Supplementary Material

Time to Shine: Fine-Tuning Object Detection Models with Synthetic Adverse Weather Images

The supplementary material adds additionals information in order to better understand and reproduce the results of our work.

1 Additional Image Corruption Details

The image corruption models utilized in our study were based on the implementation described in [5]. Some modifications were made to adapt the snowflakes layer by introducing an alpha channel, allowing control over the layer's opacity. Additionally, we extended the corruption library by incorporating a rain-streaks model derived from the snowflakes model. This involved increasing the motion blur to achieve a rain-streak-like appearance. The fog model from the corruption library remained unaltered. Details of the parameter settings for each model can be found in Tables 1, 2, and 3, respectively. The source code for the image corruption library is available on GitHub: `https://github.com/bethgelab/imagecorruptions`.

The raindrop model implementation was based on [7]. However, the original model was memory-intensive and did not fit within the available GPU memory (24GB) for an image size of 1024x1024 pixels. To address this, we made minor modifications to the code to optimize memory usage and ensure compatibility with the object detection model being evaluated. The parameterization details for the raindrop model can be found in Table 4. The source code for the raindrop model is available on GitHub: `https://github.com/astra-vision/GuidedDisent`.

The outcomes of the applied weather image overlays are presented in Figure 1, showcasing different levels of intensity.

It should be noted that in all our evaluations involving the synthetic fog overlay, it was applied to every frame rather than with a probability of 0.25 as mentioned in the paper. This was inadvertently overlooked in the main paper.

Intensity	**Center**	**Standard Deviation**	**Zoom**	**Threshold**	**Radius**	**Sigma**	**Alpha**
1	0.20	0.45	1.5	0.85	6	2	0.9
2	0.20	0.40	1.4	0.70	6	2	0.9
3	0.20	0.50	1.7	0.80	6	2	0.9
4	0.35	0.60	2.7	1.10	6	2	0.9
5	0.35	0.65	3.2	1.10	6	2	0.9

Table 1: Parameters used for the snowflake model in five different intensities.

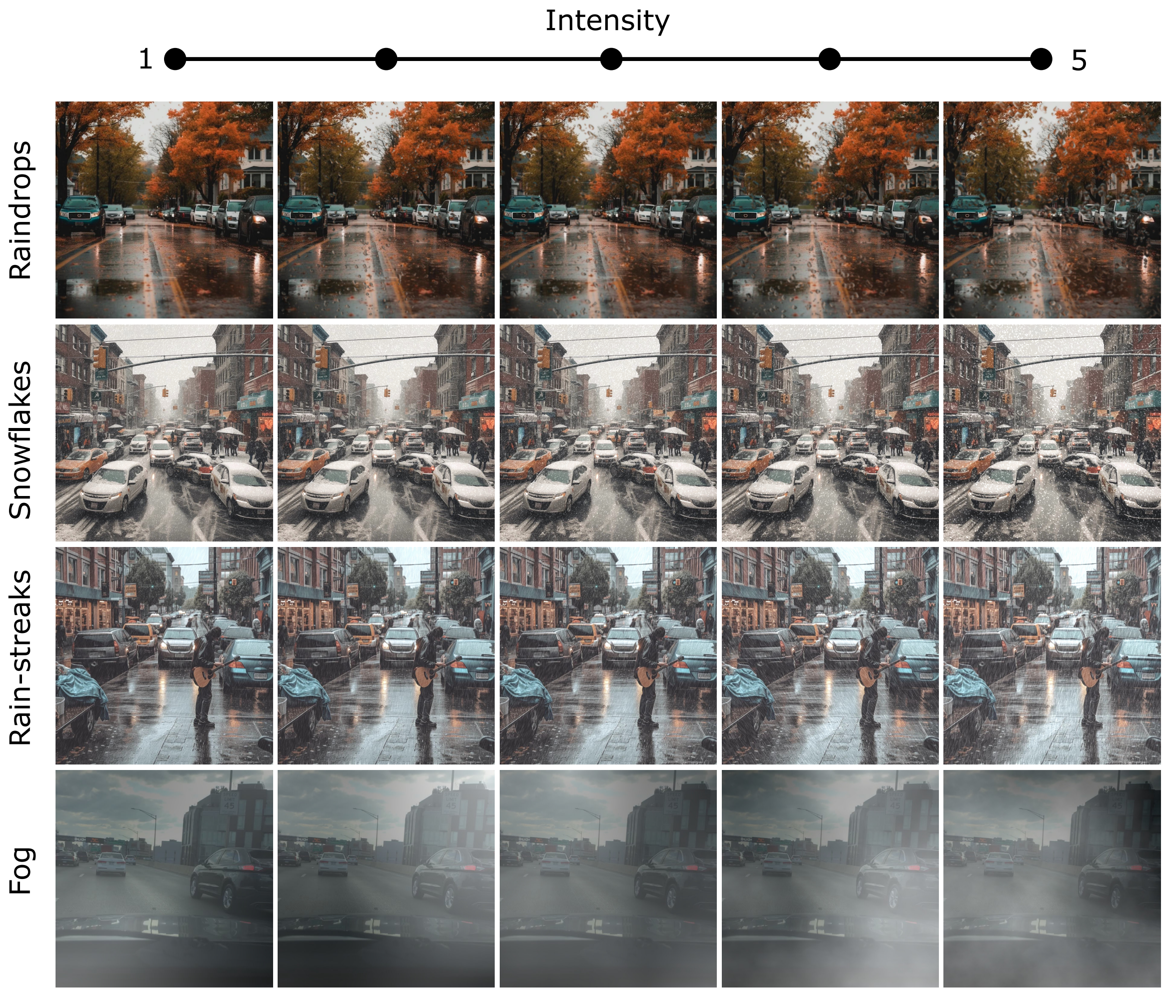

Figure 1: Examples of image corruptions applied to images in different intensities from 1 to 5.

Intensity	Center	Standard Deviation	Zoom	Threshold	Radius	Sigma	Alpha
1	0.1	0.5	1.5	0.8	10	14	0.9
2	0.2	0.5	1.5	0.8	10	14	0.9
3	0.2	0.55	1.5	0.8	10	14	0.9
4	0.3	0.55	1.5	0.8	10	14	0.9
5	0.4	0.55	1.5	0.8	10	14	0.9

Table 2: Parameters used for the rain-streak model in five different intensities.

Intensity	Density	Wibbledecay
1	1.5	2.0
2	2.0	2.0
3	2.5	1.7
4	2.5	1.5
5	3.0	1.4

Table 3: Parameters used for the fog model in five different intensities.

Intensity	Drop Size	Drop Frequency	Drop Shape	Drop Sigma
1	20	3	0.9	3
2	30	5	0.8	3
3	40	6	0.8	4
4	45	7	0.8	4
5	50	8	0.5	4

Table 4: Parameters used for the raindrop model in five different intensities.

2 Additional Evaluation Details

In this section we give some additional information on the evaluation details of the paper.

2.1 Data Variation

For the evaluation of the data variation within the Midjourney dataset we used the LPIPS metric [8] which measures the distance between two images using feature vectors of deep neural networks. A higher value indicates more diversity between two images, whereas a lower value indicates more similarity. In our work we use the LPIPS metric to measure variation and diversity inside datasets similar to [4]. We use the code provided by `https://github.com/richzhang/PerceptualSimilarity`. We provide the LPIPS score of the Midjourney dataset, as well as the scores for each dataset utilised in our benchmark, in Table 5.

2.2 Data Distribution

In our study, we presented the size distribution of cars in the Midjourney dataset categorised as small, medium and large, and established a baseline by comparing these distributions with benchmark datasets. To provide a comprehensive analysis, we present the size distribution of small, medium, and large cars for each benchmark dataset separately in Table 6.

Dataset	LPIPS	Average
Midjourney(fog)	0.731	
Midjourney(rain)	0.807	0.770
Midjourney(snow)	0.773	
Cityscapes	0.601	
ACDC(fog)	0.539	
ACDC(rain)	0.646	0.603
ACDC(snow)	0.615	
NuScenes(clear)	0.651	
NuScenes(rain)	0.564	
Adverse Dataset(fog)	0.590	
Adverse Dataset(rain)	0.704	0.680
Adverse Dataset(snow)	0.744	

Table 5: LPIPS scores for Midjourney dataset and all the analysed benchmark datasets in the paper. The ”Average” column presents the aggregated results from the corresponding sections delineated by dashed lines.

Dataset	Size	Small	Medium	Large	Total	Average(S)	Average(M)	Average(L)
ACDC(fog)	500	725	1004	405	2134	0.340	0.470	0.190
ACDC(rain)	500	566	972	479	2017	0.281	0.482	0.237
ACDC(snow)	500	561	1214	596	2371	0.237	0.512	0.251
BDD100K(clear)	4133	21948	16625	7764	46337	0.474	0.359	0.168
BDD100K(rain)	3301	13050	12919	6916	32885	0.397	0.393	0.210
BDD100K(snow)	3794	15401	14092	7942	37435	0.411	0.376	0.212
Adverse(fog)	199	300	514	255	1069	0.281	0.481	0.239
Adverse(wet)	199	765	859	387	2011	0.380	0.427	0.192
Adverse(snow)	200	319	681	526	1526	0.209	0.446	0.345
NuScenes(clear)	5710	2506	14641	10858	28005	0.089	0.523	0.388
NuScenes(rain)	5422	1916	15199	11015	28130	0.068	0.540	0.392
Cityscapes	3475	6813	13559	10332	30704	0.222	0.442	0.337
		64870	92279	57475	214624	0.282	0.454	0.263

Table 6: Distribution of small, medium and large objects in all benchmark datasets. We also calculate the average number of small, medium and large objects. ”Adverse” in the dataset column is short for Adverse Dataset.

2.3 Extreme Adverse Weather

We conducted a performance evaluation of images enhanced using standard augmentation techniques, where we compared four augmentation strategies: AutoAugment trained on ImageNet [1], RandAugment [2], AugMix [3] and TrivialAugment [6]. Among these, AugMix achieved the highest overall performance, and its scores are presented in the paper. To ensure stability amidst variations caused by random transformations, we computed the average results from 5 separate runs. The results are reported in the paper in Figure 7 next to results of models fine-tuned only on clear weather images and our synthetic adverse weather image dataset.

Figure 2: Images from the labeled Midjourney dataset in snow conditions.

3 Additional Dataset Details

In this section, we present additional example images extracted from both the labeled and unlabeled images from the Midjourney dataset. Specifically, Figure 2, 3, and 4 showcase images representing each weather domain from the labeled Midjourney dataset. Furthermore, to demonstrate the variations generated with identical text prompts, Figures 5 to 10 exhibit different variants of images sourced from the unlabeled portion of the dataset.

Figure 3: Images from the labeled Midjourney dataset in rain conditions.

Figure 4: Images from the labeled Midjourney dataset in fog conditions.

Figure 5: Multiple variations of an image depicting snowy weather conditions, all generated using the same text prompt by Midjourney.

Figure 6: Multiple variations of an image depicting snowy weather conditions, all generated using the same text prompt by Midjourney.

Figure 7: Multiple variations of an image depicting rainy weather conditions, all generated using the same text prompt by Midjourney.

Figure 8: Multiple variations of an image depicting rainy weather conditions, all generated using the same text prompt by Midjourney.

Figure 9: Multiple variations of an image depicting foggy weather conditions, all generated using the same text prompt by Midjourney.

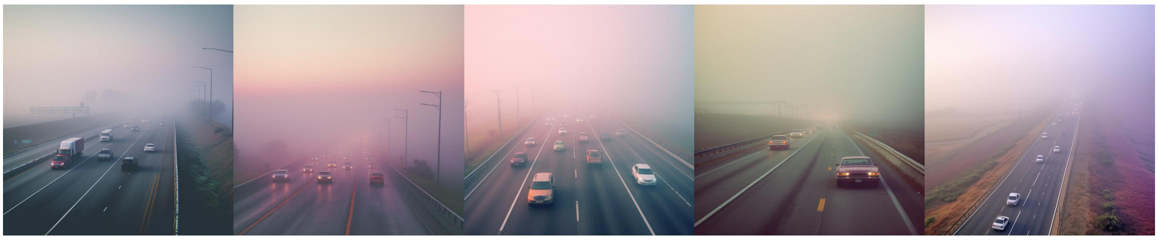

Figure 10: Multiple variations of an image depicting foggy weather conditions, all generated using the same text prompt by Midjourney.

References

[1] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning Augmentation Strategies From Data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019. IEEE.

[2] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

[3] Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *International Conference on Learning Representations*, Sept. 2019.

[4] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal Unsupervised Image-to-Image Translation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11207. Springer International Publishing, Cham, 2018.

[5] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming, Mar. 2020.

[6] Samuel G. Müller and Frank Hutter. TrivialAugment: Tuning-free Yet State-of-the-Art Data Augmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Oct. 2021.

[7] Fabio Pizzati, Pietro Cerri, and Raoul de Charette. Physics-Informed Guided Disentanglement In generative networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[8] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 2018. IEEE.