Enhancing Autonomous Vehicle Perception in Adverse Weather

Saeah Go, Max Rego, Artem Suprun, Filipp Suprun

Portland State University
CS 410/510 Visual Computing
6/6/2025

1. Introduction:

Autonomous vehicles have made significant advancements in recent years, yet their performance remains limited to ideal conditions—clear, dry roads with good visibility. In reality, environments like Portland often experience rain, fog, and occasional snow, all of which can degrade the reliability and safety of these systems. Current models struggle when exposed to visual obstructions caused by adverse weather, making robust perception a critical challenge in the development of truly reliable autonomous driving.

This project aims to address this limitation by experimenting with obscured visual data, simulating real-world conditions that autonomous vehicles are likely to encounter. Using the GTSRB dataset, we apply two different methods to obscure traffic sign images in order to mimic the visual challenges posed by environmental factors. The first method uses custom image transformation functions to simulate effects such as rain, snow, and fog. The second method leverages a generative AI technique—CycleGAN—for style transfer, transforming the original images using weather-specific domain datasets (Snow100K, Rain100, and Foggy Cityscapes) to produce realistic weather effects. By comparing the performance of models trained with each type of obscured data, we aim to identify effective strategies for maintaining high accuracy despite visual degradation, contributing to safer and more adaptable autonomous vehicle systems.

Developing a model that can help with accurately predicting traffic signs during heavy weather conditions has a direct connection to the visual computing field. Visual computing focuses on interpreting information from visual data and converting it into useful information, which this project tries to do. This project revolves around the idea of using visual data to learn visual patterns, which are then processed into useful information that we can use. To be more specific, the visual data obtained from the images are converted into classifications of different traffic signs.

2. Literature Review:

Although these automated driving systems have been rapidly evolving for the past decade, a significant challenge still remains in ensuring safety and reliability during non-ideal weather conditions. The articles we've found have assessed the challenges that autonomous vehicles face in adverse weather while also reviewing various solutions. Zhang et al. (2023), Xu and Sankar (2024), and Wang et al. (2024) each describe how adverse weather conditions hinder the vehicle's ability to detect objects, recognize road features, and navigate safely. They also indicated that while current advancements are capable of solving some of the significant issues, they still haven't been able to solve all of them.

Many of the previous articles mentioned a lack of data for adverse weather in training these models, so to improve our results, we tried finding ways for us to acquire more data with adverse weather conditions. While we could collect this data ourselves, this process would be very time consuming because we would have to collect thousands of photos, properly format them, and then label each one. So we've decided to research the benefits of modifying datasets instead. The articles by Xu et al (2023) and Shorten and Khoshgoftaar (2019) did a survey on image augmentation to find benefits for deep learning, and found that these modifications do help improve the performance of the models. The only issue the articles had found was that not all techniques work for all models, which is reasonable because not all models are alike.

A second method we could potentially employ, if we don't decide to modify the original dataset, is to make more images. In the article written by Jo, Na, & Song (2017), they had found that synthesized images were a great alternative, showing that the overall performance of the datasets were very similar. This means we could use synthesized data to help increase the number of images we have if we lack images with adverse weather effects.

For potential GenAI models we could implement, we've decided to work and explore the implementation of CycleGan, a generative adversarial network architecture introduced by Zhu et al. (2017) and researchers at UC Berkeley. CycleGAN was designed to perform image-to-image translation without requiring paired training examples, a significant innovation in the field of generative models. Traditional image translation models often rely on having a dataset with matching input-output image pairs (e.g., a photo and its corresponding stylized version). However, CycleGAN overcomes this constraint by using unpaired datasets, making it a highly flexible and scalable approach for real-world tasks where paired data is unavailable or difficult to collect.

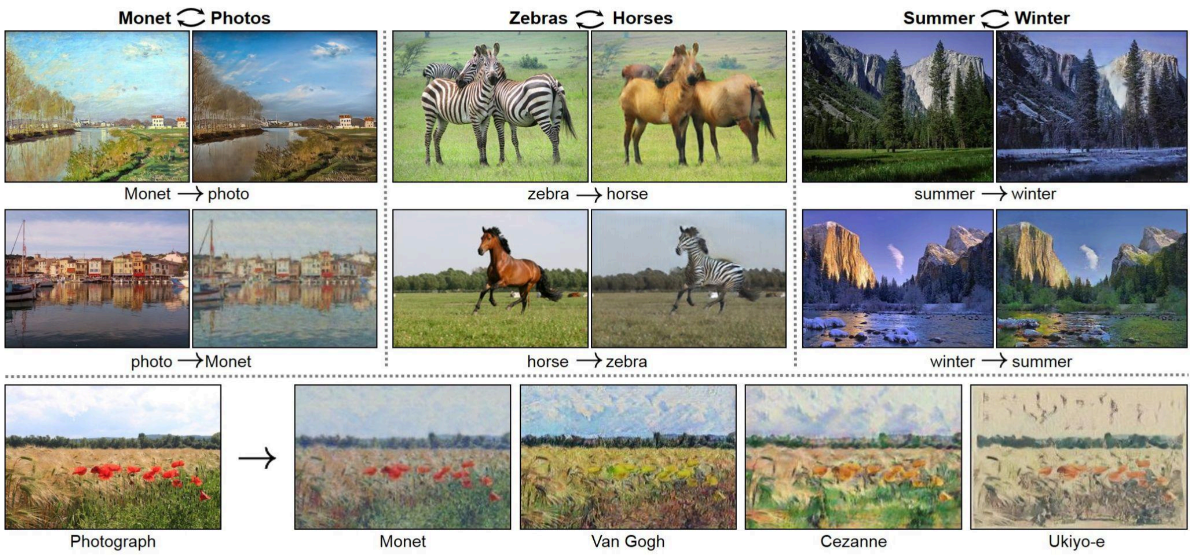
The architecture of CycleGAN consists of two generator networks and two discriminator networks. The generators learn to map images from one domain (e.g., photos) to another (e.g., Monet paintings), and vice versa. The discriminators evaluate how realistic the generated images are in their respective domains. A key innovation in CycleGAN is the cycle-consistency loss, which ensures that if an image is translated to another domain and then translated back, it should closely resemble the original input. This constraint enforces consistency and helps the model preserve structural content during style transfer.

One of the well-known demonstrations of CycleGAN's capabilities includes translating real-world photographs into Monet-style paintings and converting those paintings back into photorealistic images (Zhu et al., 2017). Such examples illustrate the model's ability to capture the stylistic and textural characteristics of one domain and convincingly apply them to another, while retaining the spatial and semantic structure of the original image (see Figure 1, top left).

In the context of our project, we leverage CycleGAN for simulating adverse weather conditions such as snow, rain, and fog. By training the model with our original traffic sign dataset (GTSRB) as the source domain and using weather-specific image datasets (e.g., Snow100K for snow, Rain100 for rain, and Foggy Cityscapes for fog) as the target domain, CycleGAN can learn to generate realistic weather-obscured images. This enables us to simulate real-world driving conditions without needing to manually collect and label thousands of weather-affected images. As such, CycleGAN provides an efficient and scalable solution for data augmentation in environments where traditional image alteration methods may fall short in realism or diversity.

Figure 1

CycleGan Example Image



Zhu, J.-Y., Park, T., & Wang, T. (2017). junyanz/pytorch-CycleGAN-and-pix2pix. GitHub. [*https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix?tab=readme-ov-file*](https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix?tab=readme-ov-file)

3. Methods:

Original Approach A: Transfer Learning on Existing Convolutional Neural Networks

We planned to start by taking a pre-trained CNN model, such as ResNet50, and use it to identify our data. We hope to use Transfer Learning, the process of using an existing trained model to solve a new problem, to get us started. We will experiment with a few existing models, with multiple experiments of varying obscuration, by obscuring 5% of the images per obscuration and by 25% per obscuration. This would give us 25% of the images obscured and the other with 50% of the images obscured. This could give insight into how the models handle difficult data.

Original Approach B: Construction of a New Convolution Neural Network

Alternatively, we considered trying to construct our own simple Convolution Neural Network, if the pre-trained models are not as accurate as we hope. We would use the previously mentioned obscuration experiments, as well as modifying the convolution, pooling, and fully-connected layers to find a sweet spot in analyzing the images. However, due to the success of the pre-trained model, we ended up not actually implementing any additional model, just creating the skeleton of one.

Original Approach C: Analysis of Difficulties

If we were unable to achieve desired accuracy using either methods with Convolution Neural Networks, we planned to look into the difficulties with this particular data. We would have looked into what challenges appeared when trying to engineer these systems, and document our possible shortcomings. We would have also looked into challenges that others in the field have run into when designing their own systems, and what can be done or overcome to fix them. We still planned to do analysis of the project, but this would've been an emergency plan, if the experiment failed.

Dataset Preparation:

To experiment a little further, we decided to work with the ideas proposed from Professor Ingle, and to also work on the original idea we planned. Professor Ingle suggested looking into generative AI. For which to use, we decided to use CycleGan, a model that works to apply effects over existing images. Our dataset now includes, our original dataset, three datasets with modified images simulating weather effects, and a dataset with simulated weather distorsions.

Another method we implemented is making custom functions to simulate weather. Originally planned to leverage albumentations, but it didn't work as our image size is too small (32x32). Instead, we obscured images using custom image transformation functions. As of Check-in #3, we've completed the image modification, and have begun testing the model.

As stated earlier in approach A, we plan to generate the obscured datasets using both CycleGAN and custom method, and plan to use them to run two separate tests. One testing dataset will contain 25% of the obscured images, and another testing set will contain 50% of the obscured images. This will be applied to each category of datasets for each method of generating the augmented images. There will be 25% obscuration rain, fog, and snow dataset from the CycleGAN method, and 25% obscuration rain, fog, and snow from the custom method, and so on for the 50% obscuration. This will hopefully provide us with results that are

The Current Method:

For our project, we stuck with Method A, the Transfer Learning Method on existing CNN models. We generated obscured image datasets using both CycleGAN and custom weather-like distortions (rain, snow, and fog). With the datasets fully prepared, we proceeded to apply transfer learning using the pre-trained ResNet50 model. We trained and evaluated the model on both 25% and 50% obscured datasets to analyze its robustness under varying degrees of visual interference.

4. Results:

Initial Model Performance (Using Original, Unobscured GTSRB Data)

Before applying any obscuration techniques, we successfully trained a ResNet50 model using the original GTSRB dataset. This model, saved as resnet50_model.path, will be used as a base for transfer learning when evaluating the performance on obscured data. The first row of the table below shows the performance metrics of the initial model.

The results that will determine how well the method has worked is based on the accuracy, precision, recall, and the f1-score of the weather/method of distortion. An original baseline of our model was added to show our current progress in this project.

CycleGan Modifications				
Weather / Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Original (Baseline)	99.13%	98.76%	98.76%	98.74%

Rain - 25%	93.29%	94.22%	93.29%	93.50%
Rain - 50%	88.59%	91.03%	88.59%	89.09%
Fog - 25%	75.21%	77.41%	75.21%	75.75%
Fog - 50%	51.39%	56.47%	51.39%	52.67%
Snow - 25%	92.83%	93.73%	92.93%	92.96%
Snow - 50%	87.76%	89.94%	87.76%	88.01%

Weather-Like Distortions				
Weather / Method	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Original (Baseline)	99.13%	98.76%	98.76%	98.74%
Rain - 25%	83.10%	90.12%	83.10%	85.06%
Rain - 50%	65.95%	83.48%	65.95%	69.97%
Fog - 25%	97.69%	97.69%	97.69%	97.69%
Fog - 50%	96.42%	96.61%	96.42%	96.42%
Snow - 25%	81.86%	91.78%	81.86%	85.02%
Snow - 50%	63.75%	86.85%	63.75%	70.25%

We modified our data with CycleGan (Figure 2), as it seems to provide a way to batch modify our existing image set, based on a passed-in trained weather type. We also developed simpler image obscurations using custom functions (Figure 3). For changes that we have made to the project, we are currently testing the effectiveness of the model at classifying the augmented images.

We evaluated the performance of a pretrained ResNet50 model fine-tuned on obscured images using two different methods: CycleGAN and custom weather-like distortions.

- The model maintained high performance on mildly obscured data (25% obscured images), with accuracies above 90% for CycleGAN cases.
- Severe distortions (50% obscured images) led to noticeable drops in accuracy and F1-score, especially with the custom distortions (63.75% accuracy for 50% snow)
- CycleGAN-obscured images generally resulted in better performance compared to the custom weather-like distortion method, particularly for moderate to high distortion levels.
- The model underperformed with CycleGAN fog, and weather-like rain and snow. The three of these sets have distortions that make the edges of shapes more difficult to discern. We

think there might be some relationship between the two, but more testing will have to be done.

- Although the model's accuracy in the CycleGAN modification table does take a hit, it shows that we've successfully synthesized data with weather effects.
- There's a noticeable reduction in the performance of the model with the given modified datasets, mostly affecting the accuracy and recall.
- In the CycleGAN modification table, it appears that when rain obscuration is 50%, the model makes fewer false positives, with the price of missing some true positives (based on how precision is greater than the recall)

Figure 2

Images generated using CycleGan, rain, snow, and fog

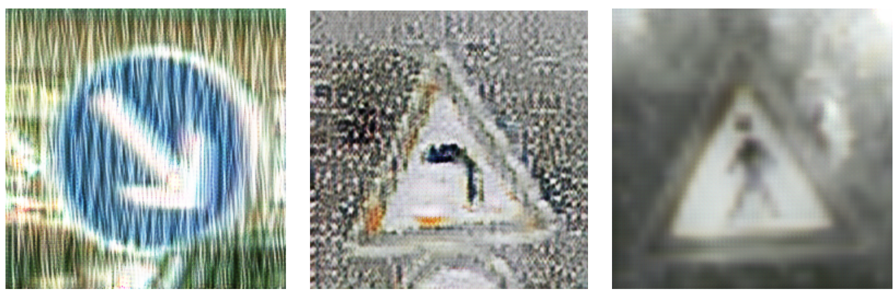


Figure 3

Obscured images generated using custom functions to simulate rain, snow, and fog effects

5. Discussion:

Roadblocks & Limitations:

One hurdle faced by the group was learning how to properly operate the CycleGAN program. Many of the members struggled with it due to technical difficulties such as systems not properly recognizing pytorch libraries, or how certain aspects of the program lacked proper documentation. This is still a current issue because we haven't fully perfected the program to produce perfect image augmentations. While this description is a bit hard to understand, if you simply browse through the augmented files, it becomes clear what we had struggled with.

Besides how CycleGAN was having issues with generating proper augmented images, we faced other limitations such as having difficulties storing the large amount of data we've generated from creating the augmented images. We had to find an alternative method of generating the results needed, without wasting too much time and space.

Strange Observations:

Some strange observations that were seen in the results was how in the custom augmented images, the rain and snow had a much lower expected result for the 50% obscuration than the ones in the CycleGAN. Also, similarly to the rain and snow, the foggy CycleGAN images had lower results, but that score wasn't present for the foggy custom augmented images.

Next Steps:

We have an idea for a potential reason why observations were strange for some of the cases. If we had more time, we would've liked to look more into how the model learns the distinctions between the traffic data. One theory we had was that edges for the shapes in the problem spots, Custom Rain and Snow, and CycleGAN fog, were disrupted with the modifications. The model has learned some key identifiers from the images, and these modifications in particular are messing with it. We would need more time to dig deeper to figure out why.

6. Conclusion:

As we run tests on our trained CNN models with our collected datasets, things seem quite positive. Our starting tests on the unaltered dataset with a pretrained model performed well enough that we proceeded with the Resnet50 Model. As we test the model, we're trying to identify how the accuracy fluctuates between different amounts of modified images within the dataset. We have two versions of the modified dataset, images modified with CycleGan and our own weather-like distortions, and introducing it to the model results in some interesting developments.

As we come to the conclusion of the testing of our Resnet50 model, we are going to spend some time analyzing how well the model performed and what we can do next. We've spent time working on transfer learning, investigating how robust pre-trained models are under increasingly degraded visual input, comparing performance between datasets with 25% and 50% obscuration levels.

As a group, we've decided that one goal for us is to look into these difficulties in the field to try and understand why things are challenging and where the challenges come from. We hope that even if our work doesn't come out with decent results, that we can still hope to learn and have something to say in the process of getting there.

Since our last check-in, we completed generating all obscured datasets, including CycleGAN-based images for rain, fog, and snow. We also conducted transfer learning using both the CycleGAN and our custom weather-like distortion datasets. For CycleGAN, transfer learning was performed on rain and snow obscured images, while for the custom method, we ran training and evaluation across all distortion types and levels.

Our findings suggest that while transfer learning enables robust recognition of traffic signs under mild distortions, model performance significantly degrades under heavy visual interference; particularly with custom distortions. This highlights the importance of incorporating a diverse and realistic augmentation pipeline during training. Moving forward, we aim to analyze these results in more detail and explore potential improvements such as domain adaptation techniques or ensemble models to boost generalization under extreme visual conditions.

7. Contributions:

Max set up a pretrained model to enable transfer learning, introduced the idea to expand the project with this approach, and researched potential models. He modified the dataset using CycleGAN to generate foggy images for testing and provided documentation and updates related to CycleGAN and the overall project progress.

Saeah proposed the project idea of traffic sign image classification, prepared the training and test datasets, and identified class imbalance issues. She initiated the use of CycleGAN for style transfer, implemented weather simulation custom functions to create three condition-specific datasets, and evaluated a pretrained ResNet50 model on obscured GTSRB images using custom image processing, ensuring the robustness testing pipeline worked for all datasets.

Artem explored convolutional neural networks and methods for weather data simulation. He generated a rainy image dataset using CycleGAN and conducted experiments to evaluate model accuracy under these altered conditions. In addition to helping expand the dataset, he contributed research findings and played an active role in writing and reviewing project documentation.

Filipp researched applications of generative AI and contributed to enhancing the project's technical documentation. He created the snow-based CycleGAN dataset used for model testing and participated in team discussions on methodology. Filipp also supported the team by contributing background research and helping refine the project write-up.

All members presented equally, and contributed making the presentation slide. Artem and Max focused more on modifying the final project report, and Saeah focused more on organizing the supplementary and writing README file.

Works Cited

Chougule, A., Chamola, V., Sam, A., Yu, F., Sikdar, B. (2023). A Comprehensive Review on Limitations of Autonomous Driving and its Impact on Accidents and Collisions | IEEE Journals & Magazine | IEEE Xplore. (n.d.). Ieeexplore.ieee.org. https://ieeexplore.ieee.org/abstract/document/10335609

Fu, H., Ye, S., Fu, X., Chen, T., & Zhao, J. (2024). New insights into factors affecting the severity of autonomous vehicle crashes from two sources of AV incident records. Travel Behaviour and Society, 38, 100934. https://doi.org/10.1016/j.tbs.2024.100934

IBM. (2024, February 12). *Transfer Learning*. Ibm.com. https://www.ibm.com/think/topics/transfer-learning

Jo, H., Na, Y.-H., & Song, J.-B. (2017). Data augmentation using synthesized images for object detection. 2017 17th International Conference on Control, Automation and Systems (ICCAS). https://doi.org/10.23919/iccas.2017.8204369

Mo, T., Zheng, S., Chan, W.-Y., & Yang, R. (2025). Review of AI Image Enhancement Techniques for In-Vehicle Vision Systems Under Adverse Weather Conditions. World Electric Vehicle Journal, 16(2), 72. https://doi.org/10.3390/wevj16020072

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0197-0

Wang, J., Wu, Z., Liang, Y., Tang, J., & Chen, H. (2024). Perception Methods for Adverse Weather Based on Vehicle Infrastructure Cooperation System: A Review. Sensors, 24(2), 374–374. https://doi.org/10.3390/s24020374

Xu, C., & Sankar, R. (2024). A Comprehensive Review of Autonomous Driving Algorithms: Tackling Adverse Weather Conditions, Unpredictable Traffic Violations, Blind Spot Monitoring, and Emergency Maneuvers. Algorithms, 17(11), 526–526. https://doi.org/10.3390/a17110526

Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. Pattern Recognition, 137, 109347. https://doi.org/10.1016/j.patcog.2023.109347

Zhang, Y., Carballo, A., Yang, H., & Takeda, K. (2023). Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. ISPRS Journal of Photogrammetry and Remote Sensing, 196(196), 146–177. https://doi.org/10.1016/j.isprsjprs.2022.12.021

Zhu, J.-Y., Park, T., & Wang, T. (2017). junyanz/pytorch-CycleGAN-and-pix2pix. GitHub. https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix?tab=readme-ov-file