

Improving autonomous vehicle perception in adverse weather through image augmentation

Ethan Kou¹, Noah Curran²

¹ Henry M. Gunn High School, Palo Alto, California

² Computer Science and Engineering, University of Michigan, Ann Arbor, Michigan

SUMMARY

Robust perception is crucial in autonomous vehicle navigation and localization. Visual processing tasks, like semantic segmentation, should work in varying weather conditions and during different times of day. Semantic segmentation is where each pixel is assigned a class, which is useful for locating overall features. To train a model to perform semantic segmentation, large amounts of data are needed, but the labeling process for segmentation data is especially tedious. Additionally, many large datasets include only images taken in clear weather. This is a problem because training a model exclusively on clear weather data hinders performance in adverse weather conditions like fog or rain. In general, visual processing tasks like segmentation have less accuracy in adverse conditions. We hypothesized that applying image augmentation during training can enhance semantic segmentation performance for autonomous vehicles in adverse weather by decreasing the loss of the model. We trained models with data from Car Learning to Act (CARLA) of both clear images and weather-affected images. Our results show that augmentations significantly improved segmentation in weathered conditions, but models trained on actual weather data still significantly outperformed models trained on augmented data in most conditions. This suggests that while augmentations don't completely solve the domain adaptation problem, they still offer an improvement.

INTRODUCTION

Perception is a crucial task for autonomous vehicles because they allow for accurate driving in a complex environment. Semantic segmentation is one example of a perception task. In semantic segmentation, each pixel in the image is given a class label (1). For example, cars are all given an ID, and roads are given a separate ID (**Figure 1**). Using this segmented image, autonomous vehicles can locate important landmarks to perform autonomous actions, making segmentation an important task. Semantic segmentation models often perform worse in adverse weather conditions due to training on only clear-sky daytime images. This is a weakness because we want autonomous vehicles to have robust perception in all weather. We addressed this problem by exploring the impact of image augmentations on model performance in simulated adverse conditions.

To train a robust segmentation model, large amounts of images and labels need to be collected. Labels are especially tedious to create for image segmentation because each pix-

el's class needs to be hand-labeled. Furthermore, most large datasets, such as CityScapes, are composed of images collected in clear weather conditions (2). This results in lower segmentation accuracy when the model is tested on images in adverse weather conditions. For example, rain can obstruct parts of the image, and fog can cause blurring.

In the past few years, there have been many approaches to the challenge of domain adaptation for perception. Domain adaptation refers to adapting a model trained on one data distribution (such as clear weather) to other distributions (such as adverse weather). A deep learning based framework for robust traffic sign detection (DFR-TSD) improved traffic sign detection by training models to reverse weather effects on images (3). Although detection was improved for traffic signs, this task is very specialized and doesn't apply to perception in general. Another paper introduced domain generalized semantic segmentation (DGSS), which generates adversarial styles using style mining for image segmentation (4). DGSS mainly tackles the issue for night-time domain adaptation, not for adverse weather in general. Previous research has also introduced tests for the robustness of models by applying various image transformations and evaluating models on the transformed images (5). Although all these papers relate to domain adaptation, none discuss autonomous vehicle semantic segmentation domain adaptation across both time and weather when only given clear weather data. Our goal was to introduce a way to improve detection in adverse weather when only given clear sky data, and we hypothesized that augmenting clear weather data in training improves semantic segmentation in adverse weather conditions. We predicted the augmented data can train the model to extract features invariant to the domain. After training and evaluating semantic segmentation models with and without augmentations, we concluded that augmentations improve perception in adverse weather conditions.

RESULTS

Our goal was to introduce a way to improve perception in adverse weather when only given clear sky data. We hypothesized that augmenting clear weather data in training improves semantic segmentation in adverse weather conditions. We trained three semantic segmentation models with the UNet architecture: M1, M2, and M3. All data was collected in Car Learning to Act (CARLA), an autonomous vehicle simulator. The first model, M1, was trained on clear weather images. M1 is the control. M2 was trained on augmented clear weather images, and M3 was trained on weather images (**Figure 2**). We measured training and validation loss and accuracy for all models (**Tables 1 and 2**). The loss value measures how much the prediction differs from the actual, so a lower loss

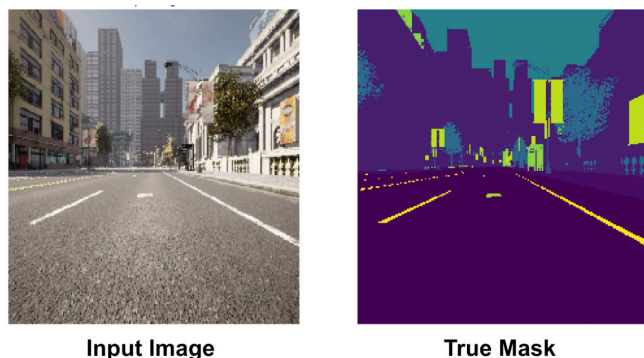


Figure 1: Semantic segmentation example. A) RGB (red green blue) and B) semantic segmentation mask, taken from our dataset.

means a model is more accurate. To acquire the loss values, we ran the models on seven test sets which each have different weather types and times of day, such as daytime clear or nighttime rainy.

With augmentation, the accuracy for random weather conditions improved from 0.8600 to 0.8822, and the loss decreased from 0.5045 to 0.4369 (**Tables 1 and 2**). Models trained on augmented data had significantly lower loss than models trained on clear data when tested on rainy night data and random weather night data ($p < 0.001$, 2-sample *t*-test, **Table 3**).

There is also less “noise” in the augmented model prediction than in the clear weather prediction (**Figure 3**). Here, “noise” is a qualitative observation that refers to how the clear weather prediction tends to have more sporadic inaccurate blotches than the augmentation model prediction. The loss of the weather data models is significantly lower than that of the augmented data models on all seven test sets except DC ($p < 0.001$, 2-sample *t*-test, **Table 3**).

DISCUSSION

Our central hypothesis was that augmenting clear weather data in training improves semantic segmentation in adverse weather conditions. After training and evaluating semantic segmentation models with and without augmentations we concluded that when only training on clear-day images, applying image augmentation improves model performance during adverse weather conditions. Our results show that applying augmentations decreased the loss when testing on random weather conditions. This indicates that image augmentation may generally improve model performance. There is also statistically significant evidence that augmentations decrease test loss in nighttime rain or nighttime random weather conditions. The randomness and diversity of augmentations could explain this trend, allowing the model to adapt better to adverse lighting conditions during the night.

Data augmentation is a commonly used technique to increase the volume of image data. The effectiveness of weather augmentations suggests that data augmentation, especially for autonomous vehicle perception, should include weather-based augmentations like the techniques tested in this paper. Despite the positive results, there are many limitations and possible improvements. Since the loss of the weather data models is significantly lower than the augmented data models on several test sets, there is room for improvement in

Model	DC	DR	DW	NC	NR	NW	W
M1	0.1423	0.525	0.2171	1.3228	1.4114	1.0673	0.5045
M2	0.1535	0.4368	0.2252	0.991	1.2082	0.846	0.4369
M3	0.1919	0.2588	0.195	0.4341	0.5546	0.4455	0.2846

Table 1: Loss of models in various test sets. The dataset name follows the convention of {D/N}{C/R/W}, where D = day, N = night, C = clear, R = rainy, and W = random weather and time.

Model	DC	DR	DW	NC	NR	NW	W
M1	0.9588	0.8606	0.9381	0.6332	0.6338	0.7145	0.8600
M2	0.955	0.8831	0.937	0.715	0.7046	0.7754	0.8822
M3	0.9455	0.9279	0.9435	0.8844	0.8601	0.8829	0.9231

Table 2: Accuracy of models in various test sets. The dataset name follows the convention of {D/N}{C/R/W}, where D = day, N = night, C = clear, R = rainy, and W = random weather and time.

the domain adaptation approach. The optimal domain adaptation approach would meet or exceed the performance of a model trained on weather data when only given ground truth clear data.

Our approach shows that augmentations during training allow for domain adaptation agnostic of the type of model. Because of this, our models’ ability to perform semantic segmentation is less powerful than state-of-the-art models because of our technique. A variety of state-of-the-art domain adaptation models and their performance are detailed in the DGSS paper (4). Future work would be to train a state-of-the-art model using our augmentation approach. Also, custom augmentations outside of the ones we used should be evaluated (6). The augmentations we used include random rain, random fog, and some others (described in Materials and Methods). Although our augmentations apply weather variation to the image, they are weather-specific augmentations, which is a drawback. As future work, the augmentations should be non-weather-specific yet still introduce realistic weather variation. For example, it’s possible that a combination of blur, noise, and other image filters could span the possible image transformations caused by weather. This could improve the detection in a wider range of weathers instead of the specific weathers that the augmentations tailor.

A simulation limitation is that CARLA, the autonomous vehicle simulation we used, does not include weather options such as snow (7). Future work would use a dataset of more diverse weather conditions. This can be done in simulation, but simulations still have drawbacks. For our data collection in CARLA, the car always spawns perfectly aligned on the road, and many factors of random noise are reduced. This level of perfection is absent in real life, so extrapolating our approach to real-life datasets is an important next step.

The significance testing method could also be improved. Since we used 10-fold cross-validation for training, some models shared data, making the loss of the models not completely independent from each other. This can cause problems with the accuracy of the significance test because the *t*-test assumes that all samples are independent. Future work would ensure every sample is independent. This can be done by acquiring a dataset large enough so that the 10 models that are trained have no shared data.

This study serves as a stepping point for the end goal of training a model to effectively perform segmentation in real-

	DC	DR	DW	NC	NR	NW	W
Clear Mean	0.1878	0.5270	0.2522	0.9467	1.2077	0.8832	0.4600
Clear Stdev	0.0064	0.0811	0.0131	0.0877	0.2240	0.1002	0.0403
Augmented Mean	0.2337	0.4752	0.3052	0.9393	0.8710	0.7312	0.4299
Augmented Stdev	0.0085	0.0863	0.0098	0.1131	0.1035	0.0530	0.0228
Weather Mean	0.2319	0.2882	0.2396	0.4222	0.5112	0.4318	0.3008
Weather Stdev	0.0076	0.0080	0.0068	0.0119	0.0149	0.0165	0.0078
Augmented lower loss than clear, p-value	1.00000	0.09175	1.00000	0.43614	0.00021	0.00025	0.02742
Weather lower loss than augmented, p-value	0.31303	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

Table 3: Loss and single-tailed 2-sample t-test p-values of 10-fold models in various test sets. Mean and standard deviation of the loss of the 10-fold models for the clear, augmented, and weather datasets. For each test set, the lowest loss is **bolded** and the second lowest loss is underlined. The p-value for the augmented dataset models having lower loss than the clear dataset models and the p-value for the weather dataset models having lower loss than the augmented dataset models are written. All p-values below 0.001 are also bolded. The dataset name follows the convention of {D/N} {C/R/W}, where D = day, N = night, C = clear, R = rainy, and W = random weather and time.

life adverse weather when only given data in clear day weather. There are still numerous ways, as described above, to improve upon our approach. Augmentations should be further researched because they are a promising technique clearly applicable to autonomous vehicle perception and most likely applicable to perception in general.

MATERIALS AND METHODS

Data collection

We collected data using CARLA, a widely used 3D autonomous vehicle simulator. CARLA has numerous towns, or different locations, each with its own unique architecture and style. Some towns are urban cities with skyscrapers, while

others are rural farmlands with barns and crops. This allows us to simulate the vehicle in a variety of environments.

CARLA sensors can be used to capture a scene in RGB and semantic segmentation. We attach both cameras to the front of the vehicle, so all images were captured on the road. The semantic segmentation is divided into 29 different classes, which are comprehensively detailed in the CARLA documentation. We collected 2 training datasets of 1200 images each. The D1 dataset includes only images taken in clear weather using the weather ClearNoon preset. The D2 dataset includes randomly generated times of day and weather from the CARLA presets.

To generate each dataset, we first set the world to Town 1 and teleported the vehicle to a random location on the road. Then, we adjusted the weather according to the desired dataset and then recorded the current snapshot from the RGB and semantic segmentation camera. We repeated this until 150 data points were recorded for Town 1. Then, we repeated everything for Towns 2-7 and 10. In total, each dataset had 1200 images because 150 data points were recorded for each of the 8 towns.

We generated seven test sets of 400 images, each under various weather conditions. We followed the same data collection procedure as the training set but only collected 50 images per town, for a total of 400 images per training set. The seven datasets were collected on clear days (DC), rainy days (DR), random weather days (DW), clear nights (NC), rainy nights (NR), random weather nights (NW), and random everything (W).

Source code for this project can be found at: <https://github.com/BubblyBingBong/AugmentationSegmentation>

Image augmentation

The goal of image augmentations is to teach the model to extract features invariant to the domain. To do this, we applied a series of weather augmentations to each image. We used the Albumentations library to make applying various types of

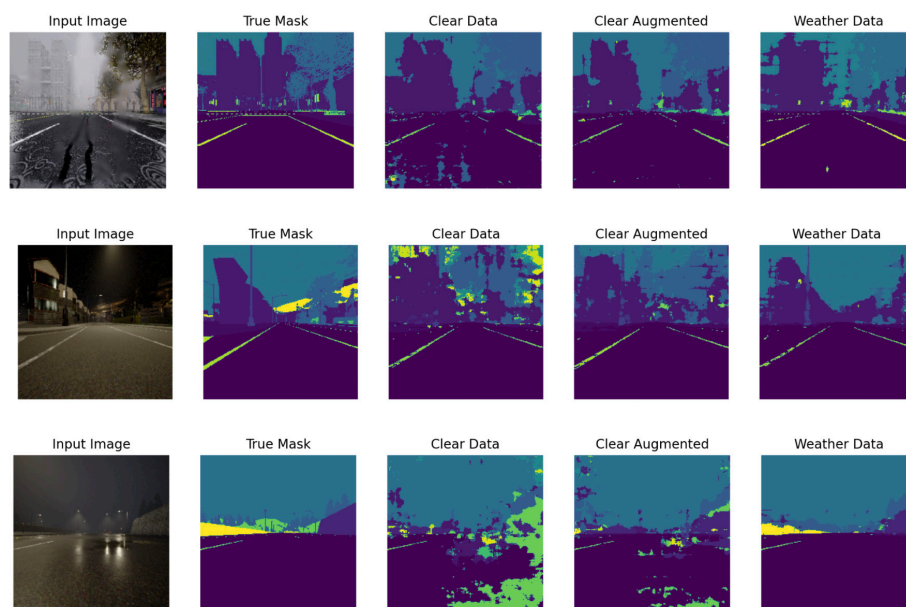


Figure 3: Example predictions on three images. A) M1 (Clear Data), B) M2 (Clear Augmented), and C) M3 (Weather Data) were tested on various images in our dataset.



Figure 4: Augmentation examples. Compared to the **A)** original image, **B)** fog adds random blurring, **C)** RGB shift randomly shifts all pixel values, **D)** random rain adds rain streaks, **E)** random sun flare adds bright spots, and **F)** random gamma adjusts overall image brightness.

image augmentations easier (6). We used the Albumentations library to generate random rain, random sun flare, random fog, random RGB shift, and random gamma. Each of the five augmentations had a 0.5 probability of being applied to an image. When these augmentations were compounded, they produced a diverse range of images that were then inputted into the model for training (Figure 4).

Overall model training and evaluation

We first trained three models. The first model M1 is trained on D1, which has all clear weather images. The second model M2 is trained on D1 after applying image augmentations. The third model M3 is trained on D2, which has images in random weather and during different times of day. We used TensorFlow's SparseCategoricalCrossentropy loss as the training metric because there are multiple label classes and the labels in our data are integers, not one-hot encoded (8).

We utilized a UNet architecture with a MobileNetV2 backbone pre-trained on ImageNet (9). We chose UNet because it's one of the most efficient model architectures for semantic segmentation. The UNet is an encoder-decoder convolutional neural network. The encoder (contracting path) detects all complexities of image features using convolutional and max pooling layers, and the decoder (expansive path) upscales these detected features to segmentation masks using convolutional and unpooling layers. There are skip connections between layers of the encoder and decoder to allow for reliable segmentation across both high-level features, like the location of an edge, and low-level features, like the class of an object. The input images are resized to 224x224 because that's the input size for MobileNetV2. For all models, we trained for 50 epochs because that minimized overfitting. We randomly selected 1,000 train samples and 200 validation samples. A 1,000-200 split ensures there's enough training data while providing enough validation data to accurately determine when overfitting starts.

Significance testing

To determine if image augmentations during training create a statistically significant decrease in the loss value, we trained 10 models each on D1, augmented D1, and D2 for 30 models total. We did not directly compare the losses for M1, M2, and M3 to determine if augmentations decrease loss because random variation could skew results. Using a significance test allows us to see if differences in loss are due to true model performance or randomness.

We used 10-fold cross-validation for each dataset because this makes sure each model has enough training data. For example, the first model was trained on all 1200 images except the first 120. The second model was trained on all 1200 images except the second 120, and so on. Each model was trained for 20 epochs because that number reduces total training time and is still high enough to prevent underfitting. We evaluated the loss of each of the 20 models on all 7 test sets.

For every test set, we had 10 loss values for each of the 3 datasets. To determine the p -value, we ran a single-tailed 2-sample t -test on corresponding sets of 10 loss values for each test set.

Received: July 15, 2024

Accepted: December 5, 2024

Published: October 23, 2026

REFERENCES

1. Isola, Phillip, et al. „Image-to-Image Translation with Conditional Adversarial Networks.“ *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul 2017. <https://doi.org/10.1109/CVPR.2017.632>.
2. Cordts, Marius, et al. „The cityscapes dataset for semantic urban scene understanding.“ *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2016. <https://doi.org/10.1109/CVPR.2016.350>.
3. Ahmed, Sabbir, et al. „DFR-TSD: A deep learning based framework for robust traffic sign detection under challen-

- ging weather conditions." *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, Feb 2021, pp. 5150-5162. <https://doi.org/10.1109/TITS.2020.3048878>.
4. Shyam, Pranjay, et al. "Dgss: Domain generalized semantic segmentation using iterative style mining and latent representation alignment." *arXiv*, 26 Feb 2022. <https://doi.org/10.48550/arXiv.2202.13144>.
 5. Pezzementi, Zachary, et al. "Putting image manipulations in context: Robustness testing for safe perception." *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics*, Sep 2018. <https://doi.org/10.1109/ssrr.2018.8468619>.
 6. Buslaev, Alexander, et al. «Albumentations: Fast and flexible image augmentations.» *Information*, vol. 11, no. 2, 24 Feb 2020. <https://doi.org/10.3390/info11020125>.
 7. Dosovitskiy, Alexey, et al. "CARLA: An open urban driving simulator." *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.
 8. "Tf.Keras.Losses.SparseCategoricalCrossentropy: TensorFlow V2.16.1." TensorFlow.
 9. Ronneberger, Olaf, et al. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *Medical Image Computing and Computer-Assisted Intervention – 2015*, vol. 9351, 18 Nov 2015, pp. 234-241. https://doi.org/10.1007/978-3-319-24574-4_28.

Copyright: © 2026 Kou and Curran. All JEI articles are distributed under the Creative Commons Attribution Noncommercial No Derivatives 4.0 International License. This means that you are free to share, copy, redistribute, remix, transform, or build upon the material for any purpose, provided that you credit the original author and source, include a link to the license, indicate any changes that were made, and make no representation that JEI or the original author(s) endorse you or your use of the work. The full details of the license are available at <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.en>.