Investigating the Influence of Image Augmentations on the Sim-to-Real Generalization of Deep Learning Perception Models

Anonymous CVPR submission

Paper ID 2

Abstract

Since large and diverse datasets required for training deep learning (DL) models are often unavailable, synthetic images generated from game-engines are increasingly used. However, DL models trained on synthetic images often struggle to generalize well to real-world images. This study systematically investigates the potential of image augmentation techniques to improve the sim-to-real generalization. To do so, we evaluate the influence of 25 basic pixel-level image augmentations on the real-world performance of various DL models trained solely on synthetic images. The comprehensive study covers multiple DL models, datasets, and perception tasks, including object detection and semantic segmentation. Our results show that image augmentations are a promising approach to increase sim-to-real generalization. Specific augmentations can significantly enhance model performance on real-world datasets, improving the median performance of the investigated models by over 5% and yielding maximum improvements of up to 26.8%. Furthermore, we show that especially differences in color and blur are significant factors contributing to the sim-to-real generalization problems of DL perception models.

1. Introduction

For many perception tasks, Deep Learning (DL) models achieve state-of-the-art results. To do so, they rely on huge and diverse datasets during training. For many use-cases, this data is not available because of safety and cost reasons, hindering the deployment of DL models. To mitigate this lack of real-world images, the use of synthetic images generated with increasingly realistic visual simulation environments has gained more and more attention. While a promising approach, DL perception models trained on synthetic images usually have problems generalizing to real-world images. Although this generalization problem is extremely relevant for real-world applications, definite reasons for it are not yet found.

The generalization problems could come from many different sources. Previous work has mostly looked into the learning process using *e.g*. domain adaptation techniques [25, 38]. On model level, [33] recently investigated the influence of the DL model architecture on the generalization capability from synthetic to real-world images and showed that different DL models differ significantly in this capability. On data level, not much work has been done.

A natural approach on data level to improve the sim-to-real generalization of DL models is to try to increase the photorealism of the visual simulation environment. Bringing the synthetic images closer to the real-world domain by increasing the photorealism should naturally improve generalization. However, increasing photorealism is a complex problem on which the gaming industry has been working for years, and is therefore infeasible for practitioners as well as industry users in the near future.

As an alternative approach, this work investigates the potential of image augmentation techniques during training of DL models to improve the sim-to-real generalization. Basic image augmentations have been shown to improve performance when using real-world datasets [2, 36] as well as to improve robustness on real-world distributions shifts [10]. However, it is not yet known to what extend image augmentations can help to reduce the shift from synthetic to real-world images. Furthermore, investigating which image augmentations are most beneficial to close the shift can also give further insights into the shortcomings of synthetic images and the differences to real-world ones that are relevant for DL models.

Specifically, this work considers 25 basic pixel-level image augmentations such as adding blur or noise, adjusting the contrast of images as well as changing image colors. It trains different DL models only on synthetic images in combination with image augmentations and evaluates the influence of the augmentations on the real-world performance. To improve the generalizability of the results, this work considers multiple DL models, datasets and perception tasks. On the one hand, it considers object detection in a visually simple in-air coupling situation. On the other hand, it considers two semantic segmentation use-cases, one coming from the autonomous driving and the other from the Unmanned Aircraft System (UAS) sector. As DL models, five semantic segmentation models on the two datasets as well as three object detection models on the corresponding dataset are considered.

The main contributions of this work are as follows. First, it gives a comprehensive study of the isolated influence of 25 image augmentations on the sim-to-real generalizability of DL models for object detection as well as semantic segmentation. It shows a positive effect of specific augmentations. Second, it expands the results by giving first insights into combinations of augmentations. Overall, it provides practical insights on the selection of image augmentations to improve sim-to-real generalization when using state-of-the-art game-engines. By doing so, this work plays a crucial role in the deployment of DL models in domains with a limited amount of real-world data.

2. Related Work

As large and diverse datasets needed by DL perception models to achieve state-of-the-art results are often not available, recent years have seen a growing interest in using synthetic images extracted from game-engines for training. It spans multiple domains like autonomous driving [13, 27, 28, 39] as well as UAS use-cases [11, 14, 18, 19, 31, 32]. While using synthetic images is a promising approach because of the increasing realism of the engines, numerous studies have shown that these images are usually insufficient to train DL models that generalize well to real-world data, as models often suffer significant performance drops when applied outside the synthetic domain [11, 14, 18, 19, 28, 39]. This issue is commonly referred to as the sim-to-real gap. While it was shown that the gap can be closed when real-world images are available, *e.g*. using domain adaptation strategies [25, 38] or by combining synthetic and real-world images during training [11, 14, 18, 19, 27, 28, 39], identifying influencing factors on the gap when only using synthetic images remains an open problem.

While [33] investigated first influences coming from the model, there is only limited work systematically exploring the influence of the synthetic data itself. There are some works investigating different image property metrics to measure the differences between the synthetic and real-world images. For example, [17] shows for the KITTI dataset [7] and its synthetic counterpart VKITTI [6] that even one metric for noise is enough to differentiate the datasets as the synthetic images contain almost no noise. They further conclude that the biggest influence on the sim-to-real gap comes from the general coloration, lighting conditions and a lack of noise [17].

There are also some works going deeper into the investigation of the influence of noise and sensor modelling for synthetic data on the sim-to-real gap [1, 4, 8, 34]. Most of these modellings are done using image augmentations. For example, [1] shows performance improvements when matching the distortion of the real-world and the virtual camera. Furthermore, [8] models sensor lens artifacts and shows that it improves the mIoU for semantic segmentation network trained on the synthetic images. Similar finding were made when modelling camera vignetting [34]. The influence of modelling different sensor effects including Chromatic Aberration, Blur, Exposure, Sensor Noise and Color Shift is investigated in [4]. It shows that applying each effect as well as the combination of all improves the performance compared to a baseline Faster R-CNN object detection model. Going in a similar direction, [29] shows in general that image augmentation can improve object detectors trained on synthetic images but does not specify which augmentations are used.

Compared to the works described above, this work investigates more sensor effects and augmentations as well as more DL models and more datasets which cover multiple domains and multiple perception tasks such as semantic segmentation and object detection. Furthermore, it presents a systematic evaluation considering the effect of each augmentation in isolation as well as combinations. It gives a detailed evaluation on which augmentations improve the real-world performance of DL models and which do not. Overall, this increases the generalizability of the results to other domains and gives more detailed insights for practical applications.

3. Experimental Setup

3.1. Method and Evaluation Metrics

To investigate the influence of applying different image augmentations during training on the sim-to-real generalizability of DL perception models, following approach is used. First, baseline models are trained only on the synthetic training datasets without the use of image augmentations. Afterwards, the models are trained on the same synthetic training datasets but using different image augmentations. As this research aims to investigate the generalization to real-world images, a real-world validation dataset is used to make sure that the model that performs best on real-world images is selected for evaluation. It is important to note that the use of a real-world validation dataset does not influence the training itself as the model weights are trained only using synthetic images. After training, all models are evaluated on real-world datasets that correspond to the synthetic training data. All models are trained three times and the best performing training run is selected for further evaluations. To measure the influence d_a of the image augmentation a on the sim-to-real generalizability of the model m, the performance $e(m_a)$ of the model m_a trained with augmentation a is related with the performance $e(m_b)$ of the baseline model m_b using

$$d_a = \frac{e(m_a) - e(m_b)}{e(m_b)}.$$

It measures the relative improvement compared to the baseline model. As mentioned in the introduction, this work considers the tasks of object detection as well as semantic segmentation to increase the generalizability of the results. Both tasks use different evaluation metrics. To evaluate the performance of the object detection models, this work employs the widely used mean Average Precision (mAP) averaged over Intersection over Union (IoU) thresholds 0.5, 0.55, ..., 0.95 from the COCO evaluation [20]. To evaluate the semantic segmentation performance, the mean Intersection over Union (mIoU) is used. As it is clear from the context which evaluation metric is used, no explicit discrimination is made in the remainder of this work.

3.2. Datasets

This section briefly describes the datasets considered in this work. They include one object detection dataset as well as two semantic segmentation datasets.

For object detection, the dataset from [32] is used. It contains real-world images showing in-air coupling maneuvers of two aircraft during air-to-air refueling recorded from the perspective of the trailing aircraft. Furthermore, it contains synthetic images showing similar situations generated using the Unreal Engine [5]. The dataset contains 8,000 synthetic training images as well as 518 real-world validation images and 4,969 real-world evaluation images. Following [33], all images are center cropped to 544x544 pixels and scaled to 320x320 pixels to improve training speed.

The first semantic segmentation dataset is from the domain of autonomous driving. As the real-world dataset, the dataset from [16] is used. It provides 445 semantic segmentation annotations for a subset of the KITTI dataset [7]. From these, 146 images are used as the real-world validation dataset and the remaining 299 images are used as the real-world evaluation dataset. As the synthetic training dataset, the Virtual KITTI 2 dataset [3] is used. The dataset is build to be similar to the original KITTI dataset by cloning five scenes using a so-called real-to-virtual cloning method. The scenes are disjoint from the real-world ones used for evaluation. Overall, 1,892 synthetic training images are available, coming from the five scenes except from scene two which was reserved for potential future evaluation on synthetic images. Because the real-world and synthetic datasets have some variations in the class annotations, the classes are merged to the final classes *Sky*, *Building*, *Vehicle*, *Vegetation*, *Sign/Pole*, *Ground* and *Other*. Furthermore, all images are center cropped to the smallest common size of 1226x370 pixels.

Table 1. Baseline results of each model on the respective real-world evaluation dataset when trained only on synthetic images combined with number of trainable weights. RS denotes Ruralscapes, MNv3 denotes MobileNetV3.

Dataset	Architecture	Backbone	Perf	Weights
RS	UPerNet	ResNet-50	49.5	66M
RS	UPerNet	ResNet-101	44.6	85M
RS	UPerNet	Swin-T	35.6	60M
RS	UPerNet	Swin-S	39.4	81M
RS	UPerNet	Swin-B	39.2	121M
KITTI	UPerNet	ResNet-50	54.3	66M
KITTI	UPerNet	ResNet-101	52.8	85M
KITTI	UPerNet	Swin-T	51.2	60M
KITTI	UPerNet	Swin-S	61.2	81M
KITTI	UPerNet	Swin-B	58.4	121M
Drogue	Faster R-CNN	MNv3-L	76.7	19M
Drogue	Faster R-CNN	ResNet-50	78.1	41M
Drogue	Faster R-CNN	VGG-16	52.4	44M

The second semantic segmentation dataset is recorded from a low-flying UAS with a tilted camera at multiple altitudes in a rural area. As the real-world dataset, the Ruralscapes dataset [22] is used. It provides 13 video sequences containing a total of 816 annotated images designed for training and 7 sequences with a total of 331 annotated images designed for evaluation. In this work, a subset of 20% of the training images, i.e. 164 images, is used as the real-world validation dataset as the remaining ones were reserved for potential future training on real-world images. The 331 designated evaluation images are used as the real-world evaluation dataset. For training, the synthetic dataset from [11] is used. It stylistically replicates the flight area of [22] in the Unreal Engine and contains images as well as labels from similar UAS perspectives. In this work, 1,569 synthetic images are available for training. Again, because the real-world and synthetic datasets have some variations in the class annotations, the classes are merged to *Street*, *Building*, *Car*, *Human*, *Greenery* and *Background*. To reduce computational load during training and evaluation, all images are scaled to 960x540 pixels.

3.3. Model Configurations and Training Settings

As mentioned in the introduction, this work investigates multiple object detection and semantic segmentation models to increase the generalizability of the results. All considered perception models are shown in Tab. 1 in combination with the number of trainable weights and the performance of the baseline model on the real-world images when trained only on synthetic images without augmentations.

For object detection, Faster R-CNN [26] architectures with VGG-16 [37], ResNet-50 [9] FPN and MobileNetV3-

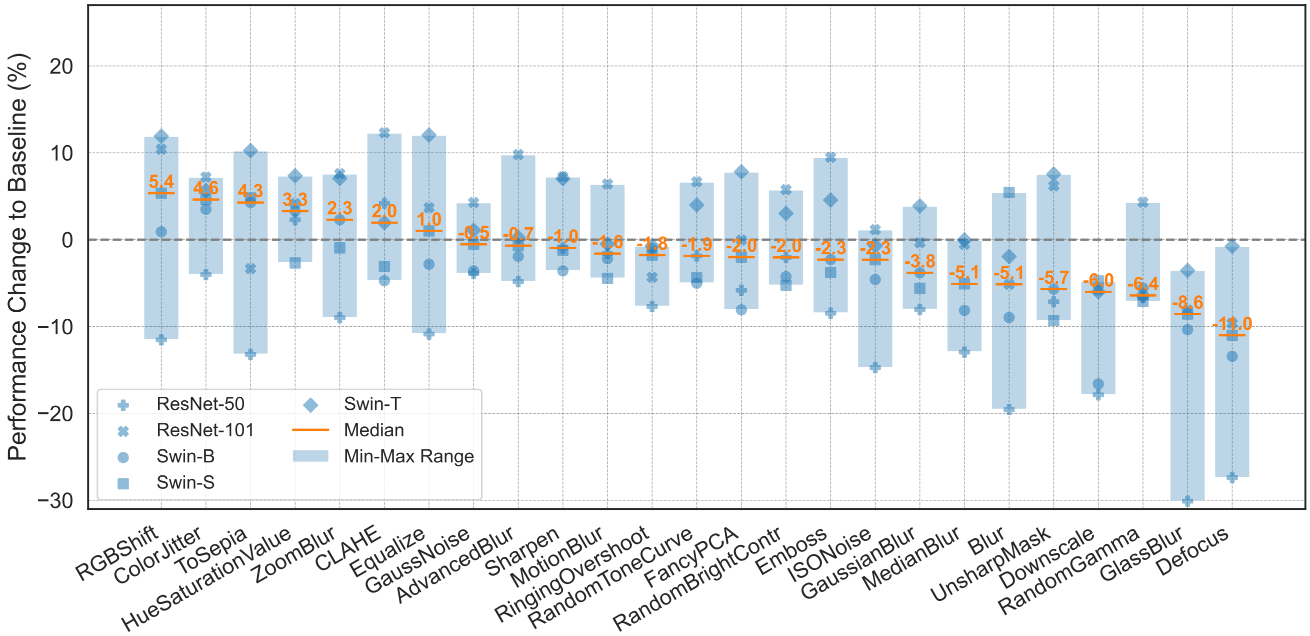

Figure 1. Change of performance d on the real-world Ruralscapes evaluation datasets when applying a specific image augmentation during training compared to the baseline model.

Large [12] FPN backbones are trained and evaluated. All models are implemented in the PyTorch library and the backbones are pre-trained on the ImageNet dataset [30] as provided by PyTorch. To avoid introducing bias because of unsuitable training hyperparameters, [33] performed a hyperparameter search to identify parameters that lead to the best performance on the real-world dataset. Following this work, all models are trained using Adam [15] optimizer. The models with VGG-16 and MobileNetV3-Large backbones use a learning rate of 0.00001 and the model with the ResNet-50 backbone a learning rate of 0.0001. Each model is trained for a maximum of 200 epochs with a training batch size of 16. To reduce computational load when the model has reached a local optimum and does not improve anymore, early stopping with a patience of 10 is used.

For semantic segmentation, UPerNet [40] architectures with ResNet-50 and ResNet-101 [9] as well as Swin-B, -S and -T [21] backbones are trained and evaluated. All models follow the implementation provided in the MMSegmentation library and are pre-trained on the ADE20k dataset [41]. Following a hyperparameter search over different optimizers and learning rates to avoid introducing bias because of unsuitable training hyperparameters, all models are trained using Stochastic Gradient Descent (SGD) opimtizer with a learning rate of 0.01, momentum of 0.9 and weight decay of 0.0005. Each model is trained for a maximum of 200 epochs with a training batch size of two. Again, to reduce computational load, early stopping with a patience of 10 is applied.

To apply augmentations during training, the implementations from the Albumentation library [2] are used. Following 25 pixel-level augmentations are used in this work: AdvancedBlur, Blur, CLAHE, ColorJitter, Defocus, Downscale, Emboss, Equalize, FancyPCA, GaussNoise, GaussianBlur, GlassBlur, HueSaturationValue, ISONoise, MedianBlur, MotionBlur, RGBShift, RandomBrightnessContr, RandomGamma, RandomToneCurve, RingingOvershoot, Sharpen, ToSepia, UnsharpMask and ZoomBlur. Each augmentation is applied with its standard parameters which include a probability of application of 50% per image. For more in-depth information about the augmentations, the reader is referred to the online documentation of the library.

Overall, all 13 model variations are trained 3 times for each of the 25 augmentations as well as the baseline resulting in 1,014 training runs.

4. Evaluation

4.1. Using Augmentations in Isolation

4.1.1. Semantic Segmentation

Figs. 1 and 2 show the effect on the real-world performance when applying the considered augmentations on the synthetic training images for the semantic segmentation datasets. The exact numerical values are given in Tab. 2. It can be seen that there are augmentations that lead to significant improvements on the sim-to-real generalizability of

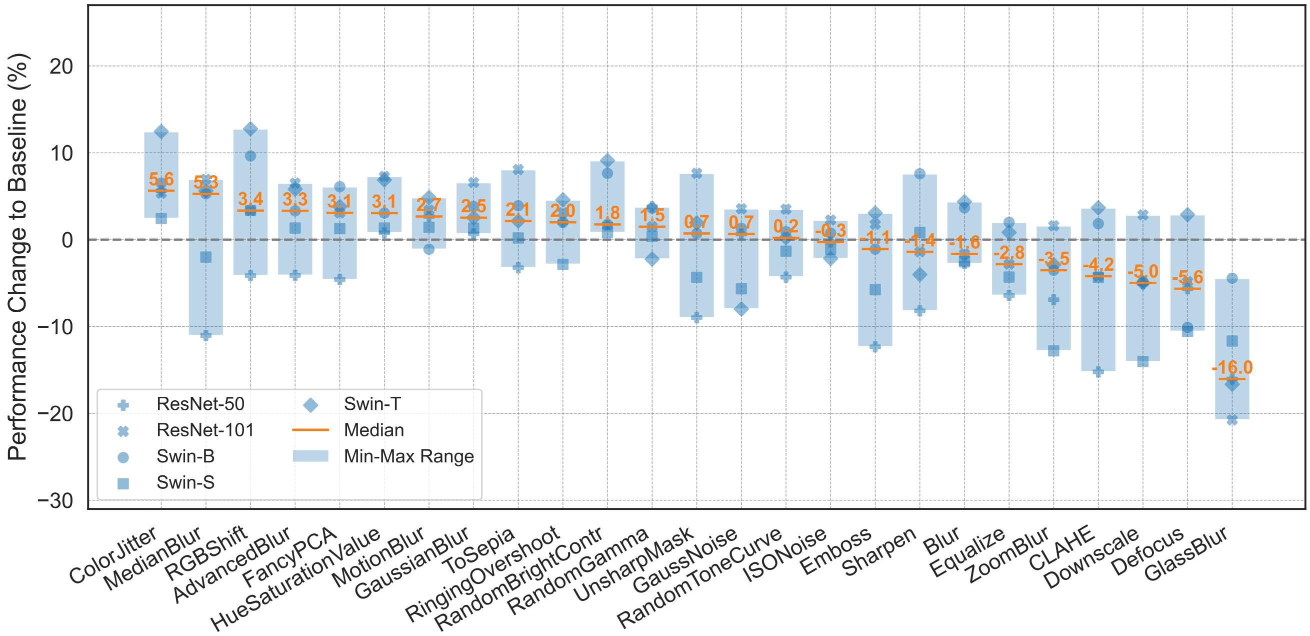

Figure 2. Change of performance d on the real-world KITTI evaluation datasets when applying a specific image augmentation during training compared to the baseline model.

the DL models. Applying just one specific augmentation on the synthetic training data improves the median performance of the models on the real-world images by over 5%. For some models, applying just one augmentation can even lead to a maximal improvement of 12.3% on Ruralscapes and 12.8% on KITTI.

There are four augmentations that improve the median performance of the trained models compared to the baseline on both datasets. These are ColorJitter, RGBShift, HueSaturationValue and ToSepia. Furthermore, it is interesting to note that while only four out of the 25 augmentations improve the median performance of the models on both datasets, the augmentations that improve the median performance the most, are similar: The top-3 augmentations on both semantic segmentation datasets contain ColorJitter and RGBShift. Both augmentations vary the colors of the synthetic images. This is an interesting finding that underlines the conclusion from [18] attributing the sim-to-real generalization problems largely to the general coloration of the synthetic data. It also aligns with [35] that experimentally shows that textures of synthetic environments are not looking completely realistic.

In general, it is very interesting to note that the transformer architectures all improve their performance when ColorJitter or RGBShift are applied, as seen in Tab. 2. Although current literature states that transformer architectures pay more attention to shape than to texture [23, 24], the color differences do seem to have a significant influence on the sim-to-real generalization problems for transformers.

There are also some augmentations that do not give any improvement on the semantic segmentation datasets but in contrary deteriorate the median performance of the DL models on both datasets. These are ISONoise, Emboss, Sharpen, Blur, Downscale, Defocus, and GlassBlur. Again, the augmentations that deteriorate the results the most are the same on both datasets, namely Defocus and GlassBlur. Furthermore, Downscale is in the bottom-4 on both datasets. This highlights that augmentations may improve the generalizability of the models but that the augmentations also have to be reasonable and reflect the phenomenons faced during real-world deployment. As is explained below, some blur augmentations help on the KITTI dataset but the real-world dataset does not contain any glass through which objects have to be detected. Furthermore, there are no images out-of-focus and all images have the same resolution. Therefore, these augmentations do not give the model useful information to learn.

There are also some interesting differences in the influence of the augmentations on the two semantic segmentation datasets. While some augmentations have a positive or negative effect on both datasets, there are also augmentations that have a different influence on the datasets. On the KITTI dataset, 15 of the 25 investigated augmentations help to improve the median of the performance of the models compared to the baseline. On the Ruralscapes dataset, only seven augmentations lead to an improvement.

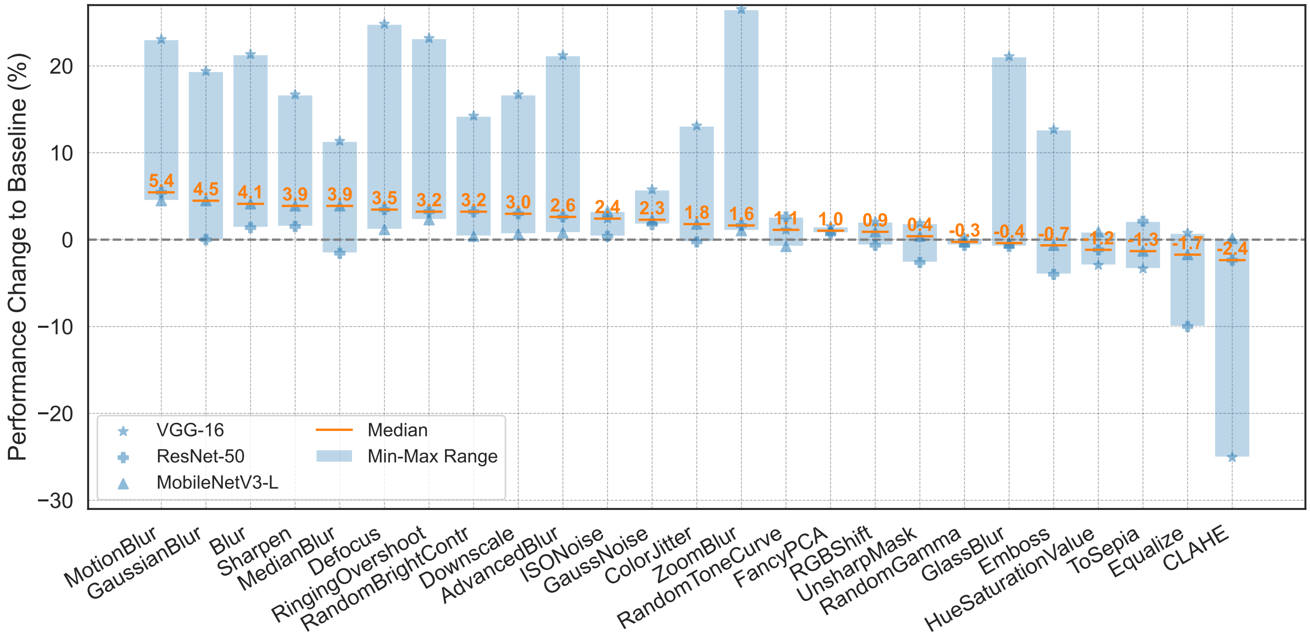

Figure 3. Change of performance d on the real-world drogue detection evaluation datasets when applying a specific image augmentation during training compared to the baseline model.

This difference mainly comes from blur augmentations. While KITTI improves with many of the blur augmentations, namely Median-, Advanced-, Motion- and Gaussian-Blur, Ruralscapes does not. Ruralscapes only improves with ZoomBlur, with which the models on the KITTI dataset do not improve. It is assumed that this difference comes from the characteristics of the datasets. As in both datasets the images are taken from a moving vehicle, it could be expected that both datasets would improve with blur augmentations. However, differences in the real-world camera settings may lead to differences in the blur of the images in both datasets, varying the effect of the augmentations.

4.1.2. Object Detection

For the object detection use-case, the augmentations show many improvements again. The median performance of the models improves on even more augmentations, namely on 18 of the 25. The size of the improvements is similar to the semantic segmentation models, except for the VGG-16 models which show improvements of more than 20% compared to the baseline multiple times. This will be considered in a later section. Furthermore, the number of augmentations that improve all models and not just the median is also much higher with 12 compared to 4 and 0 for the semantic segmentation datasets. The reasons for the higher number of augmentation that lead to an improvement could be based on the visually simple object detection dataset. As the dataset of the scenario is relatively simple and does not contain much variation, the diversity introduced by the various forms of augmentations may have a much greater impact than for the semantic segmentation datasets which are much more diverse and complex by themselves.

Similar to both semantic segmentation datasets, ColorJitter and RGBShift improve the median performance of the DL models compared to the baseline. As discussed for semantic segmentation above, this aligns with the literature stating color differences are a major factor for the sim-to-real generalization problems. However, the improvements for these color augmentations are much smaller on the object detection dataset. Also, contrary to the semantic segmentation datasets, HueSaturationValue does not give any improvements. A possible explanation is again based on the visual simplicity of the object detection dataset. As the images do not contain as much colors but mostly a blue background and gray objects, the influence of potential color differences seems to be much smaller. Nevertheless, there is a measurable influence highlighting again the problem of colors in synthetic images.

While the usage of blur for the semantic segmentation datasets was ambiguous, the object detection models improve much when using blur augmentations. The biggest improvements are achieved when using Motion Blur. This is reasonable as the drogue moves a lot in the depicted situation. On median, the models also improve using GaussianBlur, Blur, MedianBlur, AdvancedBlur and ZoomBlur. As the synthetic drogue detection dataset does not contain blurred images because of the applied data generation

Table 2. Statistical values on the change of performance of each model compared to the baseline in percent. The improvement row counts the number of augmentations with which the model improves. In the header, R abbreviates ResNet, S abbreviates Swin, MN abbreviates MobileNetV3-L and V abbreviates VGG.

	Ruralscapes					KITTI					Object Detection		
	R-101	R-50	S-B	S-S	S-T	R-101	R-50	S-B	S-S	S-T	MN	R-50	V-16
Minimum	-9.6	-30.1	-16.6	-11.0	-6.4	-20.7	-16.0	-10.1	-14.0	-16.7	-1.7	-10.0	-25.1
Maximum	12.3	4.2	4.3	5.4	12.0	8.1	5.6	9.6	3.4	12.8	4.5	5.4	26.5
Median	4.3	-7.7	-3.8	-2.7	3.0	3.1	-4.1	2.0	-1.3	2.9	0.9	0.8	12.7
Mean	2.8	-9.0	-4.1	-2.5	3.0	1.8	-4.1	1.9	-2.7	1.9	1.3	0.4	10.0
Improvement	16.0	2.0	5.0	5.0	16.0	19.0	8.0	19.0	11.0	18.0	19.0	14.0	22.0

process, these augmentations seem to align the synthetic dataset more closely with the real-world images. Especially, since the real-world images sometimes even look blurry to the human eye.

4.1.3. Differences Between the Deep Learning Models

From model perspective, the semantic segmentation model with the ResNet-101 backbone is the model that improves with the highest number of augmentations across both semantic segmentation datasets as shown in Tab. 2. Contrary, the semantic segmentation model with the ResNet-50 backbone improves with the lowest number of augmentations across both datasets. It also has the lowest median and mean improvement of all considered semantic segmentation models. Interestingly, this does not seem to be related to the baseline performance of the model. It would be reasonable to assume that models with a comparably low baseline performance have much more room to improve and vice versa. However, on the Ruralscapes dataset, the models with the ResNet-50 and ResNet-101 backbone have the best and second best baseline, respectively. A more probable explanation is that the ResNet-50 backbone with its lower number of trainable weights is already close to its capacity boarder and, therefore, the information added by the augmentations cannot be learned by the given number of weights. The ResNet-101, on the other hand, has more trainable weights and therefore might have more capacity left to learn the diversity added by the augmentations. Therefore, augmentations seem to have more benefits for models with a high capacity.

A similar phenomenon can be observed for the object detection models. The MobileNetV3-Large has by far the fewest trainable weights. It also has by far the smallest maximal deterioration and the smallest maximal improvement. Because of the resulting smaller capacity, the model seems to learn relatively robust weights by itself, preventing overfitting and instead allowing relatively good generalization. However, on the downside, the model cannot achieve much performance improvement using image augmentations.

As mentioned above, the object detection model with the VGG-16 backbone has by far the highest improvement as well as the highest number of augmentations with which it improves on the real-world evaluation data as shown in Tab. 2. As the baseline performance of the model is relatively low, it can be concluded that the model does not seem to be able to generalize well only from the synthetic images without augmentations. The architecture is the oldest one considered in this work and was improved, *e.g.* by ResNets, with the goal to improve training and overall performance. This directly translates to the sim-to-real phenomenon as it seems to need more diversity, as added from the augmentations, to be able to generalize well to real-world images. However, not all augmentations help, as it also experiences by far the most deterioration of -25.1% with certain augmentations.

Table 3. Performance improvements against the baseline for the selected combinations of augmentations in percent. CJ denotes ColorJitter, RGBS denotes RGBShift, SemS denotes the augmentations that improve the models on the semantic segmentation datasets, DS denotes Dataset Specific, and MS denotes Model Specific.

Augmentation Dataset	All	CJ+RGBS	SemS	DS	MS
Ruralscapes	-17.9	5.5	8.9	2.6	6.0
KITTI	-20.6	2.4	3.4	5.0	-1.0
Drogue	-3.8	3.3	—	-4.9	-5.3

4.2. Using Selected Combinations of Augmentations

In Sec. 4.1, it was shown that there are several augmentations that improve the model performance compared to the baseline when applied in isolation. This section presents first investigations on whether combining augmentations can improve the results further. This research is reasonable as the brute-force approach of combining all possible augmentations is not useful but deteriorates the results, as shown in Tab. 3. The median of the models deteriorates on all considered datasets. Even on the object detection dataset, on which most augmentations have a positive influence, the combination of all leads to a deterioration. As the number of potential combinations grows exponentially with the number of augmentations, only four promising combinations are evaluated in this work. The results are shown in Tab. 3.

The first considered combination aggregates the augmentations that lead to an improvement of the median performance of the models on all three datasets, namely ColorJitter and RGBShift. This combination leads to slight improvements on some datasets but does not necessarily improve the performance compared to applying only one of the augmentations. On the Ruralscapes dataset, this combination leads to slightly better results then the best augmentation in isolation. For object detection, the improvement is larger than both augmentations in isolation. While the combination still leads to an improvement on the KITTI dataset, it is worse than both augmentations in isolation.

The second considered selection combines all augmentations that lead to an improvement on both semantic segmentation datasets, namely ColorJitter, RGBShift, HueSaturationValue and ToSepia. Because of that, it is only evaluated on the semantic segmentation datasets. When combining these augmentations, it improves the median performance on both semantic segmentation datasets. While the median improvement using this combination is still smaller than some of the isolated augmentations on the KITTI dataset, it outperforms the previous best combination on the Ruralscapes dataset by more than 3%.

The third considered combination is dataset-specific and contains all augmentations that lead to a median improvement on each dataset in isolation. Therefore, for each dataset, it contains the augmentations for which the median improvement in Figs. 1 to 3 is greater than zero. The forth combination is model-specific and contains all augmentations that lead to an improvement of that model on the dataset. The results show that there is potential to be found. However, combining multiple augmentations seems to be a tricky task. While on the Ruralscapes dataset, the model-specific variant sees a median improvement larger than that of any augmentation in isolation, all the other models do not see improvements compared to augmentations in isolation. For half of the variants, the median of the model performance even deteriorates. Overall, this underscores that using augmentations is a promising approach to improve the sim-to-real generalization but combining multiple augmentations is a tricky task that does not necessarily lead to improved results.

5. Conclusion, and Future Work

This work investigates the potential of basic pixel-level image augmentations to improve the generalization capabilities of DL perception models from synthetic training to real-world evaluation images. It evaluates the influence of 25 augmentations on five semantic segmentation models on two datasets as well as three object detection models on one object detection dataset. Overall, this work shows that using augmentations is a promising approach to improve the sim-to-real generalization. Even when using only one augmentation, the median real-world performance of the models improves by more than 5% on all considered dataset. Some models even reach maximum improvements of 26.8% on the object detection, 12.8% on the autonomous driving, and 12.3% on the UAS dataset when using only one augmentation during training.

Furthermore, the results of this work underline that differences in coloration seem to have a significant influence on the problem of DL models to generalize from synthetic to real-world images, as RGBShift and ColorJitter are the only augmentations improving the median performance of the models on all three dataset. Furthermore, the performance of all transformer models improves using these two augmentations during training, indicating that color differences do seem to have a significant influence on the sim-to-real generalization problems for these architectures, although current literature states that transformer architectures pay more attention to shape than to texture.

This work further confirms findings from current literature that adding noise and blur may improve the generalization from synthetic to real-world images. However, we show that this depends on the considered dataset and that the added effects have to reflect the conditions that the model will face when deployed. For example, adding glass blur for models that will never see through glass may help to increase the diversity of the training dataset but does not seem to help improve final performance when deployed. While not all considered datasets improve with the addition of blur augmentations, the usage of blur should especially be considered when faced with situations in which the camera or the observed objects move a lot.

While this work shows many positive effects of using augmentations on the sim-to-real generalization, it also shows that not all augmentations improve the model performance on real-world images. Furthermore, combining augmentations without careful consideration may harm the real-world performance and in general combining augmentations in a useful way is a difficult task. Because of that, future work should investigate possible benefits and strategies of combining multiple augmentations and the influence of their parametrization further. As the number of possible combinations grows exponentially, using optimization frameworks or learning optimal augmentation strategies for synthetic images seem to be promising directions.

Overall, while increasing photorealism is still a promising outlook to increase the sim-to-real generalization in the long-run, image augmentations provide a simple-to-use way to improve the generalization when using existing state-of-the-art game-engines today.

References

[1] Nerea Aranjuelo, Sara García, Estíbaliz Loyo, Luis Unzueta, and Oihana Otaegui. Key strategies for synthetic data generation for training intelligent systems based on people detection from omnidirectional cameras. *Computers & Electrical Engineering*, 92:107105, 2021. 2

[2] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020. 1, 4

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 3

[4] Alexandra Carlson, Katherine A. Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. Modeling camera effects to improve visual learning from synthetic data. In *Computer Vision – ECCV 2018 Workshops*, pages 505–520, Cham, 2019. Springer International Publishing. 2

[5] Epic Games, Inc. The most powerful real-time 3d creation tool - unreal engine. accessed 15 March 2025. 3

[6] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 3

[8] Korbinian Hagn and Oliver Grau. Improved sensor model for realistic synthetic data generation. In *Proceedings of the 5th ACM Computer Science in Cars Symposium*, New York, NY, USA, 2021. Association for Computing Machinery. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, 4

[10] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 1

[11] Christoph Hinniger and Joachim Rüter. Synthetic training data for semantic segmentation of the environment from uav perspective. *Aerospace*, 10(7), 2023. 2, 3

[12] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 4

[13] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017. 2

[14] Benjamin Kiefer, David Ott, and Andreas Zell. Leveraging synthetic data in object detection on unmanned aerial vehicles. In *26th International Conference on Pattern Recognition (ICPR)*, 2022. 2

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017. 4

[16] Ivan Krešo, Denis Čaušević, Josip Krapac, and Siniša Šegvić. Convolutional scale invariance for semantic segmentation. In *Pattern Recognition: 38th German Conference, GCPR 2016, Hannover, Germany, September 12-15, 2016, Proceedings 38*, pages 64–75. Springer, 2016. 3

[17] Michael Krump and Peter Stütz. UAV based vehicle detection with synthetic training: Identification of performance factors using image descriptors and machine learning. In *International Conference on Modelling and Simulation for Autonomous Systems (MESAS)*, 2020. 2

[18] Michael Krump and Peter Stütz. UAV based vehicle detection on real and synthetic image pairs: Performance differences and influence analysis of context and simulation parameters. In *International Conference on Modelling and Simulation for Autonomous Systems (MESAS)*, 2022. 2, 5

[19] Michael Krump, Martin Ruß, and Peter Stütz. Deep learning algorithms for vehicle detection on UAV platforms: first investigations on the effects of synthetic training. In *International Conference on Modelling and Simulation for Autonomous Systems (MESAS)*, 2019. 2

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. Corresponding website: https://cocodataset.org/. 3

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 4

[22] Alina Marcu, Vlad Licaret, Dragos Costea, and Marius Leordeanu. Semantics through time: Semi-supervised segmentation of aerial videos with iterative label propagation. *Asian Conference on Computer Vision*, 2020. Corresponding image dataset is available at https://sites.google.com/site/aerialimageunderstanding/semantics-through-time-semi-supervised-segmentation-of-aerial-videos. 3

[23] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34: 23296–23308, 2021. 5

[24] Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022. 5

[25] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3), 2015. 1, 2

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region

proposal networks. *Advances in neural information processing systems (NeurIPS)*, 28, 2015. 3

[27] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[28] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[29] Frank A Ruis, Alma M Liezenga, Friso G Heslinga, Luca Ballan, Thijs A Eker, Richard JM den Hollander, Martin C van Leeuwen, Judith Dijk, and Wyke Huizinga. Improving object detector training on synthetic data by starting with a strong baseline methodology. In *Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II*, pages 333–345. SPIE, 2024. 2

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115, 2015. 4

[31] Joachim Rüter, Theresa Maienschein, Sebastian Schirmer, Simon Schopferer, and Christoph Torens. Filling the gaps: Using synthetic low-altitude aerial images to increase operational design domain coverage. *Sensors*, 24(4), 2024. 2

[32] Joachim Rüter and Rebecca Schmidt. Using only synthetic images to train a drogue detector for aerial refueling. In *International Conference on Modelling and Simulation for Autonomous Systems (MESAS)*, 2023. 2, 3

[33] Joachim Rüter, Umut Durak, and Johann C. Dauer. Investigating the sim-to-real generalizability of deep learning object detection models. *Journal of Imaging*, 10(10), 2024. 1, 2, 3, 4

[34] Kmeid Saad and Stefan-Alexander Schneider. Camera vignetting model and its effects on deep neural networks for object detection. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1–5, 2019. 2

[35] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2018. 5

[36] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1), 2019. 1

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[38] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312, 2018. 1, 2

[39] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint arXiv:1810.08705*, 2018. 2

[40] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 4

[41] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 4