Restoration of Degraded Images Captured During Adverse Weather Conditions with Generative Adversarial Networks

Final Project Report

Prasanna Gupta
Department of Automotive Engineering
Clemson University-ICAR
Greenville, SC 29607
hthalla@clemson.edu

Shailendran Poyyamozhi
Department of Automotive Engineering
Clemson University-ICAR
Greenville, SC 29607
spoyyam@clemson.edu

Hesameddin Khosravi
Department of Mechanical Engineering
Clemson University
Clemson, SC 29630
hkhosra@clemson.edu

May 6, 2022

Abstract

Perception operation in autonomous systems enables sensing environments around the system. This is crucial to make decisions, plan, and operate in real-world environments through numerous functionalities and operations from occupancy maps to object detection. To perform these essential operations, autonomous systems use a variety of sensors, namely RADAR, LiDAR, GPS, Camera, etc. Each sensor has its functionality, such as, to name a few, for path planning, GPS, for distance tracking/object detection RADAR/Lidar, for lane-keeping the Camera, etc., are used. These sensors are not responsive in adverse weather. Camera is the only sensor, which has the ability to view the environment. The main objective of present autonomous vehicle development is to use as few sensors as possible to perform perception operations without compromising safety. In this paper, we present an AI-based method named cycle-GAN to clear the adverse weather condition, which will help us perform various perception operations.

Keywords Perception · Autonomous Vehicle · Rainy weather · cycle-GAN

1 Introduction

LiDAR and Camera play a key role in sensing the environment around the system, with fine details, and help the system better understand its surrounding environment. LiDAR generates the point cloud of the surrounding environment providing the depth/distance of the objects from the vehicle. All perceptual-related tasks such as traffic sign and signal detection, lanes detection etc., are handled by the Camera. It is also possible to extract the depth information, like LiDAR, with the help of stereo cameras through correspondence. These two sensors work well under usual driving conditions. But, during adverse weather conditions such as rain, fog, and snow, LiDAR performance degrades drastically and suffers due to a phenomenon called dispersion since it works by bouncing laser beams off the surrounding environment. The quality of images captured by the Camera also degrades during such situations. Hence, the problem is improving the perception accuracy of autonomous vehicles during these adverse weather conditions for safe navigation. The concept of self-driving cars equipped with a superior level of safety came into practice in the 20th century. In 2009, when Google announced to start their research in self-driving, the concept became majorly popular. In 2015, Tesla began to commercialize 'Autopilot' features in its cars and soon became one of the top autonomous vehicle companies in the world. However, this was not just limited to automakers. Tech companies and service providers also joined the competition. Currently, several major companies are engaging in the research and development of both semi- and fully autonomous vehicles. Several automobile giants have announced their plans to launch fully autonomous cars during the last three years by 2020. The growing interest of individuals and corporations in this domain will take this industry to new heights soon. Seeing that interest and potential, we conducted a thorough market analysis of the autonomous vehicle industry. Ongoing advances in sensor technology are integral to increasing road safety. Data from these sensors is crucial for the autonomous systems to warn of any possible safety risks and intervene to prevent accidents. Some vehicles combine Autonomous vehicles systems with the infotainment system to provide more reliable data of a view of surroundings. Autonomous vehicles systems rely on embedded systems that include Cameras, Radio detection and ranging (RADAR), Light detection and ranging (LiDAR), and Ultrasonic transducers to gather valuable data for safe navigation. The challenge is to interpret the data in real-time from embedded vision to enable the system to intervene. Autonomous vehicles need higher computing power and memory to process this valuable data. Recently, state-of-the-art advanced technologies have been developed for the purpose of vehicle perception. These systems use GPU, FPGA, or custom ASIC within the electronic control units (ECUs) to efficiently handle perception tasks of an autonomous vehicle in real-time.

Initially, Ultrasonic sensors were used to detect the distance of the surrounding objects from the vehicle, but this information was not sufficient for safe navigation. To get more information such as depth information and perceptual features about these surrounding objects, additional sensors such as Camera, RADAR and LiDAR are being used for the current perception task. Further, sensor fusion technologies such as Kalman filter (KF), Extended Kalman filter (EKF), and Particle Filter are being used to increase the accuracy of captured information.

The main objective of present autonomous companies is to use as few sensors as possible to perform perception operations without compromising safety. Apart from Ultrasonic transducers and RADAR, autonomous driving giant, Tesla, relies only on cameras to see around the environment. Other self-driving vehicle companies such as Alphabet Inc.'s Waymo and GM's Cruise also use LiDAR along with Camera for this purpose. The objective of these companies is to produce a higher resolution of perception data using numerous sensors, which help the system navigate in all conditions. As companies like Tesla, John Dheere etc., generate higher resolution data using Cameras (stereo) and RADAR alone, these systems are cost-efficient compared to other self-driving companies that use LiDAR, which is expensive.

2 Camera-based perception of autonomous vehicles

During adverse weather conditions such as foggy, snowy, and rainy conditions, the performance of LiDAR and Camera degrades. But unlike LiDAR, it is possible to restore the degraded images captured by the Camera with the help of low-level image processing techniques such as image enhancement and restoration. Cameras operate on the basic principle of directing light emitted by objects onto a photosensitive surface via a lens. The light is stored as an image on the photosensitive surface, also known as the image plane. This process has the ability to gather a coloured feature of the environment which is used for vision-based processing to enable proper and safe mobility of the vehicle.

During Adverse conditions, this image plane created using a camera contains less resolution and huge noise. While advanced methods have improved recognition techniques, slight variations in weather still influence camera measurements. The Camera is susceptible when faced with adverse climatic conditions. The visibility of the Camera in an aerosol environment experience gets decreased, leading to unreliable object recognition. A camera is not recommended for environmental detection and vehicle control tasks under foggy conditions. Even though weather conditions have a significant impact on intelligent navigation perception systems, most AV research focuses on the efficiency of algorithms without taking climatic conditions into account (such as snow, sleet, rain, and fog). The ability to perceive the environment autonomously and robustly in all weather conditions has not been adequately considered. Some of the significant limitations of a self-driving vehicle's Camera during adverse conditions are:

2.1 Current industrial approaches for the perception of vehicles

The main objective of present autonomous companies is to use as few sensors as possible to perform perception operations without compromising safety. Apart from Ultrasonic transducers and RADAR, autonomous driving giant, Tesla, relies only on cameras to see around the environment. Other self-driving vehicle companies such as Alphabet Inc.'s Waymo and GM's Cruise also use LiDAR along with Camera for this purpose. The objective of these companies is to produce a higher resolution of perception data using numerous sensors, which help the system navigate in all conditions. As companies like Tesla, John Dheere etc., generate higher resolution data using Cameras (stereo) and RADAR alone, these systems are cost-efficient compared to other self-driving companies that use LiDAR, which is

Figure 1: Camera output from a self-driving vehicle during adverse condition

expensive.

2.2 Factors affecting the camera-based perception

During adverse weather conditions such as foggy, snowy, and rainy conditions, the performance of LiDAR and Camera degrades. But unlike LiDAR, it is possible to restore the degraded images captured by the Camera with the help of low-level image processing techniques such as image enhancement and restoration. Cameras operate on the basic principle of directing light emitted by objects onto a photosensitive surface via a lens. The light is stored as an image on the photosensitive surface, also known as the image plane. This process has the ability to gather a coloured feature of the environment which is used for vision-based processing to enable proper and safe mobility of the vehicle.

During Adverse conditions, this image plane created using a camera contains less resolution and huge noise. While advanced methods have improved recognition techniques, slight variations in weather still influence camera measurements. The Camera is susceptible when faced with adverse climatic conditions. The visibility of the Camera in an aerosol environment experience gets decreased, leading to unreliable object recognition. A camera is not recommended for environmental detection and vehicle control tasks under foggy conditions. Even though weather conditions have a significant impact on intelligent navigation perception systems, most AV research focuses on the efficiency of algorithms without taking climatic conditions into account (such as snow, sleet, rain, and fog). The ability to perceive the environment autonomously and robustly in all weather conditions has not been adequately considered. Zang et al. [2019] mentions some of the significant limitations of a self-driving vehicle's Camera during adverse conditions which are:

- Rain and snow conditions introduce sharp intensity fluctuations in images. This condition changes the image intensity and blurs the edges of various objects in an image. Usually, rain reduces the image intensity, and heavy snow increases the image intensity.
- Fog condition reduces the image's contrast and increases the difficulty of pattern edge recognition.
- In the case of an observed fog scene, the frequency components are concentrated at zero frequency, whereas in the absence of fog, the spectrum is widely spread. Sharp edges are defined by low and high frequencies, whereas smooth edges are determined solely by low frequencies.

This limitation has caused numerous accidents when it comes to level 2 or level 3 self-driving techniques. In September 2016, a Tesla collided with a truck on a freeway in China because of unclear image quality due to adverse weather conditions Chen et al. [2021]. This limitation of image quality produces a degraded image. There are some significant factors that these degraded images make,

- **Salt and Pepper Noise:** Salt-and-pepper noise is a type of noise that can be seen on images. It is also referred to as impulse noise. Sharp and sudden disturbances in the image signal can cause this noise. It appears as sparsely distributed white and black pixels.
- **Bilateral Image:** These are images that contain huge noise and are non-linear. This image also shows the edges between various objects of the image. But the edge also includes the noise.

Figure 2: left picture is the image captured during rain and right is the restored image with algebraic methods

- **Saturation Image:** The intensity of a color is described by its saturation. A gray-scale or black-and-white image has no colour saturation, whereas a full-color image of a field of bright wildflowers may be highly saturated.

To resolve this limitation, single or multiple thresholds, filtering, and extraction operations are performed on an image to obtain a better image output. Image efficiency can be influenced by a minor miscalculation, which machine-based learning methods and processing algorithms can address. Nonetheless, to achieve optimal results using machine learning methods, massive training data-sets must be collected, and the model must be trained for an extended period of time.

2.3 Non-AI based Methods for image restoration

Traditionally, three types of non-AI based image restoration methods Kodieswari et al. [2021] have been in use for this purpose, which work based on algebraic methods through the application of filters such as:

- **Inverse filter**: This type of restoration method is used when we have complete knowledge about the kind of blurring function/ filter leading to the degraded image.
- **Weiner filter**: This method is used if we only have partial knowledge about the blurring function/ filter leading to the degraded image.
- **Blind restoration filter**: This method is used if we do not have any prior knowledge about the blurring function/ filter causing image degradation etc.

The usage of algebraic methods for image restoration has limitations in terms of applications and accuracy, as shown in Figure 2, and cannot be used in a dynamic scenario such as in varying weather conditions. Hence, there is a need for intelligent/AI (Artificial Intelligence) based solutions to provide the best results in such dynamic conditions.

2.4 AI based Methods for image restoration

AI-based image restoration methods, such as the usage of Convolutional Neural Networks (CNNs), have resulted in better results in terms of quality and accuracy of the restored images, as shown in Figure 3 in comparison with image restoration using algebraic methods. There are many types of neural networks that are based on CNNs which have been developed over time for restoring images, such as:

- **CNN**: Plain convolutional neural networks for dirt or rain removalEigen et al. [2013].
- **DehazeNet**: Deep network for rain removalWang et al. [2021].
- **SPANet**: Spatial attentive network for rain removalPark et al. [2022].

3 Cycle-GAN as a new solution to perception improvement

As part of this project, we are planning to use AI techniques for the purpose of image restoration, i.e., using Convolutional Neural Networks (CNNs). And we aim to use a specific type of CNN which is Generative Adversarial Networks

Figure 3: Performance comparison between existing image restoration methodologies; left: DehazeNet; middle: Input image; right: Plain CNN

(GANs), for restoring the images that are degraded by rain, fog, and snow, as shown in Figure 4. because of the following advantages over other types of CNNs Isola et al. [2017]

- GANs are comparatively faster.
- These generate more realistic and high-resolution images.
- GANs are known for being good data synthesizers, i.e., these can generate data. Hence, they are not as data-hungry as other CNN types while training.
- GANs train through unsupervised learning, i.e., no data labelling is required for the input images

3.1 GAN related work in autonomous driving

Cycle GAN-based augmentation is used in the self-driving domain during the night for perception segmentation Ostankovich et al. [2020]. This study defines the perception module as a combination of object detection and road segmentation submodules. Here, GAN-based augmentation is used as a critical factor to improve the performance of perception. At night, the primary attention is on detection and segmentation issues. The initial training data comprises the BDD100K dataset, which was acquired in the winter using the front-view camera of an Innopolis University-developed self-driving car. The collected findings suggest that the segmentation task improved when Cycle GAN augmentation was used. However, due to visible artifacts, the chosen method of GAN-based augmentation has not positively impacted object detection.

Cycle-GAN is used for low illumination enhancement for object detection in self-driving applications Qu et al. [2019]. Object detection is crucial in the realm of self-driving cars. Although illumination has a significant impact on object detection, most current approaches do not adequately address the problem of object detection in low-light environments. A cycle GAN-based technique was used to optimize the image conversion. The suggested method was tested using the Oxford University Robot Car dataset. The approach can significantly enhance detection accuracy and increase the number of detected objects in low-light environments.

Under Inconsistent Illumination Conditions in Real Environment, Improved Deep Learning-Based Object Detection of Mobile Robot Vision by HSI Pre-processing Method and Cycle-GAN Method is used Wang and Chuan Tan [2019]. The goal of the family mobile robot is to deliver more intelligent services to humans in their homes. Each sensor has its own set of capabilities, with which the camera may be utilized for object identification, recognition, tracking, and other tasks. The effect of object detection is easily influenced by the outside world, such as uneven lighting, object occlusion, etc. In order to increase the detecting impact of robot vision in a dark environment, a unique approach of brightness migration based on Cycle-GAN is used. Compared to the HSI pre-processing approach, the brightness migration method based on Cycle-GAN may achieve a superior detection result in robot vision.

3.2 Dataset selection and challenges

GAN has a considerable capability when it comes to image processing and application. The most crucial factor in image data selection is the feature similarity while training the GAN. Feature similarity is the issue that the image data and the label image data should have similar features. If both data are taken using the same camera in the exact vehicle and similar locations, it overcomes the feature similarity issue. Feature similarity is observed in all the famous datasets like DAWNKENK [2020], CityscapesGarcia-Garcia et al. [2017], and ADUULM Dataset Pfeuffer et al. [2020]. These issues are addressed in Figure 4 below.

Figure 4: Various dataset images

As you can see from the figure 1image above, we have issues like, Car reflection of the windshield, Car hood, Car dashboard, and Weather color change. These issues create a noisy and clustered image in the final image of the GAN. Based on our research, we have chosen the radiate dataset because it solves the issue of feature similarity. Figure 5 shows an example of the radiate dataset Sheeny et al. [2020].

We take two types of data for our use; 1) Ground truth data which is basically self-driving data with no adverse conditions and noise. This data is considered with high accuracy and good features. 2) Adverse data which is appropriate with adverse and noise features for self-driving conditions. This is shown in Figure 6,

3.3 Cycle-GAN structure

There are a lot of GANs available, and a proper form of GAN needs to be chosen. This is determined based on the dataset form. The dataset may have similar features, but the images are not paired with the indices. The paired images and unpaired images are shown in Figure 7.

Paired images in self-driving applications are the set of two images where the location is the same with the same sensor and vehicle. Unpaired images use the exact vehicle and sensor where the locations are different, but the location features are the same as shown in Figure 7 above. For unpaired images, cycle-GAN Zhu et al. [2017] is preferred because of the ability to generate an image by using two generators and discriminators. The Cycle-GAN promotes cycle consistency by including an extra loss that measures the difference between the produced output of the second generator and the original picture, as well as the opposite. The cycle-GAN is explained below in Figure 8.

- **Generator A** basically takes the rainy image as an input and produces a clear weather image as an output.
- **Discriminator A** discriminates weather the output from generator A is a clear image or not.
- **Generator B** basically takes the clear weather image as an input and produces a rainy weather image as an output.
- **Discriminator B** discriminates weather the output from generator B is a rainy image or not.
- **Error parts**: The error of the rainy input image of generator A and the rainy output image of generator B is calculated. Based on this error, all the generators and discriminators are trained.

The above mentioned is just an overview of how a cycleGAN works. But a simple cycleGAN does have many components involved, including the loss components. These loss components are explained below.

- **Cycle Consistency loss**: A cycle in cycle-GAN means the image input is given in a model, and the same image is formed and compared with the input image. This is divided into two parts: forward and backward

Figure 5: Radiate dataset

Figure 6: left: adverse weather data; right: ground truth clear weather data

Figure 7: top: paired dataset with label; bottom: unpaired dataset

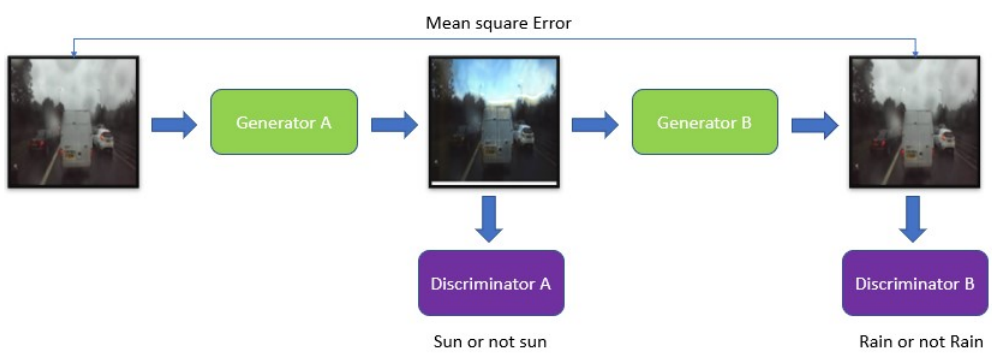

Figure 8: Cycle-GAN

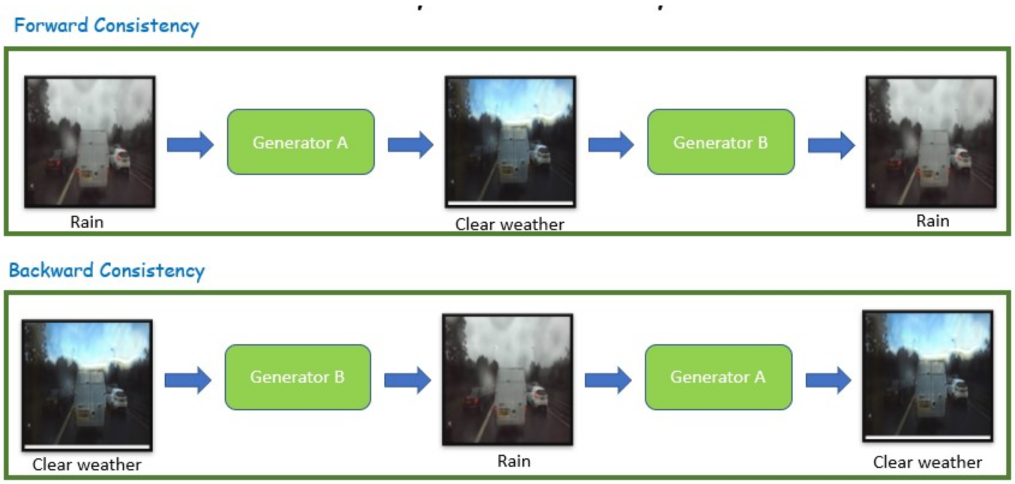

Figure 9: Cycle consistency losses

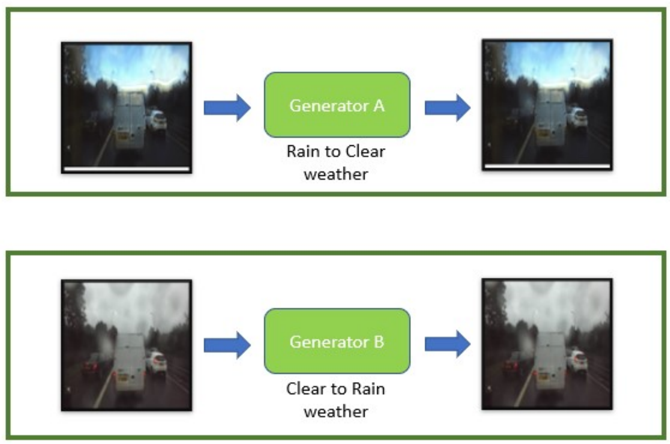

Figure 10: Identity loss

cycle consistency loss. Forward cycle consistency loss is based on the rainy image where the flow begins from generator A to generator B. Reverse cycle consistency loss is based on the clear weather image where the flow starts from generator B to generator A. MAE is used for computing this loss. This is shown in Figure 9.

- **Identity mapping loss**: Generator A converts a rainy image to a clear weather image. Generator A should have the ability to produce a clear weather image if a clear weather image is given as an input and similarly for the generator B, if rainy image is the input then the output should also be the rainy image. This is the identity mapping. This loss needs to be looked into for proper cycle-GAN training. MAE is used for computing this loss. This is shown in Figure 10.
- **Adversarial loss**: This is basic GAN loss which is basically trained based on the discriminator output where the discriminator input is from a generator output. Based on discriminator loss, the weights are tuned. MSE is used for computing this loss.

The model is basically a combination of 2 generators and discriminators. This section is going to be an explanation of each generator and discriminator. The discriminator used in our cycle-GAN is a 70 x 70 patch-GAN discriminator shown in Figure 11. Patch-GAN is a GAN discriminator that penalizes structure solely at the size of local picture patches. The Patch-GAN discriminator attempts to determine if each N × N patch in an image is real or fake. This discriminator is applied convolutionally to the picture, averaging all responses to generate the final output of the discriminator. A discriminator of this type represents the picture as a Markov random field, assuming that pixels separated by more than a patch diameter are independent. It might be seen as a lack of texture or style. The image

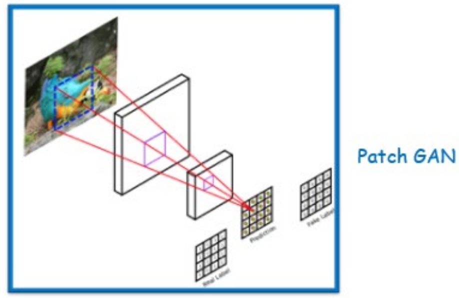

Figure 11: Patch GAN

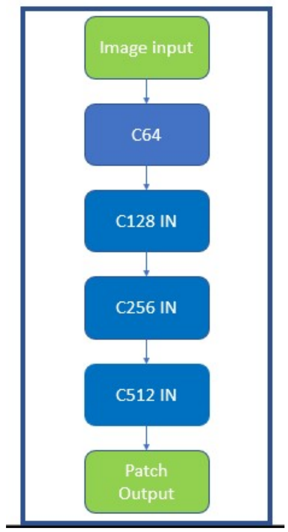

Figure 12: Discriminator model

shown below is a discriminator model in Figure 12. The discriminator is trained separately, not with the composite model.

The IN represented in each layer of the model in Figure 12 represents the instance normalization. The main difference between instance and batch normalization is that batch normalization uses the mean and variance to normalize a whole batch. Whereas, Instance normalization standardizes value on each output feature map rather than across features in a batch. This is shown in Figure 13. The generator model is shown in Figure 14 below. The model shows downscaling layers, which are represented in uN, and some upscaling layers, which are represented in dN. The main important block in the generator model is the Resnet block. Resent block is also called the residual block. The primary purpose of using a Resnet is to overcome the vanishing gradient problem in deep CNN. Vanishing gradient is an issue where derivatives become very small like weights during back propagation, making the training complex. Resnet is based on the idea of skip connections, where information from the top is concatenated with the output of the layers. This basically helps forward propagation faster. This is shown in figure 11. The generator is trained in the composite model, which is a combination of generator and discriminator.

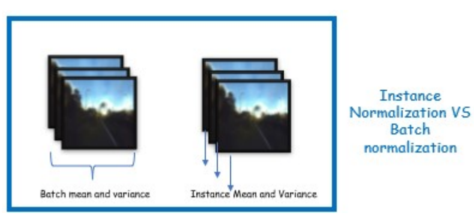

Figure 13: Instance vs Batch Normalization

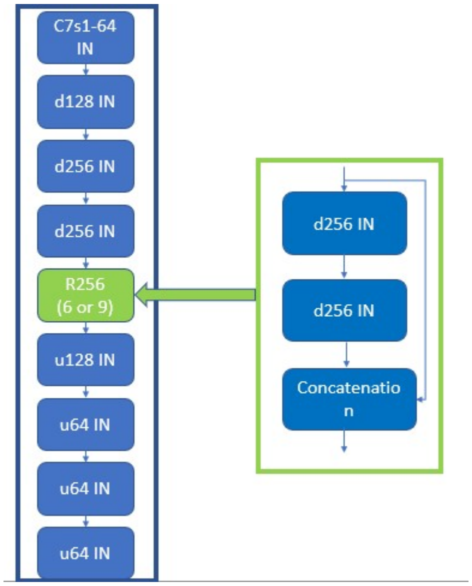

Figure 14: Generator model

3.3.1 Implementation issues and hardware configurations

In order to implement the cycle-GAN, we use the TensorFlow library to create and train the model. Mainly all models are created using a sequential model. Here we are going to use a functional API model for both the discriminator and generator models. Each composite model contains two generators and one discriminator. While training the composite model, only one generator is trainable, and the other generator and discriminator are non-trainable. The solver used here is an Adam solver. The most important part is that the discriminator has a buffer of 50 previously created images. This is done to avoid model oscillations by storing a history of generated images rather than the ones produced by the latest generators. In order to train each composite model, we used the following hardware setup:

- The number of CPU cores used was 12
- The Ram used was 300GB
- The GPU used was V100
- The simulation time is around 48 hours.

The simulation time was 70 hours for the initial runs, but we managed to get the output within 48 hours as we fine-tuned the network.

4 Results

We basically did a lot of iterations during the creation of the model, which basically clears the adverse weather image. But we have added some essential iterations that helped us make a massive difference in image generation using the cycle-GAN.

Iteration 1: This iteration is basically varying the resolution of the image on a base model. The image shapes of resolution used are 256 x 256 and 512 x 512. The results were similar, and the output was very similar in both cases. In both the configurations, the network was able to filter out the noise (droplets) from the rainy image when there are no vehicles in the Figure 15.
But the network is also removing the vehicles from the image as shared Figure 16. Even though the output is not complete, we observed that the results were comparatively faster with the 512 configuration of image shape resolution. Hence, for future iterations we have considered the input image shape as 512 x 512.

Figure 15: Rainy and restored images during iteration 1 with no vehicles in the images

Figure 16: Rainy and restored images during iteration 1 with vehicles in the images

Figure 17: Rainy and restored image during iteration 2

Iteration 2:This iteration is basically varying the loss weights of the composite model. The base configuration used in the above iteration was with weights as Adversarial loss: 1, identity loss: 5, forward consistency loss: 10, and backward consistency loss: 10. The configurations used in this iteration are:

- Adversarial loss: 1, identity loss: 1, forward consistency loss: 1, and backward consistency loss: 1
- Adversarial loss: 1, identity loss: 3, forward consistency loss: 3, and backward consistency loss: 3

The results were actually poor regarding network efficiency as shown in Figure 17, and the model was not filtering the noise even from the images without vehicles and only droplets from rain as shown below. Hence, the original weights from iteration 1 have been used for the future models.

Iteration 3: This iteration is basically tuning the generator factors. This mainly involves increasing and decreasing the layers of the resent layers and the filter size of each layer. The base filter size used was 3 x 3. The filter sizes which we used are: 5 x 5 and 7 x 7. The results were almost similar with both the filter sizes but there was an improvement(shown in Figure 18). We have observed that the network has now stopped masking the vehicles along with the droplets if there are no more than two vehicles in the image as below. Hence, for the future models we have considered the filter sizes as 5 x 5.

Iteration 4: Since the major issue has been solved from iteration 3 up to some extent, at least, we thought that maybe the discriminator was not good enough to discriminate vehicles as real and masking the vehicles as well with the droplets. Hence, in our final iteration we tried to fine tune the discriminator by increasing the number of convolution layers and trained the network with below configuration 5. The results have significantly improved. The model was able to discriminate between droplets and vehicles. This model also stopped masking vehicles with droplets, in most of the images as Figure 19.

In the Figure 20, the losses during the training (120,000 iterations total) of the cycle GAN have been plotted. In the first subplot, the discriminator 1 losses for real images (dA_{loss1}) and fake images (dA_{loss2}) have been plotted and similarly for discriminator 2 in the second subplot. In the third subplot, the generator losses for rainy to clear image (g_{loss1}) and clear to rainy image (g_{loss2}) have been plotted. From the plot it can be observed that the discriminators' loss as well as the generators' loss has decreased significantly over time with the iterations.

5 Conclusion

Even though in our final iteration we got significant results, the quality of the image generated is not very good and can be improved. In case of images with big droplets, the image restoration is getting failed and the network is unable to restore the image completely. We have also observed that in the image if the droplets are on the vehicles, the network is not clearing the droplets completely. To make this network more efficient, we may need diverse dataset with different vehicles for the generator to generate the vehicles completely that have been occluded by the droplets. Finally, we have observed that the cloudy sky with dark clouds has turned into clear sky when restored and there is a slight addition of fake realistic trees in some images.

Figure 18: Rainy and restored images during iteration 1 with no vehicles in the images

Figure 19: Rainy and restored images during iteration 4 with vehicles

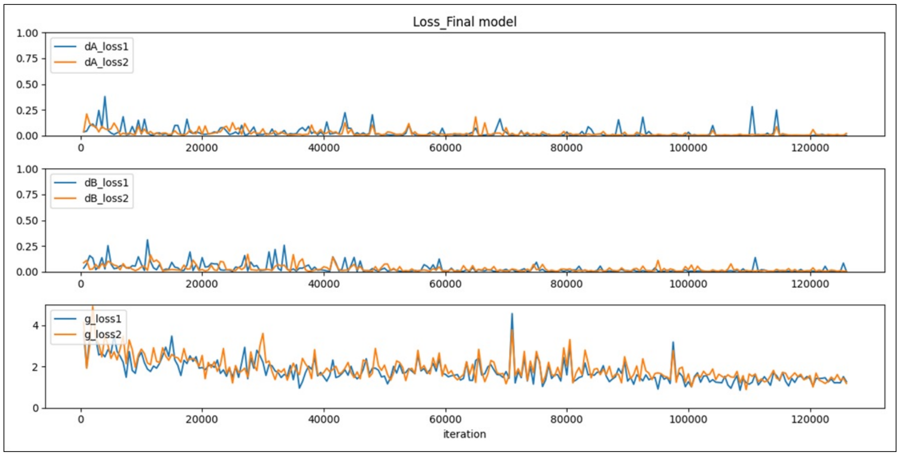

Figure 20: Discriminator and generator losses while training

References

Shizhe Zang, Ming Ding, David Smith, Paul Tyler, Thierry Rakotoarivelo, and Mohamed Ali Kaafar. The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE Vehicular Technology Magazine*, 14(2):103–111, 2019. doi:10.1109/MVT.2019.2892497.

Wei-Ting Chen, Hao-Yu Fang, Cheng-Lin Hsieh, Cheng-Che Tsai, I-Hsiang Chen, Jian-Jiun Ding, and Sy-Yen Kuo. All snow removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4176–4185, 2021. doi:10.1109/ICCV48922.2021.00416.

A. Kodieswari, V. Parameshwari, and S. Sruthi. Reduction of rain and snow within the image using image processing. 2021.

David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. *2013 IEEE International Conference on Computer Vision*, pages 633–640, 2013.

Keping Wang, Yumeng Duan, and Yi Yang. Single image dehazing algorithm based on pyramid mutil-scale transposed convolutional network. *Systems Science & Control Engineering*, 9(sup1):150–160, 2021. doi:10.1080/21642583.2020.1833780. URL `https://doi.org/10.1080/21642583.2020.1833780`.

Yeachan Park, Myeongho Jeon, Junho Lee, and Myungjoo Kang. Mcw-net: Single image deraining with multi-level connections and wide regional non-local blocks. *Signal Processing: Image Communication*, 105:116701, 2022. ISSN 0923-5965. doi:https://doi.org/10.1016/j.image.2022.116701. URL `https://www.sciencedirect.com/science/article/pii/S0923596522000431`.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

Vladislav Ostankovich, Rauf Yagfarov, Maksim Rassabin, and Salimzhan Gafurov. Application of cyclegan-based augmentation for autonomous driving at night. In *2020 International Conference Nonlinearity, Information and Robotics (NIR)*, pages 1–5, 2020. doi:10.1109/NIR50484.2020.9290218.

Yangyang Qu, Yongsheng Ou, and Rong Xiong. Low illumination enhancement for object detection in self-driving. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1738–1743, 2019. doi:10.1109/ROBIO49542.2019.8961471.

Feng Wang and Jeffrey Too Chuan Tan. Improving deep learning based object detection of mobile robot vision by hsi preprocessing method and cyclegan method under inconsistent illumination conditions in real environment. In *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 583–588, 2019. doi:10.1109/AIM.2019.8868399.

Mourad KENK. Dawn, 2020. URL `https://data.mendeley.com/datasets/766ygrbt8y/3`.

Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation, 2017. URL `https://arxiv.org/abs/1704.06857`.

Andreas Pfeuffer, Markus Schön, Carsten Ditzel, and Klaus C. J. Dietmayer. The aduulm-dataset - a semantic segmentation dataset for sensor fusion. In *BMVC*, 2020.

Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. Radiate: A radar dataset for automotive perception in bad weather, 2020. URL `https://arxiv.org/abs/2010.09076`.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017. URL `https://arxiv.org/abs/1703.10593`.