, new simulation datasets with the same number of images as the real datasets are generated. These datasets are called *Sim-Clear* and *Sim-Rain*. For an overview of the datasets, see Table 1.

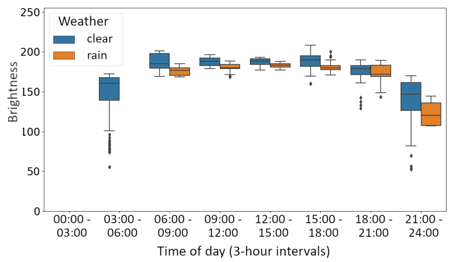
5 Difference Analysis

This section presents the difference analysis, which aims to provide a comprehensive validation of the phenomenological accuracy of the simulated images. Therefore 12 metrics are selected, covering complementary aspects of image characteristics and established analysis approaches. These metrics range from simple global properties such as brightness, to structural descriptors like edge and corner detectors, as well as perceptual similarity measures and classifier-based evaluations. This selection aims to cover both low-level image properties and higher-level, application-relevant characteristics.

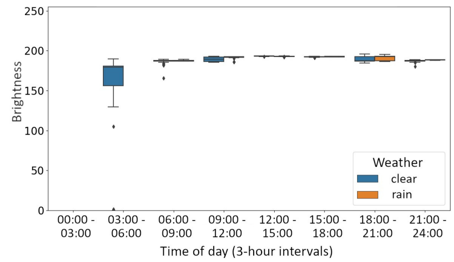
In the following subsections, the most representative metrics are discussed in detail. Specifically, brightness, structural features, i.e. edge and corner detection, the Structural Similarity Index (SSIM), and classifier-based evaluations are examined more closely. For each of these, the results are presented first, followed by an interpretation of their significance. The remaining metrics are addressed in a more compact form in the last subsection.

5.1 Brightness

Since brightness is one of the fundamental characteristics of an image the analysis starts with this measure. The brightness of the images is measured by the



(a) Brightness depending on the time of day for cropped real-world images



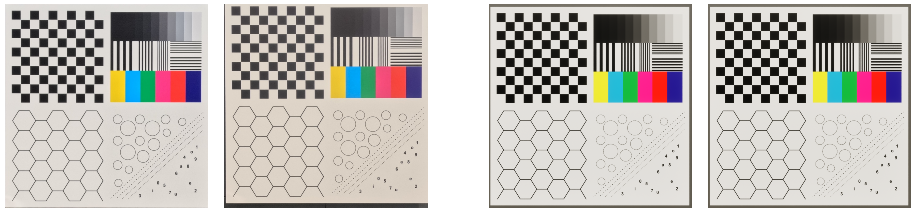
(b) Brightness depending on the time of day for cropped simulated images

Fig. 2: Daytime dependence of image brightness

quadratic mean of the pixel values after transforming the images into grayscale images. Since the background of the original images may significantly influence brightness, only cropped images were used for the analysis. As image brightness is likely correlated with the time of day, all images were grouped into non overlapping three-hour intervals (00:00-03:00, 03:00-06:00, ..., 21:00-24:00) based on their capture time.

Results The brightness of the real-world images depending on the daytime is shown in Fig. 2a. The figure shows a clear dependency of the brightness on the daytime, both for Real-Clear and Real-Rain images. Both dataset images that are taken early in the morning or late in the evening, are significantly darker than images taken during daytime. Furthermore, images of the Real-Rain dataset tend to be darker, than Real-Clear images, taken at the same time interval. and also have a smaller brightness variance for most of the time intervals. The figure also shows that no image before 3 a.m. was bright enough for the before mentioned criteria and, therefore, filtered out. Furthermore, rainy images for the time interval of 03:00 to 05:00 a.m. are missing. The brightness of the simulated images depending on the daytime is shown in Fig. 2b. As for the real-world images, the brightness of Sim-Clear and Sim-Rain images depends on the daytime, but the dependency is significantly weaker than the real-world data. Furthermore, except for images taken early in the morning, the brightness variance is much lower for the images compared to the real-world data. Simulated rain images from Sim-Rain can even be brighter than images of the Sim-Clear dataset.

Discussion In the real-world images, the dependency of the brightness on the daytime is explained by the movement of the sun and the changing ambient light. Differences between the images of the Real-Clear and the Real-Rain datasets are likely due to clouds or further cross dependencies of rainy weather. Especially lower brightness in rainy images can be coupled to clouds blocking the sunlight. The lower variance could also be attributed to the clouds, as they provide more



(a) Real-Clear image, June 21, 2025, at 2:51 p.m.
(b) Real-Clear image, July 14, 2025, at 2:53 p.m.
(c) Sim-Clear image, June 21, 2025, at 2:51 p.m.
(d) Sim-Clear image, July 14, 2025, at 2:53 p.m.

Fig. 3: Brightness differences between real-world and simulated images taken at similar times of day

consistent ambient lighting conditions. However, this needs further investigation, as a lower number of images in the Real-Rain dataset might also contribute to a lower variance. In the simulated images, the dependence of the brightness on the daytime can be explained by the simulated movement of the sun, just like for the real-world images. The lower dependence compared to real-world images, shows that the simulation of time-dependent lighting conditions is not yet realistic enough. Furthermore, cross-dependencies are missing in the simulation. Clouds in particular were not taken into account in the rain simulation, which may be one reason why the simulated rain images appear comparatively bright. On top of missing clouds, raindrops appear brighter in the simulated images than in the real world images, which further increase the brightness and could explain why some simulated rain images are even brighter than simulated clear weather images. Compared to the real-world data, the simulated images generally have a lower variance in brightness, as shown in Fig. 3. This may be attributed to the absence of interfering influences and cross-dependencies in the simulation, as well as an incorrect or imprecise simulation of the sun's position. While two real-world images taken at the same daytime and with the same weather effect can vary in their appearance, as shown in Figs. 3a and 3b, all simulated images look the same if they were simulated for the same daytime, as illustrated in Figs. 3c and 3d. In summary, the analysis of image brightness shows that cross-dependencies due to e.g. clouds and further interfering influences have a major impact on basic image characteristics and must be taken into account in simulations.

5.2 Structural Features

A structural feature analysis adds to the simple global metrics like brightness, by giving information about local structures like geometry, contours, lines and edges. For the structural feature analysis in this work, a Canny-Edge-Detector in combination with the Hough-Transformation and a Shi-Tomasi-Corner-Detector is applied to the cropped datasets. These algorithms are widely used and therefore suitable as representatives. This work highlights existing challenges when applying the algorithm for image similarity measurements and discusses the results. The algorithms feature several adjustable parameters that significantly

Table 2: Number of edges in real and simulated images depending on clear or rainy weather after the Hough-Transformation.

	lower bound	upper bound	outlier	average	median	standard deviation
Edge number (Real-Clear)	397	637	/	544	546	40
Edge number (Real-Rain)	466	638	6068	564	562	36
Edge number (Sim-Clear)	346	537	0	414	415	29
Edge number (Sim-Rain)	405	571	/	458	451	35

(a) All detected edge pixels of a slightly rotated image

(b) Zoomed staircase effect in green area of (a)

(c) Sim-Clear image with highest number of corners

(d) Detected corners of the Sim-Clear image.

Fig. 4: Challenges of using structural feature metrics

influence the results. In the future a comprehensive analysis with several sets of parameter settings will be necessary to truly understand the influence of different weather effects and the simulation. In this work, only one set of default parameters are used for each algorithm.

Results The results of the edge detection, illustrated in Table 2, show a discrepancy between the number of edges in real-world and simulated images. The simulated images generally have fewer edges than their real counterparts, both for rain and clear weather. Real-world images have an average of 544 edges per image, while simulated images have an average of only 414 edges. This is particularly interesting, as even cropped Real-Clear images and cropped Sim-Clear images exhibit different edge values, even though these configurations are very similar due to the cropping of the background and the absence of raindrops. Another noticeable effect when using the detectors is most evident in outliers in corner detection. Despite the real-world images having more edges, the images with the highest number of corners are simulated images. These images were simulated to depict the early morning hours. While the corresponding real-world images were already bright enough to be included, the simulated images appear completely black to the human eye. Nevertheless, these images have a high number of corners, as shown in Figs. 4c and 4d.

Discussion As for the brightness, this analysis used cropped images. Since the cropping was done automatically and was not checked afterwards, it is vulnerable to translations and rotations of the optical marker inside the original image. In

Table 3: SSIM results for real and simulated image pairs.

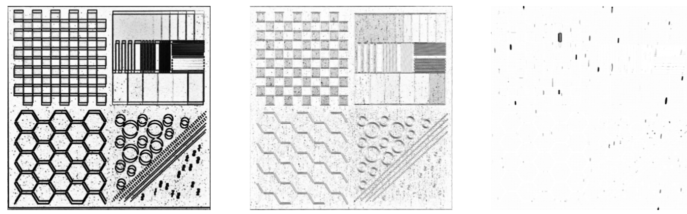
	lower bound	upper bound	outlier	average	median	standard deviation
real image pairs	0.55	0.93	0.254	0.75	0.73	0.12
simulated image pairs	0.958	0.996	/	0.99	0.99	0.006

the real-world setup, the camera can be moved slightly by the wind or whenever work needs to be done at the camera, for example when restarting the camera. These slight movements can manifest in small translations and rotations of the optical test pattern in the original images and therefore in problems with the cropped image, as illustrated in Figs. 3a and 3b. In contrast such slight movements do not occur in the simulation. These rotations in the images can result in the staircase effect for edges, as shown in Figs. 4a and 4b. Depending on the used parameters, instead of one edge several edges are identified. As only real-world images are affected from this effect, the simulated images have less edges. Yet, without analyzing every single image it is not possible to evaluate how much this effect has an influence and whether other effects may contribute to the different number of edges between real and simulated images.

5.3 SSIM

SSIM compares two images and returns a value for their similarity. This metric is based on the comparison of luminance, contrast and structure between the images and returns a value between 0 for completely different images and 1 for identical images. Accordingly it is necessary to create image-pairs. Since the real-world and simulated images have differences in brightness values, as shown in section 5.1 and rain effects in the real-world images cannot be exactly replicated by a simulation, it is not reasonable to combine real-world and simulated images for image pairs. Instead, clear weather and rainy images are combined for the following analysis. For each rainy image a clear weather image is searched which is taken at the same daytime or a maximum of 10 minutes earlier or later. In case several clear weather images are found, the image taken closest to the date of the rainy image, is used for the pair. With these criteria 124 image pairs are created each for real-world and simulated images. For all image pairs the cropped image versions were used.

Results The SSIM results for the real-world images pairs are shown in Table 3, showing values between 0.55 and 0.93 with one outlier at 0.254. The standard deviation of 0.12 highlights the large variance. The median of 0.73 is relatively low for images, that should only differ in effects based on different weather conditions. In contrast, the SSIM values of the simulated image pairs lie between 0.958 and 0.996 with a standard variation of 0.006 and a median of 0.99. This shows, that the simulated image pairs are much more similar, than their real counterparts.



(a) SSIM of two real images with rotation effects.

(b) SSIM of two real images without rotation effects but noise.

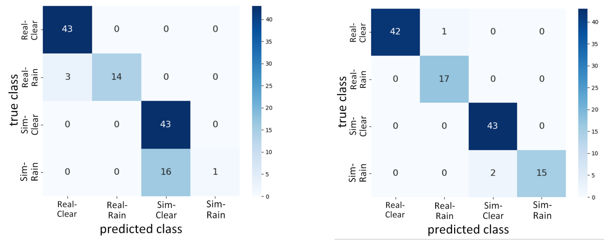
(c) SSIM of two simulated images.

Fig. 5: SSIM difference images.

Discussion The high variance of SSIM values for the real-world image pairs, cannot be explained by the expected differences between clear and rainy weather, namely falling raindrops and raindrops on the optical test pattern or the camera lens, as these differences only affect small parts of the image. Therefore, other effects must have an influence on the SSIM. Looking at difference images, one big influence is a slight rotation in the images. In Fig. 5a the difference image created using the skimage.metrics package of an image pair with strong translation and rotation influence can be seen. Further influences might also be noise resulting in difference images with mainly gray background, as shown in Fig. 5b. In contrast, the simulated image pairs all have a high SSIM value. The difference images only show the raindrops, see Fig. 5c. This accounts to the fact, that simulated images are very similar to each other and miss interfering influences like camera translation and rotation and cross dependencies. Because of the significant influence of the cropping errors, the influence of weather effects in real image pairs is obscure. Therefore the analysis of the simulated image pairs is equally unexpressive. For further studies, it must be made sure, that the differences in images of a pair is solely based on weather effects. Furthermore, in case that the SSIM values, for image pairs with only weather effect differences, are as high as for the simulated image pairs, it must be checked, whether SSIM is a good metric for such analysis at all.

5.4 ResNet-18 Classifier

Although the decision-making process of many state-of-the-art classifiers is not directly interpretable, the results can still be used to get an impression of the phenomenological accuracy of simulated images for ML models. In case simulated images are realistic to ML models, a classifier should not be able to differentiate between real-world and simulated images. Accordingly ResNet-18 models were trained to classify Real-Clear, Real-Rain, Sim-Clear and Sim-Rain images. In contrast to the previous analyses for the classification both cropped and original images were tested. First a classifier was trained solely on cropped images, then another classifier was trained on original images. The datasets were split 80-10-10 for training, validation and test.



(a) Confusion matrix for model trained on cropped images.

(b) Confusion matrix for model trained on original images.

Fig. 6: Confusion matrices for classifiers.

Results The confusion matrix for the model trained on cropped images, illustrated in Fig. 6a, indicates that the classifier can clearly distinguish between real-world and simulated images, as there are no misclassifications between those classes. Furthermore, for the real-world images, the classifier is mostly correct with the weather classes too. For the simulated images however, nearly all images are classified as clear. This results in an overall accuracy of 84.17%. Once the original images are used for training, the classifier can correctly classify all of the classes, as shown in Fig. 6b. Between real-world and simulated images there is no error. The weather classifications are not perfect, but most images are correctly classified and the accuracy for this model is 92.5%.

Discussion Seeing that the classifier does differentiate correctly between real-world and simulated images, the difference between those are still significant for the goal of reaching phenomenological accuracy. Furthermore, the analysis shows, that solely based on the optical pattern part of an image, the classifier cannot perceive clear features for rain. Using also the background seems to add enough features though. This indicates a lack of details, necessary to capture relevant features, in comparison to the real-world images. Yet due due to the limited number of images available in the dataset the results should be interpreted with caution. Although some findings, like the lack of details in simulated images, appear to be relatively clear, additional studies based on larger datasets are required to confirm their validity and generalizability.

5.5 Further Methods

Next to the described metrics the datasets were also analyzed with further metrics (see Table 4 for the results). In addition to brightness two more global measures were analyzed: contrast and MLV sharpness [5]. Their results support the findings obtained from brightness analysis, indicating that cross-dependencies, like cloud cover, can strongly influence the analysis. The analysis also suggests that the simulations lack certain interfering influences, including image noise. Slthough global metrics cannot capture local phenomena, these results prove

Table 4: Results of further metrics used for the analysis.

	lower bound	upper bound	outlier	average	median	standard deviation
contrast (Real-Clear)	21.81	68.31	/	49.28	51.22	7.48
contrast (Real-Rain)	37.10	59.62	/	53.64	54.68	4.16
contrast (Sim-Clear)	51.22	76.36	0.25 - 0.78	62.12	60.12	12.61
contrast (Sim-Rain)	58.38	76.36	/	71.57	74.89	6.28
MLV sharpness (Real-Clear)	6.42	16.89	/	14.48	14.73	1.76
MLV sharpness (Real-Rain)	11.14	17.08	/	15.57	15.87	1.32
MLV sharpness (Sim-Clear)	15.82	24.29	0.45 - 0.5	19.86	20.00	3.95
MLV sharpness (Sim-Rain)	17.00	24.39	/	22.67	23.64	1.90
NRMSE real image pairs	0.038	0.36	/	0.20	0.22	0.08
NRMSE simulated image pairs	0.006	0.06	/	0.03	0.03	0.01
PSNR real image pairs	11.81	33.33	/	17.47	15.81	4.28
PSNR simulated image pairs	26.80	47.02	/	34.74	34.25	4.13

they provide valuable initial insights into factors limiting the phenomenological correctness of the simulation. The combination of different methods may give even further insight.

Furthermore, Normalized Mean Squared Error (NRMSE) and Peak Signal to Noise Ratio (PSNR) were analyzed alongside SSIM. Similar to the SSIM results, their interpretability is limited by the sensitivity of these metrics to rotations and translations.

Finally, the ResNet-18 classifier analysis was extended by evaluating cross-dataset generalization. Consistens with the previous results, models trained on original images generalized better than those trained on cropped images. In addition, models trained on simulated images consistently underperformed compared to models trained on real-world images, suggesting that the simulatd data lacks sufficiently distinctive features for robust model training.

6 Summary and Future Work

This work presents an approach to identify differences between simulated and real-world images, working towards the identification of phenomenological correctness of simulated images. A new weather-focused real-world dataset was created, containing images of an optical test pattern designed for comprehensice optical testing. The setup was replicated in Unreal Engine. Together a dataset of 9793 real and more than 600 corresponding simulated images was created and is made available for future research. The dataset itself constitutes a major contribution of this work, as comparable datasets providing closely matched real and simulated images under controlled weather conditions are scarce, and no equivalent dataset was identified druing the literature review.

A difference analysis was conducted using 12 selected metrics, with brightness, SSIM and ResNet-18 classifiers serving as representative examples. The results indicate that realistic weather simulation requires not only modeling primary effects, such as raindrops, but also related phenomena including cloud cover and illumination changes. These cross-dependencies can influence the analysis more strongly than the investigated effects themselves, emphasizing the need for further studies to better understand and reproduce such dependencies, as well as finding new dependencies.

The presented analysis has some limitations. To reduce complexity, certain real effects were excluded, reducing the number of images in the small datasets even more before the analysis. Consequently, larger datasets (covering additional seasons, different camera configuration and scene variations) are required to validate the findings.

Furthermore this work highlights that some metrics are sensitive to image rotation, translations, noise and parameter choices. Such sensitivities should be considered in future analyses, as they may also affect downstream algorithms relying on similar image features. In addition, real-world images are subject to currently unidentified influences even though weather-independent interference was largely limited.

Overall, the results show that the presented simulation cannot yet be considered phenomenologically correct and underline the importance of comprehensice difference analyses. Future work should further improve the simulation and evaluate additional analysis methods, including ML-based feature extractors, to establish efficient and meaningful approaches for comparing simulated and real-world images.

References

1. Sun and Sky Actor in Unreal Engine | Unreal Engine 5.6 Documentation | Epic Developer Community, https://dev.epicgames.com/documentation/en-us/unreal-engine/sun-and-sky-actor-in-unreal-engine, accessed: 30 September 2025
2. Wetter und Klima - Deutscher Wetterdienst - Glossar - Niederschlagshöhe, https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?lv2=101812&lv3=101904, accessed: 22 July 2025
3. Wetter und Klima - Deutscher Wetterdienst - Glossar - Regen, https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?lv2=102134&lv3=102200, accessed: 22 July 2025
4. Avatavului, C., Prodan, M.: Evaluating image contrast: A comprehensive review and comparison of metrics. Journal of Information Systems **17** (2023)
5. Bauer, P., Thorpe, A., Brunet, G.: The quiet revolution of numerical weather prediction. Nature **525**(7567) (2015)
6. Bornscheuer, T., Leferenz, L., Schmidt, R., Rüter, J.: Mixed-reality simulation for testing complex UAS missions in safe environments. In: AIAA SCITECH Forum (2026)
7. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (1986)
8. Cooper, P.S., Colton, E., Bode, S., Chong, T.T.J.: Standardised images of novel objects created with generative adversarial networks. Scientific Data **10**(1) (2023)
9. Gunn, R., Kinzer, G.D.: The terminal velocity of fall for water droplets in stagnant air. Journal of Atmospheric Sciences (1949)
10. Halder, S.S., Lalonde, J.F., Charette, R.d.: Physics-Based Rendering for Improving Robustness to Rain (2019), arXiv:1908.10335 [cs]

11. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: Procedings of the Alvey Vision Conference. Alvey Vision Club, Manchester (1988)
12. Hartmann, N., Rüter, J., Jünger, F.: Ensuring Safety of Deep Learning Components Using Improved Image-Level Property Selection for Monitoring. In: AIAA SCITECH Forum (2025)
13. Hinniger, C., Rüter, J.: Synthetic training data for semantic segmentation of the environment from uav perspective. Aerospace **10**(7) (2023)
14. Kapoor, P., Agrawal, P., Ahmad, Z., Narayanamoorthi: Duplicate Image Detection and Comparison using Single Core, Multiprocessing, and Multithreading (2021)
15. Liu, D., Cui, Y., Cao, Z., Chen, Y.: A Large-scale Simulation Dataset: Boost the Detection Accuracy for Special Weather Conditions. In: IJCNN (2020)
16. Marchisio, A., Caramia, G., Martina, M., Shafique, M.: fakeWeather: Adversarial Attacks for Deep Neural Networks Emulating Weather Conditions on the Camera Lens of Autonomous Systems (2022), arXiv:2205.13807 [cs]
17. Marr, D., Hildreth, E.: Theory of Edge Detection. Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character **207** (1980)
18. Pagaduan, R.A., R. Aragon, M.C., Medina, R.P.: iBlurDetect: Image Blur Detection Techniques Assessment and Evaluation Study:. In: Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies (2020)
19. Rajput, P., Kumari, S., Arya, S., Lehana, P.: Effect of Contrast and Brightness on Digital Images under Outdoor Conditions. Advances in Research (2014)
20. Rosten, E., Drummond, T.: Machine Learning for High-Speed Corner Detection. In: ECCV. Springer, Berlin, Heidelberg (2006)
21. Rüter, J., Durak, U., Dauer, J.C.: Investigating the sim-to-real generalizability of deep learning object detection models. Journal of Imaging **10**(10) (2024)
22. Rüter, J., Durak, U., Dauer, J.C.: Investigating the influence of image augmentations on the sim-to-real generalization of deep learning perception models. In: Proceedings of the Synthetic Data for Computer Vision Workshop @CVPR (2025)
23. Sara, U., Akter, M., Uddin, M.S.: Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. Journal of Computer and Communications **7**(3) (2019)
24. Sobel, I.: An Isotropic 3x3 Image Gradient Operator. Presentation at Stanford A.I. Project 1968 (2014)
25. Tremblay, M., Halder, S.S., de Charette, R., Lalonde, J.F.: Rain Rendering for Evaluating and Improving Robustness to Bad Weather. International Journal of Computer Vision **129**(2) (2021)
26. Wang, Z., Bovik, A.: Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. IEEE Signal Processing Magazine **26**(1) (2009)
27. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4) (2004)
28. Yang, D., Liu, Z., Jiang, W., Yan, G., Gao, X., Shi, B., Liu, S., Cai, X.: Realistic Rainy Weather Simulation for LiDARs in CARLA Simulator (2023), arXiv:2312.12772 [cs]
29. You, Z., Zhang, X., Guo, H., Wang, J., Li, C.: Are Images Indistinguishable to Humans Also Indistinguishable to Classifiers? (2025)
30. Zhang, Y., Carballo, A., Yang, H., Takeda, K.: Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. ISPRS Journal of Photogrammetry and Remote Sensing **196** (2023)